Field Convolutions for Surface CNNs

Thomas W. Mitchel
Johns Hopkins University
tmitchel@jhu.edu

Vladimir G. Kim
Adobe Research
vokim@adobe.com

Michael Kazhdan
Johns Hopkins University
misha@cs.jhu.edu

Abstract

We present a novel surface convolution operator acting on vector fields that is based on a simple observation: instead of combining neighboring features with respect to a single coordinate parameterization defined at a given point, we have every neighbor describe the position of the point within its own coordinate frame. This formulation combines intrinsic spatial convolution with parallel transport in a scattering operation while placing no constraints on the filters themselves, providing a definition of convolution that commutes with the action of isometries, has increased descriptive potential, and is robust to noise and other nuisance factors. The result is a rich notion of convolution which we call field convolution, well-suited for CNNs on surfaces. Field convolutions are flexible, straight-forward to incorporate into surface learning frameworks, and their highly discriminating nature has cascading effects throughout the learning pipeline. Using simple networks constructed from residual field convolution blocks, we achieve state-of-the-art results on standard benchmarks in fundamental geometry processing tasks, such as shape classification, segmentation, correspondence, and sparse matching.

1. Introduction

The advent of deep learning in imaging, vision, and graphics has coincided with the development of numerous techniques for the analysis and processing of curved surfaces based on convolutional neural networks (**CNNs**). The challenge in reproducing the success of CNNs on surfaces is that classical notions of convolution and correlation in Euclidean spaces cannot simply be transposed onto curved domains. Unlike images, points on a surface have no canonical orientation, without which simple operations fundamental to the spatial propagation of information, such as moving dot products, cannot be computed in a repeatable manner.

Geometric deep learning is a young field, and many successful methods can be broadly categorized in relation to two emerging paradigms characterized by specific approaches to convolution: *diffusive* propagation and *equivariant* propagation. Diffusive approaches closely intertwine convolution operations with heat diffusion on manifolds wherein filters represented by anisotropic heat kernels or Gaussians are used to propagate scalar features [35, 6, 5, 39, 31, 47]. In contrast, equivariant convolutions distribute vector or tensor features that transform with local coordinate systems [41, 55, 46, 7, 9, 61, 62].

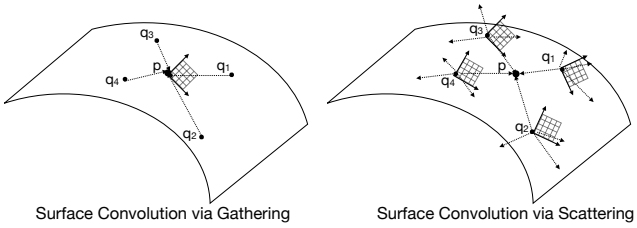

Figure 1. Prior approaches define patch-based convolution operators as *gathering* operations (left), which are sensitive to noise or disruptions in the local coordinate system. Field convolution is a *scattering* operation and is robust under perturbations as it does not rely on a single coordinate system to aggregate features.

Critically, virtually all state-of-the-art approaches sacrifice filter descriptiveness to define a notion of convolution that does not depend on the choice of local coordinate frames. Gaussian filters can facilitate efficient evaluations in the spectral domain but are individually undiscriminating. Extensions to anisotropic filtering mitigate these limitations by extending the class of filters that can be used. However, this requires defining a frame field over the surface – itself a hard problem.

Equivariant approaches have the potential to provide expressive notions of convolution on surfaces due to the encoding of geometric information in the transport of tangent vector features. However, equivariance of the response is almost universally achieved by placing constraints on the filters themselves [41, 42, 8, 9, 25, 61], limiting descriptiveness and necessitating complex architectures to support the algebraic relationships between kernels. Furthermore, these regimes formulate spatial propagation as a *gathering* operation, analogous to correlations on Euclidean domains; features are weighted based on their position relative to a coordinate frame defined at a single point (Figure 1, left),

making them sensitive to inconsistencies or disruptions in local parameterizations .

In this paper we present a novel convolution operator acting on vector fields. Our method defines the value of the convolution at a point p using a simple observation: instead of combining neighboring features by parameterizing each neighbor q_i with respect to a coordinate frame defined at p, each neighbor q_i parametrizes p within its own coordinate frames (Figure 1, right). This formulation combines intrinsic spatial weighting with parallel transport *while placing no constraints on the filters themselves*, providing a definition of convolution that commutes with the action of isometries and has increased descriptive potential. In addition, as a *scattering* operation, it is less sensitive to noise and other nuisance factors as it does not rely on a single coordinate system about each point to aggregate features. The result is a rich notion of convolution which we call *field convolution* (**FC**), well-suited for CNNs on surfaces.

Field convolutions are flexible and straight-forward to incorporate into surface learning frameworks. Their highly discriminating nature has cascading effects throughout the learning pipeline, allowing us to achieve state-of-the-art results on standard benchmarks in applications including shape classification, segmentation, correspondence, and sparse matching. All code and evaluations are publicly available at github.com/twmitchel/FieldConv.

2. Related Work

The field of geometric deep learning has grown extensively since its inception half a decade ago. Here, we only review the techniques most closely related to ours – those designed specifically for the analysis of 3D shapes. Generally speaking, these methods exist on a spectrum between extrinsic and intrinsic techniques, with the former performing signal processing using the embedding of the surface in 3D and the latter only using the Riemannian structure.

Point-based methods offer a purely extrinsic framework for applying deep learning to 3D shapes by representing them in terms of point clouds. A majority of these approaches can trace their lineage to the influential PointNet [43] and PointNet++ architectures [44] and recent approaches such as DGCNN [60], PCNN [3], KPCNN [54], TFN [55], QEC [64] and SPHNet [42] have sought to extend the framework by incorporating connectivity information, dynamic filter parameterizations, and equivariance to rigid transformations. Convolution is typically expressed by applying radially isotropic filters over local 3D neighborhoods and aggregating the results with the maximum or summation operations. This approach offers a simple foundation for extremely flexible and noise-robust networks, though at the expense of descriptive potential. More generally, these methods tend to struggle in the presence of non-rigid isometric deformations, making them less effective in scenarios like deformable shape matching [11, 19, 47].

