MPG: An Efficient Multi-Scale Point-Based GNN for Non-Uniform Meshes

Qinxin Wu[1*], Pengwei Liu[1*], Xingyu Ren[1], and Dong Ni[1] (✉)

Zhejiang University, Hangzhou, Zhejiang 310027, China
{22360247,liupw,12332063,dni}@zju.edu.cn

Abstract. Graph Neural Networks (GNNs) have become a powerful tool for modeling complex physical simulations, leveraging their ability to learn from irregular data representations. However, non-uniform meshes introduce significant challenges, particularly in adaptive multi-scale sampling, topological reconstruction, and efficient feature aggregation, often leading to high computational costs. Existing methods struggle to balance efficiency and accuracy due to their inability to dynamically adapt to irregular mesh structures. To address these limitations, we introduce Multi-Scale Point-Based Graph Neural Networks (MPG), a framework that combines point-cloud downsampling with topology-constrained strategies. MPG employs density-aware hierarchical sampling to adaptively retain critical nodes while leveraging a learnable neighborhood aggregation mechanism to enhance local structural sensitivity. Additionally, we introduce adaptive Constrained Delaunay reconstruction, which preserves global topology by eliminating invalid edges and maintaining boundary constraints during coarsening. To further improve efficiency, our model integrates lightweight residual MLPs, enabling scalable dimensions on multi-scale features. MPG's architecture supports dynamic multi-scale compression across diverse physical domains. Evaluations on fluid dynamics, thermochemical reactions, and cavity flow demonstrate that MPG reduces parameter count by at least 84.5% and accelerates training by 17.7% $\sim$ 86.1% per epoch compared to baseline models, while maintaining high single-step rollout accuracy (RMSE$<10^{-2}$). These results establish MPG as a new benchmark for efficient and accurate simulations on non-uniform meshes, offering a versatile solution for complex physical systems.

Keywords: Non-uniform meshes · Multi-scale · Graph Neural Networks · Physical simulation · Surrogate model

1 Introduction

Non-uniform meshes are widely used in computational fluid dynamics (CFD) and other physics-based simulations, as they enhance accuracy in high-gradient

[*] These authors contributed equally to this work.

regions and complex geometries. However, their irregular structure poses significant challenges for feature extraction and efficient computation. Traditional numerical solvers, such as finite difference methods[1] and finite element methods[2], provide accurate solutions but suffer from high computational costs, particularly in large-scale or real-time applications. To improve efficiency, data-driven methods have been explored for mesh-based simulations[3,4,5]. For example, convolutional neural networks (CNNs) leverage local receptive fields and weight sharing, making them effective for structured data[6,7,8]. However, their reliance on uniform grids necessitates resampling non-uniform meshes, leading to information loss and artificial smoothing. Moreover, CNNs struggle to capture the topological relationships inherent in non-uniform meshes, limiting their ability to model connectivity-dependent physical interactions.

Graph-based neural networks (GNNs) offer a more flexible framework for modeling complex physical systems[9]. By employing edge-node message passing (MP), GNNs can effectively capture local node interactions without the constraints of grid-based representations. As a result, they have been successfully applied to fluid dynamics[10], solid mechanics[11], and multi-physics coupling[12]. Among these, MeshGraphNets[10] stands out as a foundational end-to-end framework, demonstrating the potential of GNNs in mesh-based simulations. However, GNNs face notable challenges in large-scale simulations, including quadratic complexity in message passing[13] and feature oversmoothing[14], which hinder their scalability and accuracy.

To overcome the limitations of standard GNNs, multi-scale GNNs have been introduced to improve computational efficiency and enable hierarchical feature extraction[12,15,16,17,18]. These models construct coarse sub-level graphs to enable longer-range interactions and reduce MP iterations. However, existing methods face challenges in preserving graph connectivity during coarsening. Current methods primarily rely on spatial proximity[17,18,19], learnable sampling[15], manual mesh partitioning[13,16], algebraic multigrid algorithms[20], and automated bi-stride sampling[12], each with inherent limitations. Specifically, learnable or random sampling can introduce artificial partitions, hindering information exchange. Spatial proximity-based coarsening often generates incorrect edges, while algebraic multigrid algorithms suffer from cubic complexity. Bi-stride sampling preserves topology but lacks adaptability in feature dimension and sampling scale. Crucially, all these methods rely on deep MP stacks for information propagation, leading to high computational costs.

To overcome these limitations, we propose **MPG**, an efficient multi-scale GNN that integrates point-based feature extraction and graph-based topology preservation. MPG is designed to construct topology-preserving multi-scale graphs, enhance adaptive neighborhood aggregation, and achieve computational efficiency without deep MP stacking. As illustrated in Fig.1, MPG effectively coarsens non-uniform meshes while preserving key structural features. By combining hierarchical density-based sampling, topology-constrained Delaunay reconstruction, and residual MLP-based feature extraction, MPG sets a new benchmark for scal-

