计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(2): 434−447, 2026

一种基于无监督学习的智能并行结构网格生成框架

陈新海[1,2,3] 彭嘉明[1,2,3] 乔 鹏[2,3] 贾孟涵[2,3] 王庆林[1,2,3] 张 翔[1,2,3] 杨 博[1,2,3] 刘 杰[1,2,3]

[1](高端装备数字化软件湖南省重点实验室(国防科技大学) 长沙 410073)

[2](并行与分布计算全国重点实验室(国防科技大学) 长沙 410073)

[3](国防科技大学计算机学院 长沙 410073)

(chenxinhai16@nudt.edu.cn)

An Intelligent Parallel Structured Mesh Generation Framework Based on Unsupervised Learning

Chen Xinhai[1,2,3], Peng Jiaming[1,2,3], Qiao Peng[2,3], Jia Menghan[2,3], Wang Qinglin[1,2,3], Zhang Xiang[1,2,3], Yang Bo[1,2,3], and Liu Jie[1,2,3]

[1](*Laboratory of Digitizing Software for Frontier Equipment* (*National University of Defense Technology*), *Changsha* 410073)

[2](*National Key Laboratory of Parallel and Distributed Computing* (*National University of Defense Technology*), *Changsha* 410073)

[3](*College of Computer Science and Technology*, *National University of Defense Technology*, *Changsha* 410073)

Abstract With the rapid development of high-performance computing technologies, the complexity and scale of scientific computing problems have significantly increased. As a critical input for scientific computing, mesh generation has emerged as an important research direction in high-performance computing. To address challenges such as low computational efficiency and complex human-machine interaction in large-scale mesh generation, intelligent mesh generation methods have attracted increasing attention. However, achieving a deep integration of high-performance computing and artificial intelligence in mesh generation remains a critical challenge. To overcome these challenges, an intelligent parallel mesh generation framework based on unsupervised learning is proposed, enabling the efficient generation of large-scale multi-block structured meshes. The framework integrates a block partitioning method based on cross-fields, a Fourier Kolmogorov-Arnold network model, and a mesh alignment technique. It employs an unsupervised learning approach to adaptively learn high-quality mesh partitioning rules, effectively overcoming the limitations of existing intelligent methods in handling complex geometries. Additionally, a novel parallel transfer learning mechanism is introduced, significantly enhancing model convergence efficiency by transferring task similarity features across different processes. Experimental results demonstrate that the proposed framework achieves up to a 10-fold improvement in efficiency over traditional mesh generation methods for hundred-million-scale tasks. Furthermore, it outperforms existing intelligent mesh generation methods across five different quality metrics, achieving state-of-the-art performance.

Key words high-performance computing; unsupervised learning; parallel mesh generation; Fourier Kolmogorov-Arnold network; transfer learning

收稿日期: 2025-03-01; **修回日期**: 2025-08-27

基金项目: 国家重点研发计划项目(2023YFB3001903, 2021YFB0300101); 国家自然科学基金项目(12402349); 湖南省自然科学基金项目(2024JJ6468); 国防科技大学青年自主创新科学基金项目(ZK2023-11); 并行与分布计算全国重点实验室开放基金(WDZC20255290101)

This work was supported by the National Key Research and Development Program of China (2023YFB3001903, 2021YFB0300101), the National Natural Science Foundation of China (12402349), the Natural Science Foundation of Hunan Province (2024JJ6468), the Youth Foundation of the National University of Defense Technology (ZK2023-11), and the Open Fund of National Key Laboratory of Parallel and Distributed Computing (WDZC20255290101).

摘　要　随着高性能计算技术的迅猛发展，科学计算问题的复杂度和计算规模显著提升。网格生成作为科学计算的前置输入，是高性能计算领域的重要研究方向。针对大规模网格生成计算效率低、人机交互复杂等难题，发展智能化网格方法已成为研究热点，但如何在网格领域实现高性能计算与人工智能深度融合仍处于研究空白。针对上述问题，提出了一种基于无监督学习的智能并行结构网格生成框架，支持大规模多块结构网格的高效生成。框架集成了基于标架场的分块方法、傅里叶柯尔莫哥洛夫-阿诺德网络（Fourier Kolmogorov-Arnold network，傅里叶 KAN）模型和网格对齐方法，采用无监督学习模式，自适应学习高质量网格划分规则，有效解决了已有智能方法在处理复杂外形上的局限性。引入并行迁移学习机制，通过迁移不同进程间网格生成任务的相似性特征，显著提升了模型收敛效率。实验结果表明，所提出的方法在亿维网格生成任务中，相比传统网格生成实现了最高 10 倍的效率提升。同时在 5 个不同质量度量下优于已有智能网格生成方法，达到了最优水平。

关键词　高性能计算；无监督学习；并行网格生成；傅里叶 KAN；迁移学习

中图法分类号　TP391

DOI: 10.7544/issn1000-1239.202550134　　**CSTR:** 32373.14.issn1000-1239.202550134

高性能计算技术的快速发展显著提升了科学与工程计算问题求解能力，使复杂系统的精细化模拟成为可能。作为科学计算的关键前处理步骤，网格生成直接影响了计算的精度与效率。大型工程的数值仿真计算，如航空航天飞行器设计、重大装备设计、复杂电磁环境模拟、气候气象模拟，均采用基于网格离散的数值计算方法进行求解[1-3]。随着问题复杂度和计算规模显著提升，所需的网格规模呈指数级增加，且复杂网格生成质量控制需要频繁地人机交互，导致大规模网格生成的开销已成为制约科学计算效率的关键瓶颈，迫切需要发展高效的网格生成技术。

随着智能技术的发展，以神经网络为代表的人工智能技术作为一类万能函数逼近工具在科学与工程计算领域的智能化、自动化发展方面有着极大的应用空间，研究智能赋能的科学计算方法（AI for science）成为研究热点[4]。越来越多的研究人员已将这些方法应用于科学计算的核心环节，包括求解偏微分方程[5-6]、网格生成与质量评估[7-8]、流场预测问题[9]等。人工智能与传统科学计算的融合成为一种新的趋势，以突破计算的瓶颈。

在网格生成领域，已有学者开始采用有监督方法研究智能网格生成技术，通过构建大量网格数据，实现自动网格生成。MeshONet[10] 需要手动构建高质量涵盖不同外形的网格数据集，利用算子学习和双分支神经网络生成网格；ISpliter[11] 通过递归生成大量网格数据，采用有监督学习训练网络，通过预测分割线将几何模型划分为网格；Papagiannopoulos 等人[12]构建了包含不同边数的网格数据集，使用 3 个神经网络从海量数据中学习网格生成的特征并分别预测网格的顶点数量、位置和连接关系，用于训练的数据样本基于 Delaunay 三角剖分算法批量生成。上述方法均依赖于大量训练数据，但获取高质量网格数据需耗费大量人工成本与计算资源。

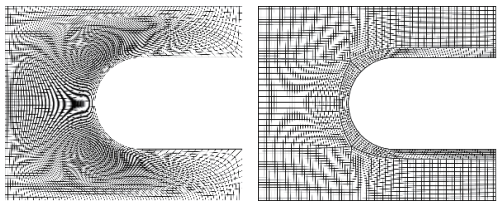
为了缓解对海量训练样本的依赖，Chen 等人[13]提出 MGNet，采用基于物理信息的无监督学习方式，将方程和边界项嵌入损失函数来约束训练，支持输入目标外形生成相应的结构网格，无需预先生成额外的网格数据。Peng 等人[14]将该方法扩展至 3 维网格生成领域，支持输入外形到目标网格的快速生成，但仅能在简单外形上生成网格。基于物理信息的方法虽消除了对训练数据的依赖，但是由于没有对复杂外形进行分块，因此在处理复杂几何体时存在收敛速度慢、训练开销大的问题。此外，已有的智能方法都尚未有效整合高性能计算资源和并行计算技术，如何在网格领域实现高性能计算与人工智能深度融合仍处于研究空白。需要借助高性能计算架构实现训练加速以及结合并行计算提升训练效率。

针对上述挑战，本文提出了一种基于无监督学习的智能并行网格生成框架，支持大规模多块结构网格的高效生成。框架集成了基于标架场的分块方法、傅里叶柯尔莫哥洛夫-阿诺德网络（Fourier Kolmogorov-Arnold network，傅里叶 KAN）模型和网格对齐方法，采用无监督学习模式，自适应学习高质量网格划分规则，有效解决了已有智能方法在处理复杂外形上的局限性。引入并行迁移学习机制，通过迁移不同进程间网格生成任务的相似性特征加速训练。实验结果表明，本框架在千万维网格生成任务中，相比传统网格生成方法实现了 1 个数量级以上

