Dual Geometric Graph Network (DG2N)
Iterative network for deformable shape alignment

Dvir Ginzburg
Tel Aviv university
dvirginzburg@mail.tau.ac.il

Dan Raviv
Tel Aviv University
darav@tauex.tau.ac.il

Abstract

We provide a novel approach for aligning geometric models using a dual graph structure where local features are mapping probabilities. Alignment of non-rigid structures is one of the most challenging computer vision tasks due to the high number of unknowns needed to model the correspondence. We have seen a leap forward using DNN models in template alignment and functional maps, but those methods fail for inter-class alignment where non-isometric deformations exist. Here we propose to rethink this task and use unrolling concepts on a dual graph structure - one for a forward map and one for a backward map, where the features are pulled back matching probabilities from the target into the source. We report state of the art results on stretchable domains' alignment in a rapid and stable solution for meshes and point clouds[1].

1. Introduction

The alignment of non-rigid shapes is a fundamental problem in computer vision. It plays an important role in multiple applications such as pose transfer [40], cross-shape texture mapping [63], 3D body scanning [3], and simultaneous localization and mapping (SLAM) [59]. The task of finding dense correspondence is especially challenging for non-rigid shapes, as the number of variables needed to define the mapping is vast, and local deformations might occur. To this end, a variety of solutions were offered to solve this problem, using axiomatic and learnable methods. From defining unique key-points or local descriptors and matching such descriptors between the shapes [49, 1, 43, 50], spectral-based methods that try to align the spectra of the shapes [29, 42, 21, 18], or template-based approaches that assume a known pre-defined structure closely resemble all shapes and find the correspondence from each shape to that template [20].

Many algorithms for non-rigid alignment relax the problem to matching probabilities which grants them the possibility to
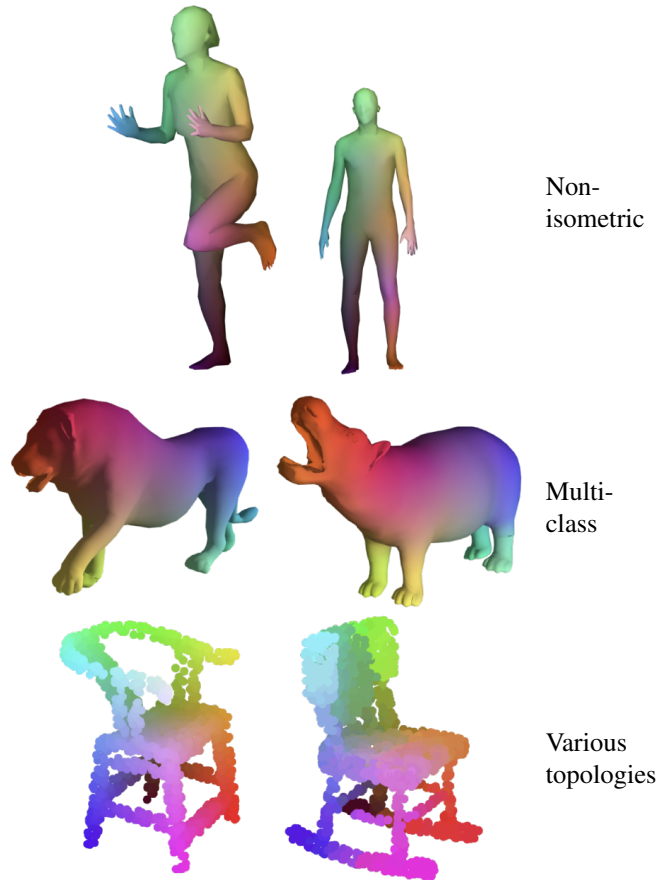

Figure 1. Dense shape correspondence maps generated by DG2N. Similar colors represents correspondence mapping from the source shape (left) to the target (right).

consider noise and variability in the pipeline. The transition from soft mapping to vertices alignment, or directly matching points, requires a post-processing step to remove outliers and smooth the results. Unfortunately, this is a slow process and is not performed in a network, as those algorithms are resource-demanding and require a large number of repetitions [54, 27].

---

[1]Our code will be publicly available upon publication

In this work, we focus on the refinement of a non-rigid alignment map. We unroll the refinement process into a multi-block graph neural network that performs the map denoising. To denoise the alignment in a learnable manner, we construct a dual graph structure, one for the forward map and one for the backward map, where we claim that the features are the actual probabilities for mapping pulled back from the target. In simple words, what best describes a point in the source, is not just its local features but how do all the points in the target resemble it. We call that structure a *Dual Geometric Graph Network (DG2N)*. As the method does not depend on the modality of the input, DG2N is able to refine both meshes and point-clouds under various deformations, as demonstrated in the experimentation section 4. We report state-of-the-art results on multiple benchmarks and succeed in providing a stable solution even under large non-isometric deformations.

Contributions   We present three key contributions:

- Build a new architecture for self-supervised non-rigid alignment based on a residual pipeline that converges into a clean, soft mapping matrix for each pair of models (zero-shot).

- Present a novel concept for graph features derived from the soft alignment map between the shapes.

- Report state of the art results in a wide range of benchmarks, including FAUST, TOSCA, SURREAL, SMAL, and SHAPENET.

2. Background

This work is focused on alignment refinement of an initial map between non-rigid models , motivated by denoising concepts of graph neural networks. Let us elaborate on each one of those elements.

Graph neural networks   While deep learning effectively captures hidden patterns in grid sampled data, we witness an increasing number of applications where the information is better represented in graphs or manifolds [26, 9, 57]. New challenges arise from a non-Euclidean structures due to the variable size of neighbors and unordered nodes.