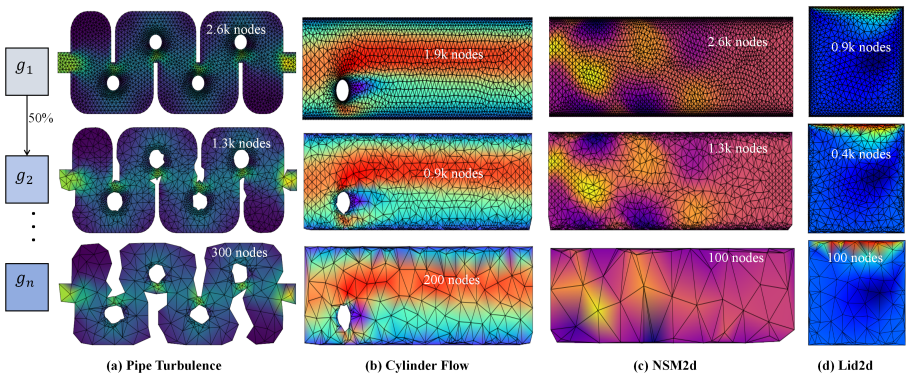


Fig. 1: Example multi-level graphs produced by our proposed method. The coarsening process is particularly challenging for non-uniform meshes with irregular structures, large holes, and non-convex regions. Our Multi-Scale Graph Construction strategy effectively maintains key features and ensures robust connectivity across arbitrary geometries.

able, high-fidelity simulations of complex physical systems. The contributions of the proposed MPG are summarized as follows:

- **Multi-Scale Graph Construction Strategy**: MPG employs a hierarchical density-aware sampling strategy, inspired by point cloud geometric sampling. It applies random sampling in dense regions and farthest point sampling (FPS) in sparse regions to ensure balanced node distributions. To preserve global topology, we incorporate Constrained Delaunay Reconstruction, which prevents incorrect topological connections and ensures accurate edge reconstruction during coarsening. This strategy significantly improves graph connectivity and leads to higher predictive accuracy, achieving RMSE$<10^{-2}$ across various physical simulation tasks.
- **Robust Neighborhood Aggregation**: Instead of relying on a fixed adjacency matrix, MPG leverages random multi-hop neighborhoods inspired by GraphSAGE[21] to improve local feature propagation. We introduce a learnable aggregation mechanism, dynamically adjusting neighbor importance to enhance adaptability across different resolutions. This approach reduces redundant computations while maintaining accuracy, leading to nearly 85% reduction in model parameters compared to conventional message-passing GNNs.
- **Computational Efficiency and Scalability**: Unlike conventional multi-scale GNNs that rely on deep MP stacking, MPG employs a hierarchical residual MLP-based feature extraction framework, inspired by point cloud feature extraction modules[22]. This design allows MPG to scale independently of mesh size, reducing redundant feature transformations while maintaining accuracy. Experimental results show that MPG accelerates training

by 17.7% $\sim$ 86.1% per epoch compared to baseline models, while achieving competitive accuracy with significantly fewer parameters.

2 Related Work

2.1 Graph-Based Method

Non-uniform meshes present significant challenges for traditional CNNs, because of their irregular structures and varying connectivity. This limitation has motivated the adoption of graph-based methods, which treat meshes as graphs to better handle irregular geometries and complex physical system predictions[20]. Pfaff et al. introduced MeshGraphNets[10], demonstrating the effectiveness of GNNs in capturing both the geometric and topological information inherent in non-uniform meshes. Building on these insights, Zhao et al. proposed a hybrid CNN-GNN model to better capture connectivity and flow pathways within porous media[23]. Expanding the scope of GNN-driven modeling, Han et al. combined GNNs with transformers architectures to predict latent states in physics simulations[24]. Meanwhile, TIE[25] departs from traditional graph edges, simplifying the model and enhancing its capacity to capture spatial relationships through self-attention mechanisms. These works demonstrate the growing potential of hybrid Graph-based models, enhancing their ability to model complex physical systems with non-uniform meshes.

2.2 Multi-Scale GNNs

While GNNs offer an effective means of handling non-uniform meshes, large-scale graphs can incur high computational costs. To address these challenges, multi-scale GNN frameworks have been proposed. Graph U-Net[15] first adapts the U-Net architecture to the graph domain by introducing learnable sampling and upsampling opeartions, whereas MS-GNN-Grid[17] relies on voxelization-based sampling, that leverages spatial coordinates for coarsening. However, depending solely on spatial positions and raw node features may be inadequate for accurately capturing the inherent complexity of non-uniform mesh distributions. To further refine multi-scale representations, Yang et al. introduced AMGNET[20], integrating multiple geometric algebra thchniques for mesh coarsening across different scales. Cao et al. proposed BSMS-GNN[12], which employs a two-step breadth-first search (BFS) sampling strategy to preserve vital local information.

2.3 Point Cloud-Based Method

Non-uniform meshes and point clouds share structural similarities, such as unstructured node distribution and geometric irregularity, making point cloud processing techniques relevant for mesh analysis. The primary challenge lies in developing effective feature aggregation operators, which can be categorized into local and global approaches[26]. Pioneering works like PointNet[27] introduced

global feature aggregation using symmetric functions, while PointNet++[28] extended this with hierarchical local feature learning. Subsequent methods, such as KPConv[29] and PointConv[30], further advanced local feature extraction through convolutions and density-adaptive techniques. A notable paradigm shift was demonstrated by Ma et al.[22] in their PointMLP framework, which achieved excellent performance using lightweight geometric affine modules within a residual structure, emphasizing that complex architectures are not always necessary for effective geometric reasoning. Inspired by these advancements, we propose leveraging multi-scale fusion and local feature learning techniques from point cloud processing to enhance mesh node downsampling and improve local information capture.