的效率提升，同时在网格质量方面优于已有智能网格生成方法，达到了最优水平。

1 相关工作

在高性能计算驱动的复杂物理场模拟中，网格直接影响计算结果的准确性和计算效率，是高性能计算领域的重要研究内容。结构网格是数值计算中常用的网格，具有求解精度高、所需单元少等特点，常用于处理边界层[15]、尾迹和激波[16]等复杂流动特征的模拟。常见的结构网格生成方法包括代数法[17]和基于偏微分方程(partial differential equation, PDE)的方法[18]。代数法中最常用的是超线性插值(transfinite interpolation, TFI)[19]，该方法用已知的边界值进行中间插值来生成内部网格，能够快速生成网格，但在复杂几何区域可能导致网格单元重叠或网格超出边界[20]。基于PDE的方法计算开销较大及网格生成时间较长，在大规模网格生成中效率较低[21]。这2种方法通常用于在规则的几何上生成网格，直接用于复杂外形的网格生成时往往会导致网格变形。如图1(a)所示，由于外形限制，直接生成的网格出现了明显的变形。



(a) 结构网格 (b) 多块结构网格

Fig. 1 Structured mesh and multiblock-structured mesh

图1 结构网格和多块结构网格

为解决上述问题，已有学者采用基于标架场的区域分块方法，将复杂的几何外形划分为更简单的子区域即“块”[22]。如图1(b)所示，每个块为一个四边形区域，块内由结构网格填充，形成多块结构网格。可以将子块分配给多个计算节点，实现并行计算和分布式计算，提高计算速度和可扩展性，同时保证网格质量。然而多块结构大规模网格生成效率仍然受限，需要结合智能方法提高网格生成效率。

近年来，一些学者尝试将深度学习引入网格生成领域，基于网格数据或物理信息训练模型，实现网格的自动生成并提升网格生成效率。ISpliter[11]通过递归生成230 037个样本数据用于训练神经网络并预测分割线的位置，通过分割线将几何模型划分为网格，但对训练数据的质量比较敏感，在低质量训练数据下表现不佳。MeshONet[10]将算子扩展到多维的输入输出，通过双分支网络架构训练生成网格，对于不同的外形需人工构建大量高质量的网格数据集。Papagiannopoulos等人[12]提出了通过3个神经网络从大量数据中学习网格生成特征，分别预测网格的顶点数量、位置和连接关系。对于具有4条边的外形，训练大约需要6 000个样本，而对于具有8条边的外形，所需的训练样本数量则增至24 000个。Kim等人[23]采用图卷积神经网络，通过学习局部误差密度来预测网格的边长，并生成16 000个控制网格和特征网格进行训练。上述方法的数据依赖性强、数据准备开销大。

为缓解智能网格生成方法对训练数据的依赖，受无监督学习特性的启发，MGNet[13]基于物理信息生成网格。该方法将网格生成所需的边界条件和方程项嵌入到损失函数中，经过训练即可生成网格。整个过程无需事先生成网格训练数据，但该方法在复杂外形上生成的网格质量较差。MeshNet[24]通过结合微分方程、边界条件和粗网格生成的先验数据来训练神经网络和学习网格生成参数。然而该方法仍依赖人工生成粗网格，限制了自动化程度。由于基于多层感知机的网络在学习非线性映射时表现较弱，KANMeshNet[25]采用KAN[26]作为骨干网络替代多层感知机，能在简单几何外形下生成2维和3维结构网格。尽管基于物理信息的网格生成方法消除了对训练数据的依赖，但仍存在2个问题：一是没有对复杂外形进行划分分块，因此在处理复杂几何体时存在收敛速度慢、训练开销大的问题；二是并行程度较低、可扩展性差的问题，模型训练效率较低，未能充分利用高性能计算系统的并行能力。因此有必要对外形分块，并结合高性能计算架构实现训练加速，通过并行计算提升训练效率。

为了解决上述问题，本文结合基于标架场的区域分块方法，首先将复杂的几何外形划分为多个子块，使得各个子块的网格生成任务可以并行计算，提升训练并行效率。同时，为了解决单个块任务训练效率受限制的问题，本文引入迁移学习，在不同子块之间共享模型参数以加速训练。

2 智能并行网格生成框架

2.1 方法概述

本节简要介绍基于无监督学习的智能并行结构

网格生成框架的工作流程。本框架以几何外形文件作为输入，首先将外形进行分块，并在各块之间实现数据并行。接着搭建 MultiMeshNet 学习网格生成规则，通过在多个进程间共享网格生成任务的相似性特征提升训练效率，最终输出多块结构网格。

MultiMeshNet 模型设计如图 2 所示，模型中的神经元被用来自然地编码底层物理定律，损失函数中嵌入了与网格生成相关的方程和边界条件。整个训练过程是无监督的，无需先验知识或标注的网格样本数据集。在训练过程中每个进程都包含 1 个独立的 MultiMeshNet 模型参数，分别在各自的块上训练网络，并结合并行训练和迁移学习机制，提高多块结构网格生成的效率和精度。整个框架的工作流程如下：

1）接收输入的几何模型文件。

2）采用基于标架场的区域分块方法对几何模型进行分解以形成多个块。

3）将分解后的块数据分发（MPI_Scatterv）到不同的进程，每个进程接收相应的分块数据。

4）每个进程在其分配到的块上训练 1 个子网络，各进程之间并行训练。

5）在训练完成后，各进程并行地在不同块上进行前向推理以生成网格。

6）聚合（MPI_Gatherv）网格数据。

7）对齐各进程生成的块网格，得到原始形状上的完整网格。

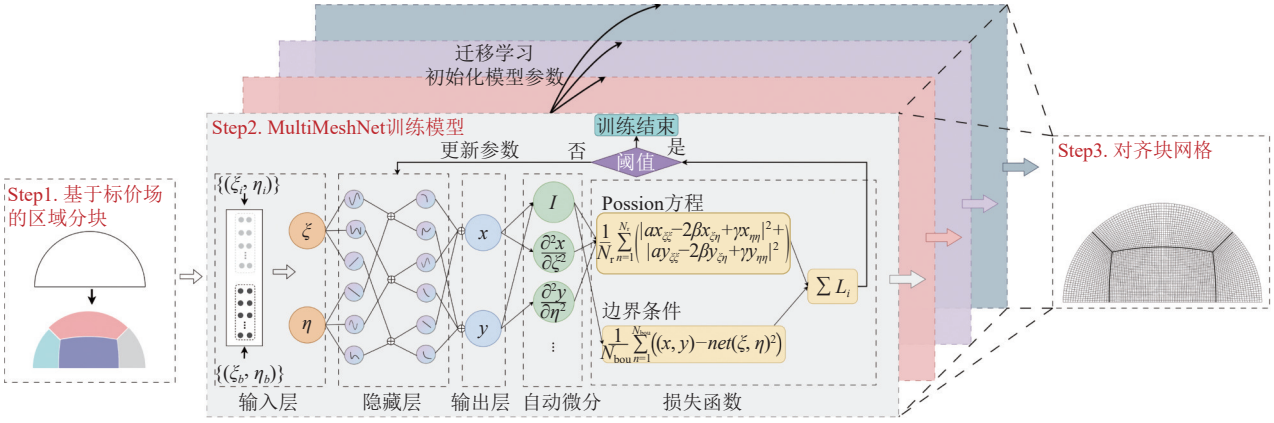


Fig. 2 The steps of intelligent parallel mesh generation and MultiMeshNet model

图 2 智能并行网格生成步骤以及 MultiMeshNet 模型

2.2 多块结构网格生成

2.2.1 基于标架场的区域分块

由于直接在复杂外形上生成网格往往难以保证网格质量，并且不利于数据并行和算法并行设计，因此框架首先对复杂的外形进行区域分块。分块后，可以将子块数据分配给多个计算节点，从而提高并行度和可扩展性。

图 3 展示了基于标架场对给定外形进行分块的主要步骤。首先，将外形的内部区域Ω离散化为三角形网格来作为标架场的背景网格。如图 3（a）所示，该方法以非结构的背景网格为输入，然后在背景网格的节点上计算标架场，在图 3（b）中展示了网格节点上的标架场。下一步分析标架场中的奇异点位置，通过迭代所有网格单元，计算每个三角形单元 3 个顶点标架场的旋转度来识别奇异点。对于识别出含有奇异点的网格单元，通过插值获得其单元内部的标架场，当标架场值为 0 时即为奇异点，图 3（c）中以圆点标示了奇异点。接下来从奇异点出发，构建图 3（d）中的流线，流线将外形划分为多个四边形子块，这些子块可用于后续的网格构建。

