arXiv:2303.12971v1 [physics.comp-ph] 23 Mar 2023

用于流固耦合的可微混合神经建模

XiantaoFan[a,b,c], Jian-xun Wang[a,b,c,d,*]

[a]航空航天与机械工程系，圣母大学，印第安纳州圣母城[b]环境变化倡议，圣母大学，印第安纳州圣母城[c]可持续能源中心（*ND Energy*），圣母大学，印第安纳州圣母城[d]*Lucy Family* 数据与社会研究所，圣母大学，印第安纳州圣母城

摘要

求解由非线性偏微分方程描述的复杂流固耦合问题，对各种科学与工程应用至关重要。传统的计算流体力学求解器难以满足日益增长的大规模、长周期模拟需求。数据的持续增长以及深度学习的快速发展，为通过数据赋能建模应对这些挑战开辟了新途径。借助可微编程框架，将深度学习与经典数值方法无缝结合，能够显著提升数据驱动建模的性能。本研究提出一种可微混合神经建模框架，用于高效模拟流固耦合问题。在该框架中，基于浸入边界法离散化的流固耦合物理规律，通过可微编程与序列神经网络无缝集成。所有模块均使用 JAX 编程，自动微分可在整个模型滚动预测轨迹上实现梯度反向传播，从而使混合神经流固耦合模型能够以端到端、序列到序列的方式进行整体训练。通过多个流固耦合基准案例，我们展示了该方法在模拟刚体和柔性体流固耦合动力学方面的优势与能力。与基准纯数据驱动神经模型、弱耦合混合神经模型以及纯数值流固耦合求解器相比，所提出的模型在准确性、鲁棒性和泛化能力方面均表现出优势。

*通讯作者。电话： +1 540 3156512电子邮箱：
jwang33@nd.edu（Jian-xun Wang）

提交至 *Elsevier* 的预印本 *2023*年*3*月*24*日

关键词：流固相互作用、可微编程、科学机器学习、深度神经网络、计算流体力学加速

1. 引言

流固耦合（FSI）广泛存在于各种科学与工程应用中，涉及从小尺度生物系统到土木或水利工程中的大型工业结构。由于结构和流体本身具有非线性，且结构、流体及其交互涉及广泛的时空尺度，双向耦合的流固耦合是一个复杂的多物理场问题 [1, 2]。计算建模对于预测和控制复杂的流固耦合动力学至关重要。然而，基于经典数值技术的传统计算模型，在以足够的准确性和较低的计算成本高效模拟流固耦合问题方面面临巨大挑战。尽管已有许多计算流体力学（CFD）技术得到发展，例如任意拉格朗日–欧拉 [3, 4], 水平集方法 [5], 和浸入边界法 [6–8], 用于模拟流体与固体动力学之间的交互并提供诸多物理见解，但求解一系列流固耦合控制常微分方程/偏微分方程（ODE/PDE）仍然成本高昂，因此难以应对实时预测或多次查询应用，例如设计优化、推理和不确定性量化。

机器学习的快速发展以及日益丰富的数据为应对这些挑战提供了新的机遇。近年来，利用深度神经网络（DNNs）对流体或流固耦合系统的复杂非线性动力学进行建模，已成为广受关注的研究方向。通过数据驱动技术（如本征正交分解（POD）和卷积自编码器方法），直接在高维空间或降阶空间中学习输入输出关系，研究人员已针对多种场景识别出映射函数，包括绕圆柱流动 [9, 10], 翼型空气动力学 [11, 12], 生物流动 [13, 14], 以及雷诺平均纳维–斯托克斯/大涡模拟闭合问题 [15–18], 等。为处理包含非结构化数据的不规则域，研究人员采用图神经网络（GNNs）构建用于流体和流固耦合动力学的快速神经模拟器 [19, 20]。例如，将

图神经网络和本征正交分解相结合，用于学习一个基准流固耦合案例中的流体和固体动力学，即弹性支撑圆柱的涡激振动 [21]。

尽管数据驱动模型在流固耦合系统建模方面取得了令人鼓舞的成果，但仍存在一些局限，包括数据采集成本高，以及泛化能力或鲁棒性可能不足。因此，重要的是平衡数据驱动模型与物理先验的运用，构建兼具两者优势的混合模型。将数据驱动技术与物理先验知识（如控制方程和约束）相结合的混合深度学习模型，能够有效提升样本效率和泛化能力，尤其是在处理小规模或稀疏数据时。对于流固耦合建模，这种混合可通过两种方式实现：（1）将深度神经网络与基于投影的模型降阶方法以及流固耦合约束相结合；（2）将控制方程纳入深度神经网络的损失函数或神经网络架构中。后一种方法也称为物理信息机器学习 [22]。

现有的大多数流固耦合问题数据驱动模型都属于第一类。通常，利用本征正交分解或卷积自编码器等技术，将高维的流体和结构动力学降维到低维潜在空间中，再由独立的深度神经网络学习潜在的流体和结构动力学 [23]。物理界面约束，例如移动界面（固体到流体的耦合）和流体力（流体到固体的耦合），通常用于耦合流体和结构深度神经网络，这些约束可以通过水平集函数 [23–25], 浸入边界法（IBM）掩码 [26], 或直接强制项 [27]等方法表示。通过这种方式，可以以一致的方式学习结构响应和流体动力学。在某些情况下，如果只关注结构响应，则可以使用更精确的数值求解器来求解固体动力学，而由深度神经网络学习流场或仅学习流体力 [28]。尽管这种方法看起来可能类似于稍后将介绍的属于第二类的混合神经模型，但这里深度神经网络的训练通常在线下进行，与数值求解器和物理约束解耦。这些松耦合混合方法通常会面临显著的稳定性

问题 [29], ，尤其是在预测长时间跨度的时空动力学时 [30]。

本文考虑的第二类混合数据驱动方法称为物理信息机器学习（PIML）。这一类别的一项重要贡献是物理信息神经网络（PINNs），它将控制方程、守恒定律和边界条件等物理先验纳入标准损失函数，以规范神经网络训练 [31]。PINNs 已在多种问题中展现出其能力，包括流体流动的正向和逆向建模 [32, 33], 固体力学 [34, 35], 以及流固耦合问题 [36, 37]。然而，PINNs 通过自动微分（AD）或数值离散重构偏微分方程残差，以软约束的方式施加物理约束，使损失景观极难优化 [38]。其成功与失效模式在很大程度上取决于用于加权各项损失的超参数设置。

近期，物理信息机器学习（PIML）的一个新方向是将已知物理规律融入神经网络架构 [39]。这一方向受到深度神经网络架构与微分方程之间关系的启发 [40, 41]。Liu 等人 [39] 提出了一种偏微分方程保持神经网络（PPNN），通过多分辨率设置下的卷积残差连接网络（卷积残差网络）块，将部分已知控制物理规律中的离散偏微分方程算子集成到深度神经网络架构中。与纯黑箱卷积残差网络和其他最先进的神经算子相比，研究已证明，PPNN 在预测各种时空物理现象时，在准确性、速度和泛化能力方面表现得好得多。从数值建模的角度看，PPNN 中保持偏微分方程的部分可解释为以卷积残差网络块实现的数值微分和时间推进，它们可以无缝集成到任何黑箱神经网络结构中，形成混合神经求解器。得益于可微编程 [42], ，梯度可以通过整个神经求解器进行反向传播，从而能够对整个模型进行高效的端到端序列到序列训练。最近，沿着可微编程的思路，利用基于经典数值技术的离散形式物理规律来构建可微混合求解器，正受到越来越多的关注 [30, 43–46]。可微编程框架已

近期已应用于湍流参数化学习 [44, 47, 48], ，其中基于后验准则学习亚网格闭合或其他相关关系。端到端可微学习显著提升了后验预测结果，相较之下，松散耦合的混合深度神经网络闭合难以解决上文提到的稳定性问题 [29]。由于缺乏自动微分能力和 GPU 兼容性，将深度神经网络集成以构建混合神经求解器，对大多数传统遗留求解器而言是一项重大挑战。为解决这一问题，越来越多的研究致力于开发可微计算流体力学求解器，例如 PhiFlow [49], Jax-CFD [43], 和 Jax-Fluids [50]。尽管混合可微神经建模范式已展现出巨大潜力，但该领域仍处于早期阶段，尚需进一步发展，尤其是在双向耦合流固相互作用动力学等复杂多物理场问题的建模方面。

本文提出了一种用于预测流固耦合动力学的可微混合神经模型。该模型将控制偏微分方程融入深度学习架构，在兼顾预测准确性与效率的同时，加速流固耦合模拟。所提出的混合神经模型基于 JAX 的可微编程框架构建；JAX 是一种面向高性能机器学习研究和可微建模的 Python 库。具体而言，我们构建了一个流固耦合混合循环网络单元，该单元以极粗网格上的可微不可压缩纳维-斯托克斯求解器为基础，并结合可训练的卷积残差网络模块，以确保流体预测准确。与此同时，通过超分辨率操作在高分辨率网格上直接求解固体动力学，并利用浸入边界法直接强制将其与流体耦合，以实现双向干扰 [51]。随后，将这些混合流固相互作用网络单元组合起来，构成基于长短期记忆网络（LSTM）的序列循环网络，从而实现序列到序列（Seq2Seq）训练与预测。通过对两个基准案例进行大量实验，并将其准确性、鲁棒性、泛化能力和效率与其他最先进的基线方法进行比较，证明了所提出的混合神经流固耦合模型具有出色的性能。据作者所知，本研究首次尝试在可微

编程范式下，将深度学习与数值求解器相结合，用于预测性流固相互作用建模。本文组织如下：第2节概述总体方法论，包括问题表述和模型设计。第3 节展示神经流固耦合模型在两个不同基准案例上的结果，并与基线方法进行比较。第4节讨论所提出的混合神经模型的鲁棒性、泛化能力和加速效果，最后在第5节总结全文。

2. 方法论

2.1. 问题表述

流固耦合问题可以用一系列具有时空非线性的耦合偏微分方程来描述。本文重点关注不可压缩流体与固体动力学之间的交互，以展示所提出的可微神经流固耦合模拟方法的基本概念。具体而言，流体动力学由不可压缩纳维-斯托克斯方程控制，

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, && \mathbf{x}, t \in \Omega_f \times [0, T] \\ \frac{\partial \mathbf{u}}{\partial t} &= -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{f}, && \mathbf{x}, t \in \Omega_f \times [0, T] \end{aligned} \tag{1}$$

其中 t 和$\mathbf{x}$ 分别表示时间和欧拉空间坐标；流体速度$\mathbf{u}(t,\mathbf{x})$和压力 $p(t,\mathbf{x})$均为定义在 $\Omega_f \subset \mathbb{R}^2$中的时空函数； ρ 和 ν 分别表示流体密度和流体的运动黏度；速度和压力的解由给定的初始条件和边界条件（IC/BCs）唯一确定。

流体动力学与可变形结构之间的交互由浸入边界法（IBM）描述，其中固体动力学的影响通过直接强制项$\mathbf{f}$ [6], 施加于流体，以满足界面$\Gamma \in \Omega_f \cap \Omega_s$处的无滑移壁面条件。结构动力学可由以下拉格朗日坐标系下的方程描述，

$$\mu_s \frac{\partial^2 \mathbf{w}}{\partial t^2} + EI \frac{\partial^4 \mathbf{w}}{\partial \mathbf{X}^4} = \mathbf{q}, \qquad \mathbf{w}, t \in \Omega_s \times [0, T] \tag{2}$$

其中，$\mathbf{X}$ 为拉格朗日空间坐标；$\mathbf{w}(t,\mathbf{X})$表示结构的动态响应，定义于 $\Omega_s \subset \mathbb{R}^2$； μ_s 为单位长度质量； EI 为柔度；$\mathbf{q}(t,\mathbf{X})$表示

由流动产生的力。一般而言，结构通常可分为两类：刚体和柔性体。对于本文考虑的二维问题，柔性体可视为基于欧拉-伯努利梁理论的刚性梁单元组合 [52]。欧拉坐标（流体）与拉格朗日坐标（结构）之间的转换可通过狄拉克δ函数实现 [7, 53],

$$\begin{aligned} \phi(\mathbf{x},t) &= \int_{\gamma_{\mathbf{X}}} \mathbf{\Phi}(\mathbf{X},t)\delta(\mathbf{x}-\mathbf{X})d\gamma_{\mathbf{X}}, \qquad \gamma_{\mathbf{X}} \in \Gamma \\ \mathbf{\Phi}(\mathbf{X},t) &= \int_{\gamma_{\mathbf{x}}} \phi(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X})d\gamma_{\mathbf{x}}, \qquad \gamma_{\mathbf{x}} \in \Gamma \end{aligned} \tag{3}$$

其中， $\mathbf{\Phi}$ 和 ϕ 分别表示拉格朗日坐标和欧拉坐标中的变量； δ表示狄拉克δ插值函数； $\gamma_{\mathbf{x}}$ 和 $\gamma_{\mathbf{X}}$ 分别对应浸入界面边界 Γ中的欧拉网格集和拉格朗日网格集。

传统上，通常采用有限差分法（FDM）或有限体积法（FVM）求解方程1，而采用有限元法（FEM）求解方程2。然而，传统流固耦合求解器的计算成本可能非常高，这推动了可微混合神经求解器的发展。该方法将离散化的流固耦合物理（方程1-2）融入深度神经网络架构，旨在实现高效可靠的流固耦合模拟。

