Zhichao WANG, Xinhai CHEN, Junjun YAN, Jie LIU, 2025. An intelligent mesh-smoothing method with graph neural networks. *Frontiers of Information Technology & Electronic Engineering*, 26(3):367-384.
https://doi.org/10.1631/FITEE.2300878

An intelligent mesh-smoothing method with graph neural networks

Key words: Unstructured mesh; Mesh smoothing; Graph neural network; Optimization-based smoothing

Xinhai CHEN
E-mail: chenxinhai16@nudt.edu.cn
ORCID: https://orcid.org/0009-0007-2931-4893

Zhichao WANG
E-mail: wangzhichao@nudt.edu.cn
ORCID: https://orcid.org/0009-0007-4034-0578

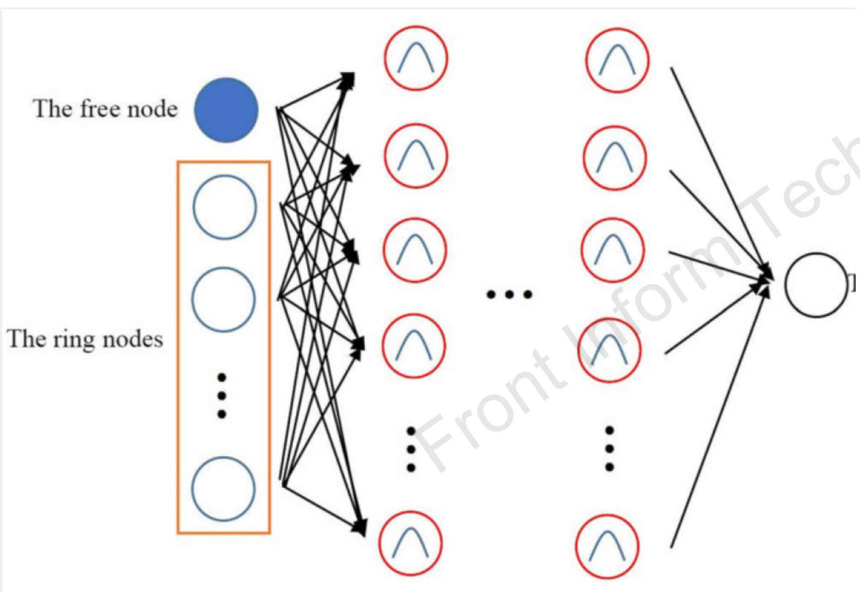
Mesh-smoothing method

- Mesh-smoothing methods are computational techniques used to improve the quality and regularity of mesh elements in numerical simulations by adjusting node positions to minimize distortions and enhance geometric properties.

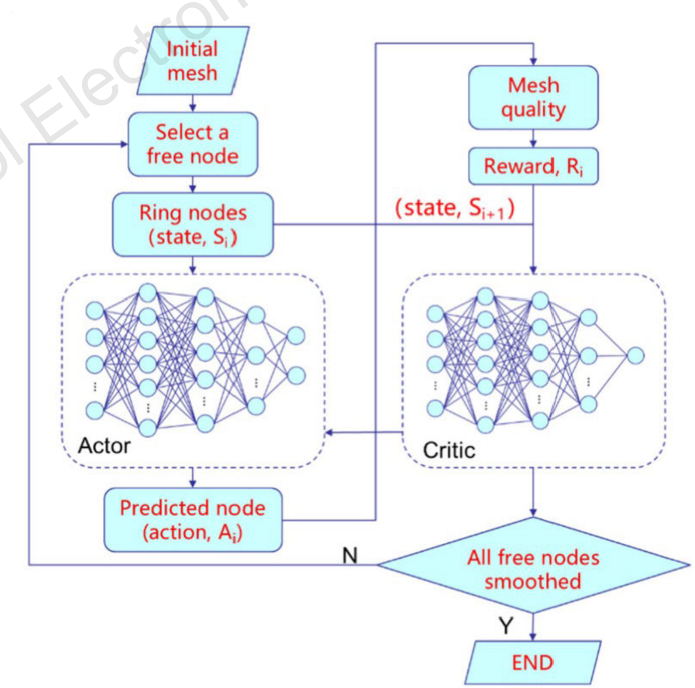
Method	Description
Laplacian smoothing	Laplacian smoothing relocates each node to the average position of its neighboring nodes to reduce irregularities and improve mesh quality.
Angle-based smoothing	Angle-based smoothing adjusts vertex positions to minimize the deviation of angles from their ideal values, thereby enhancing element quality and geometric uniformity.
CVT smoothing	CVT smoothing iteratively relocates vertices to the centroids of their corresponding Voronoi regions, ensuring a more uniform and high-quality mesh distribution.
GETMe smoothing	Smoothing method based on element transformation that aims to enhance mesh quality by applying linear transformations to mesh elements, allowing individual elements to converge to optimal shapes through multiple iterations.
Optimization-based smoothing	Optimization-based smoothing minimizes a predefined objective function, such as element distortion or energy, to iteratively adjust node positions and achieve optimal mesh quality.

