



MTGNet: multi-label mesh quality evaluation using topology-guided graph neural network

Haoxuan Zhang^{1,2} · Haisheng Li^{1,2} · Xiaoqun Wu^{1,2} · Nan Li^{1,2}

Received: 10 March 2024 / Accepted: 23 May 2024

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

Mesh quality directly affects the accuracy and efficiency of numerical simulation. Mesh quality evaluation aims to evaluate the suitability of the mesh generated in CAE pre-processing for numerical simulation. Recent work has introduced deep neural networks for mesh quality evaluation. However, these methods treat the mesh quality evaluation task as a multi-classification problem, resulting in serious correlations among different quality categories, which makes it difficult to learn the boundaries of different categories. In this paper, we propose a topology-guided graph neural network, MTGNet, which treats the mesh quality evaluation task as a multi-label task. Specifically, we first decomposed the categories in traditional multi-classification problems and obtained three completely orthogonal mesh quality labels, namely orthogonality, smoothness and, distribution. Then, MTGNet introduces a topology-guided feature representation for structured mesh data, which can generate multiple blocks of element-based graphs through the mesh topology. In order to better fuse features in different blocks, MTGNet also introduces an attention-based block graph pooling (ABGPool) method. Experimental results on the NACA-Market dataset demonstrate MTGNet shows superior or at least comparable performance to the state-of-the-art (SOTA) approaches.

Keywords Mesh quality evaluation · Structured mesh · Multi-label classification · Graph neural network · Deep learning application

1 Introduction

Mesh generation and quality evaluation are the backbone of accurate and efficient numerical simulations [1, 2]. In the American Institute of Aeronautics and Astronautics (AIAA)'s research report, titled "CFD Vision 2030 Road Map: Progress and Perspectives" [3], mesh generation is listed as one of the six important research areas in the future. The development of Computational Aided Engineering

(CAE) has revolutionized the pre-processing process of mesh generation and provides support for various simulation tasks [4–6]. The quality of generated mesh largely determines the reliability and accuracy of numerical simulation results. The critical task of mesh quality evaluation has attracted widespread attention, especially in the field of computational fluid dynamics (CFD).

Mesh quality evaluation, despite its seemingly simple nature, is marked by a lack of consensus among researchers regarding the most appropriate and effective metrics to use [7]. This divergence in opinions and practices can be attributed primarily to the absence of a universally accepted guiding framework or standardized set of criteria for the development and implementation of mesh quality metrics. Since it is difficult to define an evaluation function that takes the entire mesh as input, mesh quality evaluation typically utilizes element-based mesh quality metrics. These metrics are vital for evaluating the suitability and efficiency of mesh, particularly in computational simulations where mesh accuracy directly impacts the overall results. Li et al. [8] provided a detailed overview of mesh quality metrics, especially focusing on those relevant to mesh. In 2D mesh, the main

✉ Haisheng Li
lihsh@th.btbu.edu.cn

Haoxuan Zhang
zhxggg613@126.com

Xiaoqun Wu
wuxiaoqun@btbu.edu.cn

Nan Li
linan@th.btbu.edu.cn

¹ School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China

² Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing 100048, China

metrics of interest to researchers are shape and size, which are important to ensure that the mesh accurately represents the physical domain [9] it is intended to simulate. At the same time, the mesh quality also affects the convergence of numerical calculations.

However, the traditional mesh quality evaluation method in CFD is characterized by a reliance on existing quality metrics that may not uniformly guarantee the generation of high-quality mesh. This limitation mainly stems from the fact that these metrics are often derived with a specific focus on a particular mesh type or simulation scenario. Consequently, when applied to diverse mesh configurations, the same metrics can yield varied and sometimes contradictory results. This illustrates that existing mesh quality metrics are highly subjective and make it difficult to evaluate the quality of a mesh cell from a comprehensive perspective.

With the rapid development of graphics processing units (GPU) and the volume of big data, the application scenarios of deep learning have gradually begun to develop in the fields of computer graphics [10, 11] and Computer-Aided Design (CAD) [12, 13]. Convolutional Neural Networks (CNNs) [14] are the representative achievement of deep learning models. The sharing of convolution kernel parameters within the hidden layer and the sparsity of inter-layer connections enable the convolutional neural network to learn grid-like topology features, such as image [15, 16], video [17, 18] and audio [19], with a small amount of calculation. This results in stable effects and no additional feature engineering requirements on the data. Chen et al. [20] introduced a novel paradigm in the realm of structured mesh quality evaluation. They proposed GridNet, a model predicated on the principles of CNNs. Concurrently, they contributed to the field by proposing the NACA-Market, a structured mesh dataset designed to facilitate research and application in this domain. Liu et al. [21] proposed Gridformer, an automatic mesh quality evaluation model based on Transformer [22]. This methodology sets the task of mesh quality evaluation as an image classification challenge. Gridformer is particularly

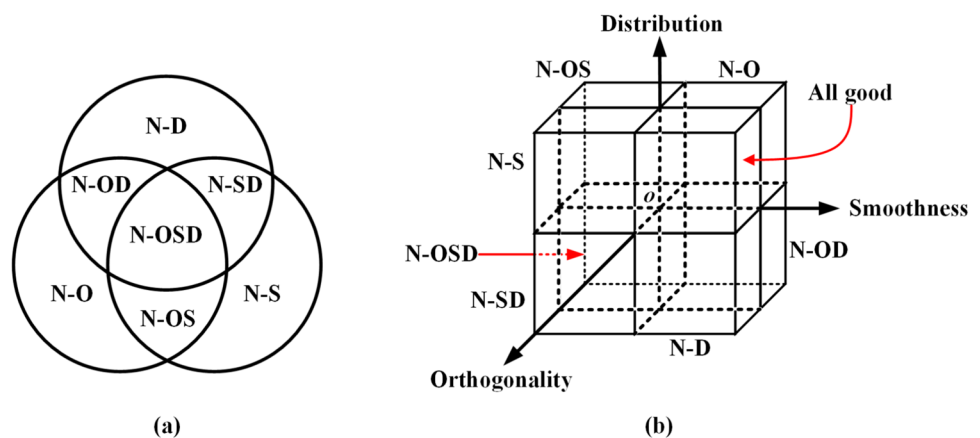
noteworthy for its selection of three pivotal features that exert a substantial impact on mesh quality.

The mesh quality evaluation networks based on deep learning proposed by existing research have performed well. However, existing research treats the mesh quality evaluation task as a multi-classification problem [20, 21, 34, 35, 44]. We experimentally found that correlations exist between each category. Taking the NACA-Market dataset constructed by Chen et al. [20] as an example, this dataset adopts three mesh evaluation metrics, namely orthogonality, smoothness, and distribution. Each metric uses two categories to form a category, so there are eight categories in total. This means that the output mesh quality evaluation result is an 8D vector. In Fig. 1a, we can see that N-OSD has a strong correlation with N-OS, N-OD, N-SD, N-S, N-D, and N-O, which seriously affects the boundaries between different categories of data. To solve this problem, we decompose the eight categories and change the mesh quality evaluation task to a multi-label problem, which means that the result is a 3D vector. Without changing the original categories, the 3D vector labels that we obtain are a completely orthogonal space (Fig. 1b), allowing the network to learn the boundaries of different categories well.

