arXiv:2512.15083v1 [cs.LG] 17 Dec 2025

用于弹性仿真的神经模块化物理

Yifei Li* **Haixu Wu Zeyi Xu TuurStuyck Wojciech Matusik**麻省理工学院计算机科学与人工智能实验室 麻省理工学院计算机科学与人工智能实验室 上海大学 Meta Reality Labs 麻省理工学院计算机科学与人工智能实验室

摘要

基于学习的方法在物理仿真方面取得了显著进展，通常通过端到端优化的单体神经网络来近似动力学。尽管这些模型提供了一种有效的仿真方式，但与传统数值模拟器相比，它们可能会丢失一些关键特性，例如物理可解释性和可靠性。受以模块化方式运行的经典模拟器启发，本文提出了用于弹性仿真的神经模块化物理（NMP），将神经网络的近似能力与传统模拟器的物理可靠性结合起来。不同于以往的单体学习范式，NMP 将弹性动力学分解为具有物理意义的神经模块，并通过中间物理量将其连接起来，从而能够直接监督中间物理量和物理约束。借助专门设计的架构和训练策略，我们的方法将数值计算流程转化为模块化神经模拟器，从而提高了物理一致性和泛化能力。实验表明，与其他神经模拟器相比，NMP 对未见过的初始条件和分辨率具有更强的泛化能力，能够稳定地进行长时程仿真，更好地保留物理属性；此外，在底层动力学未知的场景中，它也比传统模拟器具有更强的可行性。

1 引言

基于学习的模拟器已成为建模复杂物理系统的一种强大方法；在这种方法中，神经网络通常通过数据进行端到端的整体优化，以近似物体动力学。代表性的神经模拟器包括神经算子（Lu et al., 2019; Li et al., 2021; Kovachki et al., 2023; Wu et al., 2024b）和图神经网络（GNN）模拟器（Sanchez-Gonzalez et al., 2018; 2020; Pfaff et al., 2021; Lam et al., 2023; Halimi et al., 2023），它们有望实现快速、灵活且可微的仿真，并已在流体动力学（Tompson et al., 2017）、材料科学（Friederich et al., 2021）等领域展现出良好的效果。然而，上述神经模拟器通常存在物理合理性方面的关键局限。具体而言，神经模拟器往往表现为不可分解的函数，唯一可访问的物理量是模型输出，因此无法获取仿真所必需的中间物理量（例如内力、应力）。这种不透明性严重损害了模拟器的可解释性，也使得施加已知物理约束变得困难。尽管一些物理信息神经网络（Raissi et al., 2019; Cuomo et al., 2022; Guo et al., 2020; Wang & Zhong, 2024; Wang et al., 2021; Stuyck et al., 2025）能够利用物理方程显式优化模型，但它们通常面临严重的训练困难（Daw et al., 2023; Wu et al., 2024a），且主要局限于简单的方程和场景（Wang et al., 2023; 2024），难以支持本文关注的复杂长时域动力学仿真。

请注意，物理轨迹远不止是遵循模糊数据分布的状态序列。以弹性软体的动力学为例，每一步的动力学都受全局守恒定律、局部几何约束和材料特定的本构关系支配，这些因素共同将运动限制在数学上可能的状态空间的狭小子集中，因此模拟器必须具备物理合理性。事实上，与近期的神经模拟器不同，我们注意到，传统数值模拟器会仔细遵守上述所有物理定律，例如有限元方法（Sol´ın，2005年）。总体而言，这些经典模拟器采用模块化的方式运行，

*通讯作者。

将复杂物理动力学分解为若干子模块，并基于物理先验知识定义中间量之间的相互作用。经典模拟器中的模块化计算流程不仅将复杂动力学分解为更易处理的子过程，还提供了灵活的接口，用于约束物理量，以确保严格的物理合理性。这一观察促使我们将经典模拟器的模块化引入基于学习的模拟器。

此前，研究人员探索过一些混合神经-物理模拟器，方法是将经典模拟器中的某个模块替换为可学习神经网络，例如替换离散化模板（Bar-Sinai 等人，2019年）或本构定律（Ma 等人，2023年）。这些工作可视为对模块化动机偶然且初步的探索。然而，纯数值方法在信息未知的场景中（例如环境摩擦力）适应性较差，因此这些混合方法只能取得折中的性能，以牺牲数据驱动神经模拟器的场景灵活性为代价，换取更好的物理正确性。这些零散且不尽如人意的方法自然引出了一个问题：我们能否设计出一种具有完备模块化架构的神经模拟器，同时保留数据驱动的灵活性与物理合理性？经过全面的实证研究，我们发现，物理仿真的模块化远不止简单替换，还需要精心设计，才能忠实确保与数值模拟器的物理对齐，并避免复杂模块化架构中的崩溃（Mittal 等人，2022年）。“崩溃”指模块能力混乱，这是模块化神经网络的一个基础性问题（Jarvis 等人，2023年），会导致不理想的模块行为和整体性能。

本文提出了一种神经模块化物理 （NMP）框架，成功将模块化学习理念融入神经弹性模拟，其中包括专用架构和模块化物理训练策略。具体而言，我们的方法将物理模拟分解为定义明确的模块，并以专用神经网络替换本构模型和时间积分方案等关键组件。这种与物理对齐的模块化设计将神经模拟器内部的关键物理量显式呈现出来，从而能够直接监督中间量，并灵活地在神经网络与数值实现之间切换。此外，为避免模块化架构潜在的崩溃，NMP采用两阶段训练策略：首先，利用数值模拟器提供的中间物理量进行监督，对各组件独立训练；然后再将它们联合微调，从而实现各模块的专门化 ，并使其与物理更加严格地对齐。此外，得益于模块化设计，NMP能够灵活地施加物理约束。通过利用显式呈现的中间物理量（例如弹性体的变形梯度），NMP能够施加更精细的物理约束，例如局部体积保持；这是以往端到端神经模拟器无法实现的，展现了这一全新模块化学习范式的潜力。我们通过在多种弹性模拟任务上开展全面实验，展示了这些优势。我们的贡献总结如下：

- 我们提出并研究一种全新的、全面的模块化神经框架神经模块化物理，用于弹性仿真。每个经典子系统都可以替换为具有明确定义物理接口的神经模块，从而能够直接监督中间量、与基于物理的方法互换，并灵活施加物理约束。
- 我们提出专门的架构和模块化物理训练策略，以确保与数值方法更好地实现物理对齐，并避免复杂模块化架构发生坍塌，从而成功保证训练稳定性和模块专门化。
- NMP 对未见过的初始条件和分辨率展现出很强的泛化能力，能够生成准确的长时域滚动预测，其表现优于先进的神经模拟器；在信息未知的场景中，它也比混合模拟器展现出更好的灵活性。

2 相关工作

2.1 神经模拟器

深度模型已被广泛应用于物理模拟（Pfaff 等人，2021年；Ma 等人，2023年；Wu 等人，2024b）。以往的神经模拟器大致可分为以下两种范式：端到端神经模拟器和混合神经-物理模拟器。

在端到端神经模拟器中，各种架构的神经网络被用作整体近似器，通过数据监督来模拟物理动力学。例如，卷积神经网络（Guo 等人，2016；Tompson 等人，2017；Afshar 等人，2019；Liu 等人，2023）已被用于

捕捉欧拉物理系统中的空间模式。为处理复杂几何形状，基于图神经网络（GNNs）的方法，例如 MeshGraphNet 等（Sanchez-Gonzalez 等人，2020；Pfaff 等人，2021；Lam 等人，2023；Sanchez-Gonzalez 等人，2018；Halimi 等人，2023），利用消息传递对系统元素之间的复杂相互作用进行建模。近年来，神经算子（Li等人，2021；Lu 等人，2019；Kovachki 等人，2023；Wu 等人，2024b）被提出，用于在函数空间中近似物理过程，从而提供一个能够跨不同分辨率和边界条件预测物理过程的框架。尽管这些方法前景可观，但它们依赖端到端架构，用单一网络对整个物理系统建模，因此难以保证物理定律。相比之下，我们通过模块化架构解耦复杂物理，从而实现更好的物理可解释性，并支持更灵活的物理约束。

另一方面，近期物理模拟领域的进展为刚体 Popovi´c 等人（2003）；Xu 等人（2021）、软体 Hu 等人（2019）；Geilinger 等人（2020）、流体 McNamara 等人（2004）；Schenck 和 Fox（2018）；Li等人（2024b），以及布料 Li 等人（2022）；Liang 等人（2019）；Li 等人（2024a）提出了完全可微的求解器。尽管这些可微模拟器支持基于梯度的优化，但它们仍采用固定的解析物理流程，使学习的范围在很大程度上局限于参数辨识。这种僵化限制了它们融入数据驱动校正或处理未知或不完整物理规律的能力。相比之下，我们的方法以模块化的学习得到的物理组件取代固定的解析流程，使数据驱动的适应能够在仿真的多个阶段进行，同时保留物理结构。

至于神经网络-物理混合方法，这些方法将学习嵌入经典求解器的特定阶段，在保留物理基础的同时，捕捉解析模型遗漏的效应。例如，学习得到的数据驱动离散化模板（Bar-Sinai 等人，2019），以及在解析动力学基础上学习残差动力学，以补足未建模部分（Yin 等人，2021）。在弹性仿真领域，NCLaw（Ma 等人，2023）学习本构定律。然而，现有的混合方法通常只替换部分组件，其余流程仍交由经典求解器处理，因此在场景灵活性和物理一致性方面表现有所折损。我们通过将整个弹性仿真流程模块化，推进了这一方向。全面的模块化在保留经典力学组织原则的同时，发挥了学习方法的表达能力。

2.2 神经模块化网络

神经模块化网络通过将不同子任务分配给独立的子网络，促进了专门化和可解释性。Andreas 等人（2016）提出了模块化这一理念，通过动态组合面向特定任务的图来进行语言推理，随后该方法被用于视觉问答和程序综合（Kirsch 等人，2018；Lu 等人，1995）。尽管神经网络的模块化已被证明在广泛的应用中具有优势，但其有效性从根本上取决于所学习模块的专门化程度（Mittal 等人，2022年；Jarvis 等人，2023年），而这一点在物理模拟器中尚未得到充分探索。本文观察到数值模拟器内部存在模块化特性，因此提出神经模块化物理，以模块化物理计算过程，并配合专门设计的架构和训练策略来避免模块坍塌。

3 神经模块化物理

如前所述，本文旨在构建一个全面的模块化框架，用于弹性仿真，以提升场景灵活性和物理合理性。为确保物理对齐和模块专门化，我们提出神经模块化物理（NMP），并为其设计了专门的模块化架构和训练策略。本节将首先介绍经典模拟器的一些基础知识和见解，然后详细说明 NMP 的形式化设计与训练方法。

