A Structure-Aware Triangular Mesh Simplification Based on Graph Neural Networks (GNNs)-Guided Quadric Error Metrics (QEM)

Baoyi Zhang , Xi Yu , Wuyi Cai * , Xian Zhou , Binhai Wang , Tongyun Zhang

Posted Date: 27 April 2026

doi: 10.20944/preprints202604.1809.v1

Keywords: triangular mesh simplification; graph neural networks; quadric error metrics; structural awareness; spectral geometry

 doi:10.20944/preprints202604.1809.v1

Article

A Structure-Aware Triangular Mesh Simplification Based on Graph Neural Networks (GNNs)-Guided Quadric Error Metrics (QEM)

Baoyi Zhang [1], Xi Yu [1], Wuyi Cai [2,*], Xian Zhou [2], Binhai Wang [2] and Tongyun Zhang [2]

[1] Key laboratory of Metallogenic Prediction of Nonferrous Metals and Geological Environment Monitoring (Ministry of Education) / School of Geosciences and Info-Physics, Central South University, Changsha 410083, China

[2] Geological and Geographic Information Institute of Hunan Province / Geological Big Data Center of Hunan Province, Changsha 410021, China

* Correspondence: dxs_cwy0920@sina.cn (W. CAI).

Abstract

Triangular mesh is one of the most widely used representations for 3D surfaces. However, high-resolution mesh models often contain a large number of triangles, leading to significant burdens in storage, transmission, and real-time rendering. Mesh simplification aims to reduce model complexity while preserving geometric fidelity and structural features. Classical methods, such as quadric error metrics (QEM), rely solely on local geometric errors, making them difficult to distinguish between redundant regions and structurally important features, often resulting in feature loss and topological degradation. To address these limitations, this study proposes a structure-aware triangular mesh simplification framework based on graph neural networks (GNNs)-guided QEM. GNNs are employed as a structural importance estimator to predict geometric saliencies of mesh edges. The predicted importances are incorporated into the classical QEM edge collapse cost through a soft modulation mechanism. Furthermore, a geometry-saliency driven dynamic cost modulation strategy is designed, enabling the simplification process to prioritize critical features in early stages and gradually transition to global error minimization in later stages, without compromising the geometric optimality of QEM. In terms of model design, hybrid structural representation GNNs are constructed by integrating spectral geometry and a dual-branch architecture. Laplacian positional encoding is introduced to capture global topological information, while 1-hop and 2-hop message passing branches enable multi-scale representation of complex geometric structures. In addition, a staged inference strategy is adopted to dynamically update graph structural features during simplification, effectively mitigating topological drift. Experimental results on the TOSCA dataset demonstrate that the proposed method achieves stable performance across various simplification ratios. It consistently outperforms FQMS and QEM in terms of geometric error (P_{CD}) and normal consistency (P_{NE}). For structural preservation (P_{LE}), the method shows advantages, with win-rates generally exceeding 90%. Moreover, it significantly improves the preservation of local geometric details at low to moderate simplification ratios. In summary, the proposed method effectively enhances local structural preservation while maintaining global geometric topology, providing an interpretable and practical solution for integrating learning-based structural awareness with classical geometric optimization in mesh simplification.

Keywords: triangular mesh simplification; graph neural networks; quadric error metrics; structural awareness; spectral geometry

MSC: 86-10

 doi:10.20944/preprints202604.1809.v1

1. Introduction

Triangular mesh remains the most widely used data structure for representing 3D surfaces in geo-visualizations [1,2]. With the rapid development of 3D scanning, photogrammetry, and physical simulation technologies, high-resolution mesh models have found broad applications in digital humans, cultural heritage, industrial inspection, and 3D mapping. However, such models often contain hundreds of thousands or even millions of triangles, imposing significant storage, transmission, and real-time rendering burdens. To meet the demands of interactive applications and resource-constrained scenarios, such as mobile devices and large-scale scene rendering, progressive meshes (PM) [3] represent a high-resolution triangular mesh as a low-resolution base mesh combined with a sequence of refinement records. This improved representation supports multi-resolution visualization and levels of detail (LoD) rendering [4], and has been widely adopted in applications such as 3D gaming and photorealistic maps. Nevertheless, even within PM or multi-resolution frameworks, a fundamental mesh simplification challenge remains: how to significantly reduce mesh complexity while faithfully preserving the original geometric appearance and critical structural features.

Traditional methods are predominantly greedy, defining cost functions for vertices or edges and executing simplification operations according to priority. Among these, the quadric error metrics (QEM) proposed by Garland and Heckbert (1997) [5] are the most representative, and improvements to the classical QEM framework have been proposed via geometric constraint strategies [6], saliency metrics [7], or intrinsic error metrics [8]. Owing to their computational efficiency and implementation stability, such methods are widely used in industrial pipelines. However, relying solely on local geometric error often fails to distinguish between “redundant regions that can be safely collapsed” and “critical features essential for global shape and structure”, which can lead to premature loss of feature edges, unstable normals, or topological degradation. Therefore, mesh simplification is not merely a geometric optimization problem but also a structure-aware task.

Graph neural networks (GNN) [9,10] represent one of the most prominent approaches within geometric learning that identify which local structures are important to preserve during simplification, providing adaptive guidance for simplification decisions. It propagates information through the connectivity of irregular graph neighborhoods, making them ideal for representing structural patterns and contextual relationships within meshes [11,12]. Nonetheless, fully learning-based approaches often struggle to precisely control geometric error like QEM and are unstable under high simplification ratios or complex topologies. Furthermore, the mesh topology continuously evolves during simplification, rendering guidance predicted from the initial mesh progressively invalid and resulting in topological drift. Therefore, effectively integrating learned structure-aware information into classical simplification workflows remains challenging, especially when computational efficiency and geometric stability must be maintained.

To address this issue, this study proposes a structure-aware triangular mesh simplification framework based on GNN-guided QEM. Unlike end-to-end prediction approaches, our method employs GNNs as a structural importance estimator to predict the geometric significance of edges, which is then injected into the classical QEM process via soft modulation. Specifically, we introduce a geometry-saliency driven dynamic cost modulation strategy that adjusts edge collapse costs throughout the simplification process. This ensures that critical geometric and topological structures are preferentially preserved in the early stages, while gradually reverting to global error minimization in later stages, without compromising QEM’s geometric optimality.

The main contributions of this study are as follows:

(1) Triangular simplification framework based on GNN-guided QEM: We integrate GNN-predicted edge structural importances with classical QEM, injecting the importances via soft modulation. This preserves QEM’s advantages of minimal geometric error and algorithmic stability while introducing geometry-saliency driven structure awareness, significantly enhancing the protection of key features and providing an interpretable fusion of deep learning with classical geometry optimization.

 doi:10.20944/preprints202604.1809.v1

(2) Hybrid GNNs with spectral geometry and dual-branch architecture: We incorporate Laplacian positional encoding (Laplacian PE) to capture global topology and design a dual-branch message passing structure, enabling the network to simultaneously capture 1-hop fine-grained geometry and 2-hop neighborhood structures, thereby improving expressive power for complex geometries.

(3) Geometry-saliency driven dynamic cost modulation with staged inference: By constructing a dynamic soft-modulation function that evolves with simplification progress, we achieve a smooth transition from feature preservation to error control. Combined with staged inference updating graph-topology features, this strategy effectively mitigates topological drift while balancing computational efficiency and prediction accuracy.

2. Related Works

Mesh simplification is a fundamental problem in computer graphics, aiming to reduce model complexity while preserving critical features. It is essential for large-scale scene rendering [13] and real-time applications [14]. The field has evolved from traditional approaches focused on local Euclidean geometric errors toward multi-level methods integrating intrinsic geometry awareness, visual saliency, structural perception, and end-to-end differentiable learning.

Garland and Heckbert (1997) [5] laid the foundation of classical mesh simplification with their QEM, which gained widespread adoption due to its efficiency and was later extended to arbitrary dimensions [15,16]. However, the original QEM computes the sum of squared distances from a vertex to its adjacent triangle planes, which considers only local Euclidean geometry. As a result, it lacks sensitivity to surface topology, often causing distortions in texture mapping or geodesic distances. To address this problem, Liu et al (2023) [8] introduced an intrinsic error metric, which optimizes intrinsic distances rather than extrinsic geometry, effectively reducing the formation of elongated triangles on curved surfaces.

Beyond geometric refinements, researchers recognized that simplification should incorporate perceptual and structural awareness, not just geometric curvature. Hoppe (1999) [17] introduced an appearance-based metric to preserve texture, while Lee et al (2005) [7] proposed the concept of mesh saliency, computing the visual importance of regions using a center-surround operator on Gaussian-weighted average curvature and integrating it into QEM. This approach allocates more triangles to perceptually important regions rather than merely areas of high curvature, thereby overcoming the limitations of purely geometric metrics. Building on this, Song et al (2014) [18] enhanced saliency detection using spectral processing; Xu et al (2024) [19] introduced anisotropic optimization for weak features through the CWF algorithm; and Heep et al (2025) [20] made significant contributions to normal-integrated feature preservation. Despite these heuristic enhancements, feature extraction remains limited by manually designed geometric descriptors, making it difficult to adaptively capture complex global topologies.

With the rapid development of geometric deep learning, GNNs have emerged as effective tools for learning structural features of 3D meshes. A significant milestone in this direction is the emergence of collapse learning methods, which parameterize the edge collapse operation within neural architectures. Early work, such as MeshCNN [21], pioneered this approach by defining edge-centered convolutions and a learnable pooling operator based on edge collapse in a task-driven manner, allowing the network to retain topologically important features while discarding redundant ones. Furthermore, recent advances have integrated deep learning-driven feature sampling into the QEM framework to achieve more precise collapse control. For example, Lan, et al. [22] proposed the SFSP-QEM algorithm, which utilizes a deep learning-based salient feature-preserving point sampler to prioritize key geometric regions during the simplification process, which significantly reduces geometric distortion compared to vanilla QEM. In a broader context, numerous studies explored more general convolution operators on manifolds or non-Euclidean domains. For instance, MoNet [23] and SplineCNN [24] aggregate features within local neighborhoods using continuous kernels, providing a foundational approach for spatial-domain convolutions on non-Euclidean data. Building

 doi:10.20944/preprints202604.1809.v1

upon this, the expressive power for complex geometric relationships have been enhanced by enlarging neighborhood ranges or constructing multi-scale structures. For example, DGCNN [25] dynamically updates neighborhood relationships to better capture local structural features, while PD-MeshNet [26] expands the receptive field via multi-scale encoding to capture broader geometric context. Beyond neighborhood design, providing GNNs with global structural information has become a key research focus. For example, DiffusionNet [27] leverages diffusion operators from spectral geometry to achieve robust feature learning on discrete meshes. Inspired by these studies, this study incorporates Laplacian Positional Encoding into node features, using eigenvectors of the graph Laplacian to supply global structural context, thereby alleviating the limitations of traditional message passing neural networks (MPNNs) in expressing absolute node positions and modeling long-range structures.

In recent years, end-to-end learning-based mesh processing methods have gained increasing attention. For instance, Potamias et al (2022) [1] proposed an end-to-end learnable mesh simplification approach that bypasses traditional edge collapses, directly predicting simplified meshes via vertex sampling and candidate face generation. Similarly, Neural Progressive Meshes [28] emulate dynamic mesh editing through learning. In the domain of generative and representation learning, MeshMAE [29] explored self-supervised feature extraction via masked autoencoders, providing a new avenue for mesh structural priors, while Neural Subdivision [30] implemented learnable geometric subdivision operators.

Meanwhile, differentiable topology optimization has emerged as another research frontier. These methods model mesh geometry and topology as differentiable parameters, enabling gradients to directly influence vertex positions and connectivity for end-to-end optimization. For example, DeepMesh [31] applies differentiable processing to deep implicit fields to generate explicit meshes with topology changes; Differentiable Surface Triangulation [32] employs softened weighted Delaunay triangulation for differentiable optimization of discrete topology; FlexiCubes [33] introduces tunable parameters for mesh extraction, allowing flexible updates of geometry and connectivity during gradient descent; and DMesh [34] achieves differentiable representation for general triangular meshes, jointly modeling geometry and topology. Collectively, these end-to-end differentiable approaches provide high flexibility and gradient control over geometric and topological optimization. However, although these methods offer advantages in geometric representation flexibility and differentiable optimization capability, they often struggle to provide strict single-edge collapse error bounds or precise local geometric control like classical QEM, while their prediction processes lack intuitive geometric interpretability and face stability challenges.