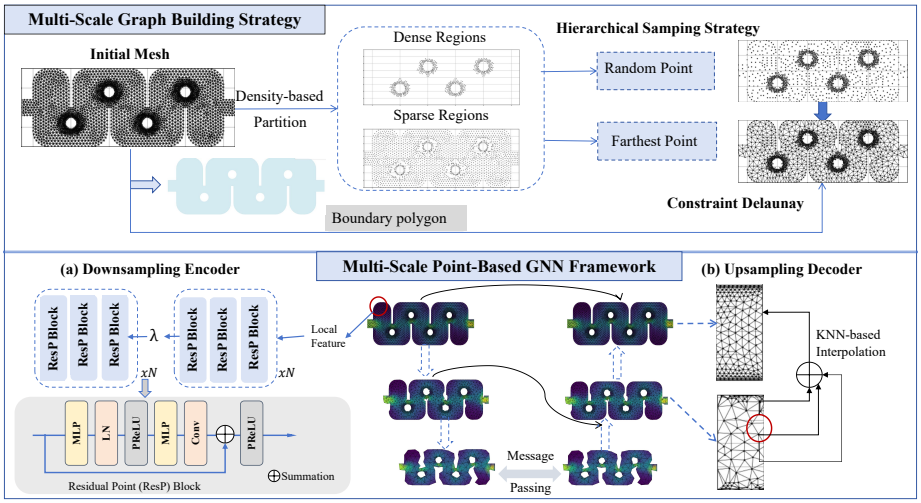


Fig. 2: A model overview of the Multi-Scale Point-Based GNN(MPG) framework. The top panel illustrates the initial mesh partitioning into dense and sparse regions, followed by constrained Delaunay reconstruction for topology preservation. The bottom panel depicts the hierarchical feature extraction and reconstruction process, utilizing lightweight residual MLPs and KNN-based interpolation for efficient multi-scale processing.

3 Multi-Scale Graph Constructing

In this section, we will introduce the hierarchical density-based sampling strategy, constrained Delaunay reconstruction, and the corresponding neighborhood definition for local feature aggregation in Sec.4. All algorithms and preprocessing steps here are completed in one pass, and we can flexibly adjust the sampling

ratio of each layer, the proportion between different density regions, and the fixed sizes of neighbor nodes for different physical simulations scenarios.

3.1 Problem Formulation

Multi-scale graph construction aims to generate hierarchical representations of complex systems while preserving essential distribution characteristics and topological structure. For non-uniform meshes, this process involves node selection, feature aggregation, and edge reconstruction, all of which must balance computational efficiency and physical accuracy. An ideal multi-scale graph representation should meet three key requirements: 1) Selected nodes should retain both local geometric details and global distribution characteristics, preserving key high-gradient regions. 2) Neighborhood aggregation: The model should efficiently extract and propagate local features, minimizing computational cost while adapting to variable mesh structures. 3) Edge reconstruction: Connectivity between coarse and fine graph layers must be structurally consistent, preventing incorrect edges that distort the underlying topology.

However, existing multi-scale GNN methods exhibit several limitations. Spatial proximity-based methods often generate incorrect edges across boundaries, distorting topological structures. Learnable sampling methods focus on locally important nodes, failing to capture global mesh distribution. Automated strategies, such as algebraic multigrid and bi-stride sampling, suffer from high computational complexity and limited adaptability in adjusting multi-scale resolutions.

To address these challenges, we propose a Multi-Scale Graph Construction strategy that integrates hierarchical density-based sampling, topology-aware edge reconstruction, and efficient neighborhood aggregation. As illustrated in Fig.2(top panel), our approach first partitions the input mesh into dense and sparse regions based on local geometric and topological properties. It then applies adaptive sampling techniques to preserve both local details and global structure described in Sec.3.2. To prevent incorrect topology, we incorporate a Constrained Delaunay edge reconstruction strategy in Sec.3.4, ensuring smooth transitions between scales. The overall process is formalized in Algorithm 1, which outlines the hierarchical mesh coarsening procedure with Delaunay triangulation.

3.2 Hierarchical Density-Based Sampling Strategy

To simplify non-uniform meshes while preserving geometric and topological fidelity, we introduce a density-based hierarchical sampling strategy. This method classifies mesh regions into dense and sparse regions based on local node degree and average edge length metrics. Nodes in dense regions, characterized by high geometric complexity or sharp gradients, require fine-grained sampling to preserve local features, while sparse regions are coarsened to maintain global structure with reduced computational costs. Given a non-uniform mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, we first compute the global average edge length:

$$\bar{l} = \frac{1}{|\mathcal{E}|} \sum_{e=(u,v)\in\mathcal{E}} \|x_u - x_v\|, \tag{1}$$

Algorithm 1 Hierarchical Mesh Coarsening with Constrained Delaunay Triangulation

Require: Mesh $\mathcal{M}$, Positions $\mathbf{P}$, Levels L, Coarsening Ratios $\mathbf{U}$, Neighbor Counts $\mathbf{K}$
Ensure: Coarsened Meshes $\{\mathcal{M}_l\}_{l=1}^{L}$, Positions $\{\mathbf{P}_l\}_{l=1}^{L}$, Sampling Indices $\{\mathbf{I}_l\}_{l=1}^{L}$, Neighbor Indices $\{\mathbf{N}_l\}_{l=1}^{L}$
Initialize coarse mesh parameters;
while not converged **do**
  Sample nodes from mesh $\mathcal{M}$ based on ratio $\mathbf{U}$;
  for each level $l = 1, \ldots, L$ **do**
    Generate sampling indices $\mathbf{I}_l$;
    Update positions $\mathbf{P}_l$;
    Construct coarse mesh $\mathcal{M}_l$ via Constrained Delaunay triangulation;
    Compute neighbor indices $\mathbf{N}_l$ using fixed-size $\mathbf{K}$;
  end for
  Update coarse mesh and parameters;
end while

