Laplacian Mesh Transformer: Dual Attention and Topology Aware Network for 3D Mesh Classification and Segmentation

Xiao-Juan Li[1,2], Jie Yang (✉)[1,2], and Fang-Lue Zhang[3]

[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] Victoria University of Wellington
{lixiaojuan,yangjie01}@ict.ac.cn fanglue.zhang@vuw.ac.nz

Abstract. Deep learning-based approaches for shape understanding and processing tasks have attracted considerable attention. Despite the great progress that has been made, the existing approaches fail to efficiently capture sophisticated structure information and critical part features simultaneously, limiting their capability of providing discriminative deep shape features. To address the above issue, we proposed a novel deep learning framework, Laplacian Mesh Transformer, to extract the critical structure and geometry features. We introduce a dual attention mechanism, where the 1st level self-attention mechanism is used to capture the critical partial/local structure and geometric information on the entire mesh, and the 2nd level is to fuse the geometrical and structural features together with the learned importance according to a specific downstream task. More particularly, Laplacian spectral decomposition is adopted as our basic structure representation given its ability to describe shape topology (connectivity of triangles). Our approach builds a hierarchical structure to process shape features from fine to coarse using the dual attention mechanism, which is stable under the isometric transformations. It enables an effective feature extraction that can tackle 3D meshes with complex structure and geometry efficiently in various shape analysis tasks, such as shape segmentation and classification. Extensive experiments on the standard benchmarks show that our method outperforms state-of-the-art methods.

Keywords: Laplacian EigenVector, Transformer, Attention Mechanism, Topology Aware, Shape Segmentation & Classification

1 Introduction

3D data analysis has been an important topic in computer graphics and computer vision research. Numerous tasks in semantic understanding [28,3], 3D detection [90,10], shape abstraction [69,65] rely on the advanced 3D shape analysis and understanding technology, especially for the urgent requirements in autonomous driving, virtual/augmented reality, robotics, and model creation.

As an essential method to represent 3D shapes, polygonal meshes have been successfully used in the above applications for efficient modeling and rendering of 3D objects. To make it possible to learn the features of 3D meshes of neural networks, many large-scale datasets (*e.g.* ShapeNet [9], ModelNet [79]) are built and made available to the public. Considering that the polygonal meshes describe the detailed surfaces (including the geometry and structures) by a set of 2D polygons [6], some voxelized approaches [46,12] extend the 2D deep learning methods to the 3D domain. However, it suffers from massive computation and memory demands and thus has a limited capacity to cope with high-resolution mesh data. Other pioneering works focus on learning features from point clouds to perform 3D data analysis, such as PointNet [54] and PointNet++ [56]. They have achieved good performances on segmentation and classification by multi-layer perceptrons (MLPs) or dynamic graphs [75]. Although point cloud is lightweight and mitigates the computation cost issue, it lacks topological information compared to the polygonal meshes. Therefore, the prior deep learning-based methods fail to capture complex structural information and partial features for shape analysis.

This work focuses on polygonal meshes and develops a novel deep architecture based on self-attention [70] to learn 3D shape features in a topology-aware manner. The design of our network is based on two key observations. Firstly, the eigenvectors obtained from a Laplacian spectral decomposition on meshes are used to shape analysis [44,41] and indicate some topological information(*e.g.* symmetry), which can be naturally used as a representation of the topology of 3D meshes. Secondly, the relationships among the elements of the structural and partial geometric features can well represent the 3D meshes and their parts in a discriminative way. The self-attention mechanism used in Transformer [70,14,13,83] has shown its capability of extracting the relationships among all the elements of input signals in natural language processing and image analysis [88,58,29], which can be adopted to analyze 3D shapes [89,18,86] more effectively, especially for 3D meshes [17].

This paper builds a dual attention architecture in a topology-aware fashion, Laplacian Mesh Transformer, to understand the complex structure and rich geometric information of 3D meshes. Our method takes raw mesh features as input and produces global features containing effective descriptions of topology and geometry information. There are two branches (see Figure 1) that simultaneously learn critical geometric features and structural features from the Euclidean coordinates and Laplacian spectral decomposition, respectively. In both branches, we extract features of different scales by four self-attentions. Then we apply a final attention-based fusion module to learn the importance of the topology and geometry information and fuse them to form the final global features when applying to different downstream tasks. With the help of the dual attention mechanism and Laplacian spectral decomposition, we build a hierarchical structure from fine (partial) to coarse (global) to process shape features. Compared to alternative methods, our approach utilizes both spatial and spectral information by dual

attention and is able to dynamically determine the contribution of topology and geometry features during inference for a better shape understanding.

Our network architecture is illustrated in Figure 1. To demonstrate the aptitude of our approach to describe mesh features, we build two downstream networks to perform shape segmentation and classification on the ShapeNet [9] and COSEG [76] datasets, which are the fundamental shape analysis tasks in computer vision. Our extensive experiments demonstrate the robustness of our Laplacian Mesh Transformer to various vertex types and different triangulations of meshes. Our method achieves remarkable performance in both segmentation and classification with a lightweight network, and it can also be potentially applied to other tasks, such as shape retrieval.

The main contributions of our method are as follows:

- We design a dual attention mechanism for learning features on 3D polygonal meshes, which takes eigenvectors from Laplacian spectral decomposition as the raw topological description;
- We propose a deep architecture that focuses on sophisticated polygonal meshes and takes the partial geometric/structural features and their importance into consideration in a fine (partial) to coarse (global) manner;
- We conduct extensive experiments on multiple 3D shape analysis tasks to demonstrate our superior effectiveness compared to state-of-the-art methods.

2 Related Works

This section briefly reviews learning-based methods in the 3D domain and then summarizes the popular self-attention-based work, which is helpful for many applications.