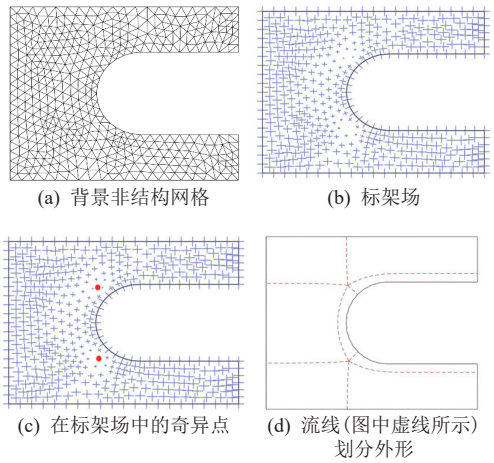


Fig. 3 Domain partitioning steps based on the cross-field

图 3 基于标架场的区域分块步骤

2.2.2 基于无监督学习的损失函数的构造

PDE 方法生成网格将结构网格的生成过程定义

为在微分几何框架下求 PDE 的边值问题。传统基于椭圆型 PDE 方程的网格生成方法通过建立物理空间和计算空间的一对一映射进而生成网格。以图 4 所示的 2 维问题为例描述这种映射关系。图 4 中的计算空间表示一个包含变量ξ和η的单位正方形，通常是一个规则且简化的空间，便于进行数值计算。计算空间可以表示为$(\xi,\eta)\subset[0,1]\times[0,1]$。物理空间是指研究的物理问题所在的空间。

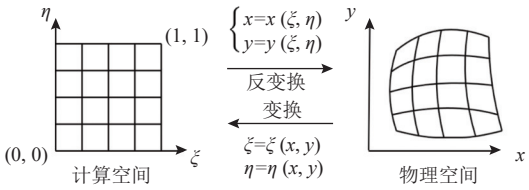


Fig. 4 The principle of structured mesh generation based on PDE method

图 4 基于 PDE 方法的结构网格生成原理

不规则的物理空间变换到规则的计算空间可用式(1)等价表示。

$$\begin{cases}\xi_{xx}+\xi_{yy}=0,\\ \eta_{xx}+\eta_{yy}=0。\end{cases}\tag{1}$$

式(1)反变换的形式为

$$\begin{aligned}&\alpha x_{\xi\xi}-2\beta x_{\xi\eta}+\gamma x_{\eta\eta}=0,\\ &\alpha y_{\xi\xi}-2\beta y_{\xi\eta}+\gamma y_{\eta\eta}=0,\\ &(\xi,\eta)\in\Omega,\end{aligned}\tag{2}$$

其中

$$\begin{aligned}&\alpha=x_\eta^2+y_\eta^2,\\ &\beta=x_\xi x_\eta+y_\xi x_\eta,\\ &\gamma=x_\xi^2+y_\xi^2,\end{aligned}\tag{3}$$

x_ξ表示x对ξ的 1 阶偏导数；x_η表示x对η的 1 阶偏导数；$x_{\xi\xi}$表示x对ξ的 2 阶偏导数；$x_{\xi\eta}$表示x先对ξ求偏导，再对η求偏导，依次类推。上述变换和反变换的详细推导见文献[18]。

而神经网络可以简化上面的步骤，直接应用神经网络学习计算空间到物理空间的映射，进而生成网格。这个过程可以表示为以计算空间坐标(ξ,η)到物理空间坐标(x,y)的一一映射。即神经网络以(ξ,η)作为输入，以物理空间坐标(x,y)作为输出，此时需要遵循反变换(式(2))。将式(2)嵌入在网络的损失函数中，可以给出以下损失函数：

$$\begin{aligned}L(\boldsymbol{\Phi};x,y)=\frac{1}{N_{\mathrm{r}}}\sum_{n=1}^{N_{\mathrm{r}}}(|\alpha x_{\xi\xi}-2\beta x_{\xi\eta}+\gamma x_{\eta\eta}|^2+|\alpha y_{\xi\xi}-2\beta y_{\xi\eta}+\\ \gamma y_{\eta\eta}|^2)+\frac{1}{N_{\mathrm{bou}}}\sum_{n=1}^{N_{\mathrm{bou}}}\left(|(x,y)-net(\xi,\eta)|^2\right),\end{aligned}\tag{4}$$

其中N_{r}代表控制方程残差点的个数，N_{bou}表示在边界上采样的点的个数，使用均方误差损失。式(4)构造了 MultiMeshNet 基于 Poisson 方程的损失函数。损失函数中的第 1 项使得生成的网格点尽量满足椭圆方程的稳态解，第 2 项使得计算空间的边界在通过神经网络后可以满足给定外形的边界。

2.2.3 MultiMeshNet 网络设计

多层感知机网络在处理复杂几何时，往往难以捕捉到网格生成中的全局规律和细节结构。而 KAN 架构通过引入更为灵活的激活函数，可以更好地捕捉网格生成中的复杂非线性关系。因此本文选用 KAN 架构而非多层感知机作为 MultiMeshNet 的网络结构。

KAN 架构的核心思想是用可学习的激活函数替代传统的线性权重，从而实现更复杂的非线性变换。KAN 将权重参数由 1 维样条函数替代，能够更好地逼近复杂函数。然而样条函数的计算复杂度较高，可能导致训练时间较长，并且在某些节点处无法保证高阶导数的连续性，影响梯度反向传播。为了解决这些问题，本文用 2 维傅里叶函数替代样条函数。傅里叶函数参数化简单，优化更易实现，并且通过全局特征分析能够快速捕捉整体规律，提升稳定性。引入傅里叶函数后，激活函数ϕ被定义为

$$\phi(\boldsymbol{P})=\sum_{d=1}^{D}\sum_{k=1}^{K}\left(\alpha_k^d\cos\left(kP^d\right)+\beta_k^d\sin\left(kP^d\right)\right)+c,\tag{5}$$

其中$\boldsymbol{P}=(\xi,\eta)$为计算空间的 2 维坐标点，$D$为特征维度的数量，$P^d$为$\boldsymbol{P}$的第$d$个特征。$K$是基频分量的个数，决定了每个输入维度中包含的正弦项和余弦项的傅里叶系数个数。α_k^d和β_k^d分别表示可学习的傅里叶系数，c是可学习的偏置项。关于傅里叶 KAN 的详细数学公式见附录 A。

2.2.4 面向多块网格的并行迁移学习机制

针对单个块训练效率受限制问题，本文基于不同子块间几何特征存在相似性的先验认知，引入迁移学习构建跨子块的参数共享机制。通过迁移学习加速模型收敛。

迁移学习的核心思想是将从一个任务中学习到的知识特征应用到另一个任务上。由于分块后的网格块在形状上具有较高的相似性，在训练新块时采用迁移学习策略，以已有的特征参数进行初始化，从而避免随机初始化提高收敛速度。为减少不同块尺寸和比例对训练的影响，首先对所有外形进行归一化处理，并采用 Hausdorff 距离作为相似性度量，寻找最匹配的块以确定权重迁移的来源。具体而言在每

个外形上均匀采样相同数量的特征点，计算其相似性并选择已训练块中最相似的权重进行初始化；若该块尚未训练，则选择次相似的块的权重进行初始化。此外由于采用的网络结构较为轻量，为保持模型的适应性，因此未采用层冻结策略。最后多块结构网格生成 MultiMeshNet 的训练流程如下：

算法 1. 多块结构网格生成的训练过程。

输入：计算空间的内部点$\{(\xi_i,\eta_i)|i=1,2,\cdots,N_r\}$，边界点$\{(\xi_b,\eta_b)|b=1,2,\cdots,N_{bou}\}$，块上的边界点$\{(x_b,y_b)|b=1,2,\cdots,N_{bou}\}$；

输出：MultiMeshNet 的网络参数。

① 根据区域分块步骤分解外形为不同的块，任选一块进行训练；

② 将$\{(x_b,y_b)\}$坐标值缩放到 0~1 的区间；

③ 构建图 2 所示的多块结构网格生成框架，初始化 MultiMeshNet 的网络参数为与其外形最相似的训练完成的模型权重；

④ 构建式（4）损失函数；

⑤ 初始化 *bestLoss* 为较大的值；

⑥ /*训练过程*/

⑦ while *epoch*<*maxEpoch*

⑧ 输入$\{(\xi_i,\eta_i)\}$和$\{(\xi_b,\eta_b)\}$到网络中；

⑨ 根据式（4）计算损失 *currentLoss*；

⑩ 反向传播更新网络参数；

⑪ if *currentLoss* < *bestLoss* then

⑫ 更新 *bestLoss* 为 *currentLoss*；

⑬ 保存当前网络参数 *bestModel*；

⑭ end if

⑮ *epoch*=*epoch*+1；

⑯ end while

⑰ /*训练新的块上的模型参数*/

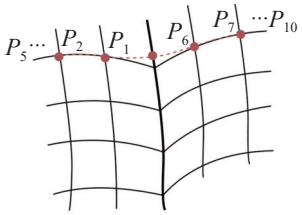
⑱ 初始化新的块上的模型参数为 *bestModel*，设定损失阈值为 10^{-5}，若新的块上的损失值小于 10^{-5} 则停止训练；

⑲ 并行重复行②~⑯训练新的块。

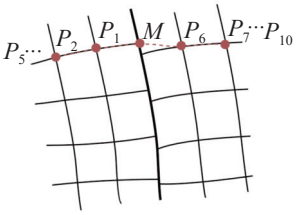
2.2.5 块网格的对齐

在多块网格生成后，需要将其对齐拼接。若所有相邻块之间的网格连接是光滑的，计算过程将更加顺畅，因为没有明显的跳跃或不连续。否则，若网格块之间存在不连续的连接，如图 5（a）（b）所示，则可能会导致误差传递到模拟结果中，从而影响计算精度。因此需要处理这些不光滑、不连续的连接方式。