3.1 经典模拟器带来的启发

由于模块专门化是神经模块化网络中的关键（Mittal 等人，2022年），因此确定模型模块通常很困难，但也至关重要。幸运的是，与语言或视觉应用不同，物理模拟有一个长期以来的参考，即经典数值模拟器。因此，我们提出遵循经典模拟器的计算过程，这不仅提供了原生的异构模块，还为后续优化保留了具有物理意义的接口。

具体而言，可变形物体的运动遵循牛顿第二定律 $\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}_{\text{int}}(\mathbf{x}) + \mathbf{f}_{\text{ext}}$, 其中 $\mathbf{x} \in \mathbb{R}^{3n}$ 按顺序堆叠了一个包含 n 个顶点的网格的节点位置，$\mathbf{M}$ 是（集总）质量矩阵，$\mathbf{f}_{\text{int}}$ 表示内部弹性力，而 $\mathbf{f}_{\text{ext}}$ 汇集了重力或接触等外载荷。

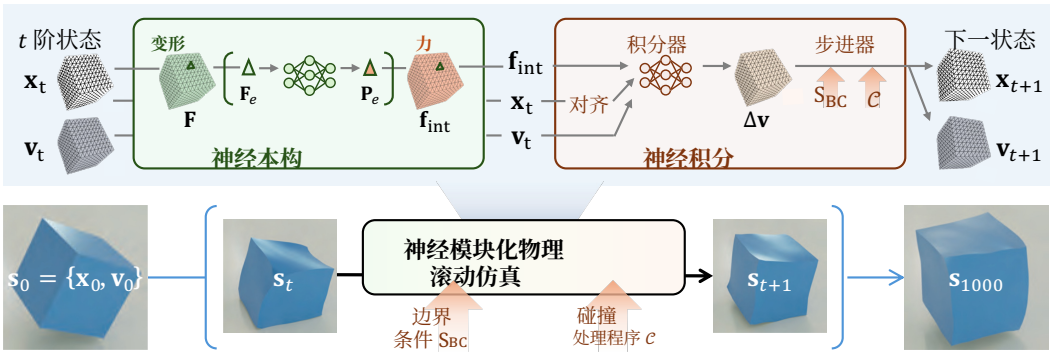


图 1：神经模块化物理（NMP）将弹性物理仿真模块化为两部分：用于计算内力的神经本构模块 $\mathbf{f}_{\text{int}}$，以及用于演化系统状态 $\mathbf{s}_t = \{\mathbf{x}_t,\ \mathbf{v}_t\}$ 并处理传统边界条件 $\mathbf{S}_{BC}$ 和碰撞 $\mathcal{C}$ 的神经积分模块。

基于四面体或六面体单元离散化的标准有限元方法（FEM）涉及以下计算过程：(i) 首先根据节点位移计算每个单元内部的变形梯度$\mathbf{F}$，然后将单元力组装到全局系统中；(ii) 同时进行时间积分以仿真动力学。上述计算过程显然导出了两个解耦的模块：空间力计算（称为本构定律）和时间积分。受此观察启发，NMP 将弹性仿真模块化为神经本构模块和神经积分模块，具体内容将在下一节中详细介绍。

3.2 弹性物理模块化

借鉴经典仿真器，我们将传统计算过程分解为两个连续的神经模块，每个模块都与定义明确的物理子过程相对应（图 1）。

具体而言，给定包含顶点位置$\mathbf{s}_t$ 和速度$\mathbf{x}_t$ 的系统状态$\mathbf{v}_t$，神经本构模块将根据节点位移计算出的变形梯度$\mathbf{F}$映射为内力$\mathbf{f}_{\text{int}}$；神经积分则演化$\mathbf{s}_t$，并在每个步骤中通过施加边界条件（$\mathbf{S}_{BC}$）和处理碰撞（$\mathcal{C}$）来完成计算，同时保持传统模拟器的计算流程（算法 1）。以下介绍各个模块的具体细节。

算法 1： 神经模块化物理

输入： $\mathbf{s}_t = \{\mathbf{x}_t, \mathbf{v}_t\}, \mathbf{S}_{BC}, \mathcal{C}$
输出： $\mathbf{s}_{t+1} = \{\mathbf{x}_{t+1}, \mathbf{v}_{t+1}\}$
// 神经本构模块
对于每个 元素 e，**执行**
  计算变形梯度 $\mathbf{F}_e$，由 $\mathbf{x}_t$
  预测应力 $\mathbf{P}_e = f_\theta(\mathbf{F}_e)$
  计算单元力 $\mathbf{f}_{\text{int}}^e = \mathbf{B}_e^\top \mathbf{P}_e V_e$
组装全局力 $\mathbf{f}_{\text{int}} = \sum_{e \text{ 整数}} \mathbf{f}^e$
// 神经积分模块
$\Delta\mathbf{v} = g_\phi(\mathbf{x}_t,\ \mathbf{v}_t,\ \mathbf{f}_{\text{int}})$
$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta\mathbf{v}$
$\mathbf{x}_{t+1} = \mathbf{x}_t + h\mathbf{v}_{t+1}$
$\mathbf{x}_{t+1},\ \mathbf{v}_{t+1} = \mathcal{C}(\mathbf{S}_{BC}(\mathbf{x}_{t+1},\ \mathbf{v}_{t+1}))$

神经本构模块 在有限元法中，内力$\mathbf{f}_{\text{int}}$ 由材料的本构定律通过以下过程确定。首先，应变能密度函数 $\Psi(\mathbf{F})$ 定义材料对变形的响应，其中$\mathbf{F} = \partial_{\mathbf{x}} \partial\mathbf{x}$ 是变形梯度，用于衡量从参考坐标$\mathbf{X}$ 到变形后坐标$\mathbf{x}$的局部几何畸变。随后，第一类皮奥拉-基尔霍夫应力$\mathbf{P}$计算为$\mathbf{P} = \partial\Psi(\mathbf{F}) \partial\mathbf{F}$，表示材料抵抗变形的能力。内部力通过对所有单元求和计算：$\mathbf{f}_{\text{int}} = \sum_e \mathbf{B}_e^\top \mathbf{P} V_e$，其中$\mathbf{B}_e$ 是单元 e 的应变-位移矩阵，V_e 是该单元在参考构形中的体积。

在此模块中，经典计算流程保持不变，只是将应力项$\mathbf{P}$ 的计算替换为可学习的神经网络 f_θ，因为在现实场景中，材料响应通常难以测量，而数据驱动范式更加灵活。此处，f_θ 配置为一种特殊的旋转等变架构，遵循 Ma 等人（2023）的方法。具体而言，给定$\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$，我们将旋转不变元组 $(\mathbf{\Sigma}, \mathbf{F}^\top\mathbf{F}, \det\mathbf{F})$ 输入两层多层感知机，再通过$\mathbf{R} = \mathbf{U}\mathbf{V}^\top$将输出旋转回去，得到$\mathbf{P} = \mathbf{R}\, f_\theta(\cdot)$。上述模块化设计保留了有限元法的逐单元计算结构，同时利用学习来捕捉传统模型在现实应用中可能无法捕捉的复杂材料行为。

值得注意的是，f_θ 采用自 Ma 等人（2023）的参数化方法只是我们全面模块化框架的一部分；该框架旨在实现完全模块化的神经模拟器，并采用更完善的方法来应对物理模块化和模型优化中新出现的难题。

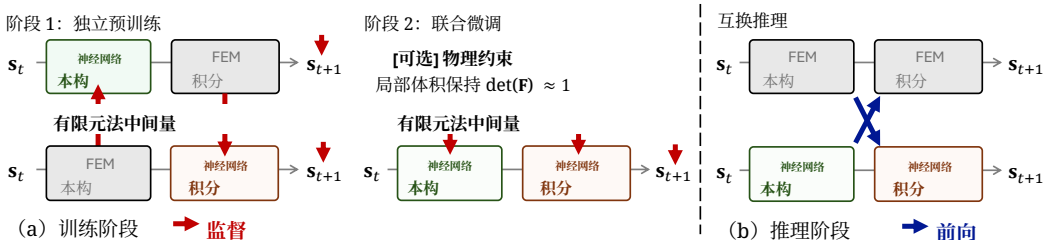


图2：模块化物理训练策略与推理方案。采用两阶段训练范式来优化神经模块：第一阶段的分离训练可有效避免模块坍塌，第二阶段的联合微调则能进一步利用物理约束。

神经积分模块 在经典有限元法中，运动方程可以通过各种方案随时间积分，其中最常见的是显式欧拉法和隐式欧拉法：

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h\mathbf{v}_{t+\alpha},\ \mathbf{v}_{t+1} = \mathbf{v}_t + h\mathbf{M}^{-1}\left(\mathbf{f}_{\text{int}}(\mathbf{x}_{t+\alpha}) + \mathbf{f}_{\text{ext}}\right), \tag{1}$$

其中， h 为时间步长， α 的选择决定了积分方案： $\alpha = 0$（显式欧拉法）和$\alpha = 1$（隐式欧拉法）。$\mathbf{M}$ 表示（集总）质量矩阵，$\mathbf{f}_{\text{ext}}$ 表示外力，两者在现实应用中都难以获取。例如，由于物体可能是中空或实心的，质量矩阵$\mathbf{M}$ 难以确定；而环境的摩擦力则取决于接触材料。因此，我们尝试以可学习的方式计算$\mathbf{M}$和 $\mathbf{f}_{\text{ext}}$ 相关项。

具体而言，神经积分模块保留了与隐式有限元法相同的接口，但使用可学习的神经网络来近似速度增量 $\Delta\mathbf{v}$ （式 1 中与$\mathbf{M}$,$\mathbf{f}_{\text{ext}}$相关的项）：

$$\Delta\mathbf{v} = g_\phi(\mathbf{x}_t, \mathbf{v}_t, \mathbf{f}_{\text{int}}) \quad \text{with} \quad \mathbf{v}_{t+1} = \mathbf{v}_t + \Delta\mathbf{v},\ \mathbf{x}_{t+1} = \mathbf{x}_t + h\,\mathbf{v}_{t+1}. \tag{2}$$

对于 g_ϕ 的构型，一个主要难点在于，不同场景和模拟步中的原始坐标$\mathbf{x}_t$ 会因全局平移和旋转而发生变化。因此，我们在前面添加一个可学习的规范化模块：一个两层 T-Net（Qi 等人，2017年）预测一个 3×3 线性映射$\mathbf{T}$ ，将变形网格对齐到一个学习得到的参考坐标系， $\tilde{\mathbf{x}}_t = \mathbf{T}(\mathbf{x}_t - \bar{\mathbf{x}}_t)$，其中 $\bar{\mathbf{x}}_t$ 是网格质心。随后，我们用多层感知机层对每个顶点的对齐后的位置、原始位置、速度和内力进行嵌入；将这些嵌入与原始输入拼接，得到一个描述网格状态的紧凑特征向量。之后，另一个多层感知机将该特征映射为速度增量 $\Delta\mathbf{v} \in \mathbb{R}^3$。

