Global Discrete Optimization for Quad-Dominant Mesh Reduction

YUZHE LUO*, State Key Lab of CAD&CG, Zhejiang University, China and LIGHTSPEED, China
JINGCHEN GAO*, Hangzhou Dianzi University, China
ZHERONG PAN, META, USA
KUI WU, LIGHTSPEED, USA
XUEBO JI, The University of Hong Kong, China
XIAOGANG JIN†, State Key Lab of CAD&CG, Zhejiang University, China
XIFENG GAO†, LIGHTSPEED, USA

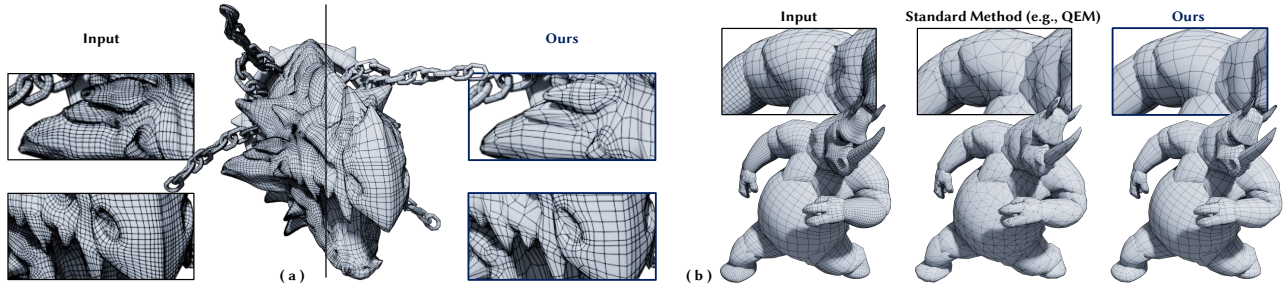

Fig. 1. High-fidelity quad-dominant mesh simplification. (a) Our method simplifies complex dense meshes (Input, 68k faces) to highly coarse versions (Ours, 14k faces) while strictly preserving the original quad-dominant topology and sharp feature flow (see close-ups). (b) Compared to standard methods (e.g., QEM), our results maintain high structural integrity, ensuring artifact-free deformations in skeletal animation even at 20% of the original resolution.

We present a global, discrete optimization framework for quad-dominant mesh reduction that formulates simplification as an edge selection problem under coupled topological and geometric constraints. Unlike greedy edge-collapse heuristics, which make irreversible local decisions, and rigid poly-chord removal schemes, which over-constrain global structure, our method casts quad-dominant reduction as an Integer Linear Programming (ILP) problem that jointly optimizes edge flow continuity and geometric fidelity in a single global optimization. To make this formulation both tractable and adaptive, we segment poly-chords at geometrically meaningful breakpoints to relax otherwise rigid topological constraints, and introduce geometry-aware soft constraints that steer the optimization toward preserving high-curvature and deformation-critical regions. Extensive experiments demonstrate that our approach consistently outperforms state-of-the-art methods in preserving quad structure and edge flow, while achieving competitive geometric accuracy and significantly benefiting downstream editing applications.

CCS Concepts: • **Computing methodologies** → **Shape modeling**.

*Both authors contributed equally to this research.
†Corresponding authors.

Authors' addresses: Yuzhe Luo, yzluo@zju.edu.cn, State Key Lab of CAD&CG, Zhejiang University, China and LIGHTSPEED, China; Jingchen Gao, lucasgao0927@qq.com, Hangzhou Dianzi University, China; Zherong Pan, zherong.pan.usa@gmail.com, META, USA; Kui Wu, walker.kui.wu@gmail.com, LIGHTSPEED, USA; Xuebo Ji, xueboji.cs@gmail.com, The University of Hong Kong, China; Xiaogang Jin, jin@cad.zju.edu.cn, State Key Lab of CAD&CG, Zhejiang University, China; Xifeng Gao, gxf.xisha@gmail.com, LIGHTSPEED, USA.

SIGGRAPH Conference Papers '26, July 19–23, 2026, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811043

ACM Reference Format:
Yuzhe Luo, Jingchen Gao, Zherong Pan, Kui Wu, Xuebo Ji, Xiaogang Jin, and Xifeng Gao. 2026. Global Discrete Optimization for Quad-Dominant Mesh Reduction. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3799902.3811043

1 INTRODUCTION

Automated mesh simplification is a ubiquitous component in computer graphics, serving as the backbone for generating hierarchical level-of-detail (LOD) models in real-time rendering pipelines. Industry-standard algorithms—most notably Quadric Error Metrics (QEM) [Garland and Heckbert 1997]—and their modern implementations in tools like Unreal Engine [Epic Games 2022] or Simplygon [Donya Labs AB 2022] excel at minimizing geometric deviation. These methods allow massive assets to be rendered efficiently by aggressively collapsing edges based on local error measurements.

However, assets manually crafted by artists are predominantly quad-dominant meshes, characterized by structured quad layouts with coherent edge flows and well-aligned edge loops. When generating LODs, artists expect the simplification process to maintain the integrity of this underlying quad topology. Preserving clean edge loops and orthogonal flows is essential for a wide range of downstream editing workflows, including local retopology, UV unfolding, texturing, skeletal animation, and physics simulation. Existing automated methods, driven largely by local geometric error metrics, often fail to meet these expectations. Being inherently myopic, these methods tend to destroy global quad structures, resulting in irregular triangulations that are ill-suited for production pipelines. Consequently, outputs from standard tools often force artists to perform

extensive manual retopology, creating a significant bottleneck in production workflows.

Existing solutions face a fundamental dichotomy between *topological integrity* and *geometric fidelity*. On one end of the spectrum, poly-chord methods [Daniels et al. 2008] treat quad topology as a rigid network of continuous strips. While this strategy preserves global structure, it suffers from severe "locking" issues: removing a long chord to simplify a flat region may cause unacceptable distortion in distant high-curvature regions. Conversely, local heuristics [Garland and Heckbert 1997] prioritize geometric error minimization. While recent variations attempt to preserve quads through local checks or hybrid pipelines [Knodt 2025; Mason 2020], they fundamentally lack a global perspective. As quad topology is inherently a global property, a sequence of locally valid edge collapses can inadvertently destroy long-range edge loops or create singularities that are impossible to resolve later.

We argue that the inability of current methods to reconcile these competing objectives stems from their underlying decision-making paradigm. These greedy algorithms [Garland and Heckbert 1997; Knodt 2025] are inherently sub-optimal due to myopic local decisions, while strict structural preservation [Daniels et al. 2008] is overly rigid and insensitive to geometric variation. Quad-dominant mesh simplification therefore involves making decisions that affect the global topology, where the preservation or removal of an edge simultaneously affects poly-chord continuity, transverse edge loops, and local surface approximation.

To bridge this gap, we propose a novel framework that reformulates quad-dominant simplification not as a sequence of local operations, but as a global optimization based on Integer Linear Programming (ILP). Instead of making myopic local decisions, our method determines the optimal subset of edges to retain in a single global solve, simultaneously balancing topological continuity and geometric error. In this formulation, topological constraints are enforced as hard constraints, while geometric objectives are incorporated as soft penalties, allowing structural requirements to be relaxed only where geometrically necessary. To make this global formulation both tractable and adaptive, we introduce two technical innovations:

(1) **Partial poly-chord segmentation** relaxes the rigidity of global poly-chords by segmenting them at geometrically meaningful breakpoints, enabling the ILP solver to collapse portions of a poly-chord while preserving poly-chord continuity and the associated topological constraints.
(2) **Geometry-aware soft constraints** incorporate curvature-sensitive information into the optimization, guiding the solver to retain sufficient edge density in high-curvature regions without enforcing overly restrictive hard constraints.

We validate our approach on a diverse dataset of artist-created quad-dominant assets. Quantitative experiments demonstrate that our method significantly outperforms state-of-the-art techniques in preserving quad topology and edge flow, while maintaining competitive geometric accuracy measured by Hausdorff and Chamfer distances. Qualitative feedback from professional artists further confirms that our output better preserves the modeling intent of the input model, thus facilitating various downstream applications.

