Generalizable Local Feature Pre-training for Deformable Shape Analysis

Souhaib Attaiki   Lei Li   Maks Ovsjanikov

LIX, École Polytechnique, IP Paris

Abstract

Transfer learning is fundamental for addressing problems in settings with little training data. While several transfer learning approaches have been proposed in 3D, unfortunately, these solutions typically operate on an entire 3D object or even scene-level and thus, as we show, fail to generalize to new classes, such as deformable organic shapes. In addition, there is currently a lack of understanding of what makes pre-trained features transferable across significantly different 3D shape categories. In this paper, we make a step toward addressing these challenges. First, we analyze the link between feature locality and transferability in tasks involving deformable 3D objects, while also comparing different backbones and losses for local feature pre-training. We observe that with proper training, learned features can be useful in such tasks, but, crucially, only with an appropriate choice of the receptive field size. We then propose a differentiable method for optimizing the receptive field within 3D transfer learning. Jointly, this leads to the first learnable features that can successfully generalize to unseen classes of 3D shapes such as humans and animals. Our extensive experiments show that this approach leads to state-of-the-art results on several downstream tasks such as segmentation, shape correspondence, and classification. Our code is available at `https://github.com/pvnieo/vader`.

1. Introduction

Extracting informative representations from 3D geometry is a central task in Computer Vision, Computer Graphics, and related fields. Classical approaches have relied on hand-crafted features derived from basic geometric principles [9, 11, 55, 87, 100]. More recently, the focus has shifted towards data-driven approaches that learn features directly from 3D data [17, 20, 46] in a task-specific manner.

In addition to methods that learn features from scratch for each application, several recent works have also advocated for general-purpose *representation learning* on geometric data [51, 103, 110]. Inspired by the success of transfer learning in other domains [117], these methods aim to learn
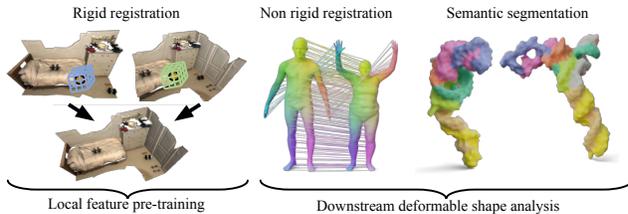

Figure 1. We present VADER, a novel feature pre-training technique aiming for deformable shapes. By pre-training *local* feature extractors on 3D scenes for rigid alignment, our approach enables transfer learning to downstream deformable shape analysis tasks, such as shape matching and semantic segmentation.

informative representations of 3D data, which can then be exploited in data-limited downstream tasks.

Despite this progress, state-of-the-art architectures in *deformable shape analysis* still either rely on classical hand-crafted features as input signals to their learning pipelines [70, 81, 88, 99], or are trained *from scratch* for each task [30, 42, 67], thus requiring significant amounts of labeled data. Unfortunately, as we demonstrate in our work, existing 3D representation learning approaches fail to provide a useful signal in tasks that involve highly deformable shapes, such as shape correspondence or segmentation.

This result is perhaps expected since existing approaches have primarily focused on transfer learning across man-made 3D objects or scenes [109], and are typically restricted to settings with significant domain overlap between training and test data. Furthermore, there is currently a lack of understanding of what makes pre-trained features transferable, especially across significantly different shape classes.

In this work, we aim to investigate the transferability of geometric features to develop representation learning approaches that are useful in downstream deformable shape analysis tasks, such as non-rigid shape matching and semantic segmentation (see Fig. 1). Taking inspiration from recent studies that emphasize the importance of low and mid-level features in enabling 2D transfer learning [78, 116], we explore the impact of feature locality on downstream task accuracy across significantly different 3D shape categories. Our study shows that, with a carefully chosen architecture, successful general-purpose represen-
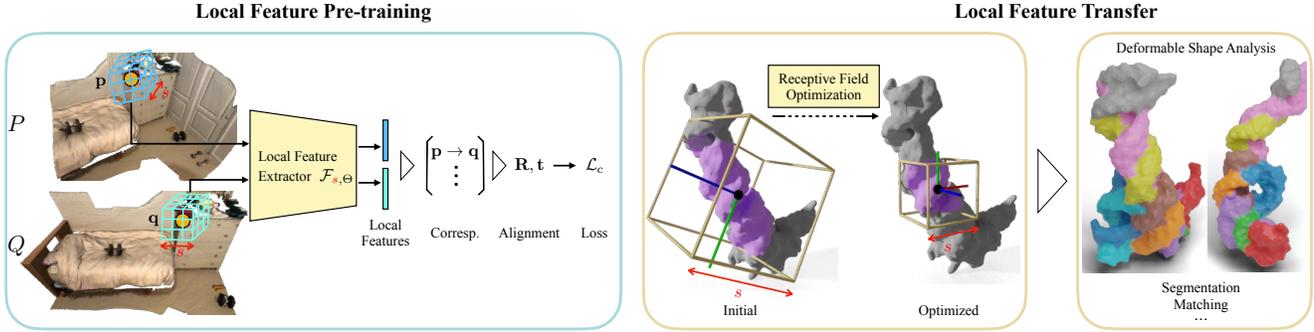
Local Feature Pre-training

Local Feature Transfer

Deformable Shape Analysis

Figure 2. **Method overview**. We propose generalizable *local feature* pre-training for deformable shape analysis. We first pre-train a local feature extractor $\mathcal{F}_{s,\Theta}$, which has a learnable receptive field size s and network parameters Θ, on a pretext task of matching local features for 3D alignment. We then propose a differentiable method for optimizing the receptive field size s to transfer $\mathcal{F}_{s,\Theta}$ to downstream tasks. For illustration purposes, we use a molecular surface segmentation task as an example on the right.

tation learning for deformable 3D shape analysis is possible. We also find that the receptive field (or local support) size plays a crucial role in the transferability of features and needs to be adapted between training and test data. To address this, we propose a receptive field optimization strategy, which, combined with a specific pre-training approach, leads to state-of-the-art results on a wide range of downstream tasks. An overview of our proposed method can be found in Fig. 2.

To summarize, our main contributions are as follows:

1. We investigate the link between the locality of geometric (3D) features and their transferability in challenging deformable shape tasks.

2. We build upon the investigation and propose a novel method for optimizing the receptive field size of local features in the context of transfer learning in 3D tasks. We demonstrate that this optimization brings significant improvement and allows pre-trainable features to generalize well to unseen data in downstream tasks.

3. We show that pre-training local features with an unsupervised cycle consistency loss outperforms the standard contrastive PointInfoNCE loss.

4. Based on all of these insights, we propose a new local feature pre-training mechanism and show its utility in a wide range of tasks involving deformable objects, going beyond man-made objects or scenes considered in previous 3D transfer learning approaches.

2. Related Work

Extracting robust local features is a key problem in 3D shape analysis, and a full review is beyond the scope of this paper. Below we discuss methods most closely related to ours and refer the readers to both early [16, 44, 49] and more recent surveys including [17, 20, 45, 46] for a comprehensive overview of local feature extraction and geometric

deep learning more broadly.

Hand-crafted features. Early efforts in designing informative features for 3D geometry have focused primarily on either ensuring invariance to rigid motion [55, 59, 60, 101, 114] or intrinsic features invariant to isometries. Intrinsic descriptors are typically based on either analysis of geodesic distances [38, 50] or derived quantities such as spectral properties arising from the eigenbases of the Laplace-Beltrami operator [9, 18, 100]. Intrinsic descriptors, such as the HKS [100] and WKS [9], are a very popular choice in deformable shape correspondence methods, especially based on the functional maps framework, e.g., [1, 19, 33, 54, 82, 92, 95], among many others.

Learning for deformable shape matching. To overcome the limitations of hand-crafted features, more recent approaches have tried to learn descriptors directly from deformable shape data [6, 15, 72, 77, 88, 108]. Remarkably, however, many learning-based works still use hand-crafted features (most commonly, SHOT, HKS, WKS, or similar) as input to their learning pipelines [32, 69, 70, 76, 96, 99, 104]. Several recent works [5, 7, 30, 35, 42, 67, 75] investigate learning robust deformable shape correspondence directly from raw geometry, either by exploiting extensive training sets or combining spatial and spectral regularization [30, 32]. Nevertheless, the features learned in these works are typically application and dataset-specific and fail to generalize to new shape classes and shape processing tasks.