Deep Learning on 3D Domain. With the increased availability of 3D models and the development of deep learning frameworks, there are various approaches to analysis and modeling 3D models, thanks to the mighty deep learning tools nowadays. For different representations of 3D data, recent works have been developed for voxels [46,12,55,78], multi-view images of 3D data [64,34,63], point cloud [54,56,19,39,1], meshes [66,73,32,23,20,82], and implicit functions [52,50,47,11]. Voxels represent values on a regular grid in three-dimensional space, which are similar to pixels inside of a 2D image. Some operators of deep learning could be extended and applied to the 3D voxels naturally. For the multi-view images of 3D data, a shape can be rendered into multiple images from different views. By applying the traditional 2D image CNNs to these 2D images from different views, the entire model is represented by aggregating the features of these images. The point cloud is a general representation of any 3D shape, which is easy to capture with portable devices. Many works solve the following challenges: noisy, sparse, and disordered. Compared to point clouds, meshes are considered a better representation of concrete geometric shapes and structures. Nevertheless, it is tough to learn on the meshes given their irregularity. [32] reconstructs a 3D mesh by the laplacian in an extrinsic/intrinsic manner, but ours uses the laplacian eigenvectors and attention to help the deep

networks understand shapes. For the comprehensive and detailed review of deep learning on 3D data, we refer the readers to these surveys [7,33,2,80].

Self-Attention Mechanism. The self-attention mechanism is widely used in many natural language processing (NLP) and computer vision tasks. The survey papers [25,37,67,42] have comprehensive discussion on the attention mechanism. Bahdanau *et al.* [4] first adopts the attention mechanism (soft-search) into the neural machine translation, the attention map is predicted to summarize the contextual relationships by bidirectional RNN [60]. Lin *et al.* [43] introduces a model for learning an interpretable embedding by self-attention. Followed by this, [70] proposed the transformer and applied it to the machine translation, which does not depend on the convolution operator, and achieves promising results by utilizing the global context. Furthermore, the researchers made a great effort to develop and expand the transformer, such as XLNET [83], a two-way transformer – BERT [15]. However, in the NLP field, the sentences are sequential and semantic meaningful, while the vertex on 3D meshes are usually disordered and have no semantics.

At the same time, self-attention also makes great potential impacts and receives more and more attention in computer vision (*e.g.* object detection – DETR [8]). Wang *et al.* [72] proposed a residual attention module for image classification in a stacked manner. Zhang *et al.* [87] designed a generative model with self-attention, which enables attention-driven and long-range dependency modeling for image GAN [22]. Recently, Visual Transformer [77] and Vision Transformer (ViT) [16] interpret an image as semantic visual tokens and sequential patches, respectively, then apply the transformer to the above sequential data. They all exceed the performance of CNN-based methods on image processing tasks when the training data is sufficient.

Inspired by the local patch structure used in ViT and the basic semantic information in language words, we propose a dual attention module based on self-attention, which can take the geometry and topology into consideration, capturing the local partial criticism on the 3D meshes and obtaining crucial semantic information.

3 Methodology

3.1 Overview

Given a 3D polygonal mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}|$ vertices and $|\mathcal{E}|$ edges, our goal is to let the network learn a function $f : V \to \mathbb{R}^{|S|}$ that maps vertex features to vector space S. For the classification task, $|S|$ could be 1 since the whole shape has one attribute; For the segmentation task, $|S| = |\mathcal{V}|$ because each vertex of the shape has a attribute. The vertex feature typically contains coordinates, normal, curvatures, PCA, etc. We aim to design a network that can learn a general function f, describing the importance of geometric and structural contexts. So we proposed a dual attention mechanism to learn a reasonable fusion between geometry and topology to improve the performance according to the global context. Our network takes the coordinates, normal, Laplacian eigenvector of the

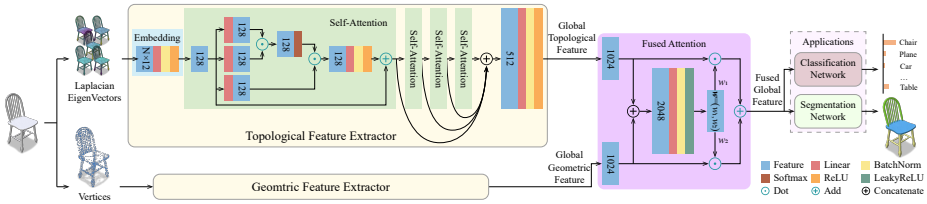


Fig. 1: **Our framework for shape analysis.** The Laplacian eigenvectors and vertex coordinates are fed the first level of our dual attention mechanism, where the two self-attention branches for topology and geometry features have the same architecture. The second level of attention in the fusing module merges the two sets of features with learned importance to generate the final global feature for the whole shape, which can be used to perform some downstream tasks.

vertices of a mesh model as input and predicts the probability matrix with size $|\mathcal{V}| \times l_s$ for each vertex for shape segmentation and probability score with size l_c for each category on the entire shape for classification. In the following sections, we briefly revisit the formulation of Laplacian spectral decomposition in Sec. 3.2. Then, we further present the dual attention mechanism (Sec. 3.3) on geometry and topology for 3D polygonal meshes, which aims to capture the partial critical features and the importance of geometry and topology. Lastly, we describe our entire network architecture (Sec. 3.4) that determines the importance of geometry and topology and feeds the fused feature to perform classification, segmentation, or other tasks. Our network considers the shape's geometry and topology simultaneously and adjusts their importance by the attention mechanism to achieve adequate shape understanding.

3.2 Laplacian Spectral Decomposition

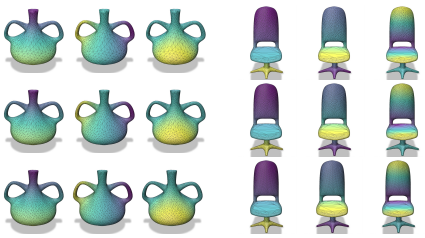
The Laplacian Spectral Decomposition effectively describes the mesh topology and geometric properties, *i.e.* the connectivity of vertex or symmetry. Figure 2 visualizes Laplacian eigenvectors on some 3D meshes. An observation is that Laplacian eigenvectors are intuitive when visualized for the segmentation task. Hence, in most cases, if our networks can determine the instances' segments by laplacian features, our network can balance the weights by our dual attention to get more accurate segmentation. We have a discussion about that in subsection 4.4.

A mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E})$ with arbitrary vertices and different connectivity can be regraded as a graph with $|\mathcal{V}|$ nodes and $|\mathcal{E}|$ relationships. We can adapt the graph Laplacian matrix on the 3D mesh to capture the topology of vertices. The Laplacian spectral decomposition depends on the number of vertices and different triangulation. In practice, the Laplacian matrix is formulated as follows:

$$\mathbf{L} = \mathbf{A}^{-1}(\mathbf{D} - \mathbf{W}) \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ is a diagonal matrix that places the weights of each vertex on the diagonal of the matrix, the weight is defined as the Voronoi area of the one-ring triangles surrounding each vertex. $\mathbf{W} \in \mathbb{R}^{|V|\times|V|} = \{w_{i,j}\}$ is a cotangent weighted adjacent matrix that is sparse and a discretization of the continuous Laplacian on smooth surfaces [48]. $w_{i,j} \neq 0$ means that vertices v_i and v_j are connected by a edge on meshes, the value describes the cotangent weight [53,48] $w_{i,j} = \frac{1}{2}(cot\alpha_{ij} + cot\beta_{ij})$ of the edge, where α_{ij}, β_{ij} are the angles opposite of the mesh edge (v_i, v_j). $\mathbf{D} \in \mathbb{R}^{|V|\times|V|}$ is the degree matrix that is a diagonal matrix and each diagonal entry $d_{i,i} = \sum_{j=1}^{|V|} w_{i,j}$ is the sum of each row of the weighted adjacent matrix $\mathbf{A}$. After that, we calculate the eigenvector x of the Laplacian matrix $\mathbf{L}$: $det(\mathbf{L}x - \lambda\mathbf{I}) = 0$ according to [61]. We sort the absolute values of all the eigenvalues in an ascending order. In our paper, We use the eigenvectors corresponding to the first 12 eigenvalues as the descriptors of the topology of 3D meshes. In Sec. 4.4, we evaluate the performance when using different numbers of eigenvectors.

In Figure 2, we illustrate some visualized Laplacian eigenvectors on different meshes. We visualize the Laplacian eigenvectors on the meshes with different vertex numbers for each example. From the results, we can see that the Laplacian eigenvectors are robust to different discretizations when the meshes can be discretized into reasonable sets of triangles, which is suitable for revealing the topology of meshes.



(a) Vases (b) Chairs

Fig. 2: **Laplacian eigenvector visualization.** For the two examples from COSEG, we only visualize the first three eigenvectors of Laplacian spectral decomposition in different columns. From the top row to the bottom row, we show the eigeowvectors for the meshes with 2000, 3500, and 5000 vertices.

3.3 Dual Attention

Given a polygonal mesh $\mathcal{M}$, we use the above formulation to encode the topology $f_t \in \mathbb{R}^{|V|\times 12}$ of that mesh in the vertex feature. We also encode the shape geometry $f_g \in \mathbb{R}^{|V|\times 3}$ of 3D meshes represented by the vertex coordinates into the vertex feature. Now that each mesh can be represented as the vertex feature set $\{f_t, f_g\}$, which is defined on the vertex set $\mathcal{V}$ of mesh $\mathcal{M}$. Turning the 3D mesh graph into a vertex-wise feature set prevents using complex graph structures when training the network. Besides, the vertex-wise feature fits the self-attention operator [70], which is permutation-invariant and independent of the connection between vertices.

The structure of our dual attention is illustrated in Figure 1. There are two attention modules: one is the self-attention (adopted from PCT [24]) with two branches, another is the fusion attention which learns the importance of geometry and topology adaptive according to the global context. The features are firstly fed to the self-attention encoder Enc_g for encoding the geometry

feature f_g and the other self-attention encoder Enc_t for encoding the topology feature f_t. The two encoders Enc_g, Enc_t have a similar structure. For the encoder $Enc_t = \{enc^{emb}, enc^{sa}, enc^{cat}, enc^{slp}\}$ in the topology branch, it contains one embedding module enc^{emb}, four self-attention operators $\{enc_i^{sa}, i = 1, 2, 3, 4\}$, and one feature concatenation block enc^{cat}. At the end of the encoder, the feature map goes through a single layer of perceptrons enc^{slp} to generate the final feature f_t' for topology. We formulate the above process as:

$$\begin{aligned} f_t &= enc^{emb}(f_t), f_t^1 = enc_1^{sa}(f_t) \\ f_t^i &= enc_i^{sa}(f_t^{i-1}), i = 2, 3, 4 \\ f_t &= enc^{cat}(f_t^1, f_t^2, f_t^3, f_t^4), \quad f_t' = enc^{slp}(f_t) \end{aligned} \tag{2}$$

where enc^{emb} consists of two FC layers with batch-normalization and ReLU activation, which embeds the features into a 128-dimensional embedding space. enc^{sa} is a standard self-attention module, and its architecture is presented in Figure 1. enc^{cat} performs the concatenation of multiple feature maps. Finally, for the topology branch, the network Enc^t maps the Laplacian eigenvector $f_t \in \mathbb{R}^{|V|\times 12}$ into the feature space $f_t' \in \mathbb{R}^{128}$. The geometry branch perform the same process, Enc_g takes the coordinates $f_g \in \mathbb{R}^{|V|\times 3}$ as input and generates the geometry feature $f_g' \in \mathbb{R}^{128}$. The two branches do not share weights.

Furthermore, we proposed an attention-based fusion module enc^{fus} to merge two features f_t', f_g' with an attention mask adaptively. The module learns to adjust the attention mask for achieving better performance for the given shape analysis tasks. It takes the generated features f_t', f_g' as inputs and predicts the attention weights w_t, w_g for the feature fusion. The weights w_g and w_t are predicted by three FC layers and batch-norm layers. For the final output, we use the exponential function $exp(\cdot)$ to ensure the weights are non-negative. Namely, the final output f is:

$$\begin{aligned} (w_t, w_g) &= enc^{fus}(f_t', f_g') \\ f &= w_t f_t' + w_g f_g' \end{aligned} \tag{3}$$

where w_t, w_g are learnable weights and $w_t + w_g = 1$. w_t and w_g describe their contribution to the global feature f. In the end, the attention-based fusion module learns to determine the importance of geometry and topology according to the global context.

3.4 Network Architecture

We build deep network models based on our proposed dual attention module to perform shape analysis tasks. The overall pipeline is illustrated in Figure 3.