2 RELATED WORK

Our work intersects three closely related research areas: general mesh reduction, structure-aware quad mesh simplification, and global optimization methods for discrete mesh topology. While each of these areas has been extensively studied, existing approaches are fundamentally limited by either myopic local decision-making or overly rigid global constraints, motivating the need for a more flexible global formulation.

2.1 General Mesh Reduction

Mesh reduction has been extensively studied for decades [Luebke et al. 2002], with the primary goal of reducing complexity while minimizing geometric deviation. A diverse array of methodologies has been proposed, including methods based on primitive clustering [Cohen-Steiner et al. 2004; Lindstrom 2000; Low and Tan 1997; Rossignac and Borrel 1993; Xu et al. 2024], manifold envelope extraction [Chen et al. 2023; Gao et al. 2022], and differentiable rendering optimization [Hasselgren et al. 2021; Knodt et al. 2023]. The seminal Quadric Error Metric (QEM) introduced by Garland and Heckbert [1997] revolutionized the field by compacting geometric history into a 4×4 matrix, allowing for efficient and high-quality iterative edge collapses. This framework was later extended to handle attributes such as UV coordinates, vertex colors, and other appearance attributes [Garland and Heckbert 1998; Hoppe 1999]. Further refinements have addressed reduction quality through improved quadric placement [Garland and Zhou 2005; Liu et al. 2023, 2025; Trettner and Kobbelt 2020]. The Quadrics-based method is also used for specific requirements such as animation mesh reduction [DeCoro and Rusinkiewicz 2005; Landreneau and Schaefer 2009].

Despite their success, these approaches are inherently *myopic* and geometry-driven: they rely on local collapse operators guided by geometric error metrics, without explicit mechanisms for preserving global topological structures. As a result, when applied to quad-dominant meshes, such methods tend to disrupt coherent edge flows and convert structured quad layouts into irregular triangulations. As observed in character modeling and rigging workflows [Raitt and Minter 1998], these irregular topologies often exhibit unstable behavior under skinning and articulated deformation. In contrast, our approach retains the geometric accuracy benefits of quadrics-based error measurement, while explicitly constraining the simplification process to preserve the global quad structure and its associated edge flow. Rather than treating the mesh as an unstructured collection of triangles, we formulate quad-dominant mesh reduction as a global optimization problem that respects topological continuity throughout the simplification process.

2.2 Structure-Aware Quad Reduction

Maintaining the specific connectivity of quad-dominant meshes requires strategies that prioritize topological invariants such as edge loops and poly-chords. Extensive work has focused on the extraction and generation of quadrilateral meshes [Campen et al. 2012; Campen and Kobbelt 2014; Dielen et al. 2021; Dong et al. 2025; Jakob et al. 2015; Marcias et al. 2015; Palmer et al. 2024; Pietroni et al. 2021; Tarini et al. 2011; Usai et al. 2015], as well as the specialized reduction of pure quad meshes [Bozzo et al. 2010; Daniels et al. 2008; Tarini et al.

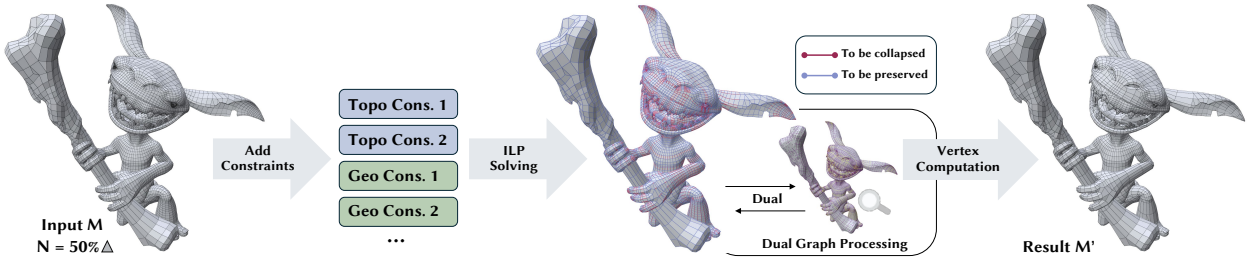

Fig. 2. Our pipeline: Given an input quad-dominant mesh and a target face count, we construct geometric and topological constraints to determine the optimal edge selection via an ILP solver. Following dual graph post-processing and vertex position calculation, we generate the final simplified mesh.

2010]. Related remeshing approaches [Daniels et al. 2009b; Panozzo et al. 2011] produce high-quality quad meshes but fundamentally aim at remeshing rather than topology-preserving simplification of the input connectivity. Recognizing that quad topology is defined by global edge loops, Daniels et al. [2008] proposed reducing mesh complexity by removing entire poly-chords, which guarantees all-quad output but enforces a rigid *all-or-nothing* constraint that often leads to the well-known *locking* problem. Daniels et al. [2009a] later proposed a localized variant with attribute-aware weighting, which mitigates locking but remains a sequential greedy approach. The Frostbite Engine [Mason 2020] employs a hybrid strategy permitting partial poly-chord reduction by locally introducing triangles, but relies on undocumented heuristics without global consistency guarantees. More recently, Knodt [2025] adapted the single edge-collapse operator for quad-dominant reduction by incorporating per-edge quadrics. Nevertheless, such greedy approaches remain inherently myopic and can inadvertently disrupt long-range loop continuity. Our approach formulates quad-dominant mesh reduction as a global ILP problem, jointly optimizing geometric objectives and topological constraints to avoid the failure modes of both rigid poly-chord removal and myopic sequential simplification.

2.3 Global Optimization using ILP

ILP has been widely used in computational geometry for problems requiring global discrete decisions on mesh topology. Bommes et al. [2009] pioneered its use for global singularity placement in quadrilateral remeshing, and Takayama et al. [2014] applied it to local patch quadrangulation. Duan et al. [2024] extended ILP to singularity structure simplification in hexahedral meshes. In quad patch layout optimization, ILP-based formulations have been employed for layout generation via graph matching [Razafindrazaka and Polthier 2017; Razafindrazaka et al. 2015], separatrix-based patch extraction [Zhang et al. 2016], and capacitive touch sensor placement [Palma et al. 2024]. While these works demonstrate the efficacy of ILP for global topology management, our method is the first to directly apply the constrained global ILP optimization paradigm to the problem of simplifying an existing quad-dominant surface mesh via optimal edge selection, thus overcoming the shortsightedness of local greedy algorithms and the rigidity of traditional global topological methods.

3 METHOD

Given an input quad-dominant mesh and a target face count, our method first discovers all poly-chords and segments them at geometrically meaningful breakpoints, while clustering loopy edges via QEM-based medial curve simplification to establish both topological and geometric constraints. These constraints are then unified within an ILP framework, which solves for the globally optimal subset of edges to retain. The resulting edge selection is refined through dual graph simplification to eliminate high-valence vertices, followed by a final QEM-based edge collapse that computes vertex positions and resolves remaining geometric degeneracies (Fig. 2).

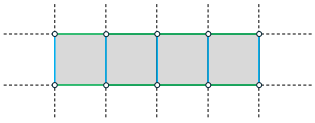

Fig. 3. Illustration of primal space edge loops and opposing quad edges.

3.1 Problem Formulation

As shown in Fig. 2, the input to our method involves a quad-dominant mesh $\mathcal{M} = (V, E, F)$ and a target triangular face count N, where we use V, E, F to denote the set of vertices, edges, and faces, respectively. Here we assume a quad-dominant mesh such that each $f \in F$ is bordering either 3 or 4 edges. Note that although our mesh might contain quads, we require the user to specify the target face (assuming triangular faces) count and we treat each quad face as two triangular faces. Our goal is to compute a simplified mesh $\mathcal{M}' = (V', E', F')$ that achieves the target face count as much as possible, while preserving the topological integrity and geometric fidelity. In terms of topological integrity, we intend to preserve the global quad layout, in particular the continuity of poly-chords and their dual orthogonal edge loops. In terms of geometric fidelity, we intend to preserve the fine surface details in high-curvature areas. We formulate quad-dominant mesh reduction as a global edge selection problem, in which each edge is treated as a binary decision variable indicating whether it is preserved or collapsed. All constraints and objectives are then defined over these variables within a unified optimization framework.