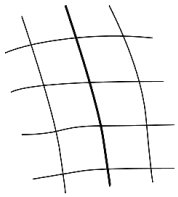
在多块结构网格中，网格块之间的连接可以分为 3 种类型：1）点对点连接但不光滑，在这种情况下，



(a) 点对点连接 (连续) 但不光滑



(b) 非点对点连接



(c) 点对点连接 (连续) 且光滑

Fig. 5 Connection types between adjacent blocks

图 5 相邻块之间的连接类型

相邻块之间有共同的节点，网格线是连续的，但是在交界面处存在不平滑的过渡，如图 5（a）所示为点对点连接但不光滑。这种连接会影响后续数值计算中的稳定性，特别是在流体模拟中可能引入误差。2）非点对点连接，此时相邻块之间没有共同的节点，穿过交界面的网格线不连续，导致块之间的连接不连贯，如图 5（b）所示。3）点对点连接（连续）且光滑，此时是最理想的连接方式，相邻块之间不仅有公共节点，而且交界面的网格线斜率连续，形成平滑过渡，如图 5（c）所示。

对于非点对点连接或点对点连接但不光滑，遵循以下步骤进行处理。1）对于交界面上的每一个点，寻找计算它与交界面两边最近的 10 个点P_1~P_{10}的位置。2）使用埃尔米特代数插值法来根据相邻点的位置计算出交界面点的光滑位置 M。以上的操作与网格平滑的操作有相似之处，需要注意的是，上述插值的操作仅仅移动了相接的线上的点的位置，并不移动内部的网格点的位置。经过以上调整，原本分块网格不光滑、不连续的连接可以变得连续光滑。

3 实验结果

本节在不同外形、控制方程以及网格规模上进

行实验来验证 MultiMeshNet 网格生成能力、对不同方程的支持能力以及网格生成的高效性。

为证明模型的学习能力, MultiMeshNet 采用了轻量级的网络架构 {2, 5, 2}, 即输入层和输出层有 2 个神经元, 隐藏层有 5 个神经元, 模型可学习的参数量仅为 102。由于选用了轻量级的网络, 所有训练和推理过程均在 Intel® Xeon® Platinum 8255C CPU @ 2.50 Hz 上完成, 以及均使用 Adam 优化器进行优化。

本文将 MultiMeshNet 生成的网格质量与传统 TFI[19] 方法以及智能网格生成方法 MGNet[13], MeshNet[24], KANMeshNet[25] 的网格生成质量进行对比分析。为保证实验的公平性, KANMeshNet 的模型架构为{2, 3, 2}, 可学习的参数量为 104; MGNet 和 MeshNet 采用相同的模型配置{2, 5, 5, 5, 5, 2}, 对应的可学习参数量为 117。

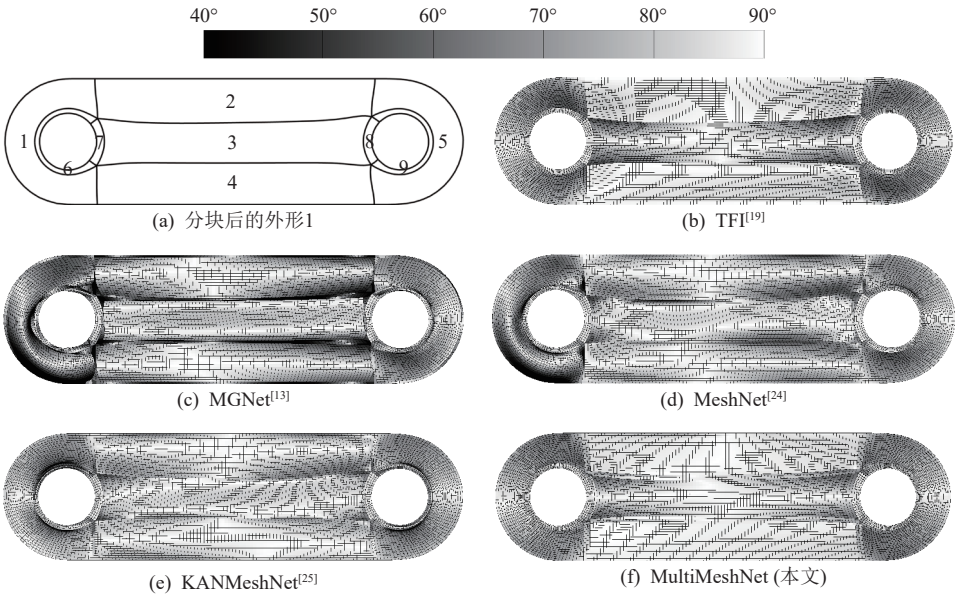
在评估生成网格的质量时, 本文使用多个指标评价生成网格的质量, 包括损失值、平均最小夹角、平均最大夹角、平均等角偏斜度和平均长宽比。具体来说, 损失值通过计算 MultiMeshNet 生成的网格点与预先生成的高质量网格点之间的逐点 L1 损失值并取其平均, 辅助判断生成的网格质量。平均最小夹角和平均最大夹角表示网格中所有单元格的最小角度和最大角度的平均值。平均等角偏斜度表示网格角度与等角网格角度的最大比值, 该值的范围从 0(较好)到 1(较差)。平均长宽比通过平均长度与平均宽度的比值来计算, 该值总是大于或等于 1, 值为 1 时表示网格为正方形。

3.1 不同外形网格生成结果对比

本节展示 MultiMeshNet 在 2 种不同外形上的实验结果, 并与其他 4 种方法生成的网格的质量进行了对比。

图 6 所示为在外形 1 上不同网格生成方法的可视化结果对比。每个图展示了采用最小夹角度量的可视化效果, 白色区域表示网格正交性良好, 黑色区域表示正交性较差。其中 MultiMeshNet 生成的网格正交性良好, 并且边界过渡均匀, 尤其在分块 2, 3, 4 上的网格质量优于其他方法。这主要得益于 MultiMeshNet 强大的非线性学习能力, 在相同的参数下能够更有效地学习网格生成的映射, 并结合了迁移学习, 网络可以更快收敛, 快速适应不同块上的网格生成。MGNet 和 MeshNet 基于多层感知机进行特征提取和网格生成, 其网格质量普遍低于基于 KAN 的方法, 这是因为多层感知机的非线性建模能力较弱, 难以捕捉网格生成中的复杂关系。TFI 方法在网格较为规则的分块 2, 4 上表现较好, 但在分块 1, 5 上的网格质量较差, TFI 方法依赖于插值, 边界外形对网格质量的影响较大。

表 1 展示了在外形 1 上使用 5 种不同方法生成结构网格的数值结果。结果表明, MultiMeshNet 的损失值最低为 0.18, 优于其他智能网格生成方法。根据平均最小夹角和平均最大夹角可评估网格的正交性。MultiMeshNet 的正交性表现优于其他方法, 相较于其他方法提高了 1°~9°。同时, 网格的平均长宽比与其他方法相近, 表明 MultiMeshNet 能够较好地保



(a) 分块后的外形1 (b) TFI[19]

(c) MGNet[13] (d) MeshNet[24]

(e) KANMeshNet[25] (f) MultiMeshNet (本文)

Fig. 6 Comparison of visualization results of different mesh generation methods on shape 1

图 6 在外形 1 上不同网格生成方法的可视化结果对比

持网格单元的几何均匀性，避免了过长或过窄的网格单元。KANMeshNet 生成的网格质量优于基于多层感知机的方法。MeshNet 优于 MGNet 的原因在于 MeshNet 在损失函数上针对网格生成任务进行了优化，相比 MGNet 能够更好地平衡方程项和边界条件项的权重。然而，由于其非线性能力较弱，MeshNet 的表现仍不及基于 KAN 的方法。

Table 1　Comparison of Mesh Generation Results on Shape 1 Using Five Methods

表 1　外形 1 上使用 5 种方法的网格生成结果对比

方法	损失值↓	平均最小夹角↑/（°）	平均最大夹角↓/（°）	平均等角偏斜度↓	平均长宽比↓
TFI[19]		80.24	99.98	0.11	1.46
MGNet[13]	0.55	72.14	108.17	0.20	**1.45**
MeshNet[24]	0.42	76.45	103.75	0.15	1.47
KANMeshNet[25]	0.22	79.40	100.83	0.12	1.49
MultiMeshNet（本文）	**0.18**	**81.23**	**98.98**	**0.10**	1.46