2.2. 可微混合神经流固耦合建模

本文提出的可微分混合神经流固耦合模型基于可微编程构建，将传统深度学习框架加以扩展，以更有效地纳入物理原理。图 1(a) 展示了所提出的混合神经网络架构概览。该架构采用长短期记忆序列网络，由一系列融入流固耦合物理的循环单元组成。如图 1(b) 所示，融入物理规律的循环单元将流固耦合物理（即离散形式的控制偏微分方程）编码为神经网络架构中不可训练的部分，并与可训练部分无缝结合，构成完整的长短期记忆序列网络。所有模块均使用 JAX 编写，其中的自动微分（自动微分）可对整个程序进行梯度反向传播，使整个混合模型能够以类似经典序列神经网络的序列到序列方式进行训练。

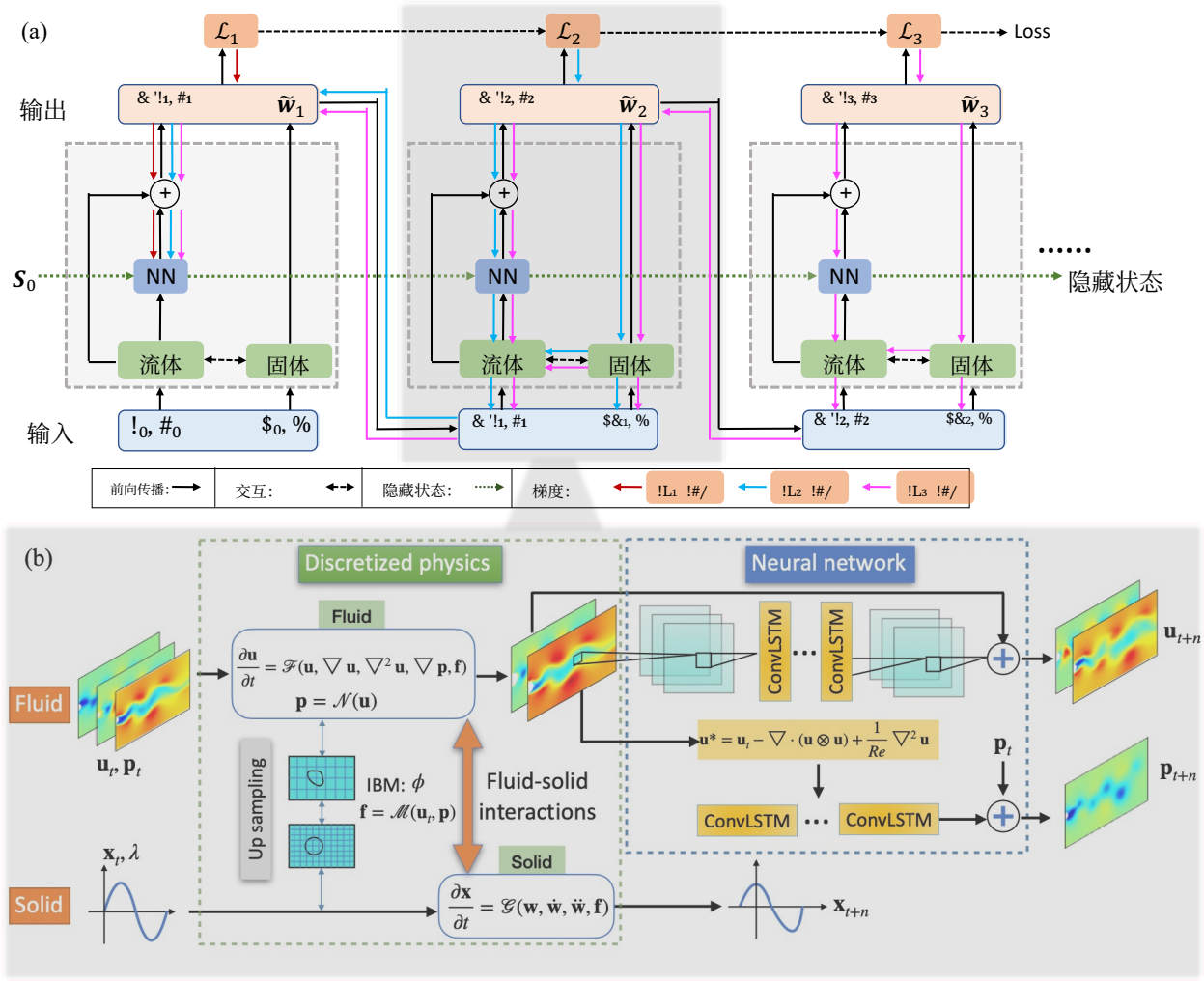

图1：(a) 用于流固相互作用仿真的可微混合神经架构概览，其中“流体”和“固体”分别表示流体与固体动力学的控制偏微分方程；“神经网络”表示可训练的神经结构。整体混合神经模型采用基于流固相互作用物理集成循环单元的长短期记忆序列网络构建。(b) 流固相互作用物理集成循环单元的详细示意图。

具体而言，对于当前时间步 t的每个循环单元，输入为前一学习步预测的速度（$\mathbf{u}_t$）、压力（p_t）和结构响应（$\mathbf{w}_t$）。输入首先经过物理编码部分，其中包含由流固相互作用控制方程中的偏微分方程算子决定的不可训练空间卷积；这可以视为在低分辨率网格上通过卷积层实现的经典数值时间步进。随后，信息继续通过额外的可训练卷积

层采用带残差连接的 ConvLSTM 形式。在物理编码部分，计算开销巨大的泊松方程完全由可训练卷积取代，以避免代价高昂的自迭代；随后按照分步法，用预测的压力校正速度 [54]。值得一提的是，学习步长（δt）可以远大于由库朗–弗里德里希–莱维（CFL）条件确定的数值时间步，从而实现快速模拟。由于固体运动的自由度较低，其控制方程可直接求解。流体与固体通过超分辨率操作进行交互，这意味着浸入边界法的掩码函数 ϕ 和直接强制法 $\mathbf{f}$ 均在超分辨率高分辨率网格上执行（见图1（b））。

2.2.1. 以离散形式编码流体物理规律

流体物理规律，即纳维-斯托克斯方程，通过分步法以基于有限体积法的离散形式进行编码。具体而言，采用前向欧拉法进行时间推进，根据离散化的动量方程估算中间速度 $\mathbf{u}^*$ ，

$$\mathbf{u}^* = -\mathbf{u}^t - \Delta t \left[\nabla \cdot (\mathbf{u}^t \otimes \mathbf{u}^t) + \nu \nabla^2 \mathbf{u}^t + \mathbf{f}^t\right], \tag{4}$$

其中， Δt 为时间步， $\otimes$ 表示张量积运算，直接浸入边界法强迫项可按以下方式计算，

$$\mathbf{f}^t = \varepsilon^t(\mathbf{x}) \left[\nabla \cdot (\mathbf{u}^t \otimes \mathbf{u}^t) - \nu \nabla^2 \mathbf{u}^t + \frac{\mathbf{u}_s^t - \mathbf{u}^t}{\Delta t}\right], \tag{5}$$

其中，$\mathbf{u}_s^t$ 是浸入式相互作用边界的目标速度，该速度由狄拉克δ函数根据结构速度进行插值（式3）； $\varepsilon^t(\cdot)$表示体积固体函数，其中 $\varepsilon^t(\mathbf{x}_s) = 1$用于固体单元$\mathbf{x}_s \in \Omega_s$，而 $\varepsilon^t(\mathbf{x}_f) = 0$用于流体单元$\mathbf{x}_f \in \Omega_f$。之后，应通过求解泊松方程来获得压力项，

$$\nabla^2 p^{t+1} = \frac{\rho \nabla \cdot [\mathbf{u}^* - \mathbf{u}^{t+1}]}{\Delta t} \tag{6}$$

不可压缩条件 $\nabla \cdot \mathbf{u}^{t+1} = 0$在流体域 Ω_f内成立，但在固体域 Ω_s 内不成立，因为 Ω_s 内的流场解不具有物理意义。因此，应针对固体域修正不可压缩条件，

$$\nabla \cdot \mathbf{u}^{t+1} = \nabla \cdot \left[\varepsilon^t(\mathbf{x})(\mathbf{u}_s^{t+1} - \mathbf{u}_c^{t+1})\right] \approx \nabla \cdot \left[\varepsilon^t(\mathbf{x})(\mathbf{u}^* - \mathbf{u}_c^t)\right] \tag{7}$$

其中 $\mathbf{u}_c^t$ 是浸没固体的精确速度。因此，流固界面区域的修正泊松方程为:

$$\nabla^2 p^{t+1} = \frac{\rho \nabla \cdot [(1-\varepsilon^t)\mathbf{u}^* + \varepsilon^t \mathbf{u}_c^t]}{\Delta t}, \tag{8}$$

注意，当 $\varepsilon = 0$时，上述方程简化为纯流体的标准泊松压力方程。然而，我们不求解计算量通常很大的泊松方程，而是采用可训练神经网络来加速这一过程。具体而言，使用可训练的卷积长短期记忆网络层来学习泊松算子，并遵循与方程8相同的物理公式,

$$p^{t+1} = \mathscr{F}_{conv_1}[(1-\varepsilon^t)\mathbf{u}^* + \varepsilon^t \mathbf{u}_c^t, \mathbf{s}_{conv_1}^t, \boldsymbol{\lambda}; \boldsymbol{\theta}_1] + p^t, \tag{9}$$

其中，$\mathscr{F}_{conv_1}[\cdot]$ 表示在时间 t、具有隐藏状态 $\mathbf{s}_{conv_1}^t$ 和可训练参数 $\boldsymbol{\theta}_1$的卷积长短期记忆网络神经函数； $\boldsymbol{\lambda}$ 表示物理参数，例如结构的材料属性。为简洁起见，下文省略输出隐藏状态$\mathbf{s}_{conv_1}^{t+1}$ 。物理规律编码部分输出的中间下一步流体速度如下所示,

$$\widetilde{\mathbf{u}}^{t+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla p^{t+1} \tag{10}$$

中间输出 $\widetilde{\ }\mathbf{u}^{t+1}$ 经过另一个带有残差连接的可训练 ConvLSTM 模块，以获得最终速度输出 $\mathbf{u}^{t+1}$,

$$\mathbf{u}^{t+1} = \mathscr{F}_{conv_2}(\widetilde{\mathbf{u}}^{t+1}, \mathbf{u}^t, \varepsilon^{t+1}, \varepsilon^t, \mathbf{s}_{conv_2}^t, \boldsymbol{\lambda}; \boldsymbol{\theta}_2) + \widetilde{\mathbf{u}}^{t+1}, \tag{11}$$

其中，固体体积分数函数 ε 用作掩码，以提供固体边界的位置信息；由可训练参数 $\boldsymbol{\theta}_2$ 参数化的卷积长短期记忆网络神经函数 $\mathscr{F}_{conv_2}$ 利用偏微分方程编码部分中的流体和固体信息，以及前序步骤的隐藏状态 $\mathbf{s}_{conv_2}^t$ ，来预测下一步速度。

2.2.2. 以离散形式编码固体动力学

固体动力学通过经典数值技术直接编码到混合神经模型中。采用标准伽辽金方法 [55], 固体变形方程（方程 2）

可离散化为常微分方程,

$$\mathbf{M}\frac{\partial^2 \mathbf{w}}{\partial t^2} + \mathbf{C}\frac{\partial \mathbf{w}}{\partial t} + \mathbf{K}\mathbf{w} = \mathbf{Q} \tag{12}$$

其中，$\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]^T$表示 n 个结构节点的动态响应向量，每个节点的响应由其 x 和 y 位移分量 w_x 和 w_y组成，即$\mathbf{w}_i = [w_{ix}, w_{iy}]^T$。全局质量、阻尼和刚度矩阵分别表示为$\mathbf{M}, \mathbf{C}$,和$\mathbf{K}$。最后，$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n]^T$表示流体力向量。流体力$\mathbf{q}_i = [q_i^D, q_i^L]^T$ 可通过沿相邻欧拉节点对直接强制法f 进行积分来计算,

$$q_i^D = -\int_{\gamma_\mathbf{x}} f_x \delta(\mathbf{x} - \mathbf{X}) d\gamma_\mathbf{x} \cong -\sum_{\mathbf{x}_i \in \gamma_\mathbf{x}} \hat{f}_x(\mathbf{x}_i)\Delta x \Delta y \tag{13}$$

$$q_i^L = -\int_{\gamma_\mathbf{x}} f_y \delta(\mathbf{x} - \mathbf{X}) d\gamma_\mathbf{x} \cong -\sum_{\mathbf{x}_i \in \gamma_\mathbf{x}} \hat{f}_y(\mathbf{x}_i)\Delta x \Delta y \tag{14}$$