Graph neural networks go back to 1997, working on acyclic graphs [48], but the notion of graph neural network was officially introduced by Gori *et al*. in 2005 [19]. Within the idea of graph neural networks, the most relevant to this work are convolutional graph neural networks, also known as ConvGNN. Under this umbrella, we can find two main streams; spectral and spatial. The first prominent research on spectral networks was presented by Bruna *et al*. [5]. On the other hand, a spatial convolutional structure was addressed more than a decade ago by Micheli [34], which has recently been resurfacing, showing its usefulness for multiple tasks in geometry and computer vision.

A variety of modern graph learning algorithms [11, 53, 26, 25, 35, 16] replaced the traditional Euclidean convolution with a general concept of pulling that can be implemented on a graph. Among popular modern graph neural network architectures for computer vision tasks, we can find PointNet [38], its successor PointNet++ [39] and DGCNN [57], which provides useful tools to convolve over a set of points.

In this paper, the unit blocks we use are based on top of the graph convolution network [25], the vertices are points in space, edges are based on the input modality, which is the triangulation for meshes or euclidean nearest-neighbors for point clouds, and the features are alignment probabilities in between the source and the target.

Non-rigid shape correspondence   Non-rigid shape matching is built out of aligning points with similar features, geometric and/or photometric, and a smoothness term, making sure a point can not be mapped farther from its neighbor. Under this umbrella, we had seen various axiomatic methods focused on distances, angles, and areas [43, 50] where a large leap forward was made when deep learning was applied on top of geometric data. We can split deep models into two categories - spatial and spectral. Under the spatial approach, we usually see a flow mechanism where the models' changes are minor [61, 30] or an all-to-all correlation approach [37] that can cope with large displacements. When the domain is well defined, then template matching showed great results, as seen in [20] and in [23]. On the spectral side, various methods based on functional maps [29, 18, 21] showed superb results on meshes and points and even excelled on partial alignment [29]. Those papers' goal was to construct deep local features such that the spectra of the shapes would align following a point-to-point soft correspondence matrix.

One of the challenges in non-rigid alignment is the lack of labeled data. That is mainly because there is no feasible way to own the exact dense correspondence of bendable and stretchable domains on real scanned sets. To overcome this obstacle and remain within the learnable regime, we consider a self-supervised approach. In the spatial domain, we have seen several useful cost functions that use templates while forcing smoothness on the structures [20]. More sophisticated assumptions on the domain, such as isometry, were able to learn a mapping by minimizing the Gromov Hausdorff metric as it only needed to compare distances between pairs [21], but failed to converge once stretching appeared. A recent mapping with a cyclic loss measuring the error only on the source showed superior results even under local stretching [18].

All those methods break once there isn't enough data to train. As reported by the authors, either the system can not converge, or we witness a high number of outliers. To overcome this limitation, we present a zero-shot alignment architecture between two-shapes, where we rethink the alignment process as denoising a soft correspondence matrix. By that, we

quickly converge into a clean outlier-free model and can cope with inter-class alignments even under large deformations.

Correspondence refinement While the methods mentioned in the previous section brought for the first time the capability to densely align between 3D objects with satisfying results, still, most output maps were noisy, partial, or sparse. For the extent of our knowledge, we are the first to offer a learnable refinement pipeline, nevertheless, many axiomatic methods have tried to iteratively refine and sharpen these maps. Most refinement methods solve the *optimal transport problem* between the shapes under various constraints derived from the input mapping. One example of such approach was even presented in the original functional maps paper, where the authors showed how ICP [2] applied to the spectral features improves the map dramatically. Others [32] solve the transport problem directly under constraints as geodesic distance preservation or variance minimization of the mapping. Methods as the Product Manifold Filter (PMF) [55] use linear or quadratic assignment solvers [27] to determine the solution to the transport problem. Lately, an important method named ZoomOut [33] showed how one can refine the initial map in the spectral domain by progressively increasing the dimension of the functional mapping, refining at each step. While the above axiomatic methods are milestones in the field, methods that try to solve the transport problem in the spatial space become computationally unfeasible and slow even under relatively sparse input sampling. On the other hand, spectral methods solve the alignment problem with a decent outcome for low-frequency maps but struggle with high frequencies. Furthermore, the functional maps setting used in methods like ZoomOut assume isometry between shapes, thus present degraded results in local scaled and deformable shape matching. Adding to the above, all spectral methods are based on the Laplace Beltrami operator which is known to be unstable and generate poor spectra under modalities different that meshes, like point clouds. Due to that, such methods tend to show inconsistent results, or even harm the initial map in some settings, as we present in the results section 4.

Graph denoising Denoising graph signals is a ubiquitous problem that plays an important role in many areas of machine learning [26, 58, 51], and was proven to improve results on a wide range of problems [8, 46]. The two main approaches for analyzing graph signals are graph regularization-based optimization and graph dictionary design [22, 47]; The optimization approach applies a regularization term that promotes certain characteristics on the model, such as smoothness or sparsity [10, 8]. The optimization function itself usually takes the form of

$$\underset{x}{\operatorname{argmin}} ||t - x||_2^2 + \lambda Q(x)$$

where t is the noisy graph signal, and Q is the regularization term.

When smoothness of the graph signal is assumed, one popular choice is the quadratic form of the graph Laplacian, the Dirichlet energy, discretized as $x^T \mathcal{L} x$ where x is the graph signal, which captures the second-order difference of a graph signal [36]. For sparsity of the graph signals, a graph total variation term that captures the first-order difference of the graph signals was proven effective [58, 9]. Recently, denoising graphs using deep architecture showed superior results by unrolling the L_1 regularization term into several layers, converging iteratively into the desired cost function [7].

