arXiv:2303.15681v1 [cs.LG] 28 Mar 2023

基于图神经网络的定常偏微分方程物理求解器

Rini Jasmine Gladstone[1], **Helia Rahmani**[2], **Vishvas Suryakumar**[2], **Hadi Meidani**[1], **Marta D'Elia**[3], *和 **AhmadZareei**[2], *

[1]土木与环境工程系，伊利诺伊大学厄巴纳-香槟分校，美国伊利诺伊州[2]Meta Reality Labs，美国华盛顿州雷德蒙德[3]Pasteur Labs，美国纽约州布鲁克林*
marta.delia@simulation.science，azar@meta.com

摘要

基于物理的深度学习框架已展现出良好的效果，能够准确建模复杂物理系统的动力学，并对不同的问题输入具备泛化能力。然而，定常问题面临一项挑战：为了获得准确预测，需要在整个计算域内进行远距离信息交换。在图神经网络（GNN）领域，这意味着需要构建更深的网络，而这反过来可能会损害或减慢训练过程。在这项工作中，我们提出了两种图神经网络架构来应对这一挑战——边增强图神经网络和多图神经网络。我们表明，在定常固体力学问题中，这两种网络的表现均显著优于基线方法（提升 1.5 至 2 倍）。此外，所提出的架构能够很好地泛化到未见过的区域、边界条件和材料。这里，通过一种新颖的坐标变换来处理可变区域，该变换能够实现旋转和平移不变性。本文拓宽了基于图神经网络的神经算子所能处理的问题范围，为其应用于复杂的科学和工业场景奠定了基础。

1 引言

用于求解偏微分方程（PDE）的数值模型，对于物理学、生物学、材料科学和金融学等众多领域的科学建模至关重要。尽管有限元法、有限差分法、有限体积法等多种技术已被开发为高保真求解器，但如何加速计算并降低其计算成本仍然是一项挑战。此外，当控制偏微分方程未知时，即使有观测数据，经典技术也无法预测物理系统。深度学习的最新进展，通过利用控制方程、观测数据或二者兼用，使得能够更快速、准确地评估物理系统响应的算法成为可能[1,2]。

通过将控制偏微分方程和边界条件加入损失函数来训练神经网络，是利用基于深度学习的代理模型学习物理系统解的一个典型例子[3–7]。在此，神经网络被用作偏微分方程解的近似表示。虽然这类代理模型有助于求得给定一组参数时偏微分方程的特定解，但只要参数、边界条件或域几何稍有变化，就需要重新训练，因此在优化、设计和不确定性量化等场景中吸引力较低。

另一类基于神经网络的求解器是卷积神经网络（CNN），它利用离散化域上的观测数据快照来预测物理解。虽然这类数据驱动方法不需要事先了解控制偏微分方程，但它们通常局限于特定的域离散化，难以推广到其他域几何形状。基于 CNN 的方法包括受偏微分方程启发的架构[8], 自回归稠密编码器—解码器网络[9], 以及符号多层神经网络[10]。

在另一类数据驱动的基于深度学习的代理模型中，神经网络用于学习一种离散化方式，然后将其用于经典求解器[11–14]。具体而言，神经网络学习一个用于粗尺度插值的场。尽管已有研究表明，这些方法能够提高精度并进一步加速传统求解器，但需要注意的是，它们仍然受到初始离散化的限制

为实现分辨率不变性，一类基于神经网络的偏微分方程求解器着重于计算域的连续图表示[15–19]。在这类表示中，网络更新的连续性确保架构不受数据分辨率影响；而基于图的网络所特有的节点间增强交互，则提高了该架构求解复杂物理系统的准确性。相关研究表明，可以优化图节点的位置，以便在不同精度级别上更好地学习解

精度[20]。此外，其中一些方法及其扩展[21,22]，着重学习偏微分方程输入与输出之间的连续映射，因此被称为神经算子。这些神经算子以及 DeepONets 等其他神经算子[23],已成功准确预测多个偏微分方程的解，并且对于偏微分方程输入参数的变化具有较好的泛化能力。值得一提的是，使用基于图的表示的一个优点是，可以通过稀疏表示恢复系统的动力学[18,21,24]。

另一类利用基于图的表示方法的网络是 MeshGraphNet 及其扩展模型。该方法最初在[25] 中针对时变问题提出，随后在[26]，中扩展至多尺度设置。这些基于网格的图神经网络将网格信息以及相应的物理参数（如载荷和边界条件）与模型参数编码到一个连通图中。除基于网格的方法外，值得一提的是，基于粒子的方法及其图表示也已成功应用于物理仿真[27]。在此类方法中，物理系统以降阶模型的方式通过粒子进行描述；随后将粒子视为图的节点，并使用消息传递神经网络学习和计算粒子的动力学。上述两类基于图的方法都能够准确近似物理仿真，并且能够很好地泛化到不同分辨率和边界条件。

在本文中，我们探讨基于网格的 GNN 架构；这类架构利用相邻节点之间的局部（或短程）相互作用来估计响应。当 GNN 的节点与偏微分方程计算域相关联时，节点之间的相互作用可以解释为偏微分方程区域中的空间相互作用。由于大多数物理系统都涉及局部相互作用，基于图神经网络的代理模型是充当有效仿真引擎的最佳候选方案之一。然而，当物理系统需要长程相互作用时，例如在静态固体力学问题中，标准的基于网格的 GNN 方法通常无法准确捕捉物理规律。这是因为一般而言，远距离节点之间的信息交换需要大量消息传递步骤。由于 GNN 随层数增加而表现出较差的尺度扩展特性，这种做法并不理想。在基于图的神经算子领域，这一问题已通过采用多分辨率图得到解决，从而实现更快的信息传播[16]。

在此，我们基于 MeshgraphNets，提出两种 GNN 架构，无需使用深层网络即可克服长距离消息传递的挑战。我们介绍边增强图神经网络（EA-GNN）和多图神经网络（M-GNN）。在 EA-GNN 中，我们引入“虚拟”边，以加快信息传播，从而提高计算效率。在 M-GNN 中，我们则采用受多重网格方法启发的多分辨率方法。与文献[26] 提出的方法不同，我们通过从原始网格中随机移除节点，并向二阶或三阶邻居添加新边，来获得低分辨率图。在此过程中，节点保留其原有的坐标属性，并逐步将与高阶邻居的交互添加到图中。此外，为了使 GNN 能够泛化到不同的几何形状，我们将物理系统映射到一个对平移和旋转不变的模拟空间，从而引入不变模拟坐标空间（关于如何在基于图的神经算子中实现相同性质的技术，参见[28] ）。这一过程有助于加快 GNN 的训练，并使其能够泛化到新的几何形状。