注：黑体数值表示最优值。“↓”表示该指标越小越好，“↑”表示该指标越大越好。

图 7 所示为在外形 2 上不同网格生成方法的可视化结果对比。MultiMeshNet 在上述 5 种方法中表现最佳，尤其在分块 19, 21, 22 上的网格质量明显优于其他方法。相较之下，MGNet 生成的网格未能完全贴合边界，特别是在分块 19, 2 上的网格质量较差。整体而言，基于 KAN 生成的网格质量优于基于多层感知机的方法，因为 KAN 引入了自适应激活函数，相比于多层感知机的固定激活函数，其非线性映射能力更强。TFI 方法在分块 5, 6 的顶点处网格质量表现一般，主要是由于该区域的网格质量受边界影响较大，特别是顶点处的夹角较大，呈钝角。TFI 方法会将这些钝角的边界信息传递到网格内部，从而影响网格质量。

表 2 展示了在外形 2 上使用 5 种方法生成结构网格的结果对比。结果表明，MultiMeshNet 的损失值低于其他智能方法。在网格正交性方面，MultiMeshNet 的平均最小夹角和平均最大夹角优于其他方法，且其网格的平均长宽比为 1.23，接近最优值 1.22，证明 MultiMeshNet 在保证网格几何形状均匀的同时，能够有效保持网格的正交性。相比之下，MGNet 生成的网格质量最差，尤其是在分块 19~23 上，生成的网格未能很好地贴合边界。值得注意的是，并非所有智能方法都优于传统方法。对于 MeshNet 和 MGNet 来说，在参数量较小的情况下，网格生成质量反而低于传统的 TFI 方法。

3.2　不同控制方程网格生成结果对比

MultiMeshNet 网格生成的核心是将网格生成相关的方程嵌入在损失函数中，不同的物理控制方程代表不同的物理现象和数学特性。为了证明框架具备对不同方程通用的支持能力，本节对 MultiMeshNet 使用 3 种不同控制方程进行网格生成的网格质量进行了分析。

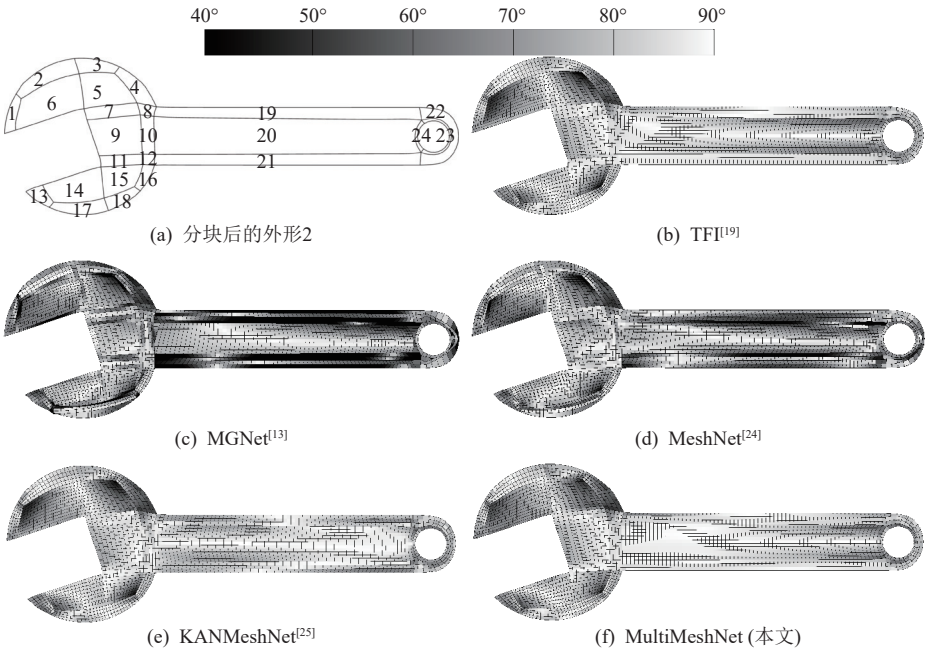


Fig. 7　Comparison of visualization results of different mesh generation methods on shape 2

图 7　在外形 2 上不同网格生成方法的可视化结果对比

Table 2 Comparison of Mesh Generation Results on Shape 2 Using Five Methods

表 2 在外形 2 上使用 5 种方法的网格生成结果对比

方法	损失值↓	平均最小夹角↑/(°)	平均最大夹角↓/(°)	平均等角偏斜度↓	平均长宽比↓
TFI[19]	-	78.93	101.31	0.13	**1.22**
MGNet[13]	0.76	65.98	114.15	0.27	1.34
MeshNet[24]	0.40	76.20	104.01	0.16	1.26
KANMeshNet[25]	0.33	79.22	101.01	0.13	1.23
MultiMeshNet（本文）	**0.26**	**80.06**	**100.15**	**0.12**	1.23

注：黑体数值表示最优值。“↓”表示该指标越小越好，“↑”表示该指标越大越好。

比较的双曲方程形式为

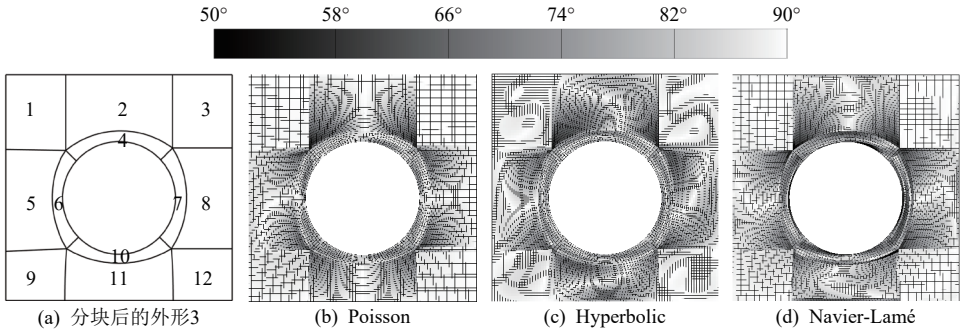
$$\begin{cases} x_\xi x_\eta + y_\xi y_\eta = 0, \\ x_\xi y_\eta - y_\xi x_\eta = S, \end{cases} \tag{6}$$

其中 S 代表网格单元的面积。Navier-Lamé方程用于网格生成的形式为

$$\begin{cases} \lambda\left(x_{\xi\xi} + y_{\eta\xi}\right) + \mu\left(2x_{\xi\xi} + x_{\eta\eta} + y_{\eta\xi}\right) = 0, \\ \mu\left(x_{\xi\eta} + y_{\xi\xi} + 2y_{\eta\eta}\right) + \lambda\left(x_{\xi\eta} + y_{\eta\eta}\right) = 0, \end{cases} \tag{7}$$

其中 μ 和 λ 是拉梅常数。

图 8 展示了不同控制方程在外形 3 上的结果对比。通过结合 3 种不同的控制方程，MultiMeshNet 能够灵活地生成具有接近质量的网格（Poisson, Hyperbolic, Navier-Lamé）。Poisson 方程是线性椭圆偏微分方程，其数值求解相对稳定，且对边界条件的敏感性较低，常被用于生成均匀的网格。Hyperbolic 方程广泛应用于描述波动问题和动态过程，在分块 1, 3, 9, 12 上生成的网格质量较好。Navier-Lamé方程主要用于描述弹性体中的应力和应变分布，可以在网格生成过程中实现对结构力学特性的有效建模，从而帮助生成能够反映实际物理条件的网格。MultiMeshNet 能够灵活地结合这些不同类型的控制方程，具有很强的适应性和通用性。



(a) 分块后的外形3 (b) Poisson (c) Hyperbolic (d) Navier-Lamé

Fig. 8 Comparison of mesh generation results using three equations on shape 3

图 8 在外形 3 上使用 3 种方程的网格生成结果对比

表 3 展示了不同控制方程在外形 3 上生成网格结果的对比。结果表明，MultiMeshNet 在使用 3 种控制方程生成的网格质量中，平均最大夹角和平均最小夹角的最优值与最差值之间的差异不超过 2°，平均等角偏斜度和平均长宽比的数值也接近。表明 MultiMeshNet 能够在不同控制方程下生成质量接近的优质网格，对于不同方程都有较好的适用性。

Table 3 Comparison of Mesh Generation Results on Shape 3 Using Three Different Equations

表 3 在外形 3 上使用 3 种不同方程的网格生成结果对比

不同控制方程	损失值↓	平均最小夹角↑/(°)	平均最大夹角↓/(°)	平均等角偏斜度↓	平均长宽比↓
Hyperbolic	0.026	83.02	97.02	0.08	1.17
Navier-Lamé	0.032	82.17	97.86	0.09	1.19
Poisson	**0.019**	**84.61**	**95.35**	**0.06**	**1.17**

注：黑体数值表示最优值。“↓”表示该指标越小越好，“↑”表示该指标越大越好。

3.3 大规模网格生成开销对比

本节探讨使用 MultiMeshNet 进行大规模网格生成的计算开销。单个计算节点配置为 Intel Xeon Platinum 8255C 处理器，主频为 2.50 GHz。实验中计算节点数量依次设为 1, 4, 8, 16, 网格规模从 10^3~10^8 跨越 6 个数量级，此外我们还在附录 B 中增加了在 GPU 上的网格生成时间。