Therefore, effectively introducing learning-driven structure awareness while maintaining geometric controllability remains a critical challenge for high-fidelity mesh simplification. To address this, we propose a hybrid optimization strategy, integrating deep learning-based structural awareness with classical geometric optimization to balance error control and topological feature preservation. Inspired by the multi-hop neighborhood aggregation of MixHop [35] and the multi-scale feature aggregation of PointNet++ [36], we design a dual-branch message passing structure. The 1-hop GCN branch captures fine-grained geometric features on the manifold surface, while the 2-hop GCN branch captures 2-hop geometric patterns and topological dependencies. Unlike fully data-driven generative models, we leverage GNN-predicted edge importance scores as guiding factors to dynamically soft-modulate QEM edge collapse costs. This GNN-guided QEM framework combines the structural perception capabilities of deep learning with the stability and interpretability of classical geometric error metrics, achieving a favorable trade-off between geometric fidelity and structural feature preservation.

3. Materials and Methods

We propose a mesh simplification framework guided by structure-aware GNNs, as illustrated in Figure 1. This framework integrates deep-learned geometric features with the classical QEM algorithm. The core innovation is a GNN-guided dynamic soft cost modulation strategy, which

 doi:10.20944/preprints202604.1809.v1

adjusts edge collapse costs based on learned edge importance scores, effectively preserving important geometric and topological features while substantially reducing mesh complexity. The input 3D triangular mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$ is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{1hop}, \mathcal{E}_{2hop})$, where $\mathcal{V}$ denotes the set of vertices, $\mathcal{E}_{1hop}$ is defined as the set of first-order adjacency edges derived directly from triangular faces $\mathcal{F}$, while $\mathcal{E}_{2hop}$ represents second-order connections from neighborhood relations. Given an input 3D mesh $\mathcal{M}$, the framework first performs feature and topology reconstruction. An initial node feature matrix $\boldsymbol{H}^{(0)}$ is constructed by concatenating geometric attributes, topological descriptors, and Laplacian Positional Encodings. Concurrently, a multi-scale graph comprising 1-hop local edges $\mathcal{E}_{1hop}$ and 2-hop far edges $\mathcal{E}_{2hop}$ is built. During the message passing phase, a dual-branch Graph Convolutional Network consisting of GCN_{1hop} and GCN_{2hop} aggregates 1-hop and 2-hop neighborhood information in parallel. Features from both branches are then fused via weighted linear combination and non-linear transformations to yield the final node embeddings $\boldsymbol{H}^{(l)}$. In the edge importance prediction stage, the endpoint node embeddings of each edge, along with their absolute difference and explicit geometric edge features, are concatenated and processed sequentially by a Multi-Layer Perceptron (MLP) and a Sigmoid activation function to generate the predicted edge importance score I_{uv}. Subsequently, these scores are directly integrated into the dynamic cost soft modulation stage: building upon the base QEM cost $C_{QEM}(u, v)$, a soft modulation function $\Omega(I, \gamma)$ magnifies the cost of candidate edges with high importance scores, strictly governed by a relaxation threshold $G(t)$ that dynamically evolves with the simplification progress t. This yields the final modulated cost $C^*(u, v)$. Finally, all candidate edges are ranked in a priority queue based on $C^*(u, v)$. Iterative edge collapse operations are executed subject to specific constraints (i.e., normal flip prevention, quality control, and manifold preservation), ensuring a seamless transition from rigorous feature preservation to global error minimization.

To ensure predictive reliability and geometric consistency under high simplification ratios, a staged inference scheme is introduced during the simplification process. The overall procedure is divided into S stages. At the beginning of each stage, the graph structure is reconstructed from the current intermediate mesh, and trained GNNs are invoked to re-estimate the geometric importance score I_{uv} for each edge. This allows the predictions to remain aligned with the evolving mesh geometry, thereby maintaining geometric consistency between the inferred importance scores and the underlying mesh structure. GNNs share the same network parameters across all stages, while only the topology of the input graph is updated dynamically. This design preserves computational efficiency and alleviates the topological drift that may arise when early predictions become inconsistent with the continuously evolving mesh connectivity. The staged inference mechanism operates in conjunction with the geometry-saliency driven dynamic soft cost modulation strategy. The former provides periodic global updates of edge importance, whereas the latter continuously modulates edge collapse costs. Together, they balance overall geometric error control with the preservation of critical structural features throughout the simplification process.

 doi:10.20944/preprints202604.1809.v1

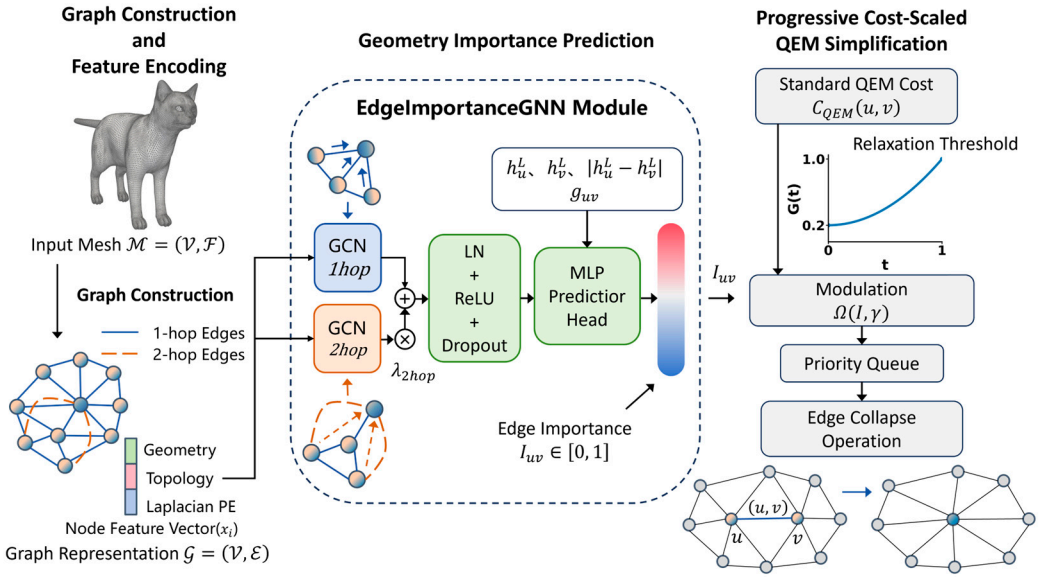


Figure 1. Mesh Simplification Framework based on GNN-guided QEM.

3.1. Graph and Its Features

3.1.1. Node Feature Initialization

To capture both local geometric details and global topological context, each vertex $v_i \in \mathcal{V}$ is initialized with a high-dimensional feature vector $\boldsymbol{x_i} \in R^{\{D_{in}\}}$, where D_{in} denotes the dimensionality of input features. The feature vector is constructed by concatenating three components:

(1) Geometric attributes. These include 3D vertex coordinates (x, y, z) and normal vectors $\boldsymbol{n} = (\boldsymbol{n}_x, \boldsymbol{n}_y, \boldsymbol{n}_z)$, obtained by accumulating and normalizing the normals of adjacent faces. Together they provide an explicit description of the local surface geometry.

(2) Topological descriptors. These consist of the normalized vertex valence and a binary boundary indicator $b_i \in \{0, 1\}$, which distinguishes interior manifold vertices from boundary vertices.

(3) Laplacian PE. Standard MPNNs aggregate features primarily from local neighborhoods under the constraint of permutation invariance [37]. However, due to the theoretical limitation imposed by the 1-Weisfeiler-Lehman (1-WL) test, such models are insensitive to absolute node positions and may fail to distinguish symmetric structures with locally isomorphic neighborhoods [38]. To enhance the expressive power, we introduce spectral positional encodings derived from the eigen-decomposition of the graph Laplacian, enabling the model to capture global topological information [39]. Specifically, the normalized graph Laplacian is computed as [40]:

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}, \tag{1}$$

Specifically, $\boldsymbol{I}$ denotes the identity matrix, $\boldsymbol{A}$ represents the adjacency matrix of the mesh graph, and $\boldsymbol{D}$ is the diagonal degree matrix with diagonal entries defined as $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$. This formulation corresponds to the symmetric normalized graph Laplacian, a classical definition in spectral graph theory [40], and has been applied in graph representation learning [39]. Compared to the unnormalized Laplacian ($\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$), the eigenvalues of the normalized Laplacian are bounded within the interval [0,2]. This property improves the numerical stability of spectral representations across graphs with varying sizes and connectivity patterns, thereby leading to more robust positional encodings derived from eigendecomposition on 3D meshes of different resolutions. Based on this, the eigenvectors corresponding to the smallest k non-zero eigenvalues are used as positional encodings, in order to avoid the trivial solution associated with the zero eigenvalue. The parameter k controls the trade-off between representation capacity and computational cost, and is empirically set to $k = 16$ in this work. These low-frequency eigenvectors characterize smooth geometric modes

 doi:10.20944/preprints202604.1809.v1

of the mesh surface and provide long-range positional references. To address the inherent sign ambiguity in eigen-decomposition, a maximal-component sign normalization scheme is adopted, aligning the direction of the largest-magnitude component to ensure deterministic and consistent input features.

For efficient processing by the graph neural network, the vertex features are stacked row-wise to form the input feature matrix:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{|V|} \end{bmatrix} \in \mathbb{R}^{|V| \times D_{in}}, \boldsymbol{H}^{(0)} = \boldsymbol{X}, \tag{2}$$

where the matrix $\boldsymbol{H}^{(0)}$ serves as the initial node embedding of the graph neural network, where $\boldsymbol{H}^{(l)}$ denotes the node embedding matrix at the l-th layer.

3.1.2. Multi-Scale Graph

To overcome the limitation that a single graph convolution layer aggregates information only from first-order neighbors, a multi-scale graph structure composed of 1-hop and 2-hop neighborhoods is constructed, in which two types of edge indices are defined:

(1) 1-hop Edges ($\boldsymbol{\mathcal{E}}_{1hop}$): Represent direct connections in the original mesh topology within the 1-hop neighborhood, capturing local geometric structures and surface continuity.

(2) 2-hop Edges ($\boldsymbol{\mathcal{E}}_{2hop}$): Inspired by the multi-scale grouping (MSG) strategy in PointNet++ [36], we introduce sparse connections based on the 2-hop neighborhood. For each vertex v_i, the strict 2-hop neighborhood (excluding itself and its 1-hop neighbors) is identified, and connections are established to the K (K=12 in this study) nearest vertices based on Euclidean distance. This allows the network to aggregate features across local variations and capture 2-hop geometric patterns.

The composition of the vertex feature vectors and the construction of the multi-scale graph are illustrated in Figure 2.

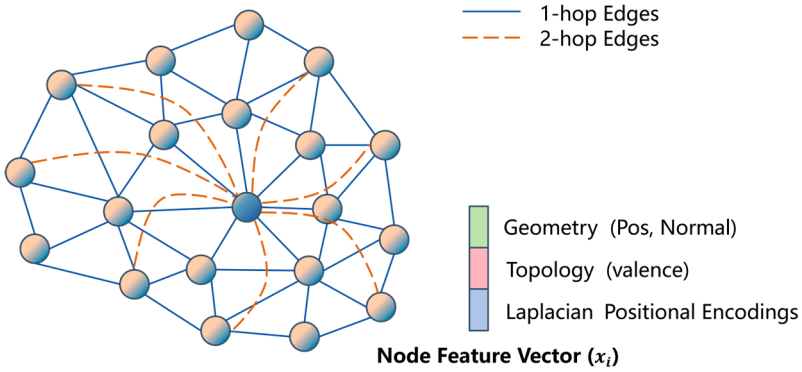


Figure 2. Illustration of vertex features and multi-scale graph construction for the triangular mesh. Solid lines indicate 1-hop edges, while dashed lines denote 2-hop edges.

