DMD-Net: Deep Mesh Denoising Network

Aalok Gangopadhyay*
CVIG Lab, IIT Gandhinagar
aalok@iitgn.ac.in

Shashikant Verma*
CVIG Lab, IIT Gandhinagar
shashikant.verma@iitgn.ac.in

Shanmuganathan Raman
CVIG Lab, IIT Gandhinagar
shanmuga@iitgn.ac.in

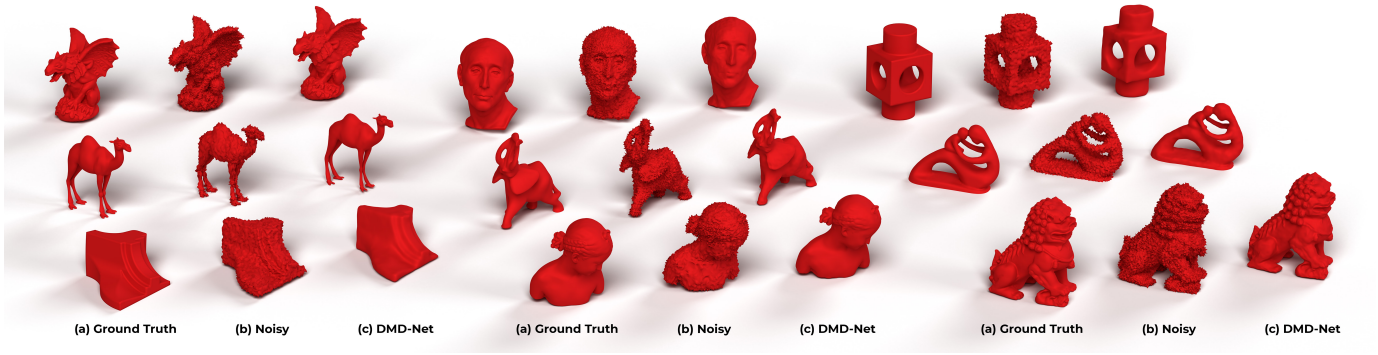


Fig. 1: Mesh denoising results obtained using DMD-Net on several meshes. The objects on the left (a) denote the original ground truth meshes. The objects in the middle (b) denote the noisy meshes obtained by adding noise to the ground truth meshes. The objects on the right (c) denote the denoised mesh results of DMD-Net acting on the noisy objects.

***Abstract*—We present Deep Mesh Denoising Network (DMD-Net), an end-to-end deep learning framework, for solving the mesh denoising problem. DMD-Net consists of a Graph Convolutional Neural Network in which aggregation is performed in both the primal as well as the dual graph. This is realized in the form of an asymmetric two-stream network, which contains a primal-dual fusion block that enables communication between the primal-stream and the dual-stream. We develop a Feature Guided Transformer (FGT) paradigm, which consists of a feature extractor, a transformer, and a denoiser. The feature extractor estimates the local features, that guide the transformer to compute a transformation, which is applied to the noisy input mesh to obtain a useful intermediate representation. This is further processed by the denoiser to obtain the denoised mesh. Our network is trained on a large scale dataset of 3D objects. We perform exhaustive ablation studies to demonstrate that each component in our network is essential for obtaining the best performance. We show that our method obtains competitive or better results when compared with the state-of-the-art mesh denoising algorithms. We demonstrate that our method is robust to various kinds of noise. We observe that even in the presence of extremely high noise, our method achieves excellent performance.**

I. Introduction

There has been a tremendous rise in the use of 3D acquisition technology for obtaining high fidelity digital models of real world 3D objects. 3D models obtained using CAD softwares are designed by humans and appear perfect in shape. On the contrary, 3D models obtained using 3D scanners contain noise owing to the measurement error of the scanning devices. In the case of 3D objects obtained using photogrammetry, considerable noise is introduced because of the reconstruction error. The task of eliminating these various types of noise to restore the object to its original shape is known as mesh denoising and is a fundamental problem in the field of 3D shape analysis.

Challenges. Given a noisy mesh, there are many possible candidates for the original noise-free mesh. This makes mesh denoising a highly ill-posed inverse problem. There is a natural trade-off involved in mesh denoising between that of eliminating the noise and preserving the high frequency details. Striking the perfect balance in this trade-off is a crucial aspect for any good denoising algorithm. Existing works in the literature have addressed these challenges and have devised innovative strategies for solving mesh denoising. While this problem has been approached by a variety of classical approaches, there are not many learning based approaches proposed that have addressed this problem.

Contributions. In this work, we design a novel GNN architecture to solve the mesh denoising problem. We use a large scale dataset to train a mesh denoising architecture in the deep learning framework. Unlike many existing learning-based methods, our method is end-to-end trainable and takes entire mesh as input instead of the patch based approach. Our contributions are three-fold:

1) We propose a novel Graph Convolutional Neural Network architecture, which employs primal-dual graph aggregation and a Feature Guided Transformer (FGT) paradigm. FGT serves the purpose of guiding the transformation from a noisy input mesh to its denoised representation.
2) Noise of varying intensity and types can be robustly denoised by our method. Moreover, the proposed method successfully preserves high frequency features and obtains the best result in terms of minimizing deviation in facet normals.
3) The architecture and loss functions proposed by us also work on meshes with non-manifold topology.

*These authors contributed equally to this work.