Representational approaches sit between extrinsic and intrinsic techniques. These methods exploit the data's underlying connectivity to form convolutional operators, often making use of well-developed techniques for graph-based learning on irregular structures [10, 63, 58, 14, 30, 18, 8, 29]. In particular, convolutions are performed using filters defined relative to the explicit graph structure as functions on edges or vertices, often with only immediate local support such as the surrounding one-ring or half-edge. A particularly notable example is MeshCNN [23], which specifically leverages the ubiquitous representation of surfaces as triangle meshes to construct a similarity-invariant convolution operator propagating edge-based features. While this enables graph-based convolutions to better handle non-rigid deformations compared to point-based approaches, it also makes them sensitive to changes in connectivity.

One approach to defining *intrinsic* convolution has been to parametrize the surface over a simple domain such as the the sphere [22], torus [34], or plane [50] where standard CNNs can be applied. However, such parameterizations depend on the genus and often exhibit significant distortion.

A second class of approaches has been to define intrinsic convolution over the Riemannian manifold, and can generally be classified in relation to two emerging paradigms: *diffusive* convolutions and *equivariant* convolutions. In the former, convolution operations are closely related to heat diffusion on surfaces wherein heat (e.g. Gaussian) kernels are used to propagate scalar features. While early diffusive approaches including GCNN [35], ADD [5], ACNN [6] and MoNet [39] perform convolutions over local patches, recent state-of-the-art networks ACSCNN [31] and DiffusionNet [47] represent convolution in the spectral domain. Despite their success in a variety of scenarios, most notably in dense shape correspondence [19, 11, 31, 47], these methods face an intractable problem: radially symmetric filters are individually undiscriminating and diffusive frameworks are not naturally suited to handle the orientation ambiguity problem introduced by the use of more descriptive, anisotropic kernels. To compensate, these methods supplement convolutions with basic orientation-aware operations on tangent vector features [47] in addition to employing various strategies that are either fragile, such as aligning kernels along the directions of principal curvature [6, 39], or discarding information by pooling over samplings of orientations or by specifying directions of maximum activation[35, 31].

Recently, several techniques have been introduced for *equivariant* surface convolutions such as MDGCNN [41], GCN [7, 9] and HSN [61]. In contrast to diffusive approaches, equivariant convolutions are designed specifically to address the rotation ambiguity problem by propagating tangent vector features that transform with local coordinate systems. To make the convolution independent of

the choice of local coordinate frame, most existing methods strongly constrain the class of filters that can be used [41, 42, 8, 9, 25, 61]. An exception to this is PFCNN [62] which also discards information by pooling over multiple kernel orientations. Often, these parameterizations are so restrictive that they necessitate complex network architectures to be effective: even the state-of-the-art HSN [61] is formulated as a multi-stream U-Net with various pooling operations. Furthermore, in moving from radially isotropic to anisotropic filters, prior equivariant regimes universally formulate spatial propagation as a *gathering* operation wherein all features in the local surface are weighted based on their position in a *single* coordinate system. While this approach may seem natural as it is analogous to correlation on Euclidean domains, a feature's dependence on a single local parameterization increases sensitivity to noise.

3. Method Overview

Field convolutions are closely related to an operation called *extended convolution*, which allows a filter to adaptively transform as it travels over a Euclidean domain or manifold [37]. In the latter case, it forms the basis for the recently proposed ECHO descriptor [38], which has been shown to significantly outperform SHOT [56] and other hand-crafted descriptors in terms of overall descriptiveness and robustness to a variety of nuisance factors.

In particular, field convolutions combine extended convolution on surfaces with parallel transport, resulting in a mapping between vector fields that only depends on the Riemannian metric. Given a feature vector field X and a filter f, the construction of the field convolution is straightforward: At each point $p \in M$ on the surface, values of X in the surrounding neighborhood $q \in \mathcal{N}_p$ are weighted relative to the position of p in the frame determined by $X(q)$, transported to p, and aggregated. This approach is agnostic to connectivity and is robust, as the assignment of weights with respect to multiple coordinate systems makes it naturally insensitive to noise and other nuisance factors. Most importantly, no constraints are placed on filters.

Field convolutions facilitate the construction of highly discriminating yet simple networks, without the need for pooling, normalization, or specialized architecture. The principal module in applications is the field convolution ResNet (**FCResNet**) block, consisting of two successive field convolutions with a residual connection between the input and output layers [24]. FCResNet blocks are self-contained and flexible, and can easily be incorporated into isometry-invariant surface learning regimes. In addition, we leverage the connection between field convolutions and the recently proposed state-of-the-art ECHO surface descriptor [38] to construct a novel final layer specifically designed for labeling tasks with isometry-invariant surface networks, which we refer to as an ECHO block. This block takes vector field channels as input, mapping them to scalar ECHO descriptors which are then fed through an MLP to make predictions, essentially converting the problem to one of image classification in the final layer of the network.

4. Field Convolution

Following the approach of Knoppel *et al.* [28], we represent tangent vectors as complex numbers. Given a surface M, at any point $p \in M$ we can assign to the tangent space T_pM an orthonormal basis $\{\mathbf{e}_1, \mathbf{e}_2\}_p$. Then T_pM can be associated with $\mathbb{C}$ such that for any $\mathbf{v} \in T_pM$, we have $\mathbf{v} \equiv r\, e^{i\theta}$, with $r = |\mathbf{v}|$ and θ the angle between $\mathbf{v}$ and $\mathbf{e}_1$.