3.2. Edge Importance GNNs

We propose an EdgeImportanceGNN model to predict the geometric importance of mesh edges. The model employs GCNs as the fundamental message-passing operator, aggregating neighborhood features to learn node embeddings, which are subsequently used to predict the importance probability $I_{uv} \in [0,1]$ for each undirected edge (u, v). The overall architecture of the proposed model is illustrated in Figure 3.

 doi:10.20944/preprints202604.1809.v1

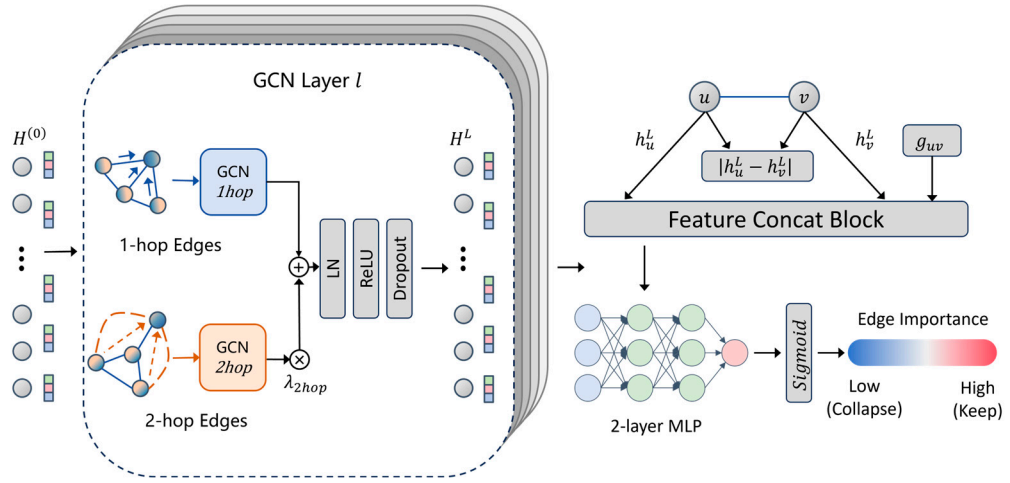


Figure 3. The EdgeImportanceGNN architecture. On the left, dual-branch GCN layers aggregate 1-hop and 2-hop neighborhood features in parallel. On the right, the edge importance prediction head combines node embeddings to estimate geometric significances of edges.

3.2.1. Dual-Branch Message Passing

Standard GCNs struggle to capture feature mixing across multi-hop neighborhoods. Inspired by the multi-hop feature mixing mechanism in MixHop [35], we design a dual-branch message passing architecture, significantly enhancing the model's ability to represent complex mesh geometries.

The architecture consists of L GCN layers, each processing multiple graph scales in parallel:

(1) 1-hop GCN branch (GCN_{1hop}): Aggregates information from 1-hop neighbors, focusing on capturing local geometric features of the manifold surface.

(2) 2-hop GCN branch (GCN_{2hop}): Captures 2-hop geometric patterns and topological correlations.

Node feature updates are performed via a linear combination of multi-scale features followed by nonlinear transformations, including layer normalization, activation, and dropout regularization:

$$\boldsymbol{H}^{(l+1)} = Dropout\left(\sigma\left(LN\left(GCN_{1hop}(\boldsymbol{H}^{(l)}, \boldsymbol{\mathcal{E}}_{1hop}) + \lambda_{2hop} \cdot GCN_{2hop}(\boldsymbol{H}^{(l)}, \boldsymbol{\mathcal{E}}_{2hop})\right)\right)\right), \tag{3}$$

where, $\boldsymbol{H}^{(l)} \in \mathbb{R}^{|V| \times D_l}$ denotes the node embeddings after the l-th layer, with D_l representing the feature dimension. $\sigma(\cdot)$ is a nonlinear activation function (ReLU in this work), and the hyperparameter λ_{2hop} balances the relative contribution of 1-hop local information and 2-hop neighborhood context during feature fusion. Notably, GCN_{1hop} and GCN_{2hop} share the standard GCN formulation but maintain independent weight matrices, each responsible for extracting spatial features at different scales, forming a functionally complementary dual-branch structure.

3.2.2. Edge Importance Prediction Head

After L layers of message passing, the final node embedding matrix $\boldsymbol{H}^L \in \mathbb{R}^{|V| \times D_L}$ is obtained, where the embedding of node i is denoted as $\boldsymbol{h}_i^L$, corresponding to the i-th row of $\boldsymbol{H}^L$. For any undirected edge (u, v), its edge feature vector $\boldsymbol{z}_{uv}$ is constructed via the Feature Concat Module. This vector is formed by concatenating the embeddings of its endpoints, their absolute difference, and explicit geometric edge attributes $\boldsymbol{g}_{uv}$:

$$\boldsymbol{z}_{uv} = Concat(\boldsymbol{h}_u^L, \boldsymbol{h}_v^L, |\boldsymbol{h}_u^L - \boldsymbol{h}_v^L|, \boldsymbol{g}_{uv}), \tag{4}$$

where the geometric attributes $\boldsymbol{g}_{uv}$ comprise edge sharpness and normalized edge length. Edge sharpness is computed from the normalized dihedral angle and is explicitly enhanced for topological boundary edges (by taking the maximum value), providing a unified numerical scale to characterize both internal sharp edges and open boundary edges. The absolute difference term $|\boldsymbol{h}_u^L - \boldsymbol{h}_v^L|$ enables the model to sensitively detect abrupt changes in node embeddings across feature boundaries, such as sharp creases, thereby assigning higher retention weight to geometrically salient regions. The edge

 doi:10.20944/preprints202604.1809.v1

importance I_{uv} is then predicted via a two-layer multilayer perceptron (MLP, φ) with a sigmoid activation:

$$I_{uv} = Sigmoid\big(\varphi(\boldsymbol{z}_{uv})\big), \tag{5}$$

3.3. QEM Simplification with GNN-Guided Dynamic Soft Modulation

We further propose a dynamic soft modulation strategy that integrates GNN-predicted edge geometric importance into the simplification process. Unlike conventional static weighting schemes, this strategy employs a dynamic cost modulation function that evolves with the simplification progress, enabling a smooth transition from strict feature preservation to global error control. Notably, this approach constitutes a soft modulation of edge collapse priorities and does not compromise the underlying QEM algorithm's geometric error minimization.

3.3.1. QEM Cost Initialization

For a candidate edge (u, v) and its optimal collapse position $\bar{\boldsymbol{v}}$, the base geometric error is determined by the accumulated quadric matrices $\boldsymbol{Q}$:

$$C_{QEM}(u, v) = \bar{\boldsymbol{v}}^T(\boldsymbol{Q}_u + \boldsymbol{Q}_v)\bar{\boldsymbol{v}}, \tag{6}$$

where $\boldsymbol{Q}_u$ and $\boldsymbol{Q}_v$ are the quadric error matrices accumulated from the planes of adjacent triangles at vertices u and v, respectively.

To prevent the formation of degenerate meshes during simplification, the following constraints are incorporated:

(1) Normal Flip Check: Collapse is prohibited if it induces a change in the angle between adjacent face normals exceeding a threshold θ_{thresh} (e.g., 60°).

(2) Quality Control: Collapses that generate extremely elongated triangles (with low aspect ratio) are penalized or disallowed.

(3) Manifold Preservation: Edge collapse is permitted only on 2-manifold edges to maintain topological consistency.

3.3.2. Dynamic Cost Soft Modulation

To regulate the strength of feature preservation throughout the simplification process, we introduce a dynamic cost soft modulation strategy. This approach incorporates a relaxation threshold that evolves with the simplification progress, balancing geometric fidelity against the reduction rate during topological evolution.

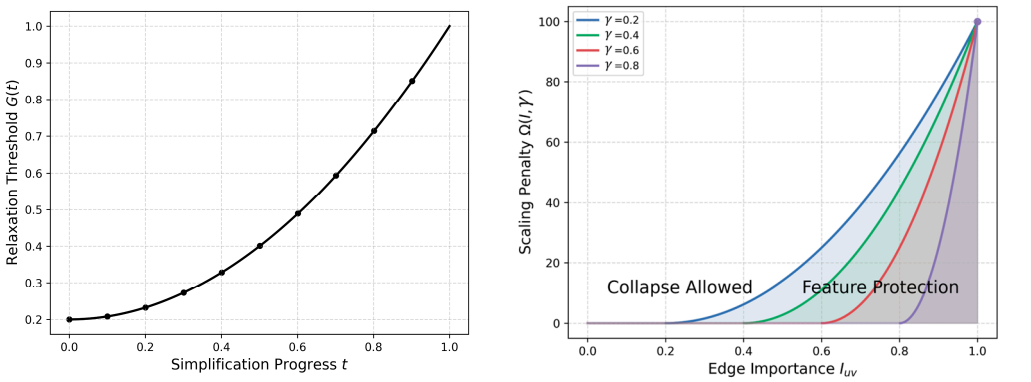
We first define a monotonic, continuous function $G(t)$ over the normalized simplification progress $t \in [0,1]$ to control the influence of geometric importance in edge collapse costs:

$$G(t) = G_{start} + (1 - G_{start}) \cdot t^p, \tag{7}$$

where $t = (N_{start} - N_{curr})/(N_{start} - N_{target})$ is the normalized simplification factor, with N denoting the current face count. G_{start} is the initial protection threshold (set to 0.2 in this study), and p ($p = 2.0$ in this study) is a power exponent controlling the relaxation rate.

This function effectively establishes a dynamic soft modulation mechanism. As illustrated in Figure 4a, $G(t)$ maintains strong constraints on high-importance edges during the early stages of simplification and gradually relaxes feature preservation requirements in later stages. This ensures that target face counts are achievable while preserving the geometric optimality of QEM and maintaining stable global error control.

 doi:10.20944/preprints202604.1809.v1



(**a**) (**b**)

Figure 4. Illustration of the QEM simplification mechanism with dynamic cost soft modulation: (a) Relaxation threshold $G(t)$ varying with simplification progress t, showing a smooth transition from strict feature preservation to global error control. (b) Continuous penalty scaling function $\Omega(I,\gamma)$, which penalizes high-importance edges to ensure early-stage protection of salient geometric features.

The final dynamically modulated edge collapse cost $C^*(u,v)$ is given by:

$$C^*(u,v) = C_{QEM}(u,v) \cdot \Big(1 + \Omega\big(I_{uv}, G(t)\big)\Big), \tag{8}$$

where Ω is a continuous scaling penalty term that integrates the GNN-predicted geometric importance score I_{uv} with the current relaxation threshold γ ,where $\gamma = G(t)$ represents the adaptive relaxation strength at simplification step t. The penalty function is defined as:

$$\Omega(I,\gamma) = \begin{cases} \lambda_{scale} \cdot ((I-\gamma)/(1-\gamma+\varepsilon))^2, & if\ I > \gamma \\ 0, & otherwise \end{cases}, \tag{9}$$

where $\varepsilon = 10^{-6}$ ensures numerical stability, and λ_{scale} is a scaling factor enhancing the early-stage protection of high-importance edges in the priority queue (set to 100.0 in this study).

To strengthen the theoretical foundation of the proposed dynamic soft modulation, we analyze the analytical properties of the scheduling function $G(t)$ and the penalty function $\Omega(I,\gamma)$, where $\gamma = G(t)$. These properties mathematically guarantee the algorithmic stability and the asymptotic convergence of our method.

(1) Monotonicity and Convexity of $G(t)$: Given $t \in [0,1]$, $G_{start} \in (0,1)$, and $p \geq 1$, the first derivative of the scheduling function with respect to the simplification progress t is $G'(t) = (1 - G_{start})pt^{p-1} \geq 0$. This ensures that $G(t)$ is strictly monotonically increasing, mapping the domain $[0,1]$ to $[G_{start}, 1]$. Furthermore, with $p = 2.0$, the second derivative $G''(t) = 2(1 - G_{start}) > 0$ indicates that $G(t)$ is a strictly convex function. From an algorithmic perspective, this convexity ensures that the relaxation of feature protection is initially slow (maintaining strong geometric constraints when the mesh is dense) and accelerates only as the mesh approaches the target decimation ratio.