where $\|x_u - x_v\|$ denotes the Euclidean distance between nodes u and v. Edges longer than $\bar{l}$ are removed to filter out incorrect long-range connections, creating a refined edge set $\mathcal{E}' = \{e \mid l(e) \leq \bar{l}\}$. On the updated graph $\mathcal{M}' = (\mathcal{V}, \mathcal{E}')$, we recalculate node degrees to determine local density:

$$\bar{d} = \frac{1}{n} \sum_{v \in \mathcal{V}} |\mathcal{N}'(v)|, \tag{2}$$

where $n = |\mathcal{V}|$ denote the number of nodes in the mesh and $\mathcal{N}'(v)$ denote the set of neighbors of node v. Nodes are classified as $\mathcal{V}_\mathrm{d}$ if $d'(v) > \bar{d}$, and $\mathcal{V}_\mathrm{s}$ otherwise.

Given a coarsening ratio U (e.g., $U = 0.5$ for 50% downsampling) and a sampling proportion m for dense regions, we define the number of sampled nodes in dense and sparse regions as:

$$k_d = \lfloor U \cdot m \cdot n \rfloor, \quad k_s = \lfloor U \cdot (1 - m) \cdot n \rfloor, \tag{3}$$

where k_d and k_s represent the number of nodes sampled from dense and sparse regions, respectively. The combined sampled node set is then defined as:

$$\begin{cases} V_d = \{v_i \mid v_i \sim \mathcal{U}(\mathcal{V}_\mathrm{d})\}, & |V_d| = k_d, \\ V_s = \{v_i \mid v_i = \mathrm{FPS}(\mathcal{V}_\mathrm{s})\}, & |V_s| = k_s, \end{cases} \tag{3}$$

Here, $\mathcal{U}(\mathcal{V}_\mathrm{d})$ denotes random sampling from dense regions, and $\mathrm{FPS}(\mathcal{V}_\mathrm{s})$ denotes farthest point sampling based on Euclidean distance. Finally, the full sampling set is $S_i = V_d \cup V_s$. As illustrated in Fig.3, our hierarchical sampling strategy effectively distinguishes regions requiring detailed local representation from those reflecting global mesh structure, preserving critical geometric details and avoiding errors during coarsening.

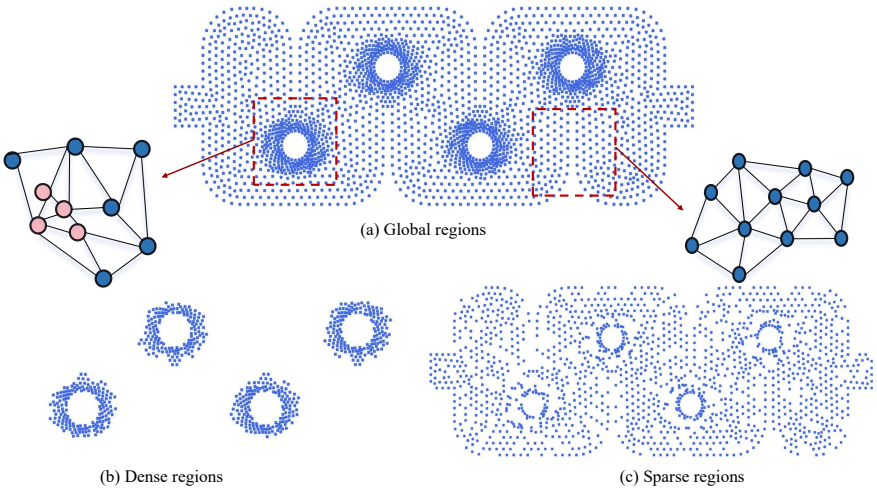


Fig. 3: Illustration of the hierarchical density-based sampling strategy applied to Pipe Turbulence.

3.3 Neighborhood Definition

To reduce computational overhead and enhance local feature aggregation, we define fixed-size neighborhood structures. In traditional GNNs such as GCN [31], stacking K message-passing layers causes each node to aggregate information from an increasingly large receptive field, potentially involving $O(d^K)$ nodes, where d is the node degree. While the per-layer complexity is linear in the number of edges, this cumulative growth increases per-node cost and can lead to over-smoothing.

Inspired by GraphSAGE [21], we instead sample a fixed number of neighbors S per node within 1-hop or 2-hop distance. This constrains the per-node cost to $O(K \cdot S)$ and the total model complexity to $O(K \cdot S \cdot n)$, where n is the number of nodes. Fixed-size neighborhoods improve scalability and ensure consistent computation across varying graph densities.

3.4 Constrained Delaunay Reconstruction

In graph-based mesh processing, accurate edge reconstruction at coarser scales is crucial for preserving geometric and topological integrity. Traditional methods rely on spatial proximity[17,18,19] or retain edges from sampled nodes[12], but often generate incorrect edges across boundaries, disrupting connectivity.