Letting $\Gamma(TM)$ be the space of vector fields on M, we express the evaluation of a vector field $X \in \Gamma(TM)$ at a point $p \in M$, in terms of the frame $\{\mathbf{e}_1, \mathbf{e}_2\}_p$, as

$$X(p) \equiv \rho_p\, e^{i\phi_p}. \tag{1}$$

Similarly, for two points $p,\ q \in M$ we denote the logarithm of p with respect to q, giving the "position" of p in T_qM, as

$$\log_q p \equiv r_{qp}\, e^{i\theta_{qp}}. \tag{2}$$

We denote by φ_{pq} the change in angle resulting from the parallel transport $\mathcal{P}_{p\leftarrow q} : T_qM \to T_pM$ along the shortest geodesic from q to p, such that for any $\mathbf{v} \in T_qM$,

$$\mathcal{P}_{p\leftarrow q}(\mathbf{v}) \equiv e^{i\varphi_{pq}}\, \mathbf{v}. \tag{3}$$

We consider filters belonging to the space of square integrable functions on the complex plane, and define the *field convolution* of a vector field $X \in \Gamma(TM)$ with a filter $f \in L^2(\mathbb{C})$ to be the vector field in $\Gamma(TM)$ with

$$\big(X * f\big)(p) = \int_M \rho_q\, e^{i(\phi_q + \varphi_{pq})}\, f\left(r_{qp}\, e^{i(\theta_{qp} - \phi_q)}\right)\, dq. \tag{4}$$

The first term is the parallel transport of the tangent vector $X(q)$ to T_pM and the second term is the evaluation of the filter at the coordinates of $\log_q p$, expressed relative to the frame $\left\{X(q)/\|X(q)\|, X^{\perp}(q)/\|X(q)\|\right\}$.[1]

In practice, we use filters compactly supported within a radius of ϵ and limit the domain of integration to the geodesic ϵ-ball about p.

Finally, noting that isometries preserve areas, and commute with the action of parallel transport and the logarithm [16], it follows that if $\Psi : M \to N$ is an isometry, we have

$$d\Psi\left[\big(X * f\big)(p)\right] = \left[d\Psi(X) * f\right]\big(\Psi(p)\big). \tag{5}$$

Or in other words, field convolutions commute with the action of isometries. A detailed proof of this claim can be found in Supplement A.

[1] Although the frame is undefined when $\rho_q = 0$, the integral remains well-defined as the value of the filter is multiplied by ρ_q in the integrand.

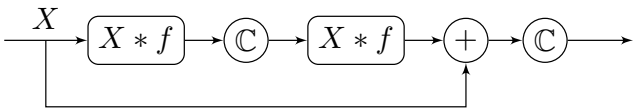

Figure 2. The FCResNet block. Here $\mathbb{C}$ denotes the complex ReLU in Equation (8).

Discretization In practice, we discretize a surface M by a triangle mesh with vertices V. To every $p \in V$, we associate the collection of vertices $\mathcal{N}_p \subset V$ belonging to the geodesic ϵ-ball about p. At each point, real-valued filters $f \in L^2(\mathbb{C})$ are supported on $\log_p(\mathcal{N}_p) \subset \mathbb{C}$, and parameterized as sums of angular frequencies with band-limit B. That is, for any $z = re^{i\theta} \in \mathbb{C}$ with $|r| \leq \epsilon$, the evaluation of f at z is expressed as

$$f(z) = \sum_{m=-B}^{B} f_m(r) \cdot e^{im\theta} \tag{6}$$

where $f_m(r) \in \mathbb{C}$ is the m-th Fourier coefficient of f, restricted to radius r and $f_{-m}(r) = \overline{f_m(r)}$ because f is real-valued. We discretize the function $f_m(r)$ using linear interpolation, setting $f_m(r) = \mathbf{r}^\top(r)\,\mathbf{f}_m$, where $\mathbf{r}(r) \in \mathbb{R}^N$ is the vector of linear interpolation weights (with $\mathbf{r}_i(r) \neq 0$ only if $i \in \{\lfloor rN/\epsilon \rfloor, \lceil rN/\epsilon \rceil\}$) and $\mathbf{f}_m \in \mathbb{C}^N$ is the vector of Fourier coefficients at the discrete radii.

Then, letting $\{w_p\} \subset \mathbb{R}_{>0}$ denote the area weights associated with vertices $p \in V$, choosing an arbitrary edge at every vertex to define a frame, and letting $X \in \mathbb{C}^{|V|}$ be a discrete vector field, the evaluation of the field convolution $X * f$ as in Equation (4) at a vertex $p \in V$ is given by

$$\big(X * f\big)(p) = \sum_{\substack{q \in \mathcal{N}_p \\ |m| \leq B}} w_q\, \rho_q\, e^{i(\phi_q + \varphi_{pq})}\, f_m(r_{qp})\, e^{im(\theta_{qp} - \phi_q)}. \tag{7}$$

The values of w_q, φ_{pq}, r_{qp}, and θ_{qp}, corresponding to the weight, transport change of angle, geodesic distance, and logarithm for each $p \in V$ and $q \in \mathcal{N}_p$ can be precomputed to speed up training. Similar to [61], we apply rotational offsets $e^{i\beta_{|m|}}$ to the coefficients corresponding to each frequency, providing additional learned degrees of freedom.

5. Surface CNNs with Field Convolutions

Field convolutions are the principle contribution of this work as they provide a robust and descriptive framework for the spatial propagation of information on surfaces. The goal of this section is to introduce the fundamental building blocks for incorporating field convolutions into isometry-invariant surface learning paradigms.

FCResNet Blocks The atomic unit for field convolutions in surface CNN frameworks is the FCResNet block, which consists of two field convolutions each followed by a non-linearity and a residual connection between the input and output streams (Figure 2). They are entirely self-contained, and map vector field features to vector field features without relying on any supporting or complementary convolution operations that are a common fixture in other equivariant approaches [41, 61]. As such, they represent a flexible and descriptive layer that can be easily employed in isometry-invariant learning pipelines.