除了精细的物理对齐之外，上述模块化设计还具有若干优点。首先，它在模块接口处公开中间量，这些中间量可在训练期间直接由有限元法的中间量进行监督，从而进一步提升模型可解释性和模块专门化。其次，神经模块与传统模拟器保持相同的接口，支持与其基于物理的对应模块互换，以便灵活部署（图 2b）。

3.3 模块化物理训练

正如 Mittal 等人（2022年）所证明的，模块化架构本身并不足以保证整个模型以模块化方式运行；训练策略同样至关重要。因此，如图 2 所示，我们提出了一种两阶段训练策略，以优化预定义的神经模块。

分离训练 在第一阶段，我们独立训练神经本构模块和神经积分模块，同时对另一个组件使用相应的可微仿真。具体而言，对于每个时间步长，监督信号可形式化如下：

$$\begin{aligned} &\text{Neural constitutive module } f_\theta\text{: } \mathcal{L}_{\text{constitutive}} = \|\mathbf{f}_{\text{int}} - \mathbf{f}^*_{\text{int}}\|_2^2 + \|\mathbf{x}_{\text{FEM-Integration}} - \mathbf{x}^*\|_2^2 \\ &\text{Neural integration module } g_\phi\text{: } \mathcal{L}_{\text{integration}} = \|\mathbf{v} - \mathbf{v}^*\|_2^2 + \|\mathbf{x}_{\text{FEM-constitutive}} - \mathbf{x}^*\|_2^2, \end{aligned} \tag{3}$$

为简化起见，此处省略时间步长下标。 $\cdot^*$ 表示由有限元法生成的监督信号。得益于有限元法独特的可处理特性，我们还可以直接监督神经模块的中间结果，确保每个网络都能学会其特定作用，即材料行为或时间动力学。此外，我们还利用可微有限元法模块及其对应的神经模块生成仿真结果，其中对最终结果$\mathbf{x}$ 的监督可作为正则化项，确保每个模块的输出都具有物理意义。

联合微调 完成分离训练后，我们获得了两个经过充分优化的模块。随后，我们对两个网络进行联合微调，使它们能够相互适应，同时保留预训练所获得的物理理解。值得注意的是，借助可获取的中间物理量（例如变形梯度），NMP 还能够解析地计算物理约束；由于单体式神经模拟器具有不透明性，此前无法实现这一点。

为了展示我们的框架在施加物理约束方面的灵活性，我们进一步探索将弹性动力学的局部体积守恒纳入损失函数，以提升仿真的物理一致性。具体而言，对于近乎不可压缩的弹性体，该守恒特性意味着变形梯度的行列式 det(**F**) 应保持接近 1。因此，我们还使用铰链损失对过大的偏差进行惩罚（Gentile & Warmuth, 1998）：

$$\mathcal{L}_{\text{volume}} = \sum\nolimits_{e} \left[\max\left(|\det(\mathbf{F}_e) - 1| - \epsilon_{\text{volume}},\, 0\right)\right]^2, \tag{4}$$

其中，e 表示元素索引，我们按照工程实践将 $\epsilon_{\text{volume}} = 0.05$ 设为 $\epsilon_{\text{volume}} = 0.05$。微调时，将这一新增惩罚项以 0.1 的权重加到中间物理量和模型预测的 L2损失上。在实验中，这一物理约束不会显著提高准确率指标，但能持续提升模型预测的物理合理性。

请注意，上述局部体积守恒物理约束仅为一个示例。借助我们的模块化物理框架，可以灵活地在训练过程中加入新的物理正则化，例如神经网络模拟器与有限元模拟器的弹性势能 $\sum_e V_e\, W_e(\mathbf{F}_{e,t})$ 之间的 L2 距离，其中 V_e 和 $W_e(\cdot)$ 分别表示静止体积和应变能密度。

4 实验

我们构建了五种测试环境，用于评估弹性仿真的不同方面。前四种环境提供完整的物理信息，用于测试模型的物理学习能力；最后一种环境则引入未知摩擦，以模拟现实世界中的挑战。

CUBE 我们仿真一个弹性立方体（1,000 个顶点，3,645 个四面体）在重力作用下掉落并在刚性地面上反弹的过程。该场景用于测试外部碰撞处理和一般变形行为。验证集通过改变初始速度和立方体朝向来测试泛化能力，并考察碰撞响应以及大幅动力学变化下的一般变形行为。

CUBEXL 此外，我们还构建了一个由 $22 \times 22 \times 22$ 网格组成的高分辨率立方体 CUBEXL（10,648 个顶点），用于对模型处理大分辨率网格时的表现进行压力测试。

SPOT 小牛“Spot”（Crane et al., 2013）（1,015 个顶点，2,752 个四面体）的头部和尾部均受到约束。该环境用于测试模型对多个固定边界点的处理，以及复杂几何形状中的应力传播。验证集改变初始速度。

BOB 一只弹性鸭子“Bob”（Bob）（849 个顶点、3,108 个四面体）悬浮在空中，头部顶点固定。此环境用于测试该方法处理固定点处狄利克雷边界条件以及复杂几何形状变形的能力。验证数据集改变了初始速度。

对于这四种环境，我们采用新胡克材料模型（Smith 等，2018），时间步长为 $h = 5 \times 10^{-4}$ 秒，并使用半隐式积分器进行计算。每个数据集包含 32 条训练轨迹，每条轨迹有 1,000 个步长，初始速度随机采样；此外还包含 8 条具有未见初始条件的验证轨迹。更多详情见附录 B。

FRICTION 最后一项基准测试引入了未知摩擦动力学。一只弹性橡胶鸭“Bob”（Bob）（5,130 个顶点、21,448 个四面体）被放落到粗糙表面上，并在摩擦接触动力学作用下开始滑动。轨迹由外部投影动力学模拟器（Du 等，2021）生成，该模拟器通过隐式积分方案同时模拟超弹性材料响应和摩擦接触。与此前接触仅体现为简单非穿透约束的环境不同，此设置要求神经网络捕捉未明确提供的丰富摩擦效应。因此，这项基准测试评估了模型泛化到更真实接触现象的能力，以及直接从数据中推断缺失动力学的能力。

所有仿真结果均可在 https://sites.google.com/view/neural-modular-physics 查看 .

4.1 主要结果

基线方法 如引言所述，我们提出的 NMP 是一个完整的神经模块化网络。因此，我们选取三种具有代表性的神经模拟器作为基线方法：基于图神经网络的网格-

表 1：我们报告了不同展开时域下，模型仿真的顶点位置与真实模拟之间的均方根误差。基线可视化见附录 C.1。

方法	CUBE			CUBEXL			SPOT			BOB		
	100	500	1000	100	500	1000	100	500	1000	100	500	1000
Transolver（2024b）	0.277	1.525	3.200	0.362	3.622	9.984	0.229	0.943	2.290	0.075	0.597	1.782
EGNO（2024年）	0.503	2.308	4.492	0.662	3.288	6.357	0.370	1.176	1.217	0.299	1.414	2.770
MeshGraphNet（2020年）	0.151	0.371	1.554	0.082	0.433	1.963	0.066	0.577	2.364	**0.048**	0.508	2.104
Ours	**0.019**	**0.159**	**0.535**	**0.020**	**0.253**	**0.440**	**0.061**	**0.217**	**0.449**	0.050	**0.316**	**0.579**

GraphNet（MGN）（Pfaff 等人，2020年），以及先进的神经算子 EGNO（Xu 等人，2024年）和基于 Transformer 的 Transolver（Wu 等人，2024b）。此外，纯神经模拟器和混合神经-物理模拟器也能够生成仿真，我们将在下一节对其进行比较。

设置 我们在所有环境中，使用来自未见过的测试轨迹的 100、500 和 1，000 步滚动推演评估所有方法，以揭示模型的长时域稳定性以及不同持续时间内累积的物理误差。请注意，训练期间，我们的模型仅针对最多 200 步的序列进行监督，因此 500 步和 1，000 步滚动推演属于时间外推。

结果 如表 1 所示，与其他神经模拟器相比，NMP 的精度显著更高，在 1,000 步滚动推演中的误差降低了 60% 以上，彰显了模块化设计在捕捉长时域动力学方面的优势。值得注意的是，尽管我们对先进的神经算子 EGNO 和 Transolver 进行了精心调优，它们在长时程仿真中仍然表现不佳。这可能是因为缺乏对中间物理量的显式监督，这些单体式神经模拟器很可能会生成毫无意义的结果，尤其是在长时程滚动推演中。可参阅附录 C.1 和项目网站查看可视化对比，附录 C.4 查看表 1 中各序列标准差的统计数据，并参阅附录 B.4 了解实现细节。

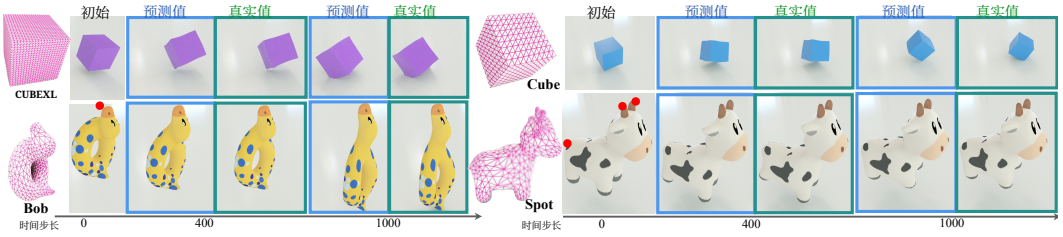


图 3：随机初始条件下 NMP 长时程仿真的可视化结果。每个数据集均展示了 静止网格以及关键帧处的仿真序列。 即使 推演中，我们的方法依然保持稳定，且视觉效 真实值（有限元法）相符。完整视频请访

可视化 如图3所示，尽管 NMP 的训练序列最多只有 200 步，但在延长的仿真过程中，它依然稳定且符合物理规律。这与整体式基线方法形成鲜明对比，后者通常在数百步后便会发散（附录 C.1）。这些结果表明，我们的设计能够有效支持稳定的外推。