Preserving the topological integrity of a quad-dominant mesh requires maintaining its fundamental topological invariants: *poly-chords* [Daniels et al. 2008]. In the dual space, a poly-chord is defined

as a polyline connecting the centroids of adjacent quadrilaterals, which represents the continuous flow direction on the mesh surface. In the primal space, as illustrated in Fig. 3, the continuous sequences of edges that follow this flow direction are referred to as *edge loops* [Knodt 2025]. The edges orthogonal to these edge loops—specifically, the edges traversing the opposite sides of the quadrilaterals—are termed *opposing quad edges*. Maintaining the continuity of these elements is critical for preserving the global flow during simplification.

3.2 ILP Optimization Framework

Unlike local greedy heuristics, which may degrade global structure, we adopt a holistic formulation based on global discrete optimization. We introduce binary decision variables $x_e \in \{0,1\}$ for each edge $e \in E$ and $x_f \in \{0,1\}$ for each face $f \in F$, where a value of 1 indicates that the corresponding element is preserved in the simplified mesh. Under such a representation, we can estimate the number of faces after reduction, denoted as $N' = |F'|$, via the following equation:

$$N' \approx |F_{\text{tri}}| + 2|F_{\text{quad}}|, \tag{1}$$

where $F_{\text{tri}} = \{f \in F | x_f = 1 \wedge \sum_{e\in\partial f} x_e = 3\}$ and $F_{\text{quad}} = \{f \in F | x_f = 1 \wedge \sum_{e\in\partial f} x_e = 4\}$ denote the number of retained triangular and quad faces. Specifically, they correspond to the faces $f \in F$ such that the number of retained edges, i.e., $\sum_{e\in\partial f} x_e$, is equal to 3 and 4, respectively. Since we want the preservation of topological integrity, requiring $N' = N$ exactly would potentially lead to conflict with other topological constraints, leading to an infeasible ILP. Therefore, we allow users to provide a slack parameter ϵ_N and add the following constraint:

$$(1-\epsilon_n)N \le N' \le (1+\epsilon_n)N, \tag{2}$$

where we have found that setting $\epsilon_n = 0.03$ yields the best overall result. In this section, we introduce a set of hard constraints to preserve topological integrity and a set of soft penalty objectives to encourage geometric similarity.

3.2.1 Edge-Face Consistency Constraint. To prevent degenerate faces, we require that a preserved face must border either 3 vertices (a triangular face) or 4 vertices (a quad face). This is realized by the following constraints:

$$\sum_{e\in\partial f} x_e \ge 3x_f \quad \text{and} \quad \sum_{e\in\partial f} x_e - 2 \le 4x_f, \tag{3}$$

where the first constraint ensures that, if a face is preserved, the number of bordering edges is at least 3. On the other hand, if the face is not preserved, then the second constraint ensures that the number of bordering edges is at most 2.

3.2.2 Poly-chord Segmentation and Consistency Constraint. As illustrated in Fig. 4, an entire poly-chord often exhibits complex structures, such as self-intersections. We can collapse an entire poly-chord by removing all the opposing quad edges between two parallel poly-chords. However, this strict constraint often leads to infeasibility in the ILP formulation or over-simplifying that significantly degrades the geometric quality. To resolve this, we propose a heuristic segmentation strategy that decomposes poly-chords into

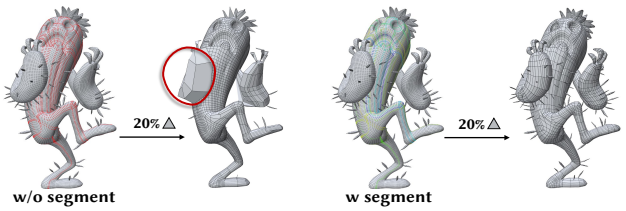

Fig. 4. Left: All the red edges on the model correspond to a single poly-chord, which is rather complicated and involves self-intersection. Requiring to preserve or discard the poly-chord altogether is too restrictive or leads to undesirable over-simplification (red circle) after face reduction. Right: We divide the poly-chord into segments as labeled by different colors, thus providing more flexibility for the ILP solver.

structurally coherent segments. We then enforce that all the opposing quad edges within each coherent segment must be retained or removed altogether.

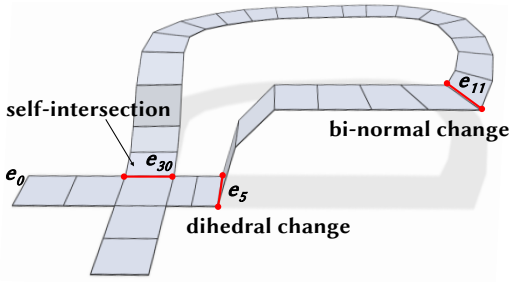

Fig. 5. Raw poly-chords are segmented at geometric breakpoints (red edges) according to one of three criteria, allowing independent simplification decisions for each segment.

Given a poly-chord with its corresponding opposing quad edges $P = \{e_0, e_1, \ldots, e_m\}$, we insert breakpoints based on three criteria:

- If the **dihedral angle** between adjacent faces exceeds a threshold ϵ_θ (typically $\pi/3$): $|\theta_{\text{dihedral}}(e_i)| > \epsilon_\theta$.
- If the **bi-normal change** between adjacent faces is larger than ϵ_θ: $|\theta_{\text{bi-normal}}(e_i)| > \epsilon_\theta$. Here we compute the bi-normal as the (normalized) cross product between the face normal vector and the direction of an edge in the edge loops.
- If a vertex incident to e_i has been previously visited by P, indicating a **self-intersection**, then e_i is designated as a split location.

Based on the detected breakpoints, P is partitioned into a disjoint union $P = \cup_{\text{seg}} P_{\text{seg}}$. We enforce the constraint that all opposing quad edges within a segment share the same preservation status:

$$x_e = x_{e'} \quad \forall e, e' \in P_{\text{seg}}. \tag{4}$$

This formulation, as visualized in Fig. 5, allows the solver to flexibly collapse redundant segments while preserving those critical to the global flow directions.

3.2.3 Geometric Similarity Constraints. In the previous section, we focus on ensuring that opposing quad edges are preserved and discarded in a topologically consistent manner. In this section, we turn our focus to the edges in the edge loops, or loopy edges. The direction changes of these loopy edges reflect the curvature of the

input mesh. Our goal is to retain a sufficient number of edges in high-curvature areas. To this end, our core idea is to utilize conventional QEM-based mesh simplification via edge collapse [Garland and Heckbert 1997]. Crucially, we do not use QEM to generate the final geometry directly; instead, we leverage the decimation history to define clustering constraints for the ILP. Specifically, we notice that an edge collapse can be interpreted as an edge clustering operator, where an edge to be collapsed is merged with a neighboring edge. We then utilize the QEM-based edge clusters to formulate our constraint and enforce that at least one edge must be retained within each cluster.

As illustrated in Fig. 6, we realize the above idea by first extracting the medial curve $\Gamma = \langle e_1, \cdots, e_m \rangle$ from the original unsegmented poly-chord between a pair of two loopy edge sequences, represented as an ordered set of medial edges, where each $e_i \in \Gamma$ connects the midpoints of two adjacent opposing quad edges. Each edge $e_i \in \Gamma$ is associated with a cluster $C(e_i)$ initialized to contain the two parallel loopy edges. Next, we conduct QEM-based edge clustering guided by the quadric error $Q(e_i)$, computed on the medial edges. We adopt the per-edge weighted quadric formulation proposed by Knodt [2025] to respect tangent plane constraints during collapse. For an edge $e_i = \langle v_i^0, v_i^1 \rangle$ bordering two vertices, the quadric error combines contributions from the surface deviation term and the in-plane distortion term:

$$Q(e_i) = w_n \cdot \mathbf{Q}(v_i^0, n_f(e_i)) + w_f \cdot \mathbf{Q}(v_i^0, \frac{v_i^0 - v_i^1}{\|v_i^0 - v_i^1\|_2} \times n_f(e_i)), \quad (5)$$

where $n_f(e_i)$ denotes the unit face normal associated with e_i, and $\mathbf{Q}$ represents the fundamental plane quadric error function [Trettner and Kobbelt 2020]. The first term penalizes deviation from the underlying surface, while the second term penalizes in-plane distortion orthogonal to the flow. Here we set $w_n = \|v_i^0 - v_i^1\|^2$, $w_f = 0.01$.