ECHO Blocks A secondary contribution of this work is the concept of an ECHO block for label-prediction tasks, which leverages the connection between vector fields and the recently proposed ECHO surface descriptor [38]. Given a scalar signal and a frame field, ECHO descriptors provide an intrinsic, isometry-invariant characterization of the local surface about a feature point in terms of the filter maximizing the response to extended convolution. This filter is constructed by having neighbors of the feature point "cast a vote", weighted by the value of the signal at that point, into the filter position corresponding to the position of the feature point, as seen from the neighbor's frame. A vector field can be used to compute ECHO descriptors at every point $p \in M$, using the magnitude and direction of each vector to define the values of the signal and transformation field at p.

The idea behind ECHO blocks is to convert feature vector fields to pointwise descriptors, turning the task of vector field classification into one of image classification in the final layer of the network. These blocks consist of two steps: 1) A field convolution layer is used to map the input feature channels to D output feature channels (with D the desired number of descriptors). These are then used to compute pointwise ECHO descriptors, resulting in H isometry-invariant scalar features per channel, where H is the number of samples used to represent the ECHO descriptor. 2) The $D \times H$ values are linearized and fed to a three-layer MLP. Like field convolutions, the computation of ECHO descriptors relies only on the logarithm map, parallel transport, and the integration weights associated with each vertex. No additional pre-processing is required.

Linearities and Non-Linearities Since we represent tangent vector features as complex numbers, we apply linearities in the form of multiplication by complex matrices in the same manner as is done for real-valued features. However, our linearities do not include translational offsets to preserve commutativity with the action of isometries.

For similar reasons, non-linearities are applied only to the radial components of features as is done in [61]. Namely, given a feature vector field $X \in \mathbb{C}^{|V|}$ we apply pointwise ReLUs with a learned offset b such that

$$\mathrm{ReLU}_b\big(X(p)\big) = \mathrm{ReLU}\left(\rho_p + b\right) e^{i\phi_p}. \tag{8}$$

FCNet: A Generic Surface CNN for Vector Fields In our experiments we use a simple, generalizable architecture we call an **FCNet**, which is simply a series of FCResNet blocks. For labeling tasks, we append an ECHO block to the end of the network to make predictions. For FCNets consisting of three or more layers, we add additional residual connections after every two FCResNet blocks as we find this significantly accelerates training. In all experiments, we take the raw 3D positions of points as inputs and use a learnable gradient-like operation (Supplement B) to map them to vector fields which are then fed to the network. We could also use the intrinsic Heat Kernel Signature [51] as input, thereby obtaining a fully isometry-invariant pipeline. However, as demonstrated by Sharp *et al.* [47], the 3D coordinates work as well in practice and are easier to compute.

Despite this elementary construction, we show that FCNets achieve state-of-the-art results in a variety of fundamental geometry processing tasks.

6. Evaluation

We compare our method against leading surface learning paradigms on four benchmarks corresponding to fundamental tasks in geometry processing: classification, segmentation, correspondence, and feature matching.

6.1. Implementation

Our framework is implemented using PyTorch Geometric [15]. We employ the same, simple FCNet architecture discussed in Section 5 in all of our experiments, varying the number of FCResNet blocks based on task complexity. For label-prediction tasks on large datasets, we append an ECHO block to the end of the network to make predictions. Otherwise we use the magnitudes of the output feature vectors.

As input, we take the 3D coordinates, which are lifted to 16 tangent vector features in the initial gradient layer, followed by either 32 or 48 features in the FCResNet stream. We use the ADAM optimizer [27] to a cross-entropy loss with an initial learning rate of 0.01 and a batch size of 1. We randomly rotate all inputs to ensure there are no consistencies in the spatial embedding of shapes.

Our pre-processing regime parallels [61], omitting the operations necessary to support their multi-scale and pooling operations. All shapes are normalized to have unit surface area and we use the Vector Heat Method [49] to compute the geodesic ϵ-ball $\mathcal{N}_p \subset V$ corresponding to each vertex $p \in V$, in addition to the logarithm and parallel transport associated with each edge $(p, q) \in \{p\} \times \mathcal{N}_p$. Area weights are assigned in the standard way, using one third of the vertex's one-ring area, and are normalized by the sum of the weights within the geodesic ϵ-ball. While we process shapes as triangle meshes in our experiments, we note that recent work by Sharp *et al.* [48] has made possible efficient computations of logarithmic parameterizations and vector transport on point clouds, with which our method can be extended to analyze point cloud shape data.

Method	Accuracy
FC (ours)	**99.2%**
DiffusionNet [47]	98.9%
MeshWalker [29]	97.1%
HSN [61]	96.1%
MeshCNN [23]	91.0%
GWCNN [13]	90.3%

Table 1. Classification accuracy on the SHREC '11 Dataset [32] .

Method	# Features	Accuracy
FC (ours)	3	**92.9%**
MeshWalker [29]	NA	92.7%
MeshCNN [23]	5	92.3%
DiffusionNet [47]	16	91.5%
HSN [61]	3	91.1%
SNGC [22]	3	91.0%
PointNet++ [43]	3	90.8%

Table 2. Segmentation accuracy on the composite dataset of [34].

6.2. Classification

First, we use an FCNet with two FCResNet blocks to classify meshes in the SHREC '11 dataset [32], containing 30 shape categories. Filters are supported on geodesic neighborhoods of radius $\epsilon = 0.2$ and are parameterized using $N = 6$ radial samples with band-limit $B = 2$. Due to the small scale of the task, we omit the ECHO block in the final layer and instead use a global mean pool over the feature magnitudes to give a prediction. As in prior works [23, 61, 47], we train on 10 samples per class and report results over three random samplings of the training data. Our FCNet converges quickly, and we train on just 30 epochs – far fewer than the 100 or more used in previous work.

Results are shown in Table 1. Due to the wide adoption of the dataset, we only list the results of methods achieving a classification accuracy of 90% or higher. Our simple FCNet achieves the highest reported accuracy, reaching a classification rate of 100% on two of the three random samplings of the training data. Like HSN and DiffusionNet who also report high classification accuracy, our FCNet uses relatively few parameters compared to other networks and is agnostic to both mesh connectivity and isometric deformations – all providing a significant advantage on the SHREC '11 dataset which has a small number of training samples and consists of poor-quality meshes with in-class deformations mainly limited to rigid articulations. The superior performance of our FCNet is likely due to the descriptiveness of field convolutions, as HSN uses specially parameterized filters.