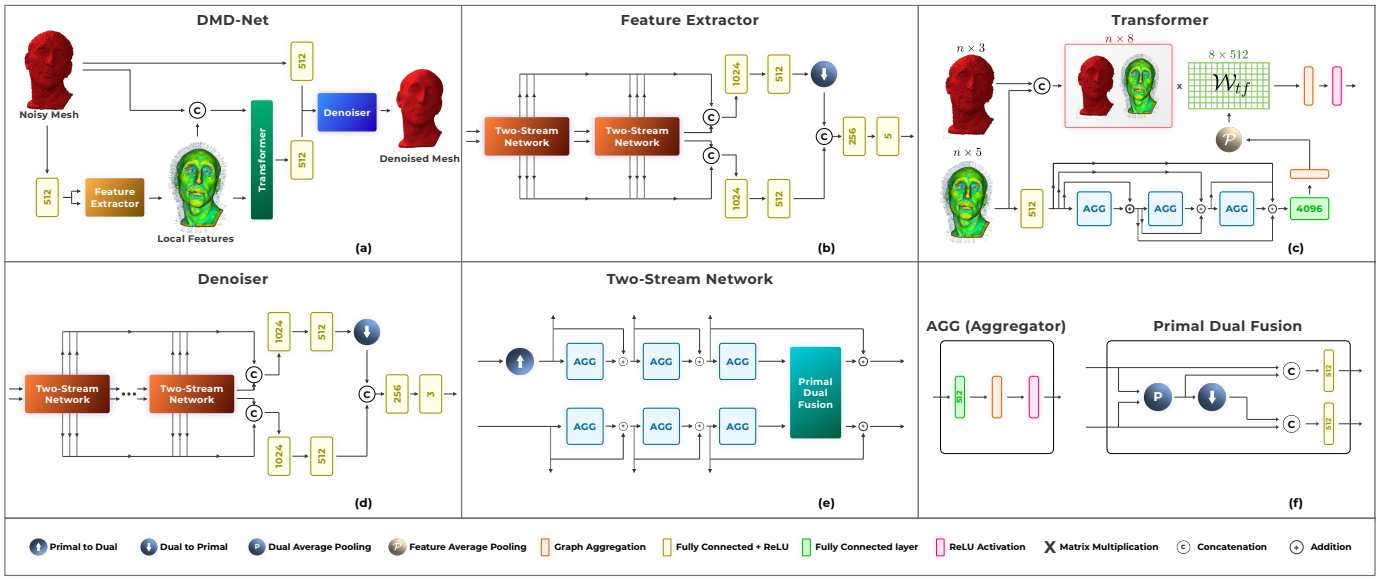


Fig. 2: The proposed DMD-Net architecture for mesh denoising: (a) Overall architecture depicting the feature guided transformer (FGT) paradigm. (b) The Feature Extractor, (c) The Transformer, (d) The Denoiser, (e) The Two-Stream Network, and (f) The Aggregator (AGG) and the Primal Dual Fusion (PDF) block.

II. Related Works

Isotropic and Anisotropic Methods. Initial works on mesh denoising such as [1], [2], [3], [4] applied mesh filters to remove noise. These methods were isotropic in nature as they did not depend on the geometry of the surface. As a result, these methods along with removing noise also blurred the high frequency details present in the mesh. In order to overcome these problems, later works have adopted anisotropic frameworks. Earlier works in this direction such as [5], [6], [7], [8], [9], [10], [11], [12] were based on anisotropic geometric diffusion, a method inspired from [13].

Bilateral Filters. Another direction of work is based on the use of bilateral filters in meshes either in the vertex domain [14], [15] or in the facet normal domain [16], [17], [18]. In [14], the vertices of the mesh are filtered in the normal direction by making use of the local neighbourhoods. In [15], a robust estimator is used to update the position of each vertex by aggregating the predictions of its spatial neighbourhood.

Multi-Step Schemes and Variants. The later anisotropic based works such as [19], [20], [21], [22], [23], [24], [25], [17], [16], [26] adopted a multi-step scheme, in which, for each face of the mesh, the normals are updated by averaging over the neighbouring normals and then the vertices are updated to be consistent with the new normals. Further, methods such as [27], [28], [29], [30], [31], [32], [33], [34], [35] have augmented the multi-step scheme with additional steps like adding feature detection and sub-neighbourhood searching. In [34], a two-stage scheme is used where the first step of normal filtering is guided by a normal field.

Sparse Optimization. These methods are based on the idea that the occurrence of sharp features in a mesh are sparse. Methods such as [36] and [37] perform L_1 optimization, [38] uses L_0 norm for surface curvature minimization, and [39] performs variational denoising using total variation regularizer along with piecewise constant function spaces.

Surface Reconstruction. The objective of methods such as [40], [41], [42], [43] is to perform surface reconstruction which is a task closely related to mesh denoising. These methods employ robust statistics to predict the surface normals.

Non-Local Similarity and Low Rank Matrix Recovery. Methods based on non-local similarity [44], [45], [46], [47] and low rank matrix recovery [48], [49], [50], [51] use the idea that there is redundancy in the patches of a mesh with similar looking patches being present in different regions. These methods further construct a patch matrix and recover its low rank.

Learning based. These include methods such as [52], [53], [54], [55], [56], [57], [58], [59], [58]. In [54], the authors have proposed the filtered facet normal descriptor (FND) and have used multiple iterations of neural network which maps the FND to the facet normals of the denoised mesh.

III. DMD-Net: Deep Mesh Denoising Network

In this section, first we introduce the mesh denoising problem. Then, we describe the structure of DMD-Net and the motivation behind it. Lastly, we mention the loss functions used for training DMD-Net.

A. Problem Statement

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P})$, in which $\mathcal{V}$ is the set of vertices, $\mathcal{E} \subseteq \mathcal{V}^2$ is the set of edges, $\mathcal{F} \subseteq \mathcal{V}^3$ is the set of faces, and $\mathcal{P}$ is the vertex-wise feature matrix. Let $|\mathcal{V}| = n$, $|\mathcal{E}| = m$, $|\mathcal{F}| = f$, and $\mathcal{P}$ is a matrix of size $n \times 3$, corresponding to the 3D spatial coordinates of the vertices. Now, suppose that due to a process of signal corruption, some noise gets introduced into the feature matrix of the graph. Let the graph obtained as a result of noise corruption be denoted as $\mathcal{G}' = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P}')$, where $\mathcal{P}' = \mathcal{P} + \mathcal{N}$, in which $\mathcal{N}$ is the noise matrix coming