In this paper, we present a topology-guided graph neural network for multi-label mesh quality evaluation. To the best of our knowledge, our proposed method is the first to employ multi-label graph neural network for the task of mesh quality evaluation. The key contributions are as follows:

- We propose MTGNet, a multi-label mesh quality evaluation network based on graph attention mechanism. MTGNet utilizes an attention-based block graph pooling (ABGPool) to extract mesh quality features in a reasonable way, namely orthogonality, smoothness and distribution, which allows the network to better learn the boundaries among different labels.
- We design a topology-guided input representation of structured mesh. Based on the complex topology in the

Fig. 1 Schematic diagram of multi-classification and multi-label problems. Where **a** is a multi-classification Venn diagram, **b** is a multi-label 3D space representation. N-SD represents mesh data with poor smoothness and distribution, and N-O represents mesh data with poor orthogonality. In (a), if the mesh data does not belong to any category in the set, then its label is “All good”



structured mesh, the mesh data is divided into multiple blocks, and then the mesh cells are regarded as graph nodes to provide input for the graph neural network. This representation can provide more property information about the mesh shape.

- We evaluate MTGNet on the mesh benchmark dataset NACA-Market. Experimental results demonstrate the superior performance of MTGNet compared to SOTA methods. By introducing a multi-label perspective, MTGNet can provide a more reliable mesh quality evaluation result during numerical simulations.

2 Related work

According to mesh topology, mesh can be divided into unstructured mesh and structured mesh. In this paper, we focus on structured mesh because it is more efficient in numerical computation, but the difficulty of generating structured mesh leads many researchers to prefer unstructured mesh. The difficulty of structured mesh generation lies in ensuring that the generated mesh is consistent with the boundary and maintains orthogonality, which may cause singular points. Singular points negatively impact the results and stability of numerical simulations. Meanwhile, another factor hindering the development of automated structured mesh generation is the difficulty in achieving intelligent mesh quality evaluation, because traditional mesh quality evaluation often requires the involvement of professionals, making the task time-consuming and highly experience-dependent.

Traditional mesh quality metrics are often evaluated from one aspect of the structured mesh. The area ratio evaluates the quality of mesh elements by comparing their areas to a reference or ideal area. The area ratio of a structured mesh is measured as:

$$f_{area} = \max\left[R, \frac{1}{R}\right] \tag{1}$$

$$R = \frac{Area(Q_{up}) + Area(Q_{left}) + Area(Q_{down}) + Area(Q_{right})}{N_{Q_{this}} * Area(Q_{this})} \tag{2}$$

where $Area(Q)$ is the area of mesh Q . Q_{this} refers to the structured mesh being calculated, and Q_{up} , Q_{down} , Q_{left} , and Q_{right} refer to the adjacent mesh on the upper, lower, left, and right sides of the structured mesh respectively. If there is no mesh in a certain direction, its area is 0. $N_{Q_{this}}$ is the number of Q_{this} adjacent structured mesh.

Knupp [23] provided explicit formulas for partial quality measures of quadrilateral mesh and, for each measure type, listed the basic abstract properties of that type. A structured

mesh has four valences. Suppose the coordinates of the four nodes be $(x_p, y_p), p = 0, 1, 2, 3$. By calculating the Jacobian matrix of each node, we can get the following matrix:

$$B_p = \begin{bmatrix} x_{p+1} - x_p & x_{p+3} - x_p \\ y_{p+1} - y_p & y_{p+3} - y_p \end{bmatrix} \tag{3}$$

So the shape quality metric for structured mesh is:

$$f_{shape} = \frac{8}{\sum_{n=0}^3 (\lambda_{11}^k + \lambda_{22}^k) / \beta_p} \tag{4}$$

where λ_{11}^k is the square of the length of the side connecting nodes k and $k + 1$, λ_{22}^k is the square of the length of the side connecting nodes k and $k + 3$, $\beta_p = \det(B_p)$.

The skewness metric aims to evaluate mesh distortions which arise from large or small angles. Unlike shape, the skewness is insensitive to length. So the skewness for structured mesh is:

$$f_{skewness} = \frac{4}{\sum_{n=0}^3 \left(\sqrt{\lambda_{11}^k \lambda_{22}^k}\right) / \beta_p} \tag{5}$$

Xie et al. [24] introduced an approach to evaluate mesh quality, shifting the focus from traditional element-based indicators to a mapping-based method. This method represents a novel perspective in mesh quality evaluation, emphasizing the relationship between the initial (or undeformed) mesh and the deformed mesh. This approach is particularly relevant in scenarios where the mesh undergoes deformation, either due to external forces or as part of the simulation process.

The incorporation of machine learning techniques [25, 26] represents an evolution beyond traditional mesh quality metrics. Chetouani [27] introduced a paradigm-shifting 3D mesh quality metric founded on the principles of feature fusion. This metric employs a Support Vector Regression (SVR) model [28] to quantitatively evaluate the quality of 3D mesh. The SVR model is trained, integrating predefined mesh quality metrics alongside geometric attributes to yield predictive quality scores. Sprave et al. [29] employed an innovative methodology, wherein they extract low-level attributes via the neighborhood graph of the mesh. This process involves an analysis of each mesh element, focusing on the aggregation and evaluation of neighborhood quality indices to determine the overall quality of the mesh. The development of reinforcement learning techniques has also brought innovation to the field. Tong et al. [30] proposed a framework called ‘‘SRL-assisted AFM’’, combined with reinforcement learning, to introduce different mesh quality metrics as reward functions in the process of generating quadrilateral meshes. This method generates quadrilateral meshes with low aspect ratios, high Jacobian, unbalanced

seeds, low EP numbers with sharp features, conformal boundaries, and boundary layers.

With the further development of research on mesh quality evaluation based on deep learning, researchers have discovered that mesh data is a kind of natural graph data, which can be processed by Graph Neural Networks (GNNs) [31, 32]. The core idea of GNNs is to aggregate the neighbor information of the node through the message-passing mechanism [33] and update the feature representation of the node. This process typically involves multiple iterations to capture more distant information in the graph. Eventually, the feature representation of each node contains information about its neighbors and further nodes. Based on this idea, Wang et al. [34] propose GMeshNet, a graph neural network to evaluate the quality of structured mesh, which converts the mesh quality evaluation task into a graph classification task. They also design a sparse-implemented algorithm to transform the structured mesh data into graph data. GMeshNet achieves SOTA performance on the NACA-Market dataset (10,240 meshes). In our previous work [35], we propose a structured mesh quality evaluation neural network based on dynamic graph attention, MQENet. We also design two improved mesh preprocessing algorithms, the point-based graph and the element-based graph, to convert mesh to graph more effectively. MQENet achieves SOTA results on ten subsets of the NACA-Market dataset (1024 meshes).