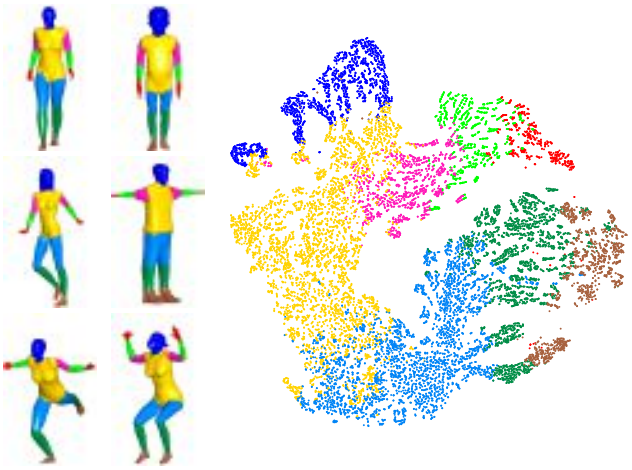

Figure 3. We visualize the descriptors computed in our FCNet's ECHO block in the segmentation task. Left: Models from the test split of the composite dataset [34], color-coded by ground-truth labels. Right: 2D projections of the descriptors using t-SNE [57].

6.3. Segmentation

Next, we apply our field convolution framework to the task of human body segmentation, using the dataset proposed by [34], which consists of a composite of various human shape datasets [1, 2, 17, 59, 4]. The varied nature of the collection of models in terms of human subjects, acquisition method, and connectivity serve to test both descriptiveness and robustness to variety of nuisance factors.

We use an FCNet with four successive FCResNet blocks ($N = 6$, $B = 2$, $\epsilon = 0.2$) followed by an ECHO block, trained to predict a body part annotation for each point on the mesh. The ECHO block computes $D = 32$ descriptors with $H = 33$ samples (corresponding to three samples per geodesic radius) for a total of 1056 scalar feature channels. The three-layer MLP first maps these features to 256 channels, then 124, and finally to the desired number of output channels. Due to the large number of vertices per model, we downsample each mesh to 1024 vertices using farthest point sampling, an approach also used by [23, 61]. Our network converges quickly and we train for only 15 epochs with a label smoothing regularization [53] factor of 0.2.

Results in the form of the percentage of correctly classified vertices across all test shapes are shown in Table 2. As in the classification experiments, we only list the results of methods that achieve a segmentation accuracy of 90% or higher on the dataset. Again, our basic network achieves state-of-the-art results, outperforming all other methods with a minimal number of input features. The improvement due to field convolutions is especially evident relative to other techniques that employ surface convolutions, such as HSN [61] and DiffusionNet [47] approaches.

To understand the features learned by our network, we use t-SNE[57, 40] to visualize the descriptors computed in the ECHO block for all models in the test dataset, color-coded using the ground-truth labels (Figure 3). We observe a distinct clustering of points, not only corresponding to similarly labeled regions but also reflecting the connectivity between adjacent regions on the meshes. This suggests that our FCNet is able to learn at least some measure of intrinsic similarities between shapes, despite starting with the extrinsic 3D coordinates as input.

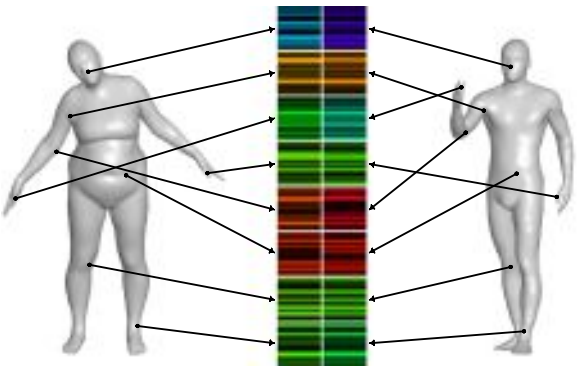

Figure 4. FCNet features at corresponding points on models in the remeshed FAUST dataset [11]. Features are drawn using the HSV scale – hue encodes the absolute magnitude and value encodes the relative magnitude with saturation fixed at one.

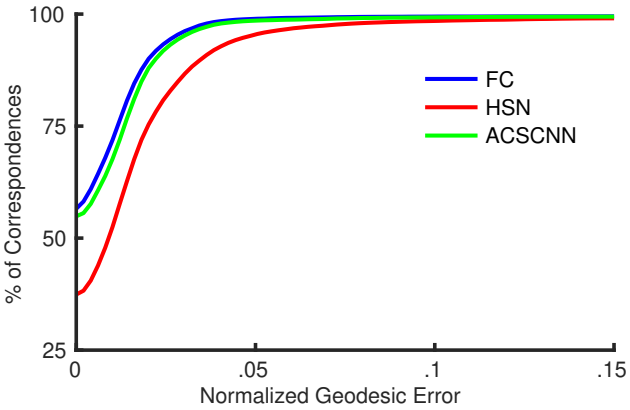

Figure 5. Percentage of correspondences for a given geodesic error on the remeshed FAUST dataset using field convolutions (FC), HSN, and ACSCNN.

6.4. Correspondence

Here we use an FCNet to find pointwise correspondences between similar shapes. Over the last half-decade, the FAUST dataset [4] has become the *de facto* standard for evaluating network performance in correspondence tasks and many recent approaches have achieved near-perfect accuracy on the dataset [14, 9, 31]. However, shapes in the dataset share the same connectivity, and there has been some question as to whether these methods have primarily learned the mesh graph structure, rather than deformation-invariant characterizations of the shape themselves [47]. To

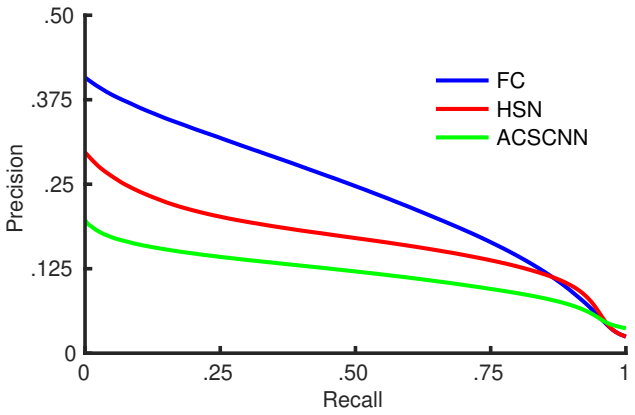

Figure 6. Results of feature matching evaluations on the SHREC 2019 Isometric and Non-Isometric Shape Correspondence dataset [12] in the form of the mean precision-recall curves.