For any mesh $\mathcal{M}$, we can obtain its geometry feature f_g and topology feature f_t by the coordinates and Laplacian spectral decomposition. Then, we can obtain the global feature f that fuses the geometry feature f_g and topology feature f_t according to their learned importance by our proposed dual attention. Since

dual attention can understand more critical factors for shape understanding, it gives a reasonable criticism during inference, conforming to human intuition. We can then feed the global shape feature f to downstream network modules for shape analysis tasks. Particularly, we implement segmentation and classification networks to demonstrate the effectiveness of our approach.

Segmentation: Part segmentation predicts a vertex-wise function that can map the vertex features to semantic labels for each vertex, *i.e.* segment the whole shape into some meaningful semantic parts (*e.g.* arm, chair back, etc.). For learning the 3D shape in a cross-category manner, we simultaneously feed the category information as a one-hot vector and fused global feature to the segmentation network as shown in Figure 3, which comprises five MLPs, each including a linear layer, a batch-norm layer, and an activation (ReLU/LeakyReLU). Nevertheless, for the final output, we use $sigmoid(\cdot)$ as activation to predict the probabilities ($\in \mathbb{R}^{|V|\times k}$) of all semantic labels for each vertex. Note that we add one dropout layer for avoiding overfitting, k is the number of part semantics. Finally, the probabilities are turned to a semantic label by an $argmax$ function.

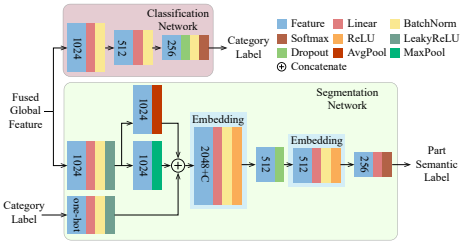


Fig. 3: **Architecture of segmentation and classification networks.** The segmentation network (bottom) first processes the fused global feature and the vector of labels. Then the processed global feature is duplicated, goes through average and max pooling, respectively, and is then concatenated with category features. Two embedding modules take the concatenated features to predict vertex-wise labels. In the classification network (top), the fused global features go through three MLPs, then a *softmax* operation is adopted to predict the probability.

Classification: This task aims to predict the probability of belonging to one semantic category for a given shape, which maps the input features to one semantic label for the whole mesh, such as chairs, tables, etc. Most parts of the classification network are the same as the segmentation network, but it predicts only one probability vector ($\in \mathbb{R}^k$) for the whole shape, where k is the number of categories.

4 Experiments & Evaluations

Laplacian Mesh Transformer is a general method for applying self-attention on triangular meshes to exploit shape features, enabling various applications, such as segmentation, classification, and shape retrieval. In this section, we present extensive quantitative and qualitative experiments on our shape classification and segmentation networks to evaluate the efficiency of the extracted features by Laplacian Mesh Transformer. We test our method and the existing deep models on three popular large scale datasets (*i.e.* ShapeNet [9], ModelNet [79],

COSEG [84]). We also perform ablation studies to demonstrate the effectiveness of our key components. The experiments were conducted on a computer with an i9-9900K CPU and an RTX 2080Ti GPU.

4.1 Implementation Details

We primarily use the above large datasets for our experiments. ShapeNet provides 16 categories with semantic part labels, and ModelNet contains 40 categories of CAD models without any semantic part labels. COSEG dataset contains models segmented and labeled over 11 categories. For COSEG, we use the three largest and most commonly used categories, i.e. Vase, Chairs, and Tele-aliens. We follow the official splits of training and test set for all the above datasets. Due to the non-manifold nature of raw 3D data, we must ensure the shape is a manifold for the Laplacian spectral decomposition. Therefore, we follow the manifold algorithm [31] to preprocess the raw data and simplify [26] these watertight meshes to roughly the same number (2048) of vertices. Note that we have demonstrated that our network is independent of the number of vertices. The input features include coordinates (3) and Laplacian eigenvectors (12) for two branches. The 12-d Laplacian eigenvectors are the absolute Laplacian eigenvectors corresponding to the 12 lowest frequencies. For all the shapes, we scale them into a unit sphere. According to the evaluation (see Table 4), we achieve the best performance using four attention blocks and 12-d Laplacian eigenvectors.

We train the dual attention and downstream networks simultaneously. Our network is trained for 1000 epochs using the Adam solver [38] with a learning rate starting from 5e-4 and decaying every 100 epochs with a decay rate of 0.8. The trainable parameters are initialized randomly with Gaussian distribution. We implemented our network in PyTorch [51]. The backbone network of self-attention is borrowed from PCT [24]. Most linear layers are composed of MLPs with ReLU activation. Empirically, our network converges in one day with a batch size of 32.

Methods	Input Type	MN10	MN40	SN
PointNet [54]	Point	-	89.2	-
PointNet++ [56]	Point	-	91.9	-
SO-Nett [40]	Point	95.7	93.4	-
PCT [24]	Point	-	93.2	-
3DShapeNets [79]	Volume	83.5	77.0	-
VoxNet [46]	Volume	91.0	84.5	-
ACNN [5]	Mesh	-	-	93.9
SyncSpecCNN [85]	Mesh	-	-	99.7
SPH [36]	Mesh	-	68.2	-
LaplacianNet [57]	Mesh	97.4	94.2	**99.8**
Ours	Mesh	**98.6**	**95.5**	99.4

Table 1: **Comparison on shape classification** of ModelNet10(MN10), ModelNet40(MN40), and ShapeNet(SN)**.** All the alternative methods are classified into three clusters according to the input type. Note that '-' indicates the number is not reported.

4.2 Shape Classification

We compare our shape classification network with state-of-the-art methods quantitatively. We evaluate all the methods on three datasets, ShapeNet (16 categories), ModelNet40 (40 categories), and ModelNet10 (10 categories), which are all widely used benchmarks for 3D shape classification. The output of our classification network is a probability score vector over all categories. We optimize

the network by minimizing the cross-entropy loss between the ground truth one-hot vector and the probability logits. We observe that our method successfully beats all the other methods on the ModelNet Benchmark, including point-based methods [54,56,40,24], volume-based methods [46,79], and mesh-based methods [5,85,36,57]. Meanwhile, our method achievess comparable performance on the ShapeNet compared to SyncSpecCNN and LaplacianNet. The results are shown in Table 1, we report the overall accuracy across all categories. The mean overall accuracy on three large datasets is 97.9%, which outperforms the attention-based models such as PCT [24] and strong mesh-based models such as [57]. Note that our method only takes the 3-d coordinates and 12-d Laplacian eigenvectors as input, more inputs features (*e.g.* normal) could further improve the performance of our network. Please refer to our supplementary for more evaluations.