Despite these advances, a limitation still exists in the machine learning techniques employed. These methods often exhibit constrained generalization capacity, which becomes particularly evident when encountering data distributions that depart from those present in the training dataset. Existing mesh quality evaluation methods based on deep learning

are usually regarded as a multi-classification problem, making it challenging to discern and accurately evaluate the quality of the mesh.

3 Method

3.1 Overview

To extract mesh quality features from structured mesh, we design a topology-guided graph neural network for multi-label mesh quality evaluation. The architecture of MTGNet is shown in Fig. 2. For input structured mesh data, we first process the mesh using a topology-guided input representation that results in a 6D feature based on mesh elements. The feature is then divided into blocks based on topology, and each block is represented as a graph. GATv2 [36] and SAGPool [37] are utilized for graph convolution layer and graph pooling layer respectively. We propose an Attention-based Block Graph Pooling (ABGPool) layer to fuse features in different blocks. Finally, a Multi-Layer Perceptron (MLP) is used to classify the mesh quality. Unlike other studies, the output of MTGNet is a 3D label vector instead of an eight-category vector.

The final result consists of three mesh quality labels: orthogonality, smoothness and, distribution. As defined by Chen et al. [20], the orthogonality metric means that the mesh element should be as orthogonal as possible. The smoothness metric means that the change of the mesh element in the model should be a smooth transition rather than a sudden transition. The distribution metric means that the need to use dense mesh elements in critical areas (e.g.,

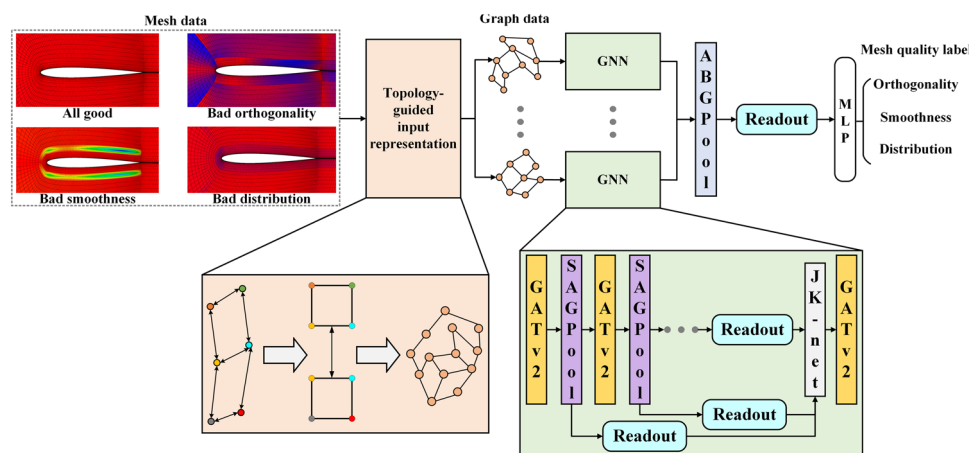


Fig. 2 The architecture of MTGNet. MTGNet first generates element-based graphs from the structured mesh data and divides the mesh data into different blocks through topology-guided input representation. Then, the graph neural network is used to perceive and extract different features in the graph. Finally, MTGNet fuses features in dif-

ferent blocks through ABGPool to obtain a quality label for the input structured mesh. For input mesh data, red mesh elements represent high quality and blue mesh elements represent low quality. Different mesh quality metrics detect low-quality mesh elements in different areas. (Color figure Online)

where there are large variations in flow field parameters or geometric curvature).

Unlike multi-classification networks, the output of multi-label networks is not a one-hot vector, and each index on the multi-label vector is evaluated independently. At the same time, since the final result is changed to a multi-label vector, this means that the network structure and the way of extracting features are very different from that of a multi-classification network.

3.2 Topology-guided input representation

Structured mesh usually contain a large number of nodes and cells, so the generated graph data also have a large number of nodes. This fact causes the neighborhood explosion problem [38] in the graph neural network during the training process. At the same time, studies by Wang et al. [34] and our previous work [35] have demonstrated that models trained using node-based structured mesh representation perform poorly (this is why we did not use node-based mesh representation in Sect. 4 experiment). To this end, we propose a topology-guided input representation method for structured mesh.

According to the theory of Xiao et al. [39], the topology of the structured mesh are composed of singular points and streamlines formed by the cross-field, as show in Fig. 3. Among them, singular points and streamlines are defined as follows:

- *Singular point.* Given a vector-field $v_f(x, y)$ over the mesh domain Ω and a point $p \in \Omega$, p is a singular point when it satisfies $v_f(p) = 0$.
- *Streamline.* Given a cross-field $v_c(p)(c = 0, 1, 2, 3)$ over the mesh domain Ω , suppose $\psi(s) \in \Omega$ is a parametric curve. $\psi(s)$ is a streamline of $v_c(p)$ when it satisfies $\forall s \in [0, 1], \exists \partial\psi(s) \times v_i(\psi(s)) = 0$.

We extract singular points P and streamlines Ψ from structured mesh domain Ω , and uses singular points as vertices and streamlines as boundaries to divide the mesh data into multiple

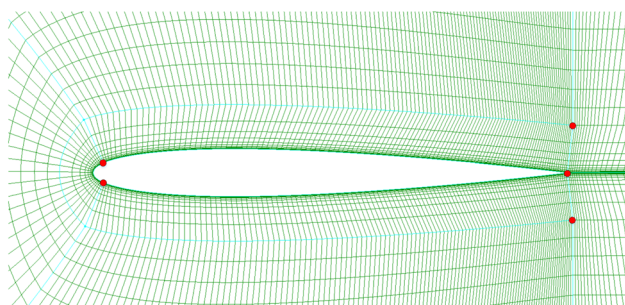


Fig. 3 A 2D airfoil structured mesh. The red points are singular points and the blue lines are streamlines

blocks. Our method can accelerate the processing speed of graph neural network on mesh quality evaluation task.

GNNs are deep learning models used for processing graph data. This paper takes structured mesh data in different blocks as input, so how to convert mesh data into graph data is a challenging task.

A graph is pair of $G = (V, E)$, where $V = \{v_i || i \in N_v\}$ is the set of vertices from mesh nodes, N_v is the number of the vertices and $E = \{e_{ij} || e_{ij} = (v_i, v_j), (v_i, v_j) \in V^2\}$ is the set of edges from connections between mesh nodes. For an undirected graph, e_{ij} is identical to e_{ji} . A graph corresponding to a mesh consists of its own nodes and elements.