We greedily collapse edges in order of increasing $Q(e_i)$, using the quadric error solely as a priority metric. We then update the medial curve as $\Gamma \leftarrow \Gamma / \langle e_i \rangle$. Further, upon collapsing the edge e_i, its associated cluster $C(e_i)$ is merged with that of a neighboring $C(e_j)$. For an edge e_i on the medial axis, there can be two neighboring clusters, and we choose to merge with the cluster having smaller Euclidean distance between the center points of the edges. The collapse terminates when any of several geometric stopping criteria is violated, including spatial deviation from the original curve, normal deviation beyond a threshold angle, or excessive normal variation within a cluster (detailed in the supplementary material). Finally, notice that in order to compute $Q(e_i)$, we need to maintain its associated face normal $n_f(e_i)$. For an edge representing the center line of a quad, its face normal is the normal of the quad face. After an edge is collapsed, we update the face normal definition to be the average of the quad face normals within a cluster.

After clustering we are left with a reduced medial axis Γ such that for each $e_i \in \Gamma$, its associated cluster $C(e_i)$ represent a subset of original edges in the input mesh that represents a salient feature of the mesh, which should not be discarded altogether. We thus require that at least one of the edges within the cluster must be retained. However, enforcing this as hard constraints can yield infeasible ILP. Therefore, we formulate this requirement as a soft penalty via an

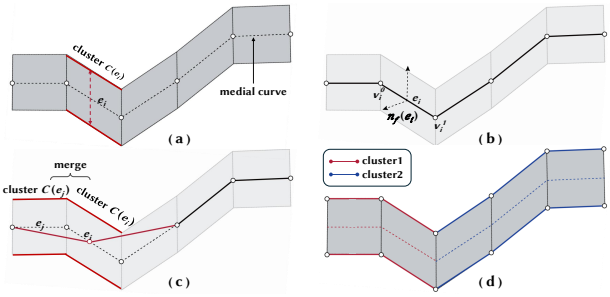

Fig. 6. Overview of the medial axis clustering. We extract the medial curve from the poly-chord (a) and calculate quadric errors for curve edges based on normal directions (b). An iterative edge collapse process is then performed to cluster edges (c). Finally, the clustered results will be used to establish piecewise geometric constraints (d).

auxiliary continuous variable $x_{C(e_i)} \in [0, 1]$ defined via the follow logic constraint:

$$\begin{aligned} x_{C(e_i)} &\leq 1 - x_e, \quad \forall e \in C(e_i), \\ x_{C(e_i)} &\geq 1 - \sum_{e \in C(e_i)} x_e. \end{aligned} \quad (6)$$

Specifically, $x_{C(e_i)} = 1$ iff all the edges in $C(e_i)$ is discarded. This triggers a high cost in the objective function (Section 3.2.5), strongly encouraging the preservation of at least one loopy edge within each region containing significant geometric features.

3.2.4 Geometric Similarity for Triangular Faces. Our method takes as input quad-dominant meshes that may contain triangular faces and therefore lack well-defined poly-chords. However, we can use a similar idea to enforce geometric similarity constraints with the help of QEM-based edge collapsing. Specifically, we first find a set of connected edges $\Gamma_t = \{e_1, \cdots, e_m\}$ bordering triangular faces using breadth-first search (BFS). Each $e_i \in \Gamma_t$ is again associated with a cluster $C_t(e_i)$ initialized to be $\{e_i\}$. Note that here e_i refers to primal mesh edges adjacent to triangular faces, and unlike Section 3.2.3, the quadric error is computed using the standard face-based QEM from adjacent triangle plane equations, rather than the per-edge formulation of Eq. 5. We greedily collapse edges in order of increasing $Q(e_i)$, terminating when the QEM cost exceeds a threshold ϵ_{tri} proportional to the squared bounding box diagonal (see supplementary material). Upon each collapse, we update $\Gamma_t \leftarrow \Gamma_t / \{e_i\}$ and $C_t(e_i)$ is combined with a neighboring $C_t(e_j)$ with smallest Euclidean distance in edge center. After the collapse, we introduce the interleaved cluster activation constraint. Specifically, if any edge in a cluster is discarded, then a neighboring cluster must be entirely preserved. To this end, we introduce the continuous variable $x_{C_t(e_i)} \in [0, 1]$ defined via the following logic constraint:

$$\begin{aligned} \sum_{e \in C_t(e_i)} x_e + |C_t(e_i)| \cdot x_{C_t(e_i)} &\geq |C_t(e_i)|, \\ x_{C_t(e_i)} &\leq |C_t(e_i)| - \sum_{e \in C_t(e_i)} x_e. \end{aligned} \quad (7)$$

Specifically, $x_{C_t(e_i)} = 1$ iff at least one edge in cluster $C_t(e_i)$ is discarded: the first constraint forces $x_{C_t(e_i)} \geq 1$ whenever any $x_e = 0$, while the second forces $x_{C_t(e_i)} = 0$ when all edges are

preserved. If this is the case, the following constraint ensures that all edges are preserved in the neighboring cluster $C_t(e_j)$:

$$|C_t(e_j)|x_{C_t(e_i)} \leq \sum_{e \in C_t(e_j)} x_e. \tag{8}$$

3.2.5 Objective Function. Our solver seeks an assignment of variables that satisfies all hard constraints while minimizing a global objective $\mathcal{L}$. We formulate $\mathcal{L}$ to balance three competing goals: adhering to the target face count, minimizing the violation of geometric soft constraints, and maximizing the retention of high-importance edges. The objective function is defined as:

$$\mathcal{L} = \underbrace{|N' - N|}_{\text{Target Precision}} + \underbrace{\lambda_{\text{geo}} \sum_{C(e_i)} x_{C(e_i)}}_{\text{Geometric Similarity}} - \underbrace{\lambda_{\text{edge}} \sum_{e \in E} Q(e) x_e}_{\text{Edge Importance}}. \tag{9}$$

Here, the first term acts as a soft regularizer to guide the solution towards the exact target count. The second penalty term discourages the total collapse of significant geometric clusters by associating high costs when some $x_{C(e_i)} = 1$. Our third term encourages an error-weighted edge preservation scheme. Minimizing the negative sum favors the preservation of edges with large geometric errors, thereby maintaining high-frequency geometric features.

3.3 Post-processing

While the primal ILP formulation enforces a series of hard constraints to decide which primal edges need to be preserved, it operates without explicit awareness of the dual graph structure. Consequently, the solution may produce "super-faces" in the dual graph $\mathcal{M}^\star$, i.e., faces $f^\star \in F^\star$ bounded by an excessive number of preserved dual edges. Here, we use a superscript * to denote variables on the dual mesh. In the primal mesh, these correspond to high-valence vertices that degrade mesh quality, as illustrated in Fig. 7. Further, our ILP does not handle other geometric degenerate cases such as self-intersection after edge collapse. To address these issues, we adopt a two-stage processing after the ILP solving.

3.3.1 Dual Graph Simplification. We eliminate "super-faces" by first identifying dual faces with more than 6 boundary edges. We then maintain these dual faces in a queue to be processed iteratively until no dual faces are remaining. Specifically, we adopt two steps to process the dual face. First, we remove dangling vertices. As shown in Fig. 7, a dual face $f^\star$ can have a boundary vertex with valence=1. Here, the valence of a dual vertex is defined as the number of *retained* (i.e., not collapsed) dual edges incident to it; collapsed edges do not participate in the valence computation. We collapse the corresponding primal edges by setting $x_e \leftarrow 0$ to remove all dangling vertices. Second, we sort the boundary vertices of $f^\star$ by their valence. Note that after our first step, the valence is at least 2. We then segment $f^\star$ by tracing an edge from the highest-valent vertex $v^\star_{\text{begin}}$ until we reach another boundary vertex $v^\star_{\text{end}}$. We determine the edge sequence connecting $v^\star_{\text{begin}}$ and $v^\star_{\text{end}}$ by a shortest-path search, where we encourage all consecutive edges lie on the same poly-chord by adding a high penalty when an edge does not follow the same edge loop as the previous edge. After each division of a dual face $f^\star$, we result in two sub-faces, and we insert them back into the queue for further processing. The process terminates when no "super-faces" are remaining.