To address these issues, we propose an adaptive Constrained Delaunay Reconstruction method tailored for multi-scale edge reconstruction, inspired by classical computational geometry techniques that robustly handle non-convex domains and regions with holes[32]. Specifically, we leverage prior mesh information to define a polygonal boundary $\mathcal{B}$, encoding the original mesh's geometry

and topology. Given a sampled node set (V'), we construct edges using standard Delaunay triangulation $T(V')$ and validate each candidate edge $e = (u, v)$ against the constraints:

$$E' = \{e \mid e \subseteq \mathcal{B}, e \in \mathcal{T}(V')\}. \tag{4}$$

Only edges completely contained within the constrained polygonal region are retained. This ensures robust edge reconstruction, preventing invalid connections and preserving mesh quality.

4 Multi-Scale Point-Based GNN(MPG)

In this section, we will introduce MPGNN, a hierarchical GNN where the multi-level structure has been determined by the input mesh and the preprocessing in Sec.3.

4.1 Problem

Conventional GNNs and multi-scale GNNs rely heavily on stacked message passing (MP) layers to aggregate node and edge features:

$$h_v^{(k)} = \text{AGGREGATE}\left(\{h_u^{(k-1)}, e_{uv}^{(k-1)} \mid u \in \mathcal{N}(v)\}\right), \tag{5}$$

where $h_v^{(k)}$ is the embedding of node v at layer k, and $e_{uv}^{(k-1)}$ is the edge embedding between nodes u and v. However, stacking MP layers significantly increases computational complexity and redundancy, especially for large-scale non-uniform meshes. Additionally, the fixed high-dimensional embeddings (e.g., 128 dimensions) further exacerbate computational overhead.

Inspired by PointMLP[22], which achieves efficient feature extraction on point clouds without complex local operations, we revisit mesh feature learning. Non-uniform meshes structurally resemble point clouds; thus, heavy reliance on MP layers may be unnecessary. We aim to replace stacked MP layers with lightweight residual MLP modules, efficiently extracting node features and improving scalability.

4.2 Encoder

The input is multi-level mesh structure $\{\mathcal{M}_l = (\mathcal{V}_l, \mathcal{E}_l)\}_{l=1}^{L}$. For each mesh level, the encoding operation can be formulated as:

$$g_i = \Phi_{\text{pos}}\left(\mathcal{A}\left(\Phi_{\text{pre}}\left(\mathcal{V}_i, \mathcal{V}_{ij}\right)\right)\right), \tag{6}$$

where $\Phi_{\text{pre}}(\cdot)$ and $\Phi_{\text{pos}}(\cdot)$ represent residual MLP blocks adapted from PointMLP[22]. Specifically, $\Phi_{\text{pre}}(\cdot)$ learns initial node embeddings within local receptive fields,

capturing preliminary geometric patterns. For neighborhood aggregation $\mathcal{A}$, we adopt a learnable weighting mechanism inspired by CNN receptive fields:

$$\mathcal{A}(f_i) = \sum_{j \in \mathcal{N}(i)} (W_{i,j} \cdot w_{ij}) \cdot f_j, \quad w_{ij} = \frac{\exp(-||\mathbf{x}_i - \mathbf{x}_j||_2)}{\sum_{k \in \mathcal{N}(i)} \exp(-||\mathbf{x}_i - \mathbf{x}_k||_2)}. \tag{7}$$

where f_j denotes neighbor features, W denotes learnable parameters, and w_{ij} dynamically adjusts neighbor importance based on spatial relationships.

Finally, the aggregated features are refined by another residual MLP block $\Phi_{\text{pos}}(\cdot)$. Formally, the residual MLP block can be expressed as:

$$\Phi(x) = act(W_2 \cdot \text{act}((W_1 \cdot x)))) + x, \tag{8}$$

where W_1 and W_2 are learnable parameters, act denotes nonlinear activation (e.g., PReLU).

By explicitly employing lightweight residual MLP-based feature extraction, the encoder effectively reduces computational overhead and improves dimension scalability. This design avoids deep message passing stacks, significantly enhancing efficiency and generalization on various non-uniform meshes.

4.3 Decoder

For decoder module, we adopt a k-nearest neighbor (KNN) interpolation to efficiently recover fine-level node features[27,28]. For each node x_i in the fine-layer X_f, we identify its k closest nodes from the coarse layer X_c based on Euclidean distance. The interpolated feature x_i is computed by distance-weighted averaging:

$$x_i = \sum_{j \in \mathcal{N}_k(i)} \frac{w_{ij} h_j}{\sum_{j \in \mathcal{N}_k(i)} w_{ij}}, \quad w_{ij} = \frac{1}{d(P_f^i, P_c^j)}, \tag{9}$$

where h_j denotes the coarse-layer feature of neighbor node j, and $d(P_f^i, P_c^j)$ denotes the Euclidean distance between fine-layer node i and coarse-layer node j.

Additionally, we incorporate skip-connections between corresponding encoder and decoder layers, facilitating efficient spatial information transfer. A residual MLP $\Phi(x)$ is further employed to refine interpolated features, enhancing reconstruction accuracy and ensuring structural consistency across mesh resolutions. This encoder-decoder architecture is specifically designed for static multi-scale feature reconstruction, providing a efficient foundation for temporal dynamics prediction in physical simulations.

5 Experiments

5.1 Experimental Setup

Datasets We evaluate our MPG model on five datasets covering various physical domains: *Cylinder Flow[14]* simulates incompressible fluid flow around a

cylinder. *Pipe Turbulence[33]* represents turbulent flow dynamicly within a complex pipe. *Thermochemical Reaction[34]* involves multi-physics complex model to simulate the flow, including fluid flow, heat transfer, mass transfer, and chemical reaction within a thermochemical heat-storage reactor. *Lid2d[35]* focus on incompressible lid-driven cavity flow with varying viscosity and boundary conditions. *NSM2d[35]* deals with incompressible fluid dynamics with complex boundary conditions and initial states.