其中，q_i^D和 q_i^L分别表示 x 和 y 方向上 i^{th} 节点受到的阻力和升力。项 $f_x = \mathbf{f} \cdot \mathbf{n}_x$ 和 $f_y = \mathbf{f} \cdot \mathbf{n}_y$ 分别表示欧拉强迫项的 x 和 y 分量，其中单位方向向量$\mathbf{n}_x$ 和$\mathbf{n}_x$ 分别沿 x 和 yˆ轴； □ 表示根据式3将力从欧拉网格扩散至拉格朗日网格。求解式12所用的时间积分格式分别为刚体 [56] 采用的四阶龙格-库塔方法，以及柔性体 [57],采用的Newmark-β方法。本文采用流体与结构之间的弱耦合。刚体和柔性体对应的式12-14的详细形式见附录A。

2.3. 可微编程支持的序列到序列训练

整个可微混合神经模型都可以像经典序列网络（例如循环神经网络或 Transformer）一样，采用序列到序列的方式进行整体训练，因为所有组件均由可微编程构建。对于序列到序列学习，同时考虑流体与固体动力学的损失函数定义如下,

$$\mathcal{L}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \alpha\left[\mathcal{L}_p(\boldsymbol{\theta_1}) + \mathcal{L}_v(\boldsymbol{\theta_2})\right] + \beta\mathcal{L}_s(\boldsymbol{\theta_1}, \boldsymbol{\theta_2}) \tag{15}$$

其中，$\boldsymbol{\theta_1}$ 和 $\boldsymbol{\theta}_2$ 是 ConvLSTM 层的可训练参数；$\mathcal{L}_p$、$\mathcal{L}_v$和 $\mathcal{L}_s$ 分别表示与流体压力、速度和固体响应相关的损失分量。各损失分量的详细表达式如下，

$$\mathcal{L}_p(\boldsymbol{\theta}_1) = \frac{1}{N}\sum_{t=0}^{N-1}\|p^t + \mathscr{F}_{conv_1}[(1-\varepsilon^t)\mathbf{u}^* + \varepsilon^t\mathbf{u}_c^t, \mathbf{s}_{conv_1}^t, \boldsymbol{\lambda}; \boldsymbol{\theta}_1] - \mathbf{p}_d^{t+1}\|_{L_2}^2 \tag{16}$$

$$\mathcal{L}_v(\boldsymbol{\theta}_2) = \frac{1}{N}\sum_{t=0}^{N-1}\|\widetilde{\mathbf{u}}^{t+1} + \mathscr{F}_{conv_2}(\widetilde{\mathbf{u}}^{t+1}, \mathbf{u}^t, \varepsilon^{t+1}, \varepsilon^t, \mathbf{s}_{conv_2}^t, \boldsymbol{\lambda}; \boldsymbol{\theta}_2) - \mathbf{u}_d^{t+1}\|_{L_2}^2 \tag{17}$$

$$\mathcal{L}_s(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{1}{N}\sum_{t=0}^{N-1}\|\mathbf{w}^{t+1} - \mathbf{w}_d^{t+1}\|_{L_2}^2 \tag{18}$$

其中，$\|\cdot\|_{L_2}$ 表示 L2 范数，N 表示滚动时间步的总数，下标 $\square_d$ 表示标注数据。α 和 β 分别是流体损失部分和固体损失部分的加权参数。这两项损失的权重取值应使其量级相近，从而平衡流体与固体动力学的贡献。通过使用随机梯度下降法，在较长的滚动轨迹上优化总损失，训练整个可微神经模型，

$$\boldsymbol{\theta}_1^*, \boldsymbol{\theta}_2^* = \underset{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2}{\operatorname{argmin}} \left[\alpha\left(\mathcal{L}_p(\boldsymbol{\theta_1}) + \mathcal{L}_v(\boldsymbol{\theta_2})\right) + \beta\mathcal{L}_s(\boldsymbol{\theta_1}, \boldsymbol{\theta_2})\right] \tag{19}$$

训练期间，神经网络模型在多个时间步上展开，可显著提升长轨迹上的推理性能，同时提高准确性和稳定性 [30, 43]。训练过程中的多步滚动预测由可微编程实现，使梯度能够沿整个滚动预测轨迹反向传播。然而，通过可微编程直接对混合神经模型进行序列到序列训练，可能会带来一些挑战：(1) 混合神经求解器容易出现不稳定，尤其是在训练初期，因为可训练参数的初始值不符合物理规律，导致离散化物理在数值上不稳定；(2) 一次性在长轨迹上训练需要占用大量 GPU 内存，因为需要存储由大量模型滚动预测步数产生的嵌套函数梯度

。为解决这些问题，整个训练轨迹被划分为若干首尾相接的子链。如附录 B.1所示，可以根据链式法则组合各子链的梯度，从而得到用于更新可训练参数的最终梯度。卷积长短期记忆网络的隐藏状态会在子链之间保留并传递，以确保训练过程的连续性并维持模型的一致性。此处介绍的用于速度修正和压力预测的两个神经网络具有相同的架构和超参数，详情请见附录 B.3 。

3. 数值结果

3.1. 案例设置

本研究采用两个具有代表性的流固耦合基准案例，展示所提出的可微混合神经流固耦合求解器的性能：刚体的涡激振动（VIV）和柔性体的流致变形（FID）。首先，在不引入任何深度学习模块的情况下，使用经典数值流固耦合求解器，以足够高的时空分辨率直接求解第2.1 节所述的模型方程。此过程旨在为所提出的神经流固耦合模型生成用于训练和验证的高保真数据。对于刚体的涡激振动，如图2（a）所示，

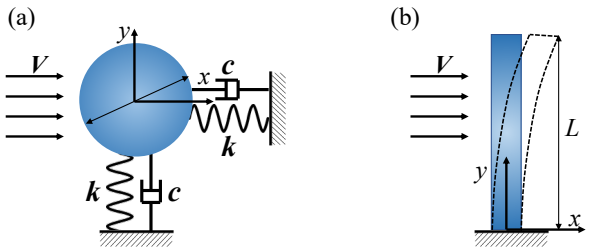

图 2：（a）涡　　激振动的刚性圆柱；（b）柔性板的流致变形　　.

问题可简化为一个典型的弹簧-阻尼（$k-c$）系统，如 [56]所述。另一方面，图2(b)中的柔性体可建模为梁，其变形采用欧拉-伯努利梁理论求解 [58]。两种情况下使用的参数

表 1：数据生成的案例参数汇总。

参数	质量比 (m^*)	固有频率 (f_n)	阻尼 (c)	-	Re
刚体	2	0.1-0.3	0	-	150
参数	质量比(m^*)	杨氏模量 (E/Pa)	阻尼比	泊松比	Re
柔性体	2.75	1×10^8	0	0.3	150

数据生成参数汇总于表1中，其中大多数参数均为无量纲参数。例如，无量纲系统质量定义为质量比 m^*,

$$m^* = \frac{4m}{\pi \rho_s B^2}, \tag{20}$$

其中， m 表示质量， B 为特征长度，其中圆柱的特征长度为 $B = D$ ，柔性梁的特征长度为 $B = L$ ； ρ_s 表示材料密度。固有频率 f_n 定义为:

$$f_n = \sqrt{\frac{k}{m}}, \tag{21}$$

其中， k 为弹簧刚度。雷诺数定义为 $Re = VB/\nu$，其中 V 为均匀来流速度的大小。为便于与以往的模拟和实验结果进行比较，通常用特征长度和固有频率对来流速度进行归一化，这称为折合速度 U_r,

$$U_r = \frac{V}{Bf_n} \tag{22}$$

刚体涡激振动（VIV）和柔性体拍振（PID）的数值结果分别展示于图3和4中。如图3所示，我们模拟的不同约化速度下的涡激振动响应结果与 Bao 等人此前得到的数值结果吻合良好。 [59]。本次模拟准确捕捉到了典型的“8”字形轨迹和“锁定”现象。图4展示了垂直柔性板变形的模拟结果。图4(a)中的振荡形态从第一模态转变为第三模态，表明本次模拟捕捉到了柔性结构的弹性。从

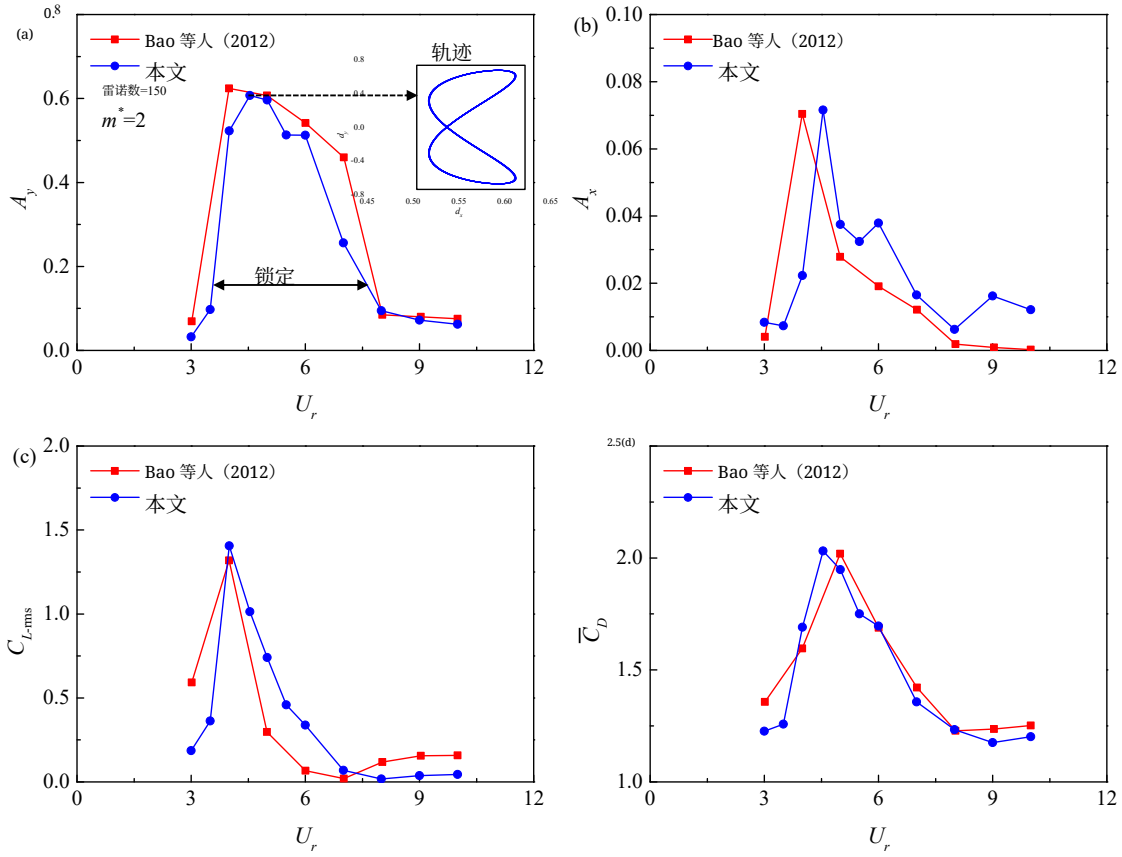

图 3：刚性圆柱涡激振动（VIV）响应的数值结果：振动幅值（A_x， A_y）分别为 (a) 横向和 (b) 流向；(c) 升力均方根C_{L_rms}；(d) 平均阻力 C_D。参考数据来自 [59]。准确模拟了典型的“8”字形轨迹和“锁定”现象。

图4(a)所示的板 的自由端随时间发生的变形可以看出，均匀流中的垂直悬臂板最初沿流向发生稳定变形，随后由于尾流中的涡脱落而开始振荡，正如 [60, 61]所报告的那样。两个不同案例所得的数值结果表明，我们的可微数值求解器能够以足够高的时空分辨率准确模拟流固耦合物理。这些高分辨率数值模拟将用于数据生成和验证研究，为所提出的混合神经模型在广泛的流固耦合场景中的训练和评估提供支持。

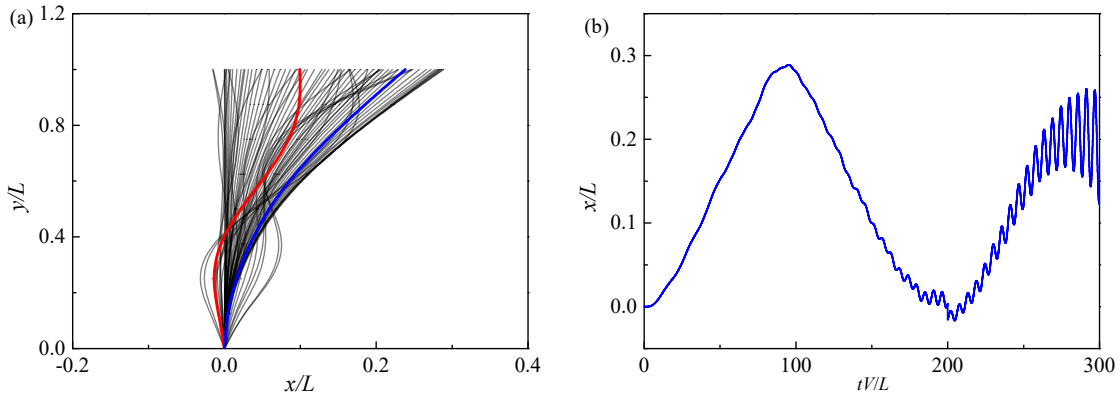

图 4：垂直柔性板流致变形（FID）响应的数值结果：(a) 板的振型从第一模态（蓝线）向第三模态（红线）变化；(b) 板自由端的时变形。

3.2. 对比基线模型

为展示我们提出的用于流固耦合模拟的可微混合神经模型的优势，我们将其与两种基线学习模型进行比较：纯数据驱动神经网络模型和弱耦合混合神经模型，如图 C.1所示。此外，我们还将其与采用相同学习模型时空分辨率的纯数值流固耦合求解器进行比较。