There are two existing algorithms for converting structured mesh data into graph data, namely, graph representation based on mesh nodes and graph representation based on mesh elements. In graph representation based on mesh nodes, the individual mesh nodes (the points where mesh lines intersect) are treated as nodes of the graph. This approach directly maps each mesh node to a graph node. The edges of the graph correspond to the edges of the mesh. In other words, if two mesh nodes are connected by an edge in the mesh, their corresponding graph nodes are also connected by an edge in the graph representation. The graph representation based on mesh elements treats mesh cells (faces) as nodes of the graph, and the edges of the graph are based on the connection strength between the two cells.

To further reduce the nodes in the graph generated from structured mesh data, we use the graph representation based on mesh elements. First, the feature matrix $X \in \mathbb{R}^{N \times f}$ and the adjacency matrix of mesh nodes $A_q \in \mathbb{R}^{4N \times 4N}$ based on node are obtained from the raw mesh file, where N is the number of mesh elements and f is the number of feature. Through the coordinates of the mesh nodes, we can calculate which mesh nodes are shared, thereby obtaining the adjacency matrix of the mesh nodes. In order to better obtain the adjacency relationship of each node on the mesh, we calculate the nodes on each mesh element independently. Then, we suppose there is an element management matrix M , where $M = [m_{ij}] \in \mathbb{R}^{4N \times N}$, and $m_{ij} = 1$ if node i in element j , otherwise $m_{ij} = 0$. Finally, we can obtain the strength matrix between two mesh elements:

$$S = M^T A_q M \tag{6}$$

If two structured mesh share one edge, the strength is 6. So the adjacency matrix of mesh elements A is

$$A_{ij} = \begin{cases} 1, & \text{if } S_{ij} = 6 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

3.3 Dynamic attention graph convolution layer

We utilize GATv2 as the graph convolution layer in MTG-Net. The advantage of GATv2 is its ability to assign dynamic

attention scores to distinct nodes, in contrast to GAT’s static attention mechanism [40], which assigns uniform attention scores across all nodes. This dynamic approach of GATv2 is suitable for situations where discerning between nodes based on their individual attributes. In the task of structured mesh quality evaluation, this capability becomes vital.

GAT uses the score function $\alpha : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ to score every edge (n_i, n_j) , which calculates the importance of the features of the neighbor n_j to the graph node n_i :

$$\alpha(n_i, n_j) = \sigma(b^T \cdot \text{cat}[wn_i, wn_j]) \tag{8}$$

where $b \in \mathbb{R}^{2d}$, $w \in \mathbb{R}^{d \times d}$ are learnable parameters through training, $\text{cat}[\cdot, \cdot]$ denotes vector concatenation, d is the number of features in each graph node and σ represents an activation function.

GAT’s static attention mechanism presents a notable limitation, particularly evident in its inability to differentiate effectively between the qualities of various structured mesh. This uniformity in attention allocation potentially hinders GAT’s capacity to accurately represent and fit the nuances that exist in the training data.

In order to improve the performance of the graph convolution layer, we can move parameter b^T out of the non-linear result before performing the operation:

$$\alpha(n_i, n_j) = b^T \sigma(w \cdot \text{cat}[n_i, n_j]) \tag{9}$$

For any two mesh elements in the graph, we can get the dynamic attention score α , as shown in Fig. 4. Meanwhile, Layer Normalization (LayerNorm) [41] and the activation function are performed. Layer Normalization plays a vital role in stabilizing the training process by normalizing the input across each layer of the network, effectively reducing internal covariate shift and leading to more efficient training dynamics. The activation function, on the other hand, introduces a non-linear aspect to the network, essential for perceiving the complex, non-linear relationships in mesh data. The feature matrix $X' \in \mathbb{R}^{N \times n}$ input to the PSAP layer is:

$$X' = \text{Dynamic}(\sigma(\text{LayerNorm}(X)), A) \tag{10}$$

where $A \in \mathbb{R}^{N \times N}$ is the element-based adjacency matrix from structured mesh. N is the number of the graph nodes, m is the number of input features and n is the number of output features.

The dynamic attention mechanism of GATv2 is not just an incremental improvement but a critical enhancement that addresses a fundamental shortcoming of its predecessor, thereby enabling more nuanced and accurate modeling in applications such as structured mesh quality evaluation.

3.4 Attention-based block graph pooling layer

The application of graph pooling techniques to mesh quality evaluation tasks faces with significant challenges, primarily due to the computational complexity associated with these techniques. A key aspect of this complexity arises from the necessity to compute a distribution matrix for node clustering within the graph. The computational and spatial complexity of this distribution matrix calculation is directly proportional to the number of nodes and edges in the graph [36]. Given that mesh data often comprise a substantial quantity of nodes and elements, this complexity makes many graph pooling methods impractical for tasks related to mesh quality evaluation.

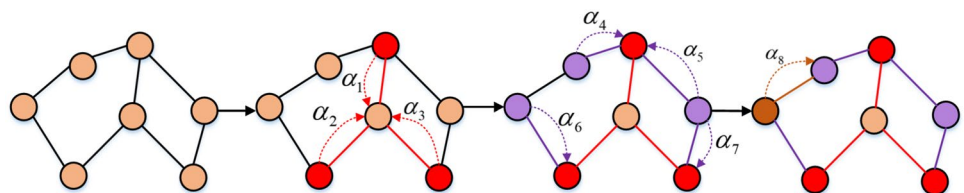
Self-Attention Pooling (SAGPool) is an implementation of hierarchical pooling. Its idea is to adaptively learn the importance of nodes from the graph through graph convolution, and then use the TopK mechanism [42] to discard nodes. The advantage of SAGPool is that it can simultaneously consider node features and graph topology, and also has reasonable complexity and end-to-end representation learning. SAGPool keeps the number of parameters consistent during training and does not need to consider the size of the input graph. This is very suitable for structured mesh data with different mesh numbers. However, SAGPool was proposed to solve the pooling problem of single graph data, and there is no solution to fuse multiple graphs. Thus, we propose ABGPool, a pooling method based on SAGPool, to pool the graph data.

SAGPool adaptively learns the importance of nodes from the graph through graph convolution, and then uses an attention-based fusion mask to fuse features from each block. Specifically, aggregation operations are used to assign an importance score to each node.

$$Z_i = \sigma \left(D_i^{-\frac{1}{2}} A_i D_i^{-\frac{1}{2}} X'_i W_{att} \right) \tag{11}$$

where $D_i \in \mathbb{R}^{N \times N}$ is the degree matrix of A_i in the i th block and $W_{att} \in \mathbb{R}^{n \times 1}$ represents the weight parameter.