我们的主要贡献包括：

- 一种新颖的坐标变换方法，使所提出的图神经网络对旋转和平移具有不变性，并能推广到新的区域几何形状。
- 一种边增强策略，可加快图中的消息传递，从而提高涉及长程相互作用问题的训练效率。
- 一种多图方法，通过以层次化方式在不同分辨率图之间传递信息，从而加快消息在区域中的传播。

2 背景

2.1 图神经网络

图神经网络（GNN）是一类作用于由节点和边组成的图结构上的深度学习方法[29]。图神经网络通过在图中相邻节点之间进行消息传递，来建模各节点特征之间的相互依赖关系。它们已成功应用于图、边或节点层面的预测和分类任务。近年来，图卷积网络（GCN）等图神经网络变体[30]，图注意力网络（GAT）[31]，图循环网络（GRN）[32] 在各种任务中展现出突破性性能。图神经网络强大的表达能力和卓越性能，使其得以应用于众多数据天然以图结构表示的领域，例如粒子物理[33]，高能物理探测器[34]，电力系统[35]，等。此外，图神经网络框架能够模拟复杂的物理域，其中流体、刚性固体和可变形材料彼此相互作用[27]。在 MeshGraphNets 中，网格物理域由网格表示；网格本质上是一种图，图神经网络可在其上学习预测物理规律（例如，参见[25]）。

2.2 挑战 使用图神经网络建模弹性材料和超弹性材料的变形 s

在这项工作中，我们研究定常弹性和超弹性问题，尤其关注 Mooney-Rivlin 超弹性模型[36]，，尽管我们提出的方法有望适用于一大类椭圆型偏微分方程。我们的目标是训练能够捕捉物理系统本构行为的图神经网络模型，而不是近似特定问题设定下的响应。为此，我们期望所提出的图神经网络代理模型能够在不同的域几何形状、边界条件、载荷和材料属性下表现良好。

为定常力学问题（以及一般的椭圆型偏微分方程）开发代理模型，一个难点在于，任意一点处的解都取决于区域内距离更远的点。事实上，这些问题中的解可以表示为整个区域上的积分。这意味着图神经网络必须在网络中包含远距离点之间的连接。从某种意义上说，所提出的架构旨在实现区域内任意点之间的快速消息传播。

2.3 用于物理仿真的 MeshGraphNet

MeshGraphNets[25] 利用了网格表示，这种表示广泛用于结构力学、空气动力学问题等领域的有限元仿真。随着自适应网格划分的发展，即使使用高度不规则的网格，也能对复杂几何形状的变形进行精确的高分辨率仿真。MeshGraphNets通过编码器将网格中的仿真状态编码为图结构，利用图神经网络的消息传递步骤近似描述物理系统内部动力学的微分算子，再通过解码器提取节点级动力学信息以更新网格，从而预测物理系统的动力学。

MeshGraphNet 在给定当前网格及先前时间步网格的条件下，学习时间 t $+1$ 时网格动态量的正向模型。编码器将当前网格编码为图，其中网格节点成为图节点，网格边成为图中的双向边。它们还添加了额外的边，即世界边，用于学习自碰撞、接触等相对于网格而言是非局部的外部动力学。世界边由空间邻近性创建，限定在固定半径内。图创建完成后，生成节点属性和边属性，并使用可学习的编码器网络对其进行编码。接下来，由 L 个可执行消息传递操作的图网络（GN）块组成的处理器，会更新世界边、网格边以及当前图的节点特征。最后，为了根据时间 t 时的状态预测时间 t $+1$ 时的状态，使用作为多层感知机的解码器网络，将更新后的节点特征转换为输出特征。

MeshGraphNets 能够准确且高效地对物理系统的动力学进行建模。它还具有良好的泛化能力，并且可以在推理时进行扩展。然而，它们尚未在静态固体力学问题等与时间无关的问题上进行测试。

在[26], 一种改进版的MGN网络，即多尺度MGN，被提出，其目的是恢复表征基于网格的离散化的空间收敛性，并提高网格细化时的训练效率。这项工作借鉴了基于网格的离散化中的多重网格方法，但其内在架构和适用范围与本文所介绍的方法不同。

3 方法

如上一节所述，求解与时间无关的固体力学问题需要在所有自由度之间快速传播信息，因为空间中每个点的变形或应力都取决于其他所有点。随着计算域分辨率的提高，保证这种传播在计算上变得极具挑战性。为应对这一挑战，在使用图神经网络作为数值求解器时，一种自然的解决方案是增加网络的深度。然而，这种方法的计算成本非常高，并且可能导致过平滑[37]，，使模型更难训练，或导致梯度消失/梯度爆炸[38]。为实现信息的快速传播并避免这些问题，我们提出了两种方法：(a) 边增强图神经网络，以及 (b) 多图神经网络。下面我们将分别介绍这两种模型。

(a) 边增强：我们将第一种方法称为对现有图进行“边增强”。由于图是由网格生成的，节点仅与其邻居相连；如果两个节点相距 n 跳，则需要 n 个 GN 模块，消息才能在这两个节点之间传递[39]。因此，随着 n 增大，需要更深的网络才能实现广泛的消息传播。为避免增加网络深度，我们在随机选取的节点之间引入增强边，以提高图的非局部性（见图 1c）。这种边增强显著减少了节点从远处节点收集消息所需的跳数，从而减少 GN 模块的数量，加快消息传播。

(b) 多重图：这种方法受到偏微分方程多重网格方法的启发，其基础是离散化（或网格）的层次结构[40]。通过依次执行一系列图下采样操作和上采样操作，我们将多重网格的概念应用于图，并以层次化方式在整个计算域中传递信息。实际应用中，下采样操作对应于网格粗化，生成规模更小的图，这些图连接着区域中相距较远的点