图 C.1(a) 所示的纯数据驱动模型是一种黑箱序列到序列神经网络，不包含任何物理先验知识。流体与固体动力学分别由两个独立的循环神经网络建模： NN_1，一个卷积长短期记忆网络，其可训练神经架构和参数与所提出的混合神经模型相同；以及 NN_2，一个基于多层感知机（MLP）的长短期记忆网络。值得注意的是，流体网络NN_1 与固体网络 NN_2 基于浸入边界法（IBM）进行耦合，其中固体网络将浸入边界法掩码信息 NN_2 提供给流体网络 NN_1 ，以实现流固相互作用。

弱耦合混合神经模型与所提出的可微模型具有相同的神经网络架构和可训练参数。如图C.1(b) 所示，它也保留了流体与固体动力学的物理规律，并按照第 2.2.1 节和第 2.2.2 节所述的相同方式进行离散化。然而，这个基线模型与所提出的模型之间的关键区别在于，其物理编码部分不可微。因此，梯度会

在每个可训练网络组件处中断，无法通过整个模型展开轨迹进行反向传播，因此该模型等同于以教师强制方式训练的离线混合网络。也就是说，该模型同样利用了流固耦合动力学的控制偏微分方程信息，但采用弱耦合方式。这个基线模型属于第 1 节所述混合模型的第一类，从求解器的角度来看，它代表了数值求解器与离线训练的神经网络修正的混合。

此后，这两种基线学习模型分别称为“纯数据驱动”和“弱耦合”模型。仅保留离散化物理部分、且不包含任何可训练组件的模型，等同于传统显式浸入边界法流固耦合求解器，以下称为“纯求解器”。

3.3. 刚体的涡激振动（*VIV*）

为评估神经模型的时间预测能力，我们在时间窗口 $tV/D = [0, 40]$内对其进行训练。随后，通过提供初始条件，并从 $tV/D = 0$到 $tV/D = 90$进行模型滚动预测，评估这些训练后模型的性能。具体而言，窗口 $tV/D = [0, 40]$ 内的模型滚动预测代表训练区域内的推理，而窗口 $tV/D = [40, 90]$ 内的滚动预测则对应时间预测。图 5 和图 6展示了所提出的混合神经求解器与基线模型的结果对比。结果表明，所提出的混合神经求解器仅使用有限的训练数据，便能准确预测较长时间跨度内的结构响应和涡旋模态。涡旋模态从发展状态转变为稳定的“2S”模态，这一过程被我们的混合神经模型准确捕捉（2rd 图 5中的一列）。这些结果与真实值（1st 图 5中的一列）及已发表的研究结果 [62]一致。相比之下，时空分辨率相同的纯数值流固耦合求解器难以捕捉真实的涡旋动力学，如 4th 图5中的一列所示。在这种情况下，大部分流动特征都因数值耗散过强而衰减消失；这是基于学习的神经求解器所采用的时空分辨率极低所导致的。同样，纯求解器对结构响应（即流向位移和横向位移 d_x， d_y）的预测

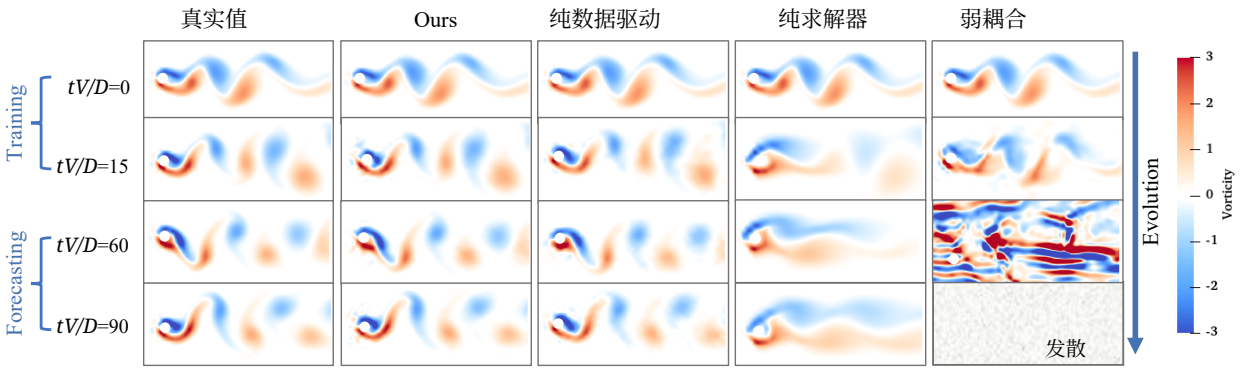

图 5：弹簧刚度和折合速度设为 $k = 0.16$，$U_r = 6.25$ 时，不同神经模型的时空涡量预测结果对比。

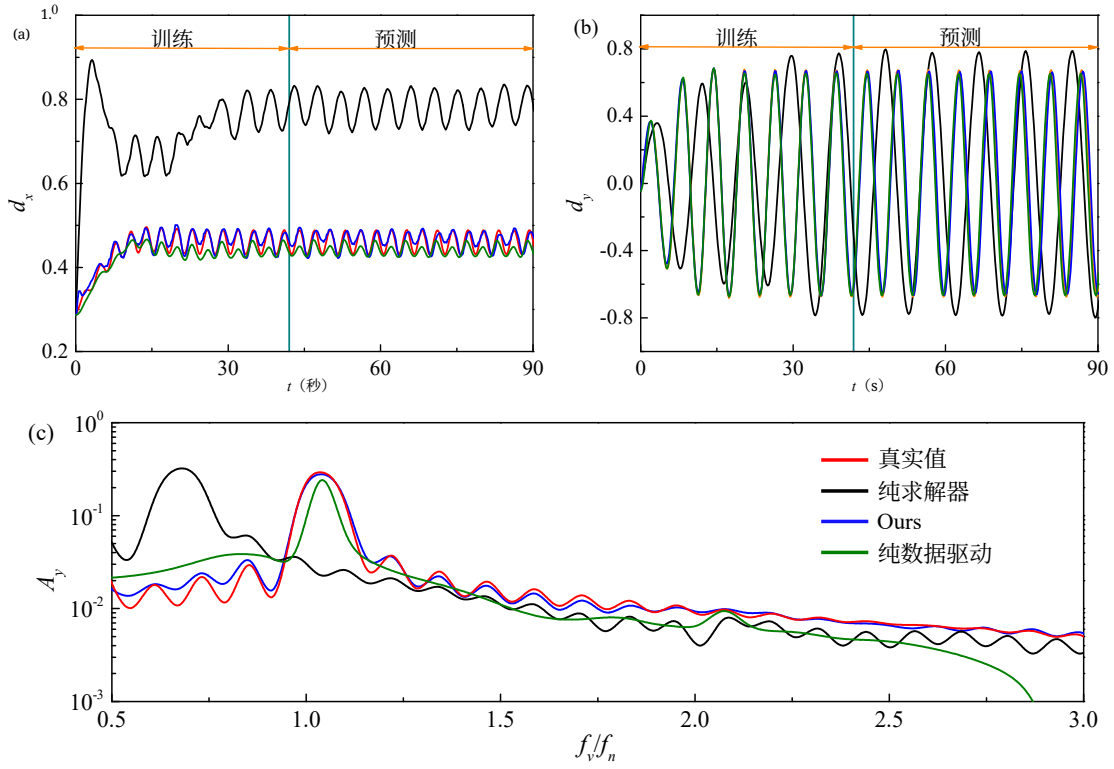

图 6：不同模型预测的结构响应对比（$k = 0.16$, $U_r = 6.25$）：(a) 流向位移；(b) 横向位移；(c) 横向位移相对于无量纲涡脱落频率的频谱 f_v。由于模拟发散，未展示弱耦合模型的结果

（绿色）也与真实值（红色）存在显著偏差，如图 6 所示；相比之下，所提出的混合模型（蓝色）的预测结果几乎与参考值重合。

请注意，此处的模型是在训练期间见过的相同参数（$k = 0.16$，$U_r = 6.25$）下进行测试的，旨在考察其时间预测性能。对于在“见过”的参数下进行测试的情况，纯数据驱动模型的时间预测性能也相当不错。纯数据驱动模型的良好表现也得益于涡激振动流动的重复性。其预测的流场和结构响应的准确性均远胜于纯数值求解器。然而，与所提出的可微混合神经模型相比，纯数据驱动模型的准确性较低。首先，纯数据驱动模型预测的远尾流区涡旋模式不够平滑，且含有明显噪声，如图3^{rd} 列所示，见图5。此外，流向位移被低估，横向位移的高频特征也未能捕捉到，如图 6 所示。尤其是，从位移预测结果的频谱分析（图 6a）可以看出，纯数据驱动模型的结果与真实值存在明显差异，而我们的混合神经模型预测结果几乎与真实值完全一致。

得益于在可微编程框架中融入流固耦合动力学的离散物理规律，所提出的混合神经模型在时间预测能力方面优于纯数据驱动模型和纯数值流固耦合求解器。尽管弱耦合模型也纳入了离散的控制偏微分方程，并显著降低了训练损失，但在测试过程中却出现不稳定和快速发散的情况，如图5最右列所示。这凸显了嵌入神经网络的求解器存在严重的稳定性问题：这类求解器使用先验损失准则进行离线训练。这也凸显了所提出的可微混合神经模型的优势：它通过端到端训练满足后验准则，并在整个解轨迹上反向传播梯度。

为进一步考察所有模型的误差随时间的传播情况，模型在每个时间步的预测相对误差 ϵ^t t 计算如下，

$$\epsilon^t = \frac{1}{N_\lambda} \sum_{i=0}^{N_\lambda} \left(\frac{\|f(\widetilde{\mathbf{u}}^t, \lambda_i; \boldsymbol{\theta}^*) - \mathbf{u}_d^t\|_2}{\|\mathbf{u}_d^t(\lambda_i)\|_2} + \frac{\|\mathbf{w}^t - \mathbf{w}_d^t\|_2}{\|\mathbf{w}_d^t(\lambda_i)\|_2} \right) \tag{23}$$

其中 N_λ 为参数集合的大小； f 表示使用优化参数 $\boldsymbol{\theta}^*$训练得到的神经流固耦合求解器；$\mathbf{w}$ 为神经网络模型计算得到的结构响应；$\mathbf{u}_d^t$ 和 $\mathbf{w}_d^t$ 为真实数据。如图 7所示，

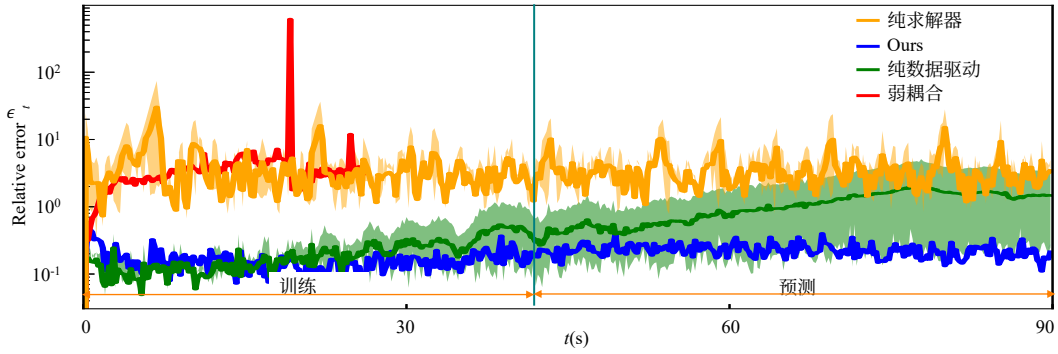

图7：不同模型的相对预测误差随时间的演化，其中粗线表示平均值，阴影区域表示误差散布。

表2：不同模型预测结果的平均相对误差

$\bar{\epsilon}$	差分混合神经（我们的方法）	纯数据驱动	纯求解器	弱耦合
流体	**0.16**	0.67	3.76	∞
结构	**0.02**	0.11	2.49	∞

纯求解器（黄色）的误差从一开始就高出数个数量级，这是由于时空分辨率较低。弱耦合神经求解器（红色）的滚动误差即使在训练范围内也会迅速增大，并在 $t = 28$s 时迅速发散。对于纯数据驱动求解器（绿色），滚动误差在训练范围（$0 < t < 40$s）内起初较低，但在预测范围（$t > 40$s）内由于误差累积而迅速增大一个数量级；这在大多数数据驱动时间预测模型中是众所周知的问题。相比之下，我们提出的可微混合神经模型（蓝色）保持了令人印象深刻的低误差，即使将预测范围外推至训练范围长度的两倍也是如此。此外，与纯数据驱动基线相比，误差散布显著减小，表明所提出模型具有出色的鲁棒性。最后，完整滚动轨迹上的平均相对误差列于