Learning for man-made shape matching. A parallel line of studies has focused on learning local geometric features for man-made object or scene alignment. Many efforts have been made to explore different representations for local 3D geometry [29, 41, 52, 56, 66, 115]. With the advancement of 3D deep learning techniques, networks based on PointNet [4, 105, 106, 112, 113], sparse convolution [24, 26], or kernel point convolution [10] have been applied to dense

feature extraction. However, these networks have very limited generalization ability even across 3D scene datasets, such as from indoor scenes to outdoor scans [10], due to local features being coupled with global scene structures.

Feature pretraining. Most closely related to ours are the recent PointContrast [110] and its follow-up works [23, 51, 103] that explore the potential of learning informative representations for 3D data, which can then be leveraged in downstream tasks. In contrast to these works, which focus on man-made objects or scenes, we consider generalizable feature pre-training for *deformable shape analysis*, and study generalizability across significantly different 3D shape categories. Most importantly, our work highlights the impact of feature locality on transferability, which is lacking in prior works.

3. Motivation

Our main objective is to build a general-purpose feature extractor that can be applied to highly deformable shape analysis tasks, such as matching human shapes or segmentation of molecular surfaces, among others. We first examine existing designs of geometric feature pre-training and perform a pilot study to understand their generalization power in such tasks. Our key insight is that the *locality* of geometric features plays a crucial role in their transferability across different categories, which has so far been overlooked in prior works. To this end, we first perform an in-depth analysis of feature locality versus transferability in a representative deformable shape matching task.

Revisiting PointContrast. PointContrast [110] is a recent feature pre-training framework, in which a geometric feature extractor is pre-trained by a pretext task involving correspondences on 3D scenes [28] related by rigid motion. Specifically, PointContrast uses a fully-convolutional sparse U-Net [26] to output a feature vector for every point in the input. During pre-training, given two scene fragments, a contrastive PointInfoNCE loss is used to minimize the feature distance for corresponding points and maximize it for non-corresponding ones:

$$\mathcal{L}_{nce} = - \sum_{(i,j) \in \Omega} \log \frac{\exp(\mathbf{f}_i \cdot \mathbf{f}_j / \tau)}{\sum_{(\cdot, k) \in \Omega} \exp(\mathbf{f}_i \cdot \mathbf{f}_k / \tau)}, \quad (1)$$

where Ω is the set of corresponding point pairs in the overlap region, $\mathbf{f}_\square$ denotes the learned point-wise feature vector, and τ is a temperature hyper-parameter.

Pilot study on deformable shapes. We study the link between the locality of geometric features and their transferability through the lens of a downstream deformable shape matching task [30, 32, 48, 96, 98]. Specifically, we leverage a widely-used human shape dataset, FAUST-Remeshed (FR) [12, 93], consisting of 100 humans in diverse poses.
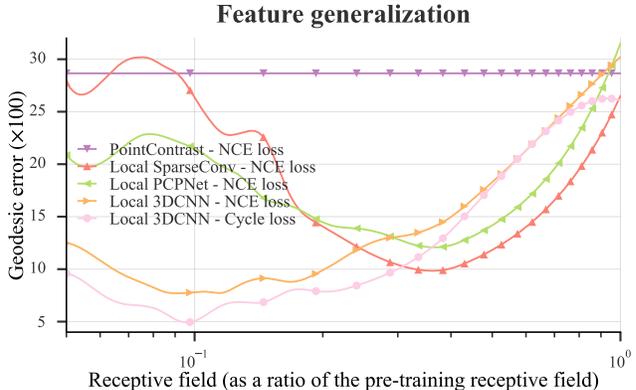

Figure 3. Feature locality vs. transferability in a downstream task of non-rigid shape matching on the FAUST-Remeshed dataset. *Local* in the legend denotes local patch-based input at each point.

Given a pair of 3D shapes as input, we first use pre-trained feature extractors to compute a feature vector for every point in the input. We then find correspondences via nearest neighbor search between the extracted features and apply a lightweight refinement with ZoomOut [80], a common practice in prior works. We use the standard mean geodesic error [57] as the evaluation metric. Note that there is no fine-tuning of network weights, and thus this study provides a good indication of how informative and transferable the pre-trained features are in downstream applications.

Feature locality vs. transferability. We evaluated the features produced by PointContrast in this context and obtained a matching error of **28.7**, compared to **6.1** achieved by a recent axiomatic method [92]. We attribute this limited utility of PointContrast features for deformable shapes to the global structure of its network, which employs a fully-convolutional U-Net design. Furthermore, this network is trained on entire 3D scenes with a global receptive field, making it significantly less likely to generalize to unseen shape categories.

To address this issue, we propose to limit the receptive field size and pre-train feature extractors *that take as input only a local patch centered at each point*, and output a feature vector for the center point (Fig. 2). Intuitively, the space of local patches is significantly smaller than the space of shapes, thus potentially enabling generalization across different shape categories [3, 21, 37, 39, 43, 102].

To evaluate this general approach, we select three different architectures for pre-training a local feature extractor: a) SparseConv [25, 110], a sparse tensor-based network, which also constitutes the backbone of PointContrast; b) PCPNet [43, 90], a PointNet-based architecture; and c) 3D CNN [41, 65], operating on voxel grids. We then pre-train these local feature extractors for *the rigid alignment task* on 3D man-made scenes [115], following a similar strategy as

in PointContrast. We follow the standard design choices and optimal pre-training patch size as used in the existing literature. Please see the exact architectures and pre-training details in the supplementary.

Given these pre-trained local feature extractors, a natural question would be how to adapt the receptive field (patch) size between pre-training and downstream 3D data, which may consist of significantly different shape classes, to make the local feature extractors generalize well. For this, we test a wide range of receptive field sizes (as ratios of the pre-training one) and plot their corresponding matching performance on FR in Fig. 3.

When comparing PointContrast and Local SparseConv in Fig. 3, we observe that making the network *local* and operating on patches significantly improves feature transferability, especially for some specific receptive field size in this downstream task. Moreover, we observe that Local 3D CNN has the best generalization performance on FR, compared to Local SparseConv and Local PCPNet.

Most importantly, this pilot study highlights the importance of feature locality and the crucial role that the optimal receptive field size plays in the downstream task for successful transfer learning. In practice, performing an exhaustive search of receptive field sizes is typically infeasible. To address this issue, we propose a differentiable approach to feature locality optimization for downstream 3D geometric data (Sec. 4.2).

4. Methodology

In this section, we first briefly introduce our chosen local feature pre-training method in Sec. 4.1. We then describe our novel differentiable method for optimizing the receptive field size of local features for transfer learning across datasets in Sec. 4.2.

In the following, we represent our local feature extractor as $\mathcal{F}_{s,\Theta}$, where $s \in \mathbb{R}$ is a *learnable* receptive field size (Sec. 3) and Θ denotes the parameters of the network backbone. Given a shape P represented as either a triangle mesh or a point cloud and a point $\mathbf{p} \in P$, the feature extractor $\mathcal{F}_{s,\Theta}$ associates a feature vector to $\mathbf{p}$, which summarizes the local 3D geometry around $\mathbf{p}$. The feature extractor is composed of two stages: local patch extraction, denoted by $\mathcal{G}_s(\mathbf{p}, P)$, and geometric feature extraction, denoted by $\mathcal{H}_\Theta(\mathcal{G}_s(\mathbf{p}, P)) = \mathcal{F}_{s,\Theta}(\mathbf{p}, P)$.