未见初始条件的泛化 我们针对 BOB 和 CUBE 评估了两种未见过的初始条件——新颖的速度，以及速度与姿态偏移的组合 +（图4，左）。我们的模块化仿真器能够适应这两种情况，生成稳定、合理且长达 1，000 步的轨迹；尽管训练中没有此类示例，生成的轨迹仍能紧密贴合真实值。SPOT 和 CUBEXL 的结果见附录 C.3。

更高分辨率下的泛化 我们还通过在比训练中见过的更精细的离散化网格上进行评估，测试了网格无关泛化能力（10，$648 \to 21$，952 个顶点；图4，右）。尽管我们的模型仅在一种较粗糙的分辨率下训练，但逐顶点神经模块能够无缝适配更密集的网格，生成的运动依然稳定且符合物理规律。

4.2 与物理仿真器的比较

本节详细介绍 NMP 与物理模拟器之间的比较。具体而言，我们将*(i)* 传统物理模拟器和*(ii)* 混合神经网络-物理方法 NCLaw（Ma 等人，2023年）纳入基线方法；后者通过学习得到的材料模型扩展数值积分器。

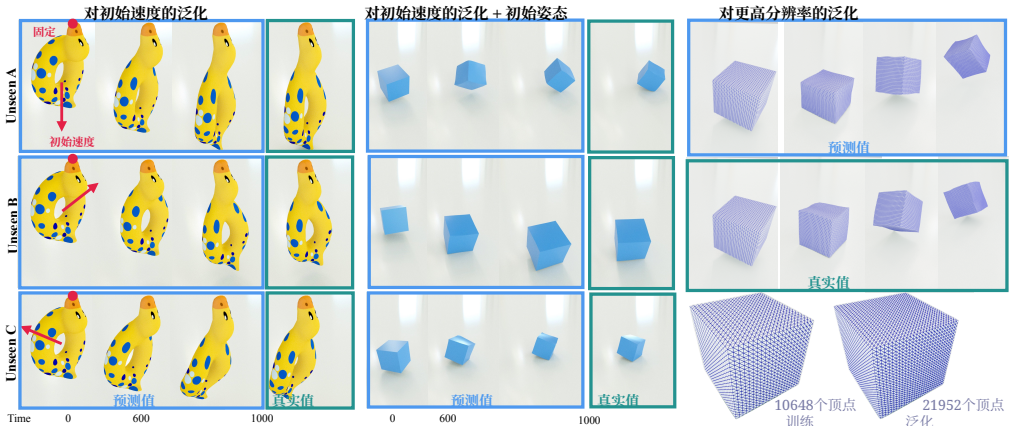


图4：NMP 在未见过的初始条件和更高分辨率下的仿真。从左到右，我们的方法与真实值：BOB 上未见过的初始速度、未见过的速度+姿态、CUBEXL 上的网格分辨率泛化（在 10k 个顶点上训练，在 21k 个顶点上测试）。视频请见网站。

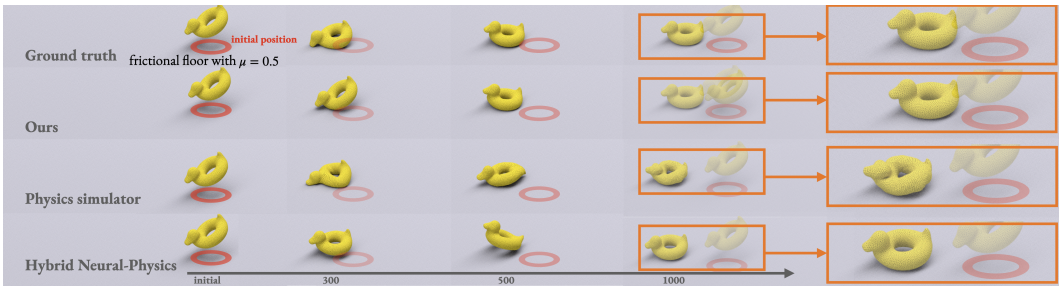


图5：在具有未知摩擦动力学的 FRICTION 环境中进行仿真对比。我们用红圈标出鸭子与表面接触的初始位置。

场景灵活性 我们在 FRICTION 环境中进行评估，其中弹性鸭子通过未知摩擦动力学与表面发生交互。每种方法都在未见过的 1,000 步推演上进行测试。请注意，若无法获取底层接触定律，纯 物理模拟器和混合 物理模拟器都无法捕捉未知摩擦效应。因此，我们通过系统辨识，为这两个基线方法估算物理模拟器输入参数的合理值。相比之下，我们的模型能够从数据中学习，因此无需进行系统辨识。

如图5所示，我们的方法再现了摩擦的作用：鸭子短暂滑动后便停下，而不是毫无阻力地滑行。这表明我们的模型成功学会了未知的摩擦相互作用。尽管 NCLaw 学会了材料响应，但 NCLaw 和纯物理模拟器都未能捕捉到底层接触动力学。这些结果表明，在物理信息不完整的情况下，我们的神经模拟器具有更强的灵活性。数值模拟器在信息完整的场景中表现良好，而这项基准测试凸显了神经方法处理未知动力学的独特优势。

效率比较 除了适应不同场景的灵活性外，神经模拟器的另一项优势是效率（Li等人，2021年；Wu 等人，2024b）。我们还测量了各种方法的运行效率。如表2所示，所有神经基线方法和我们的方法，效率都显著高于物理与神经物理混合方法。这一优势源于神经模拟器简化的推理范式：它只涉及神经网络的前向传播，而且张量运算已在现代设备上得到极大优化（Paszke等人，2019年）。具体而言，NMP采用了一种特殊的

表2：BOB 任务中不同方法的效率比较。记录了进行 1,000 步滚动推演的推理时间。

方法	时间（毫秒）
Transolver（2024b）	11796
EGNO（2024年）	26351
MeshGraphNet（2020年）	27603
NCLaw（神经物理）（2023年）	97121
物理模拟器	90275
NMP（我们的方法）	**2250**

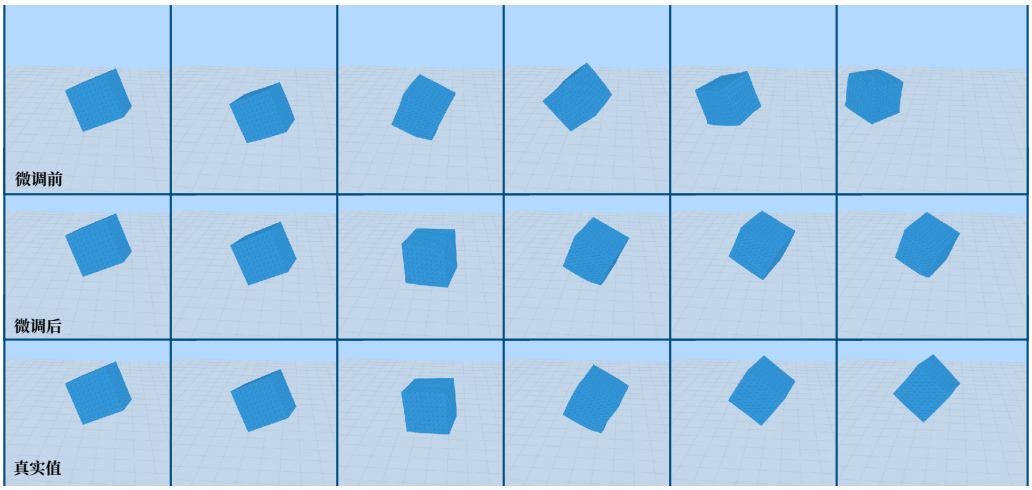


图 6：“从仿真到现实”迁移的概念验证实验。我们在 CUBE 数据集上预训练 NMP 模型，并在外力未知且材料更柔软的场景中对其进行微调。为了模拟现实世界的仿真任务，微调期间只能获取位置信息。

模块化架构，使我们只需引入轻量级神经模块，而无需使用笨重的单体模型。因此，NMP的速度约为传统物理模拟器的 40×倍，并且比先进的神经算子Transolver（Wu 等人，2024b）快 5× 倍以上；后者每一步都必须对完整而笨重的模型进行推理。

迁移能力 由于 NMP 采用两阶段训练策略，训练期间需要内力（公式 3）才能确保成功实现模块化。在仿真场景中可以获取这种中间物理信息，但在现实世界中无法记录。这里，我们展示了利用神经网络的迁移能力，NMP 也可以应用于现实世界的仿真。

这里我们考虑一个新场景：一个立方体在未知外力作用下下落并弹起，其材料未知（比 CUBE 数据集中的材料更柔软），且只能获取位置。由于这种部分可观测仿真无法提供两阶段训练所需的中间物理信息，我们采用深度学习中的预训练-微调范式，先在仿真的 CUBE 数据集上预训练 NMP，使模型实现良好的模块化，然后仅依据位置信息对其进行微调。图 6 显示，NMP 仍能准确生成仿真结果。这一结果不仅证明了 NMP 的实用性，也凸显了神经模拟器在迁移能力方面的优势。

4.3 模块化的独特优势

值得注意的是，要确保复杂的神经网络模块化网络按预期工作并非易事。若没有 NMP 中精心设计的物理对齐架构和专门化训练策略，我们就无法实现模块化的独特优势。为了直观地说明这一点，我们将两阶段方法与从头开始对两个网络进行联合训练的方法进行比较，见

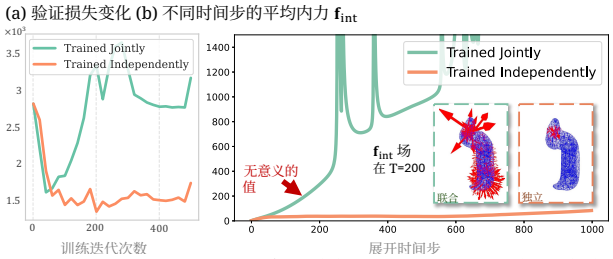
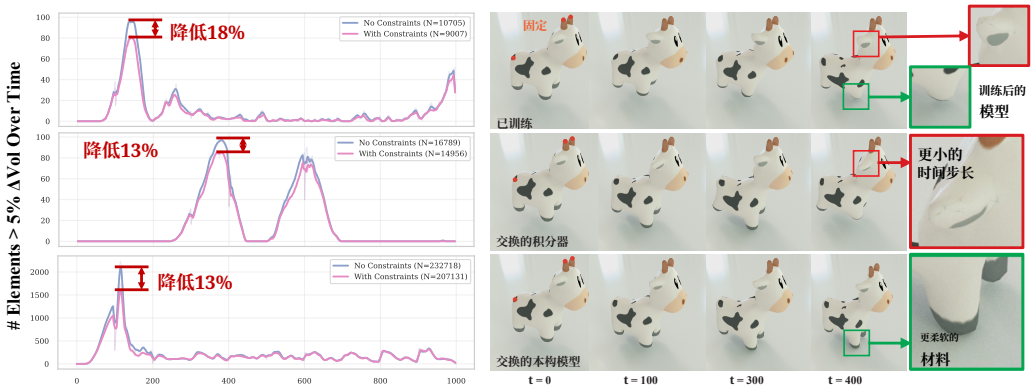