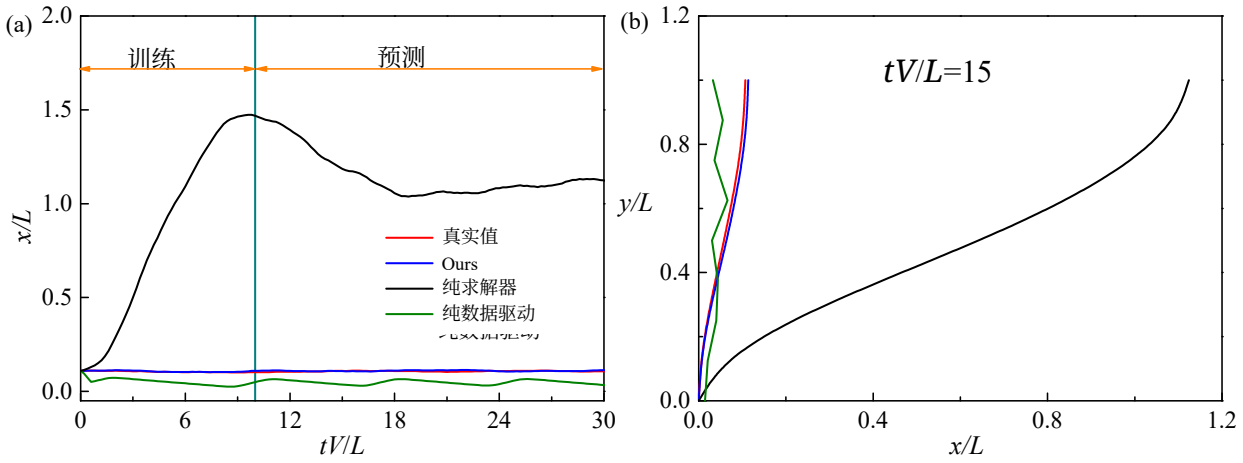

图 8：不同模型的时间序列结构响应：(a) 柔性板自由端的流向位移随时间的变化；(b) 柔性板在 $tV/L = 15$ 时的振动模态。

中。2，这展示了所提出模型相较于其他基线方法的显著优势。

3.4. 柔性结构的流致变形（*FID*）

为全面展示所提出的混合神经流固耦合求解器在各种流固耦合场景中的卓越性能，我们对图中所示柔性板的流致变形进行建模。1与第一个案例相比，由于结构柔度增加，本案例中的流动模式和结构响应周期性较弱。我们采用与第一个案例架构和超参数完全相同的卷积长短期记忆网络来构建该混合神经模型。如第3.1节所述，垂直柔性板主要沿流向发生变形，呈现前三阶振动模态。为展示本方法的优势，我们选取最复杂的时段，即板以第三模态振动的时段。模型在时间窗口 $tV/L = [0, 10]$内进行训练，随后通过输入初始条件，并在 $tV/D = 0$ 至 $tV/D = 30$范围内进行模型滚动预测，来评估其性能。具体而言，窗口 $tV/D = [0, 10]$ 内的模型滚动预测代表训练范围内的推理，而窗口 $tV/D = [10, 30]$ 内的滚动预测则对应时间预测。结果见图8和图9，其中未包括对弱耦合模型的分析

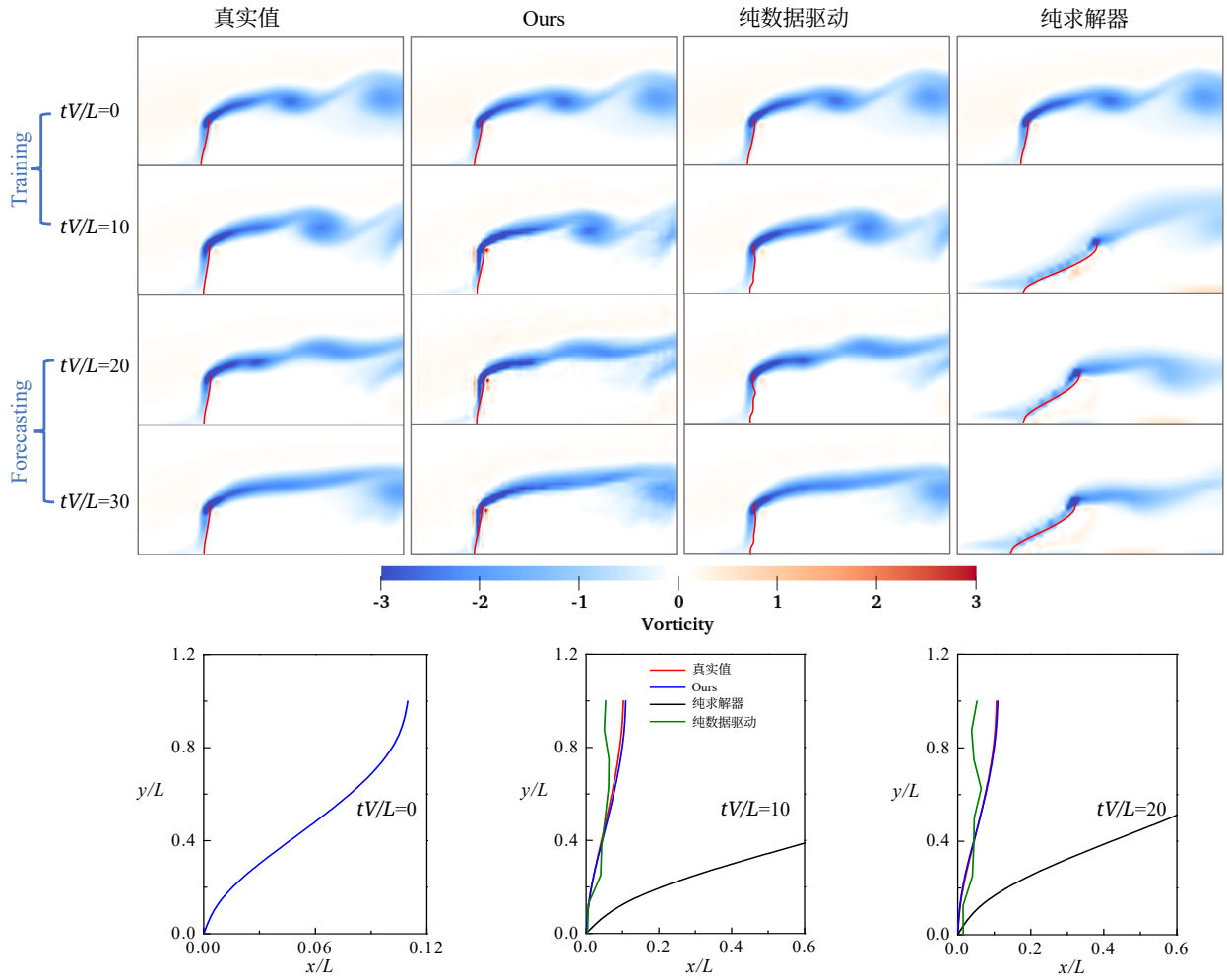

图 9：柔性结构不同模型的涡量和位移对比，其中以红线表示柔性结构以第三模态振动的情况。

，因为该模型在测试期间再次迅速发散。以柔性板自由端的位移为例，显然，所提出的可微混合神经模型能够准确预测自由端位移的时程，与真实值吻合良好（见图8(a)）。与第一个案例类似，纯粗网格流固耦合求解器难以捕捉结构响应，因为它显著高估了自由端位移。尽管纯数据驱动模型给出的位移预测在幅值上似乎合理，但图8(b) 中展示的振动模态表明，其结果完全不符合物理规律，这说明不含任何

物理组件在涉及柔性体的复杂流固耦合场景中表现较差。相比之下，所提出的混合神经流固耦合模型基于物理的控制方程求解结构响应，确保结果不仅准确，而且符合物理规律。在这种情况下，混合神经模型和纯数据驱动模型都能够捕捉非周期性流动模式，如图9所示。

4. 讨论

4.1. 对未见材料参数的泛化能力

本小节旨在通过评估模型在参数空间中多个测试点上的预测表现，考察所提出模型的泛化能力。具体而言，我们研究模型对涡激振动中未见材料参数（例如弹簧刚度）进行准确预测的能力，这些参数未包含在训练数据集中。为此，我们将不同弹簧刚度值（即 $k = 0.18$ 和 $k = 0.14$）对应的初始条件输入训练好的神经流固耦合模型，然后展开模型预测较长时间范围内的系统行为。

我们以外推的刚度参数 $k = 0.18$ 为例。图10展示了不同时刻的预测流场，圆柱从 $tV/D = 0$ 到 $tV/D = 60$ 的时变位移历程则绘制于图11中。神经流固耦合求解器成功捕捉了涡旋模态从$tV/D = 0 - 20$到 $tV/D = 40 - 60$的发展过程，与真实值高度吻合。这种准确性也体现在结构响应模拟中，涵盖了过渡阶段和稳定的循环阶段。这些结果表明，即使材料参数超出训练集范围，训练后的混合神经模型仍具有出色的预测能力。然而，正如预期的那样，纯求解器完全无法准确捕捉流固耦合动力学。面对训练集之外的新材料参数情形时，其他数据驱动基线模型则遇到了更大的困难。如图10所示，弱耦合神经模型在最初几个展开步数内便迅速发散。纯数据驱动模型虽然在训练时间范围内的预测尚属合理（如图所示）

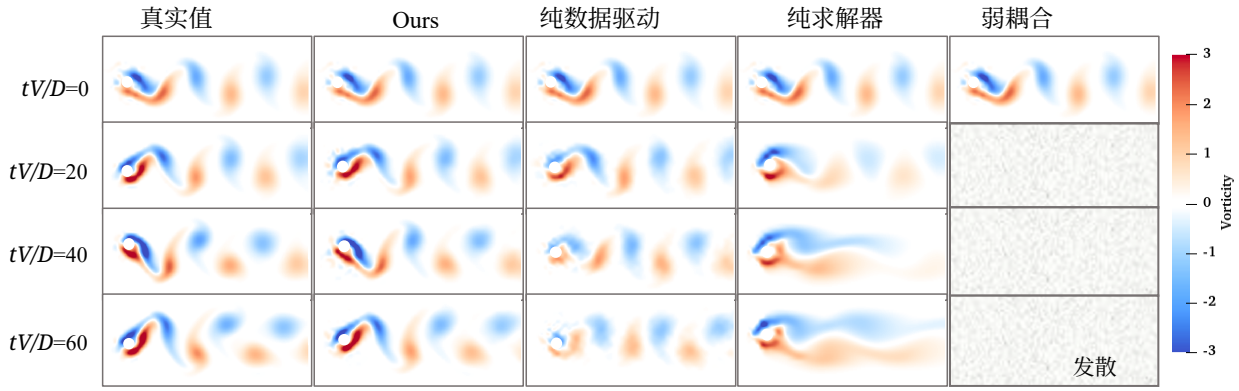

图10：未见测试参数$k = 0.18(U_r = 5.56)$下不同模型的涡量时间序列对比。

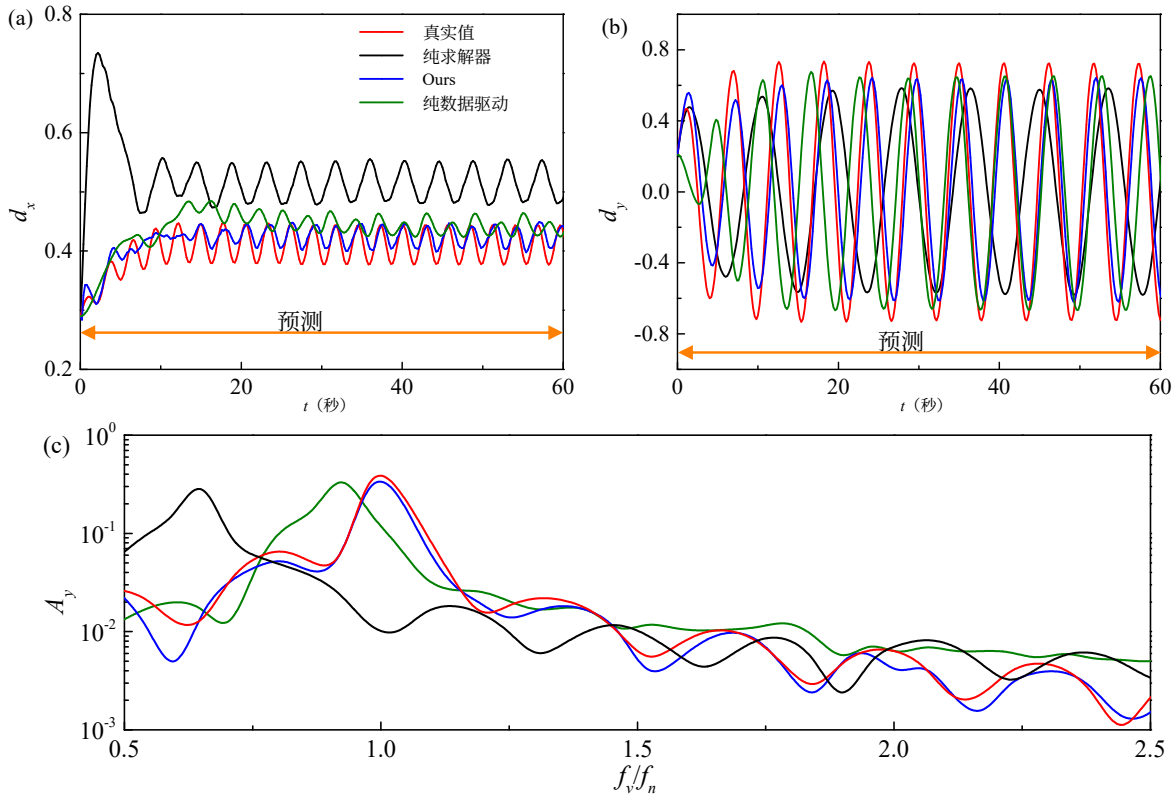

图11：未见参数 $k = 0.18(U_r = 5.56)$下的结构响应时间序列：(a) 流向位移；(b) 横向位移；(c) 横向位移的频谱。由于求解器在未见参数下会直接完全崩溃，因此此处未列出弱物理信息模型的结果。