4.1. Local Feature Pre-training

Feature extraction. Following the transferability study in Fig. 3, we build $\mathcal{F}_{s,\Theta}$ by adapting the local feature extractor architecture from WSDesc [65], which is based on a voxel-based representation [41, 115] and a 3D CNN for geometric learning. Fig. 4 illustrates the local patch extraction procedure. A voxel grid $V_\mathbf{p} = \mathcal{G}_s(\mathbf{p}, P)$ is anchored at $\mathbf{p}$,
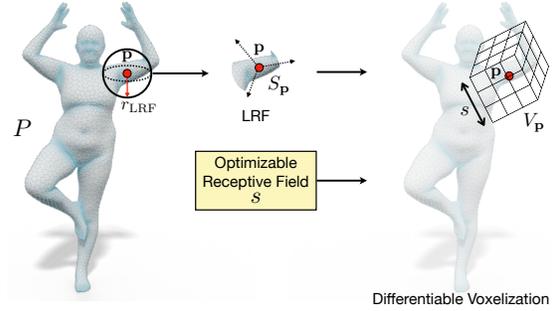

Figure 4. To extract local features, we parameterize the 3D geometry around point $\mathbf{p} \in P$ with a voxel-based representation $V_\mathbf{p}$, reoriented by a local reference frame (LRF) and endowed with an optimizable receptive field size s.

whose size is determined by the optimizable s, to capture the local geometric structure. $V_\mathbf{p}$ is reoriented by a local reference frame (LRF) for rotation invariance in local features [41, 65]. The voxel values in $V_\mathbf{p}$ are computed by a differentiable voxelization layer based on probabilistic aggregation [65] for back-propagating gradients to s. Next, for geometric feature extraction, $V_\mathbf{p}$ is fed to a 3D CNN $\mathcal{H}_\Theta$, which consists of six convolutional layers with ReLU and normalization in-between [41]. The resulting feature vector $\mathcal{H}_\Theta(V_\mathbf{p})$ is 32-dimensional.

Pre-training loss. We adopt the pretext task of matching local features for 3D scene alignment, as advocated in PointContrast [110], to pre-train the feature extractor $\mathcal{F}_{s,\Theta}$. This allows us to gain access to large-scale 3D training data, which has been absent thus far for deformable shape analysis. In this work, we examine two different loss functions for pre-training: one is the PointInfoNCE loss $\mathcal{L}_{\text{nce}}$ (Sec. 3), and the other is an unsupervised loss $\mathcal{L}_c$ based on cycle consistency, introduced in [65] and briefly described below for completeness.

Cycle consistency evaluates the extracted features in a rigid registration pipeline without using ground truth labels. Given two shapes P and Q and their per-point features extracted by $\mathcal{F}_{s,\Theta}$, putative correspondences between P and Q are first constructed by differentiable nearest neighbor search [86] in the learned feature space. Then a 3D transformation aligning P to Q is estimated from the correspondences and represented as a rotation $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ and a translation $\mathbf{t} \in \mathbb{R}^3$. Let $\mathbf{R}'$ and $\mathbf{t}'$ denote the computed alignment, in the reverse direction, from Q to P. The cycle consistency loss enforces the composition of $(\mathbf{R}, \mathbf{t})$ and $(\mathbf{R}', \mathbf{t}')$ to be an identity matrix as follows:

$$\mathcal{L}_c = \|\mathbf{R}\mathbf{R}' - \mathbf{I}\|_1 + \|\mathbf{R}\mathbf{t}' + \mathbf{t}\|_1, \qquad (2)$$

where $\mathbf{I}$ is an identity matrix. We refer the reader to [65] for more details.

As shown in Fig. 3, pre-training with $\mathcal{L}_c$ leads to local

features having significantly better transferability on FR, compared to pre-training with $\mathcal{L}_{\text{nce}}$.

This is remarkable as the PointInfoNCE loss is a go-to choice for representation learning across many domains and application scenarios [23, 34, 64]. We ascribe this performance difference partly to the fact the PointInfoNCE loss focuses on whether *individual* point correspondences are correct or not, and does not consider local or global feature consistency [64]. In contrast, the cycle consistency loss jointly assesses all the extracted local features and thus imposes correspondence consistency between P and Q for successful global alignment. As a result, $\mathcal{L}_c$ promotes more consistent (and smoother) local features to be learned for nearby points than $\mathcal{L}_{\text{nce}}$.

To validate this hypothesis, we measure the Dirichlet energy [84] of the pre-trained features (the lower, the smoother) on the human shapes of FR, resulting in 86.8 for $\mathcal{L}_{\text{nce}}$ and 75.4 for $\mathcal{L}_c$. These measurements confirm that the local features pre-trained by the cycle consistency loss are smoother (and thus more consistent) than those by the PointInfoNCE loss, which can potentially benefit downstream applications, especially across different datasets.

4.2. Local Feature Transfer

As we demonstrated in Sec. 3, the size of the receptive field is of crucial importance for downstream tasks. In this section, we introduce a receptive field optimization method, which automatically determines the best receptive field for each downstream dataset. To this end, we take advantage of the full differentiability of our network design and optimize the learnable receptive field size s, so that the distribution of features extracted on the target dataset is similar to that of features extracted by the pre-trained network on the pre-training dataset.

We draw inspiration from the domain adaptation literature [53, 62, 68, 73], where the main goal is to transfer knowledge from a network trained in a domain with abundant labeled data to a domain of interest with minimal or unlabeled data. This is achieved by minimizing the *domain shift* between the feature distribution of the two domains using some distance in the probability measure space. Thus we propose to minimize the discrepancy between the distribution of features extracted from the pre-training dataset, and the distribution of features extracted from the target (downstream) dataset. For this, we base our approach on the Maximum Mean Discrepancy (MMD) used in previous works [14, 73, 74, 91]. Intuitively, networks are pre-trained to have a strong response signal under the distribution of training local patches, and thus our goal is to optimize the local patch sizes, given *unlabeled* downstream data, so that the extracted features have a similar distribution.

In practice, after pre-training the feature extractor, we sample n_s points from the pre-training (source) dataset and

extract the features from these points with the pre-trained network. We denote these features as $F^s = \{f^s_i\}_{i=1..n_s}$. For each new downstream (target) dataset, we formulate an optimization problem, where we freeze the network backbone weights Θ but optimize the receptive field size s, so that the network produces features as close to F^s as possible, using the MMD metric. Our network is fully differentiable, and we solve this problem via gradient descent to find the optimal receptive field s. This optimized receptive field is then used for all downstream tasks involving this dataset. At each iteration during optimization, we randomly sample n_t points from the target dataset and compute features for them with the pre-trained network. We denote these features as $\{f^t_i\}_{i=1..n_t}$ and formulate the loss as follows:

$$E_{mmd} = \frac{1}{n_s n_s} \sum_{i,j=1}^{n_s} \kappa(f^s_i, f^s_j) + \frac{1}{n_s n_t} \sum_{i,j=1}^{n_s, n_t} \kappa(f^s_i, f^t_j)$$
$$+ \frac{1}{n_t n_t} \sum_{i,j=1}^{n_t} \kappa(f^t_i, f^t_j), \tag{3}$$

where κ is the Radial Basis Function (RBF) kernel. The Adam optimizer [58] is used in gradient descent.
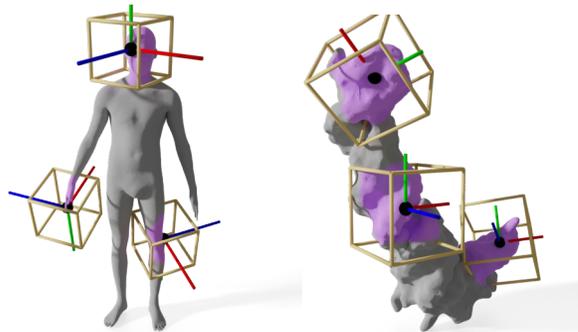

Figure 5. Qualitative visualization of the receptive field of local patches used in our downstream experiments on both the human and RNA datasets. The points of interest are indicated in black. The surface areas in pink represent the local receptive fields *after optimization*. We also show the corresponding voxel grids and the XYZ axes of the local reference frames.

5. Experiments

In this section, we provide extensive experiments to highlight the generalization power of our local feature pre-training and receptive field size optimization methods in a suite of downstream shape analysis tasks especially involving highly deformable, organic shapes. We consider diverse benchmarks including human and partial animal shape matching as well as molecular surface segmentation. We reuse the same pre-trained local feature extractor $\mathcal{F}_{s,\Theta}$ and then perform our receptive field size optimization once individually on each deformable shape dataset (Sec. 4.2).