图 7：BOB 中模块化网络联合训练与我们的两阶段独立训练策略对比。

图 7。可以观察到，联合训练由两个模块化网络构成的 NMP，会使接口物理量$\mathbf{f}_{\text{int}}$ 失去物理意义，产生数值大得不可能的噪声力场。这种混乱的中间表示表明模块化网络未能按预期工作，即出现了坍塌问题（Mittal 等人，2022年）。相比之下，我们的专门化训练策略有助于模块化网络学习符合实际的物理量，并稳定训练过程。因此，经过适当专门化的模块化网络赋予模型有趣的特征。

直接物理约束 如前所述，我们的模块化设计与传统模拟器保持严格的物理对齐，从而能够无缝实现物理约束（公式 4）。这一特性是以往的单体式模拟器所无



(a) 直接物理约束的优势　　(b) 互换推理可视化

图 8：**(a)**物理约束对 CUBE 中三条轨迹的影响。曲线记录了每个时间步体积变化超过 5% 的四面体数量，图例则统计了 1,000 个时间步内违反约束的四面体总数。**(b) 互换推理。** 上：由我们训练好的基础模型生成的序列。中：将神经积分模块替换为时间步长小得多的半隐式积分器。下：将神经本构模块替换为更柔软的圣维南材料，从而增大变形，这一点可从脚部位置看出。

法实现的。为了进一步展示引入物理约束带来的益处，我们将完整方法与一个未使用体积保持损失进行训练的变体进行比较，并记录仿真过程中体积变化超过 5% 的四面体单元数量。如图 8(a) 所示，采用局部体积守恒进行训练的模型具有更好的物理合理性。具体而言，该损失还能将违规数量的最大值进一步降低 10%。尽管 NMP 的仿真仍不完美，但这种特殊的模块化设计为我们提供了一个可显式约束物理规则的接口。

互换推理 得益于我们对模块化接口的专门设计，NMP 能够与传统模拟器无缝实现互换推理。为验证互换的稳定性，我们首先将神经积分器替换为时间步长极小的半隐式数值积分器 (5×10^{-6})，其时间步长比训练期间小得多（图 8(b)，中）。这能够捕捉更高频的动力学，例如 Spot 的耳朵运动，表明我们学习到的本构模块仍与其他数值积分器兼容。其次，我们将神经本构模型替换为更柔软的圣维南材料（杨氏模量 5×10^4），从而产生更大的变形（图 8(b)，下）。这些实验凸显了我们的框架在不同积分器和材料之间即插即用的灵活性，使其无需重新训练即可快速适应新行为。

5 结论与讨论

本文提出了神经模块化物理（NMP），这是一个全面的模块化框架，将经典模拟器的物理可靠性与神经模拟器的场景灵活性相结合。在专用架构和两阶段训练策略的支持下，NMP 成功地将弹性仿真的整体计算流程转化为一系列神经模块。本文提出的模块化物理设计：（i）公开中间物理量；（ii）支持解析实现与学习得到的实现之间的即插即用式互换；（iii）实现物理约束的解析计算，而这些功能在以往的单体式神经模拟器中均无法实现。在实验中，与神经方法和混合方法相比，NMP 展现出更高的精度和更好的长时域稳定性，同时在不同初始条件和分辨率下保持良好的泛化能力，并且比传统模拟器表现出更好的场景灵活性和效率。

本文仅关注弹性仿真，这是一个已经十分广泛且基础的领域，应用涵盖图形学、机器人学、生物力学和材料科学。在未来，我们希望将模块化理念进一步扩展到更广泛的物理领域，例如流体仿真和多物体交互。传统的流体仿真同样具有原生的模块化结构，通常包含四个步骤：平流、力处理、压力投影和边界处理，这为神经模块设计提供了启发。至于多物体交互，过去的论文对此探索较少，因此需要神经或传统的碰撞模块来处理复杂的接触。

参考文献

Bob 救生圈鸭。https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/。访问日期：2025年1月28日。Yaser Afshar、Saakaar Bhatnagar、Shaowu Pan、Karthik Duraisamy 和 Shailendra Kaushik。使用卷积神经网络预测空气动力学流场。《计算力学》，64：525–545，2019年。Jacob Andreas、Marcus Rohrbach、Trevor Darrell 和 Dan Klein。神经模块网络。载于 *IEEE* 计算机视觉与模式识别会议论文集，第 39–48 页，2016年。Yohai Bar-Sinai、Stephan Hoyer、Jason Hickey 和 Michael P. Brenner。学习用于偏微分方程的数据驱动离散化。《美国国家科学院院刊》，116(31)：15344–15349，2019年7月。ISSN 1091-6490。doi：10.1073/pnas.1814058116。Keenan Crane、Ulrich Pinkall 和 Peter Schr¨oder。通过共形曲率流实现稳健的公平化。《*ACM*图形学汇刊》*(TOG)*，32(4)：1–10，2013年。

Salvatore Cuomo、Vincenzo Schiano Di Cola、Fabio Giampaolo、Gianluigi Rozza、Maziar Raissi 和 Francesco Piccialli。基于物理信息神经网络的科学机器学习：现状与未来方向。《科学计算杂志》，92(3):88，2022年。

Arka Daw、Jie Bu、Sifan Wang、Paris Perdikaris 和 Anuj Karpatne。使用保留-重采样-释放（r3）采样缓解物理信息神经网络中的传播失效。发表于国际机器学习大会，2023年。

Tao Du、Kui Wu、Pingchuan Ma、Sebastien Wah、Andrew Spielberg、Daniela Rus 和 Wojciech Matusik。DiffPD：可微投影动力学。《*ACM*图形学汇刊》*(TOG)*，41 (2):1–21，2021年。

Pascal Friederich、Florian H¨ase、Jonny Proppe 和 Al´an Aspuru-Guzik。面向下一代物质模拟的机器学习势函数。《自然·材料》，20(6):750–761，2021年。doi: 10.1038/ s41563-020-0777-6。综述文章，在线发表于2021年5月27日。

Moritz Geilinger、David Hahn、Jonas Zehnder、Moritz B¨acher、Bernhard Thomaszewski 和 Stelian Coros。ADD：面向具有摩擦接触的多体系统的解析可微动力学。《*ACM*图形学汇刊》（*TOG*），39(6):1–15，2020年。

Claudio Gentile 和 Manfred KK Warmuth。线性合页损失与平均间隔。《神经信息处理系统进展》，11，1998年。Xiaoxiao Guo、Wei Li 和 Francesco Iorio。用于稳态流近似的卷积神经网络。载于*ACM SIGKDD*知识发现与数据挖掘国际会议，第481–490页，2016年。

Yanan Guo、Xiaoqun Cao、Bainian Liu 和 Mei Gao。利用深度学习和物理约束求解偏微分方程。《应用科学》，10(17)，2020年。ISSN 2076-3417。doi: 10.3390/app10175917。

Oshri Halimi、Egor Larionov、Zohar Barzelay、Philipp Herholz 和 Tuur Stuyck。Physgraph：使用图神经网络进行基于物理的集成。*arXiv* 预印本 *arXiv:2301.11841*，2023年。Yuanming Hu、Jiancheng Liu、Andrew Spielberg、Joshua B Tenenbaum、William T Freeman、Jiajun Wu、Daniela Rus 和 Wojciech Matusik。Chainqueen：用于软体机器人的实时可微物理模拟器。载于*2019*年国际机器人与自动化会议（*ICRA*），第6265–6271页。IEEE，2019年。Devon Jarvis、Richard Klein、Benjamin Rosman 和 Andrew M Saxe。论神经模块的专门化。载于国际学习表征会议，2023年。Louis Kirsch、Julius Kunze 和 David Barber。模块化网络：学习分解神经计算。《神经信息处理系统进展》，31，2018年。

Nikola Kovachki、Zongyi Li、Burigede Liu、Kamyar Azizzadenesheli、Kaushik Bhattacharya、Andrew Stuart 和 Anima Anandkumar。神经算子：学习函数空间之间的映射及其在偏微分方程中的应用。《机器学习研究杂志》，24(89):1–97，2023年。

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Fer- ran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu 等。学习高水平的中期全球天气预报。《科学》，382(6677):1416–1421，2023年。

YifeiLi、Tao Du、Kui Wu、Jie Xu 和 Wojciech Matusik。Diffcloth：通过干摩擦接触实现的可微布料仿真。《*ACM*图形学汇刊》*(TOG)*，42(1):1–20，2022年。

Yifei Li、Hsiao-yu Chen、Egor Larionov、Nikolaos Sarafianos、Wojciech Matusik 和 Tuur Stuyck。DiffAvatar：借助可微仿真实现可用于仿真的服装优化。发表于*IEEE/CVF*计算机视觉与模式识别会议 *(CVPR)*论文集，洛斯阿拉米托斯，加利福尼亚州，美国，2024年6月a。IEEE计算机学会。doi: 10.1109/CVPR52733.2024. 00418。URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733. 2024.00418。

Yifei Li、Yuchen Sun、Pingchuan Ma、Eftychios Sifakis、Tao Du、Bo Zhu 和 Wojciech Matusik。Neuralfluid：基于可微仿真的神经流体系统设计与控制，2024b。

Zongyi Li、Nikola Borislavov Kovachki、Kamyar Azizzadenesheli、Burigede liu、Kaushik Bhattacharya、Andrew Stuart 和 Anima Anandkumar。用于参数化偏微分方程的傅里叶神经算子。载于国际学习表征会议，2021年。

Junbang Liang、Ming Lin 和 Vladlen Koltun。用于逆问题的可微布料仿真。载于 H. Wallach、H. Larochelle、A. Beygelzimer、F. d'Alch´e-Buc、E. Fox 和 R. Garnett（编），《神经信息处理系统进展》，第 32 卷。Curran Associates, Inc.，2019年。网址：https://proceedings.neurips.cc/paper/2019/file/28f0b864598a1291557bed248a998d4e-Paper.pdf。

Jinxian Liu、Ye Chen、Bingbing Ni、Wei Ren、Zhenbo Yu 和 Xiaoyang Huang。通过动态多尺度网格划分快速流体模拟。载于*AAAI*人工智能大会论文集，第37卷，第1675–1682页，2023年。

B-L Lu、K Ito、H Kita 和 Y Nishikawa。用于构造性学习的并行模块化多筛选神经网络架构。载于国际人工神经网络大会，第92–97页。IET，1995年。