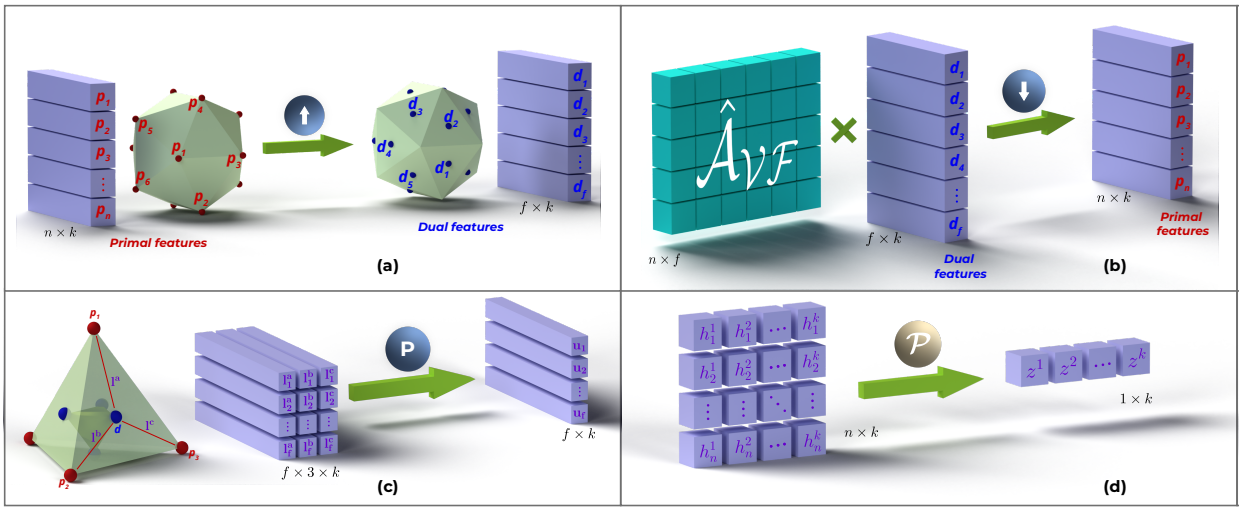


Fig. 3: The proposed DMD-Net architecture for mesh denoising: (a) Primal to Dual (P2D) layer, (b) Dual to Primal (D2P) layer, (c) Dual Average Pooling (DAP) layer, and (d) Feature Average Pooling (FAP) layer.

from some noise distribution. Note that both $\mathcal{G}$ and $\mathcal{G}'$ share the same edge and face connectivity. Recovering $\mathcal{G}$ from $\mathcal{G}'$ is the major goal of graph denoising. Since, the graphs under consideration are 3D triangulated meshes, graph denoising in this case will be referred to as mesh denoising.

B. Network Architecture

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{P})$, let $\mathcal{A}_\mathcal{V}$, $\mathcal{A}_\mathcal{F}$, and $\mathcal{A}_{\mathcal{VF}}$ denote the vertex adjacency matrix, the face adjacency matrix, and the vertex-face adjacency matrix, respectively. DMD-Net is a Graph-CNN that is designed to solve the problem of mesh denoising. It performs graph aggregation in both the primal graph $\mathcal{G}_\mathcal{V} = (\mathcal{V}, \mathcal{A}_\mathcal{V}, \mathcal{X}_\mathcal{V})$ as well as the dual graph $\mathcal{G}_\mathcal{F} = (\mathcal{F}, \mathcal{A}_\mathcal{F}, \mathcal{X}_\mathcal{F})$, where $\mathcal{X}_\mathcal{V}$ and $\mathcal{X}_\mathcal{F}$ are the primal and the dual features, respectively. Here, $\mathcal{G}_\mathcal{V}$ and $\mathcal{G}_\mathcal{F}$ are defined as triplets for the sake of convenience. A brief audio-visual explanation of DMD-Net is included in the supplementary video.

DMD-Net* is based on the Feature Guided Transformer (FGT) paradigm and consists of three main components: the feature extractor, the transformer, and the denoiser as shown in Figure 2(a). We train the feature extractor to estimate normal vector, mean curvature, and Gaussian curvature for each vertex of the original mesh from the given noisy mesh. These estimated local features serve as guidance for the transformer to compute a transformation matrix $\mathcal{W}_{tf}$.

The feature extractor (Figure 2(b)) internally contains a pair of two-stream networks which have two parallel streams, the upper one called the dual stream and the lower one called the primal stream. The transformer network consists of different layers as shown in Figure 2(c). We pool the final features using feature average-pooling layer (FAP) (Figure 3(d)) to learn the transformation $\mathcal{W}_{tf}$. The noisy input mesh is concatenated with the local features extracted by feature extractor network, on which the transformation $\mathcal{W}_{tf}$ is applied. These transformed features guide the denoiser network to accurately estimate a denoised representation of the input noisy mesh. The denoiser has a structure identical to that of the feature extractor as depicted in Figure 2(d).

The two-stream network is an asymmetric module consisting of two parallel streams, the lower one for performing aggregation in the primal graph and the upper one for performing aggregation in the dual graph. It consists of the primal-to-dual layer, a cascade of aggregator layers and a primal dual fusion layer. The primal-to-dual layer (Figure 3(a)) converts the primal graph features $\mathcal{X}_\mathcal{V}$ into the dual graph features $\mathcal{X}_\mathcal{F}$. In dual graph representation, we represent the feature of each face as the centroid of the features of the vertices constituting that face. The Aggregator (AGG) (Figure 2(f)) performs graph aggregation by pooling in the features of the neighbouring nodes. We use the graph aggregation formulation as described in [60]. The input to AGG is $\mathcal{X}$ (feature matrix) and $\mathcal{A}$ (adjacency matrix). The input graph to AGG can be in both forms, primal as well as dual. The output of AGG is given by $g(\mathcal{X}, \mathcal{A}) = \sigma(\hat{\mathcal{D}}^{-\frac{1}{2}}\hat{\mathcal{A}}\hat{\mathcal{D}}^{-\frac{1}{2}}\mathcal{X}\mathcal{W})$. Here, σ is the ReLU activation function, $\mathcal{W}$ is the learnable weight matrix, $\hat{\mathcal{A}} = \mathcal{A} + \mathcal{I}$ ($\mathcal{I}$ being the identity matrix), and $\hat{\mathcal{D}}$ is the diagonal node degree matrix of $\hat{\mathcal{A}}$. The primal dual fusion (PDF) block (Figure 2(f)) fuses the aggregated features from both the streams at the facet level. This fusion serves as a point of communication between the primal-dual streams, allowing flow of information from one stream to the other. PDF employs a dual average pooling layer that intermixes the primal and dual stream features as shown in (Figure 3(c)). The fused feature representation after dual-average pooling is in dual form $\mathcal{X}_\mathcal{F}$, and is converted to primal form $\mathcal{X}_\mathcal{V}$ using dual-to-primal layer (Figure 3(b)). We obtain $\mathcal{X}_\mathcal{V}$ by pre-multiplying $\mathcal{X}_\mathcal{F}$ with the degree normalized vertex-face adjacency matrix $\hat{\mathcal{A}}_{\mathcal{VF}}$.