Method / Dataset	FR-SH	SR-SH
SURFMNET [96]	30.1	28.6
Cyclic FMaps [40]	36.5	38.6
WSupFMNet [98]	26.3	30.2
Deep Shells [32]	26.3	22.8
DiffusionNet - XYZ	22.4	23.3
DiffusionNet - HKS	10.4	15.4
DiffusionNet - WKS	9.3	24.0
DiffusionNet - SHOT	10.8	21.5
DiffusionNet - PCH	25.8	33.2
DiffusionNet - PCN	22.6	36.2
DiffusionNet - VADER (ours)	**6.4**	**6.9**

Table 1. Performance of various features for unsupervised deformable shape matching on un-aligned data. X-Y means training on X and testing on Y. Values are mean geodesic error $\times 100$ on unit-area shapes.

Implementation. We denote our features as VADER for Voxelized Alignment-based DEscriptoR. In the pre-training stage, we use the 3DMatch dataset [115] and train the local feature extractor $\mathcal{F}_{s;\Theta}$ for 16K steps. We use the Adam optimizer with a learning rate of 10^{-3} for network weight update. For receptive field size optimization, we use the same learning rate for Adam, and we take $n_s = 10^4$ extracted patches from the pre-training dataset.

Baselines. We compare a wide spectrum of hand-crafted and pre-trained features in deformable shape tasks. For the hand-crafted features, we consider the Heat Kernel Signature (HKS) [100], Wave Kernel Signature (WKS) [9], and SHOT descriptors [101], as well as the straightforward vertex positions (XYZ). For the pre-trained features, we use the PointContrast features learned with the PointInfoNCE loss (PCN) or a hardest-contrastive loss (PCH) [110].

5.1. Human Shape Matching

Unsupervised matching. We perform unsupervised shape matching [32, 96] on the FAUST-Remeshed (FR), SCAPE-Remeshed (SR), and SHREC'19 datasets (SH) [2, 12, 79, 93], consisting of 100, 71, and 44 *unaligned* human shapes in different poses, respectively. The same train/test splits in prior works [32, 98] are adopted. We feed the above baseline features and our VADER respectively as input to a surface learning backbone DiffusionNet [99], which produces a functional map [70, 82] as output for a given pair of shapes. We leverage the unsupervised functional map losses [98] for training the backbone. The evaluation metric is the mean geodesic error of predicted maps with respect to the ground truth on unit-area shapes [57]. We use X-Y to denote training on dataset X and testing on dataset Y.

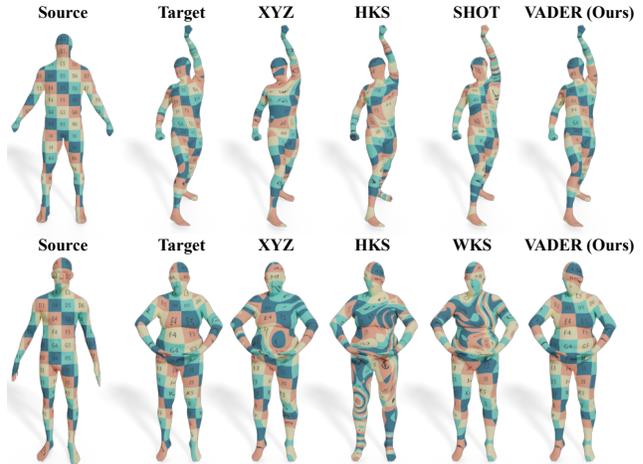As shown in Tab. 1, our approach yields the best re-

Figure 6. Qualitative comparisons of human shape matching by texture transfer. Top: results of unsupervised matching on SH. Bottom: results of supervised matching on FQ. The best three performing competitors are shown.

sults for unsupervised shape correspondence in both FR-SH and SR-SH settings, while the other tested features fail to achieve reasonable matching performance. We stress that this is a challenging test case as most existing unsupervised methods rely on aligned shapes, e.g., [31, 98]. Note that the PointContrast features perform worse than the hand-crafted features, indicating the overfitting to the pre-training data distribution, as discussed in Sec. 3, and thus the limited transferability. Our approach also outperforms several recent unsupervised approaches in Tab. 1, including SURFM-NET [96], Cyclic FMaps [40], WSupFMNet [98], and Deep Shells [32]. The comparisons clearly demonstrate the utility of incorporating generalizable pre-trained features, which is missing in prior works. We provide qualitative comparisons in Fig. 6 (Top), showing that only our approach leads to visually plausible results.

Robustness to meshing. We evaluate the performance of supervised shape matching [30] on the original FAUST (FO) dataset [12] and its remeshed version by quadratic error simplification (FQ) [99] to demonstrate the robustness and generalization of our approach against significant mesh connectivity changes across datasets. We build a point-wise MLP network [70] (to reduce the dependence on the backbone architecture) on top of the baseline features and our VADER respectively, and then predict functional maps for shape pairs. The MLP backbones are trained on FO with predictions supervised by the ground-truth maps with a simple L_2 loss. We report the mean geodesic error metric on FQ.

Fig. 7 shows the correspondence quality with a varying error threshold. It can be seen that hand-crafted features such as SHOT degrade rapidly under remeshing. Although
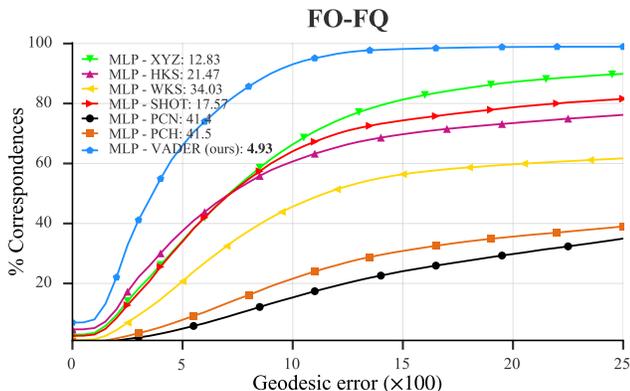
Figure 7. Accuracy of various features for supervised shape matching when the connectivity changes from training to test (mean errors $\times 100$ are reported in the legend).

Method	Accuracy $\pm$ s.d		
	Full Dataset	50 Shapes	100 Shapes
PointNet++ [90]	74.4%	–	–
PCNN [8]	78.0%	–	–
SPHNet [89]	80.1%	–	–
SplineCNN [35]	53.6%	–	–
SurfaceNetworks [61]	88.5%	–	–
DiffusionNet - XYZ [99]	90.5 $\pm$ 0.6%	82.7 $\pm$ 0.63%	83.4 $\pm$ 0.67%
DiffusionNet - HKS	90.6 $\pm$ 0.15%	82.7 $\pm$ 0.16%	84.5 $\pm$ 0.09%
DiffusionNet - WKS	88.7 $\pm$ 0.26%	77.6 $\pm$ 0.16%	81.2 $\pm$ 0.20%
DiffusionNet - SHOT	92.1 $\pm$ 0.08%	81.6 $\pm$ 0.31%	85.7 $\pm$ 0.11%
DiffusionNet - PCH	90.3 $\pm$ 0.1%	79.9 $\pm$ 0.59%	83.6 $\pm$ 0.08%
DiffusionNet - PCN	90.1 $\pm$ 0.09%	80.1 $\pm$ 0.29%	83.4 $\pm$ 0.28%
DiffusionNet - VADER (ours)	**92.6 $\pm$ 0.02%**	**83.2 $\pm$ 0.20%**	**86.8 $\pm$ 0.09%**

Table 2. Accuracy of various mesh and point cloud based methods for RNA segmentation. The reported numbers are mean accuracy over 5 runs randomly initialized. $\pm$ denotes standard deviation.

WKS and HKS are intrinsic features and do not depend on the meshing connectivity, they are not expressive enough by themselves and need to be combined with a powerful network backbone such as DiffusionNet, instead of the pointwise MLPs used in this experiment. Differently, our approach achieves superior generalization performance in this challenging setting, showing that VADER is highly robust to remeshing and resampling, and effectively captures local geometric structures in deformable shapes. Fig. 6 (bottom) presents a qualitative comparison of the computed maps with the three best-performing competitors.