Lu Lu、Pengzhan Jin、Guofei Pang、Zhongqiang Zhang 和 George Em Karniadakis。基于算子的通用逼近定理，通过DeepONet学习非线性算子。《自然·机器智能》，3：218–229，2019年。

Pingchuan Ma、Peter Yichen Chen、Bolei Deng、Joshua B Tenenbaum、Tao Du、Chuang Gan 和 Wojciech Matusik。从运动观测中学习神经本构定律，以实现可泛化的偏微分方程动力学。载于国际机器学习大会，第23279–23300页。PMLR，2023年。

Antoine McNamara、Adrien Treuille、Zoran Popovi´c 和 Jos Stam。使用伴随法进行流体控制。《*ACM*图形学汇刊》（*TOG*），23(3):449–456，2004年。

Sarthak Mittal、Yoshua Bengio 和 Guillaume Lajoie。模块化架构是否足够？《神经信息处理系统进展》，35:28747–28760，2022年。

Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga、Alban Desmaison、Andreas Kopf、Edward Yang、Zachary DeVito、Martin Raison、Alykhan Tejani、Sasank Chilamkurthy、Benoit Steiner、Lu Fang、Junjie Bai 和 Soumith Chintala。PyTorch：一种命令式风格的高性能深度学习库。载于 H. Wallach、H. Larochelle、A. Beygelzimer、F. d'Alch´e-Buc、E. Fox 和 R. Garnett（编），《神经信息处理系统进展》，第32卷，2019年。

Tobias Pfaff、MeireFortunato、Alvaro Sanchez-Gonzalez 和 PeterBattaglia。利用图网络学习基于网格的仿真。载于《国际学习表征会议》，2020年。

Tobias Pfaff、Meire Fortunato、Alvaro Sanchez-Gonzalez 和 Peter Battaglia。利用图网络学习基于网格的仿真。载于《国际学习表征会议》，2021年。Jovan Popovi´c、Steven Seitz 和 Michael Erdmann。用于控制刚体仿真的运动草图。《ACM图形学汇刊》，22(4):1034–1054，2003年10月。ISSN 0730-0301。Charles Ruizhongtai Qi、Li Yi、Hao Su 和 Leonidas J Guibas。Pointnet++：在度量空间中的点集上进行深度层次特征学习。《神经信息处理系统进展》，30，2017年。M. Raissi、P. Perdikaris 和 G.E. Karniadakis。物理信息神经网络：用于求解涉及非线性偏微分方程的正问题和反问题的深度学习框架。《计算物理学杂志》，378:686–707，2019年。ISSN 0021-9991。doi: https://doi.org/10.1016/j.jcp.2018.10.045。Alvaro Sanchez-Gonzalez、Nicolas Heess、Jost Tobias Springenberg、Josh Merel、Martin Riedmiller、Raia Hadsell 和 Peter Battaglia。将图网络用作可学习物理引擎，以进行推理和控制。载于《国际机器学习大会》，第4470–4479页。PMLR，2018年。Alvaro Sanchez-Gonzalez、Jonathan Godwin、Tobias Pfaff、Rex Ying、Jure Leskovec 和 Peter Battaglia。利用图网络学习仿真复杂物理。载于《国际机器学习大会》，第8459–8468页。PMLR，2020年。Connor Schenck 和 Dieter Fox。SPNets：面向深度神经网络的可微流体动力学。机器人学习会议（CoRL），2018年。Breannan Smith、Fernando De Goes 和 Theodore Kim。稳定的新胡克软组织仿真。《ACM图形学汇刊》，37(2):1–15，2018年。ˆPavel Sol´ın。《偏微分方程与有限元法》。约翰·威立父子出版公司，2005年。

Tuur Stuyck、Gene Wei-Chin Lin、Egor Larionov、Hsiao-yu Chen、Aljaz Bozic、Nikolaos Sarafianos 和 Doug Roble。Quaffure：实时准静态神经毛发仿真。载于计算机视觉与模式识别会议论文集，第 239–249 页，2025年。

Jonathan Tompson、Kristofer Schlachter、Pablo Sprechmann 和 Ken Perlin。利用卷积网络加速欧拉流体仿真。载于国际机器学习大会，第 3424–3433 页。PMLR，2017年。

Sifan Wang、YujunTeng 和 Paris Perdikaris。理解并缓解物理信息神经网络中的梯度流病态问题。《*SIAM* 科学计算杂志》，43(5)：A3055–A3081，2021年。

Sifan Wang、Shyam Sankaran、Hanwen Wang 和 Paris Perdikaris。物理信息神经网络训练专家指南。*arXiv* 预印本 *arXiv:2308.08468*，2023年。Sifan Wang、Shyam Sankaran 和 Paris Perdikaris。训练物理信息神经网络时遵循因果性。《计算机辅助应用力学与工程》，2024年。Yifan Wang 和 Linlin Zhong。NAS-PINN：神经架构搜索引导的物理信息神经网络，用于求解偏微分方程。《计算物理学杂志》，496:112603，2024年。Haixu Wu、Huakun Luo、Yuezhou Ma、Jianmin Wang 和 Mingsheng Long。RoPINN：区域优化的物理信息神经网络。《神经信息处理系统进展》，37:110494–110532，2024a。Haixu Wu、Huakun Luo、Haowen Wang、Jianmin Wang 和 Mingsheng Long。Transolver：用于一般几何上偏微分方程的快速 Transformer 求解器。发表于国际机器学习大会，第53681–53705页。PMLR，2024b。Jie Xu、Tao Chen、Lara Zlokapa、Michael Foshey、Wojciech Matusik、Shinjiro Sueda 和 Pulkit Agrawal。端到端可微的接触感知机器人设计框架。发表于机器人学：科学与系统论文集，线上，2021年7月。doi：10.15607/RSS.2021.XVII. 008。

Minkai Xu、Jiaqi Han、Aaron Lou、Jean Kossaifi、Arvind Ramanathan、Kamyar Azizzadenesheli、Jure Leskovec、Stefano Ermon 和 Anima Anandkumar。用于建模三维动力学的等变图神经算子。发表于国际机器学习大会，第55015–55032页。PMLR，2024年。

Yuan Yin、Vincent Le Guen、J´er´emie Dona、Emmanuel de B´ezenac、Ibrahim Ayed、Nicolas Thome 和 Patrick Gallinari。利用深度网络增强物理模型，实现复杂动力学预测。《统计力学杂志：理论与实验》，2021年(12):124012，2021年12月。ISSN 1742-5468。doi: 10.1088/1742-5468/ac3ae5。

A 补充网站

匿名补充网站：https://sites.google.com/view/neural-modular- physics。该网站包含视频，展示我们的方法在所有测试场景中的表现，并可视化不同初始条件和分辨率下的预测动力学。

B 实现细节

在本节中，我们将详细介绍正文中实验的更多细节。

B.1 真实值模拟器

我们使用 PyTorch 实现了真实值模拟器，以实现端到端可微性。该模拟器采用半隐式积分方案:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + dt(\mathbf{M}^{-1}\mathbf{f} + \mathbf{g}) \tag{5}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + dt\mathbf{v}_{t+1}, \tag{6}$$

其中， $dt = 5 \times 10^{-4}$ 秒表示时间步长，$\mathbf{M}^{-1}$ 表示逆质量矩阵，$\mathbf{g}$ 表示重力加速度，在我们的实验中将其设置为 $-9.8m/s^2$ 。
积分后，模拟器会处理外部约束。对于涉及地面接触的场景（例如 CUBE 场景），低于地平面的顶点会被投影回地面，其垂直速度设为零。对于具有固定边界条件的场景（BOB 和 SPOT），指定的顶点在整个仿真过程中都保持在其静止位置，且速度为零。

B.2 数据集生成

时间步长设置对于前四种环境，为确保稳定性，我们使用较小的内部时间步长 $dt = 10^{-5}$ 秒生成真实值轨迹（每个输出帧包含 50 个子步）。每条轨迹包含 1000 帧，分辨率较低，时间步长为 $dt = 5 \times 10^{-4}$，相当于每 50 个模拟步输出一帧。值得注意的是，我们的模型仅在这些较低分辨率的帧上进行训练和评估，这意味着神经积分器以较大的有效时间步长 $dt = 0.0005$ 运行。对于 FRICTION 环境，我们以 $dt = 0.0001$ 生成并测试模拟器，以测试更大的时间步长。

上述设置本身就要求模型预测较长时间间隔内的动力学，从而鼓励时间泛化。训练时使用的滚动推演最多为 200 步，但正文中报告的所有结果，包括 1000 步预测，都要求模型泛化到远超其训练时域的范围。

Ma材料构型 对于前四种环境的数据集生成，我们使用了 Neo-HooHookean 材料模型:

$$W_{NH}(\mathbf{F}) = \frac{\mu}{2}(I_C - 3) - \mu\ln(J) + \frac{\lambda}{2}(\ln(J))^2, \tag{7}$$

其中 μ 和 λ 是 Lam´e 参数， $J = \det(\mathbf{F})$ 是体积变化比，而 $I_C = \mathrm{tr}(\mathbf{F}^T\mathbf{F})$ 是右柯西－格林变形张量的第一不变量。
各场景的材料参数配置如下。我们为不同场景配置了不同的材料刚度，同时保持泊松比 $\nu = 0.45$不变。CUBE 场景使用较硬的材料，杨氏模量为 $E = 5 \times 10^5$ Pa；而 Bob 和 Spot 场景均使用较柔顺的材料，杨氏模量为 $E = 1 \times 10^5$ Pa，以允许更大的变形。

对于 FRICTION 环境，我们使用了可微模拟器 DiffPD（Du 等，2021）。它基于投影动力学，因此需要采用投影动力学形式来表示材料能量。有关能量离散化和求解器的完整细节，请参阅原论文。为与其他环境保持一致，我们以相同方式设置材料参数，在保持泊松比 $\nu = 0.45$ 不变的同时改变杨氏模量 E ，以控制材料刚度。

B.3 网络架构与训练细节

模型架构 神经本构模块采用两层多层感知机实现，分别包含 80 个和 96 个神经元。神经积分模块根据位置 $\mathbf{x}$、速度 $\mathbf{v}$ 和内力 $\mathbf{f}_{\text{int}}$ 预测逐顶点速度更新$\Delta\mathbf{v}$ 。三个输入分别由一个采用 SiLU 激活函数的单层多层感知机独立处理，生成维度均为 32 的嵌入。此外，我们使用学习式对齐模块（TNET）生成对齐后的输入网格坐标：它通过四元数回归器输出一个 3×3 旋转矩阵，将每个网格对齐到规范坐标系。这有助于实现姿态不变性，并提升模型对不同初始条件的泛化能力。将对齐后的网格位置、原始网格位置、速度和力与原始输入拼接后，输入一个包含 64 个隐藏单元的三层多层感知机，以回归 $\Delta\mathbf{v}$。为使训练初期更加稳定，我们参照 MeshGraphNet（Pfaff 等人，2020）的做法，将预测速度的大小限制在最高 30 m/s。