3. Method

The proposed method is a self-supervised learnable pipeline. To align two non-rigid models, we use point embeddings generated by a black box model we refer as the initiator. Using these embeddings we define the soft-alignment map as the cosine similarity between the source and target embeddings. We learn how to iteratively update the soft correspondence matrix to improve the results and remove outliers 3.1. Our cost function is based on the understanding that each point's best features are the correspondence probabilities to the target points. Specifically, if P is a soft correspondence matrix, i.e., if $P \in \mathbb{R}^{N \times M}$, and $P_{ij} \in [0, 1]$ is the probability that point i matches point j, then in the primal graph, the features of point i are the i'th row of P, and the features of the dual graph are represented by the columns (or rows of P^T). We denote this primal-dual structure as the Dual Graph Geometric Network (DG2N).

While we use the cosine-based soft correspondence map as $\mathcal{P}$, DG2N can work with any soft-alignment matrix, such as the soft-alignment map proposed in [29], and extended in [21, 18].

In the scenario of point clouds, where there is no relevant dense-correspondence initiator (Section 4), we use a rigid-alignment algorithm such as DCP [56] as our feature extraction network. This is an extremely weak learner that produces outliers and inaccurate alignments, but it is sufficient to train the proposed DG2N architecture and converge to a very good mapping.

DG2N is composed of our new graph attention mechanism, activated on the two graphs (primal and dual) simultaneously. We refer to the new convolution blocks by differential-GAT, or DGAT. Inspired by [53], we consider a pulling strategy in-between points and their neighbors, where we concatenate the differences between the node features for a fixed number of neighbors.

To keep improving the outcome and not collapsing during the denoising process, we present four cost functions. We require the alignment to be injective, smooth, not too far from the previous iteration, and to keep the most valuable points in place.

In what follows, we elaborate on the main three components of the architecture's pipeline and the four cost functions.
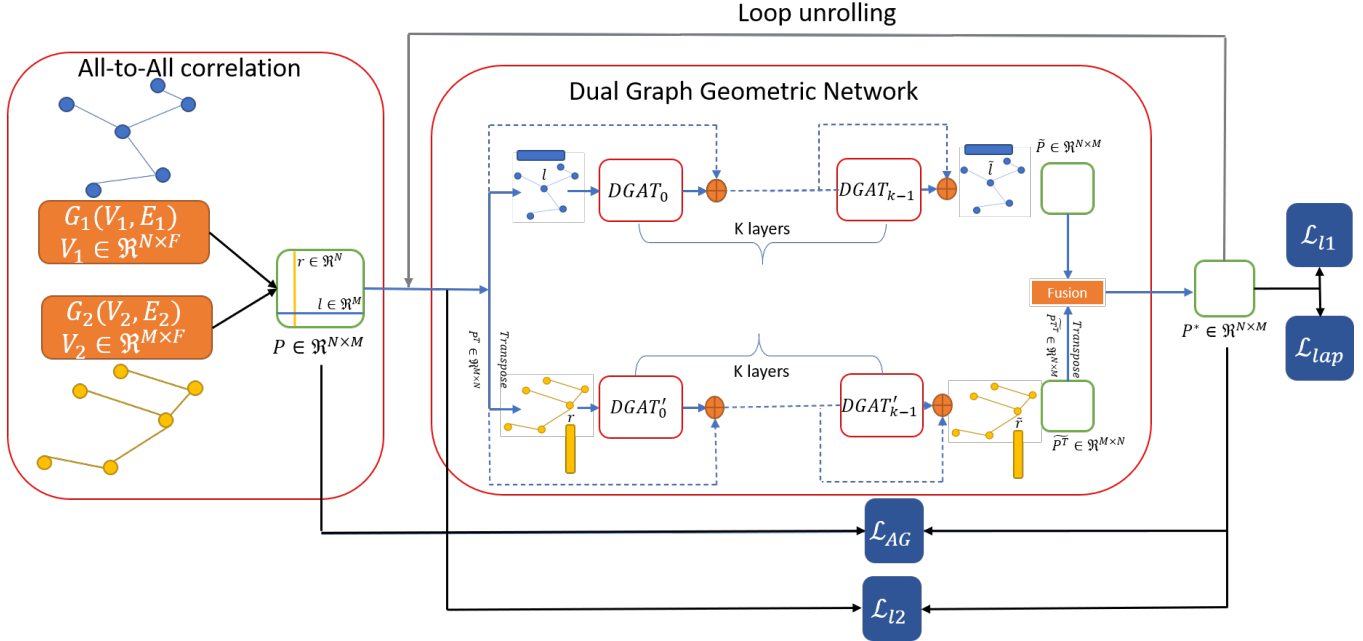
Figure 2. Our dual graph geometric network. Given two input graphs we first pass them through an initiator to have the initial soft correspondence matrix $\mathcal{P}$, representing correspondence probabilities between all vertex-pairs in the mapping. We then pass the graphs induced by $\mathcal{P}$ and $\mathcal{P}^T$ through stacked layers of $DGAT$ with residual connections, where the output is a refined soft correspondence matrix. The 4 loss objectives (Section 3.2) allows iterative refinement over $\mathcal{P}$, where the input $\mathcal{P}$ for the next iteration is the output of the previous one.

3.1. Architecture

All-to-All mapping To achieve a coherent and smooth correspondence map between two shapes, our dual graph unit (DG2N) uses the soft correspondence mapping $\mathcal{P}$ as an input. We can use any known method which has a soft correspondence matrix in the pipeline as an initiator, for example [18, 29, 21].