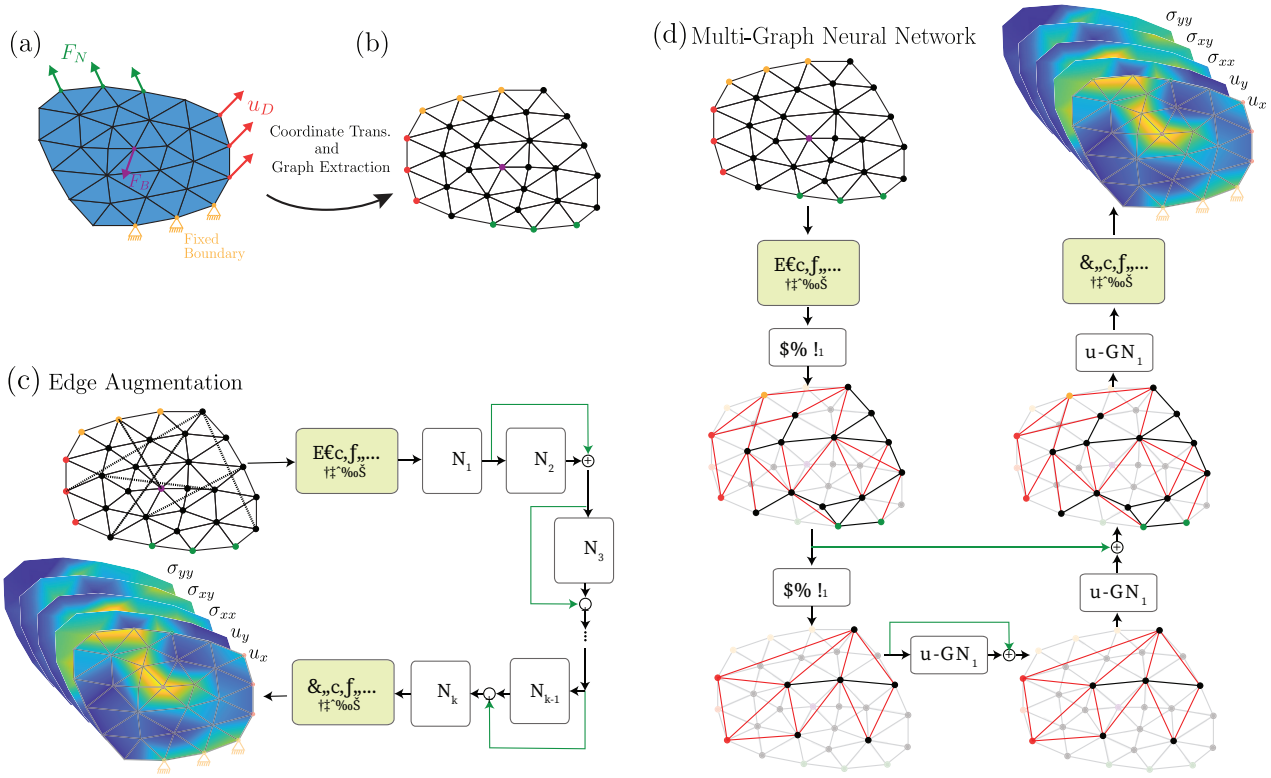

图1。 (a) 二维时不变（静态）固体力学问题示意图。(b) 将物理域转换为归一化坐标。(c) 边增强图神经网络示意图（增强边以虚线标示），其中归一化图被传递至多层感知机编码器，经过多次 GN 模块处理后，再由解码器恢复整个区域内的位移和应力向量。(d) 多图神经网络示意图，其中编码后的特征经过 M-Net模块处理，并对网格进行下采样/上采样。

。另一方面，在上采样过程中，信息会重新分布到整个图中。这种方法类似于图像处理任务中使用的图U-Net架构，[41] 例如图像分类和分割。

对于这两种方法，我们假设计算域为三角形网格，该网格会转换为图，具体过程详见第 4.1 节，并如图1所示。

3.1 边增强图神经网络（EA-GNN）

为了实现节点间信息的更快传播，我们从图中选取一组节点；如果这些节点之间不存在边，就添加一条双向边。需要指出的是，由于这条边不同于其他边，我们会在边属性中添加一个额外特征，以表明该边属于增强集合。增强图中的额外边数量是一个超参数，会在训练过程中手动调节。此外，增强边的端点是按照均匀分布随机选取的。

所提出的 EA-GNN 与 MeshGraphNet 具有相同的高层网络架构[25] ，其构建模块包括：(i) 编码器，(ii) m 个 GN 模块，以及 (iii) 解码器。如第 4.1 节所述，对于本文考虑的与时间无关的问题，为了使图具有平移和旋转不变性，在创建图之前，网格会被转换到主轴坐标系（称为仿真坐标系）中。随后，网格会被转换为图结构：将网格顶点视为节点，将网格顶点之间的连接视为边（见图 1a、b）。节点属性包括仿真坐标系中的节点位置（x、y）、节点类型（内部或边界）、所施加边界条件的类型（齐次和非齐次狄利克雷条件、诺伊曼条件）、边界条件的方向和大小、体力标记及其在仿真坐标系中的大小和方向。边属性包括节点间的欧几里得距离，以及 x 和 y方向上的位置差，即 $\Delta x = x_2 - x_1$ 和 $\Delta y = y_2 - y_1$。

生成图后，节点属性和边属性会通过编码器编码到潜在空间中。

随后，编码节点和边属性会经过 m 个 GN 模块并得到更新。每个 GN 模块由边更新模块和节点更新模块组成。设 u'_i 为节点 i 的编码节点属性向量，u'_j 为节点 j 的编码节点属性向量，其中 $j \in M(i)$，M(i) 表示节点 i 的邻域。设 e'_{ij} 为 i 和 j 之间边的编码边属性向量。边更新模块 χ 是一个多层感知机，它接收连接该边的节点属性以及边属性，并返回更新后的属性，即 $e'_{ij} = \chi(e'_{ij})$。

节点更新模块由三个多层感知机组成，分别为 ϕ、γ 和 β；节点 i 的编码后节点属性 u'_i 按如下方式更新：

$$\begin{aligned} \mathbf{m}_{ij} &= \phi(\mathbf{u}'_i, \mathbf{u}'_j, \mathbf{e}'_{ij}) \\ \mathbf{u}'_i &= \gamma(\mathbf{u}'_i, \frac{1}{N}\sum_{j \in M(i)} \mathbf{m}_{ij}) \\ \mathbf{u}'_i &= \mathbf{u}'_i + \beta(\mathbf{u}'_i). \end{aligned} \tag{1}$$

每个 GN 模块之间都会添加一个跳跃连接，以避免过平滑[38]。以下方程描述了网络中两个连续 GN 模块GN_k 和 GN_{k+1}的更新：

$$\begin{aligned} \mathbf{u}'^{(k)}, \mathbf{e}'^{(k)} &= GN_k(\mathbf{u}'^{(k-1)}, \mathbf{e}'^{(k-1)}) \\ \mathbf{u}'^{(k)} &= \mathbf{u}'^{(k)} + \mathbf{u}'^{(k-1)} \\ \mathbf{u}'^{(k+1)}, \mathbf{e}'^{(k+1)} &= GN_{k+1}(\mathbf{u}'^{(k)}, \mathbf{e}'^{(k)}) \\ \mathbf{u}'^{(k+1)} &= \mathbf{u}'^{(k+1)} + \mathbf{u}'^{(k)}. \end{aligned} \tag{2}$$