Tendency and challenges

- AI-based mesh smoothing methods are advancing towards automation, efficiency, and intelligence, leveraging deep learning to adaptively optimize mesh quality, significantly enhancing computational accuracy and efficiency.



NN-Smoothing method



DRL-Smoothing method

Motivation

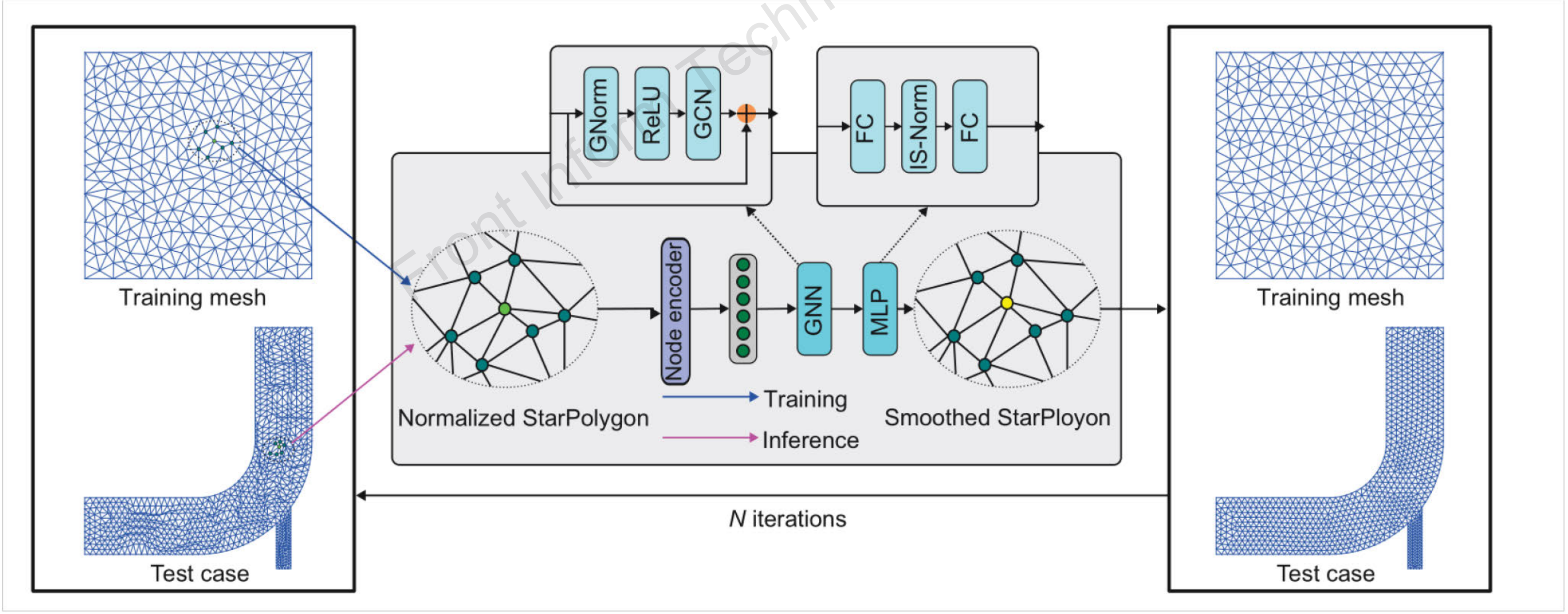
- The paper aims to overcome the computational inefficiency of optimization-based mesh smoothing and the dependency on labeled high-quality meshes in supervised learning approaches, which struggle with varying node degrees and input sequence issues.
- A lightweight, graph neural network based model is introduced to intelligently smooth meshes without labeled data, incorporating fault tolerance and a novel loss function for efficient, high-quality results with minimal computational cost.