在前一种情况下，对于新的刚度参数，模型难以从最初几个展开步开始准确捕捉流体动力学和结构响应。预测的涡旋模态在近尾流区域不准确，在远尾流区域则大幅衰减（见 3rd 图10列）。涡脱落频率预测不准确，也导致纯数据驱动模型的结构响应预测结果与真实值之间存在较大差异，如图 10所示。

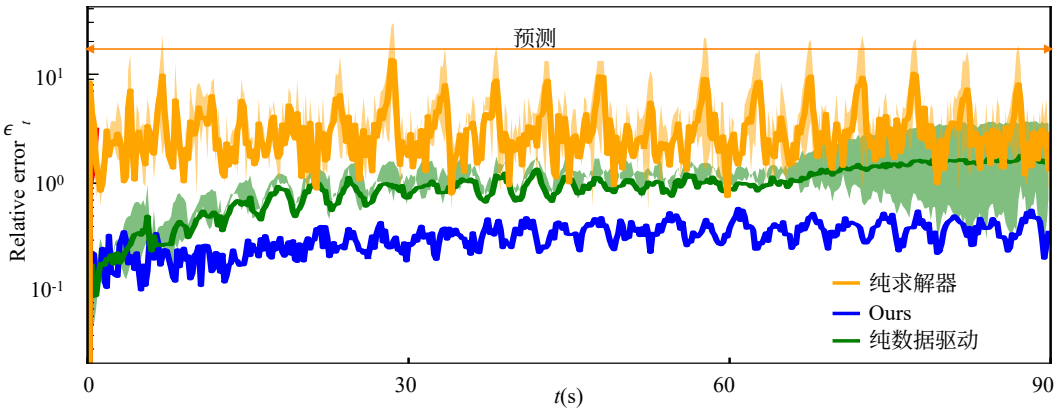

图12：不同模型相对预测误差的时间演化，其中粗线表示平均值，阴影区域表示所有测试参数下的误差散布。

Ta表3：不同模型预测结果的平均相对误差（未见刚度）

$\bar{\epsilon}$	差分混合神经（我们的方法）	纯数据驱动	纯求解器	弱耦合
流体	**0.21**	1.01	4.00	∞
结构	**0.05**	0.64	2.44	∞

图中比较了不同模型的相对预测误差随时间的演化情况。12，表中汇总了整个滚动轨迹上的平均误差值。3。显然，在相同的时空分辨率下，我们提出的模型的预测误差显著低于其他基线模型或纯数值求解器。值得注意的是，与纯数据驱动模型形成鲜明对比的是，所提出的混合神经模型几乎没有误差累积；纯数据驱动模型的预测误差则从模型滚动预测开始便持续增长。

图中汇总了两个训练参数和两个未见参数（k 或 U_r）下模型预测结果的统计量。13。如图所示，对于插值参数（$k = 0.14$， $U_r = 7.14$）和外推参数（$k = 0.18$， $U_r = 5.56$），所提出的神经求解器都能准确预测流向位移，包括波动分量和均值分量，如图13（a）和13（c）所示。对于训练参数之间插值得到的参数，横向振动幅值预测得非常准确；然而，对于外推参数，预测值略有低估。

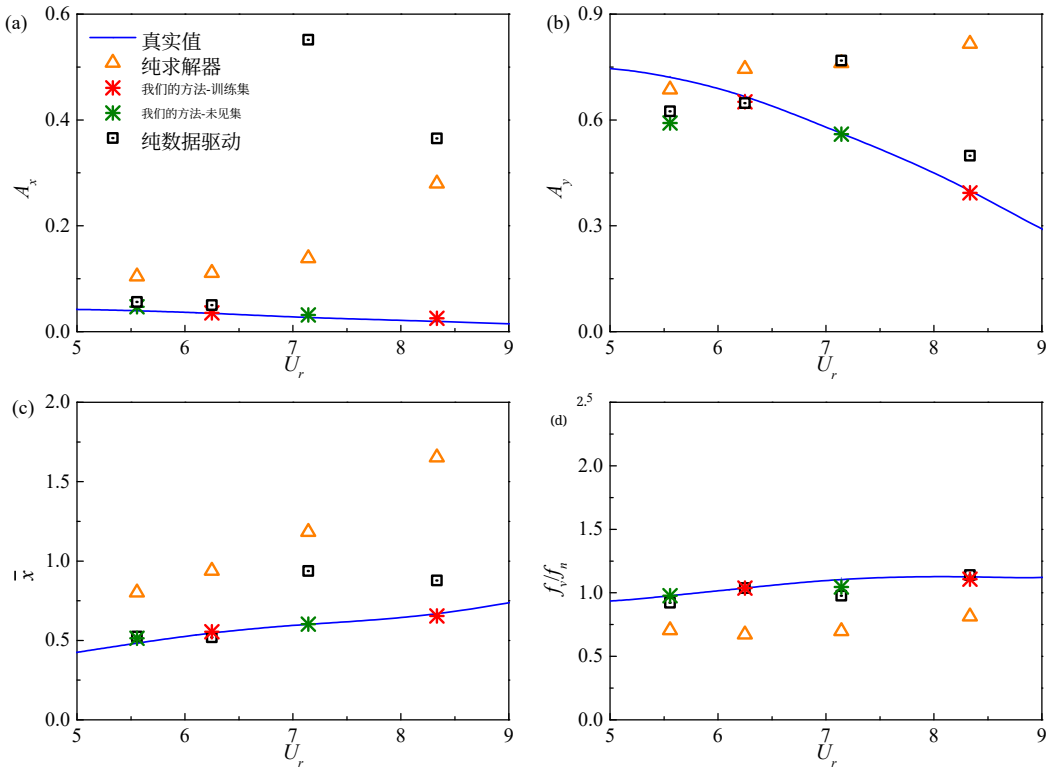

图13：参数空间中的预测结果对比：(a) 流向波动位移的均方根（RMS）；(b) 横向位移的均方根（RMS）；(c) 流向平均位移；(d) 归一化涡脱落频率。

基于上述分析，所提出的混合神经求解器能够准确预测未见参数下的流固耦合动力学，表现出良好的鲁棒性和泛化能力。它优于现有的用于流固耦合的数据驱动 [21]和混合模型 [23, 24]；这些模型只能预测针对训练参数的时间序列响应

。此外，该分析表明，由可微编程实现的序列学习策略在学习复杂的时空流固耦合动力学方面优于下一步网络。除了采用序列网络（卷积长短期记忆网络）外，当前神经网络模型中的离散化物理规律也采用递归形式进行表述，可将其理解为一个序列网络，其不可训练参数由经典数值方法预先设定。因此，所提出的神经流固耦合模型能够在长期滚动预测中，预测双向流固相互作用的时间连贯性。

4.2. 离线训练成本与在线推理加速

如图 14所示，我们评估了各种模型在在线预测阶段（亦称推理阶段）的运行时间。对于单个学习步长，我们训练得到的混合神经流固耦合模型的推理成本仅为数值流固耦合求解器在细网格上执行一个数值时间步成本的 20%。考虑到每个学习步长通常涵盖多个数值时间步，所提出的神经模型在长时间跨度预测中可实现显著加速。例如，如图 14中橙色柱状图所示，该图展示了二维涡激振动算例的相对成本；在模拟相同物理时间长度时，混合神经模型可将计算速度提高 8.3 倍。这种加速得益于采用了空间上较粗的网格和较大的时间步进间隔，后者是数值时间步的两倍。在某些情况下，由于结构柔性导致的数值不稳定性，需要采用强流固耦合方案和较小的数值步长。强耦合方案还要求通过子迭代来求解每个时间步的结构响应。这意味着高分辨率模拟的计算成本将进一步大幅增加，尤其是对于多自由度结构而言。然而，只需利用有限的高分辨率数据完成训练，经过优化的混合神经模型便能维持简单的弱耦合方案，并以更大的学习步长进行推演。因此，混合神经模型带来的加速效果可能会更加显著。

使用 RTX A6000 GPU 训练神经模型所需的运行时间，与在 8 个 CPU 上进行一次高保真流固耦合模拟的运行时间相当。然而，训练属于一次性的离线开销，而在线评估

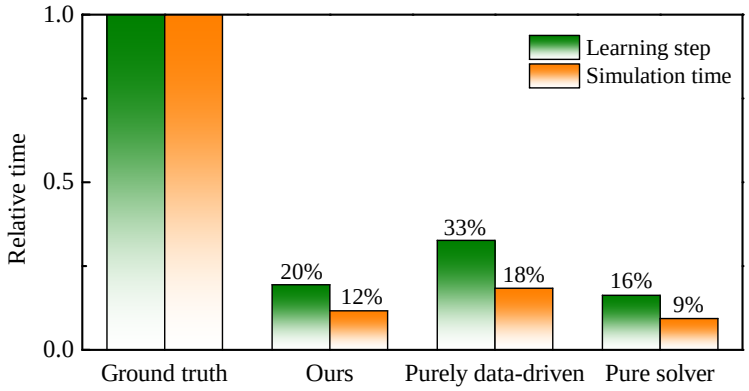

图14：不同模型的相对运行时间，用于评估神经流固耦合求解器的加速效果。真实数据是在 512×256 个网格上计算的。神经模型运行在 64×32 个网格上，每个维度的网格数减少至原来的 $8\times$ 。神经求解器的学习步长是细网格高保真流固耦合模拟数值时间步的两倍。

训练好的模型的成本则非常低。因此，对于需要长期预测并在参数空间内进行大量模型查询的场景而言，这是一种理想的解决方案。

例如，非线性流固耦合问题通常需要进行长时间的模拟，以获得可靠的统计量。在这类情况下，只要能通过短期数值模拟确定准确的初始条件，就可以使用训练好的神经网络模型预测长期动力学行为。在此背景下，与直接进行高保真模拟相比，训练和推理的成本会显著降低。此外，混合神经建模还能大幅提升参数空间内的设计优化和不确定性量化效率，因为训练好的模型无需重新训练即可推广到新的参数。

我们必须承认，目前神经流固耦合求解器的训练成本高于纯数据驱动模型。对于包含 300 个展开步数的序列到序列训练，神经求解器的训练成本是纯数据驱动模型的 7.7 倍（以运行时间衡量）。成本增加是因为反向传播过程会追踪所有可微模块中的梯度，因此既消耗大量内存又耗时。不过，推理性能和泛化能力的显著提升可以抵消这部分训练开销，正如第4.1节所示。

5. 结论

本文提出了一种可微混合神经模型，用于高效模拟流固耦合问题。通过将采用有限体积法和浸入边界法进行数值离散的流固耦合物理规律，与采用可微编程的深度序列神经网络相结合，我们实现了对整个混合模型进行端到端的序列到序列训练，以进行长期模型滚动预测。

通过两个典型的流固耦合基准问题，分别涉及刚体和柔性体的动力学，验证了神经流固耦合求解器的优点和有效性。即使训练数据集有限，该求解器也能在较长时间步内准确地模拟流动模式、流体力和结构响应。与黑箱神经网络模型相比，所提出的混合神经求解器在泛化能力、鲁棒性和预测速度方面展现出显著优势。它还克服了弱耦合混合模型遇到的数值不稳定性问题，因为全可微架构支持基于后验 准则的序列到序列训练。离散化的控制偏微分方程被集成到基于长短期记忆网络的序列网络中，从而显著缓解了误差累积问题；这一问题在大多数数据驱动预测模型中都十分常见。

尽管离线训练过程较为耗时，但这只是一次性投入。模型训练完成后，其在线推理速度显著快于经典数值方法，同时能够达到相近的准确性。对于长期预测，加速比还可进一步提高至 8.3 倍。这表明，所提出的神经求解器非常适用于许多需要长时段模拟或重复查询模型的流固耦合应用，例如设计优化和不确定性量化。

总之，通过可微编程框架将深度学习与经典数值技术无缝融合，成功发挥了两种方法各自的优势，从而为复杂流固耦合动力学建模提供了一种强有力的方法。这一创新方法在开发新一代数据赋能神经求解器方面展现出巨大潜力，可兼顾准确性与计算效率，是应对大规模、长时流固耦合模拟所带来挑战的理想工具。

致谢

作者谨此感谢美国海军研究办公室通过项目编号 N00014-23-1-2071 提供的资助，以及美国国家科学基金会通过项目编号 OAC-2047127 提供的资助。XF 还谨此感谢圣母大学环境变化倡议和可持续能源中心提供的奖学金。