最后，更新后的节点属性 u′ 会经过解码器，以输出节点变形或应力值。编码器和解码器函数均为多层感知机网络。

3.2 多图神经网络（M-GNN）

这种方法的灵感来自多重网格求解器，并依赖于层次化学习。该架构包含三个组成部分：(i) 编码器，(ii) M-Net 模块，以及 (iii) 解码器（见图 1d）。编码器和解码器具有与 EA-GNN 相同的网络架构和功能；不过，出于本节后文将阐明的原因，此处我们不考虑边属性。

M-Net 模块以单个图网络模块 GN 开始，用于更新从编码器接收的编码节点属性。GN 模块使用 GraphSAGE 算子[42]进行节点更新。编码后的节点属性 u′通过 GN 模块按如下方式更新。

$$\begin{aligned} \mathbf{u}'_i &= \text{GraphSAGE}(\mathbf{u}'_i, \mathbf{u}'_j), \forall j \in M(i) \\ \mathbf{u}'_i &= \mathbf{u}'_i + \beta(\mathbf{u}'_i), \end{aligned} \tag{3}$$

其中，β是一个多层感知机。更新后的节点属性 $\mathbf{u}'$随后经过一系列交替的下采样操作层和 GN 模块。接着，再经过一系列交替的上采样操作层和 GN 模块。下采样层和上采样层的数量由多图深度超参数d决定。下采样层的输出会与对应上采样层的输出相加，从而得到相同网格细化程度（或图规模）的结果。请注意，所有 GN 模块更新均采用公式 3 所示的相同结构，且最后一个上采样层会使图恢复到原始规模。

下采样层通过自适应选择对应更粗网格的节点子集来对数据进行下采样；下采样的节点数量由超参数 r 决定，该参数表示下采样比例。为自适应选择节点，我们采用“U-net 子采样”算法[41]。具体而言，将节点属性投影到可训练向量 p 上，得到标量投影 $u_i^T p$，并根据投影值选取排名前 k 的节点。由于标量积衡量了节点 i 投影到 p 上时保留的信息量，因此采样排名前 k 的节点可确保较小的图保留最多信息。上采样操作通过记录对应下采样层所选节点的位置，对图进行上采样，并利用这些信息将节点放回图中的原始位置。

为了确保下采样后没有不连通节点，并增强节点之间的连通性，我们计算 l^{th} 图幂，方法与 [41]，类似，并使用所得的图。此操作会在图中距离不超过 l 跳的节点之间建立连接。这可以通过将图的邻接矩阵与自身相乘 l 次来实现。训练时，我们选择 $l = 3$，并在每个

下采样层中使用连通性更好的增强图。这一步对于基于网格的物理仿真尤为重要，可确保信息在节点之间不间断地传播。

GraphSAGE 算子[42], 在 GN 模块中使用，是一种仅考虑节点属性的消息聚合算法。实际上，由于节点经过下采样，已有的边会丢失，同时会引入新的边以提升连通性。这使得更新边属性（包括已有边和新建边的属性）并非易事。由于节点位置被视为节点属性，模型可以根据节点属性间接计算欧几里得距离和位置差等边属性。因此，移除边属性不会导致信息丢失。

4 结果与讨论

4.1 数据生成

数据生成过程包含三个步骤：(i) 网格生成，(ii) 有限元仿真，(iii) 图变换。我们使用贝塞尔曲线生成二维随机几何形状，以确保区域具有多样性和非线性。在边界上随机选取的位置施加不同的边界条件（狄利克雷和诺伊曼）。对于每种几何形状，边界条件的长度、位置、大小和方向均随机选取。我们还在区域内部随机选取的位置施加体力。网格使用 `Gmsh` Python 包[43]。使用 Abaqus[44], 进行有限元仿真，以获得节点变形和应力值。

坐标变换 由于输出值（变形和应力）取决于坐标，且训练集中的每个样本都具有不同的几何形状，因此有必要将节点坐标指定为节点属性。然而，这会使图对平移和旋转具有变体性，即如果网格发生平移或旋转，网络就会将其视为不同的几何形状，并输出不同的变形和应力值。这使训练过程变得困难，因为需要在训练中加入冗余数据，才能使模型对旋转和平移保持不变。为了解决这一问题，我们将群等变性作为归纳偏置，通过将几何形状变换到主轴坐标系中，确保图对平移和旋转保持不变。因此，当几何形状发生旋转或平移时，节点坐标保持不变，进而使变换后的区域对旋转和平移保持不变。从现在起，我们将这些坐标称为“仿真坐标”（SC）。

数据增强 为了提高模型对未见过的几何形状的泛化能力，我们引入了两种数据增强策略。首先，我们同时使用德洛内算法和平行四边形填充算法进行网格生成。其次，我们通过对节点坐标添加噪声来增强数据：为每个节点添加一个从正态分布 $\mathcal{N}(0,0.01)$ 中随机采样的值，从而平移节点坐标；该噪声平均相当于边长的 10%。

数据转换 在第 §3.1 节所述的网格到图转换中，节点变形和应力值被赋予节点属性和边属性。对于采用边增强的图神经网络，会在现有边连接中添加增强边，并将增强标记（增强边为 1，否则为 0）作为边属性。随后，利用节点属性、边属性和边连接，通过 PyTorch Geometric 生成图对象。

4.2 模型架构*4.2.1*

EA-GNN

创建图时，通过从图中采样节点，并在它们之间添加双向边（如果尚无现有边）来进行边增强。增强边的百分比由超参数 augmented- edge-percentage A^{perc} 决定。我们的实验采用 $A_{perc} = 20\%$。

增强图的节点属性和边属性编码器是只有一个隐藏层并采用 ReLU 激活函数的多层感知机。节点属性的多层感知机网络架构为 $14 - 64 - 128$（即 14 个输入特征、隐藏层中有 64 个节点以及 128 个输出特征），边属性的多层感知机网络架构为 $4 - 64 - 128$。随后，编码后的图依次经过交替排列的图网络（GN）模块层和多层感知机层。网络中总共加入了 6 个 GN 模块。每个 GN 模块由以下部分组成：(a) 边更新模块——边更新函数 χ 是一个多层感知机，具有单个隐藏层，其架构为 $3 \times 128 - 128 - 128$，并采用 ReLU 激活函数。输入规模源于边属性（128 个特征）与该边连接的两个节点的属性（2×128 个特征）拼接。输出为大小为 $n_e \times 128$ 的更新后边属性，其中 n_e 是边的总数。(b) 节点更新模块——该模块包含三个函数：ϕ 用于消息传递，γ用于消息聚合和节点更新，β用于最终的节点更新。这三个函数均为只有一个隐藏层并采用 ReLU 激活函数的多层感知机。函数 ϕ 的网络架构为 $2 \times 128 - 128 - 128$，其输入特征由邻居节点的属性与相应边属性拼接而成。将所有邻居节点传来的消息相加，并与选定节点的属性拼接，所得结果作为 γ 的输入，以计算节点属性