Comparison among optimization-based smoothing, NN-Smoothing, and GMSNet

Method	Speed	Labeled high-quality mesh	Varying node degrees	Node input order
Optimization-based smoothing	Low	Not acquiring	Not affected	Not affected
NN-Smoothing	High	Acquiring	Training separate models	Performing data augmentation
GMSNet	High	Not acquiring	Not affected	Not affected

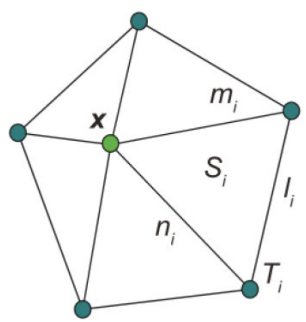
1) Lightweight model design

- The paper proposes GMSNet, a lightweight graph neural network (GNN)-based model that normalizes node inputs and uses residual GCN layers to efficiently compute optimal node positions for mesh smoothing.
- GMSNet leverages GNN's permutation invariance to process nodes with varying degrees and remain unaffected by input order, eliminating the need for separate models or data augmentation.



2) Model training

- The paper introduces a novel loss function, MetricLoss, which eliminates the need for labeled high-quality meshes by directly optimizing geometric metrics for stable and rapid convergence during training.
- Through MetricLoss, we introduce an unsupervised learning mechanism that enables the training of mesh smoothing models using only unlabeled mesh data.



$$|S(\boldsymbol{x})| = 5$$

$$\mathcal{L}(\boldsymbol{x}, S(\boldsymbol{x})) = \frac{1}{|S(\boldsymbol{x})|}\sum_{i=1}^{|S(\boldsymbol{x})|}\left(1-\frac{1}{q_i}\right) = \frac{1}{|S(\boldsymbol{x})|}\sum_{i=1}^{|S(\boldsymbol{x})|}\left(1-\frac{4\sqrt{3}S_i}{m_i^2+n_i^2+l_i^2}\right)$$

MetricLoss

Algorithm 2 Training of GMSNet

Input: Mesh dataset $M = \{\mathcal{M}_i\}_{i=1}^{n}$, MetricLoss $\mathcal{L}$, number of training epochs N, batch size $\mathcal{B}$, learning rate α, and GMSNet parameter $\boldsymbol{W}$

1: **for** $j \leftarrow 1$ to N **do**
2: **for** mesh $\mathcal{M}_i$ in M **do**
3: Sample $\mathcal{B}$ mesh nodes from $\mathcal{M}_i$: $\{\boldsymbol{x}_i\}_{i=1}^{\mathcal{B}}$
4: Compute the optimized node position: $\boldsymbol{x}_i^* = \text{GMSNet}(\boldsymbol{x}_i, S(\boldsymbol{x}_i))$ for $\boldsymbol{x}_i \in \{\boldsymbol{x}_i\}_{i=1}^{\mathcal{B}}$
5: Truncate $\Delta\boldsymbol{x}_i$ by Algorithm 1 if $\Delta\boldsymbol{x}_i$ results in negative volume elements
6: **end for**
7: Update model parameters: $\boldsymbol{W} \leftarrow \boldsymbol{W} - \frac{\alpha}{\mathcal{B}}\sum_{i=1}^{\mathcal{B}}\nabla\mathcal{L}(\boldsymbol{x}_i^*, S(\boldsymbol{x}_i^*))$ /* We use stochastic gradient descent as the optimization method */
8: **end for**

Training of GMSNet

3) Fault-tolerant mechanism

- Shift Truncation is introduced as a technique to address the issue of node displacement during the mesh smoothing process. It ensures that the movement of nodes does not exceed a predefined threshold, thereby preventing the generation of invalid elements and maintaining the stability and quality of the mesh.