(2) Non-negativity and Asymptotic Convergence to Standard QEM: $\Omega(I,\gamma) \geq 0$ strictly holds for all I and γ. Consequently, the modulated cost satisfies $C^*(u,v) \geq C_{QEM}(u,v)$. This implies that the modulation only penalizes feature-rich edges and never underestimates the base geometric error. Crucially, as the simplification reaches its target ($t \to 1$), the scheduling function converges to its upper bound: $\lim_{t\to 1} G(t) = 1$. Since the predicted importance score is bounded by $I_{uv} \in [0,1]$, the activation condition $I_{uv} > G(t)$ becomes mathematically impossible to satisfy. Therefore, $\lim_{t\to 1} \Omega(I,\gamma) = 0$, leading to $\lim_{t\to 1} C^*(u,v) = C_{QEM}(u,v)$. This asymptotic behavior rigorously proves that the barrier naturally vanishes at the tail end of the simplification process, guaranteeing that our framework degrades to the exact classical QEM, inheriting its fundamental local error minimization guarantees.

(3) C^1 Continuity (Smoothness) of the Penalty Function: The penalty term Ω is defined as a piecewise function, which introduces a potential risk of algorithmic instability (e.g., abrupt priority

 doi:10.20944/preprints202604.1809.v1

inversions in the min-heap) if the cost landscape is discontinuous. However, our formulation is deliberately constructed to be differentiable. Let $x = I - \gamma$. The function transitions at $x = 0$. The left limit approaches 0, and the right limit is $\lim_{x\to 0^+} \lambda_{scale}\left(\frac{x}{1-\gamma+\epsilon}\right)^2 = 0$. Thus, Ω is C^0 continuous. Furthermore, evaluating the first partial derivative with respect to I: $\frac{\partial \Omega}{\partial I} = \begin{cases} \frac{2\lambda_{scale}(I-\gamma)}{(1-\gamma+\epsilon)^2}, & \text{if } I > \gamma \\ 0, & \text{otherwise} \end{cases}$. At the boundary $I = \gamma$, the right derivative evaluates to 0, perfectly matching the left derivative. Thus, $\Omega(I, \gamma)$ is C^1 continuous. This mathematical smoothness ensures that as edges traverse the activation threshold during the dynamic relaxation of $G(t)$, their collapse costs experience continuous and differentiable changes. This prevents sudden stochastic jumps in the priority queue, ensuring deterministic and robust topological evolution.

As shown in Figure 4b, the continuous penalty function $\Omega(I, \gamma)$ increases sharply for edges whose importance exceeds the current dynamic threshold, thereby amplifying their collapse cost, reducing their priority in the queue, and preventing premature removal of geometrically important features during early simplification stages.

3.3.3. Overall Implementation Pipeline

The overall execution procedure of the proposed GNN-guided mesh simplification framework is presented in pseudocode form (see Table 1). The method follows a dual-timeline architecture that combines staged global inference with continuous local optimization, which can be decomposed into three core phases: (1) Staged global graph inference. A multi-scale graph is reconstructed at each stage and an EdgeImportanceGNN is used to predict edge importance scores I. These scores remain fixed within each stage to preserve global consistency. (2) QEM initialization and priority queue construction. Quadric error costs are computed for all edges and modulated by GNN-predicted importance to form final collapse priorities. All edges are inserted into a priority queue for subsequent optimization. (3) Dynamic edge collapse and local update. The algorithm iteratively collapses the minimum-cost edge under validity constraints and updates local topology and quadric errors. Neighborhood edge costs are continuously re-evaluated during the process. The three stages are executed sequentially and repeated for S stages, forming a progressive refinement pipeline. The algorithm terminates once the target compression ratio is achieved, producing the final simplified mesh M_{out}.

Table 1. GNN-guided dynamic soft modulation QEM mesh simplification algorithm.

Algorithm 1: GNN-Guided Dynamic Soft Modulation QEM Simplification
Input:
$\mathcal{M}_{in} = (\mathcal{V}, \mathcal{F})$: Original 3D mesh
N_{target}: Target face count
S: Total number of staged inference steps
Θ: Trained EdgeImportanceGNN parameters
Output:
$\mathcal{M}_{out}$: Simplified 3D mesh
Definitions:
GNN$(G, H; \Theta)$: Returns predicted edge importance scores I for the graph
$G(t)$: Computes dynamic relaxation threshold at normalized progress t
$\Omega(I, \gamma)$: Computes soft penalty scale given importance I and threshold γ
ExtractMin(PQ): Pops and returns the edge with the tuple $(u, v, C^*(u, v), \bar{v}_{uv})$ with the minimum cost
ValidCollapse$(u, v, \bar{v})$: Boolean check for manifold, normal flip, quality, and valence

 doi:10.20944/preprints202604.1809.v1

Constraints

Procedure:

$\mathcal{M} \leftarrow \mathcal{M}_{in}$, $N_{start} \leftarrow N_{curr}$

Compute per-stage geometric decay ratio: $\rho \leftarrow (N_{target}/N_{start})^{\left(\frac{1}{S}\right)}$

for stage $s = 1$ to S do

    // Phase 1: Staged GNN Inference

    Set stage target: $N_{stage} \leftarrow \max\left(N_{start} \cdot \rho^{s}, N_{target}\right)$

    // Update stage progress and dynamic scheduling threshold

    Reconstruct multi-scale graph $G = (V, \mathcal{E}_{1hop}, \mathcal{E}_{2hop})$ from current mesh $\mathcal{M}$

    Extract node feature matrix $H^{(0)}$ (Geometric, Topological, Laplacian PE)

    Predict edge importance: $I \leftarrow \mathrm{GNN}(G, H^{(0)}; \Theta)$

    // Phase 2: QEM & Priority Queue Initialization

    Compute base quadric matrices Q_v for all vertices $v \in V$

    Initialize an empty priority queue PQ

    Compute stage initial progress: $t \leftarrow 0$

    Compute dynamic relaxation threshold: $\gamma \leftarrow G(t)$

    for each valid edge $(u, v) \in \mathcal{E}_{1hop}$ do

        Solve optimal collapse position $\bar{v}$ and compute base cost $C_{QEM}(u, v)$

        Modulated cost $C^{*}(u, v) \leftarrow C_{QEM}(u, v) \cdot (1 + \Omega(I_{uv}, \gamma))$

        Insert $(u, v, C^{*}, \bar{v}_{uv})$ into PQ

    end for

    // Phase 3: Dynamic Edge Collapse (Continuous Local Optimization)

    while $N_{curr} > N_{stage}$ and PQ is not empty do

        $(u, v, C^{*}, \bar{v}_{uv}) \leftarrow \mathrm{ExtractMin}(PQ)$

        if edge (u, v) or vertices u, v are dead then continue

        if not ValidCollapse$(u, v, \bar{v}_{uv})$ then continue

        Collapse edge $(u, v) \rightarrow \bar{v}_{uv}$ and update mesh connectivity $\mathcal{M}$ and N_{curr}

        Update quadric strictly via accumulation: $Q_{\bar{v}_{uv}} \leftarrow Q_u + Q_v$

        // I_{uv} remains fixed within the current stage; surviving edges retain their scores

        Update normalized stage progress:

            $t = (N_{start} - N_{curr})/\left(N_{start} - N_{target}\right)$

        Update dynamic relaxation threshold: $\gamma \leftarrow G(t)$

        for each affected edge (x, y) in local 1-ring neighborhood of $\bar{v}_{uv}$ do

            Solve new optimal position $\bar{v}_{xy}$ and recompute base cost $C_{QEM}(x, y)$

            Recompute final cost $C^{*}(x, y) \leftarrow C_{QEM}(x, y) \cdot (1 + \Omega(I_{xy}, \gamma))$

            Push updated $\left(x, y, C^{*}, \bar{v}_{xy}\right)$ into PQ

        end for

    end while

end for

return $\mathcal{M}_{out} \leftarrow \mathcal{M}$

 doi:10.20944/preprints202604.1809.v1

3.4. Loss Function

Since an ideal simplified mesh does not possess a unique "ground-truth" topology, it is difficult to directly supervise the evolution strategy through explicit labels. To address this issue, we construct a composite objective function $\mathcal{L}$ consisting of structural representation loss, geometry-aware guidance loss, and spatial regularization terms. This formulation enables joint self-supervised and weak-supervised training without requiring manual annotations.

3.4.1. Structural Contrastive Loss ($\mathcal{L}_{con}$)

To learn topology-aware latent representations, a margin-based contrastive objective is introduced to impose self-supervised constraints on node embeddings. The loss encourages adjacent nodes to maintain high similarity in the embedding space while suppressing excessive similarity between non-adjacent nodes, thereby forming a structured feature representation space:

$$\mathcal{L}_{con} = \mathbb{E}_{(u,v)\in\mathcal{E}}[1 - sim(h_u, h_v)] + \mathbb{E}_{(u,k)\notin\mathcal{E}}[\max(0, sim(h_u, h_k) - m)], \tag{10}$$

where $sim(\cdot)$ denotes the cosine similarity operator, h represents the vertex embeddings learned by the GNN, and m is the similarity margin parameter (set to $m = 0$ in this study). This objective encourages the embedding space to explicitly encode mesh adjacency relationships, while pushing the representations of non-connected nodes toward orthogonality, thereby providing stable structural features for subsequent geometric importance prediction.

3.4.2. Geometry-Aware Loss ($\mathcal{L}_{geo}$)

To convert explicit geometric priors into learnable supervisory signals, we design a unified geometry-aware loss that combines an absolute importance loss and a relative ranking loss, guiding the network to learn an edge-importance distribution consistent with geometric characteristics:

$$\mathcal{L}_{geo} = \mathcal{L}_{hinge} + \lambda_{rank}\mathcal{L}_{rank}, \tag{11}$$

(1) Hinge Loss

For the set of sharp feature edges $\mathcal{E}_{sharp}$, identified by high dihedral-angle thresholds or boundary conditions, a hinge loss is imposed to ensure that their importance scores do not fall below an empirical threshold τ_{min}:

$$\mathcal{L}_{hinge} = \mathbb{E}_{e\in\mathcal{E}_{sharp}}[max(0, \tau_{min} - I_e)], \tag{12}$$

where τ_{min} is the minimum importance threshold (set to 0.6).

(2) Pairwise Ranking Loss

To strengthen the distinction between feature regions and flat areas, a pairwise ranking loss is introduced. This loss samples edge pairs (e_i, e_j) from the sharp-feature set $\mathcal{E}_{sharp}$ and the flat-region set $\mathcal{E}_{flat}$, respectively, and enforces a minimum margin μ between their predicted importance scores:

$$\mathcal{L}_{rank} = \mathbb{E}_{e_i\in\mathcal{E}_{sharp}, e_j\in\mathcal{E}_{flat}}\left[\max\left(0,\ \mu - \left(I_{e_i} - I_{e_j}\right)\right)\right], \tag{13}$$

where the ranking margin is set to μ=0.1. The set $\mathcal{E}_{flat}$ corresponds to geometrically smooth regions automatically identified using a low dihedral-angle threshold. By injecting geometric priors through relative ranking, this constraint encourages the network to assign significantly higher importance scores to sharp edges than to flat edges, ensuring that salient geometric structures are preferentially preserved during simplification.

3.4.3. Local Smoothness Regularization ($\mathcal{L}_{sm}$)

To suppress local prediction noise and maintain continuity in topological evolution, a local smoothness regularization term is introduced. This term encourages edges sharing the same vertex to have similar importance scores, preventing irregular collapse ordering within local regions:

 doi:10.20944/preprints202604.1809.v1

$$\mathcal{L}_{sm} = \mathbb{E}_{(u,v)\in\mathcal{E}} \left| I_{uv} - \frac{\bar{I}_u + \bar{I}_v}{2} \right|, \tag{14}$$

where $\bar{I}_u$ denotes the mean importance of all edges incident to vertex u. By penalizing abrupt variations in local importance scores, this regularization improves the stability of the simplification process during topological evolution.

3.4.4. Total Loss Function

The final training objective is defined as a weighted linear combination of the above loss terms:

$$\mathcal{L} = \lambda_{con}\mathcal{L}_{con} + \lambda_{geo}\mathcal{L}_{geo} + \lambda_{sm}\mathcal{L}_{sm}, \tag{15}$$

where $\lambda_{con}, \lambda_{geo}$, and λ_{sm} denote weighting coefficients of loss terms, allowing a controllable trade-off between structural consistency, geometric preservation, and prediction smoothness.