4.3 Shape Segmentation

Mesh segmentation is a critical and challenging task supporting methods for shape understanding and synthesis. Here, we evaluate our dual-attention mechanism on ShapeNet [9] and COSEG [84] datasets for part segmentation, which aims to divide a mesh into meaningful parts. Our network architecture for part segmentation is illustrated in Figure 3. In ShapeNet, we train our network in the cross-categories setup where there are 16 categories and 50 different parts in total. Compared to ShapeNet, some of the categories in COSEG contain fewer data, bringing difficulties to deep learning methods. The three large categories of COSEG are: Vase, Chair, Tele-Alines, which contain 200, 300, 400 shapes, respectively. Moreover, its ground truth labels are annotated on point clouds sampled from the meshes. Since we need the Laplacian eigenvectors on the manifold meshes with graph structure in our input, we turn the raw meshes to manifold meshes [31] and transfer the labels on the point cloud to the nearest mesh vertices in the data preparation stage. More evaluations are presented in supplementary. The input features include three parts: coordinates (3), Laplacian eigenvectors (12), and the category label (one-hot vector). Our network generates vertex-level semantic probabilities on the input meshes. The cross-entropy loss is used to supervise the output of the network according to ground truth one-hot vectors. Following previous works, we evaluate the performance of each method by the widely

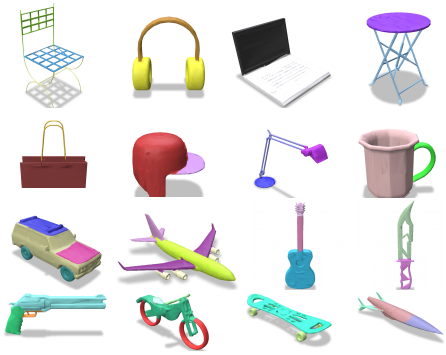


Fig. 4: **Part segmentation results.** We examples from different categories of ShapeNet. Note that the performance on Motor is lower than most other categories as in Table 2, due to its more complex topology and the larger number of mechanical parts.

	Method	Mean	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife	Lamp	Laptop	Motorbike	Mug	Pistol	Rocket	Skateboard	Table
Accuracy	Shapeboost [35]	77.2	85.8	93.1	85.9	79.5	70.1	81.4	89.0	81.2	71.1	86.1	77.2	94.9	88.2	79.2	91.0	74.5
	Guo *et al.* [27]	77.6	87.4	91.0	85.7	80.1	66.8	79.8	89.9	77.1	71.6	82.7	80.1	95.1	84.1	76.9	89.6	77.8
	ShapePFCN [34]	85.7	90.3	94.6	94.5	**90.2**	82.9	84.9	91.8	82.8	78.0	95.3	**87.0**	96.0	91.5	81.6	91.9	84.8
	LaplacianNet [57]	91.5	89.6	90.2	88.2	88.2	83.2	82.3	95.6	88.7	87.4	96.3	70.6	97.0	92.7	82.2	**94.7**	**92.6**
	Ours	**92.6**	**90.7**	**96.5**	**95.0**	89.1	**92.7**	**93.2**	**96.9**	**93.5**	**90.3**	**97.1**	85.7	**98.6**	**94.5**	**82.5**	92.5	**92.6**
IoU	FeaStNet [71]	81.5	79.3	74.2	69.9	71.7	87.5	64.2	90.0	80.1	78.7	94.7	62.4	91.8	78.3	48.1	71.6	79.6
	ACNN [5]	79.6	76.4	72.9	70.8	72.7	86.1	71.1	87.8	82.0	77.4	95.5	45.7	89.5	77.4	49.2	82.1	76.7
	VoxelCNN [85]	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.5	58.7	91.8	76.4	51.2	65.3	77.1
	Yi *et al.* [85]	**84.7**	81.6	81.7	81.9	75.2	90.2	74.9	**93.0**	86.1	**84.7**	**95.6**	**66.7**	92.7	81.6	62.1	82.9	82.1
	LaplacianNet [57]	84.3	82.9	83.4	81.7	**80.0**	75.4	71.8	91.9	81.0	80.9	92.5	59.2	93.5	86.3	**74.3**	**90.3**	86.4
	Ours	83.7	**83.2**	**92.1**	**87.2**	71.0	**91.2**	**80.6**	91.3	**86.9**	81.9	93.4	60.6	**94.6**	**87.3**	63.9	85.6	**88.4**

Table 2: **Comparison with different shape segmentation methods on ShapeNet.** Based on the output of different methods, we compare our method with the others using accuracy and/or IoU. Ours outperforms SOTA algorithms in 13/16 categories on the accuracy metric.

used accuracy and IoU (Intersection-over-Union). We compare our method with the state-of-the-art shape segmentation methods [35,27,34,57,71,5,85].

In Table 2, we report the accuracy and IoU scores over ShapeNet of all the methods, which demonstrates that our method achieves the best performance on the average scores. For the accuracy and IoU on each category, our method outperforms the prior methods on 13 and 9 categories, respectively. Table 3 presents the accuracy score on COSEG dataset. We follow the SubdivNet for spliting training and testing sets for each category. In this table, we compare with some mesh-based segmentation methods [26,49,81,74,57]. We can observe that our method beats all the alternative methods on three datasets and achieves 97.7% on the average accuracy. For the Tele-alines dataset, we outperform SubdivNet [30] by a small margin. Figure 4 shows some examples of the shape segmentation task on ShapeNet (16 categories).

Methods	Vases	Chairs	Tele-aliens	Mean
MeshCNN [26]	85.2	92.8	94.4	90.8
PD-MeshNet [49]	81.6	90.0	89.0	86.8
SubdivNet [30]	96.7	96.7	97.3	96.9
Xie *et al.* [81]	87.1	85.9	83.2	85.4
Wang *et al.* [74]	95.9	91.2	90.7	92.6
LaplacianNet [57]	94.2	92.2	93.9	93.4
Ours	**98.1**	**97.7**	**97.4**	**97.7**

Table 3: **Mesh segmentation accuracy on COSEG [76] of each method.** Our method achieves the best performance.