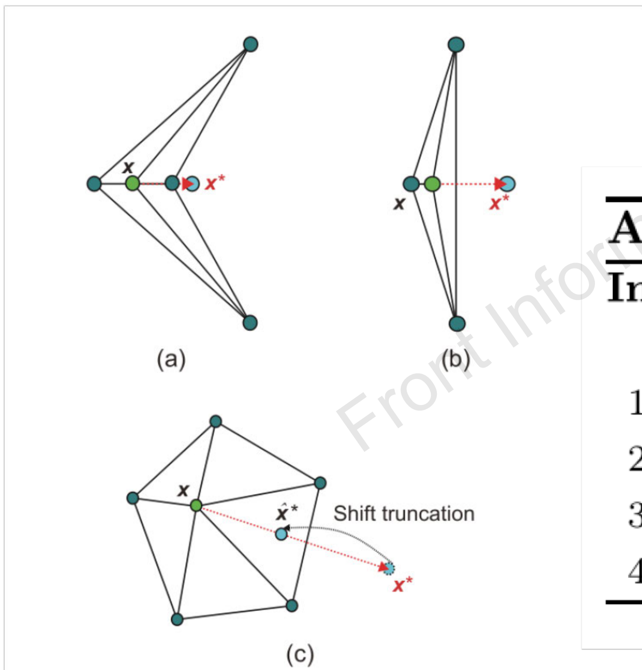


Illustration of shift truncation

Algorithm 1 Shift truncation operation

Input: Mesh node i, original position $\boldsymbol{x}_i$, optimized position $\boldsymbol{x}_i^*$, and the position shift $\Delta\boldsymbol{x}_i$