学习式对齐模块 为了提升模型在不同初始姿态下的泛化能力，我们采用学习式空间对齐模块，通过为每个样本预测一个全局 3×3 线性变换矩阵来规范化输入位置。输入点云$\mathbf{x} \in \mathbb{R}^{N\times3}$ 首先由一个共享的 MLPNET 编码器进行编码，该编码器独立地应用于每个顶点。随后，通过最大池化聚合得到的逐点特征，再将其输入两层多层感知机，以回归一个 9 维向量，并将其重塑为一个 3×3 矩阵。为确保稳定性，预测矩阵的初始化值设为接近单位矩阵。与先前将输出限制为旋转（例如通过四元数或投影至 SO(3)）的方法不同，我们允许使用完整的线性变换，使网络能够直接从数据中学习缩放、剪切及其他规范化变形。

训练配置 两个网络均使用 Adam 优化器进行训练，并采用余弦退火学习率调度。训练过程分为两个阶段：首先分别对每个网络进行 100 个训练轮次的独立预训练，随后再进行 100 个训练轮次的联合微调。在预训练期间，每个网络使用来自仿真数据集的真实值输入进行训练；微调则使网络能够适应彼此学到的行为。
为稳定序列训练并防止误差累积，我们在预训练和微调期间均采用教师强制策略。在每个训练轮次处理序列时，会定期将模拟状态重置为真实值。重置间隔按照余弦退火计划从 60 步逐渐增加至 200 步，从而逐步促使网络在不受干预的情况下处理更长的预测序列。

B.4 基线实现

我们实现了三种具有代表性的神经模拟器：Transolver（Wu 等人，2024b）、EGNO（Xu 等人，2024）和 MeshGraphNet（MGN）（Pfaff 等人，2020）。为确保公平性和可复现性，所有模型均基于其官方代码库或库构建，并在适用时遵循推荐的预处理方法和数据流程。所有基线方法均采用归一化均方误差（NMSE）损失 $\text{NMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\|\hat{\mathbf{y}}-\mathbf{y}\|_2^2}{\|\mathbf{y}\|_2^2}$进行优化，学习率固定为 0.1，训练 100 个训练轮次。具体构型如下：

- **Transolver**（Wu 等人，2024b）：我们采用 Transolver 非规则网格变体，并使用 15 个 Transolver 块，模型宽度为 256、4 个注意力头、多层感知机比例为 2，随机失活率为 0.1；同时将切片数量设为 32，参考标记数量设为 16。
- **EGNO**（Xu 等人，2024年）：使用 15 层、隐藏维度为 256、4 个向量消息头、随机失活率为 0.1，以及维度为 12 的径向基函数嵌入。
- **MeshGraphNet**（Pfaff 等人，2020年）：进行 15 步消息传递，隐藏维度为 128。

为提升鲁棒性并稳定归一化统计量，我们在训练期间向所有 三种神经模拟器（Transolver、EGNO 和 MeshGraphNet）的输入中注入少量高斯噪声。具体而言，在每一步中，我们对位置和速度进行扰动，设为 $x_t \leftarrow x_t + \varepsilon_x, v_t \leftarrow v_t + \varepsilon_v$ with$\varepsilon_x,\ \varepsilon_v \sim \mathcal{N}(0,\ \sigma^2)$，其中我们使用 $\sigma = 3 \times 10^{-3}$。

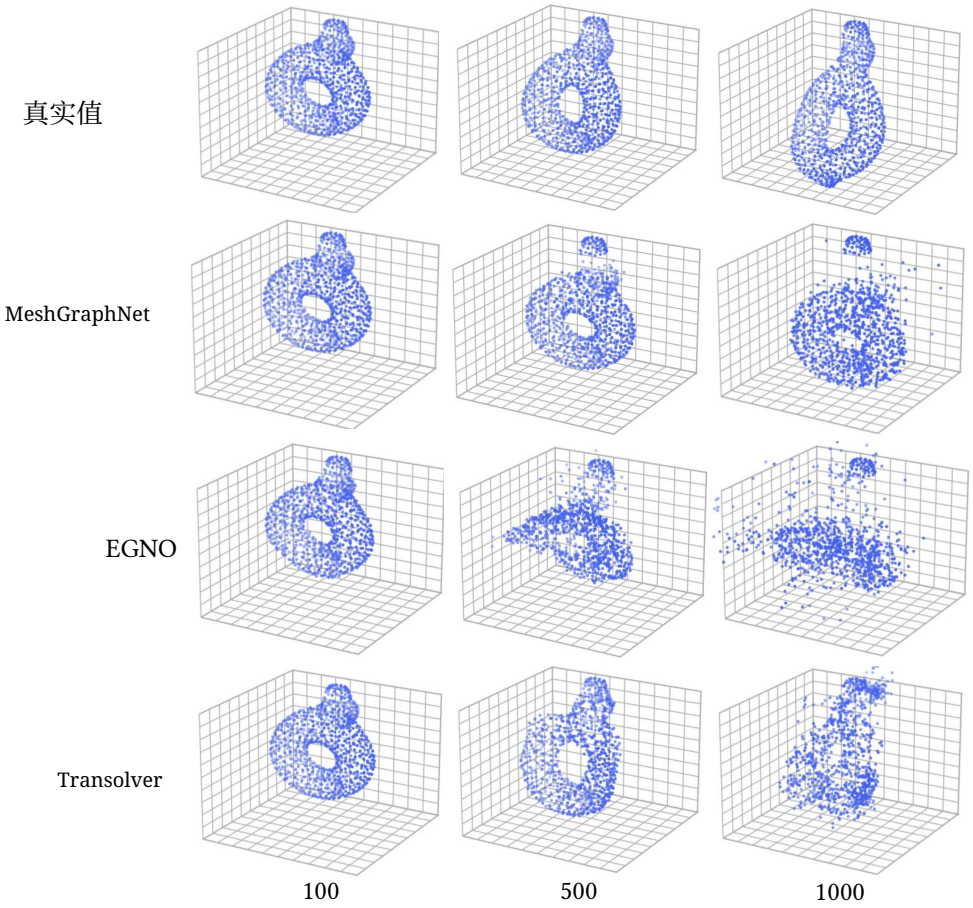


图 9：BOB 上有限元真实值与三种基线方法在第 100、500 和 1000 步的推演定性比较。从上到下，各行分别显示真实值、MeshGraphNet、EGNO 和 Transolver。各列显示第 100、500 和 1000 步的快照帧。

对于混合物理模拟器，我们直接遵循 NCLaw（Ma 等人，2023年）的模型架构配置所采用的官方配置。对于物理模拟器基线，我们使用第 B.1 节所述的真实值模拟器。

B.5 互换推理实验设置

在材料模型互换实验中，我们使用了圣维南-基尔霍夫模型：

$$W_{StVK}(\mathbf{F}) = \frac{\lambda}{2}\left(\mathrm{tr}(\mathbf{E})\right)^2 + \mu \mathrm{tr}(\mathbf{E}^2), \tag{8}$$

其中 $\mathbf{E} = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{I})$ 是格林应变张量。

C 补充结果

本节将提供更多可视化内容和结果，作为正文的补充。

C.1 与基线方法的比较

视觉比较如图9所示，我们将 Transolver（Wu 等人，2024b）、EGNO（Xu 等人，2024）和 MeshGraphNet（MGN）（Pfaff 等人，2020）在第 100、500 和 1000 步时的结果与真实值进行了比较。可视化结果揭示了各基线方法性能中的几种不同模式。

在这三种神经模拟器中，Transolver 在较短时间范围内最能保持几何形状。在第 100 步时，其模拟形状和姿态仍与参考结果接近；到第 500 步时，结果依然连贯，沿着轨迹演变，仅出现中等程度的漂移。然而，到第 1000 步时，Transolver 已无法保留局部结构，只剩下粗略的轮廓，反映出累积误差以及弹性记忆的丧失。
EGNO 退化得早得多。到 500 步时，构型已下陷成模糊的粒子云，并出现明显的向下偏移；到 1000 步时，物体已基本变得不成形，原始几何形状也难以辨认。

MeshGraphNet 比 EGNO 更长时间地保持了可辨认的轮廓，但其运动受到的阻尼过强。振荡迅速消退，接触响应变得模糊，粒子呈现出非自然的下沉或凝聚趋势，而非弹性回弹。到 1000 步时，几何形状沿垂直方向拉伸，整体连贯性也遭到破坏。

这些结果表明，单体式神经模拟器难以准确捕捉长时间范围内的物理动力学。相比之下，如图3所示，我们提出的神经模块化物理框架能够在更长的时间范围内生成稳定且符合物理规律的仿真。

新的自回归预测 在神经模拟器基线方法中，其官方配置采用单步自回归预测。因此，在实验中，我们的方法和基线方法也都遵循这一设置。增强这些基线方法的一种可能方式是让模型一次预测100步；这样，1000步仿真只需进行10次滚动预测，这是一种切实可行的减少误差累积的技巧。

表3：与增强基线方法进行比较，其中每个模型在一次推理中直接预测 100 步。记录 1,000 步的均方根误差。

方法（BOB模拟）	RMSE
Transolver（2024b）（预测100步）	0.75
EGNO（2024年）（预测100步）	2.00
MeshGraphNet（2020年）（预测100步）	2.61
NMP（我们的方法，预测单步）	**0.58**

这里，我们还与增强基线方法进行了比较。如表 3 所示，直接预测 100 步可以降低神经模拟器的长期模拟误差，尤其是 Transolver，其均方根误差从 1.782（表 1）降至 0.75。然而，这些基线方法仍逊于我们的方法，而我们的方法
彰显了模块化的优势。

C.2 初始条件分布示意图

如正文所述，不同的仿真轨迹始于不同的初始条件，这自然会导致动力学轨迹出现分歧。为直观展示这一效应，我们绘制了 CUBE 数据集中 40 条轨迹，其中包括训练集中的 32 个案例和测试集中的 8 个案例。这些案例的差异源于不同的初始速度和立方体朝向。如图 10 所示，即使第一步的初始速度或朝向只有细微差异，也可能显著改变后续动力学，从而使训练轨迹和测试轨迹之间产生明显差距。这一可视化结果凸显了我们实验中泛化所面临的固有挑战。