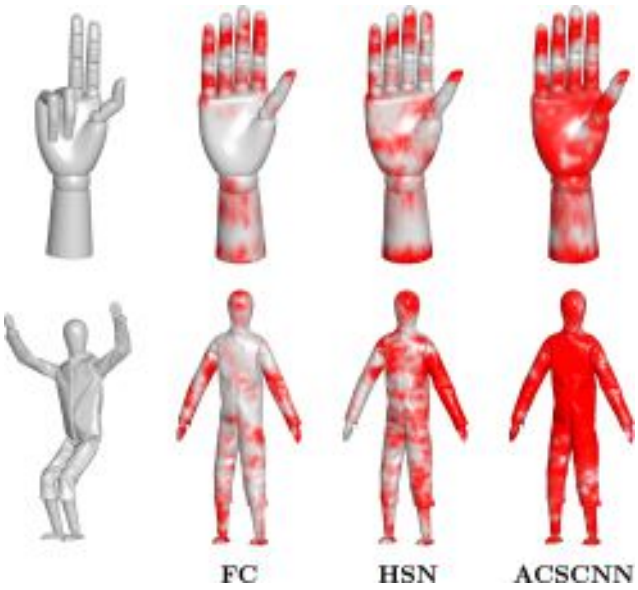

Figure 7. Feature-space distances: For each feature from the models on the right, we rank order the features on the model to the left by feature distance. Vertices are then colored from gray to red, showing how deeply one must traverse the rank ordered list before encountering a corresponding feature.

this point, we perform evaluations on a fully remeshed version of the dataset [11], a more challenging task better representative of real-world applications. As in prior work, we train on the first 80 models out of the 100 in the dataset, and use the remainder for testing.

We train an FCNet to predict the indices of corresponding vertices on a template shape. Due to the degree of precision required by this task, we use a deeper network, consisting of eight FCResNet blocks ($N = 3$, $B = 1$, $\epsilon = 0.05$) followed by an ECHO block ($D = 12$, $H = 13$) with a 124–64–32 MLP, and additional residual connections after every two FCResNet blocks. To make predictions, we add two linear layers after the ECHO block, taking the 32 features first to 256 channels, and then to the number of vertices on the template shape, with a $p = 0.5$ dropout layer in-between. Visualizations of some of the 32 channel features in our FCNet at the bottleneck before the dense final layers are shown in Figure 4.

Prior methods have typically used high-dimensional SHOT [56] descriptors as inputs for this task, which we feel to be unnecessary due to the expressiveness of the field convolution framework. As such, we train HSN and ACSCNN [31] with raw 3D coordinates inputs for comparison – two recent methods which have reported state-of-the-art results in similar classification tasks. The results are shown in Figure 5, giving the percentage of total correspondences as a function of the normalized geodesic error. Our FCNet achieves the best performance, followed by ACSCNN.

Recent spectral-based networks, ACSCNN and DiffusionNet [47], have significantly outperformed comparable equivariant networks in correspondence related tasks. This is likely for two reasons: 1) In contrast to the local patch-based convolution operators used in equivariant networks, spectral-based convolutions are formulated in a Laplace-Beltrami basis, providing an inherently global characterization of shape less sensitive to point-wise noise or local mislabeling; 2) The ability to essentially band-limit convolutions by working in basis of low-frequency eigenfunctions allows spectral-based networks to easily scale to high resolutions whereas equivariant networks must decrease both filter support and the number of parameters to process the same meshes. This makes the performance of our FCNet particularly notable as it suggests that the network is able to overcome the relative limitations of equivariant frameworks in dense correspondence tasks specifically due to the robust construction of field convolutions as a scattering operation – keeping them stable despite smaller supports – and due to their descriptiveness, the latter of which does not diminish significantly even with fewer parameters.

6.5. Feature Matching

Last, we train an FCNet to compute point-wise surface feature descriptors on shapes from the SHREC 2019 Isometric and Non-Isometric Shape Correspondence dataset [12]. The dataset consists of 50 meshes constructed from 3D scans of a jacketed humanoid figurine and a bare and gloved articulated wooden hand with 76 pre-defined pairs of meshes. We consider this dataset to be extremely challenging with significant non-isometric deformations and topological changes between pairs; as real-world scans the meshes also contain noise, varying triangulations, occluded geometry and various other sources of interference.

To ensure an even distribution of meshes in both the training and testing data, we group all pairs into three categories based on scan source (humanoid, hand, and gloved

hand) and randomly select 20% of the pairs in each category to form the test split. We randomly sample 2048 points on both meshes in each pair and use the ground truth correspondence to assign corresponding and non-corresponding points. We learn compact, 16-dimensional descriptors at each point using a twin network [33, 35, 52], where each mesh in a pair is processed by the same network and a twin loss function is minimized, weighting the descriptor distances between corresponding and non-corresponding points. We use precision-recall curves to evaluate performance on the test pairs, as they have been shown to well-characterize feature descriptiveness [26, 36] and are the standard metric in the surface feature descriptor literature [56, 45, 21, 20, 38]. A detailed explanation of our experimental regime can be found Section C of the supplement.