1: **while** $\boldsymbol{x}_i^*$ results in negative elements **do**

2: $\quad \Delta\boldsymbol{x}_i = 0.5\Delta\boldsymbol{x}_i$

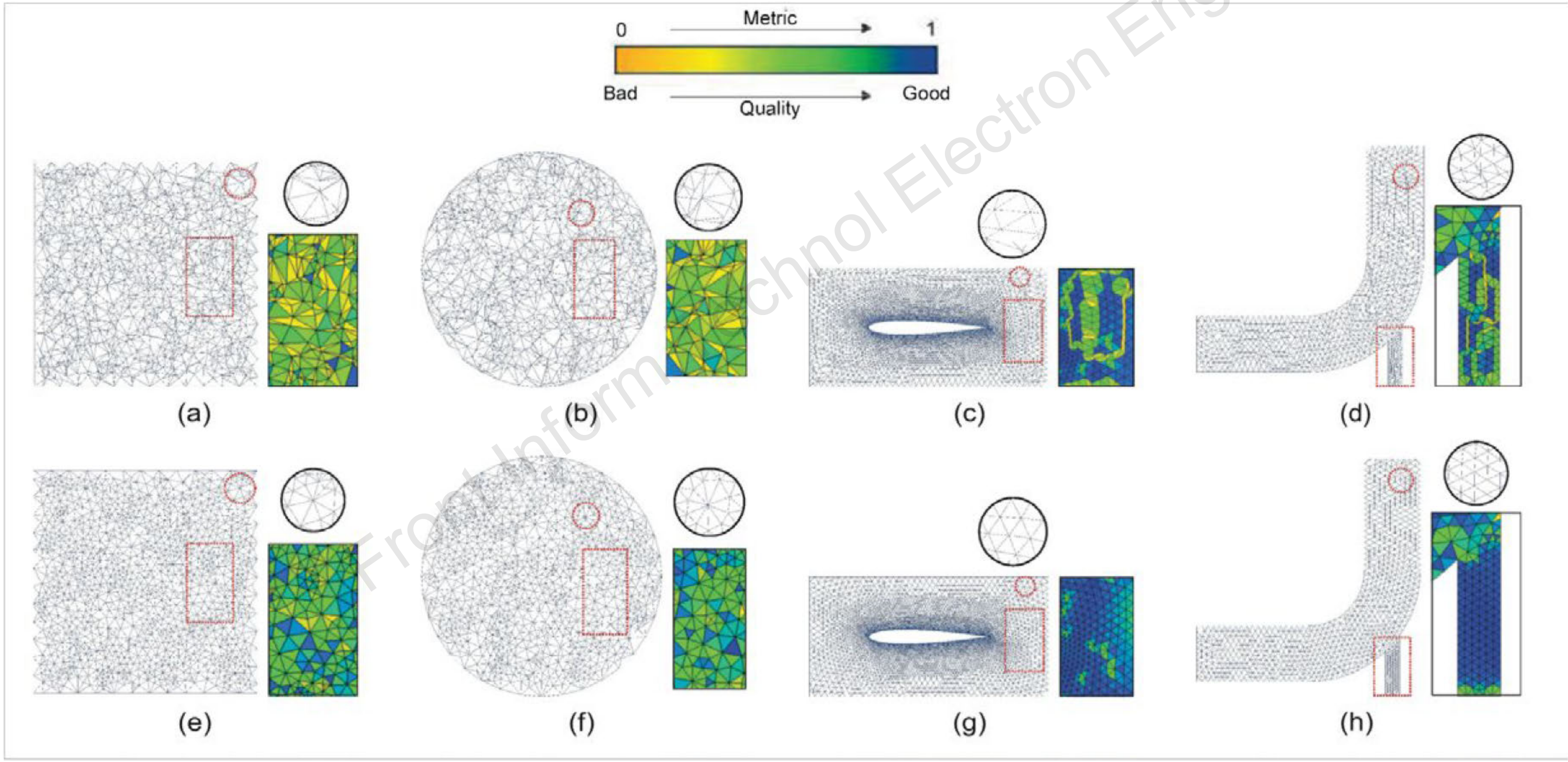
3: $\quad \boldsymbol{x}_i^* = \boldsymbol{x}_i + \Delta\boldsymbol{x}_i$

4: **end while**

Algorithm of shift truncation

4) Model performance

- The proposed model demonstrates exceptional performance across various mesh test cases with different geometries, significantly improving the quality of mesh elements.



Smoothing result of GMSNet

4) Model performance

- Compared to other traditional mesh smoothing models and intelligent mesh optimization models, the proposed model achieves optimal results in both smoothing efficiency and smoothing quality.

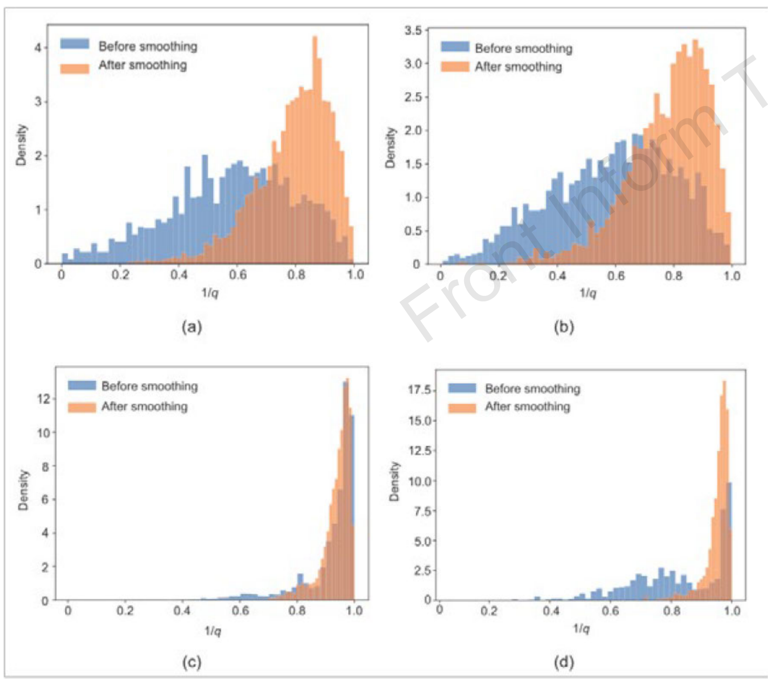
Table 2 Performance of mesh-smoothing algorithms