对应节点的属性。因此，γ 的网络架构为 $2\times128-128-128$。来自 γ 的输出会传递给 β，以获得更新后的节点属性，如方程 1所示。表示 β 的多层感知机网络架构为 $128-128-128$。

这四个函数的参数（χ、ϕ、γ和 β）在所有 GN 模块中共享。每个 GN 模块之后都会添加一个跳跃连接，如公式（2）所示。最终 GN 模块更新后的节点属性将作为解码器的输入。解码器是一个仅包含一个隐藏层的多层感知机，使用 ReLU 激活函数，其架构为 $128-64-2$，用于预测位移（u_x 和 u_y）；架构 $128-64-3$ 则用于预测应力（σ_{xx}、σ_{xy} 和 σ_{yy}）。我们在编码器之后以及每个 GN 模块之间使用 Dropout 层，丢弃率为 0.1，以减少过拟合。

4.2.2 M-GNN

M-GNN 的编码器架构与 EA-GNN 的相似。不过，由于不使用边属性，编码器只需处理节点属性。编码器是一个架构为 $14-64-128$ 的多层感知机。编码后的图会经过一个消息聚合模块，随后进入 ReLU 激活层。此处使用的消息聚合模块是 GraphSAGE 算子[42],，它会更新完整图中的节点属性。

接下来是一系列下采样模块，由以下操作组成：(a) 连通性增强– 通过连接相距 3 跳的节点来增强图的连通性。具体做法是将邻接矩阵 A 与自身相乘两次，即 $A^{upd}=A\times A\times A$。(b) 下采样– 将节点属性投影到一个一维可训练投影向量 $\mathbf{p}$上，并根据投影值采样 k 个节点的层。k 的取值由下采样比例 r决定。在实验中，我们设定 $r=0.6$。(c) 节点更新– 由 GraphSAGE 算子和函数 β组成的模块，如公式 3所述。

下采样模块的数量由多图深度超参数 d决定。调整超参数后，我们将该值设为 $d=3$。随后是数量相同的上采样模块，由以下操作组成：(a) 跳跃连接– 将对应下采样层的节点属性添加到更新后的节点属性中。(b) 上采样– 将先前移除的节点及其更新后的节点属性重新添加回图中的层。(c) 节点更新– 由 GraphSAGE 算子和函数 β组成的模块，如公式 3所述。

最后一个池化模块会恢复完整图，其中包含所有节点及其更新后的节点属性。随后，解码器将其解码，输出节点位移或应力。该解码器是一个网络，其架构与 EA-GNN 使用的架构相同。与 EA-GNN 类似，我们在编码器之后以及每个 GN 模块之间使用 Dropout 层，丢弃率为 0.1，以减少过拟合。

4.3 实验设置

我们采用第 §4.1 节中详述的数据生成过程，生成以下三个数据集，并按第 §4.1 节所述进行数据增强。(a) 我们生成 3000 个随机几何形状，并假设材料为均匀线弹性材料，杨氏模量 $E=100.0$和泊松比 $\nu=0.3$为常数；针对每种几何形状，设置 10 种不同的边界条件。此数据集用于评估模型捕捉几何非线性的能力。(b) 我们生成 5000 个随机几何形状，并假设材料为非线性超弹性材料。针对每种几何形状，我们同样考虑 10 种不同的边界条件。所选的非线性材料模型为 Mooney-Rivlin 模型[45],，其材料属性为 $C_{01}=0.3$ 和 $C_{10}=0.03$[46]。使用此数据集进行训练，可以测试网络对系统非线性的处理能力。(c) 我们生成 10,000 个随机几何形状，这些几何形状具有不同的超弹性材料属性；针对每种几何形状，同样考虑 10 种不同的边界条件。每种几何形状的材料属性均从均匀分布中随机选取：$C_{01}\sim\mathscr{U}(0.034, 0.34)$ 和 $C_{10}\sim\mathscr{U}(0.005,0.05)$。我们使用此数据集训练和测试架构对未见过的材料的泛化能力。

在所有实验中，数据的 70%、10% 和 20% 分别用于训练、验证和测试。对于每个数据集，我们训练四种网络来预测节点变形 u_x 和 u_y。我们将 MeshGraphNet 作为基线模型（B）。由于仿真坐标变换对模型的泛化能力有重要影响，我们还考虑了经过此类变换后训练的 MeshGraphNet 版本，并将其记为 B +SC。此外，我们还提供了使用仿真坐标的边增强图神经网络（EA-GNN +SC）和多图神经网络（M-GNN +SC）的结果。同样，我们训练这四种网络来预测节点应力值 σ_{xx}、σ_{xy} 和 σ_{yy}。

基线模型采用[25], 中使用的均方误差损失函数进行训练，而我们使用经过缩放的平均绝对误差损失函数 L 训练网络 B+SC、EA-GNN+SC 和 M-GNN+SC。缩放方式取决于与每个样本相关的边界条件组合；形式化地说，

$$L(\theta)=\frac{1}{n}\sum_{n=1}^{N}(\|\mathbf{d}_n\|_{\ell^1}+\|\mathbf{n}_n\|_{\ell^1})\|\mathbf{y}_n-\widehat{\mathbf{y}}_n\|_{\ell^1}, \tag{4}$$

其中，y 和 ˆy 分别为真实输出和预测输出，d_n 和 n_n 分别为第 n 个样本的狄利克雷载荷和诺伊曼位移向量。

对于所有网络，我们均使用 Adam优化器，并在 Tesla V100-SXM2 上对所有网络训练 1500 个周期。对于 B、B+SC 和 EA-GNN+SC，我们设定的学习率范围为 1E-4 至 1.5E-4，权重衰减为 1E-5。我们使用带热重启的余弦退火调度器来降低学习率[47]，，以避免陷入局部极小值而停滞。对于 M-GNN+SC，我们使用的学习率范围为 2E-3 至 3E-3，权重衰减为 1E-6。

4.4 数值结果

我们在第 §4.3 节中详述的三个数据集上测试所提出的模型，并将其与基线模型进行比较。为了比较模型的预测能力，我们将相对误差定义为

$$e(f) = \frac{\|\widehat{f} - f\|_{\ell^1}}{\|f\|_{\ell^1}}.$$

ˆ其中 f 表示感兴趣变量（位移值 u_x,u_y或应力值 $\sigma_{xx},\sigma_{yy},\sigma_{xy}$），f 表示网络预测的 f 值。