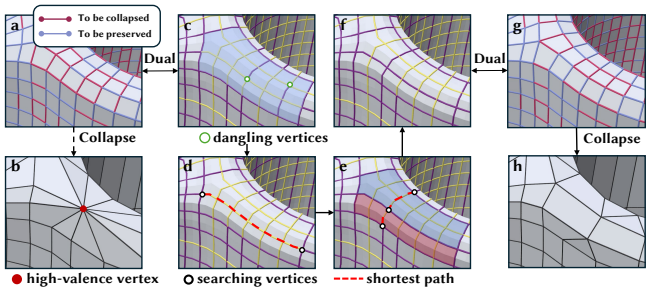

Fig. 7. Dual graph post-processing. In (c–f), violet edges denote retained dual edges and yellow edges denote collapsed dual edges. (a) Directly using ILP results to compute the simplified mesh produces high-valence vertices (b). We perform post-processing in the dual space (c) by first identifying super-faces (blue regions) and their dangling vertices. We then erode these dangling vertices (d) and iteratively repartition super-faces $f^\star$ using shortest-path searches from high-valence vertices. In (e), the initial super-face is divided into two sub-faces with 7 (blue) and 6 (red) boundary edges respectively; since $7 > 6$, the blue sub-face requires further partitioning (f). After clearing all super-faces, we perform QEM-based simplification in the primal space (g) to obtain the final result (h).

3.3.2 QEM-based Simplification. Our final step after the dual graph simplification is to use QEM-based edge collapse. Rather than collapsing edges independently, we perform a connectivity-based cluster collapse: vertices connected by collapsed edges are grouped into connected components via Union-Find, and each component is merged into a single vertex. The position of each merged vertex is determined by minimizing the accumulated quadric error (see Algorithm 3 in supplementary material). This allows users to adopt additional geometry constraints, such as self-intersection free and bounded Hausdorff distance from the original mesh, by preventing edge-collapse when such constraints are violated. We further adopt the per-vertex weight in the quadric error as proposed in [Knodt 2025], in order to better prevent volume shrinkage and preserve sharp features.

4 EXPERIMENTS

To evaluate our method, we collected 50 complex quad-dominant mesh models from Sketchfab [Sketchfab 2022] that were created by artists with proper licensing. We compare our method against the single edge collapse-based method [Knodt 2025] (denoted as **Single**) and the quadric decimation method of MeshLab [Cignoni et al. 2008] (denoted as **MeshLab**). Note that we do not compare with methods designed specifically for pure quad mesh simplification, as they are unable to handle the complex quad-dominant topologies present in our dataset, and reliable open-source implementations are lacking. In our experiments, we compare the results of different methods that simplify the input mesh's face count to 80%, 60%, 40%, and 20% of the original, respectively. Our method is implemented in C++ using the CP-SAT solver within Google OR-Tools [Google 2024] to solve the optimization problem. All experiments are conducted on an Intel i7-9700 8-core CPU.

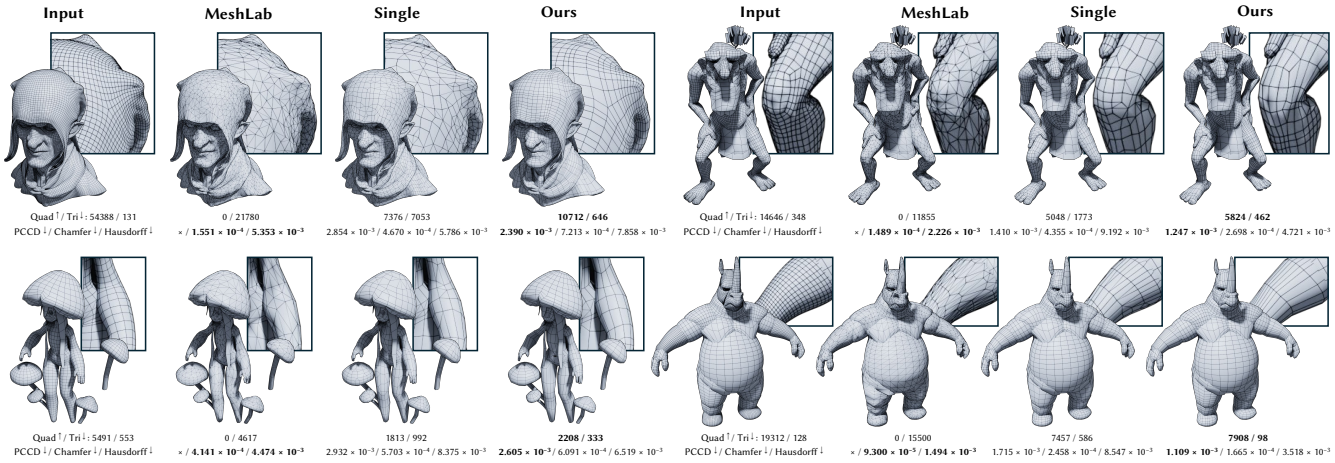

Fig. 8. Qualitative comparison. Our method significantly improves quad preservation, especially the crucial edge flow around joints for animation, all while maintaining comparable geometric fidelity.