We train an FCNet consisting of eight FCResNet blocks ($N = 6, B = 1, \epsilon = 0.1$) on the downsampled 2048-vertex mesh pairs, using the magnitudes of the output features as point-wise descriptors. HSN and ACSCNN are trained on the downsampled and full-resolution meshes, respectively. We report results averaged over three random samplings of the test-train split (Figure 6); to ensure fair comparisons, we compute the average precision-recall curves over all test pairs using the same set of correspondences for all methods. Our FCNet achieves the best performance by a significant margin, followed by HSN. The difference is likely explained by the increased descriptiveness of field convolution and its robust formulation as a scattering operation, making it better able to characterize flat, featureless areas (Figure 7, palm of the hand) and insensitive to high-frequency perturbations of the surface (Figure 7, folds in the figurine jacket), as compared to the gathering-based convolution operations used by HSN which rely on strongly constrained filters. We believe that ACSCNN under-performs relative to the other methods because methods like ACSCNN which depend on the (global) spectral decomposition of the Laplace-Beltrami operator are less stable in the presence of non-isometric deformations, geometric occlusions, and changes in topology between corresponding pairs. While still not giving excellent performance, methods like FCNet and HSN, which use filters with local support, tend to be more robust.

6.6. Performance

Field convolutions are among the most efficient equivariant convolution operations, requiring few parameters per convolution operation. On an RTX 2080 GPU and 3.8 GHz CPU, our deepest FCNet trains at approximately 3 min/epoch on the full resolution meshes in the dense correspondence task. Field convolutions use a similar number of parameters as HSN [61] per convolution and with half the memory footprint. HSN's multi-stream convolutions learn a weight matrix for the radial profile and rotational offset corresponding to each stream and the connections between them, resulting in $(N + 1)\, S^2$ total parameters per convolution, with N the number of radial samples and S the number of streams. Similarly, we learn a complex radial profile and rotational offset for each non-negative frequency up to the number of band-limited frequencies with $N(2B + 1) + B + 1$ total parameters per convolution. In the classification and segmentation experiments, HSN reports results using $N = 6$ radial samples and $M = 2$ streams resulting in 28 total parameters per convolution. In the same experiments, our FCNet achieves state-of-the-art performance with 33 total parameters per convolution, as we use filters with band-limit $B = 2$ and the same number of radial bins. However, HSN stores features for both streams, increasing spatial complexity by a factor of two.

More generally, we see our state-of-the-art results on the segmentation task using the composite dataset [34] as particularly notable in that other top-performing methods, including MeshCNN [23] and HSN, use the deepest versions of their network for this task despite the small number of labels involved (eight classes), presumably because of the large size of the training dataset. Our FCNet outperforms these networks with a much shallower architecture and only in the dense correspondence and feature matching tasks – both of which involve learning granular distinctions between large numbers of similar points – do we increase the depth of our network. This suggests that unlike most networks, the depth of an FCNet (or other network built on field convolutions) necessary to achieve good performance is not strongly dependent on the size of the dataset, and scales primarily with task complexity.

7. Conclusion

We present a novel definition of surface convolution acting on vector fields, combining invariant spatial weighting with the parallel transport of features in a scattering operation *while placing no constraints on the filters themselves*. This formulation is highly descriptive, insensitive to a variety of nuisance factors, and straight-forward to implement; with it, we construct simple networks that achieve state-of-the-art results in fundamental geometry-processing tasks.

While the complexity of our method is comparable to existing equivariant approaches, it shares the same drawbacks as filter supports and parameter counts must be limited to process meshes at full resolution. More generally, existing successful surface learning frameworks (including ours) are designed to handle only isometric or nearly-isometric shape deformations and fail to achieve adequate performance in the presence of the kinds of complex deformations, geometric occlusions, and topological changes found in real shape data. In the future, we plan to expand our framework to handle more challenging classes of deformations, beginning with invariance to conformal automorphisms.

References

[1] Adobe. Adobe mixamo 3D characters, 2016. www.mixamo.com. 6

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. *Transactions on Graphics*, 24(3):408–416, 2005. 6

[3] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018. 2

[4] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition*. IEEE, 2014. 6

[5] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 3189–3197, 2016. 1, 2

[6] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library, 2016. 1, 2

[7] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *International conference on Machine learning*, pages 1321–1330. PMLR, 2019. 1, 2

[8] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *International conference on Machine learning*, volume 97, pages 1321–1330, 2019. 1, 2, 3

[9] Pim de Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. *arXiv preprint arXiv:2003.05425*, 2020. 1, 2, 3, 6

[10] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, page 3844–3852, 2016. 2

[11] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Computer Vision and Pattern Recognition*, pages 8592–8601, 2020. 2, 6, 7

[12] R. M. Dyke, C. Stride, Y.-K. Lai, P. L. Rosin, M. Aubry, A. Boyarski, A. M. Bronstein, M. M. Bronstein, D. Cremers, M. Fisher, T. Groueix, D. Guo, V. G. Kim, R. Kimmel, Z. Lähner, K. Li, O. Litany, T. Remez, E. Rodolà, B. C. Russell, Y. Sahillioğlu, R. Slossberg, G. K. L. Tam, M. Vestner, Z. Wu, and J. Yang. Shape correspondence with isometric and non-isometric deformations. In Silvia Biasotti, Guillaume Lavoué, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 7

[13] Danielle Ezuz, Justin Solomon, Vladimir G. Kim, and Mirela Ben-Chen. GWCNN: A metric alignment layer for deep shape analysis. *Computer Graphics Forum*, 36(5):49–57, 2017. 5

[14] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. SplineCNN: Fast geometric deep learning with continuous B-spline kernels. In *Computer Vision and Pattern Recognition*, pages 869–877, 2018. 2, 6

[15] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 5

[16] Jean H Gallier and Jocelyn Quaintance. *Differential Geometry and Lie Groups: A Computational Perspective*, volume 12. Springer Nature, 2020. 3

[17] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. Shape retrieval contest 2007: Watertight models track. *SHREC competition*, 8(7), 2007. 6

[18] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Computer Vision and Pattern Recognition Workshops*, 2019. 2

[19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *European Conference on Computer Vision.*, pages 230–246, 2018. 2