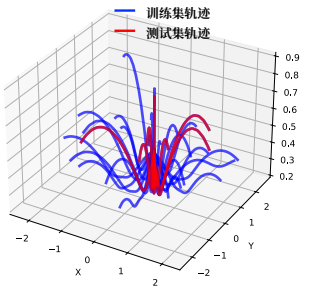


图 10：CUBE 上不同初始条件下轨迹的可视化。

C.3 对未见初始条件的泛化

图 11 展示了我们的方法在不同初始条件下对 Spot 场景的泛化能力。与正文讨论的 Bob 场景类似，我们在多个未见过的初始速度下评估了模型。结果表明，模型在不同场景中表现稳定——每一行都展示了一条独特的轨迹，模型在准确保持头部和尾部的定点边界条件的同时，维持了稳定的运动。在 $t = 1000$，我们的预测结果（蓝框内）与真实配置（绿框内）高度吻合，表明尽管双定点约束颇具挑战性，模型仍具备稳健的泛化能力。

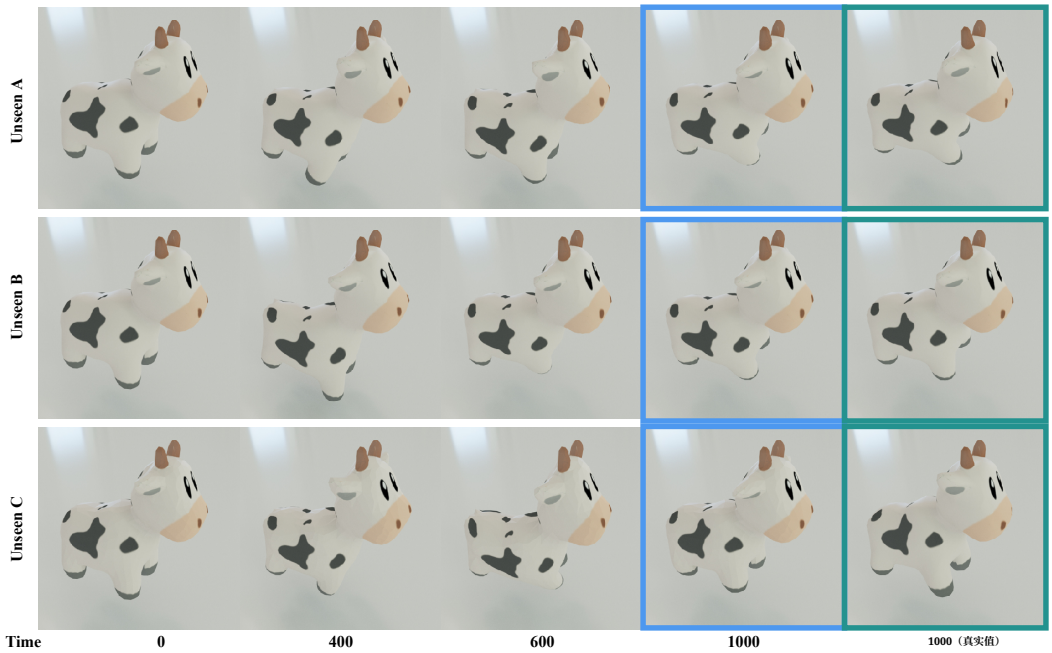


图11：对 SPOT 的未见初始条件的泛化。每一行展示了具有不同初始速度的一组序列，证明了我们的模型能够处理多样的动力学，同时保持不动点约束。我们的预测与在 t=1000 时的真实值非常接近。

C.4 表 1 的标准差

表4 报告了与表 1 所列平均性能相对应的逐序列标准差（基于 8 条验证轨迹）。

表4：表 1 中均方根误差的标准差（基于 8 条验证轨迹计算）

方法	CUBE			CUBE XL			SPOT			BOB		
	100	500	1000	100	500	1000	100	500	1000	100	500	1000
Transolver	±0.069	±0.364	±0.706	±0.073	±0.726	±2.323	±0.062	±0.343	±0.673	±0.043	±0.230	±0.563
EGNO	±0.197	±0.979	±2.001	±0.103	±0.815	±1.533	±0.098	±0.333	±0.273	±0.110	±0.524	±1.038
MeshGraphNet	±0.002	±0.009	±0.002	±0.010	±0.034	±0.631	±0.001	±0.003	±0.001	±0.001	±0.001	±0.002
Ours	±0.010	±0.046	±0.161	±0.010	±0.149	±0.107	±0.001	±0.135	±0.162	±0.021	±0.136	±0.348

C.5 物理验证指标

除了均方根误差，仿真器的物理合理性也是一项重要指标。因此，作为图 8 的补充，我们还统计了在 1,000 步滚动推演中体积变化超过 5% 的四面体数量，以衡量不同仿真器的局部体积保持能力。如表 6 所示，在这项物理验证指标上，NMP 优于所有其他基线方法。这些结果进一步证明了神经模块化物理在确保物理合理性方面的优势。

表 5：不同模型的物理验证结果，其中记录了体积变化超过 5% 的四面体数量。数量越少，表示物理合理性越好。

方法（CUBE 仿真）	数量
Transolver（2024b）	1271827
EGNO（2024年）	1595317
MeshGraphNet（2020年）	479566
NMP（我们的方法）	**231094**

D 消融研究

这里，我们对神经模块的超参数和构型进行了一些消融研究。

模型复杂度分析 我们通过消融神经本构模型的规模，评估其对网络复杂度的敏感性。我们选择原始架构（2 个隐藏层，分别包含 96 和 80 个神经元），以最大限度地利用训练期间的 GPU 显存。将架构缩减为由 64 和 40 个神经元组成的 2 层网络，并在使用真实值积分器的 CUBE 数据集上进行评估后，我们发现其 1000 步均方根误差（0.337±0.270）高于原始模型（0.220±0.174），凸显了模型规模的重要性。进一步扩大 NMP 模型是一个很有前景的方向。

超参数分析 如公式 4 所示，加入物理约束也会引入一个权重超参数。在实验中，我们选择了一个体积正则化常数，使约束项在训练期间约占总损失的 1%。这一选择体现了鼓励物理合理性与保持可学习性之间的平衡。在此，我们评估了模型在一系列体积损失权重$\{0, 1, 10, 1000\}$下的性能。如右图所示，适度的正则化（例如 1.0）可以提高长时域精度，而过大的值（例如 1000）则会降低性能，因为过大的权重会阻碍网格实现预期的变形。

表 6：物理约束损失权重分析。以下测试在图 6 所示的 “ 真实世界 ”CUBE 场景中进行。

损失权重	1000 步均方根误差
0	0.397
1	0.337
10	0.604
1000	2.579

E 失败模式分析

为了进一步展示我们模型的仿真特性，我们从 BOB 任务中选取了最差情况（均方根误差最大）。图 12 中给出了视觉对比。可以观察到，即使在最差情况下，NMP 仍能准确仿真 Bob 的主要动力学特征。此外，我们还注意到，大变形动力学确实给仿真带来了很大挑战。

F 局限性

本文提出了一种新框架，将弹性仿真模块化为经过精心设计和优化的神经网络，从而在场景灵活性和物理约束方面带来了若干独特优势。尽管神经模块化物理旨在成为一种通用理念，但正如本文标题所示，我们目前的实验仅涵盖弹性仿真。将该框架应用于其他物理领域（例如流体动力学）将是一个有前景的研究方向。然而，流体仿真需要一种不同的模块化架构，以适配传统计算流体力学的流程。因此，我们将当前研究范围限定于弹性仿真，并计划将 NMP 在其他领域的探索留作未来工作。

大语言模型使用声明：大语言模型仅用于协助写作和润色语言 .

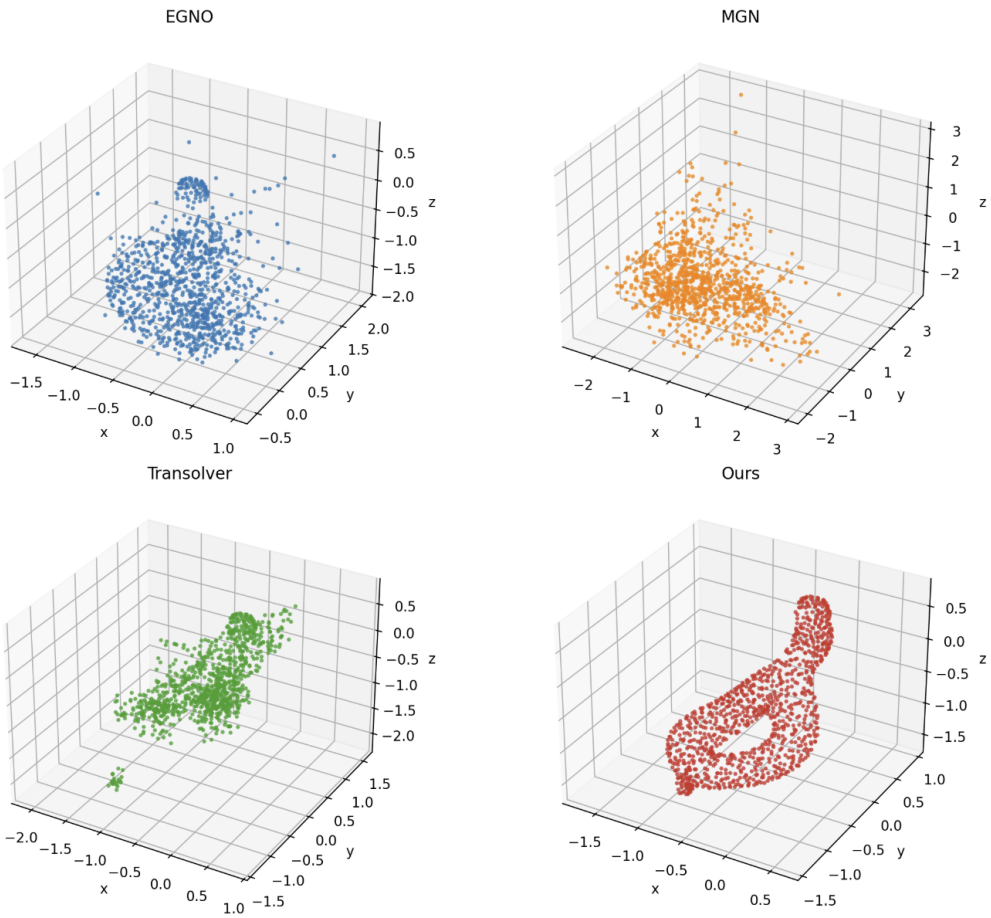


图 12：失效模式分析。我们将最差情况下的最后一帧进行可视化，该情况根据 NMP 的均方根误差选定。同时也展示了其他基线方法的预测结果。