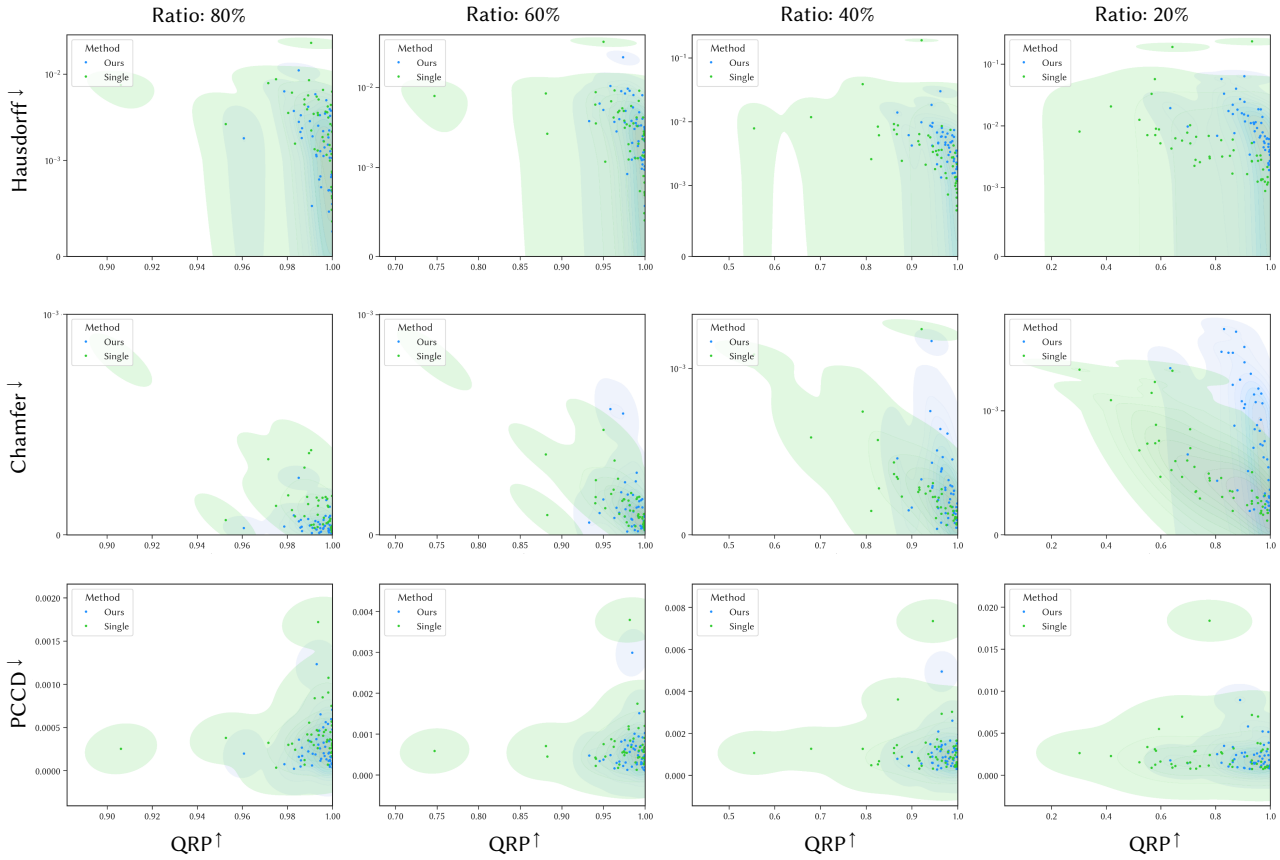

Fig. 9. Comparison of our method (blue) against the Single baseline (green) across reduction ratios from 80% to 20%. The scatter plots illustrate the distribution of regularity (QRP) versus geometric error; Hausdorff and Chamfer distances are shown on a logarithmic scale. The shaded density regions are estimated using a Gaussian kernel with Scott's rule bandwidth. A distribution further to the right indicates better quad preservation, while a distribution further down indicates lower geometric distance. The plots demonstrate that our method consistently achieves superior topology preservation and geometric similarity across all reduction ratios.

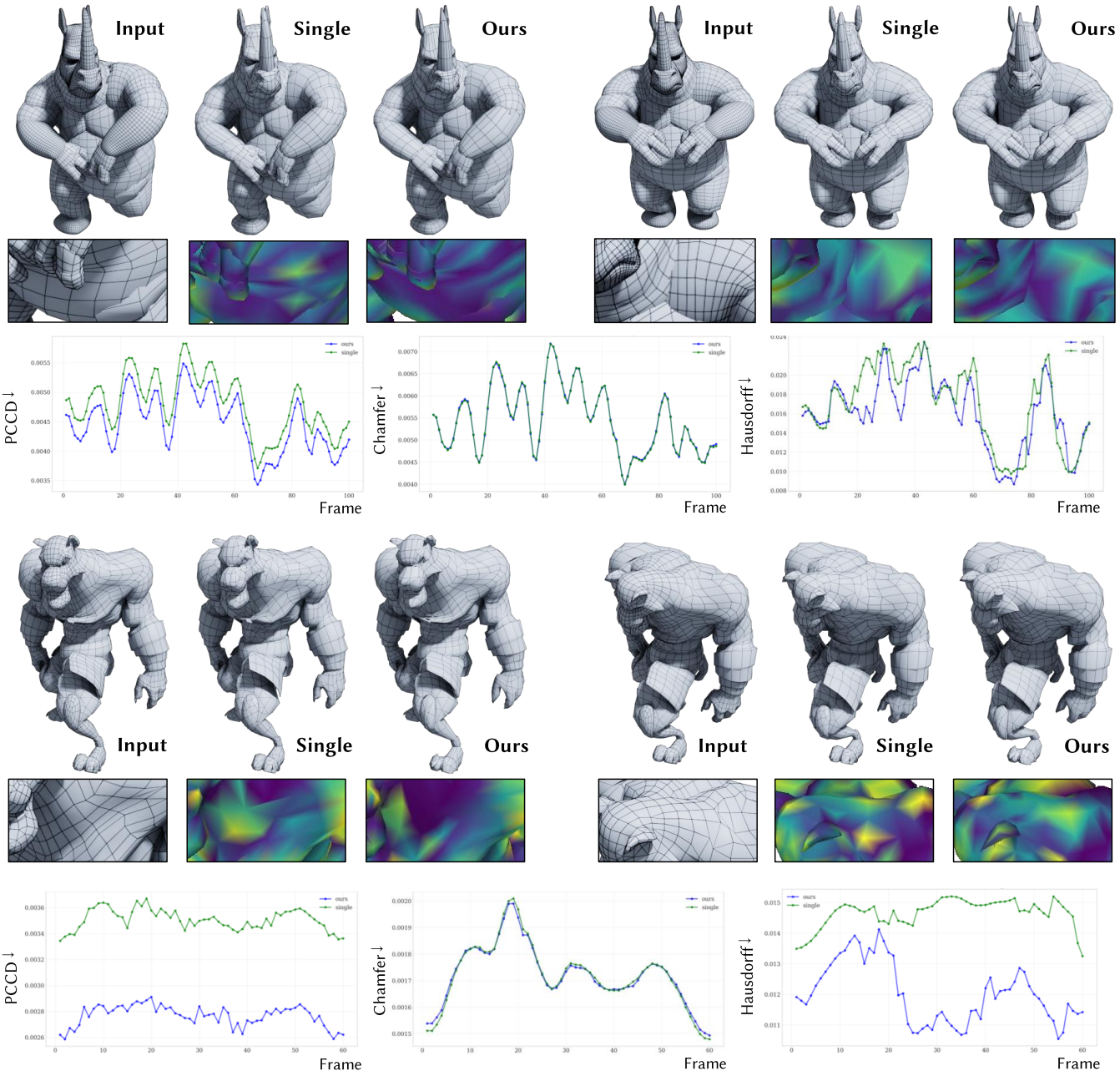

Fig. 10. Frame-by-frame geometric quality comparison for animated meshes. The table presents the PCCD, Chamfer, and Hausdorff distance variations across all frames. Our method demonstrates reduced deformation artifacts owing to its superior edge flow preservation. The inset error maps visualize the per-vertex geometric deviation from the input mesh, showing that our method significantly reduces motion-induced stretching artifacts compared to the Single baseline.

4.1 Results

We evaluate mesh quality using standard geometric metrics (Chamfer and Hausdorff distances) and two topology-aware metrics. All models are normalized to a unit bounding box prior to evaluation, so all distance-based metrics are scale-invariant and directly comparable across models. Quad Preservation Ratio (QPR) [Knodt 2025], defined as $\frac{\text{Out Quads}}{\text{Out Tris}} / \frac{\text{In Quads}}{\text{In Tris}}$, quantifies quad structure retention where values closer to 1.0 indicate better preservation. To evaluate the preservation of global topological flow, as illustrated in Fig. 12, we introduce Poly-Chord Chamfer Distance (PCCD), which computes the Chamfer distance between the medial curves of extracted poly-chords from the original and simplified meshes.

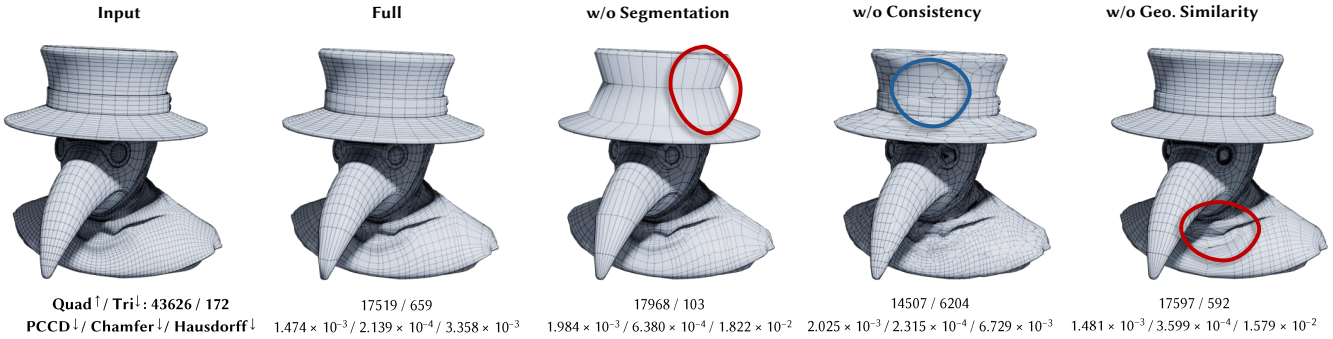

Fig. 11. Qualitative ablation study. By employing multiple constraints, our method effectively avoids undesirable disruption of edge flow (blue circle) and over-simplification (red circle).