In detail, [29] showed that using the functional mapping $\mathcal{C}$, with the graphs laplacian eigendecomposition of the shapes Φ, Ψ the soft correspondence is constructed by

$$\mathcal{P} \propto |\Psi \mathcal{C} \Phi^T|. \tag{1}$$

In the scenarios where spectral methods are unstable or fail to create reasonable results (Section 4) we show here that an elementary all-to-all correlation matrix can be constructed from popular rigid alignment networks such as DCP [56]. We found that to be good enough as an initiator for the refinement. Specifically, we use the last hidden layer of DCP as a point descriptor h_{x_i}. The soft correspondence is constructed by the cosine similarity between the descriptors:

$$\mathcal{P}_{i,j} = \frac{h_{x_i} \cdot h_{x_j}}{||h_{x_i}||_2 \cdot ||h_{x_j}||_2}, \tag{2}$$

where h_{x_i} and h_{x_j} represent two feature vectors of points x_i and x_j.

In Section 4 we show such a simple solution finds a noisy correspondence between non-isometric pairs but provides sufficient initialization for our architecture.
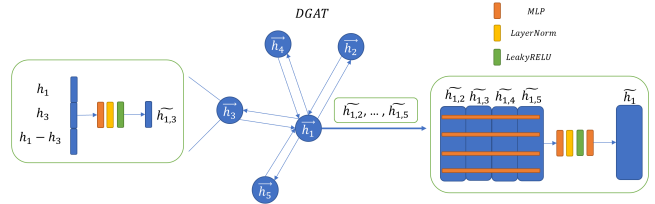

Figure 3. Single DGAT layer. Phase I: Create the *difference guiding feature vector* $[\vec{h_i}|\vec{h_j}|\vec{h_i} - \vec{h_j}] \in \mathbb{R}^{3M}$ and pass it through a first learnable architecture. Phase II: stack phase I features $\tilde{H}_i \in \mathbb{R}^{M \times K}$ and regress each feature individually through the second module forming the output feature per point $\tilde{h}_i \in \mathbb{R}^M$.

Differentiable GAT DG2N is composed of two parallel GNN modules based on the Differential Graph Attention (DGAT) layer. Inspired by GAT [53], DGAT perform weighted local pooling, only here we stack the output feature vector from the per-pair stage and apply a per-feature network to learn the best refinement step.

The most generalized structure associated with GNNs is

$$f_i' = \gamma_\Theta(f_i, \square_{j \in \mathcal{N}(i)} \phi_\Theta(\psi(f_i, f_j, e_{j,i})))$$

where f and f' represent the input and output data channels respectively, $\square$ is some differentiable aggregation function, $\gamma_\Theta, \phi_\Theta$ denote non-linear transmission functions, and ψ defines the feature-fusion method applied before the features propagate through ϕ_Θ.

$DGAT$ set $\square$ to be stacking of the per-neighbor output fea-

ture vector, ψ is the guiding vector function

$$\psi(f_i, f_j, e_{j,i}) = [f_i | f_j | f_i - f_j], \qquad (3)$$

and the graph's edges determine the neighborhood. γ_Θ, and ϕ_Θ are variants of a multi-layer perception (MLP) with normalization and non-linear activation function layers. In practice, DGAT takes the form

$$f_i' = DNN_2(\underset{j \in \mathcal{N}(i)}{\|}(DNN_1([f_i | f_j | f_i - f_j])) \qquad (4)$$

where $\underset{j \in \mathcal{N}(i)}{\|}$ is the concatenation of DNN_1 outputs. An illustration can be found in Figure 3.

One crucial emphasis here is the role of DNN_2 and the distinction to other suggested aggregation functions. One optional aggregation would be to concatenate the output difference features resulting in a feature vector of dimension $\tilde{H}_i \in \mathbb{R}^{KM}$, while we stack the features resulting in $\tilde{H}_i \in \mathbb{R}^{M \times K}$. Our construction not only produces a learnable module with a factor of M^2 fewer parameters as $DNN2$ is applied on the k dimensional vectors but, more significantly, acts as a learnable weighting function that incorporates the various per-neighbor refinements into a refinement step per-node.

Dual Geometric Graph Network In the heart of the proposed architecture, is our understating that the soft correspondence matrix $\mathcal{P}$ induces a graph. The nodes of the *primal* graph are the source points, and the features are the correspondence measure for all target points, i.e., the rows of $\mathcal{P}$ are the features. The *dual* graph has the same structure only based on $\mathcal{P}^T$. Here the nodes are the vertices of the target shape, and the columns of $\mathcal{P}$ are the features. We provide a visualization of the architecture in Figure 2. Each primal-dual pair $(\mathcal{P}, \mathcal{P}^T)$ is passed through k layers of $DGAT$ in a res-net structure; i.e. the output of each $DGAT$ layer is $DGAT(\mathcal{P}) + \mathcal{P}$ for an input soft correspondence matrix $\mathcal{P}$, and similar for $\mathcal{P}^T$, the dual graph pipeline. In each iteration, we fuse $\mathcal{P}$ and $\mathcal{P}^T$ into one aligned soft correspondence matrix[2].

The output of each iteration of DG2N refinement is also a soft correspondence matrix. As the correspondence statistics vary between one iteration to the next, we use different weights per iteration. As we continue to iterate, the soft correspondence matrix improves and converges to a clean, outlier-free soft mapping. At inference, the output map is the maximum-likelihood solution derived from the soft correspondence matrix, which is:

$$\pi(\mathcal{X})_i = \underset{j}{\arg\max} \, \mathcal{P}_{ij}. \qquad (5)$$

Where $\mathcal{X}, \mathcal{Y}$ are the source and target shapes respectively.

---

[2]There are several reasonable options for the fusion, as element-wise max or mean. In practice, no consistent improvement was noted by one option over the other.