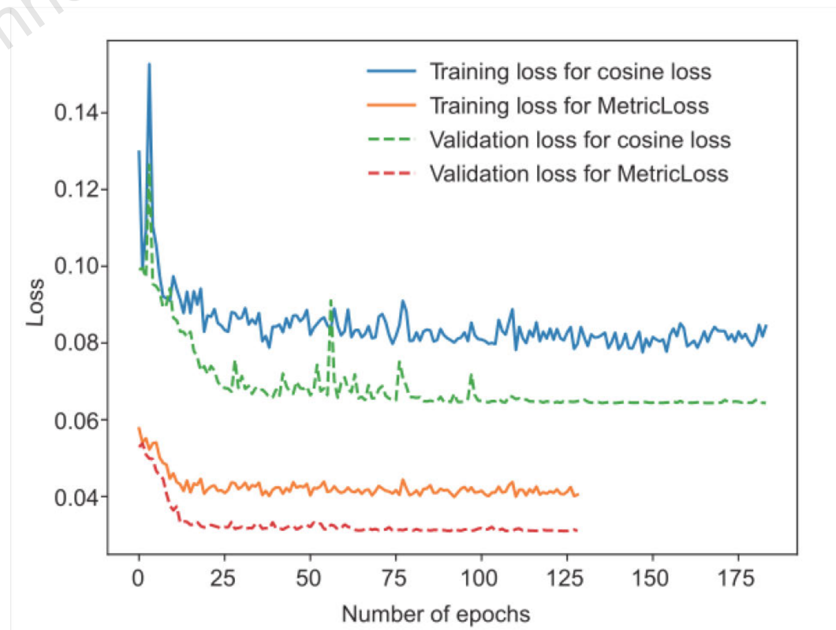
Mesh	Algorithm	Min angle		Max angle		$1/q$		Speed (s/node)
		Min	Mean	Max	Mean	Min	Mean	
Square	Origin	0.04	29.65	179.82	95.09	0.00	0.58	–
	Laplacian smoothing	12.38	**43.97**	147.67	**79.44**	0.25	**0.79**	1.28E-04
	Angle-based smoothing	10.52	41.26	148.60	81.76	0.22	0.76	5.66E-04
	CVT smoothing	2.55	40.88	170.60	84.04	0.05	0.75	1.17E-03
	GETMe smoothing	13.30	36.30	146.88	87.63	0.25	0.69	2.10E-03
	Optimization-based smoothing	**16.33**	43.94	**136.99**	80.70	**0.32**	**0.79**	9.27E-03
	NN-Smoothing	9.14	43.58	158.97	79.79	0.16	**0.79**	4.67E-04
	GMSNet	13.99	43.81	151.39	79.81	0.22	**0.79**	6.39E-04
Circle	Origin	0.29	30.38	179.05	94.60	0.01	0.59	–
	Laplacian smoothing	1.43	42.86	174.53	**81.19**	0.03	**0.78**	1.31E-04
	Angle-based smoothing	1.95	39.47	173.97	84.23	0.04	0.73	5.95E-04
	CVT smoothing	3.19	39.98	172.98	85.59	0.05	0.73	1.20E-03
	GETMe smoothing	1.17	36.83	172.90	88.08	0.03	0.69	2.18E-03
	Optimization-based smoothing	**6.52**	**43.31**	**154.66**	81.90	**0.15**	**0.78**	8.78E-03
	NN-Smoothing	1.26	41.10	176.04	83.86	0.03	0.75	4.83E-04
	GMSNet	2.20	43.00	169.21	81.29	0.05	**0.78**	6.58E-04
Airfoil	Origin	0.25	53.25	178.91	67.84	0.01	0.92	–
	Laplacian smoothing	26.36	**54.91**	111.02	65.78	0.51	**0.94**	1.34E-04
	Angle-based smoothing	20.26	53.49	118.76	66.64	0.42	0.93	6.11E-04
	CVT smoothing	25.49	52.08	115.79	68.19	0.48	0.91	1.22E-03
	GETMe smoothing	**36.02**	52.90	**95.78**	67.89	**0.65**	0.92	2.24E-03
	Optimization-based smoothing	30.53	54.90	108.85	**65.72**	0.54	**0.94**	8.40E-03
	NN-Smoothing	27.69	54.06	113.06	66.57	0.51	0.93	4.88E-04
	GMSNet	27.17	54.50	110.47	66.08	0.52	**0.94**	6.71E-04
Pipe	Origin	4.10	46.28	170.48	76.81	0.07	0.82	–
	Laplacian smoothing	27.91	**56.55**	112.45	**63.93**	0.52	**0.96**	1.30E-04
	Angle-based smoothing	24.28	54.62	101.93	65.69	0.53	0.94	5.81E-04
	CVT smoothing	27.78	54.22	97.80	66.28	0.62	0.93	1.18E-03
	GETMe smoothing	**33.76**	51.07	**93.51**	69.61	**0.65**	0.90	2.14E-03
	Optimization-based smoothing	32.39	56.41	106.07	64.14	0.57	**0.96**	9.01E-03
	NN-Smoothing	28.28	53.72	112.79	66.69	0.51	0.93	4.74E-04
	GMSNet	28.23	55.76	112.27	64.71	0.52	0.95	6.47E-04