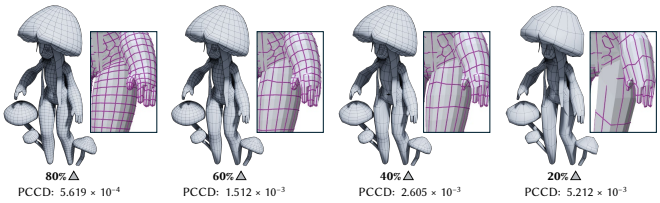

Fig. 12. Illustration of the PCCD metric for simplification results at different ratios. The medial curves of poly-chords are displayed in purple. We assess the preservation of the original mesh's quad topological flow by comparing the similarity of these curves.

Table 1 presents our quantitative comparison across four simplification levels, where our method consistently achieves the highest QPR and PCCD, highlighting superior preservation of quad structures and edge flow even under aggressive simplification. Metrics such as QPR and PCCD are not applicable to MeshLab's output as it produces a pure triangle mesh lacking quad structure. Regarding geometric accuracy, MeshLab unsurprisingly achieves the best results due to its purely geometry-driven approach. We further illustrate the trade-off in Fig. 9, which plots geometric quality against QPR of Single and our method. In this chart, being further to the right indicates a higher quad ratio, while being further down indicates higher geometric similarity. Compared to the Single method, our approach strikes a more favorable balance between geometric fidelity and topological integrity. These results collectively validate the effectiveness of our ILP-based framework in preserving quad-dominant structures while maintaining competitive geometric quality. The qualitative comparison in Fig. 8 clearly shows that our method yields better results in preserving the initial edge flow.

To demonstrate the impact of preserved edge flow on animated meshes, we compare the deformation quality of meshes simplified to 20% using the Single method and our approach. As shown in Fig. 10, we evaluate the geometric distance variation throughout a complex animation sequence. The results indicate that our method achieves a lower geometric distance for the vast majority of frames. Visually, our method exhibits significantly fewer distorted regions, highlighting the practical benefit of our topology-aware simplification for animation applications.

Table 1. Comparison of our approach against Single and MeshLab. Our method achieves superior preservation of quadrilaterals and edge flow while maintaining competitive geometric accuracy. MeshLab requires full triangulation of the input mesh and thus cannot preserve any quad structures.

	Metric	Ours	Single	MeshLab
80% △	Avg. QPR↑	0.9941	0.9920	×
	Avg. PCCD↓	2.596×10^{-4}	4.157×10^{-4}	×
	Avg. Chamfer↓	3.506×10^{-5}	8.828×10^{-4}	1.185×10^{-5}
	Avg. Hausdorff↓	2.220×10^{-3}	3.594×10^{-3}	1.099×10^{-3}
60% △	Avg. QPR↑	0.9859	0.9749	×
	Avg. PCCD↓	5.671×10^{-4}	6.956×10^{-4}	×
	Avg. Chamfer↓	1.099×10^{-4}	1.330×10^{-4}	8.960×10^{-5}
	Avg. Hausdorff↓	4.078×10^{-3}	3.998×10^{-3}	2.436×10^{-3}
40% △	Avg. QPR↑	0.9716	0.9332	×
	Avg. PCCD↓	1.030×10^{-3}	1.219×10^{-3}	×
	Avg. Chamfer↓	2.701×10^{-4}	2.640×10^{-4}	2.001×10^{-4}
	Avg. Hausdorff↓	5.355×10^{-3}	8.170×10^{-3}	3.888×10^{-3}
20% △	Avg. QPR↑	0.9281	0.7917	×
	Avg. PCCD↓	2.185×10^{-3}	2.554×10^{-3}	×
	Avg. Chamfer↓	1.211×10^{-3}	5.390×10^{-4}	5.940×10^{-4}
	Avg. Hausdorff↓	1.301×10^{-2}	1.445×10^{-2}	9.916×10^{-3}

Runtime and ILP Optimality. The ILP solving stage dominates runtime, consistently reaching the 180 s time limit, while all other stages complete in under 12 s. The solver returns high-quality feasible solutions across all tested models, though provable optimality is not guaranteed within this budget. A detailed per-stage runtime breakdown is provided in the supplementary material.

Outlier Analysis. We identify a small number of Hausdorff outliers per ratio, primarily on models with low face counts and elevated triangle content. Fig. 13 shows a representative case: the sparse input and fragmented poly-chords limit the ILP solver's flexibility, causing increased geometric error under aggressive simplification.

4.2 Ablation Studies

We perform an ablation study on several key components of our method, including poly-chord consistency constraints, poly-chord segmentation, geometric similarity constraints, and the dual graph post-process. The results, as presented in Table 2 and Fig. 11, demonstrate that our full method achieves the best edge flow preservation

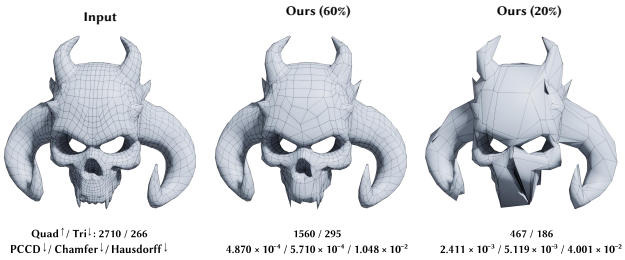

Fig. 13. Outlier analysis on *Demon Mask*. From left to right: input mesh, our result at 60%, and our result at 20%. At 60% reduction, the quad structure and geometric detail are well preserved. At 20%, the low face count and high triangle ratio fragment poly-chords into short segments, limiting the ILP solver's flexibility and leading to visible geometric degradation in thin structures such as the horns and fangs.

and the optimal geometric distance. Notably, removing the poly-chord consistency constraints severely disrupts the original quad structure and edge flow. While removing poly-chord segmentation yields a higher number of preserved quads, the resulting over-constraints induce over-simplification, which significantly degrades geometric quality. Similarly, the absence of geometric similarity constraints leads to over-simplification in many regions, causing a substantial drop in geometric quality.

Table 2. Ablation study on the 40% reduction level. We evaluate the impact of each core component of our method. "w/o" stands for "without".

Metric	Full	w/o Consist.	w/o Segmt.	w/o Geo.	w/o Dual
Avg. QPR$^{\uparrow}$	0.9716	0.8537	0.9966	0.9791	0.9791
Avg. PCCD$^{\downarrow}$	1.030×10^{-3}	2.081×10^{-3}	1.247×10^{-3}	1.213×10^{-3}	1.044×10^{-3}
Avg. Chamfer$^{\downarrow}$	2.701×10^{-4}	3.810×10^{-4}	5.161×10^{-4}	4.035×10^{-4}	2.808×10^{-4}
Avg. Hausdorff$^{\downarrow}$	5.355×10^{-3}	2.043×10^{-2}	6.299×10^{-3}	2.108×10^{-2}	5.909×10^{-3}

5 DISCCUSION

We presented a global ILP framework for quad-dominant mesh reduction that balances topological integrity with geometric fidelity. Utilizing partial poly-chord segmentation and geometry-aware soft constraints, our method preserves critical edge flows and significantly outperforms local heuristics, facilitating downstream editing workflows. Future work will focus on scalable optimization strategies to handle larger assets effectively.

ACKNOWLEDGMENTS

Xiaogang Jin was supported by the National Natural Science Foundation of China (Grants: 62472373, U25A20440).

REFERENCES

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 1–10.

Agostino Bozzo, Daniele Panozzo, Enrico Puppo, Nico Pietroni, and Luigi Rocca. 2010. Adaptive quad mesh simplification. In *Eurographics Italian Chapter Conference*. 95–102.

Marcel Campen, David Bommes, and Leif Kobbelt. 2012. Dual loops meshing: quality quad layouts on manifolds. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–11.

Marcel Campen and Leif Kobbelt. 2014. Dual strip weaving: Interactive design of quad layouts using elastica strips. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 1–10.

Zhen Chen, Zherong Pan, Kui Wu, Etienne Vouga, and Xifeng Gao. 2023. Robust low-poly meshing for general 3d models. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–20.

Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. 2008. MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference*, Vittorio Scarano, Rosario De Chiara, and Ugo Erra (Eds.). The Eurographics Association. doi:10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136

David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. 2004. Variational shape approximation. In *ACM SIGGRAPH 2004 Papers*. 905–914.

Joel Daniels, Cláudio T. Silva, and Elaine Cohen. 2009a. Localized quadrilateral coarsening. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1437–1444.

Joel Daniels, Claudio T Silva, and Elaine Cohen. 2009b. Semi-regular quadrilateral-only remeshing from simplified base domains. In *Computer Graphics Forum*, Vol. 28. Wiley Online Library, 1427–1435.

Joel Daniels, Cláudio T Silva, Jason Shepherd, and Elaine Cohen. 2008. Quadrilateral mesh simplification. *ACM Transactions on Graphics (TOG)* 27, 5 (2008), 1–9.

Christopher DeCoro and Szymon Rusinkiewicz. 2005. Pose-independent simplification of articulated meshes. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*. 17–24.

Alexander Dielen, Isaak Lim, Max Lyon, and Leif Kobbelt. 2021. Learning direction fields for quad mesh generation. *Computer Graphics Forum* 40, 5 (2021), 181–191.

Qiujie Dong, Jiepeng Wang, Rui Xu, Cheng Lin, Yuan Liu, Shiqing Xin, Zichun Zhong, Xin Li, Changhe Tu, Taku Komura, et al. 2025. CrossGen: Learning and generating cross fields for quad meshing. *ACM Transactions on Graphics (TOG)* 44, 6 (2025), 1–15.

Donya Labs AB. 2022. Simplygon 10. https://www.simplygon.com/

Junyi Duan, Xiaopeng Zheng, Na Lei, and Zhongxuan Luo. 2024. Singularity structure simplification for hex mesh via integer linear program. *Computer-Aided Design* 168 (2024), 103654.

Epic Games. 2022. Unreal Engine 5. https://www.unrealengine.com/en-US/unreal-engine-5

Xifeng Gao, Kui Wu, and Zherong Pan. 2022. Low-poly mesh generation for building models. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9.

Michael Garland and Paul S Heckbert. 1997. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. 209–216.

Michael Garland and Paul S Heckbert. 1998. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings Visualization'98 (Cat. No. 98CB36276)*. IEEE, 263–269.

Michael Garland and Yuan Zhou. 2005. Quadric-based simplification in any dimension. *ACM Transactions on Graphics (TOG)* 24, 2 (2005), 209–239.

Google. 2024. Google OR-Tools. https://github.com/google/or-tools.

Jon Hasselgren, Jacob Munkberg, Jaakko Lehtinen, Miika Aittala, and Samuli Laine. 2021. Appearance-driven automatic 3d model simplification. *EGSR (DL)* 29 (2021), 85–97.

Hugues Hoppe. 1999. New quadric metric for simplifying meshes with appearance attributes. In *Proceedings Visualization'99 (Cat. No. 99CB37067)*. IEEE, 59–510.

Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant field-aligned meshes. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–15.

Julian Knodt. 2025. Single edge collapse quad-dominant mesh reduction. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–23.

Julian Knodt, Zherong Pan, Kui Wu, and Xifeng Gao. 2023. Joint UV optimization and texture baking. *ACM Transactions on Graphics (TOG)* 43, 1 (2023), 1–20.

Eric Landreneau and Scott Schaefer. 2009. Simplification of articulated meshes. *Computer Graphics Forum* 28, 2 (2009), 347–353.

Peter Lindstrom. 2000. Out-of-core simplification of large polygonal models. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. 259–262.

Hsueh-Ti Derek Liu, Mark Gillespie, Benjamin Chislett, Nicholas Sharp, Alec Jacobson, and Keenan Crane. 2023. Surface simplification using intrinsic error metrics. *ACM Transactions on Graphics (TOG)* 42, 4 (2023).

Hsueh-Ti Derek Liu, Xiaoting Zhang, and Cem Yuksel. 2025. Simplifying textured triangle meshes in the wild. *ACM Transactions on Graphics (TOG)* 44, 6 (2025), 1–16.

Kok-Lim Low and Tiow-Seng Tan. 1997. Model simplification using vertex-clustering. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. 75–ff.

David Luebke, Martin Reddy, Jonathan D Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. 2002. *Level of detail for 3D graphics*. Elsevier.

Giorgio Marcias, Kenshi Takayama, Nico Pietroni, Daniele Panozzo, Olga Sorkine-Hornung, Enrico Puppo, and Paolo Cignoni. 2015. Data-driven interactive quadrangulation. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.

Ashton Mason. 2020. Quad mesh simplification in frostbite. https://ubm-twvideo01.s3.amazonaws.com/o1/vault/gdc2020/presentations/Quad_Mesh_Simplification_Mason_Ashton.pdf

Gianpaolo Palma, Narges Pourjafarian, Jürgen Steimle, and Paolo Cignoni. 2024. Capacitive Touch Sensing on General 3D Surfaces. *ACM Transactions on Graphics (TOG)*

43, 4, Article 103 (2024).
David Palmer, Albert Chern, and Justin Solomon. 2024. Lifting directional fields to minimal sections. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.
Daniele Panozzo, Enrico Puppo, Marco Tarini, Nico Pietroni, and Paolo Cignoni. 2011. Automatic construction of quad-based subdivision surfaces using fitmaps. *IEEE Transactions on Visualization and Computer Graphics* 17, 10 (2011), 1510–1520.
Nico Pietroni, Stefano Nuvoli, Thomas Alderighi, Paolo Cignoni, Marco Tarini, et al. 2021. Reliable feature-line driven quad-remeshing. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–17.
Bay Raitt and Greg Minter. 1998. Digital Sculpture Techniques (How to Apply the Principles of Traditional Sculpture to make Stunning 3D Characters). https://web.archive.org/web/20160218133259/https://www.ualberta.ca/~cwant/blender/derived-surfaces.pdf#expand
Faniry H. Razafindrazaka and Konrad Polthier. 2017. Optimal base complexes for quadrilateral meshes. *Computer Aided Geometric Design* 52 (2017), 63–74.
Faniry H Razafindrazaka, Ulrich Reitebuch, and Konrad Polthier. 2015. Perfect matching quad layouts for manifold meshes. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 219–228.
Jarek Rossignac and Paul Borrel. 1993. Multi-resolution 3D approximations for rendering complex scenes. In *Modeling in Computer Graphics: Methods and Applications*. Springer, 455–465.
Sketchfab. 2022. The best 3D viewer on the web. https://sketchfab.com/
Kenshi Takayama, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. Pattern-based quadrangulation for N-sided patches. *Computer Graphics Forum* 33, 5 (2014), 177–184.
Marco Tarini, Nico Pietroni, Paolo Cignoni, Daniele Panozzo, and Enrico Puppo. 2010. Practical quad mesh simplification. *Computer Graphics Forum* 29, 2 (2010), 407–418.
Marco Tarini, Enrico Puppo, Daniele Panozzo, Nico Pietroni, and Paolo Cignoni. 2011. Simple quad domains for field aligned mesh parametrization. In *Proceedings of the 2011 SIGGRAPH Asia Conference*. 1–12.
Philip Trettner and Leif Kobbelt. 2020. Fast and robust qef minimization using probabilistic quadrics. *Computer Graphics Forum* 39, 2 (2020), 325–334.
Francesco Usai, Marco Livesu, Enrico Puppo, Marco Tarini, and Riccardo Scateni. 2015. Extraction of the quad layout of a triangle mesh guided by its curve skeleton. *ACM Transactions on Graphics (TOG)* 35, 1 (2015), 1–13.
Rui Xu, Longdu Liu, Ningna Wang, Shuangmin Chen, Shiqing Xin, Xiaohu Guo, Zichun Zhong, Taku Komura, Wenping Wang, and Changhe Tu. 2024. CWF: consolidating weak features in high-quality mesh simplification. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–14.
Sen Zhang, Hui Zhang, and Jun-Hai Yong. 2016. Automatic quad patch layout extraction for quadrilateral meshes. *Computer-Aided Design and Applications* 13, 3 (2016), 409–416.