*A detailed description of network architecture (Feature extractor, Transformer, Denoiser), FGT paradigm, and proposed layers (P2D, D2P, DAP, FAP) is included in the supplementary material. We also present exhaustive ablation studies on the importance of each block and the final choice of DMD-Net architecture in the supplementary material.

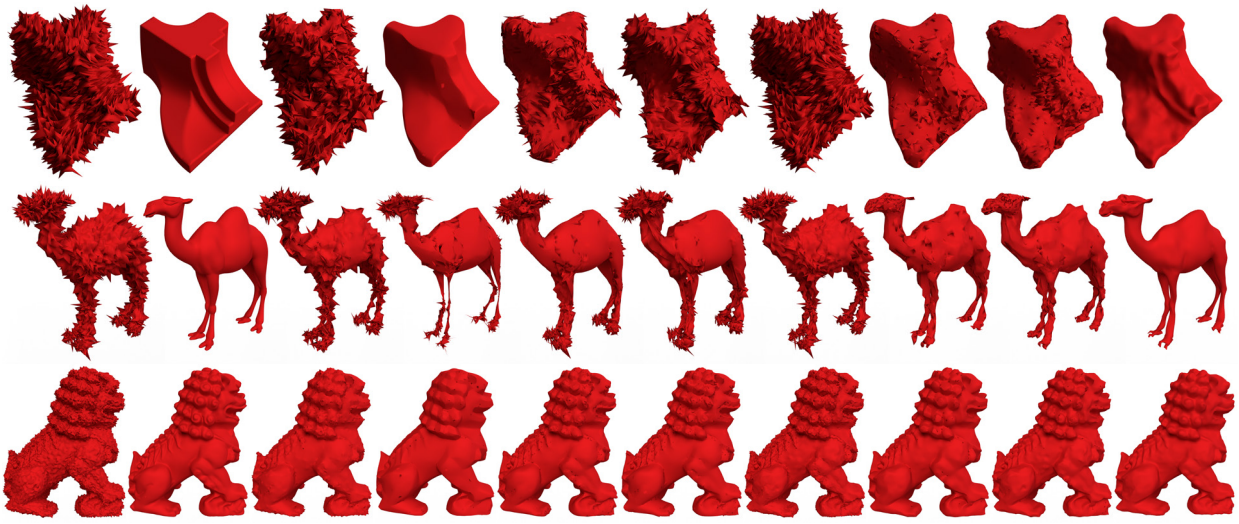


Fig. 4: Comparison of different mesh denoising methods on Fandisk (Noise: Gaussian $\mathcal{N}(0, 0.05)$), Camel (Noise: Gaussian $\mathcal{N}(0, 0.025)$) and Chineese Lion (Noise: Gaussian $\mathcal{N}(0, 0.005)$).

TABLE I
METRIC COMPARISON OF OUR METHOD WITH STATE-OF-THE-ART MESH DENOISING ALGORITHMS FOR SIX DIFFERENT MESHES

Object	Noise	Metric	**BMD**[14]	**L0M**[38]	**BNF**[16]	**FEFP**[24]	**NIFP**[15]	**GNF**[34]	**CNR**[54]	**Ours**
Block	$\mathcal{N}(0, 0.05)$	Normal	73.01	20.786	55.516	66.434	74.962	49.81	47.361	**20.315**
		Vertex ($\times 10^{-3}$)	2.173	0.231	0.713	1.172	2.215	0.393	0.402	**0.155**
		Chamfer	0.49	0.185	0.183	0.331	0.521	0.212	0.206	**0.135**
Bunny	$0.015 \cdot \Gamma(2, 2)$	Normal	20.094	8.679	24.487	33.516	28.954	18.928	11.747	**7.769**
		Vertex ($\times 10^{-3}$)	0.236	**0.218**	0.263	0.27	0.315	0.255	0.24	0.29
		Chamfer	**0.193**	0.216	0.207	0.195	0.252	0.226	0.228	0.258
Fertility	$0.035 \cdot \mathcal{U}(-1, 1)$	Normal	60.853	49.574	30.182	37.627	65.037	33.676	24.268	**11.285**
		Vertex ($\times 10^{-3}$)	0.294	0.209	0.07	0.135	0.259	0.085	**0.054**	0.063
		Chamfer	0.142	0.093	0.045	0.064	0.121	0.061	**0.043**	0.048
Chinese Lion	$\mathcal{N}(0, 0.005)$	Normal	20.505	16.864	16.19	17.581	29.482	14.547	15.068	**12.881**
		Vertex ($\times 10^{-3}$)	0.014	0.02	0.007	0.008	0.009	0.007	**0.006**	0.039
		Chamfer	0.012	0.019	0.005	0.007	0.008	0.006	**0.005**	0.023
Camel	$\mathcal{N}(0, 0.025)$	Normal	71.597	65.904	60.208	64.855	73.529	49.176	51.623	**24.726**
		Vertex ($\times 10^{-3}$)	0.519	0.393	0.288	0.313	0.449	0.132	0.114	**0.108**
		Chamfer	0.13	0.154	0.074	0.089	0.115	0.077	**0.059**	0.061
Fan Disk	$\mathcal{N}(0, 0.05)$	Normal	72.47	**17.249**	47.726	58.636	73.392	40.586	42.508	19.82
		Vertex ($\times 10^{-3}$)	2.166	0.213	0.656	0.954	2.064	0.398	0.414	**0.176**
		Chamfer	0.538	0.181	0.195	0.298	0.529	0.233	0.207	**0.146**