表 4 展示了 5 种方法在不同网格规模和不同数量计算节点下的网格生成时间。与传统的 TFI 方法相比，MultiMeshNet 在推理效率上提升了 2~10 倍。由于 TFI 方法扩展性较差，随着计算节点数量的增加，其网格生成时间变化不明显。相较于结构相似的 KANMeshNet, MultiMeshNet 将网格生成时间减少到 1/2~1/4。MeshNet 和 MGNet 在推理时间上更短，但其生成的网格质量较低。相比之下 MultiMeshNet 在保持较低生成开销的同时，也保证了较高的网格质量，在速度与质量之间实现了良好的平衡。此外

Table 4 Comparison of Mesh Generation Time for Five Methods on Various Numbers of CPU Computing Nodes

表 4 在不同数量的 CPU 计算节点下 5 种方法的网格生成时间对比

方法	网格规模	网格生成时间/s			
		1 个节点	4 个节点	8 个节点	16 个节点
TFI[19]	9.84×10^3	0.03	0.03	0.03	0.03
	5.18×10^5	2.18	1.96	1.92	1.83
	2.40×10^7	48.23	45.27	43.70	43.20
	1.10×10^8	4 839	3 846	3 624	3 547
KANMeshNet[25]	9.84×10^3	1.35	2.73	1.52	1.06
	5.18×10^5	10.91	3.22	2.75	2.45
	2.40×10^7	164.88	74.67	42.57	24.66
	1.10×10^8	4 237	2 318	1 625	1 232
MeshNet[24]	9.84×10^3	0.19	0.10	0.07	0.04
	5.18×10^5	1.93	0.52	0.29	0.36
	2.40×10^7	72.99	18.89	10.95	6.09
	1.10×10^8	3 033	799	691	523
MGNet[13]	9.84×10^3	0.20	0.06	0.03	0.02
	5.18×10^5	2.13	0.59	0.30	0.35
	2.40×10^7	75.46	20.14	10.89	6.13
	1.10×10^8	3 037	768	686	503
MultiMeshNet（本文）	9.84×10^3	1.36	0.95	0.81	0.67
	5.18×10^5	3.41	2.10	1.35	0.89
	2.40×10^7	92.34	25.61	13.08	8.23
	1.10×10^8	2 848	762	432	327

注：网格生成时间不包括文件读写时间和通信时间。

MultiMeshNet 展现出较强的可扩展性。在网格规模为 2.40×10^7 时，当计算节点数量从 1 增至 4，推理时间由 92 s 降至 25 s，显示出良好的强可扩展性。

表 5 展示了不同方法在弱可扩展性测试下的网格生成时间比较。从弱可扩展性的角度比较，当节点数从 1 增加到 4，网格规模从 6.0×10^5 扩展到 2.4×10^6，扩大了 4 倍，而每个节点的平均任务量保持一致。MultiMeshNet 的网格生成时间仅从 4.2 s 升至 5.3 s，表明其在处理更大规模问题时，能够较好地维持计算效率，具备良好的弱可扩展性。KANMeshNet 则从 12.9 s 升至 19.6 s，扩展性不如其他方法。基于多层感知机的时间变化也比较稳定，优于基于 KAN 架构的方法。MultiMeshNet 略差于 MGNet 和 MeshNet，但是 MultiMeshNet 可以兼顾网格生成质量，达到质量和速度的平衡。

尽管对表 4 已讨论了无 I/O 情况下的网格生成时间，但在实际应用中，网格生成过程通常伴随显著的 I/O 和通信操作，对整体计算效率产生重要影响。因此有必要进一步分析 MultiMeshNet 在不同计算节点数量下的网格生成总时间，此处的总时间涵盖了智能并行网格生成流程中的 3 个步骤。表 6 展示了不同规模网格在多种计算节点配置下的生成总时间。

对于网格规模较小的任务，如 9.84×10^3 和 5.18×10^5，增加计算节点反而导致运行时间增加，主要是由于过度并行化带来了额外的开销。随着节点数的增加，任务划分变得过于细碎，导致每个节点的计算任务非常小，从而显著增加了计算与调度之间的开销。此外节点间的通信和同步成本也随之上升，因此对于较小规模的网格，过多的并行节点反而会增加运行时间。

网格规模为 2.40×10^7 时节点 1~4 的加速效果较好。但随着节点数的进一步增加，性能提升的幅度逐渐减缓。当计算节点从 8 个增至 16 个时，计算时间几乎没有变化，这主要是由于节点间的通信和同步开销逐步增加。

网格规模为 1.10×10^8 时加速效果显著。计算节点从 1 个增至 16 个时，系统的并行计算效率均在 80% 以上，可以获得较好的加速效果。计算任务得以高效分配到各个节点，保证了每个节点的计算性能，同时节点间的通信开销相对于整体计算任务较低，优化了并行计算的整体性能，整体上 MultiMeshNet 在中大规模场景下具备稳定的并行性能。

3.4 基于迁移学习的损失函数分析

为解决单个块训练效率受限的问题，本文基于

Table 5 Comparison of Mesh Generation Time for Four Methods Under Weak Scalable Testing

表 5 4 种方法在弱可扩展性测试中的网格生成时间对比

网格规模	节点数	每节点平均任务量	网格生成时间/s			
			KANMeshNet[25]	MGNet[13]	MeshNet[24]	MultiMeshNet（本文）
6.0×10^5	1	6×10^5	12.9	3.1	3.3	4.2
2.4×10^6	4	6×10^5	19.6	3.8	3.9	5.3
2.4×10^7	8	3×10^6	42.6	10.9	11.0	13.1
4.8×10^7	16	3×10^6	58.9	13.4	14.4	16.5

注：网格生成时间不包括文件读写时间和通信时间。

Table 6 Total Time Required to Generate Meshes at Different Scales Using MultiMeshNet on Different Numbers of CPU Computing Nodes

表 6 使用 MultiMeshNet 在不同数量的 CPU 计算节点上生成不同规模网格的所需总时间

网格规模	网格生成总时间/s			
	1 个节点	4 个节点	8 个节点	16 个节点
9.84×10^{3}	10.04	9.53	16.86	30.20
5.18×10^{5}	54.52	17.45	18.27	31.14
2.40×10^{7}	504.54	134.74	75.04	74.86
1.10×10^{8}	5 189.58	1 384.34	712.26	425.76

不同子块几何特征的相似性引入迁移学习，以实现跨子块的参数共享。通过迁移学习机制，模型能够将一个块上学习到的特征和参数迁移到其他块的模型上，从而加速后续训练的收敛。与随机初始化不同，迁移学习让网络在开始训练时已经具备了一定的网格生成能力，使模型更容易且更迅速地优化。

如图 9 所示为不同方法在训练过程中单块上的损失值。实验结果明表明，迁移学习显著加速了 MultiMeshNet 网络的收敛过程，仅需运行 3 500 轮损失值就达到 10^{-5} 量级，此时停止训练。而未使用迁移学习的 MultiMeshNet(没有迁移学习)的损失函数值则显著高于使用了迁移学习的 MultiMeshNet(有迁移学习)，无论是在初始化损失还是收敛速度上，引入了迁移学习的效果更佳。MultiMeshNet(有迁移学习)在训练 3 500 轮后，相较于其他方法的损失值低 1 个数量级，证明了迁移学习在加速模型训练和提升收敛速度方面的有效性。损失曲线显示 KANMeshNet 的收敛也较快，但其训练时长大约是 MultiMeshNet 的 2 倍。这是因为 KANMeshNet 使用样条函数作为激活函数，其计算复杂度较高，相比之下 MultiMeshNet 的训练开销更小。

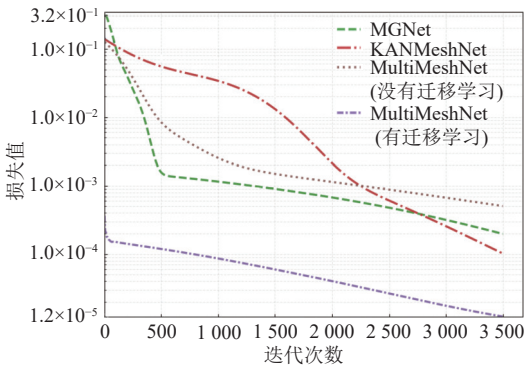


Fig. 9 Loss values during the training process of the five methods

图 9 5 种方法在训练过程中的损失值

3.5 不同参数量下的模型性能对比

我们详细对比了 4 种模型在不同参数量下的性能。如图 10 所示为 4 种方法在不同参数量下的最低损失值的对比，测试了 MGNet, MeshNet, KANMeshNet, MultiMeshNet 在参数量为 0~5 000 的表现。在每次训练过程中，每个模型均迭代 3 000 轮，并记录损失函数的最小值。