4. Results

4.1. Dataset and Baselines

To test the proposed method, we use the benchmark TOSCA dataset, which contains 80 high-resolution meshes [41]. To ensure robust evaluation, we implement a five-fold cross-validation strategy. Specifically, the dataset is partitioned into five mutually exclusive folds, with each fold utilizing 64 meshes for training and the remaining 16 meshes reserved exclusively for testing. For comparison, we consider two representative baselines, including a classical geometry-saliency driven approach and a feature-preserving engineering-oriented method. For each of the 80 meshes, we perform five simplification experiments with target reduction ratios of 0.05, 0.1, 0.2, 0.5, providing a comprehensive assessment of method performance under varying levels of mesh simplification.

QEM: As a theoretical baseline for geometric error minimization, we adopt the classical QEM algorithm proposed by Garland and Heckbert (1997) [5]. The experiments are conducted using the standard implementation provided by Open3D [42] to ensure reproducibility and fair comparison. This implementation explicitly preserves the original mesh boundaries. Consequently, non-zero P_{WA} values appear for TOSCA models with open boundaries (e.g., cat and dog), reflecting the intrinsic topological characteristics of the data rather than algorithmic degradation. For watertight models (e.g., centaur), the implementation correctly maintains topological integrity and yields 0% P_{WA}. Compared with implementations that ignore boundaries or implicitly fill holes, the Open3D strategy provides a more conservative and consistent evaluation protocol.

Fast Quadric Mesh Simplification (FQMS): To evaluate the performance differences between learning-based approaches and engineering-level mesh simplification strategies, we select FQMS as a representative mesh simplification method. FQMS is a highly engineered implementation of the classical QEM framework. It accelerates the simplification process by using threshold-based edge collapses, avoiding the computational cost of global sorting, and aims to balance computational efficiency with geometric fidelity. Owing to its exceptional runtime performance and robust handling of watertight volumes, FQMS has been integrated as a core component in authoritative medical imaging platforms such as 3D Slicer and serves as the underlying engine of the Python geometry processing library pyfqmr. Since FQMS is primarily designed to maximize processing speed and topological robustness, rather than explicitly preserve local geometric or visual features, it may exhibit certain geometric deviations at high simplification ratios. Including FQMS in the comparison allows an objective assessment of the proposed method's improvements in key metrics, including geometric accuracy, normal consistency, and topological fidelity, relative to commonly used engineering-level tools. For experimental reproducibility, we employ the publicly available open-source implementation [43].

 doi:10.20944/preprints202604.1809.v1

4.2. Experimental Environment and Model Settings

The experiments in this study were implemented using the PyTorch and PyTorch Geometric (PyG) deep learning frameworks [44]. The specific hardware and software configurations of the experimental platform are detailed in Table 2.

Table 2. Experimental environment configuration.

Configuration	Value
CPU	Intel(R) Core(TM) i7-10700 @ 2.90 GHz
GPU	NVIDIA GeForce RTX 2060 SUPER (8GB)
RAM	32GB
Operating System	Windows 10 Pro 64-bit
DL Framework	PyTorch 2.4.0 + PyTorch Geometric 2.5.3
CUDA Version	12.1

The specific network architecture configurations and training hyperparameters are summarized in Table 3. The network encoder takes 21-dimensional composite node features as input, which consists of a 3D surface normal, a 2D structural prior, and a 16-dimensional Laplacian positional encoding. The encoder utilizes three graph convolutional layers (GCNConv) with a hidden dimension of 64. For edge importance prediction, a geometry late-fusion decoder constructs a 194-dimensional feature vector for each edge and processes it via a two-layer MLP to output importance scores. The model is trained for 50 epochs with a batch size of 1. A fixed random seed (seed=42) is applied to minimize performance variance induced by randomness.

Table 3. Architecture configurations and training hyperparameters of EdgeImportanceGNN.

Category	Configuration Item	Value / Setting
Input Features	Surface normal dimensionality	3
	Structural feature dimensionality	2
	Laplacian positional encoding dimensionality (k)	16
Network Architecture	GNN convolution operator	GCNConv
	Number of GNN layers (L)	3
	Hidden feature dimensionality (C_{hidden})	64
	Normalization strategy	LayerNorm
	Dropout rate	0.15
Multi-scale Graph	Residual fusion weight for 2-hop neighbors (λ_{2hop})	0.5
	Maximum number of far neighbors (K)	12
Edge Decoder	Geometric edge feature dimensionality	2
	Edge MLP input feature dimension	194
Training Settings	Optimizer	Adam
	Initial learning rate	1×10^{-3}
	Training epochs	50
	Batch size	1
Loss Weights	Structural contrastive loss weight (λ_{con})	1.0
	Geometric hinge loss weight (λ_{hinge})	0.7
	Pairwise ranking loss weight (λ_{rank})	0.25
	Local smoothness regularization weight (λ_{sm})	0.2

4.2. Evaluation Metrics

To objectively evaluate performances of different mesh simplification methods in terms of geometric accuracy, normal consistency, topological integrity, and feature preservation, this study adopts the evaluation protocol proposed by Potamias et al (2022) [1]. The metrics include the

 doi:10.20944/preprints202604.1809.v1

percentage of wrong adjacency (P_{WA}), point-wise chamfer distance (P_{CD}), point-sampled normal error (P_{NE}), Laplacian spectrum error (P_{LE}), as well as the simplification time.

4.2.1. Percentage of Wrong Adjacency (P_{WA})

The percentage of wrong adjacency (P_{WA}) is used to measure the topological integrity of the simplified mesh. In an ideal closed manifold mesh, each edge should be shared by exactly two faces. Edges that violate this condition are classified as non-manifold edges, including boundary edges and singular edges. $\mathcal{E}_{total}$ denote the set of all edges in the simplified mesh, and $\mathcal{E}_{wrong}$ denote the set of edges that are either shared by only one triangle (boundary edges) or by more than two triangles (non-manifold edges). Compared with manifold-based indicators that only check manifoldness, P_{WA} more effectively captures triangulation defects caused by holes or cracks in the mesh, as follows:

$$P_{WA} = \frac{|\mathcal{E}_{wrong}|}{|\mathcal{E}_{total}|} \times 100\%, \tag{16}$$

A lower P_{WA} value indicates that the simplified mesh is closer to a closed manifold in terms of topology.

4.2.2. Point-Wise Chamfer Distance (P_{CD})

The point-wise chamfer distance (P_{CD}) measures the global geometric discrepancy between the simplified mesh and the original model. A fixed number of points are uniformly sampled on the surfaces of both meshes. The nearest-point distances between the two point-sets are then computed in a bidirectional manner:

$$P_{CD} = \frac{1}{N}\sum_{\{p\in P_o\}} \min_{q\in M_s} \|p-q\|_2 + \frac{1}{N}\sum_{\{q\in P_s\}} \min_{p\in M_o} \|q-p\|_2, \tag{17}$$

where P_o and P_s denote the sampled point sets from the original mesh and the simplified mesh, respectively, and $\|\cdot\|_2$ represents the Euclidean distance. A smaller P_{CD} indicates that the simplified mesh better approximates the overall geometry of the original model.

4.2.3. Point-Sampled Normal Error (P_{NE})

The point-sampled normal error (P_{NE}) evaluates the consistency of local surface orientations before and after simplification. For each pair of nearest points, the angular difference between their unit normals is computed, and the bidirectional average is taken:

$$P_{NE} = \frac{1}{2}\left(\frac{1}{N}\sum_{p\in P_o} \theta\left(\boldsymbol{n}_p, \boldsymbol{n}_{\hat{q}}\right) + \frac{1}{N}\sum_{q\in P_s} \theta\left(\boldsymbol{n}_q, \boldsymbol{n}_{\hat{p}}\right)\right), \tag{18}$$

where $p \in P_o$ and $q \in P_s$ denote points sampled from the surfaces of the original mesh M_o and the simplified mesh M_s, respectively. For any sampled point p, $\hat{q}$ denotes its closest point on the simplified mesh M_s. Similarly, for any sampled point q, $\hat{p}$ denotes its closest point on the original mesh M_o. The unit normals at the corresponding locations are denoted by n_p, $n_{\hat{q}}$, n_q, and $n_{\hat{p}}$.

The angular function is used to measure the geometric deviation between two unit normal vectors. For two normalized surface normals $\mathbf{n}_p$ and $\mathbf{n}_{\hat{q}}$, their dot product and the angle θ satisfy the standard Euclidean identity:

$$\boldsymbol{n}_p \cdot \boldsymbol{n}_{\hat{q}} = cos\ \theta, \tag{18}$$

Accordingly, the angle can be obtained via the inverse cosine function, and the angular function is defined as:

$$\theta\left(\boldsymbol{n}_p, \boldsymbol{n}_{\hat{q}}\right) = \arccos\left(clip\left(\boldsymbol{n}_p \cdot \boldsymbol{n}_{\hat{q}}, -1, 1\right)\right), \tag{19}$$

where $\text{clip}(\cdot, -1, 1)$ constrains the dot product to the interval $[-1, 1]$, preventing numerical instability caused by floating-point errors. The P_{NE} value is reported in radians, where smaller values indicate better preservation of surface normal consistency.

 doi:10.20944/preprints202604.1809.v1

4.2.4. Laplacian Spectrum Error (P_{LE})

The Laplacian spectrum error (P_{LE}) evaluates the consistency of global structural and intrinsic topological properties between the original and simplified meshes. Since the number of vertices typically changes after simplification, eigenvectors cannot be directly aligned due to dimensional mismatch. Therefore, this work adopts the eigenvalue spectrum of the normalized Laplacian matrix, commonly referred to as Shape-DNA [45], as a global structural descriptor. This formulation ensures that the metric focuses on intrinsic geometric properties while remaining invariant to changes in vertex count.

The normalized Laplacian matrix is defined as:

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}, \tag{20}$$

where $\boldsymbol{A}$ is the adjacency matrix, and $\boldsymbol{D}$ is the degree matrix.

The first k smallest eigenvalues $\{\lambda_i\}_{i=1}^{k}$ are extracted, and the metric is defined as:

$$P_{LE} = \frac{1}{k}\sum_{i=1}^{k}(\lambda_i^o - \lambda_i^s)^2. \tag{21}$$

If the simplified mesh contains fewer than k valid eigenvalues, the missing components are padded with zeros to penalize the loss of topological modes. A smaller P_{LE} value indicates that the simplified mesh better preserves the intrinsic structural characteristics of the original model.

4.3. Model Performances

4.3.1. Model Comparison

To comprehensively evaluate the proposed method's geometric preservation capability, we compared three approaches: QEM, FQMS, and our method across multiple evaluation metrics at different simplification levels. For fair comparison, all methods were evaluated on the same 5-fold data splits, where the proposed method follows a cross-validation protocol while the baseline methods (QEM and FQMS) are directly applied without training. Table 4 summarizes the mean and median values of P_{WA}, P_{CD}, P_{NE}, and P_{LE} for simplification ratios of 0.05, 0.1, 0.2 and 0.5. Overall, as the simplification ratio increases, all methods show a gradual reduction in P_{CD}, P_{NE}, and P_{LE} errors, indicating that models retaining more faces better approximate the original geometry.

Table 4. Performances of different methods on the TOSCA dataset under various simplification ratios ($Ratio = \frac{N_S}{N_{org}}$).