Baselines We benchmark MPG against five baselines. *MeshGraphNets[10]* is a foundational graph-based model for mesh representation, evaluated with noise injection(NI). *Graph U-Net[15]* employs learnable graph sampling within a U-Net architecture, specifically tailored to graph structure. *PointMLP[22]* is a purely point-based method for feature extraction without topology structure. *AMGNET[20]* integrates algebraic multigrid algorithms within GNNs graph coarsening. *BSMS-GNN[12]* introduces a novel bi-stride sampling strategy for multi-scale mesh representation. And to our knowledge, it's the current state-of-the-art (SOTA) model.

Evaluation Protocol We evaluate MPG through two experiments: *1) Feature Reconstruction Evaluation:* We compare MPG with PointMLP, BSMS - GNN, and AMGNET on Cylinder Flow, Pipe Turbulence, and Thermochemical Reaction datasets. Based on multi-scale meshes $\{\mathcal{M}_l^t = (V_l^t, \mathcal{E}_l^t)\}_{l\in(1,L)}^{t\in(1,T)}$ (L multi-scale layers, T time steps), MPG encodes features via the Encoder and reconstructs via the Decoder. *2) Rollout Evaluation:* To assess MPG's generalization in non-uniform mesh simulations, we conduct rollout experiments on long-trajectory Cylinder Flow, Lid2d, and NSM2d datasets with MeshGraphNets, AMGNET, and BSMS-GNN. As MPG focuses on feature reconstruction without explicit temporal modeling, we use BSMS-GNN's Message Passing layer for prediction from $\mathcal{M}_l^t$ to $\mathcal{M}_l^{t+1}$. Both MPG and BSMS-GNN use two multi-scale layers with a fixed 50% downsampling ratio per layer.

In training process, we use both the L2 loss (Eq.(10)) for model optimization. In feature reconstruction, y is from the original $\mathcal{M}_0^t$ and $\hat{y}$ from the reconstructed one. In rollout, y is the ground truth of $\mathcal{M}_l^{t+1}$ and $\hat{y}$ its prediction. For evaluation, RMSE is the primary metric.

$$L_{l2} = \frac{1}{D}\sum_{k=1}^{D}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{m}\sum_{i=1}^{m}\frac{\|y_{k,i}^t - \hat{y}_{k,i}^t\|_2}{\|y_{k,i}^t\|_2} \tag{10}$$

5.2 Results and Discussion

According to the description of 5.1, we evaluate in two distinct tasks: feature reconstruction and rollout prediction. The key observations from these evaluations are summarized as follows:

Table 1: Feature Reconstruction RMSE ($\times 10^{-4}$) comparison on three datasets. Lower is better. Best result highlighted in bold.

Model	Cylinder Flow	Pipe Turbulence	Thermochemical Reaction
PointMLP	52.50	60.52	109.45
AMGNET	15.76	17.63	22.17
BSMS-GNN	3.23	3.32	3.96
MPG (ours)	**2.72**	**1.74**	**3.94**

Feature Reconstruction Accuracy. Tab.1 summarizes the RMSE results for the feature reconstruction task conducted on Cylinder Flow, Pipe Turbulence, and Thermochemical Reaction datasets. All models demonstrate good reconstruction performance (RMSE$<10^{-3}$), while MPG achieves comparable or better accuracy. This highlights that our proposed multi-scale encoding strategy effectively captures essential geometric and physical features, despite using significantly fewer parameters.

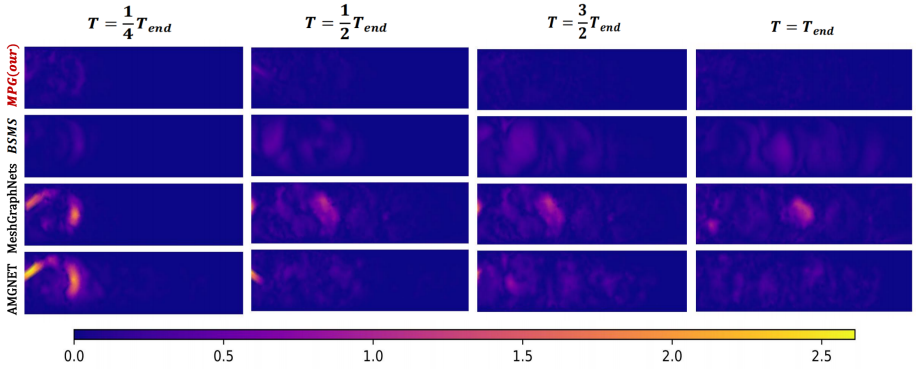


Fig. 4: Rollout Error Snapshots of MeshGraphNets, AMGNET, BSMS, and Our Proposed Model (MPG) on the NSM2d Dataset.

Rollout Prediction and Error Accumulation. For the rollout prediction task (Tab.2), conducted on Cylinder Flow, Lid2d, and NSM2d datasets, our MPG model exhibits notably lower error accumulation over the trajectory compared to baselines, particularly on complex mesh structures. Even though single-step prediction errors for Cylinder Flow and NSM2d are similar to BSMS-GNN, MPG achieves these results with approximately only 15% of its parameter count. Fig.4 illustrates the error accumulation over the trajectory for different models, emphasizing the superior performance of MPG in maintaining lower error rates over time.