C. Loss Functions

To train our network, we use the following loss functions[†]: (a) **Vertex loss** ($\mathcal{L}_{vertex}$) - which computes the mean Euclidean distance between the corresponding vertices, (b) **Normal loss** ($\mathcal{L}_{normal}$) - which computes the mean angular deviation between the normals of the corresponding faces, (c) **Curvature loss** ($\mathcal{L}_{curvature}$) - where we compute the mean curvature error and the Gaussian curvature error of the vertices [4], (d) **Chamfer loss** [61] ($\mathcal{L}_{chamfer}$), and (e) **Feature Extractor loss** ($\mathcal{L}_{FE}$). During training, we use a linear combination of the loss functions described above. The weights used for vertex loss, normal loss, curvature loss, chamfer loss and the feature extractor loss are $\lambda_V = 1$, $\lambda_N = 0.2$, $\lambda_\kappa = 0.01$, $\lambda_C = 0.05$ and $\lambda_{FE} = 1$, respectively.

[†]A detailed description and mathematical formulation of all the loss functions is included in supplementary material. We also present ablation studies on the importance and choice of training weights for each loss function in the supplementary material.

IV. Results and Discussions

A. Dataset Generation

We use ShapeNet [62], a large scale repository of 3D models, as our dataset for training the network. ShapeNet has around 50k objects spanning 55 categories. The entire dataset is split into three parts: a) train, b) test-intra, and c) test-inter. For each of the 50 categories, 80% of the 3D models are included in train and the rest 20% is included in test-intra. All the objects in the remaining 5 held out classes are included in

TABLE II
PERFORMANCE OF DMD-NET ON DIFFERENT NOISE TYPES AND LEVELS

Noise		test-intra + test-inter			
		Vertex ($\times 10^{-4}$)		Normal (degrees)	
		Loss	Reference	Loss	Reference
Gamma noise	$0.03 \cdot \Gamma(2,2)$	3.995	13.345	33.239	72.186
	$0.04 \cdot \Gamma(2,2)$	4.699	23.724	37.331	75.908
	$0.05 \cdot \Gamma(2,2)$	5.587	37.069	40.88	78.39
Gaussian Noise	$\mathcal{N}(0, 0.025)$	3.285	6.176	26.692	67.482
	$\mathcal{N}(0, 0.05)$	3.952	24.705	34.372	77.088
	$\mathcal{N}(0, 0.1)$	6.478	98.82	45.334	83.268
Impulse Noise	$0.05 \cdot \delta(0.15, 0.15)$	3.333	7.408	27.424	47.756
	$0.1 \cdot \delta(0.15, 0.15)$	4.118	29.633	35.377	52.65
	$0.2 \cdot \delta(0.15, 0.15)$	7.074	118.531	46.306	55.351
Uniform noise	$0.05 \cdot \mathcal{U}(-1, 1)$	3.366	8.233	28.158	71.154
	$0.1 \cdot \mathcal{U}(-1, 1)$	4.254	32.932	36.677	79.704
	$0.2 \cdot \mathcal{U}(-1, 1)$	7.698	131.729	48.307	84.799

test-inter. The objective of test-inter dataset is to evaluate how well our network performs on categories it has not seen before. We include a detailed description of data preparation process and creation of noisy and its corresponding ground-truth counterpart in supplementary material. We further augment the data by rotating the mesh in random orientation in every epoch of training.

The final dataset contains 80357 meshes to which rotation augmentation and noise is added. This dataset is split into train, test-intra and test-inter containing 61842, 15377, and 3138 meshes respectively. During training our network, each epoch consists of 10^4 meshes which are randomly selected from the training set.

B. Results

In Table I, we quantitatively compare our method with several existing mesh denoising algorithms on some of the popular meshes which are not part of the ShapeNet dataset. To each of these meshes we add a different type and level of noise. To compare the different methods we use the following metrics: Normal loss, Vertex loss and Chamfer loss. Note that in terms of angular divergence (Normal loss), our method shows significantly better performance compared to other methods. The resulting denoised mesh obtained using various method are visually compared in Figure 4. Further, we show the visual results of DMD-Net on nine popular meshes in Figure 1. A low noise of $\mathcal{N}(0, 0.01)$ is used for all the nine meshes.

Since CNR[54] uses a learning based framework, we perform the following two comparisons. First we train both DMD-Net and CNR network on the ShapeNet training dataset and compare their performance on test-intra and test-inter sets. We then train both DMD-Net and CNR network on the CNR train dataset and then compare their performance on the CNR test dataset. This comparison is shown in Table III. We show the visual results of the four methods, as mentioned above, in Figure 6. We find the results of DMD-Net visually more plausible and close to the ground truth in the visual sense. Further, in Figure 7, we present a visual comparison of several denoising methods on Kinect scans from the CNR dataset, and show that, DMD-Net achieves competitive performance.

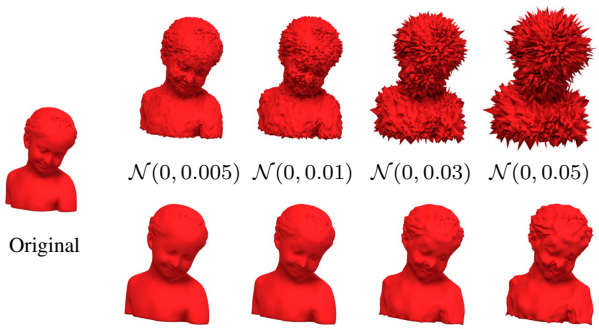


Fig. 5: Output of DMD-Net on the bust model under Gaussian noise of various levels. The top row denotes the noisy mesh and the bottom row denotes the denoised result of DMD-Net.

During training, we use four different noise types each with five different noise levels. We show qualitative results under different noise levels in Figure 5. We also evaluate the performance of DMD-Net in case of noise levels that were not included during training in Table II. Here, reference value refers to the metric distance between the noisy and the original mesh. As can be seen in Figure 5, our method performs very well even in the presence of high noise.