测试数据集上节点位移值预测的相对误差（u_x 和 u_y）以及节点应力值预测的相对误差（σ_{xx}， σ_{yy} 和 σ_{xy}）列于表 1 和 2。其中，E 表示线弹性模型，HE 表示超弹性（Mooney-Rivilin）材料模型。此外，S 表示单一材料，即整个数据集中使用同一种材料；V 表示变化材料，即数据集中使用不同的材料。

MAT.	模型	参数。	$e(u_x)\downarrow$	$e(u_y)\downarrow$
E,S	B	7.2E5	0.64	0.63
E,S	B+SC	7.2E5	0.25	0.26
E,S	EA-GNN+仿真坐标	7.2E5	**0.05**	**0.05**
E,S	M-GNN+仿真坐标	2.8E5	0.13	0.13
HE,S	B	7.2E5	0.81	0.81
HE,S	B+SC	7.2E5	0.33	0.34
HE,S	EA-GNN+仿真坐标	7.2E5	**0.08**	**0.09**
HE,S	M-GNN+仿真坐标	2.8E5	0.16	0.16
HE,V	B	7.2E5	0.73	0.74
HE,V	B+SC	7.2E5	0.28	0.29
HE,V	EA-GNN+仿真坐标	7.2E5	**0.09**	**0.09**
HE,V	M-GNN+仿真坐标	2.8E5	0.16	0.15

表1. 在不同线性和非线性材料选择及模型下预测节点位移 u_x 和 u_y 的相对误差。

MAT.	模型	$e(\sigma_{xx})\downarrow$	$e(\sigma_{yy})\downarrow$	$e(\sigma_{xy})\downarrow$
E,S	B	0.78	0.79	0.49
E,S	B+SC	0.30	0.30	0.19
E,S	EA-GNN+仿真坐标	**0.09**	**0.09**	**0.05**
E,S	M-GNN+仿真坐标	0.17	0.18	0.11
HE,S	B	0.88	0.88	0.64
HE,S	B+SC	0.30	0.32	0.29
HE,S	EA-GNN+仿真坐标	**0.11**	**0.13**	**0.09**
HE,S	M-GNN+仿真坐标	0.18	0.19	0.15
HE,V	B	0.92	0.91	0.85
HE,V	B+SC	0.39	0.39	0.29
HE,V	EA-GNN+仿真坐标	**0.12**	**0.12**	**0.09**
HE,V	M-GNN+仿真坐标	0.18	0.20	0.16

表2. 在给定不同线性和非线性材料以及神经网络模型的情况下，预测节点应力的相对误差$\sigma_{xx},\sigma_{xy},\sigma_{yy}$

如表 1、2所示，无论是否进行坐标变换（即 B 和 B+SC），EA-GNN 和 M-GNN 的表现都显著优于基线模型。这表明，边增强和多图修改都能够进一步提升消息传递效果。我们还观察到，加入坐标变换后，基线模型的性能显著提升，但与我们提出的方法相比，误差仍然很大。通过这一比较，我们可以区分坐标变换和图修改所带来的提升。值得注意的是，边增强技术始终优于所有其他方法，最佳模型的数值以粗体标出。比较单一线性弹性材料（E,S）和单一非线性超弹性材料（HE,S）对应的行，可以发现，所有模型的性能都有所下降，相对绝对误差也如预期般增大。非线性给系统带来了新的复杂性，因此学习需要更多数据。需要注意的是，进一步增加复杂性，即在测试案例之间改变非线性材料（即 HE,V）时，我们发现性能保持不变，而且所提出的模型能够泛化到未见过的材料属性。与基线方法相比，我们的模型在 HE,V 测试数据集上的测试表现相近，这表明我们的模型具有泛化能力，确实能够学习真实的物理规律，并准确预测未见过的非线性材料属性。

除相对误差值外，我们还分别在图 2a 和图 2xb 中展示了所有模型在位移 y、 σ_{xx}、 σ_{yy}uσ_{xy}和应力值 u_x、y在测试数据集上的损失值。曲线表示移动

平均值，而阴影对应方差。M-GNN 和 EA-GNN 中观察到的损失值突然跳变对应于退火过程。总体而言，我们发现坐标变换能够改善基线模型的训练效果，但只有采用边增强或多图方法时，性能才会得到显著提升。

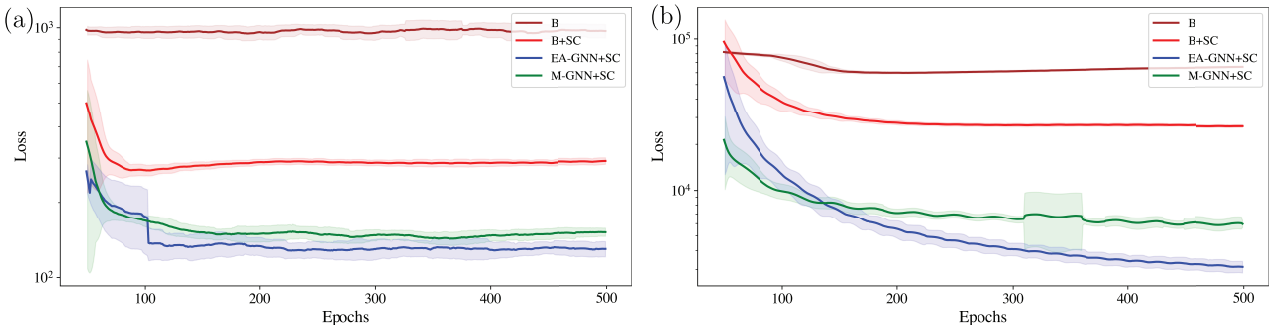

图2. 四种模型——B、B+SC、EA-GNN+SC 和 M-GNN+SC——在测试数据上预测单一材料弹性模型的（a）变形和（b）应力时的损失

接下来，为了评估模型的泛化能力，并确认模型确实学到了真实的物理规律，我们在分布外数据集上测试训练好的网络，即这些数据集不属于训练时使用的数据分布。为此，我们考虑以下三种样本外分布：（a）更小或更大的物理域，其特征是区域尺寸为原来的二分之一或两倍，而网格特征长度保持不变。尺度 = 0.5 和 2.0 的结果见表 3。（b）在边界的不同部分施加新的狄利克雷和诺伊曼边界条件。具体而言，狄利克雷和诺伊曼边界条件施加于互不相连的边界节点集合，而非相连的集合。分布外边界条件的结果见表 3，标记为 BC。（c）结合上述所有变化，并加入随机旋转和平移，即将上述数据中的图平移并旋转至不同的坐标系，以评估模型是否具有旋转不变性和平移不变性。旋转和平移的结果也列于表 3，标记为 Rot+Tra。