5.2. Molecular Surface Segmentation

Next, we conduct experiments in the molecular surface segmentation task, which aims to segment RNA molecules into functional components. We use the dataset introduced in [89], consisting of 640 RNA triangle meshes, where each vertex is labeled into one of 259 atomic categories. The dataset has an 80/20% split for training and test sets. We

Method / Dataset	SHREC'16 CUTS Animals
PFM [94]	8.8
FSP [71]	12.2
DiffusionNet - XYZ	4.9
DiffusionNet - SHOT	4.6
DiffusionNet - VADER (ours)	**3.7**

Table 3. Performance (mean geodesic error $\times 100$) of various features on the SHREC'16 CUTS Animals benchmark.

Figure 8. Qualitative comparisons of partial animal matching by texture transfer on the cat class of the SHREC'16 CUTS Animals benchmark.

feed the baseline features and our VADER respectively as input to DiffusionNet and train it to predict a label at each vertex as output.

As shown in Tab. 2, our approach achieves state-of-the-art segmentation performance when used with the full training set, outperforming both hand-crafted and pre-trained PointContrast features as well as several recent shape segmentation networks, such as [61]. We also perform experiments in more challenging settings, where only a fraction of the training set, with respectively 50 and 100 shapes (corresponding to 9% and 18% of the training set), is used. We observe in Tab. 2 that our method consistently outperforms the competitors by a significant margin when given limited training data. The results highlight that our pre-training and receptive field size optimization strategies bring significant improvement to downstream organic shape analysis tasks.

5.3. Partial Animal Matching

We also evaluate how well different geometric features perform on deformable shapes in the presence of significant partiality. For this, we test on the challenging SHREC16' Cuts dataset [27], where the animal classes (cat, centaur, dog, horse, and wolf) are used for partial shape matching. We follow the setup of DiffusionNet described in Sec. 5.1 for correspondence prediction.

We compare our approach to XYZ and SHOT as they are widely used in partial matching pipelines, in addition to full-fledged methods PFM [94] and FSP [71], specifically tailored toward partial shape matching. The results are summarized in Tab. 3. Qualitative results are visualized in Fig. 8. Observe that our approach outperforms the competitors in this setting by a significant margin, including specially-tailored partial matching methods PFM and FSP.

% train data	PointContrast		VADER (ours)	
	From scratch	PointInfoNCE	From scratch	PointInfoNCE
1% data	53.2	60.7	59.5	**66.5**
10% data	74.4	73.7	72.2	**77.2**
100% data	76.9	77.2	79.0	**81.2**

Table 4. ShapeNet classification accuracy with limited labeled training data for fine-tuning.

Dataset	FR-SR	SR-FR
DFAUST	27.7	4.4
3DMatch	**4.1**	**3.8**

Table 5. Results of using features pre-trained on different datasets in the downstream task of unsupervised non-rigid shape matching. Values are mean geodesic error $\times 100$ on unit-area shapes.

5.4. Shape Classification

We use the ShapeNet dataset [22] to demonstrate the effectiveness of our pre-trained feature extractor on man-made objects. We follow the classification setup from Point-Contrast [110], using pre-trained weights as initialization for fine-tuning a classification network. Here our feature extractor is pre-trained with the PointInfoNCE loss. We conduct experiments on the ShapeNetCore v2 dataset with the same train/test split as PointContrast. We also consider a limited fine-tuning data setup, using a fraction of the fine-tuning data (1% or 10%). The classification accuracy comparison is summarized in Tab. 4. It can be seen that feature pre-training improves performance compared to training from scratch. Also, our network achieves higher classification accuracy than PointContrast in all training setups.

5.5. Ablation Study

We also evaluate the role of the pretraining dataset for local feature learning. For this, we compare 3DMatch used in our experiments to DFAUST [13], a large-scale dataset of human subjects in motion. We use the unsupervised shape matching task and the evaluation protocol introduced in Sec. 5.1 to test the generalization of a pre-trained feature extractor to the FR and SR datasets.

The comparisons in Tab. 5 show that pre-training on 3DMatch leads to more generalizable features and consistent matching performance, even though DFAUST has greater similarity to the downstream human shape datasets FR and SR. We attribute this to the fact that *local geometries* in 3DMatch, which consists of real-world scans, are richer and more complex than those in template-fitted DFAUST, leading to a more universally useful pre-training signal.

To validate this, we perform PCA analysis [36] on the local patches of 3DMatch and DFAUST. For each dataset, we first randomly extract 200K local patches. We then encode each patch as a high dimensional vector by first orienting
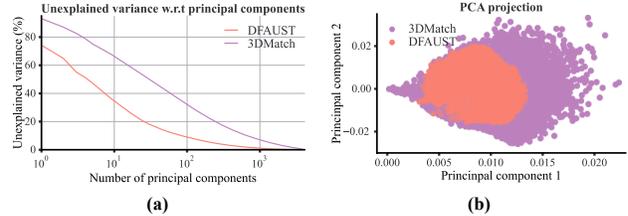

(a)   **(b)**

Figure 9. Comparing richness of local geometries in 3DMatch and DFAUST via PCA. (a) We perform PCA on sampled 3D local patches and plot the unexplained variance w.r.t. the number of principal components. (b) We project the local geometries onto the first two principal components.

it using a local reference frame and then voxelizing it to a small 3D grid of resolution $= 16^3$ using the method of [41]. The resulting vectors are 4096-dimensional and are fed as input to PCA. In Fig. 9 (a), we report the unexplained variance as a function of the number of principal components. It can be seen that 3DMatch is significantly more diverse than DFAUST since more principal components are needed to explain its full variance. In Fig. 9 (b), we visualize the projection of patches in the first two principal components and observe that local patches in DFAUST (red dots) are included in 3DMatch (blue dots), demonstrating the diversity and richness of 3DMatch once more.

6. Conclusion and Limitations

In conclusion, we have shown that our method of pre-training local features on rigid 3D scenes can generalize well to new and unseen classes of deformable organic shapes, enabling effective performance in various shape analysis tasks. Our study has highlighted the importance of selecting the right receptive field size to ensure feature transferability, which has led to the *first general-purpose local feature pre-training* for deformable shape analysis tasks. This research also sheds light on the relationship between rigid and non-rigid processing tasks, providing a link between two fields that have traditionally used different tools.

One limitation of our method is its reliance on differentiable voxelization, which can be memory and time-consuming, particularly during pre-training. Nonetheless, our results outperform PointContrast [110], a point-based method that requires *more training data* and has limited generalizability. Another limitation is that our features rely on LRF estimation, which might lack robustness to thin structures or boundaries of partial shapes. Exploring alternative scalable and robust local feature pre-training strategies is an fascinating direction for future work.

Supplementary Materials for:
Generalizable Local Feature Pre-training for Deformable Shape Analysis

Souhaib Attaiki   Lei Li   Maks Ovsjanikov

LIX, École Polytechnique, IP Paris

In this document, we collect all the results and discussions, which, due to the page limit, could not find space in the main manuscript. This supplementary material consists of two parts. First, in Appendix A, we describe more implementation details mainly regarding our pilot study, local feature pre-training, and experiments on downstream deformable shape data. Next, in Appendix B, we present additional experimental results and analysis of our local feature pre-training strategy and its generalization in downstream tasks, including deformable shape matching and segmentation.

A. Implementation Details

A.1. Feature Locality vs. Transferability

In Sec. 3 of the main text, we conducted a pilot study on feature locality vs. transferability on deformable shapes. We tested three different architectures for pre-training a *local* feature extractor, and their details are as follows.

SparseConv. We used the `ResNet14` architecture introduced in [24]. During pre-training, given a 3D point cloud P, a fixed-size local patch with a radius of 0.15 is cropped at point $\mathbf{p} \in P$ and then reoriented with a local reference frame (LRF) computed by the method in [41] for rotation invariance. The resulting local patch is fed to the sparse convolution network, which extracts a 32-dimensional feature vector for point $\mathbf{p}$.