Ratio	Method	P_{WA}		P_{CD}		P_{NE}		P_{LE}	
		mean	median	mean	median	mean	median	mean	median
0.05	QEM	0.1234±0.0191	0.1428±0.0111	0.6632±0.1977	0.5144±0.1469	0.1494±0.0166	0.1379±0.0195	0.3761±0.0509	0.3329±0.0545
	FQMS	0.1668±0.0253	0.1573±0.0175	0.8936±0.2517	0.6923±0.1689	0.1630±0.0168	0.1517±0.0153	0.3758±0.0508	0.3317±0.0547
	Proposed	0.1622±0.0270	0.1982±0.0205	0.6043±0.1750	0.4772±0.1288	0.1431±0.0154	0.1323±0.0174	0.3759±0.0510	0.3327±0.0544
0.1	QEM	0.1081±0.0257	0.1008±0.0170	0.3283±0.1041	0.2515±0.0782	0.1049±0.0137	0.0936±0.0170	0.2715±0.0367	0.2399±0.0393
	FQMS	0.1488±0.0468	0.1178±0.0436	0.4575±0.1352	0.3445±0.0891	0.1193±0.0138	0.1091±0.0138	0.2714±0.0368	0.2397±0.0393
	Proposed	0.1053±0.0181	0.1326±0.0159	0.3020±0.0923	0.2365±0.0691	0.1011±0.0129	0.0905±0.0153	0.2714±0.0368	0.2402±0.0392
0.2	QEM	0.0810±0.0185	0.0766±0.0086	0.1390±0.0529	0.1013±0.0453	0.0620±0.0109	0.0532±0.0144	0.1999±0.0271	0.1766±0.0287
	FQMS	0.1032±0.0287	0.0876±0.0239	0.2174±0.0716	0.1634±0.0523	0.0783±0.0112	0.0690±0.0134	0.1999±0.0271	0.1765±0.0287
	Proposed	0.0825±0.0186	0.0880±0.0089	0.1294±0.0456	0.0970±0.0381	0.0598±0.0101	0.0514±0.0133	0.1999±0.0271	0.1766±0.0286
0.5	QEM	0.0641±0.0145	0.0562±0.0029	0.0200±0.0097	0.0128±0.0099	0.0149±0.0042	0.0115±0.0063	0.1412±0.0191	0.1249±0.0199
	FQMS	0.0759±0.0182	0.0718±0.0176	0.0409±0.0190	0.0273±0.0175	0.0220±0.0057	0.0175±0.0082	0.1411±0.0191	0.1249±0.0200
	Proposed	0.0791±0.0213	0.0838±0.0134	0.0210±0.0079	0.0166±0.0084	0.0154±0.0035	0.0130±0.0049	0.1411±0.0190	0.1248±0.0199

To ensure fairness and statistical consistency in our evaluation, we adopted a strict co-removal strategy. Specifically, any sample exhibiting catastrophic numerical divergence was classified as extreme failure case if its error exceeded the median by more than one order of magnitude (i.e., >10 ×), and was subsequently excluded from all evaluated methods. This procedure effectively mitigates statistical bias introduced by such outliers, guaranteeing that comparisons are conducted on a consistently aligned set of valid samples.

 doi:10.20944/preprints202604.1809.v1

Analysis of the excluded cases further reveals stark differences in robustness among the methods. Across all tested simplification ratios, classical heuristic-based algorithms (QEM and FQMS) exhibited numerical explosions on six distinct models (cat1, david1, dog1, gorilla1, victoria1, and victoria2), manifesting as severe geometric instability. In contrast, the proposed neural-guided method demonstrated substantially more stable numerical behavior, failing on only a single extreme case (david1). These observations suggest that incorporating deep-learning priors effectively alleviates matrix ill-conditioning and topological degradation under complex geometric configurations, thereby improving overall algorithmic robustness in challenging scenarios.

In terms of topological preservation measured by P_{WA}, the proposed method is highly competitive with QEM and consistently outperforms FQMS. Notably, at a simplification ratio of 0.1, our method achieves a mean value of 0.1053 ± 0.0181, which is slightly better than QEM (0.1081 ± 0.0257) and significantly superior to FQMS (0.1488 ± 0.0468). This indicates comparable or even improved topological integrity over classical QEM under moderate compression. However, at the extreme simplification ratio of 0.05, the proposed method yields a moderately higher P_{WA} value (0.1622 ± 0.0270) compared to QEM (0.1234 ± 0.0191). This does not indicate performance degradation, but rather reflects the inherent behavior of the GNN-based feature-aware collapse strategy under severe vertex budget constraints. By prioritizing the preservation of high-curvature and structurally salient regions, the method actively avoids aggressive boundary collapses that are typically forced under extreme compression, which often lead to severe geometric distortion. Although this design choice may introduce a small fraction of local non-manifold configurations, it effectively circumvents catastrophic geometric collapse and maintains robust overall representational quality, as further evidenced by improvements in P_{CD} and P_{NE}. This demonstrates a strategic trade-off favoring geometric expressiveness over strict topological watertightness.

Regarding geometric deviation measured by P_{CD}, the proposed method consistently achieves lower or comparable errors across the majority of settings. Under the most challenging simplification ratio of 0.05, our approach achieves 0.6043 ± 0.1750, outperforming QEM (0.6632 ± 0.1977) and significantly surpassing FQMS (0.8936 ± 0.2517). This advantage remains stable at ratios of 0.1 and 0.2, consistently outperforming both baselines. Even at a high simplification ratio of 0.5, where all methods naturally converge to relatively small errors, the proposed method remains closely aligned with QEM while maintaining a clear advantage over FQMS. These results indicate that neural-extracted features more effectively guide the edge-collapse sequence, thereby preserving the global geometric structure more faithfully.

Normal consistency P_{NE} further demonstrates a stable advantage across all simplification levels. At a ratio of 0.05, the proposed method achieves 0.1431 ± 0.0154, noticeably lower than QEM (0.1494 ± 0.0166) and FQMS (0.1630 ± 0.0168). This superiority persists at ratios of 0.1 and 0.2, confirming that the method is more effective at preserving local surface orientations and reducing directional distortions during simplification.

Finally, concerning global structural preservation as reflected by P_{LE}, all methods exhibit highly consistent behavior across different simplification ratios, particularly under medium and high compression levels where differences become virtually negligible. This indicates that while the proposed method significantly improves local geometric fidelity (P_{CD} and P_{NE}), it does so without disrupting the global structural consistency of the original mesh. Consequently, the proposed framework achieves a highly favorable balance between local geometric precision and global structural stability.

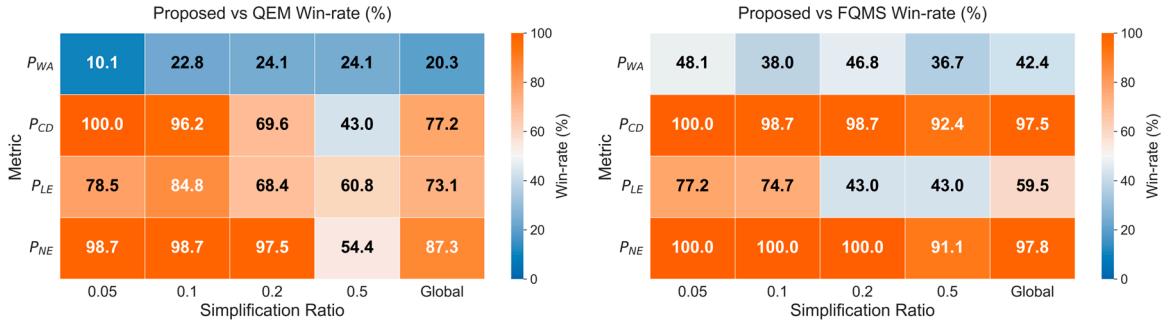
Regarding simplification time, a strictly fair evaluation against C++-based baselines (QEM/FQMS) is limited due to implementation differences. The baselines are highly optimized in C++, whereas our method is implemented in Python + PyTorch and integrates a Python-based QEM module coupled with a dynamic edge-collapse modulation mechanism. This introduces additional overhead in large-scale topological updates within the interpreter. In terms of GPU memory consumption, the proposed method shows moderate and stable usage across different models, ranging from 209.58 MB (Centaur) to 678.44 MB (Michael), indicating controllable memory overhead

 doi:10.20944/preprints202604.1809.v1

under varying geometric complexity. Future work will migrate the QEM module to optimized C++/CUDA implementations to enable more direct runtime comparisons and improve deployment efficiency.

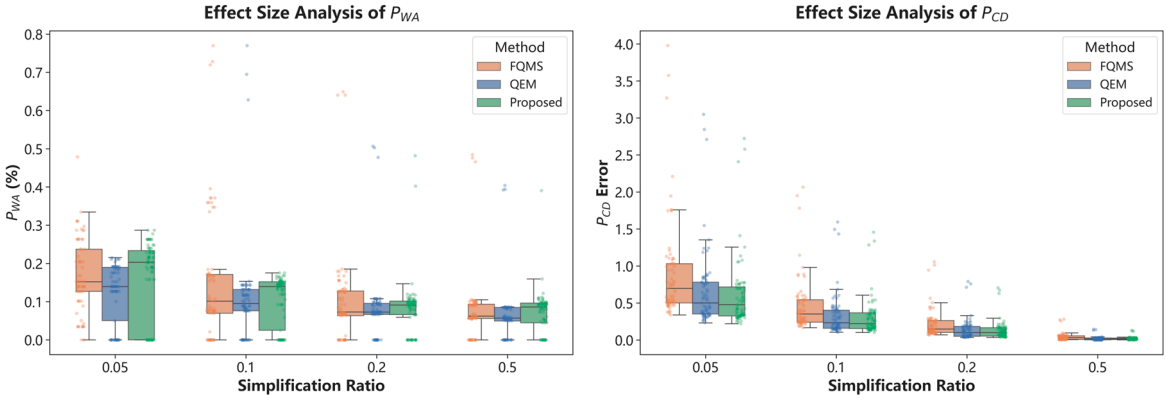
4.3.2. Win-Rates and Effect Size Analysis

To quantitatively evaluate the performance advantages of the proposed method, we computed its win-rate relative to two baseline methods, QEM and FQMS, across different simplification ratios and multiple geometric error metrics. The win-rate is defined as the percentage of test models for which the proposed method achieves a lower error on a given metric. This indicator provides an intuitive measure of the overall superiority of the proposed approach over the baselines. Furthermore, to address the potential limitation that win-rates alone might obscure the actual magnitude of improvements (i.e., effect sizes), we supplement the win-rate heatmaps with absolute error distribution boxplots. Figure 5 present heatmaps of the win-rate (%) for the proposed method relative to QEM (left) and FQMS (right) across varying simplification ratios and evaluation metrics. Figure 6 displays the boxplots of absolute error distributions for all evaluation metrics, providing a transparent view of the quantitative effect sizes.



(**a**) (**b**)

Figure 5. Heatmaps of the win-rate (%) for the proposed method relative to (**a**) QEM and (**b**) FQMS across varying simplification ratios and evaluation metrics. Values exceeding 50% indicate that the proposed method outperforms the corresponding baseline for the given metric.



(a) (b)

 doi:10.20944/preprints202604.1809.v1

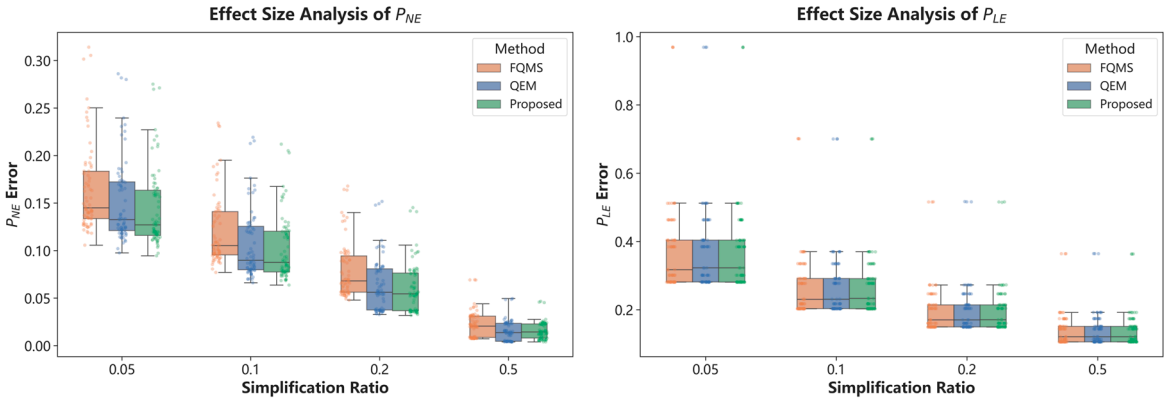


Figure 6. Boxplots illustrating the effect sizes (absolute error distributions) for (a) P_{WA}, (b) P_{CD}, (c) P_{NE}, and (d) P_{LE} across tested simplification ratios.