4.4 Ablation Studies

We perform five sets of ablation studies to demonstrate the necessity and effectiveness of our key designs. We first evaluate the dual attention and Laplacian features by checking the performance on the shape segmentation.Then, we validate the choice of the number of Laplacian eigenvectors and self-attention blocks. Finally, we demonstrate that our Laplacian mesh transformer is robust to various triangulation and different numbers of vertices. Table 4 shows all the ablation studies quantitatively on the COSEG dataset for the part segmentation.

With *v.s.* Without dual attention (DA). Our critical designs, dual attention, discriminates the importance of topology and geometry of the input shape for the specific task. Here, we aim to demonstrate the importance of using the two-stage attention for shape analysis quantitatively and qualitatively, including self-attention based topology/geometry

feature extractor and attention-based fusion (FA) module. We built and trained two ablated networks: The first one (denoted as Ours-w/o DA) only has one attention-based feature extractor and no fusing module. We directly feed the concatenation of coordinates (3) and Laplacian eigenvectors (12) to the feature extractor and use the processed feature in the following segmentation network. The second ablated version (denoted as Ours-w/o FA) adopts the original two branches to process topological and geometric inputs separately and replaces the fusing module with a simple concatenation operation. Then, the concatenated features are fed to the following segmentation network. Table 4 reports their performance on part segmentation. The quantitative results demonstrate that the dual attention mechanism brings a large improvement.

Methods	Vases	Chairs	Tele-aliens	Mean
Ours (#LEV 6)	95.3	96.2	90.0	93.8
Ours (#LEV 18)	95.9	96.6	96.7	96.4
Ours (#SA 2)	95.2	95.2	95.8	95.4
Ours (#SA 3)	97.6	97.0	96.1	96.9
Ours (#SA 5)	95.4	97.8	96.9	96.7
Ours (#V 2000)	97.9	97.5	97.3	97.5
Ours (#V 3500)	97.2	97.0	97.4	97.2
Ours (#V 5000)	97.9	97.1	97.2	97.4
Ours (Remesh)	98.0	97.7	97.1	97.6
Ours (w/o LEV)	95.5	95.7	88.0	93.1
Ours (w/o DA)	95.1	93.3	88.4	92.2
Ours (w/o FA)	95.4	96.2	89.3	93.6
Ours full-version (#SA 4, #LEV 12)	**98.1**	**97.7**	**97.4**	**97.7**

Table 4: **Ablation studies.** We evaluate the architectures without Dual Attention (DA), Fused Attention (FA), and test different numbers of Laplacian Eigenvectors (LEV), Self-Attention (SA) blocks, and different triangulations.

For our dual attention, we aim to learn the importance in a self-supervised fashion and use the important feature to determine more accurately. Figure 5 illustrates some segmentation results using our full network, where we also visualize the learned importance of the topology and geometry of different shapes, which contributes significantly to the better segmentation results of our method. The results and the visualization (Figure 5) demonstrate that our model can simultaneously learn the structural information and critical features without any supervision. Some amount of supervision could be good guidance for training the network, but the data is hard to annotate and very time-consuming. Moreover, how to balance the importance of each task in multi-task learning is very difficult to supervise.

With *v.s.* Without Laplacian Eigenvectors (LEV). In this experiment, we tested a network where all the Laplcian EigenVectors are replaced by the vertex coordinates (3) in the upper branch. In Table 4, we see that removing the Laplacian eigenvectors gives worse performance than our full model, which shows the critical role of the Laplacian eigenvectors for representing the shape topology.

Number of LEVs and Self-Attention (SA) blocks. We tested different numbers of LEVs (4, 6, 18) and SA blocks (2, 3, 5) used in our network on part segmentation. The results in Table 4 show that the combination of 4 SAs and 12 LEVs achieves the best performance. Therefore, we set the default number of LEVs and SA blocks to 12 and 4, respectively. From the results, we find that

more LEVs can result in more noises for the input features, and fewer LEVs are not sufficient to represent meaningful topological information. Besides, the network's performance reaches saturation as the number of SA increases.

Different Triangulation and Vertex Numbers. To demonstrate that our network is independent of triangulation and vertex numbers, we conducted experiments on processed meshes with different triangulation and various vertex numbers. We first subdivide [45] the meshes and simplify or sample [21,68] them to 2000, 3500, and 5000 vertices to train the models Ours-#V 2000, Ours-#V 3500, and Ours-# 5000 respectively. We use a mixture of two categories for training and the third for testing. The official splits of training and test data are applied to the three datasets. As shown in Table 4, the above models achieve similar performance as our original model.

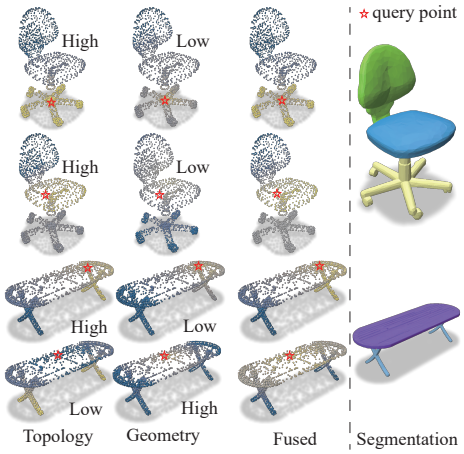


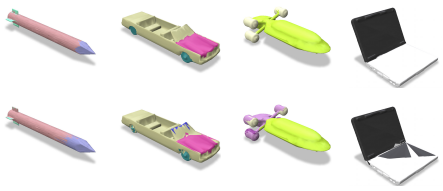
Fig. 5: **Importance visualization.** We visualize the attention maps from the self-attention (geometry & topology) and fused attention modules. For each shape, we show the three vertex-wise attention maps for two different query points in a row vertex-wise attention maps. We can observe that our dual attention is able to determine which is more important on the specific task, *e.g.* part segmentation. Blue to yellow means increasing weights.

Moreover, we re-mesh [59] the data of COSEG and simplify them to around 2048 vertices. Figure 2 shows that LEV is independent of the connectivity of triangles since it is induced from the geodesic distance and invariant under the isometric transformation, so that our method can resist the instability of different discretizations. Table 4 reports the quantitative evaluation result. The performance on the re-meshed datasets is close to the original performance of our complete network.