The best results are highlighted in bold, while the second-best results are annotated with a gray background. q is the aspect ratio

4) Model performance

- The model significantly reduces the proportion of low-quality mesh elements while increasing the proportion of high-quality mesh elements.
- Additionally, the paper discusses the training process of the model under different unsupervised loss functions for mesh optimization, demonstrating the effectiveness of MetricLoss in stabilizing the training process and promoting model convergence.



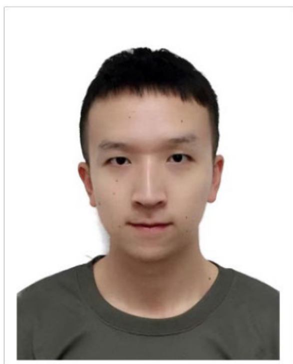


Future outlook

- **Generalization to 3D meshes:** future research will focus on extending the proposed method to three-dimensional (3D) mesh smoothing, addressing the additional complexities and challenges in 3D geometry.
- **Integration with real-time applications:** exploring the integration of the model into real-time simulation and engineering applications to enhance computational efficiency and practical usability.
- **Advanced loss functions:** investigating more sophisticated loss functions and optimization strategies to further improve mesh quality and model convergence in diverse scenarios.

Author Biography

Zhichao WANG earned his Master of Science degree in Computer Technology from the National University of Defense Technology (NUDT), China, in 2022. Currently, he is advancing his academic journey as a doctoral candidate at NUDT. His scholarly pursuits span a diverse array of research domains, notably intelligent mesh generation, sophisticated mesh refinement techniques, the integration of artificial intelligence in computational fluid dynamics (CFD), and the exploration of physics-informed neural networks (PINNs).



Xinhai CHEN obtained his Ph.D. in Computer Science and Engineering from the National University of Defense Technology (NUDT), China, in 2021. Currently, he serves as an Associate Professor at the Science and Technology on Parallel and Distributed Processing Laboratory, NUDT, China. His research expertise spans deep learning applications in computational fluid dynamics (CFD), intelligent mesh generation, physics-informed neural networks (PINNs), and intelligent flow prediction. Dr. Chen has authored and co-authored over 20 peer-reviewed papers in prestigious international conferences and journals, contributing significantly to these cutting-edge research areas.

Jie LIU earned both his Master of Science and Doctor of Philosophy (PhD) degrees in Computer Science from the National University of Defense Technology (NUDT) in Changsha, China. Currently, he holds the position of Professor at the College of Computer, NUDT. His research focuses on high-performance computing (HPC), computational fluid dynamics (CFD), and machine learning. With a prolific academic career, Professor Liu has authored and co-authored over 100 papers in these fields, making significant contributions to the advancement of HPC, CFD, and machine learning technologies.