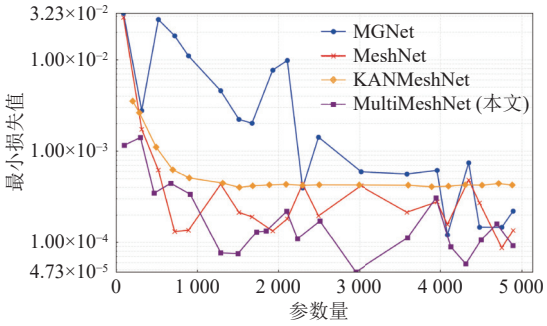


Fig. 10 Comparison of minimum loss values for four methods at different parameter counts

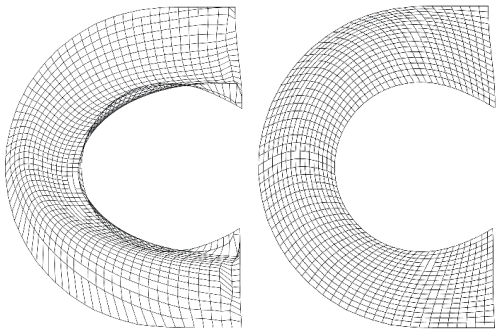
图 10 4 种方法在不同参数量下的最小损失值对比

MultiMeshNet 和 MeshNet 的最小损失值较低，且 MultiMeshNet 在大多数情况下能够达到 4 种方法中的最低损失。在参数量为 0~5 000 时，MultiMeshNet 模型在大部分参数量下都优于其他 3 种方法。MGNet 的损失函数难以有效平衡边界条件损失与控制方程损失，导致模型收敛效果较差。对于 KANMeshNet，当参数量超过 1 000 时，最低损失值几乎不再下降，原因在于该方法需要更长的迭代轮数才能继续收敛。

3.6 不同基函数的敏感度实验

本节对比了基于傅里叶基函数与样条函数的 2 种方法在网格生成任务中的表现，验证了傅里叶基函数的优势。实验在相同的外形条件下分别使用 KANMeshNet 与 MultiMeshNet 模型，通过 Ray Tune 对各自网络结构进行超参数寻优。最终确定的最优结构分别为 KANMeshNet 的 {2, 10, 2}与 MultiMeshNet 的{2, 19, 2}。使用各自的最优架构对网格进行训练与生成。

图 11 展示了 KANMeshNet 与 MultiMeshNet 在相同外形下生成的网格对比结果。KANMeshNet 在网格内部区域能够生成具有良好正交性的网格，但在边界区域出现一定程度的扭曲，表明其在局部细节特征捕捉方面存在一定不足。MultiMeshNet 生成的网格未出现明显的边界扭曲，整体结构更加均匀，网格质量更高，体现出其在保持几何连续性与局部适应性方面更优。



(a) KANMeshNet (b) MultiMeshNet

Fig. 11 Comparison of mesh generation results of KANMeshNet and MultiMeshNet

图 11 KANMeshNet 和 MultiMeshNet 网格生成结果对比

4 结论与展望

本文提出了一种基于无监督学习的智能并行网格生成框架(MultiMeshNet)，支持大规模多块结构网格的高效生成。框架集成了基于标架场的分块方法、傅里叶 KAN 和网格对齐方法，自适应地从物理信息中学习网格划分规则，有效解决了已有智能方法无法处理复杂外形的局限性。引入迁移学习机制，通过迁移不同进程间网格生成任务的相似性特征，显著提升了模型收敛效率。实验结果表明，MultiMeshNet 在亿维网格生成任务中，相比传统并行网格生成实现了 2~10 倍的效率提升。同时在网格质量正交性方面优于已有智能网格生成方法 2°~9°，达到了最优水平。此外 MultiMeshNet 设计充分考虑了轻量化需求，不仅适用于高性能计算平台，也能在普通的个人电脑或服务器上部署和运行。此外框架同样支持在 GPU 上运行，可以进一步提升模型的训练速度和推理效率。未来的工作将进一步缩短训练开销，加速模型收敛。

作者贡献声明：陈新海、彭嘉明提出并设计了基于迁移学习的智能网格生成框架，优化了网络结构并撰写论文；乔鹏、贾孟涵负责构建并预处理数据集，生成与评估不同网格外形，参与论文修改；王庆林、张翔负责整理实验数据并分析结果；杨博、刘杰负责指导并提供计算资源。

参 考 文 献

[1] Gao Xiang, Zhang Xiang, Xu Chuanfu, et al. Research on general mesh generation software for scientific engineering computing[J]. Computer Engineering & Science: Special Issue on High Performance Computing, 2020, 42(10): 1897−1904 (in Chinese)
(高翔, 张翔, 徐传福, 等. 面向科学工程计算的通用网格生成软件系统研究[J]. 计算机工程与科学：高性能专刊, 2020, 42(10)：1897−1904)

[2] Chen Xinhai, Wang Zhichao, Deng Liang et al. Towards a new paradigm in intelligence-driven computational fluid dynamics simulations[J]. Engineering Applications of Computational Fluid Mechanics, 2024, 18(1): 2407005

[3] Secco N R, Kenway G K, He Ping, et al. Efficient mesh generation and deformation for aerodynamic shape optimization[J]. AIAA Journal, 2021, 59(4): 1151−1168

[4] Wei Zishu, Han Yue, Liu Sihao, et al. Lookahead analysis and discussion of research hotspots in artificial intelligence from 2021 to 2023[J]. Journal of Computer Research and Development, 2024, 61(5): 1261−1275 (in Chinese)
(魏子舒, 韩越, 刘思浩, 等. 2021 至 2023 年人工智能领域研究热点分析述评与展望[J]. 计算机研究与发展, 2024, 61(5)：1261−1275)

[5] Yan Junjun, Chen Xinhai, Wang Zhichao, et al. ST-PINN: A self-training physics-informed neural network for partial differential equations[C]//Proc of the 33rd Int Joint Conf on Neural Networks. Piscataway, NJ: IEEE, 2023: 1−8

[6] Chen Xinhai, Gong Chunye, Wan Qian, et al. Transfer learning for deep neural network-based partial differential equations solving[J]. Advances in Aerodynamics, 2021, 3: 1−14

[7] Chen Xinhai, Liu Jie, Gong Chunye, et al. MVE-Net: An automatic 3-D structured mesh validity evaluation framework using deep neural networks[J]. Computer-Aided Design, 2021, 141: 103104

[8] Wang Zhichao, Chen Xinhai, Li Tieju, et al. Evaluating mesh quality with graph neural networks[J]. Engineering with Computers, 2022, 38(5): 4663−4673

[9] Xiao Qisong, Chen Xinhai, Liu Jie, et al. MH-DCNet: An improved flow field prediction framework coupling neural network with physics solver[J]. Computers & Fluids, 2024, 284: 106440

[10] Xiao Jing, Chen Xinhai, Wang Qingling, et al. MeshONet: A generalizable and efficient operator learning method for structured mesh generation[J]. arXiv preprint, arXiv: 2501.11937, 2025

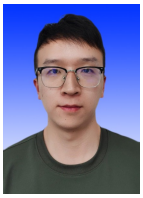
[11] Liu Zengsheng, Chen Shizhao, Gao Xiang, et al. ISpliter: An intelligent and automatic surface mesh generator using neural networks and splitting lines[J]. Advances in Aerodynamics, 2023, 5(1): 1−25

[12] Papagiannopoulos A, Clausen P, Avellan F. How to teach neural networks to mesh: Application on 2-D simplicial contours[J]. Neural Networks, 2021, 136: 152−179

[13] Chen Xinhai, Li Tiejun, Wan Qian, et al. MGNet: A novel differential mesh generation method based on unsupervised neural networks[J]. Engineering with Computers, 2022, 38(5): 4409−4421

[14] Peng Jiaming, Chen Xinhai, Liu Jie. 3DMeshNet: A three-dimensional differential neural network for structured mesh generation[J]. Graphical Models, 2025, 139: 101257

[15] Chen Linqi, Zhou Ying. Boundary layer mesh optimization algorithm based on local feature size[J]. Modeling and Simulation, 2023, 12(4): 3464−3471 (in Chinese)
(陈林祺, 周颖. 基于局部特征尺寸场的边界层网格生成算法[J]. 建模与仿真, 2023, 12(4): 3464−3471)
[16] Ripley R, Lien F S, Yovanovich M. Numerical simulation of shock diffraction on unstructured meshes[J]. Computers & Fluids, 2006, 35(10): 1420−1431
[17] Smith R E, Eriksson L E. Algebraic grid generation[J]. Computer Methods in Applied Mechanics and Engineering, 1987, 64(1/2/3): 285−300
[18] Thompson J F, Thames F C, Mastin C W. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies[J]. Journal of Computational Physics, 1974, 15(3): 299−319
[19] Gordon W J, Hall C A. Construction of curvilinear co-ordinate systems and applications to mesh generation[J]. International Journal for Numerical Methods in Engineering, 1973, 7(4): 461−477
[20] Allen C. Towards automatic structured multiblock mesh generation using improved transfinite interpolation[J]. International Journal for Numerical Methods in Engineering, 2008, 74(5): 697−733
[21] Babuska I, Flaherty J E, Henshaw W D, et al. Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations [M]. Berlin: Springer, 2012
[22] Xiao Zhoufang, He Shouping, Xu Gang, et al. A boundary element-based automatic domain partitioning approach for semi-structured quad mesh generation[J]. Engineering Analysis with Boundary Elements, 2020, 113: 133−144
[23] Kim M, Lee J, Kim J. GMR-Net: GCN-based mesh refinement framework for elliptic PDE problems[J]. Engineering with Computers, 2023, 39(5): 3721−3737
[24] Chen Xinhai, Liu Jie, Zhang Qingyang, et al. Developing a novel structured mesh generation method based on deep neural networks[J]. Physics of Fluids, 2023, 35(9): 097137
[25] Peng Jiaming, Chen Xinhai, Zhang Qingyang, et al. A data-free Kolmogorov–Arnold network-based method for structured mesh generation[J]. Physics of Fluids, 2024, 36(11): 117146
[26] Liu Ziming, Wang Yixuan, Vaidya S, et al. KAN: Kolmogorov-Arnold networks[J]. arXiv preprint, arXiv: 2404.19756, 2024