Computational Efficiency. Notably, the statistics for training and inference time exclude the multi-scale construction, which acts as an independent preprocessing module. As summarized in Tab.2, MPG achieves significant improvements in computational efficiency. Specifically, our method reduces training time

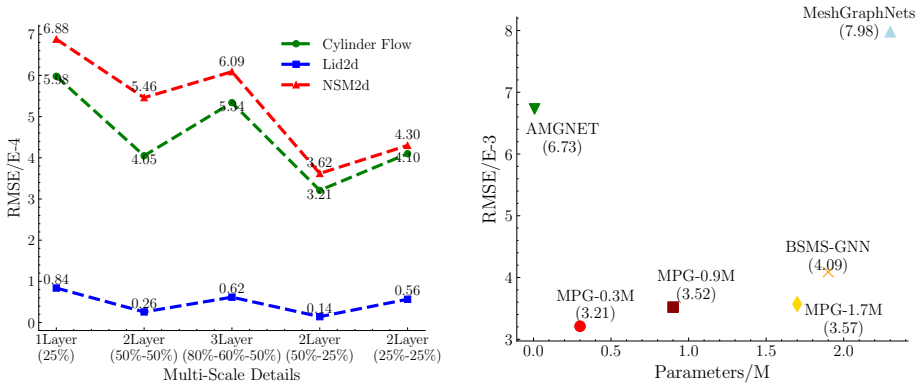
per epoch by 63.1% $\sim$ 86.1% compared to BSMS-GNN, 17.7% $\sim$ 65.3% compared to MeshGraphNets, and 45.0% $\sim$ 73.8% compared to AMGNET across different datasets. Additionally, MPG exhibits a substantial reduction in inference time in the Cylinder Flow and Lid2d datasets. Regarding memory usage, MPG requires less memory than MeshGraphNets and BSMS-GNN. MPG also maintains a lightweight model size, reducing the parameter count by **87.3%** compared to MeshGraphNets and **84.5%** compared to BSMS-GNN. Even compared to the most parameter-efficient AMGNET model, MPG strikes a better balance between model size and predictive accuracy, as AMGNET's extreme parameter reduction leads to higher rollout errors. These gains are enabled by MPG's hierarchical multi-scale construct strategy, adaptive neighborhood aggregation, and lightweight residual MLP framework, establishing it as a new benchmark for scalable and efficient simulations on non-uniform meshes. In summary, the proposed MPG framework demonstrates superior efficiency and accuracy, effectively balancing model compactness, training speed, and predictive performance, setting a new standard for efficient multi-scale GNN simulations.

Table 2: Performance comparison on Rollout Task. Best results are in **bold**, second-best results are underlined.

Measurements	**Case**	**MPG (Ours)**	MeshGraphNets	BSMS	AMGNET
RMSE-1 ($\times 10^{-3}$)	Cylinder Flow	4.98	7.98	**4.09**	9.71
	Lid2d	**0.26**	1.22	0.35	0.95
	NSM2d	5.75	23.50	**5.46**	45.9
RMSE-50 ($\times 10^{-2}$)	Cylinder Flow	**3.71**	5.05	4.56	7.34
	Lid2d	**0.95**	4.66	1.13	1.76
	NSM2d	**15.81**	21.19	20.29	18.24
RMSE-all ($\times 10^{-2}$)	Cylinder Flow	**53.16**	63.95	57.27	62.82
	Lid2d	**7.15**	12.04	8.76	9.61
	NSM2d	**30.07**	35.33	33.34	32.49
Training time (s/epoch)	Cylinder Flow	**160.83**	209.26	435.49	440.08
	Lid2d	**140.13**	403.52	647.49	254.96
	NSM2d	**148.71**	180.61	1071.72	568.16
Inference time (ms/step)	Cylinder Flow	**14.13**	16.74	67.45	56.76
	Lid2d	**9.54**	12.42	72.76	10.42
	NSM2d	17.99	**17.88**	136.59	59.14
Training RAM (MiB)	Mean	8964	22583	12572	**6882**
Params (#)	All	296,755	2,331,778	1,917,442	**7,168**

5.3 Ablation Study

Network Details Study We investigate how network depth and sampling ratios influence reconstruction performance, as illustrated in Fig.5(a). Specifically, we analyze two key aspects: 1) varying the number of coarse-graining layers while keeping the overall sampling ratio fixed, 2) adjusting the sampling ratios within



(a) (b)

Fig. 5: (a) RMSE across different datasets under varying multi-scale settings. (b) RMSE across different hidden state dimensions with various parameters.

a fixed two-layer structure. Results demonstrate that a two-layer structure consistently yields better reconstruction accuracy compared to deeper hierarchical designs. Furthermore, within the two-layer setting, a $50\% - 25\%$ sampling ratio slightly outperforms the standard $50\% - 50\%$ ratio, which used for fair and direct comparisons with some fix-size baselines in main experiments.

Parameter Study We further evaluate how parameters influence MPG's reconstruction accuracy. Fig.5(b) demonstrates that MPG achieves better accuracy and computational efficiency without relying on deep MP stacks. Notably, MPG consistently outperforms baseline models across all parameter scales, highlighting the effectiveness of our design even at smaller model sizes.