3.2. Losses

We combine four different losses in this pipeline. $\mathcal{L}_L$ Laplacian loss, $\mathcal{L}_{l1}$ Sparsity loss, $\mathcal{L}_{AG}$ Anchors guidance loss and $\mathcal{L}_{l2}$ Denoising regularization.

These constraints form together the loss objective of a single refinement step of DG2N, which is:

$$\mathcal{L} = \mathcal{L}_L + \mathcal{L}_{l1} + \mathcal{L}_{AG} + \mathcal{L}_{l2} \qquad (6)$$

All four losses are evaluated separately and summed together both for the primal and dual graphs and executed for every iteration output.

Let us elaborate on each term in the loss.

Laplacian loss Laplacian regularization term pushes toward graph smoothness. It takes the form of:

$$\mathcal{L}_L = \lambda_L \mathcal{P}^T L_\mathcal{P} \mathcal{P} = \lambda_L \sum_{(i,j) \in E_\mathcal{X}} w_{i,j} ||\mathcal{P}_{i,\star} - \mathcal{P}_{j,\star}||_2^2$$

where $L_\mathcal{P} = D - A$ is the graph Laplacian of the source shape $\mathcal{X}$, D is the degree of each node, A is its adjacency matrix, and $\mathcal{P}_{i,\star}$ is the i'th row of $\mathcal{P}$. Two important items to note here, the first is that this term can be used on any structure inducing a graph. Second, while all other methods use Laplacians on the shape coordinates in space, we claim that smoothness should apply directly to the soft correspondence matrix. Since the features are the mapping probabilities, we claim the smoothness on $\mathcal{P}$ is a better goal.

Sparsity regularization We add the $\mathcal{L}_1$ regularization on the rows of $\mathcal{P}$. Specifically,

$$\mathcal{L}_1 = \lambda_{l1} \sum_{i=1}^{N} |\mathcal{P}_{i,\star}|_1. \qquad (7)$$

As the rows of $\mathcal{P}$ represent the alignment probabilities, we wish to promote sparsity. Each source point corresponds to a single target point, meaning one element should hold most of the energy, and the rest should decline rapidly. Note that we are not normalizing the rows or columns in each iteration, thus $\mathcal{P}$ is in fact a pseudo probability matrix, where each row does not guaranteed to sum to one.

Anchors guidance loss One of the caveats with the Laplacian regularization is its tendency for over-smoothing and thus hurt the overall performance [60]. In our case, this phenomenon takes the shape of pushing all correspondence probabilities of $\mathcal{P}$ towards the average. While this decreases the Laplacian loss, it results in significant degradation of the results, as shown in the ablation study (Table 3).

To solve the mentioned problem, we present a self-supervised anchor guidance mechanism. Motivated by node classification tasks [17, 44], a few anchor points are sampled from the initial soft correspondence map, and we seek

5

to use their initial mapping as guidance through the refinement. Bear in mind that those points are not fixed and are part of the learnable pipeline, only we provide extra attention to points we believe in their mapping.

Analyzing soft correspondence mappings generated from different pipelines (FMnet [29], SURFMNet [42], DCP [56]), we observed two attributes that reoccur by all algorithms:

1. Source nodes where the highest correspondence probability of $\mathcal{P}$ is two orders of magnitude larger than the average probability ($\frac{1}{M}$) usually point to the true correspondence.

2. High probability correspondences reside in clusters, that is, if a source node corresponds to some target node with high probability, it is usually the case its neighbors will also have high probability correspondence to some neighbor of this corresponding point.

We utilize the above observations to attend the over-smoothing caused by the Laplacian. For each $\mathcal{X}, \mathcal{P}, \mathcal{Y}$ we first sample $k \leq |V_{\mathcal{X}}|$ disconnected nodes using FPS [15] noted as $V_{K_{\mathcal{X}}}$, and assign their soft label by defining

$$\hat{y}_i = \underset{j}{\arg\max}\, \mathcal{P}_{i,*} \quad \forall v_i \in V_{K_{\mathcal{X}}}$$
$$C(\hat{y}_i) = \mathcal{P}_{i,\hat{y}_i} \tag{8}$$

where $C(\cdot)$ is the confidence x_i corresponds to $\hat{y}_i$.

Using the above formulation, we define a soft-classification problem. We constrain the network to label the anchor points similarly to how they were classified before the refinement layer. The penalty for each wrong classification is directly proportional to the confidence $C(\cdot)$.

We use the anchors' notations and define the anchor loss as the cross-entropy between the presumed label, and the output features of DG2N Layer. Specifically,

$$\mathcal{L}_{AG} = \sum_{x_i \in V_{K_{\mathcal{X}}}} C(\hat{y}_i)^2 \tag{9}$$
$$\left(-x_i[\hat{y}_i] + \log \Big(\sum_{j=0}^{|V_{K_{\mathcal{X}}}|-1} \exp\left(x_i[j]\right) \Big) \right),$$

where $x_i[m]$ is the m'th element in the row of $\mathcal{P}$ corresponding to vertex x_i.

Denoising regularization For each iteration, we assume the output is similar to the input. By that we force the network to penalize for large gaps, and de-facto promote minor updates, usually referred to as noise or outliers in this paper. Denoting the previous layer matrix as $\mathcal{P}$ and the output of a single DG2N iteration as $\mathcal{P}^*$, the denoising regularization takes the form of:

$$\mathcal{L}_{l2} = \lambda_{l2} ||\mathcal{P}^* - \mathcal{P}||_2^2. \tag{10}$$

4. Experiments

The following section presents multiple scenarios in which our self-supervised architecture surpasses current state-of-the-art algorithms for non-rigid alignment. In addition, we will present our zero-shot pipeline that achieves near-perfect results for non-isometric deformable shape matching. To adjust DCP [56] for the dense correspondence task we follow FMnet [29] loss and optimize for $\mathcal{L} = ||P - I||_2^2$, as during DCP training the map is given by $\pi(\mathcal{X}_i) = \mathcal{Y}_i$.