4.5 Limitations & Failure Cases

Our approach is limited by the geometric properties of 3D meshes. Although there are many available mesh datasets, the meshes are non-manifold and have complex topological structures, which could lead to problematic/non-robust Laplacian spectral decomposition results. For example, most meshes from ShapeNet [9] are created by artists who do not consider geometric properties. Hence, all the meshes need a pre-process to be manifold to achieve successful decomposition and have to be simplified to a specific number of vertices before feeding into networks. We also show several representative failure cases as shown in Figure 6: In Figure 6 (a), our method failed to cope with

tiny parts and recognized them as the noises of the main body of the rocket. In Figure 6 (b), for a car with no roof, our network expects a complete topology structure and recognizes the top of the window as the roof. The lack of training data on certain parts may cause a failure of our network, like in Figure 6 (c), where the engines are recognized as a tail. Figure 6 (d) shows a failure example caused by our simplification step. We use the quadratic edge collapse method [21] to simplify the meshes, which generates sparse vertices on *flat* surfaces, such as the keyboard part of the laptop here. Although only the label of one vertex is mispredicted, it still produces apparent artifacts.



(a) (b) (c) (d)

Fig. 6: **Failure Cases.** The first row is groundtruth and the second row is segmented by our method.

5 Conclusions & Future Works

In this paper, we present a novel shape analysis framework, *Laplacian Mesh Transformer*, which efficiently utilizes the shape topology and geometry information in deep feature extraction for polygonal meshes. More particularly, inspired by the recent advances of Transformer-based models in natural language processing and 2D image analysis, we propose the dual attention mechanism that achieves higher performance than prior works. In its two-stage process, we first explore the relationships between the elements of geometry features and topology features extracted by Laplacian spectral decomposition and then adopt a fusing attention module to merge the features effectively. Such a hierarchical structure to process features from fine to coarse can tackle 3D meshes with complex structure and geometry, benefiting shape analysis tasks. An avenue for future research is to apply the proposed learning framework to other potential tasks, such as shape retrieval or generative modeling. Another direction of future work is to design a transformer operator on irregular meshes, like the 2D CNN kernels in images. Furthermore, we hope to integrate the Laplacian spectral decomposition into our network architecture in an end-to-end manner, enabling the network to take raw non-manifold mesh data with arbitrary connectivities and vertex numbers as input, *e.g.* [62]. If the meshes have a large number of vertices, segmenting the meshes into patches would be a good solution to extend our scalability. For unstructured data such as point clouds, we can extend our work by constructing graphs based on proximity (as done by SyncSpecCN [85]) and using graph Laplacian, such as the raw LiDAR data for autonomous driving.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61872440).

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: ICML. pp. 40–49 (2018)
2. Ahmed, E., Saint, A., Shabayek, A.E.R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., Ottersten, B.: Deep learning advances on different 3D data representations: A survey. arXiv preprint arXiv:1808.01462 **1** (2018)
3. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1534–1543 (2016)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: International Conference on Learning Representations (2015), http://arxiv.org/abs/1409.0473
5. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.: Learning shape correspondence with anisotropic convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 3189–3197 (2016)
6. Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B.: Polygon mesh processing. CRC press (2010)
7. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End object detection with transformers. CoRR **abs/2005.12872** (2020), https://arxiv.org/abs/2005.12872
9. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
10. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1907–1915 (2017)
11. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
12. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: ECCV. pp. 628–644. Springer (2016)
13. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.:

An image is worth 16x16 words: Transformers for image recognition at scale. CoRR **abs/2010.11929** (2020), https://arxiv.org/abs/2010.11929
17. Dwivedi, V.P., Bresson, X.: A generalization of transformer networks to graphs. arXiv preprint arXiv:2012.09699 (2020)
18. Engel, N., Belagiannis, V., Dietmayer, K.: Point transformer. IEEE Access **9**, 134826–134840 (2021)
19. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)
20. Gao, L., Yang, J., Wu, T., Yuan, Y.J., Fu, H., Lai, Y.K., Zhang, H.: SDM-NET: Deep generative network for structured deformable mesh. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019) **38**(6), 243:1–243:15 (2019)
21. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. pp. 209–216 (1997)
22. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
23. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: AtlasNet: A papier-mâché approach to learning 3D surface generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
24. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)
25. Guo, M.H., Xu, T.X., Liu, J.J., Liu, Z.N., Jiang, P.T., Mu, T.J., Zhang, S.H., Martin, R.R., Cheng, M.M., Hu, S.M.: Attention mechanisms in computer vision: A survey (2021)
26. Hanocka, R., Hertz, A., Fish, N., Giryes, R., Fleishman, S., Cohen-Or, D.: Meshcnn: a network with an edge. ACM Transactions on Graphics (TOG) **38**(4), 1–12 (2019)
27. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
28. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4421–4430 (2019)
29. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3464–3473 (2019)
30. Hu, S.M., Liu, Z.N., Guo, M.H., Cai, J.X., Huang, J., Mu, T.J., Martin, R.R.: Subdivision-based mesh convolution networks. arXiv preprint arXiv:2106.02285 (2021)
31. Huang, J., Su, H., Guibas, L.: Robust watertight manifold surface generation method for shapenet models. arXiv preprint arXiv:1802.01698 (2018)
32. Huang, R., Rakotosaona, M.J., Achlioptas, P., Guibas, L.J., Ovsjanikov, M.: Operatornet: Recovering 3d shapes from difference operators. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8588–8597 (2019)
33. Ioannidou, A., Chatzilari, E., Nikolopoulos, S., Kompatsiaris, I.: Deep learning advances in computer vision with 3d data: A survey. ACM Computing Surveys (CSUR) **50**(2), 1–38 (2017)
34. Kalogerakis, E., Averkiou, M., Maji, S., Chaudhuri, S.: 3D shape segmentation with projective convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3779–3788 (2017)