Fig. 4 The calculation procedure of attention scores in graph data



As shown in Fig. 5, ABGPool uses an attention-based feature fusion method to combine the resulting mesh quality features in different blocks, thereby enhancing the feature representation ability of the pooled graph. This method is the design of a learnable fusion weight matrix, denoted as W . This matrix is not static; rather, it is optimized through the learning process of the network. The network is trained to discern and assign appropriate weights to the features by minimizing the feature fusion loss, represented as L_{fus} . This loss function serves as a guide, directing the network towards a more effective combination of features. Once the optimal weights are determined, the features in each block are combined using a weighted averaging method. By allowing the network to learn and optimize the weights for feature fusion, we ensure that the final feature representation is suited to the characteristics of the mesh data. The following equations show this feature fusion method:

$$Z = \sum_{i=1}^n W_i \otimes Z_i, \quad \sum_{i=1}^n W_i = 1 \tag{12}$$

The pooling operation can be performed according to the importance score and the topology of the graph. Based on the score calculated by Eq. 12, only $[kN]$ nodes are reserved:

$$idx = \text{top-rank}(Z, [\eta N]), \quad Z_{\text{mask}} = Z_{idx} \tag{13}$$

where $\eta \in (0, 1]$ is the pooling ratio.

Repeatedly stacking SAGPool, as shown in Fig. 1, can perform graph pooling, and finally use a PSAPool to fuse features from each block. The final feature matrix and adjacency matrix are obtained by

$$X_f = X'_{[idx, :]} \odot Z_{\text{mask}}, \quad A_f = A_{[idx, idx]} \tag{14}$$

ABGPool can effectively extract and fuse features in each block and avoid the neighbor explosion problem caused by the massive nodes in mesh data. This is crucial for tasks that

require high computational efficiency and accuracy, such as mesh quality evaluation.

3.5 Graph readout operation

In the architecture of MTGNet, an approach is employed to extract and interpret graph data, addressing the challenges posed by the variability of node positions within the graph. This process involves enhancing the completeness of the graph data through a combination of global readout operations (Fig. 1).

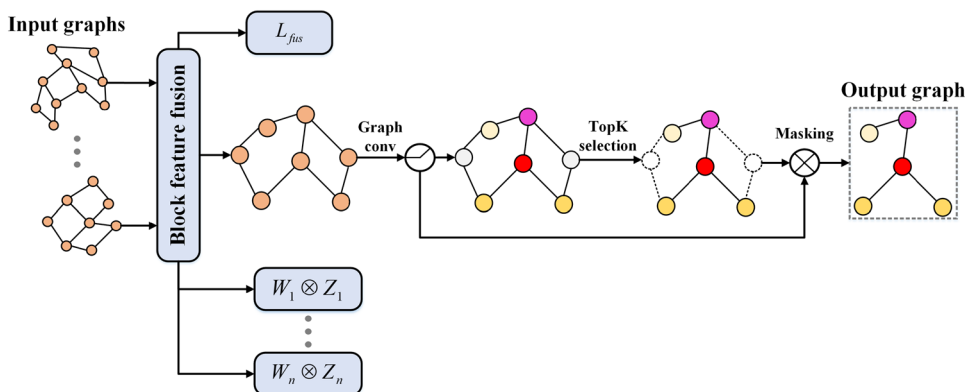
Specifically, the graph data is enriched by concatenating two distinct global readout operations: the global average readout and the global maximum readout. The global average readout operation computes the mean of features across all nodes, providing a generalized representation of the graph’s overall characteristics. In contrast, the global maximum readout focuses on capturing the most prominent features across the nodes, thereby highlighting the most significant aspects of the graph’s structure.

Given the complexity of accurately representing each node due to their differing positions in the graph, MTGNet introduces the Jumping Knowledge Network (JK-net) [43] into its framework. JK-net is good at obtaining precise representations at various levels, enabling the network to effectively capture and integrate information from different neighborhood ranges for each node. This leads to a more structure-aware representation of the graph, taking into account the diverse local contexts of individual nodes.

4 Experiments

In this section, we evaluate MTGNet on the mesh benchmark dataset NACA-Market. The datasets, implementation details and experimental results are introduced.

Fig. 5 The procedure of attention-based block graph pooling layer



4.1 Datasets

The current landscape of benchmark datasets for structured mesh evaluation tasks is notably limited, with only a few datasets available, such as NACA-Market and AirfoilSet [44]. Among these, NACA-Market stands out as a publicly accessible dataset, making it a valuable resource for research and development in mesh quality evaluation. This dataset focuses on the NACA0012 airfoil and provides the basis for mesh quality evaluation networks. NACA-Market has a total of 10,240 2D structured mesh divided into eight categories. However, in Sect. 3.2 of this paper, we changed the annotation method of this dataset to multi-label, resulting in three labels. This dataset has a total of 10 sizes of airfoil structured mesh, with each size including 1024 structured meshes. In the experiments, we separately use the entire dataset of NACA-Market and its subsets to train MTGNet and evaluate its performance.

4.2 Implementation details

During the training process, we set the division ratio of the training data, validation data and test data to 80%, 10%, 10%, and shuffle the data to ensure the distribution. We set the batch size to 32 and the pooling ratio for all pooling layers to 0.7. The number of input features, output features, network layers and hidden layer units to 6, 3, 8 and 8, respectively. We select AMS-Grad [45] with an initial learning rate of $1e-2$ as the optimizer, and dynamically decrease the learning rate according to the training situation. In order to satisfy the final multi-label output, the final activation function of the MLP layer is Sigmoid. Furthermore, we use binary cross-entropy loss as the loss function and add L2 regularization with $1e-4$ weight decay. The experiments are carried out on an NVIDIA RTX A6000-48G. Part of the visualization of the mesh data are completed in the MeshLink platform [46].

4.3 Evaluation metric

Since we treat the mesh quality evaluation task as a multi-label problem, accuracy can no longer be used to evaluate network performance. In this paper, we choose F1-score as the evaluation metric of neural network. The calculation formula of macro-F1-score is as follows:

$$F1 = \frac{2TP}{2TP + FN + FP} \quad (15)$$

where TP represents the number of true positive examples, FP represents the number of false positive examples, and FN represents the number of false negative examples.

In addition, the F1-score is also divided into micro-F1 and macro-F1. Micro-F1 first calculates the TP , FP , and FN of all categories, then sums up to obtain the overall TP_{all} , FP_{all} , and FN_{all} . By calculating the overall precision and recall, we can finally calculate the micro-F1 value. The characteristic of this calculation method is that the calculated F1 value is susceptible to the influence of a class with a large number of samples. Macro-F1 first calculates the F1 of all classes, then directly calculates the arithmetic mean of all F1 to obtain the macro-F1 value. This method treats all classes equally, regardless of the importance of different classes. Therefore, macro-F1 is susceptible to sample classes corresponding to high Precision or high Recall. Based on the characteristics of micro-F1 and macro-F1, we found that the macro-F1-score is more suitable for multi-label mesh quality evaluation tasks.

To establish the superiority of MTGNet over existing solutions, comparative experiments were conducted against GMeshNet [34] and GridNet [20], which are the SOTA methods on the whole NACA-Market datasets. The performance evaluation focused on two key metrics: recall and F1-score. As shown in Table 1, the comparison reveals specific areas where MTGNet outperforms GMeshNet and GridNet on the whole NACA-Market dataset (10,240 meshes). In terms of recall for orthogonality and smoothness metrics, MTGNet achieves 99.94% and 98.32% respectively, better than the recall rates of GMeshNet and GridNet. For the distribution metric, MTGNet is slightly lower than GridNet. We believe that compared to GNNs, the receptive field of CNNs is smaller, which allows GridNet to better capture the local features of distribution in structured mesh. The F1-score of MTGNet reaches an impressive 99.47%, significantly outperforming GMeshNet and GridNet. More importantly, MTGNet divides the structured mesh data to reduce the depth and width of the neural network, thereby reducing the cost of the network.