We evaluate DG2N on a wide range of popular datasets for dense shape correspondence. To assess the network's robustness, we test it on multiple datasets with different statistical and topological attributes as humans datasets (FAUST[3] and SURREAL [52]), animals (SMAL [64] and TOSCA [4]) or chairs and plains (SHAPENET[6]). We use a remeshed and down-sampled version of FAUST, SURREAL, and SMAL, as suggested by [41]. In the generated datasets, each shape has approximately 1000 vertices. These re-meshed datasets offer significantly more variability in terms of shape structures and connectivity than the original datasets [14].

Mesh Error Evaluation The measure of error for the correspondence mapping between two shapes will be according to the Princeton benchmark [24], that is, given a mapping $\pi_\rightarrow(\mathcal{X}, \mathcal{Y})$ and the ground truth $\pi_\rightarrow^*(\mathcal{X}, \mathcal{Y})$, the error of the correspondence matrix is the sum of geodesic distances between the mappings for each point in the source figure, divided by the area of the target figure.

$$\epsilon(\pi_\rightarrow) = \sum_{x \in \mathcal{X}} \frac{\mathcal{D}_{\mathcal{Y}}(\pi_\rightarrow(x), \pi_\rightarrow^*(x))}{\sqrt{area(\mathcal{Y})}}, \tag{11}$$

where the approximation of $area(\cdot)$ for a triangular mesh is the sum of its triangles area.

Humans datasets - FAUST and SURREAL We follow the suggested setting [14] for these human datasets and split both datasets into training sets (80 shapes) and test sets (20 shapes). The specific shape splits are identical for all tested methods for a fair comparison. We test two scenarios, one in which we train and evaluate on the same dataset and one in which we test on the other dataset (e.g., training on FAUST evaluating on SURREAL). This experiment aims attesting the generalization power of all methods to small re-meshed datasets, as well as their ability to adapt to a different dataset at test time.

Table 1 stresses some of the key advantages of DG2N compared to other self-supervision methods and refinement techniques, as robustness and generalization. While almost all learnable methods perform reasonably well on the same-dataset benchmark, we see significant performance gaps compared to other methods when conducting the cross-dataset test; this is due to the fact we are self-supervised and shape-pair specific, thus are almost invariant to noise

	FAUST [3]	SURREAL [52]	F on S	S on F	SMAL [64]	SMAL on TOSCA [4]
FMNet [29]	12.1	18.7	35.3	33.4	*	*
3D-CODED [20]	8.5	15.5	28.5	26.0	8.8	35.7
Deep GeoFM [14]	3.8	4.2	7.8	14.2	*	*
DCP(Unsup) [56]	19.3	21.2	26.4	28.9	16.8	27.3
SURFMNet(Unsup) [42]	7.1	11.3	31.5	42.3	*	*
Unsup FMNet(Unsup) [21]	13.1	14.6	33.2	38.5	*	*
PMF [55] on DCP	18.1	19.8	21.6	25.8	14.0	23.9
PMF [55] on GeoFM	3.5	4.1	6.6	8.5	*	*
ZoomOut [33] on DCP	22.9	18.5	28.0	29.9	14.6	26.8
ZoomOut [33] on GeoFM	**2.9**	**3.8**	6.3	8.4	*	*
Ours(Unsup) on DCP [56]	15.3	12.9	21.1	25.4	**7.9**	**19.5**
Ours(Unsup) on FMNet [21]	9.6	11.8	13.5	14.9	*	*
Ours(Unsup) on SURFMNet [42]	5.9	8.1	9.3	10.5	*	*
Ours(Unsup) on GeoFM [14]	3.4	4.1	**6.2**	**8.1**	*	*

Table 1. Mean geodesic error (MGE) comparison by different methods on FAUST(F), SURREAL(S), SMAL and TOSCA datasets. No post processing filters are used for any of the methods except the specified refinement procedures. We remark that due to the numerical instabilities of the Laplacian decomposition we were not able to run the spectral methods with the code published by the authors (results marked with *) on our re-sampled SMAL dataset. ZoomOut presents slightly better results than DG2N for refining GeoFM, where GeoFM already achieves remarkable results on FAUST, without any post-processing filters. However, when refining imperfect initiators, such as DCP, ZoomOut compromises the initiators' results, *degrading* the MGE by 3.6 points. In contrast, our DG2N is robust to imperfect initiators, and *improves* the noisy correspondence of DCP by 4.0 points.

added to the system by changing the statistical attributes of the data. Comparing to PMF [55], having an average refinement time of two minutes per shape-pair, we see that DG2N outperforms its results in both settings while refining in two orders of magnitude faster. PMF is extremely sensitive to the optimization hyper-parameters, and in our experiments we observed variability of up to 5X in the MGE score under different parameters. We report here the results under the optimal parameters. Compared to ZoomOut refining GeoFMNet, which receives remarkable results without any post processing filters, ZoomOut presents 0.5 cm MGE less in the same-dataset setting, and worst results compared to DG2N in all other experiments. When ZoomOut is examined on noisy initial maps, as in the case of using unsupervised initiators, ZoomOut presents results that are often worse than not using filters at all. We attribute that phenomena to the use of the hard mapping $\pi(\mathcal{X})$ by ZoomOut. $\pi(\mathcal{X})$ of noisy initiators often include cases where neighbor points map to geodesically distant target vertices, using such outliers as the initial conditions for the refinement process may cause divergence, as happened in our evaluations. Unlike ZoomOut, DG2N takes advantage of the soft alignment matrix, which indicates the pipeline for possible outliers, and offer other mappings that are coherent to the point neighborhood.