[20] 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*, 116:66–89, 2016. 8

[21] Yulan Guo, Ferdous Sohel, Mohammed Bennamoun, Min Lu, and Jianwei Wan. Rotational projection statistics for 3D local surface description and object recognition. *International Journal of Computer Vision*, 105:63–86, 2013. 8

[22] Niv Haim, Nimrod Segol, Heli Ben-Hamu, Haggai Maron, and Yaron Lipman. Surface networks via general covers. In *International Conference on Computer Vision*, pages 632–641, 2019. 2, 5

[23] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: A network with an edge. *Transactions on Graphics*, 38(4):90, 2019. 2, 5, 6, 8

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[25] Wenchong He, Zhe Jiang, Chengming Zhang, and Arpan Man Sainju. *CurvaNet: Geometric Deep Learning Based on Directional Curvature for 3D Shape Analysis*, page 2214–2224. Association for Computing Machinery, New York, NY, USA, 2020. 1, 3

[26] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition*, volume 2, pages 506–513. IEEE, 2004. 8

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[28] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. Globally optimal direction fields. *Transactions on Graphics*, 32(4), 2013. 3

[29] Alon Lahav and Ayellet Tal. Meshwalker: Deep mesh understanding by random walks. *Transactions on Graphics*, 39(6):1–13, 2020. 2, 5

[30] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *Transactions on Signal Processing*, 67(1):97–109, 2018. 2

[31] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic chebyshev spectral CNNs. In *Computer Vision and Pattern Recognition*, pages 14658–14667, 2020. 1, 2, 6, 7

[32] Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Nguyen, Ryutarou Ohbuchi, Yuki Ohkita, Yuya Ohishi, Fatih Porikli, Martin Reuter, Ivan Sipiran, Dirk Smeets, Paul Suetens, Hedi Tabia, and Dirk Vandermeulen. Shrec '11 track: Shape retrieval on non-rigid 3d watertight meshes. In *Eurographics Workshop on 3D Object Retrieval*, pages 79–88, 01 2011. 5

[33] Roee Litman and Alexander M Bronstein. Learning spectral descriptors for deformable shape correspondence. *Transactions on Pattern Analysis and Machine Intelligence*, 36(1):171–180, 2013. 8

[34] 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. *Transactions on Graphics*, 36(4):71, 2017. 2, 5, 6, 8

[35] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *International Conference on Computer Vision*, pages 37–45, 2015. 1, 2, 8

[36] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630, 2005. 8

[37] Thomas W Mitchel, Benedict Brown, David Koller, Tim Weyrich, Szymon Rusinkiewicz, and Michael Kazhdan. Efficient spatially adaptive convolution and correlation. *arXiv preprint arXiv:2006.13188*, 2020. 3

[38] Thomas W Mitchel, Szymon Rusinkiewicz, Gregory S Chirikjian, and Michael Kazhdan. Echo: Extended convolution histogram of orientations for local surface description. In *Computer Graphics Forum*. Wiley Online Library, 2020. 3, 4, 8

[39] 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 *Computer Vision and Pattern Recognition*, volume 1, page 3. IEEE, 2017. 1, 2

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 6

[41] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *Transactions on Graphics*, 37(6):236:1–236:14, 2018. 1, 2, 3, 4

[42] Adrien Poulenard, Marie-Julie Rakotosaona, Yann Ponty, and Maks Ovsjanikov. Effective rotation-invariant point CNN with spherical harmonics kernels. In *3D Vision*, pages 47–56, 2019. 1, 2, 3

[43] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Computer Vision and Pattern Recognition*, pages 652–660, 2017. 2, 5

[44] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[45] 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, 2014. 8

[46] Stefan C Schonsheck, Bin Dong, and Rongjie Lai. Parallel transport convolution: A new tool for convolutional neural networks on manifolds. *arXiv preprint arXiv:1805.07857*, 2018. 1

[47] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusion is all you need for learning on surfaces. *arXiv preprint arXiv:2012.00888*, 2020. 1, 2, 5, 6, 7

[48] Nicholas Sharp and Keenan Crane. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum*, 39(5), 2020. 5

[49] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. The vector heat method. *Transactions on Graphics*, 38(3), 2019. 5

[50] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision.*, pages 223–240. Springer, 2016. 2

[51] 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. 5

[52] Zhiyu Sun, Yusen He, Andrey Gritsenko, Amaury Lendasse, and Stephen Baek. Embedded spectral descriptors: learning the point-wise correspondence metric via siamese neural networks. *Journal of Computational Design and Engineering*, 7(1):18–29, 2020. 8

[53] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 6

[54] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Computer Vision and Pattern Recognition*, pages 6411–6420, 2019. 2

[55] Nathaniel Thomas, Tess Smidt, Steven M. Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *arXiv:1802.08219*, 2018. 1, 2

[56] 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. 3, 7, 8

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 6

[58] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Computer Vision and Pattern Recognition*, pages 2598–2606. IEEE, 2018. 2

[59] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic. Articulated mesh animation from multi-view silhouettes. *Transactions on Graphics*, 27(3), 2008. 6

[60] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *Transactions on Graphics*, 38(5):1–12, 2019. 2

[61] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. CNNs on surfaces using rotation-equivariant features. *Transactions on Graphics*, 39(4):92–1, 2020. 1, 2, 3, 4, 5, 6, 8

[62] Yuqi Yang, Shilin Liu, Hao Pan, Yang Liu, and Xin Tong. PFCNN: Convolutional neural networks on 3d surfaces using parallel frames. In *Computer Vision and Pattern Recognition*, June 2020. 1, 3

[63] Li Yi, Hao Su, Xingwen Guo, and Leonidas J. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Computer Vision and Pattern Recognition*, July 2017. 2

[64] Yongheng Zhao, Tolga Birdal, Jan Eric Lenssen, Emanuele Menegatti, Leonidas Guibas, and Federico Tombari. Quaternion equivariant capsule networks for 3d point clouds. In *European Conference on Computer Vision.*, pages 1–19. Springer, 2020. 2