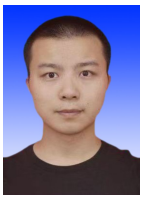
Chen Xinhai, born in 1995. PhD, assistant professor. His main research interest includes intelligent mesh generation.
陈新海, 1995 年生。博士, 助理研究员。主要研究方向为智能网格生成。



Peng Jiaming, born in 2000. PhD candidate. Her main research interest includes intelligent mesh generation.
彭嘉明, 2000 年生。博士研究生。主要研究方向为智能网格生成。



Qiao Peng, born in 1988. PhD, assistant professor. His main research interests include high-performance computing and intelligent computing.
乔　鹏, 1988 年生。博士, 助理研究员。主要研究方向为高性能计算、智能计算。



Jia Menghan, born in 1994. PhD, assistant professor. His main research interests include high-performance computing and intelligent computing.
贾孟涵, 1994 年生。博士, 助理研究员。主要研究方向为高性能计算、智能计算。



Wang Qinglin, born in 1987. PhD, associate professor. His main research interests include high-performance computing and parallel computing.
王庆林, 1987 年生。博士, 副研究员。主要研究方向为高性能计算、并行计算。



Zhang Xiang, born in 1986. PhD, associate professor. His main research interests include high-performance computing and intelligent computing.
张　翔, 1986 年生, 博士, 副研究员。主要研究方向为高性能计算、智能计算。



Yang Bo, born in 1987. PhD, associate professor. His main research interest includes intelligent computing.
杨　博, 1987 年生。博士, 副研究员。主要研究方向为智能计算。



Liu Jie, born in 1974. PhD, professor. His main research interest includes intelligent computing.
刘　杰, 1974 年生。博士, 研究员。主要研究方向为智能计算。

附录 A.

KAN 的理论基础来源于 Kolmogorov-Arnold 表示定理，该定理指出任何多变量连续函数 $f(x)$ 都可以表示为有限个单变量函数的组合与加法。可以形式化为

$$f(\boldsymbol{P})=f(P^1,P^2,\cdots,P^d,\cdots,P^D)=\sum_{q=1}^{2D+1}\varphi^q\left(\sum_{d=1}^{D}\phi^{q,d}\left(P^d\right)\right), \tag{A1}$$

其中输入点为 $\boldsymbol{P}=(\xi,\eta)$ 为计算空间的 2 维坐标点，$\phi^{q,d}$ 是可学习的激活函数，φ^q 是单变量函数。为将 Kolmogorov-Arnold 表示定理扩展到任意深度和宽度，定义了一个具有 n_{in} 输入维度和 n_{out} 输出维度的 KAN 层

$$\begin{aligned}&\boldsymbol{\Phi}=\left\{\phi^{q,d}\right\},\\&d=1,2,\cdots,n_{\text{in}},\\&q=1,2,\cdots,n_{\text{out}}。\end{aligned} \tag{A2}$$

式（A2）为构建更深的 KAN 架构提供了基础，使得其可以扩展到任意深度。一个 L 层的 KAN 可以表示为多个 KAN 层的组合，表示为

$$KAN(\boldsymbol{P})=(\boldsymbol{\Phi}_{L-1}\cdot\boldsymbol{\Phi}_{L-2}\cdot\cdots\cdot\boldsymbol{\Phi}_1\cdot\boldsymbol{\Phi}_0)\boldsymbol{P}, \tag{A3}$$

其中 $\boldsymbol{\Phi}_i$ 表示第 i 层的 KAN，$\boldsymbol{\Phi}_i$ 由 $\phi^{q,d}$ 组成。在 KAN[26] 中，ϕ 是通过样条函数来参数化的。但是本文用 1 维傅里叶函数替代样条函数，引入傅里叶函数后，激活函数 ϕ 被定义为

$$\phi(\boldsymbol{P})=\sum_{d=1}^{D}\sum_{k=1}^{K}\left(\alpha_k^d\cos\left(kP^d\right)+\beta_k^d\sin\left(kP^d\right)\right)+c, \tag{A4}$$

其中 D 表示特征维度的数量，P^d 表示 $\boldsymbol{P}$ 的第 d 个特征。K 是基频分量的个数，决定了每个输入维度中包含多少个正弦和余弦项的傅里叶系数。α_k^d 和 β_k^d 分别表示可学习的傅里叶系数，c 是可学习的偏置项。

附录 B.

如表 B1 所示是 5 种方法在生成不同规模网格时，在不同数量 NVIDIA GeForce RTX 3090 计算节点下的网格生成时间。在网格规模为 9.84×10^3，且网络结构本身相对轻量时，GPU 由于需要承担启动与数据传输的固定开销，反而可能导致其运行时间劣于 CPU。随着网格规模增大，当网格规模为大于等于 5.18×10^5 时，智能网格生成方法能够充分利用 GPU，此时其生成时间小于在 CPU 平台上的生成时间。进一步地，当网格规模达到 1.10×10^8 时，GPU 的高并行度与显存带宽优势更加凸显，使得网格生成时间远小于 CPU 平台，整体加速效果明显。

Table B1 Mesh Generation Time for Five Methods on Various Numbers of GPU Computing Nodes

表 B1 在不同数量的 GPU 计算节点下的 5 种方法的网格生成时间

方法	网格规模	网格生成时间/s			
		1 个节点	2 个节点	4 个节点	8 个节点
TFI[19]	9.84×10^3	0.19	0.16	0.16	0.15
	5.18×10^5	2.95	2.82	2.15	2.1
	2.40×10^7	112.3	109.5	105.32	102.3
	1.10×10^8	465.71	461.21	460.87	451.6
KANMeshNet[25]	9.84×10^3	0.67	0.43	0.22	0.15
	5.18×10^5	7.65	4.21	2.92	2.16
	2.40×10^7	142.21	88.55	56.9	32.71
	1.10×10^8	1 304.5	745.17	474.88	315.54
MeshNet[24]	9.84×10^3	0.81	0.42	0.27	0.19
	5.18×10^5	1.33	0.92	0.5	0.38
	2.40×10^7	55.16	30.83	21.65	11.26
	1.10×10^8	1 070.7	642.81	332.59	184.34
MGNet[13]	9.84×10^3	0.88	0.65	0.38	0.20
	5.18×10^5	1.32	0.96	0.58	0.42
	2.40×10^7	55.78	30.47	22.02	12.23
	1.10×10^8	1 074.5	634.01	352.63	186.56
MultiMeshNet（本文）	9.84×10^3	0.92	0.62	0.39	0.21
	5.18×10^5	2.53	1.63	0.98	1.62
	2.40×10^7	64.44	35.15	23.3	15.87
	1.10×10^8	1 084.2	696.86	350.88	189.14

注：网格生成时间不包括文件读写时间和通信时间。

如表 B2 所示，MultiMeshNet 在生成不同规模网格时，网格生成的总时间随着 NVIDIA GeForce RTX 3090 计算节点数量的变化而不同。在较小的网格规模，如 9.84×10^3 和 5.18×10^5 时，各节点之间的加速效果并不明显。然而随着网格规模的增大，尤其是在 2.40×10^7 和 1.10×10^8 时，GPU 的高并行度和显存带宽的优势显著体现，生成时间明显减少。表明在处理大规模网格时，使用更多计算节点能够减少总时间开销。

Table B2 Total Time Required to Generate Meshes at Different Scales Using MultiMeshNet on Different Numbers of GPU Computing Nodes

表 B2 使用 MultiMeshNet 在不同数量的 GPU 计算节点上生成不同规模网格所需总时间

网格规模	网格生成总时间/s			
	1 个节点	2 个节点	4 个节点	8 个节点
9.84×10^3	6.78	8.69	7.69	10.2
5.18×10^5	10.1	18.6	15.89	16.62
2.40×10^7	356.03	280.36	211.85	216.78
1.10×10^8	1 525.44	1 125.10	853.13	624.12