Animals datasets - SMAL and TOSCA To better understand the different models' generalization capabilities and ensure the models are not hand-crafted for human-like structures, we also assess the network's performance on animal datasets. SMAL [64] dataset provides a generative model for synthetic animals creation in different categories as cats,

horses, etc.; SMAL is extracted from a continuous parametric space with a fixed number of vertices and same triangulation for all shapes, with the possibility of generating "infinitely many" training samples. Unlike SMAL, TOSCA [4] contains a fixed selection of shapes, including 9 cats, 11 dogs, 3 wolves, etc. Which is both dramatically smaller and has no topological guarantees, meaning no two shapes have the same triangulation. The animals datasets experiment was conducted as follows: For each SMAL category, we create 80 shapes for training and 20 for the test, resulting in 500 samples. We must emphasize that previous methods [20] that worked with SMAL used two orders of magnitude more training samples in their experiments. Table 1 expresses the advantages of DG2N over previous works that are considered state-of-the-art in this regime. The tested spectral based methods (FMnet variant [29, 21]) failed to converge on the remeshed datasets, probably due to the unstable and noise process of the decomposition of the Laplacians. Compared to ZoomOut and PMF we see similar trains to the FAUST and SCAPE experiments, where both achieve an average of 4 cm MGE worse results than DG2N.

Deformable irregular correspondences - point clouds registration Point cloud registration is undoubtedly one of the hardest registration tasks for 3D shapes, while it is the most common scenario in real-world cases. We evaluate the different methods of chosen classes from SHAPENET [6], namely chairs, cars, and plains; Each category contains multiple subjects, where no pair is isometric, nor has the same number of points. Unlike meshes, point clouds suffer from noise and topology ambiguity due to the sampling process

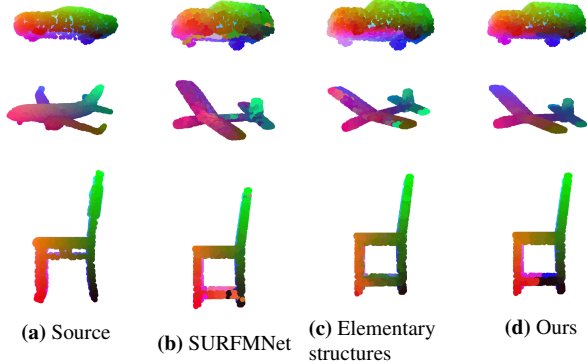|			
(a) Source	**(b)** SURFMNet	**(c)** Elementary structures	**(d)** Ours

Figure 4. Dense correspondence on point clouds - spectral solutions fail to create smooth or coherent results due to the noisy nature of the Laplacian for point clouds, Elementary structures [12] and other reconstruction based methods do not enforce smoothness resulting in noisy maps.

involved in generating them and the surface-approximation heuristics needed to define each point's neighborhood. Spectral based methods undergo significantly degradation in the results, since spectral decomposition of point clouds is inaccurate and unstable [28, 31]. GeoFMNet is a supervised pipeline, thus irrelevant for this evaluation as ShapeNet doesn't contain any correspondence labels. The authors of SURFMNet did not evaluate on point-clouds nor offered the tools for such evaluation. For fairness, we used [45] which is a new tool for Laplacian decomposition on point clouds for the evaluation of SURFMNet. No dataset currently exists with ground-truth correspondences between deformable point clouds, so we turn to evaluate the performance of the different methods visually, in terms of smoothness, coherence[3], and robustness to deformations.

Zero-shot correspondence "Zero-shot" self-supervised methods are essential and brought great achievements and new capabilities in other domains as super-resolution and image generation [13, 62]. Having a zero-shot registration method for 3D shapes is considered exceptionally difficult, with only a few [18, 21] that tried to tackle the problem. Unfortunately, as seen in previous experiments (Section 4), spectral methods are sensitive and limited in terms of the input domain. To present our self-supervision capabilities, we chose randomly 10 inter-class shape pairs for the FAUST-remeshed dataset. For each pair, we trained DCP [56] only on the inference pair until convergence exposing new linear augmentations each training step and ran the inference. On the provided output we ran DG2N refinement scheme. Naturally, only unsupervised methods are relevant for comparison. We present a comparison to other zero-shot methods in Table 2.

Method	Mean geodesic error
SURFMNet(Unsup) [42]	36.2
Unsup FMNet(Unsup)[21]	16.5
Cyclic-FMnet(Unsup) [18]	14.1
DCP(Unsup) [56]	19.5
Ours(Unsup) on DCP	**11.0**

Table 2. Mean geodesic error in a zero-shot setting on FAUST-remeshed. We present best results among all unsupervised methods that are relevant to this experiment setting.

4.1. Ablation

DG2N training is constructed of 4 different loss functions, each plays an important and substantial role in the refinement process. We provide table 3 as numerical evidence to the significance of the different cost functions, as well as our DGAT module importance. While some objectives improve the refinement effect, some, as $\mathcal{L}_{AG}$ or $\mathcal{L}_{l2}$ are indispensable, with substantial degradation to the results without their regularization effect to the denoising process. Inspecting the effect of replacing DGAT with GAT [53] or GCN [25] we witness substantial performance decrease, where the baseline alternatives bring degraded results compared to the initiator mapping. The ablation was done on FAUST resampled, with the initial correspondence generated by SURFMNet.