PCPNet. It is a variant of PointNet [90] endowed with a quaternion spatial transformer. We used the single-scale architecture proposed by [43]. PCPNet is designed to be a local network requiring input patches to have a fixed number of points. Thus during pre-training, a fixed-size local patch (radius = 0.15) is cropped at point $\mathbf{p}$ and reoriented by an LRF. The local patch is then resampled to 1,024 points and fed to the network, resulting in a 32-dimensional feature vector for point $\mathbf{p}$.

3DCNN. We used the architecture from [65] with a learnable receptive field size and differentiable voxelization, the same as our VADER in Sec. 4.1 of the main text. More details can be found in Appendix A.2.

Dataset. We pre-trained the above local networks on the 3DMatch dataset, which is a collection of RGB-D scan datasets with 62 indoor scenes and 4,142 point cloud fragments. There are 13K points on average in a fragment after downsampling.

Loss. We used the PointInfoNCE loss, in which 300 point correspondences were randomly sampled for a pair of point clouds for faster training and the temperature parameter τ was set to 0.07.

We also used the cycle consistency loss $\mathcal{L}_c$. During pre-training, we use the extracted features to build correspondences for rigid alignment between shapes P and Q. The intuition for $\mathcal{L}_c$ is that the estimated transformation $(\mathbf{R}, \mathbf{t})$ aligning P to Q should be the inverse of the transformation $(\mathbf{R}', \mathbf{t}')$ aligning Q to P. Mathematically, this can be expressed as:

$$\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}' & \mathbf{t}' \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}\mathbf{R}' & \mathbf{R}\mathbf{t}' + \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{I} \quad (4)$$

Application to deformable shape matching. In Fig. 3 of the main text, we have shown the results of shape matching on the Faust Remeshed dataset, directly using the pre-trained feature extractors. Given two shapes S_1, and S_2, we compute their respective point-wise features F_1 and F_2 using a specific pre-trained model. We first produce an estimate of the point-to-point maps T_{21}^{nn} and T_{12}^{nn} using nearest neighbor search between F_1 and F_2. We then filter the correspondences by mutual check: a pair of points $x \in S_1, y \in S_2$ is considered to be in correspondence, if and only if in the feature space, x is the nearest neighbor of y, and y is the nearest neighbor of x. This results in two filtered maps T_{21}^{mf} and T_{12}^{mf}. Finally, we further refine these two maps using the ZoomOut method [80], which is based on navigating between the spectral and spatial domains while progressively increasing the number of spectral basis functions. We emphasize that if the initial point-to-point map is noisy or contains strong ambiguities like symmetry ambiguities, ZoomOut is not able to remedy these errors, thus leading to final correspondences of bad quality. We perform 10 iterations of ZoomOut, starting from 30 eigenfunctions up to 100 eigenfunctions.

A.2. Local Feature Pre-training

In Sec. 4.1 of the main text, we introduced our local feature pre-training strategy.

Feature extraction. We use $r_{LRF} = 0.3$ and $\sigma = 10^{-3}$ for differentiable voxelization [65], and the voxel grid resolution is set to 16^3. We pre-trained on the 3DMatch dataset introduced in Appendix A.1.

Pre-training loss. For the PointInfoNCE loss $\mathcal{L}_{nce}$, its settings are described in Appendix A.1. For the cycle consistency loss $\mathcal{L}_c$, 300 points were randomly sampled on each point cloud for feature extraction and alignment estimation. A relaxation-based solver is used in $\mathcal{L}_c$ for estimating a 3D transformation between two point clouds, and its details can be found in [65].

In the main text, we investigated the performance difference between the cycle consistency loss and PointInfoNCE loss w.r.t learned feature smoothness. Suppose that $F \in \mathbb{R}^{m \times n}$ is the matrix of extracted n-dimensional pointwise features for a shape of m vertices, we measure the Dirichlet energy as follows:

$$E_{Dirichlet}(F) = \frac{1}{n} \sum_{i=1}^{n} F_i^\top W F_i, \qquad (5)$$

where F_i is the ith column of F, and W is the standard stiffness matrix computed using the classical cotangent discretization scheme of the Laplace-Beltrami operator [85].

A.3. Baselines

In Sec. 5 of the main text, we tested our proposed VADER features against a wide spectrum of competitors, including both hand-crafted and learned features.

Specifically, the Heat Kernel Signature (HKS) and Wave Kernel Signature (WKS) features are both sampled at 100 values of energy *t*, logarithmically spaced in the range proposed in their respective original papers. SHOT descriptors are 352-dimensional, and we used the implementation from the PCL library [97]. PointContrast features are 32-dimensional, and we used the publicly available implementation and the pre-trained weights released by the authors[1].

A.4. Downstream Shape Analysis Training

In Sec. 5 of the main text, we used DiffusionNet on top of the baselines features and our VADER respectively, in both the shape matching and segmentation tasks. We employed the publicly available implementation of DiffusionNet released by the authors[2]. Unless specified otherwise, in our experiments, we used four DiffusionNet blocks of width = 128. The DiffusionNet is trained by an ADAM optimizer [58] with an initial learning rate of 10^{-3}.

In Sec. 5.1 of the main text, we also used a point-wise MLP network on top of the baselines features and our

---

[1]https : / / github . com / facebookresearch / PointContrast
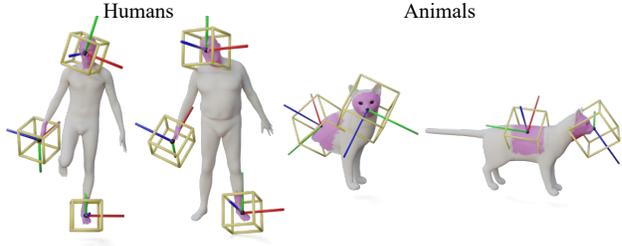[2]https://github.com/nmwsharp/diffusion-net

Figure 10. Visualizing the optimized receptive field for shape pairs.

VADER respectively for supervised shape matching. For this, we use the same MLP architecture as in FMNet [70]. After computing the point features with the MLP, we use them to compute the predicted functional map C_{pred} as in [30] and penalize its deviation from the ground-truth map C_{gt} using the L2 loss: $L = \|C_{pred} - C_{gt}\|_2^2$.

A.5. Computational Specifications

All our experiments were executed using Pytorch [83], on a 64-bit machine, equipped with an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and an RTX 2080 Ti Graphics Card.

In terms of computational time, pertaining our method takes about 12 hours on a single RTX 2080 Ti Graphics Card, in contrast to the 64 hours required for PointContrast. The receptive field optimization takes about 20 minutes per dataset. For feature extraction, our method takes 3 seconds to extract local features for a 5000-vertex shape, which is on par with other local features like SHOT [101], but slower than PointContrast (0.1s). Finally, the forward pass using VADER takes the same time as for all baseline features, e.g., 0.2 seconds per iteration for the unsupervised shape-matching experiment in Sec 5.1 of the main text.

B. Additional Results and Analysis

B.1. Size of the learned receptive field

Fig. 5 of our paper provides an illustration of the optimized receptive field in downstream tasks. In Fig. 10, we include more visualizations for shape *pairs* for both humans and animals. Observe that the optimized receptive field indeed corresponds to interpretable concepts, such as the head or foot of a human, and is consistent across shape pairs.

B.2. Human Shape Matching

In Sec. 5.1 of the main text, we performed unsupervised shape matching on the FAUST-Remeshed (FR), SCAPE-Remeshed (SR), and SHREC'19 datasets (SH) and reported the matching performance in Tab. 1. We provide additional quantitative results of the FR-SR and SR-FR settings in Tab. 6. Compared with the baseline features, our VADER

Method / Dataset	FR-SR	SR-FR
SURFMNET	15.2	9.5
Cyclic FMaps	23	23.2
WSupFMNet	27.1	14.2
Deep Shells	6.0	**3.4**
DiffusionNet - XYZ	25.7	8.4
DiffusionNet - HKS	7.9	23
DiffusionNet - WKS	4.2	24.1
DiffusionNet - SHOT	7.2	4.1
DiffusionNet - PCH	11.4	8.7
DiffusionNet - PCN	20.4	9.1
DiffusionNet - VADER (ours)	**4.1**	3.9

Table 6. Accuracy of various features for unsupervised shape matching on un-aligned data. X-Y means train on X and test on Y. Values are mean geodesic error $\times 100$ on unit-area shapes.