模型	B	B+SC	EA-GNN+仿真坐标	M-GNN+仿真坐标
尺度 = 0.5				
$e(u_x)$	0.87 ± 0.56	0.47 ± 0.29	**0.09** ± **0.07**	0.20 ± 0.11
$e(u_y)$	0.96 ± 0.64	0.47 ± 0.33	**0.09** ± **0.07**	0.20 ± 0.11
尺度 = 2				
$e(u_x)$	1.09 ± 0.67	0.85 ± 0.41	**0.22** ± **0.12**	0.29 ± 0.11
$e(u_y)$	0.95 ± 0.51	0.76 ± 0.33	**0.21** ± **0.10**	0.28 ± 0.09
BC				
$e(u_x)$	0.87 ± 0.55	0.57 ± 0.42	**0.11** ± **0.08**	0.22 ± 0.10
$e(u_y)$	0.95 ± 0.65	0.60 ± 0.43	**0.13** ± **0.09**	0.24 ± 0.10
ROT + TRA				
$e(u_x)$	0.91 ± 0.69	0.57 ± 0.42	**0.11** ± **0.08**	0.22 ± 0.10
$e(u_y)$	1.04 ± 0.78	0.60 ± 0.43	**0.13** ± **0.09**	0.24 ± 0.10

表3. 不同样本外数据分布下节点变形预测的相对误差

首先可以观察到，基于仿真坐标的训练有助于全面提升泛化能力。接下来，我们观察到，EA-GNN 的表现始终优于其他架构，其次是 M-GNN；两者均在仿真坐标系中进行评估。EA-GNN 和 M-GNN 的相对误差都接近表 1 中半缩放情况下的分布内测试误差。对于双倍缩放的情况，模型表现更差，因为图中几乎有一半对应于模型此前未见过的坐标

位置。对于 B+SC、EA-GNN+SC 和 M-GNN+SC，旋转和平移情况下的误差没有变化，因为图在输入模型前会转换到仿真坐标系；正如预期，只要模型使用仿真坐标进行训练，表现就会很好。在所有情况下，基线模型 B 的表现始终最差，其误差比 B+SC 大 1.5 到 2 倍。这是唯一一个对平移和旋转不具不变性的模型，从 B 在旋转和平移情况下较高的误差可以看出这一点。

从这些实验中，我们可以推断，EA-GNN 和 M-GNN 都能够成功学习数据背后的物理规律，并很好地泛化到未见过的区域、几何形状和边界条件，而基线模型无法做到这一点。在仿真坐标系中训练和评估模型，使其具有旋转不变性和平移不变性，因此这些模型能够有效加速物理仿真，尤其适用于需要描述区域不同部分之间远程相互作用的时不变物理仿真。

5 结论

最先进的 GNN 架构无法高效地捕捉静力学行为，因为它们难以有效描述远程相互作用。本文表明，边增强 GNN 和多重 GNN 凭借增强的连通性，能够准确且高效地捕捉静态弹性材料和超弹性材料的本构行为。此外，通过在参考坐标系中学习物理规律，我们的模型能够自动实现旋转不变性和平移不变性。通过多项数值测试，我们表明，所提出的两种架构都能高效学习时不变固体力学，并很好地泛化到未见过的材料、边界条件和区域。值得注意的是，所提出的坐标变换方法可以轻松应用于任何基于图的架构，我们也以 MeshGraphNets 为例进行了展示。据我们所知，这是首次成功利用基于图的神经网络代理模型，以平移不变和旋转不变的方式求解时不变固体力学问题，并保证泛化能力。我们的方法易于实现，可用于通过深度学习来学习具有挑战性的时不变物理系统，因此是模拟科学和工程领域复杂静态系统的理想方案。

参考文献

1. Thuerey, N. 等。《基于物理的深度学习》（WWW，2021）。2. Karniadakis, G. E. 等。物理信息机器学习。《自然评论：物理》3，422–440（2021）。3. Lagaris, I. E.、Likas, A. 与 Fotiadis, D. I. 用于求解常微分方程和偏微分方程的人工神经网络。《IEEE 神经网络学报》9，987–1000（1998）。4. Raissi, M.、Perdikaris, P. 与 Karniadakis, G. E. 物理信息神经网络：用于求解涉及非线性偏微分方程的正向和逆向问题的深度学习框架。《计算物理学杂志》378，686–707（2019）。5. Sirignano, J. 与 Spiliopoulos, K. DGM：用于求解偏微分方程的深度学习算法。《计算物理学杂志》375，1339–1364（2018）。6. Yu, B. 等。深度 Ritz 方法：一种用于求解变分问题的基于深度学习的数值算法。《数学与统计学通讯》6，1–12（2018）。7. Khoo, Y.、Lu, J. 与 Ying, L. 使用人工神经网络求解参数化偏微分方程问题。《欧洲应用数学杂志》32，421–435（2021）。8. Ruthotto, L. 与 Haber, E. 由偏微分方程启发的深度神经网络。《数学成像与视觉》62，352–364（2020）。9. Geneva, N. 与 Zabaras, N. 使用物理约束深度自回归网络对偏微分方程系统的动力学进行建模。《计算物理学杂志》403，109056（2020）。10. Long, Z.、Lu, Y. 与 Dong, B. PDE-Net 2.0：利用数值-符号混合深度网络从数据中学习偏微分方程。《计算物理学杂志》399，108925（2019）。11. Bar-Sinai, Y.、Hoyer, S.、Hickey, J. 与 Brenner, M. P. 学习偏微分方程的数据驱动离散化。《美国国家科学院院刊》116，15344–15349（2019）。12. Kochkov, D. 等。机器学习加速的计算流体力学。《美国国家科学院院刊》118，e2101784118（2021）。13. Zhuang, J.、Kochkov, D.、Bar-Sinai, Y.、Brenner, M. P. 与 Hoyer, S. 二维湍流中被动标量平流的学习得到的离散化。《物理评论：流体》6，064605（2021）。