	Ablation	MGE
Baseline	SURFMNet	7.1
	$\mathcal{L}_L + \mathcal{L}_{l1}$	47.5
	$\mathcal{L}_L + \mathcal{L}_{l1} + \mathcal{L}_{AG}$	38.1
Loss	$\mathcal{L}_{l2} + \mathcal{L}_{AG}$	9.9
	$\mathcal{L}_L + \mathcal{L}_{l1} + \mathcal{L}_{l2}$	6.7
	$\mathcal{L}_L + \mathcal{L}_{l2} + \mathcal{L}_{AG}$	6.2
	DGCNN [57]	25.1
GNN	GCN [25]	21.9
	GAT [53]	14.3
Full	DG2N	**5.9**

Table 3. Ablation study

5. Summary

We presented a novel line of thought for aligning non-rigid domains using a learnable iterative pipeline. Motivated by graph denoising and presenting a dual graph structure built on top of soft correspondences, we rapidly converge into an accurate and free of outliers mapping even under severe non-isometric deformations. We report state-of-the-art results on multiple benchmarks and different scenarios, where other methods suffer poor outcomes or fail altogether.

---

[3]A good alignment will map a guitar neck of one shape to the other.

References

[1] 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] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 3

[3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014. 1, 6, 7

[4] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 6, 7

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

[6] 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 preprint arXiv:1512.03012*, 2015. 6, 7

[7] Siheng Chen, Yonina C Eldar, and Lingxiao Zhao. Graph unrolling networks: Interpretable neural networks for graph signal denoising. *arXiv preprint arXiv:2006.01301*, 2020. 3

[8] Siheng Chen, Aliaksei Sandryhaila, José MF Moura, and Jelena Kovacevic. Signal denoising on graphs via graph filtering. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 872–876. IEEE, 2014. 3

[9] Siheng Chen, Aliaksei Sandryhaila, José MF Moura, and Jelena Kovačević. Signal recovery on graphs: Variation minimization. *IEEE Transactions on Signal Processing*, 63(17):4609–4624, 2015. 2, 3

[10] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. American Mathematical Soc., 1997. 3

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering, 2017. 2

[12] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems*, pages 7435–7445, 2019. 8

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018. 8

[14] 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. 6, 7

[15] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997. 6

[16] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 2

[17] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998. 5

[18] Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. *arXiv*, pages arXiv–1912, 2019. 1, 2, 3, 4, 8

[19] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734. IEEE, 2005. 2

[20] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *ECCV*, 2018. 1, 2, 7

[21] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised Learning of Dense Shape Correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019. 1, 2, 3, 4, 7, 8

[22] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. 3

[23] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[24] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. 6

[25] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. 2, 8

[26] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018. 2, 3

[27] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 1, 3

[28] Rongjie Lai, Jiang Liang, and Hong-Kai Zhao. A local mesh method for solving pdes on point clouds. *Inverse Problems & Imaging*, 7(3):737, 2013. 8

[29] Or Litany, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Deep Functional Maps: Structured prediction for dense shape correspondence. *CoRR*, abs/1704.08686, 2017. 1, 2, 3, 4, 6, 7

[30] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019. 2

[31] Chuanjiang Luo. *Laplace-based Spectral Method for Point Cloud Processing*. PhD thesis, The Ohio State University, 2014. 8

[32] Manish Mandad, David Cohen-Steiner, Leif Kobbelt, Pierre Alliez, and Mathieu Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 3

[33] Simone Melzi, Jing Ren, Emanuele Rodolà, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *CoRR*, abs/1904.07865, 2019. 3, 7

[34] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009. 2

[35] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns, 2016. 2

[36] Todd K Moon and Wynn C Stirling. *Mathematical methods and algorithms for signal processing*, volume 1. Prentice hall Upper Saddle River, NJ, 2000. 3

[37] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Flot: Scene flow on point clouds guided by optimal transport, 2020. 2

[38] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[39] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2

[40] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images, 2018. 1

[41] 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. 6

[42] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE international conference on computer vision*, pages 1617–1627, 2019. 1, 6, 7, 8

[43] Raif M Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233, 2007. 1, 2

[44] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008. 5

[45] Nicholas Sharp and Keenan Crane. A Laplacian for Non-manifold Triangle Meshes. *Computer Graphics Forum (SGP)*, 39(5), 2020. 8

[46] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013. 3

[47] David I Shuman, Christoph Wiesmeyr, Nicki Holighaus, and Pierre Vandergheynst. Spectrum-adapted tight graph wavelet and vertex-frequency frames, 2013. 3

[48] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997. 2

[49] 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

[50] 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. 1, 2

[51] Dinh-Hoan Trinh, Marie Luong, Francoise Dibos, Jean-Marie Rocchisani, Canh-Duong Pham, and Truong Q Nguyen. Novel example-based method for super-resolution and denoising of medical images. *IEEE Transactions on Image processing*, 23(4):1882–1895, 2014. 3

[52] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 6, 7

[53] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. 2, 3, 4, 8

[54] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3327–3336, 2017. 1

[55] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alexander M. Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. *CoRR*, abs/1701.00669, 2017. 3, 7

[56] 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. 3, 4, 6, 7, 8

[57] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019. 2, 8

[58] Yu-Xiang Wang, James Sharpnack, Alexander J Smola, and Ryan J Tibshirani. Trend filtering on graphs. *The Journal of Machine Learning Research*, 17(1):3651–3691, 2016. 3

[59] Diedrich Wolter and Longin J Latecki. Shape matching for robot mapping. In *Pacific Rim international conference on artificial intelligence*, pages 693–702. Springer, 2004. 1

[60] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr. au2, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks, 2019. 5

[61] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds, 2020. 2

[62] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016. 8

[63] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7982–7991, 2019. 1

[64] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 6, 7