From the results, several key observations can be drawn:

(1) The proposed method maintains strong normal consistency and local geometric fidelity with significant effect sizes across most simplification ratios.

Experimental results show that, in terms of the P_{NE} metric, the proposed method achieves a win-rate of 90% against FQMS across all simplification ratios (0.05–0.5). Compared with QEM, the win-rate reaches 90% at low simplification ratios of 0.05 0.1, and 0.2, respectively, and remains above 50% even at medium and high ratios. More importantly, these high win rates are not only reflected in improved mean values but also correspond to a clear downward shift in the overall error distribution. The method reduces both the median normal deviation and its variance, effectively suppressing the extreme angular errors frequently observed in baseline approaches. This indicates that by leveraging GNNs to predict edge importance and assigning higher weights to high-curvature regions and sharp features during edge collapse, the algorithm preferentially preserves geometrically salient structures. As a result, normal deviation is effectively reduced, and the stability of local surface orientation is maintained throughout the simplification process, which aligns well with the design objective of the proposed method.

(2) The method shows clear advantages in geometric distance at low to moderate simplification ratios, while the advantage diminishes at higher ratios.

For the P_{CD} metric, the win-rate with respect to QEM reaches 100% at a simplification ratio of 0.05 but decreases to 96.2%, 69.6%, and 43.0% at ratios of 0.1, 0.2 and 0.5, respectively. In the P_{CD} boxplot, the proposed method demonstrates clear superiority under extreme simplification, while the performance gap narrows as the simplification ratio increases. This behavior can be attributed to the fact that QEM directly minimizes the quadratic error between vertices and planes, inherently favoring the global minimization of geometric distance. Consequently, under moderate simplification conditions, QEM tends to achieve smaller P_{CD} values. In contrast, the proposed method predicts edge importance using GNNs and applies a soft modulation to the collapse cost in the early stages of simplification, thereby prioritizing the preservation of geometrically salient regions. This strategy significantly improves local geometric fidelity at low to moderate simplification ratios. Moreover, the method maintains a win-rate of 92%-100% against FQMS across all simplification ratios, indicating a clear advantage over this baseline in controlling overall geometric error. These results highlight the inherent trade-off between preserving local geometric features and minimizing global geometric distance.

(3) The method demonstrates robust structural preservation but exhibits certain limitations in topological watertightness.

The P_{LE} metric remains stable across all three methods. This indicates that all approaches are capable of preserving the overall structural characteristics of the mesh in most models. Since P_{LE} is derived from the Laplacian spectrum and reflects intrinsic structural properties of the mesh, these

 doi:10.20944/preprints202604.1809.v1

results suggest that key structural relationships are effectively maintained during simplification. However, the global average win-rates for the P_{WA} metric relative to QEM and FQMS are only 20.3% and 42.4%, respectively. It should be noted that P_{WA} measures the proportion of non-manifold edges and is used to evaluate the topological watertightness of the simplified mesh. As shown in the P_{WA} boxplot, the method performs comparably to the baselines at low and moderate simplification ratios, but exhibits a higher median under extreme simplification. These results indicate that, at very high compression levels, the proportion of boundary or non-manifold edges produced during simplification is higher than that of the baseline methods. This phenomenon may arise because the method prioritizes the preservation of geometrically salient regions, causing areas with lower geometric importance to undergo more aggressive collapses under high compression ratios, which can lead to local topological anomalies or degeneracies. It is also worth noting that P_{WA} is highly sensitive to local topological irregularities; even a small number of non-manifold edges can affect the statistic. Therefore, this metric primarily reflects changes in local topological consistency during simplification rather than directly corresponding to overall geometric error.

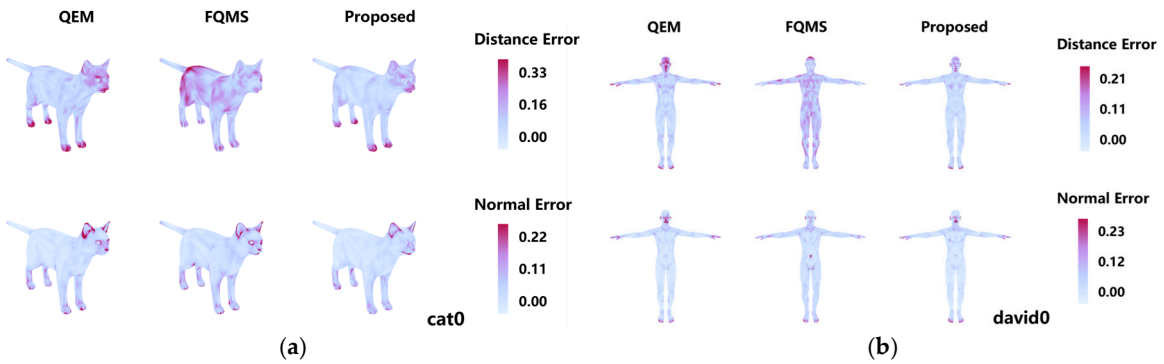
(4) The proposed method achieves an effective balance between local structure preservation and global geometric fidelity.

Considering all evaluation metrics, the global average win-rate indicates that the proposed method achieves the most significant improvements on P_{CD} and performs excellently on P_{NE}, while maintaining comparable P_{LE} performance across all methods. Conversely, performance on P_{WA} remains solid under moderate compression ratios and performs comparably to QEM on geometrically flatter models (e.g., the dog model), but exhibits a relative decline under extreme ratios. This reflects the inherent topological trade-off discussed earlier, where restricting operations in feature-rich areas forces the algorithm to concentrate allowable edge collapses in geometrically flat regions, occasionally impacting local topological watertightness under extreme compression. From a results viewpoint, this metric distribution further validates the effectiveness of the GNN-guided soft modulation strategy. The superior P_{CD} (Chamfer distance) and P_{NE} (normal consistency) indicate that the strategy successfully prioritizes the preservation of high-scoring feature edges, thereby effectively reducing global geometric error while retaining geometrically salient structures. Concurrently, the stable P_{LE} performance further demonstrates that enhancing local feature protection does not compromise the overall structural preservation capability.

Overall, the method demonstrates strong capability in preserving local geometric structures while maintaining geometric distance errors comparable to those of the classical QEM method in most cases. This suggests that incorporating learned geometric importance into the traditional QEM simplification framework enables an effective balance between structural preservation and global geometric fidelity without altering its fundamental optimization mechanism, thereby providing a practical approach for integrating learning-based techniques with classical geometric simplification algorithms.

4.3.3. Error Fields

To further investigate the spatial distribution of errors produced by different simplification methods, error fields were generated for results at a simplification ratio of 0.1, as shown in Figure 7.



 doi:10.20944/preprints202604.1809.v1

Figure 7. Error fields of mesh simplification results for different models of (**a**) cat0, (**b**) david0, (**c**) dog3, and (**d**) centaur2 at a simplification ratio of 0.1. In each model, the first row illustrates the geometric distance error, while the second row presents the normal deviation. The color map transitions from white to deep purple, representing increasing error magnitude.

From the perspective of spatial distribution, clear differences can be observed among the methods. FQMS tends to produce concentrated high-error regions in areas with high curvature or complex structures. For instance, in the dog3 model, prominent deep-purple regions appear around the limb joints and facial features, indicating noticeable local geometric deviations. This pattern is consistent in both distance error and normal error fields. The quantitative metrics further support this observation: for the dog3 model, the P_{CD} value of FQMS is 0.461, whereas the proposed method reduces it to 0.333. Meanwhile, P_{NE} decreases from 0.144 to 0.131. The reduction in the area of high-error regions observed in error fields aligns with these numerical improvements.

In comparison, the QEM method exhibits a more uniform overall error distribution, although certain local accumulations of error remain in detailed regions. For example, in the cat0 and centaur2 models, the error maps of QEM do not show extreme error peaks, but relatively elevated error regions still spread around structural details. Correspondingly, the P_{CD} value of QEM on the cat0 model is 0.181, slightly higher than 0.174 obtained by the proposed method. On the centaur2 model, the proposed method reduces P_{CD} from 0.779 to 0.762, while P_{NE} decreases from 0.176 to 0.168. Supported by these significant numerical improvements, the error fields reveal that the spatial distribution of high-error regions becomes more concentrated and smoother. A more pronounced improvement can be observed on the david0 model. The proposed method reduces P_{CD} from 0.167 (FQMS) to 0.102(outperforming QEM's 0.106), while the normal error decreases from 0.081 to 0.064(surpassing QEM's 0.066). The corresponding error fields show a clear reduction of high-error areas and a more continuous overall error distribution.

Overall, rather than merely maintaining a comparable global error level to QEM, the proposed method achieves superior quantitative accuracy and effectively mitigates the concentration of errors in local regions, leading to a smoother and more evenly distributed spatial error pattern. This indicates that the proposed structure-aware strategy can allocate geometric errors more effectively during the simplification process, preserving key structural features while maintaining overall geometric fidelity.

5. Discussions

The comprehensive experimental results and multi-metric evaluations indicate that the proposed method achieves clear advantages in the P_{CD}, P_{NE}, and P_{LE} metrics, demonstrating the effectiveness of the GNN-guided structure-aware strategy in capturing complex geometric features. In this framework, GNNs are employed as a structural importance estimator. By incorporating Laplacian positional encoding and a dual-branch message-passing architecture, the model is able to simultaneously capture fine-grained 1-hop geometric details and broader 2-hop topological context, thereby enhancing its ability to perceive structural characteristics. While the classical QEM algorithm performs well in minimizing global geometric distance, its inherently greedy optimization based on

 doi:10.20944/preprints202604.1809.v1

point-to-plane distances often overlooks semantically important regions with high curvature during simplification. In contrast, the proposed method embeds the learned importance weights into the QEM edge collapse cost through a soft modulation strategy, introducing geometry-saliency driven structural awareness without altering the fundamental optimization framework of QEM. This design significantly improves the identification and preservation of complex geometric structures. Experimental results confirm consistent advantages across multiple metrics. For example, at a simplification ratio of 0.05, the proposed method achieves a P_{NE} value of 0.1431, outperforming QEM (0.1494) and FQMS (0.1630), indicating superior preservation of surface normal consistency. Meanwhile, the proposed method maintains geometric errors comparable to QEM across different simplification levels. At a ratio of 0.05, the P_{CD} value is 0.6043, lower than QEM (0.6632) and significantly better than FQMS (0.8936), demonstrating that the learning-guided strategy effectively improves simplification quality while preserving geometric accuracy.

Regarding the dynamic regulation mechanism, the proposed progress-aware threshold function $G(t)$, together with the staged inference strategy, plays a critical role. This mechanism enables a smooth transition from early-stage feature preservation to late-stage global error optimization. The effectiveness of this design is further supported by the win-rate analysis, where the proposed method achieves strong advantages across most metrics. Specifically, the average win-rate with respect to QEM exceeds 70% for both P_{LE} and P_{NE}, while comparisons with FQMS show near 100% win-rates on P_{NE} and P_{CD}. These results indicate that the method is highly effective in preserving overall geometric structure. Moreover, the staged update of graph features helps mitigate topological drift during simplification, ensuring stable normal consistency even under high compression ratios. In terms of spatial error distribution, the proposed method produces smoother and more continuous error fields, effectively reducing local error concentrations and yielding simplified meshes that more closely follow the original geometric contours, particularly in high-curvature regions where more natural structural transitions are preserved.

The relatively weaker performance in the P_{WA} metric reflects a fundamental trade-off between strict feature preservation and uniform error distribution. By explicitly protecting high-curvature structures and enforcing rigorous topological checks (e.g., the Link Condition), edge collapses under high compression are inevitably concentrated in geometrically flat regions, causing localized error accumulation. Consequently, while our method preserves sharp features significantly better than QEM on complex models, it sacrifices some P_{WA} performance. This effect naturally weakens and may even vanish on simple geometric shapes (e.g., the dog model). Ultimately, this represents a controlled compromise for structural fidelity, while future work will further reduce GNN inference overhead and explore the incorporation of non-geometric attributes such as texture.