遵守伦理标准

利益冲突：作者声明不存在任何利益冲突。

附录A. 两类固体的控制方程

结构动力学控制方程（式12）的具体形式因刚性固体和柔性体而异。对于忽略变形的刚性固体，方程可简化为质量-阻尼器-弹簧系统，对应于涡激振动（VIV）现象。在这种情况下，式$\mathbf{M}$、$\mathbf{K}$和$\mathbf{C}$ 中的矩阵12 均简化为标量值。时间步 t 处的二自由度方程可改写为：

$$m \begin{bmatrix} \ddot{w}_x \\ \ddot{w}_y \end{bmatrix} + k \begin{bmatrix} w_x \\ w_y \end{bmatrix} + c \begin{bmatrix} \dot{w}_x \\ \dot{w}_y \end{bmatrix} = \begin{bmatrix} F_D \\ F_L \end{bmatrix}, \tag{A.1}$$

其中， m 为质量； $k = 2m \cdot (2\pi f_n)^2$, 和 f_n 为固有频率； $c = 2m \cdot 2\pi f_n \cdot \xi_s$，其中$\xi_s$为结构阻尼比； $F^D = \sum_{i=1}^{n} q_i^D$和 $F_L^t = \sum_{i=1}^{n} q_i^D$分别为阻力和升力， n为拉格朗日节点总数。对于一般的柔性固体，结构通过组装梁单元进行建模。仅考虑二维域中梁单元的弯曲时，可构建柔性固体的一致质量矩阵$\mathbf{M}$ 和刚度矩阵$\mathbf{K}$ 。由于每个梁单元由 i^{th}和$(i+1)^{th}$ 节点组成，因此可以推导出单个单元的局部刚度矩阵和质量矩阵：

$$\mathbf{K}^e = \frac{EI}{L_e^3} \begin{bmatrix} 12 & 6L_e & -12 & 6L_e \\ 6L_e & 4L_e^2 & -6L_e & 2L_e^2 \\ -12 & -6L_e & 12 & -6L_e \\ 6L_e & 2L_e^2 & -6L_e & 4L_e^2 \end{bmatrix}, \tag{A.2}$$

$$\mathbf{M}^e = \frac{\mu_s L_e}{420}\begin{bmatrix} 156 & 22L_e & 54 & -13L_e \\ 22L_e & 4L_e^2 & 13L_e & -3L_e^2 \\ 54 & 13L_e & 156 & -22L_e \\ -13L_e & -3L_e^2 & -22L_e & 4L_e^2 \end{bmatrix}, \tag{A.3}$$

其中 $L_e = z_{i+1} - z_i$。这里，阻尼矩阵基于瑞利阻尼构建 [63]:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}, \tag{A.4}$$

其中系数为 $\alpha = 2\xi_s \frac{\omega_i \cdot \omega_j}{\omega_i + \omega_j}$, $\beta = \frac{2\xi_s}{\omega_i + \omega_j}$。$\omega_i$ 是通过以下方式求解的 ith 固有圆频率:

$$(\mathbf{K} - \omega^2\mathbf{M})\mathbf{\Phi} = 0, \tag{A.5}$$

其中 $\omega = diag[\omega_1 \cdots \omega_i \cdots \omega_j \cdots \omega_n]$ 是特征值，$\mathbf{\Phi}$ 是特征向量。

附录 B. 混合神经模型的实现与训练细节

附录 *B.1.* 分段长轨迹训练

序列到序列训练过程非常耗费内存，尤其是在结合可微物理组件时。为克服内存约束，当前训练中将整个轨迹划分为多个子链。完整轨迹的均方误差（MSE）损失计算如下:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=0}^{N}(f(\tilde{\mathbf{u}}_t;\boldsymbol{\theta}) - \mathbf{u}_t^d)^2, \tag{B.1}$$

其中 $f(\tilde{\mathbf{u}}_t;\ \boldsymbol{\theta})$ 是混合神经流固耦合模型的输出；$\mathbf{u}_t^d$是标签数据；而 N是时间步数。如果将整个轨迹划分为 n 时间跨度相等的子轨迹，每个子轨迹包含 N_{sub} 个时间步。公式 $\mathcal{L}$ 中的损失B.1 可

计算如下:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=0}^{n}\sum_{t=0}^{N_{sub}}(f(\tilde{\mathbf{u}}_{t,i};\boldsymbol{\theta})-\mathbf{u}_{t,i}^{d})^{2}\\ &= \frac{N_{sub}}{N}\sum_{i=0}^{n}\frac{1}{N_{sub}}\sum_{t=0}^{N_{sub}}(f(\tilde{\mathbf{u}}_{t,i};\boldsymbol{\theta})-\mathbf{u}_{t,i}^{d})^{2}\\ &= \frac{N_{sub}}{N}\sum_{i=0}^{n}\frac{1}{N_{sub}}\|f(\tilde{\mathbf{u}}_{t,i},\boldsymbol{\theta})-\mathbf{u}_{t,i}^{d}\|_{2}^{2}\\ &= \frac{N_{sub}}{N}\sum_{i=0}^{n}\mathcal{L}_{i}\end{aligned}$$

因此，总梯度可写为:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}} = \frac{N_{sub}}{N}\sum_{i=0}^{n}\frac{\partial\mathcal{L}_{i}}{\partial\boldsymbol{\theta}}, \tag{B.2}$$

其中， N_{sub} 表示每个子轨迹中的时间步数； n 表示子轨迹的数量。需要注意的是，在训练过程中，必须将最终输出和隐藏状态传递至后续子轨迹。所提出的混合神经流固耦合模型的训练算法见算法 1。

Algo算法 1 可微混合神经流固耦合模型的训练算法 el

数据: 加载流体和/或结构数据标签

初始化: 初始化神经网络和离散化物理

$epoch \leftarrow 0$ **当** $epoch \leqslant N$ **执行时**

  对于 $t \in [1, N_{sub}]$ **，执行**

    $\tilde{}\mathbf{u}_t^* \leftarrow \mathcal{F}$ （$\mathbf{u}_{t-1}$， $\mathbf{x}_{t-1}$） ▷ 求解中间速度

    $\mathbf{p}_t \leftarrow \mathscr{F}$($\tilde{}\mathbf{u}_t^*$， $\mathbf{x}_{t-1}$，状态； $\boldsymbol{\theta}_1$) ▷ 预测压力

    $\widetilde{\mathbf{u}}_t \leftarrow \widetilde{\mathbf{u}}_t^* - \frac{\nabla\mathbf{p}_t\Delta t}{\rho}$ ▷ 获取低分辨率速度

    $\mathbf{u}_t \leftarrow \mathscr{F}$($\tilde{}\mathbf{u}_t$， $\mathbf{x}_{t-1}$，状态； $\boldsymbol{\theta}_2$) ▷ 预测速度

    $\mathbf{x}_t \leftarrow \mathcal{G}$ （$\mathbf{u}_t$， $\mathbf{x}_{t-1}$， λ） ▷ 直接求解结构响应

  end

  $\mathcal{L} \leftarrow$ ||预测 –标签||$_{\mathrm{L}}^{2}$ $_{2}$

  $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 - l\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}_1}$

  $\boldsymbol{\theta}_2 \leftarrow \boldsymbol{\theta}_2 - l\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}_2}$

end

附录 *B.2.* 高分辨率真实数据的下采样

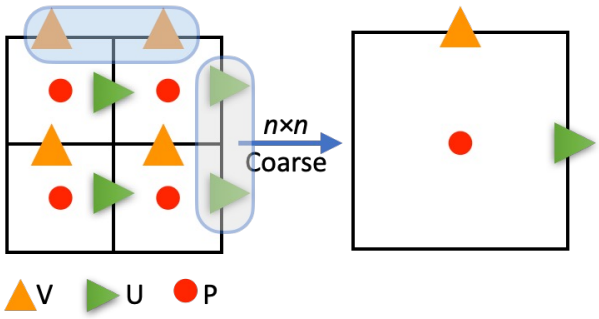

图 B.1：将交错变量下采样至粗 2 倍网格的示意图。对位于粗网格单元各面的细网格速度分量（如阴影区域所示）取平均，以计算粗网格上的速度。对位于细网格单元中心的压力取平均，以计算粗网格上的压力。单元内部的速度分量将被舍弃。

附录 *B.3.* 可训练卷积长短期记忆网络模块的超参数

在混合神经流固耦合模型中，所有 ConvLSTM 模块均相同，由五层组成，通道配置为 $[32, 64, 64, 64, 32, 1]$，并采用可训练的 3×3 卷积核。激活函数为修正线性单元（ReLU），最后一层除外，该层不使用激活函数。优化设置如下：

- 初始学习率 $= 10^{-4}$
- 优化器 $=$ Adam
- 调度器 =余弦衰减调度 (alpha=10^{-9})

流体损失项和固体损失项的加权参数 α 和 β 均设为 1。

附录 C. 基准数据驱动神经模型

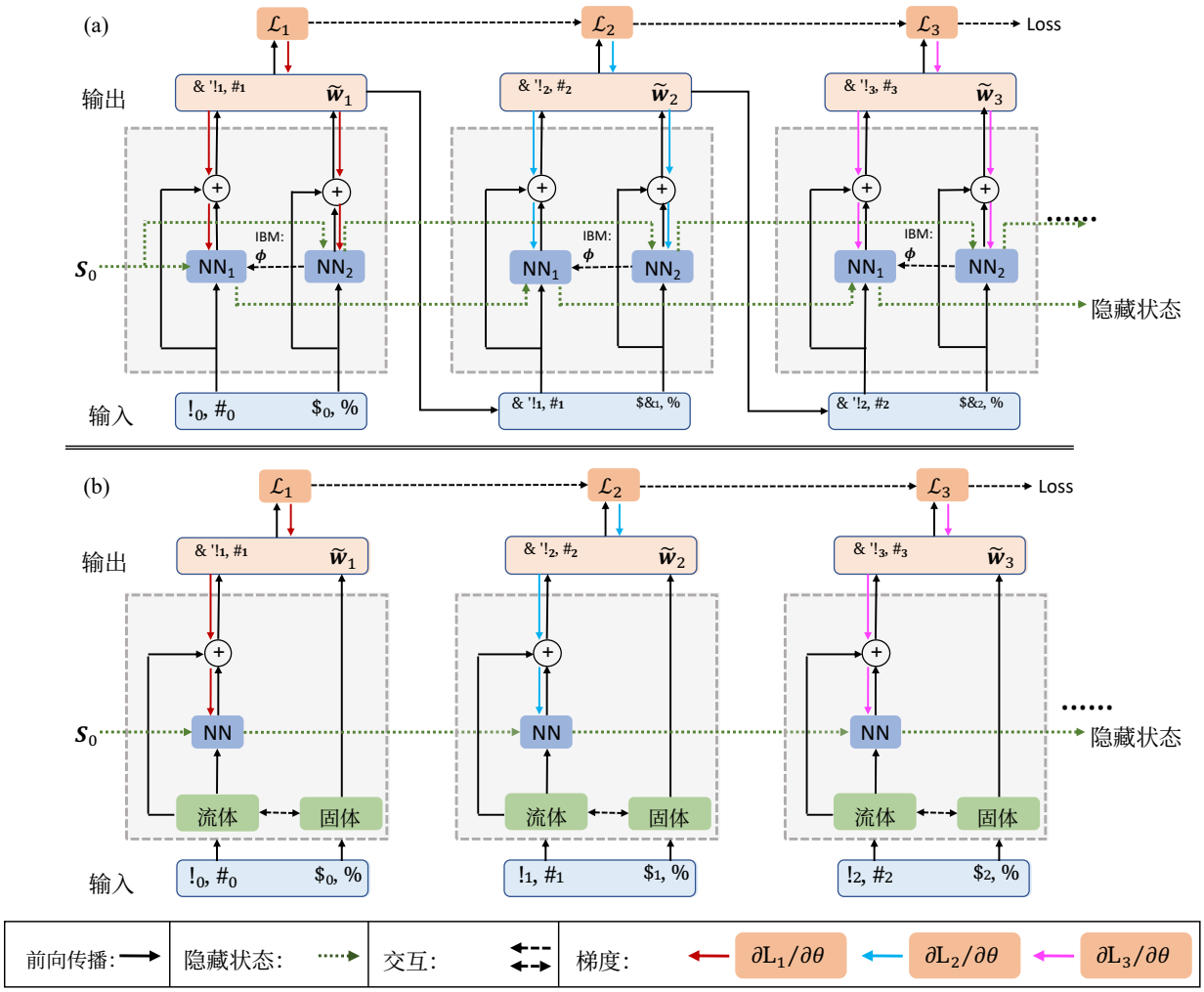

图 C.1: （a）纯数据驱动模型，其中已知物理规律被完全移除，流体与固体在两个时间步之间的动力学分别由 LSTM 神经网络映射。流体网络和固体网络通过 IBM 掩码连接。（b）弱耦合模型，其中保留了离散化物理规律，但梯度无法通过这些物理规律进行反向传播，相当于以教师强制方式训练的离线网络。其可训练神经架构与所提出的神经模型保持相同。