Table 1 Performance comparison of MTGNet and other methods on the whole NACA-Market dataset.

Mesh property	Network		
	MTGNet(%)	GMeshNet(%)	GridNet(%)
Orthogonality	99.94	99.81	96.95
Smoothness	98.32	88.16	93.46
Distribution	99.73	99.21	99.92
F1	99.47	90.79	–

The significance of bold in the table indicates that the value is the maximum value in its row.

Due to the difficulty of structured mesh generation, it is challenging to form relevant structured mesh datasets. Therefore, it is necessary to test the ability of neural networks on small mesh datasets to provide solutions for structured mesh quality evaluation of more models in the future. In this paper, we test the generalization ability of different models on small datasets using the subsets of the NACA-Market dataset (1024 meshes). We choose MQENet [35] and GMeshNet [34], two SOTA methods for small mesh datasets, as comparison methods. We first train several networks separately on small structured mesh datasets. As shown in Table 2, the comparison reveals specific areas where MTGNet outperforms MQENet and GMeshNet in the subsets of the NACA-Market dataset. In terms of recall for all three metrics, MTGNet achieves 88.06%, 89.94%, and 86.24% respectively, better than the recall of MQENet and GMeshNet. The F1-score of MTGNet, standing at 88.93%, significantly outperforms MQENet and GMeshNet. We can see that MTGNet can achieve satisfactory F1-score and recall when the size of the dataset changes, demonstrating MTGNet's ability to adapt and generalize to datasets of different sizes.

It is worth noting that MQENet, GMeshNet, and GridNet all evaluate the input mesh directly. This is the reason why MTGNet is superior to these SOTA methods. After topology-guided input representation, our network can capture those blocks that contribute the most to the mesh quality. This makes it easier for the neural network to focus on the effective features in the graph. Finally, through the feature fusion module in ABGPool, most of the effective features can be utilized as the basis for the final mesh quality evaluation, thereby obtaining more reasonable mesh quality evaluation results.

In order to prove that our proposed MTGNet has improved efficiency, we measured the model parameters of different neural networks after training on the whole NACA-Market, as shown in Fig. 6. We selected ResNet [47], Efficientnet [48], VIT [49], T2T-VIT [50], GridNet, Gridformer [21], and GMeshNet for comparison. It can be seen that MTGNet has the smallest model parameters, which means that the time

required for structured mesh quality evaluation is reduced. This is because MTGNet adopts a topology-guided input representation that decomposes structured mesh data into multiple blocks. For small structured mesh data, we only need to design a small graph neural network to effectively extract the mesh quality features. Other methods directly input the entire structured mesh data into the neural network, making it difficult for small neural networks to extract deep-level mesh quality features.

We also illustrate the superiority of MTGNet by comparing it with traditional metrics. In our research, we pay special attention to structured mesh data similar to that described in Fig. 7, which belongs to the labels of bad distribution, good orthogonality and smoothness. However, traditional metrics incorrectly classified this mesh as "All good". This misclassification highlights the limitations of traditional mesh quality metrics, which can overlook significant deficiencies when other metrics perform satisfactorily. In contrast, our proposed MTGNet demonstrates its ability to accurately detect such defects by leveraging the topology-guided input representation method, GATv2, and ABGPool, which can effectively identify hidden patterns in structured meshes that traditional evaluation methods cannot identify.

4.4 Ablation studies

In this section, to illustrate the effectiveness of MTGNet, this paper performs ablation studies from two aspects: hyper-parameters and network structure. Since training the network on the whole dataset is time-consuming, we perform ablation studies on a subset (1024 meshes) of the NACA-Market dataset.

4.4.1 Analysis of hyper-parameters

In this part, we analyze the impact of different hyper-parameters on MTGNet. Choosing appropriate hyper-parameters can improve the efficiency and accuracy of neural networks. We discuss two factors, network depth and pooling ratio. The experimental results are shown in Table 3.

The pooling ratio in the graph pooling layer dictates the proportion of nodes to be retained in each layer during the pooling process. A higher pooling ratio, such as 0.9, implies that a larger proportion of nodes are kept during the pooling process. The observations from the network performance at this ratio indicate that retaining more features tends to yield better results. This is likely because more information about the mesh is preserved, allowing the network to make more informed and accurate predictions. Conversely, a lower pooling ratio, like 0.3, means that a smaller fraction of nodes is retained after each pooling operation. The poorer performance at this ratio suggests that discarding a significant number of nodes can

Table 2 Performance comparison of MTGNet and other methods on the subsets of the NACA-Market dataset.

Mesh property	Network		
	MTGNet (%)	MQENet (%)	GMeshNet (%)
Orthogonality	88.06	82.96	80.73
Smoothness	89.94	81.64	78.57
Distribution	86.24	82.48	82.69
F1	88.93	82.36	84.90

The significance of bold in the table indicates that the value is the maximum value in its row.