6 Conclusion

In this work, we introduced MPG, a novel multi-scale point-based GNN designed for efficient and accurate simulation on non-uniform meshes. MPG integrates hierarchical density-based sampling, constrained Delaunay reconstruction, and lightweight feature aggregation to balance computational efficiency and predictive accuracy. By drawing inspiration from point cloud processing and constrained Delaunay triangulation, MPG constructs robust multi-scale representations while preserving essential geometric and topological structures. Experimental results demonstrate that MPG significantly reduces computational overhead, achieving up to 86.1% faster training, lower inference time, and reduced memory consumption, while maintaining high single-step rollout accuracy across each tasks. These results highlight MPG as an efficient and scalable solution for non-uniform mesh simulations, enabling high-fidelity modeling of complex physical systems. Future work could focus on optimizing memory-efficient architectures to support training on even larger-scale meshes with higher resolution.

References

1. Haixin Wang, Yadi Cao, Zijie Huang, Yuxuan Liu, Peiyan Hu, Xiao Luo, Zezheng Song, Wanjia Zhao, Jilin Liu, Jinan Sun, et al. Recent advances on machine learning for computational fluid dynamics: A survey. *arXiv preprint arXiv:2408.12171*, 2024.
2. Olek C Zienkiewicz and Robert L Taylor. *The finite element method set*. Elsevier, 2005.
3. Octavi Obiols-Sales, Abhinav Vishnu, Nicholas Malaya, and Aparna Chandramowliswharan. Cfdnet: A deep learning-based accelerator for fluid simulations. In *Proceedings of the 34th ACM international conference on supercomputing*, pages 1–12, 2020.
4. Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
5. Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.
6. Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *International conference on machine learning*, pages 3424–3433. PMLR, 2017.
7. Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. In *Computer graphics forum*, volume 38, pages 59–70. Wiley Online Library, 2019.
8. Stathi Fotiadis, Eduardo Pignatelli, Mario Lino Valencia, Chris Cantwell, Amos Storkey, and Anil A Bharath. Comparing recurrent and convolutional neural networks for predicting wave propagation. *arXiv preprint arXiv:2002.08981*, 2020.
9. Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.
10. Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.
11. Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.
12. Yadi Cao, Menglei Chai, Minchen Li, and Chenfanfu Jiang. Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network. In *International Conference on Machine Learning*, pages 3541–3558. PMLR, 2023.
13. Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. Multiscale meshgraphnets. *arXiv preprint arXiv:2210.00612*, 2022.
14. Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9:e15, 2020.
15. Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.

16. Wenzhuo Liu, Mouadh Yagoubi, and Marc Schoenauer. Multi-resolution graph neural networks for pde approximation. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III 30*, pages 151–163. Springer, 2021.
17. Mario Lino, Stathi Fotiadis, Anil A Bharath, and Chris D Cantwell. Multi-scale rotation-equivariant graph neural networks for unsteady eulerian fluid dynamics. *Physics of Fluids*, 34(8), 2022.
18. Mario Lino, Chris Cantwell, Anil A Bharath, and Stathi Fotiadis. Simulating continuum mechanics with multi-scale graph neural networks. *arXiv preprint arXiv:2106.04900*, 2021.
19. Mario Lino, Stathi Fotiadis, Anil A Bharath, and Chris Cantwell. Towards fast simulation of environmental fluid mechanics with multi-scale graph neural networks. *arXiv preprint arXiv:2205.02637*, 2022.
20. Zhishuang Yang, Yidao Dong, Xiaogang Deng, and Laiping Zhang. Amgnet: multi-scale graph neural networks for flow field prediction. *Connection Science*, 34(1):2500–2519, 2022.
21. Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
22. Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.
23. Qingqi Zhao, Xiaoxue Han, Ruichang Guo, and Cheng Chen. A computationally efficient hybrid neural network architecture for porous media: Integrating convolutional and graph neural networks for improved property predictions. 2023.
24. Xu Han, Han Gao, Tobias Pfaff, Jian-Xun Wang, and Li-Ping Liu. Predicting physics in mesh-reduced space with temporal attention. *arXiv preprint arXiv:2201.09113*, 2022.
25. Yidi Shao, Chen Change Loy, and Bo Dai. Transformer with implicit edges for particle-based physics simulation. In *European Conference on Computer Vision*, pages 549–564. Springer, 2022.
26. Huang Zhang, Changshuo Wang, Shengwei Tian, Baoli Lu, Liping Zhang, Xin Ning, and Xiao Bai. Deep learning-based 3d point cloud classification: A systematic survey and outlook. *Displays*, 79:102456, 2023.
27. Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
28. Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
29. Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.
30. Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630, 2019.
31. Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.

32. L Paul Chew. Constrained delaunay triangulations. In *Proceedings of the third annual symposium on Computational geometry*, pages 215–222, 1987.
33. Gengxiang Chen, Xu Liu, Qinglu Meng, Lu Chen, Changqing Liu, and Yingguang Li. Learning neural operators on riemannian manifolds. *arXiv preprint arXiv:2302.08166*, 2023.
34. Gang Xiao, Zhide Wang, Dong Ni, and Peiwang Zhu. Kinetics and structural optimization of cobalt-oxide honeycomb structures based on thermochemical heat storage. *Energies*, 16(7):3237, 2023.
35. Pengwei Liu, Zhongkai Hao, Xingyu Ren, Hangjie Yuan, Jiayang Ren, and Dong Ni. Papm: A physics-aware proxy model for process systems. *arXiv preprint arXiv:2407.05232*, 2024.