14. Han, J.、Jentzen, A. 与 E, W. 使用深度学习求解高维偏微分方程。《美国国家科学院院刊》115，8505–8510（2018）。15. Li, Z. 等。神经算子：用于偏微分方程的图核网络。arXiv 预印本 arXiv:2003.03485（2020）。16. Li, Z. 等。用于参数化偏微分方程的多极图神经算子。《神经信息处理系统进展》33，6755–6766（2020）。17. You, H.、Yu, Y.、D'Elia, M.、Gao, T. 与 Silling, S. 非局部核网络（nkn）：一种稳定且与分辨率无关的深度神经网络。《计算物理学杂志》469，111536（2022）。18. Iakovlev, V.、Heinonen, M. 与 Lähdesmäki, H. 使用图神经网络从稀疏数据中学习连续时间偏微分方程。arXiv 预印本 arXiv:2006.08956（2020）。19. Belbute-Peres, F. D. A.、Economon, T. 与 Kolter, Z. 结合可微偏微分方程求解器和图神经网络进行流体流动预测。载于《国际机器学习会议》，2402–2411（PMLR，2020）。20. Alet, F. 等。图元素网络：自适应的结构化计算与记忆。载于《国际机器学习会议》，212–222（PMLR，2019）。21. Li, Z. 等。用于参数化偏微分方程的傅里叶神经算子。arXiv 预印本 arXiv:2010.08895（2020）。22. Li, Z.、Huang, D. Z.、Liu, B. 与 Anandkumar, A. 具有学习得到的变形、适用于一般几何的偏微分方程傅里叶神经算子。arXiv 预印本 arXiv:2207.05209（2022）。23. Lu, L.、Jin, P.、Pang, G.、Zhang, Z. 与 Karniadakis, G. E. 基于算子普适逼近定理，通过 DeepONet 学习非线性算子。《自然·机器智能》3，218–229（2021）。24. Poli, M. 等。图神经常微分方程。arXiv 预印本 arXiv:1911.07532（2019）。25. Pfaff, T.、Fortunato, M.、Sanchez-Gonzalez, A. 与 Battaglia, P. W. 使用图网络学习基于网格的模拟。CoRR abs/2010.03409（2020）。2010.03409。26. Fortunato, M.、Pfaff, T.、Wirnsberger, P.、Pritzel, A. 与 Battaglia, P. 多尺度 MeshGraphNets。arXiv 预印本 arXiv:2210.00612（2022）。27. Sanchez-Gonzalez, A. 等。使用图网络学习模拟复杂物理。载于 III, H. D. 与 Singh, A.（编），《第 37 届国际机器学习会议论文集》，第 119 卷《机器学习研究论文集》，8459–8468 （PMLR，2020）。28. Liu, N.、Yu, Y.、You, H. 与 Tatikola, N. Ino：具有动量守恒、用于学习复杂物理系统的不变神经算子。arXiv 预印本 arXiv:2212.14365（2022）。29. Zhou, J. 等。图神经网络：方法与应用综述。AI Open 1，57–81（2020）。30. Yao, L.、Mao, C. 与 Luo, Y. 用于文本分类的图卷积网络。载于《AAAI 人工智能会议论文集》，第 33 卷，7370–7377（2019）。31. Veli˘ckovi´c, P. 等。图注意力网络。arXiv 预印本 arXiv:1710.10903（2017）。32. Seo, Y.、Defferrard, M.、Vandergheynst, P. 与 Bresson, X. 使用图卷积循环网络进行结构化序列建模。载于《国际神经信息处理会议》，362–373（Springer，2018）。33. Shlomi, J.、Battaglia, P. 与 Vlimant, J.-R. 粒子物理中的图神经网络。《机器学习：科学与技术》2，021001（2020）。34. Ju, X. 等。用于高能物理探测器中粒子重建的图神经网络。arXiv 预印本 arXiv:2003.11603（2020）。35. Donon, B.、Donnot, B.、Guyon, I. 与 Marot, A. 电力系统图神经求解器。载于《2019 年国际神经网络联合会议（IJCNN）》，1–8（IEEE，2019）。36. Bergström, J. 5 - 弹性/超弹性。载于 Bergström, J.（编）《固体聚合物力学》，209–307，DOI: https://doi.org/10.1016/B978-0-323-31150-2.00005-4（William Andrew Publishing，2015）。37. Zhao, L. 与 Akoglu, L. PairNorm：应对图神经网络中的过平滑。arXiv 预印本 arXiv:1909.12223 （2019）。38. Wu, Z. 等。利用全局注意力为图神经网络表示长程上下文。《神经信息处理系统进展》34，13266–13279（2021）。39. Hamilton, W. 人工智能与机器学习综合讲座。图表示学习（2020）。

40. Briggs, W.、Henson, V. 与 McCormick, S.《多重网格教程》，第 2 版（2000 年）。41. Gao, H. 与 Ji, S. 图 u-net。载于 Chaudhuri, K. 与 Salakhutdinov, R.（编）《第 36 届国际机器学习会议论文集》，《机器学习研究论文集》第 97 卷，2083–2092 页（PMLR，2019 年）。42. Hamilton, W.、Ying, Z. 与 Leskovec, J. 大型图上的归纳式表示学习。载于 Guyon, I. 等（编）《神经信息处理系统进展》第 30 卷（Curran Associates, Inc.，2017 年）。43. Geuzaine, C. 与 Remacle, J.-F. Gmsh：具有内置前处理与后处理功能的三维有限元网格生成器。《工程数值方法国际期刊》79，1309–1331 页（2009 年）。44. Manual, A. U. Abaqus 用户手册。Abacus（2020 年）。45. Liu, I.-S. 等。关于 Mooney–Rivlin 材料模型的注记。《连续介质力学与热力学》24，583–590 页（2012 年）。46. Kumar, N. 与 Rao, V. V. 超弹性 Mooney-Rivlin 模型：材料常数的确定与物理解释。《参数》2，01（2016 年）。47. Loshchilov, I. 与 Hutter, F. SGDR：采用热重启的随机梯度下降。arXiv 预印本 arXiv:1608.03983（2016 年）。

6 致谢

R.G. 和 A.Z. 感谢 Prajjwal Jamdagni 就有限元数值模拟进行了深入而细致的讨论。A.Z. 和 V.S. 感谢 Abhishek Sharma 在项目启动之初给予的鼓励与支持，并将始终铭记他的指导与培养。

作者贡献声明

A.Z. 和 V.S. 提出了本研究的构想。R.G. 和 H.R. 负责开展训练人工智能模型所需的有限元方法模拟。R.G.、M.D. 和 A.Z. 设计了人工智能架构，R.G. 则负责实施并开展人工智能模型的训练与测试。所有作者均参与了结果分析、稿件撰写以及最终版本的审阅。

其他信息

待补充 , in 顺序如下：登录代码（如适用）；利益冲突（强制性声明） .

通讯作者负责代表论文的所有作者提交一份利益冲突声明。该声明必须包含在提交的文章文件中。