35. Kalogerakis, E., Hertzmann, A., Singh, K.: Learning 3d mesh segmentation and labeling. ACM Transactions on Graphics (TOG) **29**(4), 102 (2010)
36. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3 d shape descriptors. In: Symposium on geometry processing. vol. 6, pp. 156–164 (2003)
37. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. arXiv preprint arXiv:2101.01169 (2021)
38. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
39. Li, C.L., Zaheer, M., Zhang, Y., Poczos, B., Salakhutdinov, R.: Point cloud GAN. arXiv preprint arXiv:1810.05795 (2018)
40. Li, J., Chen, B.M., Lee, G.H.: So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9397–9406 (2018)
41. Lim, D., Robinson, J., Zhao, L., Smidt, T., Sra, S., Maron, H., Jegelka, S.: Sign and basis invariant networks for spectral graph representation learning. arXiv preprint arXiv:2202.13013 (2022)
42. Lin, T., Wang, Y., Liu, X., Qiu, X.: A survey of transformers. arXiv preprint arXiv:2106.04554 (2021)
43. Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: International Conference on Learning Representations. OpenReview.net (2017), https://openreview.net/forum?id=BJC_jUqxe
44. Litany, O., Remez, T., Rodola, E., Bronstein, A., Bronstein, M.: Deep functional maps: Structured prediction for dense shape correspondence. In: Proceedings of the IEEE international conference on computer vision. pp. 5659–5667 (2017)
45. Loop, C.: Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics (1987)
46. Maturana, D., Scherer, S.: VoxNet: A 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 922–928. IEEE (2015)
47. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
48. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and mathematics III, pp. 35–57. Springer (2003)
49. Milano, F., Loquercio, A., Rosinol, A., Scaramuzza, D., Carlone, L.: Primal-dual mesh convolutional neural networks. arXiv preprint arXiv:2010.12455 (2020)
50. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
51. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
52. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks (2020)
53. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. Experimental mathematics **2**(1), 15–36 (1993)

54. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
55. Qi, C.R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L.J.: Volumetric and multi-view cnns for object classification on 3d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5648–5656 (2016)
56. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
57. Qiao, Y.L., Gao, L., Rosin, P., Lai, Y.K., Chen, X., et al.: Learning on 3d meshes with laplacian encoding and pooling. IEEE Transactions on Visualization and Computer Graphics (2020)
58. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. arXiv preprint arXiv:1906.05909 (2019)
59. Rineau, L., Yvinec, M.: A generic software design for delaunay refinement meshing. Computational Geometry **38**(1-2), 100–110 (2007)
60. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE transactions on Signal Processing **45**(11), 2673–2681 (1997)
61. Sharp, N., Crane, K.: A Laplacian for Nonmanifold Triangle Meshes. Computer Graphics Forum (SGP) **39**(5) (2020)
62. Sharp, N., Crane, K.: A laplacian for nonmanifold triangle meshes. In: Computer Graphics Forum. vol. 39, pp. 69–80. Wiley Online Library (2020)
63. Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.H., Kautz, J.: SplatNet: Sparse lattice networks for point cloud processing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2530–2539 (2018)
64. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3D shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)
65. Sun, C.Y., Zou, Q.F., Tong, X., Liu, Y.: Learning adaptive hierarchical cuboid abstractions of 3d shape collections. ACM Transactions on Graphics (TOG) **38**(6), 1–13 (2019)
66. Tan, Q., Gao, L., Lai, Y.K., Xia, S.: Variational autoencoders for deforming 3d mesh models. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5841–5850 (2018)
67. Tay, Y., Dehghani, M., Bahri, D., Metzler, D.: Efficient transformers: A survey. arXiv preprint arXiv:2009.06732 (2020)
68. Trappolini, G., Cosmo, L., Moschella, L., Marin, R., Melzi, S., Rodolà, E.: Shape registration in the time of transformers. Advances in Neural Information Processing Systems **34**, 5731–5744 (2021)
69. Tulsiani, S., Su, H., Guibas, L.J., Efros, A.A., Malik, J.: Learning shape abstractions by assembling volumetric primitives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2635–2643 (2017)
70. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
71. Verma, N., Boyer, E., Verbeek, J.: Feastnet: Feature-steered graph convolutions for 3d shape analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2598–2606 (2018)

72. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 6450–6458. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.683, https://doi.org/10.1109/CVPR.2017.683
73. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3D mesh models from single RGB images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018)
74. Wang, P., Gan, Y., Shui, P., Yu, F., Zhang, Y., Chen, S., Sun, Z.: 3d shape segmentation via shape fully convolutional networks. Computers & Graphics **76**, 182–192 (2018)
75. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
76. Wang, Y., Asafi, S., Van Kaick, O., Zhang, H., Cohen-Or, D., Chen, B.: Active co-analysis of a set of shapes. ACM Transactions on Graphics (TOG) **31**(6), 1–10 (2012)
77. Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Tomizuka, M., Keutzer, K., Vajda, P.: Visual transformers: Token-based image representation and processing for computer vision. CoRR **abs/2006.03677** (2020), https://arxiv.org/abs/2006.03677
78. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 82–90 (2016)
79. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
80. Xiao, Y.P., Lai, Y.K., Zhang, F.L., Li, C., Gao, L.: A survey on deep geometry learning: From a representation perspective. Computational Visual Media **6**(2), 113–133 (2020)
81. Xie, Z., Xu, K., Liu, L., Xiong, Y.: 3d shape segmentation and labeling via extreme learning machine. In: Computer graphics forum. vol. 33, pp. 85–95. Wiley Online Library (2014)
82. Yang, J., Mo, K., Lai, Y.K., Guibas, L.J., Gao, L.: Dsg-net: Learning disentangled structure and geometry for 3d shape generation. arXiv preprint arXiv:2008.05440 **3**, 3 (2020)
83. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems **32** (2019)
84. Yi, L., Kim, V.G., Ceylan, D., Shen, I.C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A., Guibas, L.: A scalable active framework for region annotation in 3d shape collections. ACM Transactions on Graphics (ToG) **35**(6), 1–12 (2016)
85. Yi, L., Su, H., Guo, X., Guibas, L.J.: Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2282–2290 (2017)
86. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 12498–12507 (2021)
87. Zhang, H., Goodfellow, I.J., Metaxas, D.N., Odena, A.: Self-attention generative adversarial networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Interna-

tional Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 7354–7363. PMLR (2019), http://proceedings.mlr.press/v97/zhang19d.html
88. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10076–10085 (2020)
89. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268 (2021)
90. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4490–4499 (2018)