Fig. 6 Parameters of different neural networks on the NACA-Market

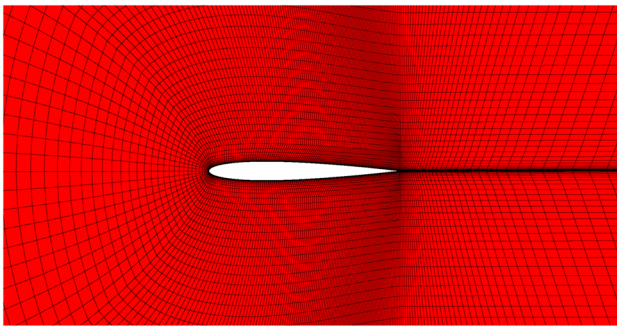
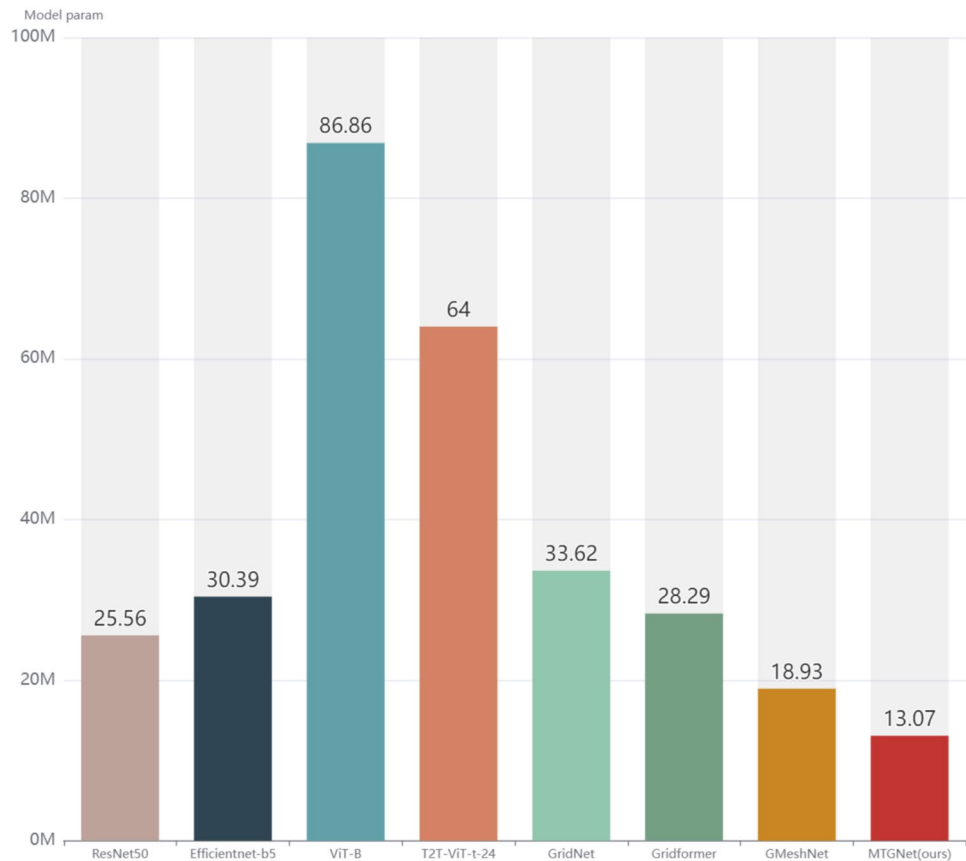


Fig. 7 Visualization of classification result on NACA-Market with traditional metrics. This mesh is classified as “all good” according to traditional metrics and actually the distribution is bad

Table 3 F1 for different hyper-parameters

Pooling ratio	Network depth		
	4 (%)	8 (%)	16 (%)
0.3	82.89	83.85	83.03
0.5	84.18	86.56	87.29
0.7	86.01	88.93	88.18
0.9	86.98	89.50	89.63

lead to a loss of critical information, which adversely affects the accuracy of the network. While a lower pooling ratio contributes to reducing the size and computational complexity of the neural network, it comes at the cost of diminished accuracy due to insufficient feature representation. However, we find that when the pooling ratio is 0.7, MTGNet can achieve performance that is quite close to that with a pooling ratio of 0.9. This means we can sacrifice a small amount of performance in exchange for improving the efficiency of the network.

The other important hyper-parameter is the network depth. We can see that when the pooling ratio is 0.5 and 0.9, the network performs better as the network depth increases. This means that deeper networks can better learn the boundaries between different categories. However, when the pooling ratio is 0.3 and 0.7, a too-deep network degrade the performance of the network. We believe that small pooling ratios coupled with deep network layers cause some of the critical quality features in structured mesh data to be discarded (because the size of the final graph is too small). And when the pooling ratio is more appropriate, too deep network layers make the model over-fit the distribution on the training set, leading to a decrease in accuracy on the validation and test

Table 4 F1-score for different networks

Pooling method	Convolution method	F1 (%)
TopK+block feature fusion	GCN	74.48
	GraphConv	78.81
	GAT	80.42
	GATv2	83.81
ABGPool(Ours)	GCN	75.12
	GraphConv	79.56
	GAT	83.79
	GATv2	88.93

sets. When the number of network layers is 8, it not only ensures the performance of the network, but also improves the efficiency of the network.

4.4.2 Analysis of network structure

In addition to the influence of hyper-parameters on the MTGNet, the network selection of the network structure is also important. The core parts of GNNs are composed of graph convolution layers and graph pooling layers. So here we choose some classic graph convolution methods and graph pooling methods to compare with the methods we use (GATv2 and SAGPool), namely GCN [51], GraphConv [52], GAT [40], and TopK [42]. Experimental results of different networks are shown in Table 4. From the results, we can see that our network performs better than other networks on NACA-Market datasets.

The use of GATv2 in the task of structured mesh quality evaluation provides a significant enhancement over other graph convolutional layers, particularly due to its dynamic attention mechanism. This feature of GATv2 plays a crucial role in enhancing the representation of hidden features within structured mesh and enables more effective classification of mesh with different quality labels. At the same time, we can see that the static attention mechanism performs better than the network that does not use any attention mechanism.

In terms of pooling layer, TopK samples a subset of basic nodes by manipulating a trainable projection vector. ABGPool further applies self-attention and graph convolution to improve TopK. From the experimental results, we can see that the network using ABGPool performs better. This is because ABGPool introduces a self-attention mechanism, which allows ABGPool to more effectively filter out graph nodes that meet the specified tasks. This shows that, whether in the graph convolution layer or the graph pooling layer, deploying the attention mechanism is crucial for the mesh quality evaluation task.

5 Conclusions

In this paper, we introduce MTGNet, a novel topology-guided graph neural network designed for multi-label mesh quality evaluation. MTGNet distinctly contrasts with other methodologies by conceptualizing mesh quality evaluation as a multi-label problem, thus allowing for a more objective and comprehensive analysis of mesh quality. MTGNet, which includes innovative elements like the topology-guided input representation and attention-based block graph pooling, demonstrates superior performance in evaluating structured mesh quality. The experiments conducted on the NACA-Market dataset show the effectiveness of MTGNet, with its ability to outperform existing SOTA methods in terms of accuracy and computational efficiency.

In the future, our proposed neural network can be further explored and developed for more computer graphics and computer-aided design related tasks. In addition, we will conduct an in-depth study of the application of feature representation methods and GNNs in the quality evaluation tasks of 3D structured mesh. For 2D unstructured mesh, we can also convert it into graph data and then use MTGNet for mesh quality evaluation. However, there are no existing mesh quality benchmark datasets for unstructured mesh. Since MTGNet is a data-driven and supervised model, it is difficult to perform the unstructured mesh quality evaluation task without datasets. In the future, we will consider constructing related datasets for unstructured mesh. We believe that MTGNet will also have excellent performance on unstructured mesh.

Funding This work is supported by Beijing Natural Science Foundation (No. L233026), and National Natural Science Foundation of China (Nos. 62277001, 62272014).