C. Hyperparameters

DMD-Net contains around 30 million learnable parameters. We use ADAM [63] optimizer with the following exponential decay rates: $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set our initial learning rate as $lr = 10^{-4}$. We train our final network for 200 epochs. One epoch takes about an hour on NVIDIA Quadro RTX 5000.

D. Ablation Studies

We conduct several ablation studies[‡], where, we devise several variants of the proposed approach and show that the proposed approach outperforms all the variants. These include Training Scheme Ablation, Dropout Rate Ablation, DMD-Net Structure Ablation, Loss Function Ablation, and Depth Ablation studies. In all of our ablation studies, we train the networks on mixed noise, that is, we randomly choose the noise type and the noise level in each iteration. For the depth ablation, we train all the variants for 200 epochs. For the rest of the ablation studies, we train all the variants for 60 epochs. Except the loss ablation study and the training scheme ablation study, all other ablation studies use a linear combination of loss functions with the following weights: $\lambda_V = \lambda_N = \lambda_\kappa = \lambda_{FE} = \lambda_C = 1$.

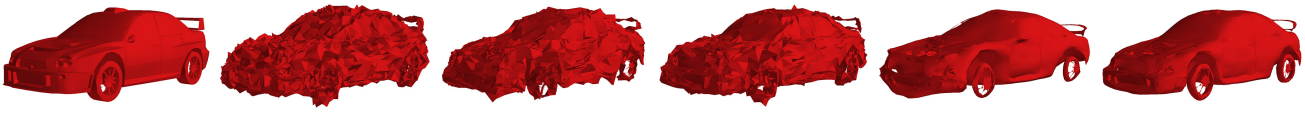
E. Transformation Equivariance

Let $\mathcal{T}$ be a transformation operator, let $\mathcal{D}$ be a mesh denoising algorithm and let $\mathcal{G}$ denote a mesh. We say that $\mathcal{D}$ has $\mathcal{T}$-equivariance, if for any given mesh, $\mathcal{G}$ we have $\mathcal{T}(\mathcal{D}(\mathcal{G})) = \mathcal{D}(\mathcal{T}(\mathcal{G}))$. That is, $\mathcal{D}$ is $\mathcal{T}$-equivariant, if $\mathcal{D}$ and $\mathcal{T}$ commute. We now discuss whether DMD-Net is equivariant with respect to the following mentioned transformations.

[‡]Ablation study Tables and their detailed explanations are presented in the supplementary material.

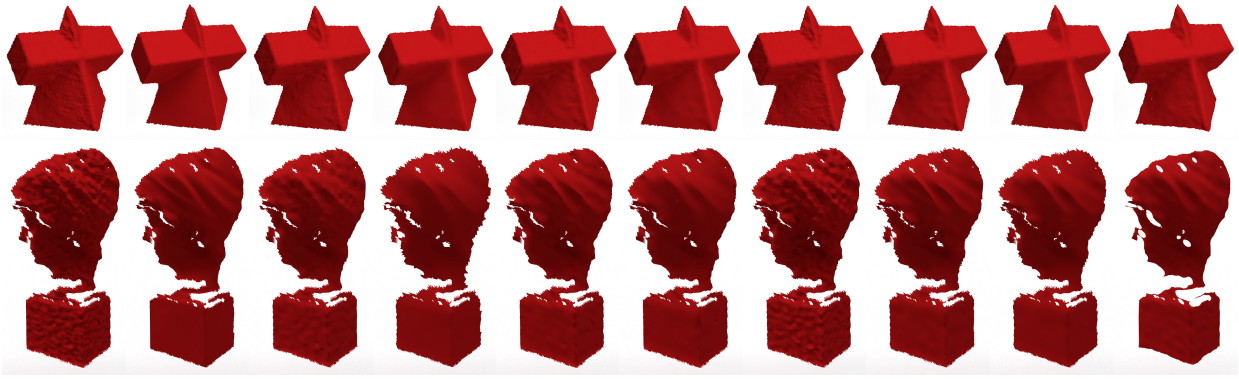
TABLE III
COMPARISON OF DMD-NET WITH CNR[54] ON BOTH THE SHAPENET AS WELL AS THE CNR DATASET

Model	Trained On ShapeNet dataset						Trained on CNR dataset		
	test-intra ShapeNet			test-inter ShapeNet			test CNR Dataset		
	Vertex ($\times 10^{-4}$)	Normal (degrees)	Chamfer ($\times 10^{-4}$)	Vertex ($\times 10^{-4}$)	Normal (degrees)	Chamfer ($\times 10^{-4}$)	Vertex ($\times 10^{-4}$)	Normal (degrees)	Chamfer ($\times 10^{-4}$)
CNR[54]	4.522	54.221	1.789	4.661	53.824	1.947	**0.978**	23.043	**0.591**
DMD-Net	**3.301**	**25.37**	**1.786**	**3.271**	**25.053**	**1.815**	2.561	**14.369**	1.372



(a) (b) (c) (d) (e) (f)

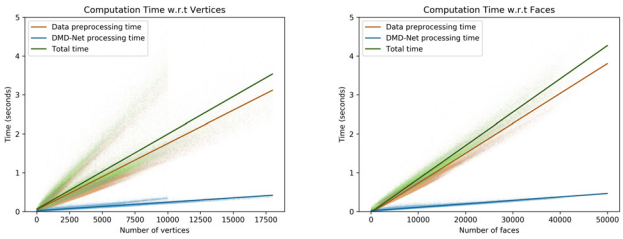
Fig. 6: Comparison of DMD-Net and CNR[54] trained on both ShapeNet and CNR dataset. (a) Ground truth, (b) Noisy Input, Output of (c) CNR network trained on CNR dataset, (d) CNR network trained on ShapeNet, (e) DMD-Net trained on CNR dataset, and (f) DMD-Net trained on ShapeNet.



Noisy Original BMD[14] L0M[38] BNF[16] FEFP[24] NIFP[15] GNF[34] CNR[54] Ours

Fig. 7: Comparison of different mesh denoising methods on Kinect scans from the CNR dataset.