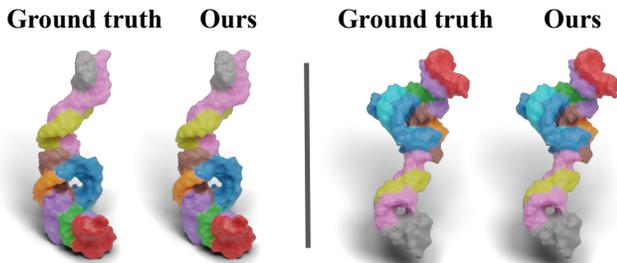**Ground truth   Ours   Ground truth   Ours**

Figure 11. Qualitative evaluation of RNA segmentation on the dataset of [89]. Left: ground truth. Right: prediction by DiffusionNet + VADER.

has the best and most consistent performance in both settings.

B.3. Molecular Surface Segmentation

In Fig. 11, we show qualitative results of RNA segmentation using DiffusionNet + VADER. It can be seen that the challenging RNA molecules can be robustly segmented into functional components with our pre-trained features.

B.4. Human Shape Segmentation

We performed an additional experiment on the human shape segmentation task. We used the dataset introduced in [76], which combines segmented human models taken from a variety of existing datasets. We used the same train/test split of 380 training and 18 test shapes as in prior works. We compared our VADER only with methods that used the original evaluation protocol as in [76], i.e., without using post-processing and evaluating the results on the full shape resolution (techniques such as Mesh Walker [63] are thus excluded).

We ran each experiment five times and report the mean and standard deviation of the accuracy in Tab. 7. Our

Method	Accuracy $\pm$ s.d
GCNN [77]	86.4%
ACNN [15]	83.7%
Toric Cover [76]	88.0%
PointNet++ [90]	90.8%
MDGCNN [89]	88.6%
DGCNN [107]	89.7%
SNGC [47]	91.0%
CGConv [111]	89.9%
DiffusionNet - XYZ	91.9 $\pm$ 0.27%
DiffusionNet - HKS	91.5 $\pm$ 0.21%
DiffusionNet - WKS	91.8 $\pm$ 0.33%
DiffusionNet - SHOT	91.5 $\pm$ 0.77%
DiffusionNet - PCH	85.6 $\pm$ 0.75%
DiffusionNet - PCN	87.3 $\pm$ 0.57%
DiffusionNet - VADER (ours)	**92.4 $\pm$ 0.25%** (+0.9)

Table 7. Human shape segmentation on the dataset of [76]. Our VADER achieves the state-of-the-art performance among methods that do not perform post-processing and evaluate on the full shape resolution. The reported numbers are the mean and standard deviation of the accuracy over five runs initialized randomly.

Ground truth   Ours   Ground truth   Ours

Figure 12. Qualitative evaluation of human shape segmentation on the dataset of [76]. Left: ground truth. Right: prediction by DiffusionNet + VADER.

VADER features achieve an accuracy of $92.4 \pm 0.25\%$, the state-of-the-art result on this dataset. In Fig. 12, we present qualitative results of human segmentation using DiffusionNet + VADER. Note that the segmentation results are simply the network predictions, and we do not perform any complex post-processing to the segmentation.

B.5. Robustness to Noise

We performed an additional experiment to evaluate the robustness of our features to noise. For this, we followed the same setup as in Sec. 5.1 of the main text and in Appendix B.2, by performing unsupervised learning on FR and testing on SR with an increasing amount of noise as input. We compared our method to the best three competing features. The results are shown in Fig. 13 - left. It can be seen
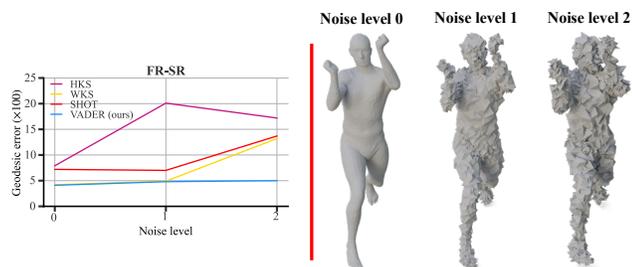
Figure 13. Left: Evolution of the geodetic error as a function of different input noise levels. Right: Qualitative visualization of noise levels.
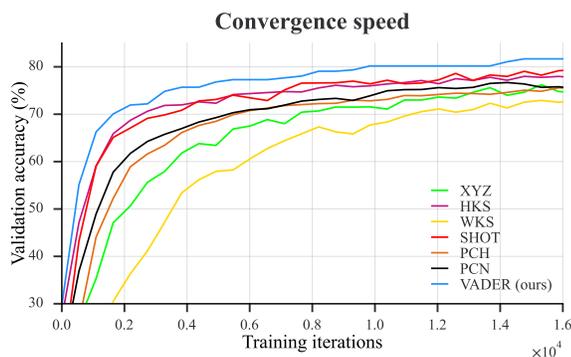

Figure 14. Evolution of the RNA segmentation accuracy on the validation set, during the training of DiffusionNet with different features.

that our features are more robust to noise, i.e., the performance does not vary much with different noise levels (the intensity of the noise can be seen in Fig. 13 - right), which is not the case with other features, such as SHOT, whose performance degrades very quickly.

B.6. Convergence Speed

In our experiments, we observed that our VADER descriptors take less time to train and facilitate learning. To demonstrate this, we show in Fig. 14 the evolution of validation accuracy during learning of the RNA segmentation task (Sec. 5.2 of the main text). It can be seen that compared to the other features, VADER requires far fewer training iterations to achieve similar performance. This clearly indicates the better descriptiveness and generalizability of our features.

References

[1] Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *International Journal of Computer Vision*, 118(3):380–392, 2016. 2

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIG-GRAPH*, pages 408–416. 2005. 6

[3] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2021. 3

[4] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019. 2

[5] Souhaib Attaiki and Maks Ovsjanikov. NCP: Neural correspondence prior for effective unsupervised shape matching. In *Advances in Neural Information Processing Systems*, 2022. 2

[6] Souhaib Attaiki and Maks Ovsjanikov. Understanding and improving features learned in deep functional maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 2

[7] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. DPFM: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*. IEEE, Dec. 2021. 2

[8] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 7

[9] 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. 1, 2, 6

[10] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3Feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020. 2, 3

[11] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. *Advances in neural information processing systems*, 13:831–837, 2000. 1

[12] 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. 3, 6

[13] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 8

[14] Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22 14:e49–57, 2006. 5

[15] Davide Boscaini, Jonathan Masci, Emanuele Rodoià, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3197–3205, 2016. 2, 11

[16] Alexander M Bronstein, Michael M Bronstein, and Maks Ovsjanikov. Feature-based methods in 3d shape analysis. In *3D Imaging, Analysis and Applications*, pages 185–219. Springer, 2012. 2

[17] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 1, 2

[18] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1704–1711. IEEE, 2010. 2

[19] Oliver Burghard, Alexander Dieckmann, and Reinhard Klein. Embedding shapes with Green's functions for global shape matching. *Computers & Graphics*, 68:1–10, 2017. 2

[20] Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020. 1, 2

[21] Rohan Chabra, Jan E. Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. *Lecture Notes in Computer Science*, pages 608–625, 2020. 3

[22] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3d model repository. *arXiv*, 2015. 8

[23] Yujin Chen, Matthias Nießner, and Angela Dai. 4DContrast: Contrastive learning with dynamic correspondences for 3d scene understanding. *arXiv*, 2021. 3, 5

[24] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 2, 9

[25] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3

[26] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019. 2, 3

[27] Luca Cosmo, Emanuele Rodola, Michael M Bronstein, Andrea Torsello, Daniel Cremers, and Y Sahillioglu. Shrec'16: Partial matching of deformable shapes. *Proc. 3DOR*, 2(9):12, 2016. 7

[28] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3

[29] Haowen Deng, Tolga Birdal, and Slobodan Ilic. PPFNet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 195–205, 2018. 2