6. Conclusions

We propose a triangular mesh simplification method guided by structure-aware GNNs, which integrates learned structural importance with the classical QEM. This approach enhances the preservation of critical structural features while maintaining overall geometric stability. Overall, the proposed method demonstrates consistent performance across multiple dimensions, including normal consistency, global structural integrity, and geometric error control, albeit with an inherent trade-off regarding local topological manifoldness. The main conclusions are as follows:

(1) GNN-guided QEM hybrid mesh simplification framework enables interpretable fusion of data-driven structural awareness and classical geometric optimization.

(2) Spectral geometry and multi-scale neighborhood-based structural representation network improves recognition of complex geometric structures.

(3) Dynamic cost soft modulation and staged inference achieve a balance between feature preservation and error control.

Despite the demonstrated effectiveness, several limitations remain for future investigation:

(1) While the overall effectiveness of the proposed framework has been demonstrated, future work will include a systematic ablation study of its key components, namely the dual-branch

 doi:10.20944/preprints202604.1809.v1

architecture, dynamic soft modulation, and staged inference, to rigorously isolate their individual contributions.

(2) Although staged inference reduces computation, each stage still requires graph reconstruction and GNN estimates. Furthermore, compared with purely geometric baselines that are highly optimized in C++, our current Python-based implementation introduces additional interpreter overhead during large-scale dynamic topological updates. This combined overhead may limit scalability to very large meshes. To improve overall efficiency and deployment capability, future work will migrate the core Python-based QEM module to optimized C++/CUDA implementations. In parallel, future research could also explore lightweight network architectures, graph sampling strategies, or incremental update mechanisms to improve efficiency.

(3) The current structural importance prediction relies mainly on geometric attributes such as dihedral angles, edge lengths, and topology, without explicitly incorporating texture, color, or semantic information. For models with complex appearance (e.g., 3D geological models, scanned cultural heritage objects), such information is crucial for visual fidelity. Future work could integrate texture features, semantic labels, or multimodal cues to achieve a more comprehensive, structure-aware mesh simplification.

Author Contributions: B.Z.: conceptualization, validation, writing—review and editing, and funding acquisition; X.Y.: software, data curation, and writing—original draft preparation; W.C.: methodology, software, and writing—original draft preparation; X.Z.: software; B.W.: writing—original draft preparation; T.Z.: investigation, data curation, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by grants from the National Science and Technology Major Project of Deep Earth Probe and Mineral Resources Exploration, China (Grant No. 2024ZD1001201), the National Natural Science Foundation of China (Grant No. 42572387), the Scientific Research Project of Geological Bureau of Hunan Province, China (Grant No. HNGSTP202301), and the Research Project of Geological and Geographic Information Institute of Hunan Province, China (Grant No. HNDX2024001).

Data and Code Availability Statement: The source code for the GNN-QEM is available in Python at Github (https://github.com/Geo3D-AI-CSU/GNN-QEM, Zhang et al., 2025), and the TOSCA dataset is available at OpenDataLab (https://opendatalab.com/OpenDataLab/TOSCA, accessed on 27 January 2026).

Acknowledgments: We thank SHI Yu-Zheng (Geological and Geographic Information Institute of Hunan Province) and ZOU Bin (Central South University) for their kind assistance in the data collection. Special thanks are extended to the MapGIS Laboratory Co-Constructed by the National Engineering Research Center for Geographic Information System of China and Central South University for providing the MapGIS® V10.7 software (Wuhan Zondy Cyber-Tech Co., Ltd., Wuhan, China).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Potamias, R.A.; Ploumpis, S.; Zafeiriou, S. Neural mesh simplification. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022; pp. 18583-18592.
2. Zhang, B.; Zhu, Y.; Zhang, T.; Zhou, X.; Wang, B.; Kablan, O.A.B.K.; Huang, J. Three-Dimensional Stratigraphic Structure and Property Collaborative Modeling in Urban Engineering Construction. *Mathematics* **2025**, *13*, 345.
3. Hoppe, H. Progressive meshes. In Proceedings of the Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 1996; pp. 99–108.
4. Biljecki, F.; Ledoux, H.; Stoter, J. An improved LOD specification for 3D building models. *Comput. Environ. Urban Syst.* **2016**, *59*, 25-37, doi:10.1016/j.compenvurbsys.2016.04.005.

 doi:10.20944/preprints202604.1809.v1

5. Garland, M.; Heckbert, P.S. Surface simplification using quadric error metrics. In Proceedings of the Proceedings of the 24th annual conference on Computer graphics and interactive techniques, 1997; pp. 209-216.
6. Lindstrom, P.; Turk, G. Fast and memory efficient polygonal simplification. In Proceedings of the Proceedings of the conference on Visualization '98, Research Triangle Park, North Carolina, USA, 1998; pp. 279–286.
7. Lee, C.H.; Varshney, A.; Jacobs, D.W. Mesh saliency. In *ACM SIGGRAPH 2005 Papers*; 2005; pp. 659-666.
8. Liu, H.T.D.; Gillespie, M.; Chislett, B.; Sharp, N.; Jacobson, A.; Crane, K. Surface Simplification using Intrinsic Error Metrics. *Acm Transactions on Graphics* **2023**, *42*, 17, doi:10.1145/3592403.
9. Gori, M.; Monfardini, G.; Scarselli, F. A new model for learning in graph domains. In Proceedings of the Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., 31 July-4 Aug. 2005, 2005; pp. 729-734 vol. 722.
10. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, April 24, 2017 - April 26, 2017, 2017.
11. Zhang, B.; Li, M.; Huan, Y.; Khan, U.; Wang, L.; Wang, F. Bedrock mapping based on terrain weighted directed graph convolutional network using stream sediment geochemical samplings. *Transactions of Nonferrous Metals Society of China* **2023**, *33*, 2799-2814, doi:10.1016/S1003-6326(23)66299-5.
12. Wang, L.; Jiang, Z.; Song, L.; Yu, X.; Yuan, S.; Zhang, B. A groundwater level spatiotemporal prediction model based on graph convolutional networks with a long short-term memory. *Journal of Hydroinformatics* **2024**, *26*, 2962-2979, doi:10.2166/hydro.2024.226.
13. Choi, J.; Shah, R.; Li, Q.; Wang, Y.; Saraf, A.; Kim, C.; Huang, J.-B.; Manocha, D.; Alsisan, S.; Kopf, J. Ltm: Lightweight textured mesh extraction and refinement of large unbounded scenes for efficient storage and real-time rendering. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024; pp. 5053-5063.
14. Takikawa, T.; Litalien, J.; Yin, K.; Kreis, K.; Loop, C.; Nowrouzezahrai, D.; Jacobson, A.; McGuire, M.; Fidler, S. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In Proceedings of the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021; pp. 11358-11367.
15. Garland, M.; Zhou, Y. Quadric-based simplification in any dimension. *Acm Transactions on Graphics* **2005**, *24*, 209-239, doi:10.1145/1061347.1061350.
16. Lindstrom, P. Out-of-core simplification of large polygonal models. In Proceedings of the Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 2000; pp. 259-262.
17. Hoppe, H. New quadric metric for simplifying meshes with appearance attributes. In Proceedings of the Proceedings Visualization'99 (Cat. No. 99CB37067), 1999; pp. 59-510.
18. Song, R.; Liu, Y.H.; Martin, R.R.; Rosin, P.L. Mesh Saliency via Spectral Processing. *Acm Transactions on Graphics* **2014**, *33*, 17, doi:10.1145/2530691.
19. Xu, R.; Liu, L.; Wang, N.; Chen, S.; Xin, S.; Guo, X.; Zhong, Z.; Komura, T.; Wang, W.; Tu, C. CWF: consolidating weak features in high-quality mesh simplification. *ACM Transactions on Graphics (TOG)* **2024**, *43*, 1-14.
20. Heep, M.; Behnke, S.; Zell, E. Feature-Preserving Mesh Decimation for Normal Integration. In Proceedings of the Proceedings of the Computer Vision and Pattern Recognition Conference, 2025; pp. 5783-5792.
21. Hanocka, R.; Hertz, A.; Fish, N.; Giryes, R.; Fleishman, S.; Cohen-Or, D. MeshCNN: A Network with an Edge. *Acm Transactions on Graphics* **2019**, *38*, 12, doi:10.1145/3306346.3322959.
22. Lan, J.M.; Zeng, B.; Li, S.Q.; Zhang, W.H.; Shi, X.Y. A Deep Learning-Based Salient Feature-Preserving Algorithm for Mesh Simplification. *CMC-Comput. Mat. Contin.* **2025**, *83*, 2865-2888, doi:10.32604/cmc.2025.060260.
23. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 5115-5124.

24. Fey, M.; Lenssen, J.E.; Weichert, F.; Müller, H. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; pp. 869-877.
25. 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)* **2019**, *38*, 1-12.
26. Milano, F.; Loquercio, A.; Rosinol, A.; Scaramuzza, D.; Carlone, L. Primal-dual mesh convolutional neural networks. *Advances in Neural Information Processing Systems* **2020**, *33*, 952-963.
27. Sharp, N.; Attaiki, S.; Crane, K.; Ovsjanikov, M. DiffusionNet: Discretization Agnostic Learning on Surfaces. *Acm Transactions on Graphics* **2022**, *41*, 16, doi:10.1145/3507905.
28. Chen, Y.-C.; Kim, V.; Aigerman, N.; Jacobson, A. Neural progressive meshes. In Proceedings of the ACM SIGGRAPH 2023 Conference Proceedings, 2023; pp. 1-9.
29. Liang, Y.; Zhao, S.; Yu, B.; Zhang, J.; He, F. Meshmae: Masked autoencoders for 3d mesh data analysis. In Proceedings of the European conference on computer vision, 2022; pp. 37-54.
30. Liu, H.-T.D.; Kim, V.G.; Chaudhuri, S.; Aigerman, N.; Jacobson, A. Neural subdivision. *arXiv preprint arXiv:2005.01819* **2020**.
31. Guillard, B.; Remelli, E.; Lukoianov, A.; Yvernay, P.; Richter, S.R.; Bagautdinov, T.; Baque, P.; Fua, P. DeepMesh: Differentiable Iso-Surface Extraction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 7072-7087, doi:10.1109/tpami.2024.3392291.
32. Rakotosaona, M.J.; Aigerman, N.; Mitra, N.J.; Ovsjanikov, M.; Guerrero, P. Differentiable Surface Triangulation. *Acm Transactions on Graphics* **2021**, *40*, 13, doi:10.1145/3478513.3480554.
33. Shen, T.; Munkberg, J.; Hasselgren, J.; Yin, K.; Wang, Z.; Chen, W.; Gojcic, Z.; Fidler, S.; Sharp, N.; Gao, J. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)* **2023**, *42*, 1-16.
34. Son, S.; Gadelha, M.; Zhou, Y.; Xu, Z.; Lin, M.C.; Zhou, Y. Dmesh: A differentiable representation for general meshes. *CoRR* **2024**.
35. Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Harutyunyan, H.; Alipourfard, N.; Lerman, K.; Steeg, G.V.; Galstyan, A.G. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In Proceedings of the International Conference on Machine Learning, 2019.
36. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 2017; pp. 5105–5114.
37. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the International conference on machine learning, 2017; pp. 1263-1272.
38. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* **2018**.
39. Dwivedi, V.P.; Joshi, C.K.; Luu, A.T.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking graph neural networks. *Journal of Machine Learning Research* **2023**, *24*, 1-48.
40. Chung, F.R.K. *Spectral graph theory*; American Mathematical Soc.: 1997; Volume 92.
41. Bronstein, A.M.; Bronstein, M.M.; Kimmel, R. *Numerical geometry of non-rigid shapes*; Springer Science & Business Media: 2008.
42. Zhou, Q.-Y.; Park, J.; Koltun, V. Open3D: A modern library for 3D data processing. *arXiv preprint arXiv:1801.09847* **2018**.
43. Forstmann, S. Fast Quadric Mesh Simplification. Available online: https://github.com/sp4cerat/Fast-Quadric-Mesh-Simplification (accessed on 27 January 2026).
44. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv* **2019**, *1903.02428*.
45. Reuter, M.; Wolter, F.-E.; Peinecke, N. Laplace–Beltrami spectra as 'Shape-DNA'of surfaces and solids. *Computer-Aided Design* **2006**, *38*, 342-366.