Given a mesh, we first normalize it to fit inside a unit cube and shift the mesh to the origin. We then denoise it using DMD-Net by un-normalizing it back to it's original scale and shifting it back to its original location. Thus, DMD-Net is scale and translation equivariant as it first converts the mesh into a cannonical representation before denoising. DMD-Net is not rotation equivariant in the theoretical sense. However, the network learns to preserve equivariance to a high level. The study on rotation equivariance is presented in supplementary material along with both quantitative and qualitative results.



(a) (b)

Fig. 8: Growth of computation time with increase in (a) number of vertices and (b) number of faces.

F. Computation Time

The visual results shown in Figure 8 contains a scatter-plot of data points along with the trendlines. As can be seen in the figures, our algorithm possesses a linear time growth which is an advantage of our method. When executed on a GPU, there is a significant speedup in the inference time of DMD-Net. The high speed and accuracy of DMD-Net makes it a suitable candidate for real-time mesh denoising.

V. CONCLUSION AND FUTURE WORK

We have proposed a deep graph learning based framework to solve the mesh denoising problem. Through various experiments, we show that our method outperforms all proposed variants in the ablation studies. We also show that our method strikes a good balance between eliminating noise and avoiding over-smoothing. In this work, we make the assumption that the noise introduced during degradation is additive in nature. It would be interesting to explore how our methods performs on different types of noise models. DMD-Net is highly efficient in terms of computation time. In future, we propose to make it memory efficient by using model compression techniques, that would enable us to efficiently implement our method on a mobile phone or to directly integrate it into a portable 3D scanner device.

VI. Acknowledgements

This work is supported by Science & Engineering Research Board, India under SERB IMPRINT 2 and SERB MATRICS grants.

References

[1] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 351–358.

[2] J. Vollmer, R. Mencl, and H. Mueller, "Improved laplacian smoothing of noisy surface meshes," in *Computer graphics forum*, vol. 18, no. 3. Wiley Online Library, 1999, pp. 131–138.

[3] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. Citeseer, 1999, pp. 317–324.

[4] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and mathematics III*. Springer, 2003, pp. 35–57.

[5] C. L. Bajaj and G. Xu, "Anisotropic diffusion of surfaces and functions on surfaces," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 1, pp. 4–32, 2003.

[6] U. Clarenz, U. Diewald, and M. Rumpf, *Anisotropic geometric diffusion in surface processing*. IEEE, 2000.

[7] A. F. El Ouafdi and D. Ziou, "A global physical method for manifold smoothing," in *2008 IEEE International Conference on Shape Modeling and Applications*. IEEE, 2008, pp. 11–17.

[8] Y. Ohtake, A. G. Belyaev, and I. A. Bogaevski, "Polyhedral surface smoothing with simultaneous mesh regularization," in *Proceedings Geometric Modeling and Processing 2000. Theory and Applications*. IEEE, 2000, pp. 229–237.

[9] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Anisotropic feature-preserving denoising of height fields and bivariate data." in *Graphics interface*, vol. 11, no. 10. Citeseer, 2000, pp. 145–152.

[10] S. Tsuchie and M. Higashi, "Surface mesh denoising with normal tensor framework," *Graphical Models*, vol. 74, no. 4, pp. 130–139, 2012.

[11] H. Huang and U. Ascher, "Fast denoising of surface meshes with intrinsic texture," *Inverse Problems*, vol. 24, no. 3, p. 034003, 2008.

[12] ——, "Surface mesh smoothing, regularization, and feature detection," *SIAM Journal on Scientific Computing*, vol. 31, no. 1, pp. 74–93, 2008.

[13] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

[14] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM transactions on graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 950–953.

[15] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," in *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 943–949.

[16] Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai, "Bilateral normal filtering for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1521–1530, 2010.

[17] K.-W. Lee and W.-P. Wang, "Feature-preserving mesh denoising via bilateral normal filtering," in *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*. IEEE, 2005, pp. 6–pp.

[18] J. Solomon, K. Crane, A. Butscher, and C. Wojtan, "A general framework for bilateral and mean shift filtering," *arXiv preprint arXiv:1405.4734*, vol. 1, no. 2, p. 3, 2014.

[19] G. Taubin *et al.*, "Linear anisotropic mesh filtering," *Res. Rep. RC2213 IBM*, vol. 1, no. 4, 2001.

[20] Y. Ohtake, A. Belyaev, and I. Bogaevski, "Mesh regularization and adaptive smoothing," *Computer-Aided Design*, vol. 33, no. 11, pp. 789–800, 2001.

[21] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh smoothing via mean and median filtering applied to face normals," in *Geometric Modeling and Processing. Theory and Applications. GMP 2002. Proceedings*. IEEE, 2002, pp. 124–131.

[22] H. Yagou, Y. Ohtake, and A. G. Belyaev, "Mesh denoising via iterative alpha-trimming and nonlinear diffusion of normals with automatic thresholding," in *Proceedings Computer Graphics International 2003*. IEEE, 2003, pp. 28–33.

[23] Y. Shen and K. E. Barner, "Fuzzy vector median-based surface smoothing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 3, pp. 252–265, 2004.

[24] X. Sun, P. L. Rosin, R. Martin, and F. Langbein, "Fast and effective feature-preserving mesh denoising," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 5, pp. 925–938, 2007.

[25] X. Sun, P. L. Rosin, R. R. Martin, and F. C. Langbein, "Random walks for feature-preserving mesh denoising," *Computer Aided Geometric Design*, vol. 25, no. 7, pp. 437–456, 2008.

[26] S. K. Yadav, U. Reitebuch, and K. Polthier, "Robust and high fidelity mesh denoising," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 6, pp. 2304–2310, 2018.

[27] H. Fan, Y. Yu, and Q. Peng, "Robust feature-preserving mesh denoising based on consistent subneighborhoods," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 2, pp. 312–324, 2009.