[30] 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. 1, 2, 3, 6, 10

[31] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021. 6

[32] 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. 2, 3, 6

[33] Davide Eynard, Emanuele Rodola, Klaus Glashoff, and Michael M Bronstein. Coupled functional maps. In *3D Vision (3DV)*, pages 399–407. IEEE, 2016. 2

[34] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege. Deep graph matching consensus. In *International Conference on Learning Representations (ICLR)*, 2020. 5

[35] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018. 2, 7

[36] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 8

[37] Kent Fujiwara, Ko Nishino, Jun Takamatsu, Bo Zheng, and Katsushi Ikeuchi. Locally rigid globally non-rigid surface registration. In *2011 International Conference on Computer Vision*, pages 1527–1534. IEEE, 2011. 3

[38] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. Pose-oblivious shape signature. *IEEE transactions on visualization and computer graphics*, 13(2):261–271, 2007. 2

[39] Ran Gal, Ariel Shamir, Tal Hassner, Mark Pauly, and Daniel Cohen-Or. Surface reconstruction using local shape priors. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 253–262. Eurographics Association, 2007. 3

[40] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes, 2019. 6

[41] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019. 2, 3, 4, 8, 9

[42] 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. 1, 2

[43] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. Pcpnetlearning local shape properties from raw point clouds. *Computer Graphics Forum*, 37(2):75–85, May 2018. 3, 9

[44] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, 2014. 2

[45] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116(1):66–89, 2016. 2

[46] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 1, 2

[47] Niv Haim, Nimrod Segol, Heli Ben-Hamu, Haggai Maron, and Yaron Lipman. Surface networks via general covers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 632–641, 2019. 11

[48] 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. 3

[49] Paul Heider, Alain Pierre-Pierre, Ruosi Li, and Cindy Grimm. Local shape descriptors, a survey and evaluation. In *Proceedings of the 4th Eurographics conference on 3D Object Retrieval*, pages 49–56, 2011. 2

[50] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212, 2001. 2

[51] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. *arXiv*, 2020. 1, 3

[52] Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics (TOG)*, 37(1):1–14, 2017. 2

[53] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data, 2021. 5

[54] Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)*, 33(4):36, 2014. 2

[55] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999. 1, 2

[56] Marc Khoury, Qian-Yi Zhou, and Vladlen Koltun. Learning compact geometric features. In *Proceedings of the IEEE*

[57] international conference on computer vision*, pages 153–161, 2017. 2

[57] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics*, 30(4):1–12, July 2011. 3, 6

[58] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5, 10

[59] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. Hough transform and 3d surf for robust three dimensional classification. In *European Conference on Computer Vision*, pages 589–602. Springer, 2010. 2

[60] Marcel Körtgen, Gil-Joo Park, Marcin Novotni, and Reinhard Klein. 3d shape matching with 3d shape contexts. In *The 7th central European seminar on computer graphics*, volume 3, pages 5–17. Citeseer, 2003. 2

[61] Ilya Kostrikov, Zhongshi Jiang, Daniele Panozzo, Denis Zorin, and Joan Bruna. Surface networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2540–2548, 2018. 7

[62] Jogendra Nath Kundu, Naveen Venkat, Rahul M V, and R. Venkatesh Babu. Universal source-free domain adaptation. June 2020. 5

[63] Alon Lahav and Ayellet Tal. Meshwalker: deep mesh understanding by random walks. *ACM Transactions on Graphics (TOG)*, 39(6):1–13, 2020. 11

[64] Lei Li, Souhaib Attaiki, and Maks Ovsjanikov. SRFeat: Learning locally accurate and globally consistent non-rigid shape correspondence. In *International Conference on 3D Vision*, 2022. 5

[65] Lei Li, Hongbo Fu, and Maks Ovsjanikov. WSDesc: Weakly supervised 3d local descriptor learning for point cloud registration. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 3, 4, 9, 10

[66] Lei Li, Siyu Zhu, Hongbo Fu, Ping Tan, and Chiew-Lan Tai. End-to-end learning local multi-view descriptors for 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1919–1928, 2020. 2

[67] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14658–14667, 2020. 1, 2

[68] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, 2020. 5

[69] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[70] 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. 1, 2, 6, 10

[71] 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. 7

[72] Roee Litman and Alexander M Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2013. 2

[73] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation, 2021. 5

[74] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *2013 IEEE International Conference on Computer Vision*, pages 2200–2207, 2013. 5

[75] Riccardo Marin, Souhaib Attaiki, Simone Melzi, Emanuele Rodolà, and Maks Ovsjanikov. Why you should learn functional basis, 2021. 2

[76] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G Kim, and Yaron Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph.*, 36(4):71–1, 2017. 2, 11

[77] 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. 2, 11

[78] Christos Matsoukas, Johan Fredin Haslum, Moein Sorkhei, Magnus Söderberg, and Kevin Smith. What makes transfer learning work for medical images: Feature reuse & other factors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9225–9234, 2022. 1

[79] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Matching humans with different connectivity. *Eurographics Workshop on 3D Object Retrieval*, 2019. 6

[80] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout. *ACM Transactions on Graphics*, 38(6):1–14, Nov 2019. 3, 9

[81] 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. 1

[82] 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. 2, 6

[83] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library.

[84] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*, 2(1):15–36, 1993. 5

[85] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, Jan. 1993. 10

[86] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. *Advances in Neural information processing systems*, 31, 2018. 4

[87] Helmut Pottmann, Johannes Wallner, Qi-Xing Huang, and Yong-Liang Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26(1):37–60, 2009. 1

[88] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 1, 2

[89] Adrien Poulenard, Marie-Julie Rakotosaona, Yann Ponty, and Maks Ovsjanikov. Effective rotation-invariant point cnn with spherical harmonics kernels. In *2019 International Conference on 3D Vision (3DV)*, pages 47–56. IEEE, 2019. 7, 11

[90] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 3, 7, 9, 11

[91] Can Qin, Haoxuan You, Lichen Wang, C. C. Jay Kuo, and Yun Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation, 2019. 5

[92] 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. 2, 3

[93] 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. 3, 6

[94] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, Feb. 2016. 7

[95] 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. 2

[96] 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. 2, 3, 6

[97] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011. 10

[98] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and

In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019. 10

H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19264–19275. Curran Associates, Inc., 2020. 3, 6

[99] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. DiffusionNet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022. 1, 2, 6, 7

[100] 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. 1, 2, 6

[101] 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. 2, 6, 10

[102] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. Patchnets: Patch-based generalizable deep implicit 3d shape representations. *Lecture Notes in Computer Science*, pages 293–309, 2020. 3

[103] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 1, 3

[104] Yiqun Wang, Jing Ren, Dong-Ming Yan, Jianwei Guo, Xiaopeng Zhang, and Peter Wonka. Mgcn: descriptor learning using multiscale gcns. *ACM Transactions on Graphics (TOG)*, 39(4):122–1, 2020. 2

[105] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019. 2

[106] Yue Wang and Justin M Solomon. PRNet: Self-supervised learning for partial-to-partial registration. *arXiv*, 2019. 2

[107] 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. 11

[108] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. Cnns on surfaces using rotation-equivariant features. *ACM Transactions on Graphics (TOG)*, 39(4):92–1, 2020. 2

[109] Aoran Xiao, Jiaxing Huang, Dayan Guan, and Shijian Lu. Unsupervised representation learning for point clouds: A survey. *arXiv preprint arXiv:2202.13589*, 2022. 1

[110] 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. 1, 3, 4, 6, 8

[111] Zhangsihao Yang, Or Litany, Tolga Birdal, Srinath Sridhar, and Leonidas Guibas. Continuous geodesic convolutions for learning on 3d shapes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 134–144, 2021. 11

[112] Zi Jian Yew and Gim Hee Lee. 3DFeat-Net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 607–623, 2018. 2

[113] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020. 2

[114] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu Horaud. Surface feature detection and description with applications to mesh matching. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–380. IEEE, 2009. 2

[115] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 2, 3, 4, 6

[116] Nanxuan Zhao, Zhirong Wu, Rynson W. H. Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1

[117] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 1