Data availability The datasets during the current study are available in <https://github.com/chenxinhai1234/NACA-Market>.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. George PL (1996) Automatic mesh generation and finite element computation. *Handb Numer Anal* 4:69–190
2. Katz A, Sankaran V (2011) Mesh quality effects on the accuracy of CFD solutions on unstructured meshes. *J Comput Phys* 230(20):7670–7686
3. Cary AW, Chawner J, Duque EP et al (2021) CFD vision 2030 road map: progress and perspectives. In: *AIAA Aviation 2021 Forum*, p 2726

4. Alauzet F, Loseille A (2016) A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Comput-Aid Des* 72:13–39
5. Gibson RF (2010) A review of recent research on mechanics of multifunctional composite materials and structures. *Compos Struct* 92(12):2793–2810
6. Um ES, Kim SS, Fu H (2017) A tetrahedral mesh generation approach for 3D marine controlled-source electromagnetic modeling. *Comput Geosci* 100:1–9
7. Knupp PM (2001) Algebraic mesh quality metrics. *SIAM J Sci Comput* 23(1):193–218
8. Li H (2012) Finite element mesh generation and decision criteria of mesh quality. *China Mech Eng* 23(3):368
9. Eriksson LE (1982) Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation. *AIAA J* 20(10):1313–1320
10. Zheng Y, Zeng G, Li H et al (2022) Colorful 3D reconstruction at high resolution using multi-view representation. *J Vis Commun Image Represent* 85:103486
11. Lei N, Li Z, Xu Z et al (2023) What's the situation with intelligent mesh generation: a survey and perspectives. *IEEE Trans Vis Comput Graph* 01:1–20
12. Li H, Zheng Y, Cao J et al (2019) Multi-view-based siamese convolutional neural network for 3D object retrieval. *Comput Electr Eng* 78:11–21
13. Zhang H, Li H, Wang P et al (2024) Surface structured mesh generation system based on symmetry preserving parameterization. *Int J Wavelets Multiresolut Inf Process* 2350065
14. Li Z, Liu F, Yang W et al (2021) A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Trans Neural Netw Learn Syst* 33:6999–7019
15. Zhang M, Li W, Du Q (2018) Diverse region-based CNN for hyperspectral image classification. *IEEE Trans Image Process* 27(6):2623–2634
16. Tian C, Xu Y, Li Z et al (2020) Attention-guided CNN for image denoising. *Neural Netw* 124:117–129
17. Xu K, Wen L, Li G et al (2019) Spatiotemporal CNN for video object segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 1379–1388
18. Davy A, Ehret T, Morel JM et al (2019) A non-local CNN for video denoising. In: *2019 IEEE international conference on image processing (ICIP)*. IEEE, pp 2409–2413
19. Schmid F, Koutini K, Widmer G (2023) Efficient large-scale audio tagging via transformer-to-CNN knowledge distillation. In: *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pp 1–5
20. Chen X, Liu J, Pang Y et al (2020) Developing a new mesh quality evaluation method based on convolutional neural network. *Eng Appl Comput Fluid Mech* 14(1):391–400
21. Liu Z, Liu H, Chen Y et al (2023) Evaluating airfoil mesh quality with transformer. *Aerospace* 10(2):110
22. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. *Adv Neural Inf Process Syst* 30:5999–6009
23. Knupp PM (2003) Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elem Anal Des* 39(3):217–241
24. Xie C, Jia S, Li Y et al (2020) Mapping based quality metrics for mesh deformation algorithms using radial basis functions. *Appl Sci* 11(1):59
25. Karniadakis GE, Kevrekidis IG, Lu L et al (2021) Physics-informed machine learning. *Nat Rev Phys* 3(6):422–440
26. Fidkowski KJ, Chen G (2021) Metric-based, goal-oriented mesh adaptation using machine learning. *J Comput Phys* 426:109957
27. Chetouani A (2017) A 3D mesh quality metric based on features fusion. *Electron Imaging* 29:4–8
28. Drucker H, Burges C J, Kaufman L et al (1996) Support vector regression machines. *Adv Neural Inf Process Syst* 9
29. Sprave J, Drescher C (2021) Evaluating the quality of finite element meshes with machine learning. arXiv preprint [arXiv:2107.10507](https://arxiv.org/abs/2107.10507)
30. Tong H, Qian K, Halilaj E et al (2023) SRL-assisted AFM: generating planar unstructured quadrilateral meshes with supervised and reinforcement learning-assisted advancing front method. *J Comput Sci* 72:102109
31. Scarselli F, Gori M, Tsoi AC et al (2008) The graph neural network model. *IEEE Trans Neural Netw* 20(1):61–80
32. Wu Z, Pan S, Chen F et al (2020) A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 32(1):4–24
33. Gilmer J, Schoenholz SS, Riley PF et al (2017) Neural message passing for quantum chemistry. In: *International conference on machine learning*. PMLR, pp 1263–1272
34. Wang Z, Chen X, Li T et al (2022) Evaluating mesh quality with graph neural networks. *Eng Comput* 38(5):4663–4673
35. Zhang H, Li H, Li N et al (2023) MQENet: A mesh quality evaluation neural network based on dynamic graph attention. arXiv preprint [arXiv:2309.01067](https://arxiv.org/abs/2309.01067)
36. Brody S, Alon U, Yahav E (2021) How attentive are graph attention networks?. arXiv preprint [arXiv:2105.14491](https://arxiv.org/abs/2105.14491)
37. Lee J, Lee I, Kang J (2019) Self-attention graph pooling. In: *International conference on machine learning*. PMLR, pp 3734–3743
38. Zeng H, Zhou H, Srivastava A et al (2019) Graphsaint: graph sampling based inductive learning method. arXiv preprint [arXiv:1907.04931](https://arxiv.org/abs/1907.04931)
39. Xiao Z, He S, Xu G et al (2020) A boundary element-based automatic domain partitioning approach for semi-structured quad mesh generation. *Eng Anal Bound Elem* 113:133–144
40. Velickovic P, Cucurull G, Casanova A et al (2017) Graph attention networks. arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903)
41. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint [arXiv:1607.06450](https://arxiv.org/abs/1607.06450)
42. Gao H, Ji S (2019) Graph u-nets. In: *International conference on machine learning*. PMLR, pp 2083–2092
43. Xu K, Li C, Tian Y et al (2018) Representation learning on graphs with jumping knowledge networks. In: *International conference on machine learning*. PMLR, pp 5453–5462
44. Chen X, Gong C, Liu J et al (2022) A novel neural network approach for airfoil mesh quality evaluation. *J Parallel Distrib Comput* 164:123–132
45. Reddi S J, Kale S, Kumar S (2019) On the convergence of Adam and beyond. arXiv preprint [arXiv:1904.09237](https://arxiv.org/abs/1904.09237)
46. Zhang H, Li H, Li N (2024) MeshLink: a surface structured mesh generation framework to facilitate automated data linkage. *Adv Eng Softw* 194:103661
47. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
48. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: *International conference on machine learning*. PMLR, pp 6105–6114
49. Dosovitskiy A, Beyer L, Kolesnikov A et al (2020) An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929)
50. Yuan L, Chen Y, Wang T et al (2021) Tokens-to-token vit: training vision transformers from scratch on imagenet. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 558–567
51. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
52. Morris C, Ritzert M, Fey M et al (2019) Weisfeiler and leman go neural: higher-order graph neural networks. In: *Proceedings*

of the AAAI conference on artificial intelligence, vol 33(01), pp 4602–4609

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.