[28] M. Wei, J. Yu, W.-M. Pang, J. Wang, J. Qin, L. Liu, and P.-A. Heng, "Bi-normal filtering for mesh denoising," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 1, pp. 43–55, 2014.

[29] L. Zhu, M. Wei, J. Yu, W. Wang, J. Qin, and P.-A. Heng, "Coarse-to-fine normal filtering for feature-preserving mesh denoising based on isotropic subneighborhoods," in *Computer Graphics Forum*, vol. 32, no. 7. Wiley Online Library, 2013, pp. 371–380.

[30] J. Wang, X. Zhang, and Z. Yu, "A cascaded approach for feature-preserving surface mesh denoising," *Computer-Aided Design*, vol. 44, no. 7, pp. 597–610, 2012.

[31] X. Lu, Z. Deng, and W. Chen, "A robust scheme for feature-preserving mesh denoising," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 3, pp. 1181–1194, 2015.

[32] M. Centin and A. Signoroni, "Mesh denoising with (geo) metric fidelity," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 8, pp. 2380–2396, 2017.

[33] M. Wei, L. Liang, W.-M. Pang, J. Wang, W. Li, and H. Wu, "Tensor voting guided mesh denoising," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 931–945, 2016.

[34] W. Zhang, B. Deng, J. Zhang, S. Bouaziz, and L. Liu, "Guided mesh normal filtering," in *Computer Graphics Forum*, vol. 34, no. 7. Wiley Online Library, 2015, pp. 23–34.

[35] T. Li, W. Liu, H. Liu, J. Wang, and L. Liu, "Feature-convinced mesh denoising," *Graphical Models*, vol. 101, pp. 17–26, 2019.

[36] R. Wang, Z. Yang, L. Liu, J. Deng, and F. Chen, "Decoupling noise and features via weighted $\ell 1$-analysis compressed sensing," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 2, p. 18, 2014.

[37] X. Wu, J. Zheng, Y. Cai, and C.-W. Fu, "Mesh denoising using extended rof model with L_1 fidelity," in *Computer Graphics Forum*, vol. 34, no. 7. Wiley Online Library, 2015, pp. 35–45.

[38] L. He and S. Schaefer, "Mesh denoising via L_0 minimization," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 64, 2013.

[39] H. Zhang, C. Wu, J. Zhang, and J. Deng, "Variational mesh denoising using total variation and piecewise constant function space," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 7, pp. 873–886, 2015.

[40] S. Fleishman, D. Cohen-Or, and C. T. Silva, "Robust moving least-squares fitting with sharp features," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 544–552.

[41] H. Sheung and C. C. Wang, "Robust mesh reconstruction from unoriented noisy points," in *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*. ACM, 2009, pp. 13–24.

[42] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, "Edge-aware point set resampling," *ACM transactions on graphics (TOG)*, vol. 32, no. 1, p. 9, 2013.

[43] O. Remil, Q. Xie, X. Xie, K. Xu, and J. Wang, "Surface reconstruction with data-driven exemplar priors," *Computer-Aided Design*, vol. 88, pp. 31–41, 2017.

[44] Q. Zheng, A. Sharf, G. Wan, Y. Li, N. J. Mitra, D. Cohen-Or, and B. Chen, "Non-local scan consolidation for 3d urban scenes," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 94–1, 2010.

[45] G. Rosman, A. Dubrovina, and R. Kimmel, "Patch-collaborative spectral point-cloud denoising," in *Computer Graphics Forum*, vol. 32, no. 8. Wiley Online Library, 2013, pp. 1–12.

[46] S. Yoshizawa, A. Belyaev, and H.-P. Seidel, "Smoothing by example: Mesh denoising by averaging with similarity-based weights," in *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. IEEE, 2006, pp. 9–9.

[47] B. Dong, J. Ye, S. Osher, and I. Dinov, "Level set based nonlocal surface restoration," *Multiscale Modeling & Simulation*, vol. 7, no. 2, pp. 589–598, 2008.

[48] X. Xie, X. Guo, G. Liu, and J. Wang, "Implicit block diagonal low-rank representation," *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 477–489, 2017.

[49] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 11, pp. 2268–2281, 2015.

[50] M. Wei, J. Huang, X. Xie, L. Liu, J. Wang, and J. Qin, "Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery," *IEEE transactions on visualization and computer graphics*, 2018.

[51] J. Wang, J. Huang, F. L. Wang, M. Wei, H. Xie, and J. Qin, "Data-driven geometry-recovering mesh denoising," *Computer-Aided Design*, vol. 114, pp. 133–142, 2019.

[52] S. Nousias, G. Arvanitis, A. S. Lalos, and K. Moustakas, "Fast mesh denoising with data driven normal filtering using deep autoencoders," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, vol. 1, 2019, pp. 260–263.

[53] J. R. Diebel, S. Thrun, and M. Brünig, "A bayesian method for probable surface reconstruction and decimation," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 1, pp. 39–59, 2006.

[54] P.-S. Wang, Y. Liu, and X. Tong, "Mesh denoising via cascaded normal regression." *ACM Trans. Graph.*, vol. 35, no. 6, pp. 232–1, 2016.

[55] W. Zhao, X. Liu, Y. Zhao, X. Fan, and D. Zhao, "Normalnet: Learning based guided normal filtering for mesh denoising," *arXiv preprint arXiv:1903.04015*, 2019.

[56] M. Wei, X. Guo, J. Huang, H. Xie, H. Zong, R. Kwan, F. L. Wang, and J. Qin, "Mesh defiltering via cascaded geometry recovery," *Computer Graphics Forum*, vol. 38, no. 7, pp. 591–605, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13863

[57] G. Arvanitis, A. Lalos, and K. Moustakas, "Feature-aware and content-wise denoising of 3d static and dynamic meshes using deep autoencoders," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 97–102.

[58] X. Li, R. Li, L. Zhu, C.-W. Fu, and P.-A. Heng, "Dnf-net: A deep normal filtering network for mesh denoising," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[59] M. Armando, J.-S. Franco, and E. Boyer, "Mesh denoising with facet graph convolutions," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[60] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[61] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, Tech. Rep., 1977.

[62] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.