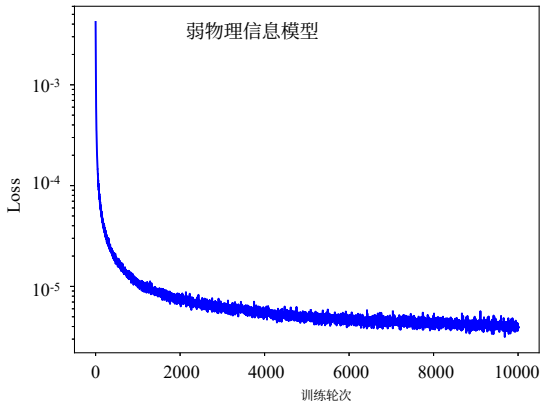

图 C.2：弱耦合混合神经模型的训练损失。

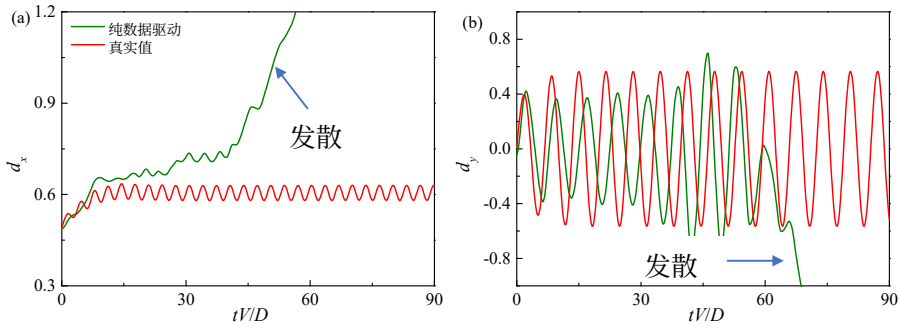

图 C.3：纯数据驱动模型预测的发散结构响应（适用于 $k = 0.14$、 $U_r = 7.14$）：(a) 流向位移；(b) 横向位移。

参考值

[1] J. Boustani、M. F. Barad、C. C. Kiris、C. Brehm，《一种用于薄型高柔顺性壳体结构的浸入边界流固耦合方法》，Journal of Computational Physics 438 (2021) 110369。

[2] B. E. Griffith, N. A. Patankar，《用于流固耦合的浸入方法》，《流体力学年度评论》52（2020年），421。

[3] B. Ramaswam y，M. Kawahara，非定常拉格朗日–欧拉有限元方法 d

用于非定常、对流、不可压缩黏性自由表面流体流动，《国际流体数值方法杂志》7 (10) (1987) 1053–1075。

[4] J. Donea、S. Giuliani、J.-P. Halleux，瞬态流固交互的任意拉格朗日-欧拉有限元方法，《计算力学与工程方法》33 (1-3) (1982) 689–723。

[5] J. A. Sethian，《水平集方法与快速行进方法：计算几何、流体力学、计算机视觉和材料科学中的界面演化》，第 3 卷，剑桥大学出版社，1999。

[6] C. S. Peskin，浸入边界法，《数值纪事》11 (2002) 479–517。

[7] C. S. Peskin，心脏瓣膜周围的流动模式：一种数值方法，《计算物理学杂志》10 (2) (1972) 252–271。

[8]E. Fadlun、R. Verzicco、P. Orlandi、J. Mohd-Yusof，结合浸入边界有限差分方法进行三维复杂流动模拟，《计算物理学杂志》161 (1) (2000) 35–60。

[9] R. Fu、D. Xiao、I. Navon、C. Wang，基于自编码器、自注意力和深度学习方法的流体流动数据驱动降阶模型，arXiv 预印本 arXiv:2109.02126（2021）。

[10] M. Guo、J. S. Hesthaven，面向时变问题的数据驱动降阶建模，《计算力学与工程方法》345（2019）75–99。

[11] S. Bhatnagar、Y. Afshar、S. Pan、K. Duraisamy、S. Kaushik，使用卷积神经网络预测气动流场，《计算力学》64（2019）525–545。

[12] L. Zhu、W. Zhang、J. Kou、Y. Liu，翼型周围亚音速流动的湍流建模机器学习方法，《流体物理学》31 (1) (2019) 015105。

[13] P. Du、X. Zhu、J.-X. Wang，基于深度学习的三维患者特异性计算流体力学代理模型，《流体物理学》34 (8) (2022) 081906。

[14]Y. Zhang, W. Jiang, L. Sun, J.-X. Wang, X. Zheng, Q. Xue, 一种基于深度学习的正常发声期间声门气流广义经验流动模型，《生物力学工程学杂志》（2022年2月）。arXiv:https://asmedigitalcollection.asme.org/biomechanical/article-pdf/doi/10.1115/1.4053862/6848039/bio-21-1098.pdf， doi:10.1115/1.4053862。

[15] J.-X. Wang, J.-L. Wu, H.Xiao，基于直接数值模拟数据重构雷诺应力建模偏差的物理信息机器学习方法，《物理评论：流体》2（3）（2017）034603。

[16] S. Pan, K. Duraisamy，封闭模型的数据驱动发现，《SIAM应用动力系统杂志》17(4)(2018) 2381–2413。

[17] R. Maulik, O. San, A. Rasheed, P. Vedula，基于神经网络的二维湍流亚格子建模，Journal of Fluid Mechanics 858 (2019) 122–144。

[18] H. J. Bae, P. Koumoutsakos，用于湍流壁面模型的科学多智能体强化学习，《自然》通讯 13 (1) (2022) 1443。

[19] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P. W. Battaglia，利用图网络学习基于网格的模拟，arXiv 预印本 arXiv:2010.03409（2020年）。[20] X. Han, H. Gao, T. Pfaff, J.-X. Wang, L. Liu，利用时间注意力在网格降阶空间中预测物理规律，载于：国际学习表征会议，2022年。[21] R. Gao, R. Jaiman，用于流固耦合的准整体式图神经网络，arXiv 预印本 arXiv:2210.04193（2022年）。[22] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang，物理规律知情的机器学习，《自然综述：物理》3（6）（2021年）422–440。

[23] R. Gupta, R. Jaiman，面向移动界面和流固耦合的混合分区深度学习方法，《计算机与流体》233（2022年）105239。

[24] S. R. Bukka, R. Gupta, A. R. Magee, R. K. Jaiman，《基于混合深度学习的降阶模型的非定常流动预测评估》，《流体物理学》33 (1) (2021) 013601。[25] H. von Wahl, T. Richter，《利用深度神经网络预测不可压缩流中欠分辨三角形刚体的运动》，International Journal for Numerical Methods in Fluids 93 (12) (2021) 3364–3383。[26]N. Wandel, M. Weinmann, R. Klein，《从头学习不可压缩流体动力学——迈向可泛化的快速可微流体模型》，载于：国际学习表征会议，2020年。[27] D. Fang, J. Tan，《用于流固耦合的浸入边界物理信息机器学习方法》，Ocean Engineering 263 (2022) 112360。

[28] M. Minakowska, T. Richter, S. Sager，用于建模非球形颗粒悬浮液的有限元/神经网络框架，《越南数学杂志》49 (1) (2021) 207–235。

[29] J. Wu, H. Xiao, R. Sun, Q. Wang，采用显式数据驱动雷诺应力闭合的雷诺平均纳维–斯托克斯方程可能存在病态问题，《流体力学杂志》869 (2019) 553–586。

[30] K. Um, R. Brand, Y. R. Fei, P. Holl, N. Thuerey，求解器在环：从可微物理中学习，以与迭代 PDE 求解器交互，《神经信息处理系统进展》33 (2020) 6111–6122。

[31] M. Raissi、P. Perdikaris、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与逆问题的深度学习框架》，Journal of Computational Physics 378 (2019) 686–707。

[32] L. Sun、H. Gao、S. Pan、J.-X. Wang，《基于物理约束深度学习、无需模拟数据的流体流动代理建模》，Computer Methods in Applied Mechanics and Engineering 361 (2020年) 112732。

[33] A.Arzani, J.-X. Wang, R. M. D'Souza，利用物理信息神经网络从稀疏数据中揭示近壁血流，《流体物理学》33 (7) (2021) 071905。doi:10.1063/5.0055600。

[34] H. Gao, M. J. Zahr, J.-X. Wang，物理信息图神经伽辽金网络：用于求解由偏微分方程支配的正问题与逆问题的统一框架，《计算力学与工程方法》390 (2022) 114502。

[35] Z. Lai, C. Mylonas, S. Nagarajaiah, E. Chatzi，基于物理信息神经常微分方程的结构识别，《声音与振动杂志》508 (2021) 116196。

[36] E. Kharazmi, D. Fan, Z. Wang, M. S. Triantafyllou，利用物理信息神经网络推断柔性圆柱的涡激振动，《流体与结构杂志》107 (2021) 103367。

[37] H.Tang、Y. Liao、H.Yang、L.Xie，一种用于涡激振动的迁移学习-物理信息神经网络（tl-pinn），Ocean Engineering 266 (2022) 113101。

[38] A. Krishnapriyan、A. Gholami、S. Zhe、R. Kirby、M.W. Mahoney，《物理信息神经网络中可能失效模式的表征》，《神经信息处理系统进展》34 (2021) 26548–26560。

[39] X.-y. Liu, L. Lu, H. Sun, J.-X. Wang，通过多分辨率偏微分方程结构保持深度学习预测参数化时空动力学，arXiv（2022）。[40] Z. Long, Y. Lu, X. Ma, B. Dong，PDE-Net：从数据中学习偏微分方程，载于：国际机器学习大会，PMLR，2018年，第3208–3216页。

[41] R. T. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud，神经常微分方程，《神经信息处理系统进展》第31卷（2018）。[42] M. Innes, A. Edelman, K. Fischer, C. Rackauckas, E. Saba, V. B. Shah, W. Tebbutt，一种连接机器学习与科学计算的可微编程系统，arXiv预印本 arXiv:1907.07587（2019）。[43]D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer，机器学习加速的计算流体力学，《美国国家科学院院刊》118（21）（2021）e2101784118。[44] B. List, L.-W. Chen, N. Thuerey，利用可微流体求解器进行学习式湍流建模：基于物理的损失函数与优化时域，Journal of Fluid Mechanics 949（2022）A25。

[45] L. G. Wright, T. Onodera, M. M.Stein, T. Wang, D. T. Schachter, Z. Hu, P. L. McMa- hon，《反向传播训练的深度物理神经网络》，《自然》601 (7894) (2022) 549–555。

[46] D. Akhare, T.Luo, J.-X.Wang，《面向复合材料制造的物理集成神经可微分（PiNDiff）模型》，《计算力学与工程中的计算方法》406 (2023) 115902。

[47] H. Frezat,J. Le Sommer, R.Fablet, G. Balarac, R. Lguensat，《准地转湍流参数化的后验学习》，《地球系统建模进展》14 (11) (2022) e2022MS003124。

[48]V. Shankar, V. Puri, R. Balakrishnan, R. Maulik，《可微物理驱动的 Burgers湍流闭合建模》，《机器学习：科学与技术》（2023）。[49] P. Holl, N. Thuerey, V. Koltun，《利用可微物理学习控制偏微分方程》，载于：国际学习表征会议，2019。

[50] D. A. Bezgin, A. B. Buhendwa, N. A. Adams，Jax-Fluids：用于可压缩两相流的全可微高阶计算流体力学求解器，Computer Physics Communications 282 (2023) 108527。[51] J. Yang, F. Stern，一种简单高效的流固相互作用直接强制浸入边界框架，Journal of Computational Physics 231 (15) (2012) 5029– 5061。[52] O. A. Bauchau, J. I. Craig，欧拉-伯努利梁理论，Springer，2009，第 173–221 页。[53] M. Uhlmann，一种用于模拟颗粒流的直接强制浸入边界法，Journal of computational physics 209 (2) (2005) 448–476。[54] A. J. Chorin，纳维-斯托克斯方程的数值解，Mathematics of com- putation 22 (104) (1968) 745–762。[55] O. C. Zienkiewicz, R. L. Taylor，固体与结构力学有限元法，Elsevier，2005。[56] X. Fan, K. Guo, Z. Jia, Y. Wang, W. Tan，亚临界雷诺数下串列圆柱的振动模态与速度干扰机制，Journal of Wind Engi- neering and Industrial Aerodynamics 199 (2020) 104136。[57] R. Kamakoti, Y. Lian, S. Regisford, S. Kurdila, W. Shyy，使用基于压力的求解器进行计算气动弹性分析，载于：第 40 届 AIAA 航空航天科学会议暨展览，2002，第 869 页。[58] O. A. Bauchau, J. I. Craig，欧拉-伯努利梁理论，Springer，2009，第 173–221 页。[59] Y. Bao, C. Huang, D. Zhou, J. Tu, Z. Han，固有频率比变化时孤立圆柱和串列圆柱上的二自由度流致振动，Journal of Fluids and Structures 35 (2012) 50–75。

[60]F.-B. Tian, H. Dai, H. Luo, J. F. Doyle, B. Rousseau, 《涉及大变形的流固耦合：三维模拟及其在生物系统中的应用》, Journal of Computational Physics 258 (2014) 451–469。[61] B. S. Connell, D. K. Yue, 《均匀流中旗帜的拍动动力学》, Journal of Fluid Mechanics 581 (2007) 33–67。[62] H. Wan, S. S. Patnaik, 《利用热效应抑制圆柱体的涡激振动》, 《流体物理学》28 (12) (2016) 123603。[63] M. Liu, D. G. Gorman, 《瑞利阻尼公式及其扩展》, 《计算机与结构》57 (2) (1995) 277–285。