arXiv:2304.14214v1 [cs.LG] 27 Apr 2023

部分变量、部分参数、部分时间，在已知部分物理学的情况下：部分信息下的辨识

Saurabh Malani普林斯顿大学化学与生物工程系

Tom Bertalan约翰斯·霍普金斯大学怀廷工程学院化学与生物分子工程系

Tianqi Cui约翰斯·霍普金斯大学怀廷工程学院化学与生物分子工程系

Jos´e L.Avalos普林斯顿大学化学与生物工程系、安德林格能源与环境中心、普林斯顿生物工程计划、分子生物学系、高草甸环境研究所

Michael Betenbaugh
系 ecular 工程 怀廷工程学院，约翰·霍普金斯大学
Chemical and Biomol

伊奥尼斯·G·凯夫雷基迪斯*化学与生物分子工程系、应用数学系 怀廷工程学院，约翰·霍普金斯大学

摘要

实验数据通常由独立测量的变量组成，这些变量具有不同的采样率（连续测量之间存在非均匀的 Δt 间隔）；在某个特定时间点，可能只能对所有变量中的一部分进行采样。针对从此类数据中辨识动力系统的方法，通常会在学习之前使用插值、插补或子采样来重新组织或修改训练数据。我们还可能拥有部分物理知识先验 （准确或近似的），而数据驱动技术可以对这些知识进行补充。在此，我们利用基于数值积分方法和先验 物理知识的神经网络架构，来辨识底层控制微分方程的右端项。这类神经网络模型的迭代使得模型能够从任意时间点采样的数据中进行学习，无需 修改数据。重要的是，我们将网络与可用的部分物理知识整合到“物理信息灰箱”中；这使得模型能够学习未知的动力学速率或微生物生长函数，同时估计实验参数。

关键词：系统辨识、动力系统、部分信息、循环神经网络、灰箱

1 引言

动力系统存在于科学和工程的各个领域，从时空混沌流或湍流，到瞬态化学反应器动力学、复杂生物系统、传染病模型，以及更广泛的其他领域。这些动力系统遵循基本的物理或数学支配定律，其中某些组成部分可能先验未知，或者只能近似获知：我们可能知道函数形式，但缺少某些参数值（例如动力学常数）；或者，我们甚至可能无法准确知道正确的函数形式（例如正确的动力学）。系统辨识是一门利用实验数据提取底层动力系统相关信息或知识，或进一步推导出有用替代模型的科学；这类模型可用于预测、将系统行为外推至训练数据范围之外、优化系统性能，或将其控制到期望的设定点。当然，这已经是系统理论中一个成熟且内容丰富的分支，并且已有许多成功应用；然而，近年来机器学习算法的爆发式发展，使人们重新关注从时间序列数据中学习模型时出现的若干问题，包括可解释性不足、能否实现有效泛化，以及如何处理缺失数据和部分观测。

神经常微分方程（Neural ODEs）和循环神经网络（RNNs）的许多应用，都侧重于学习整个时间序列的函数，例如将心电图信号分类为不同的心跳类型[52, 62]。本文关注的是从部分信息中学习自治动力系统，不过我们的方法也可以扩展到非自治系统。

在开始之前，我们先说明本文中某些术语的定义：(a) **完整观测**：在每个采样时间，都会测量所有相关因变量；(b) **部分观测**：在每个采样时间，都会测量部分变量；(c) **隐变量**：某些变量在任何时间都不会被测量；(d) **数据**Δt 是相邻采样时间之间的时间间隔；(e) **学习** δt 是我们对积分时间步长的估计，该步长应足够长以发挥作用，同时也要足够短以保证准确性；(f) **固定频率采样**：以恒定的采样频率对变量进行采样（数据 Δt 固定）；(g) **可变频率采样**：以任意采样频率对变量进行采样（数据 Δt 可变）。

系统辨识传统上是一个数据驱动过程，即使用数学、统计或机器学习技术处理输入输出数据，以提取有关底层系统规律的信息。系统理论的这一分支历史悠久，其发展历程始于著名的卡尔曼滤波器 [25]（一种线性方法），经过扩展卡尔曼滤波器 [33], 、无迹卡尔曼滤波器 [59], 和粒子滤波器 [19]等非线性变体，直至神经网络 [38, 60, 31, 12, 24, 49]等新技术。然而，经验数据虽然是这项任务不可或缺的组成部分，却并不是关于所关注系统通常可获得的唯一信息来源。控制方程、守恒定律、对称性、物理理论和/或模型，即使并不完善或不完整，也同样构成关键信息。在本文中，我们致力于考虑所有这些信息来源，开发一种混合灰箱模型，将黑箱数据驱动技术与白箱物理模型相结合。

相关方法正日益取得成功（并在近期文献中越来越受欢迎），例如物理信息神经网络（PINNs） [47] ，其中通过使用深度学习技术来施加物理知识，而物理知识通常以偏微分方程 [46]的形式给出——其中可能包含未知参数，甚至未知算子 [35]。然而，PINNs 通常用于寻找与完全已知的底层支配常微分方程/偏微分方程一致的解；我们的方法则着眼于利用已知的物理相关解（实验数据）来辨识底层系统本身的（部分）内容。尽管“PINN”一词已用于指代正向（问题求解）任务，但同期的工作 [47] 也展现了逆向辨识任务的一些特征，尽管不具备灰箱结构。随后出现了更为通用的算子学习工作 [35], ，其中学习一般的函数算子，包括积分；这种算子学习任务的较早版本例如可见于 [18]。利用机器学习，尤其是神经网络，执行这种“黑箱”任务，是一条始于20世纪80年代末和20世纪90年代初 [15, 49, 45], 的研究方向，并随着神经常微分方程或常微分方程网络 [11, 30]的出现而重新焕发生机，得到更加广泛的应用。

当已知底层微分方程右端项的部分信息时，这一辨识过程的“灰箱”版本会利用神经网络的表达能力，弥补部分已知或近似已知物理模型的不完整性或不准确性，或施加附加结构、已知的全局定律或约束。这些修正可以采用加法方式（例如 [48, 32, 26, 37, 55, 53]）或乘法方式（例如 [34, 9]），也可以将神经网络以函数方式与物理学结合（例如 [26, 44, 21, 13, 40, 57]）。

本文将重点介绍缺失参数和缺失函数项的识别：通过公式化表达强制施加质量/能量平衡或耦合关系等已知控制原理，而黑箱神经网络则学习微生物生长速率或化学反应速率等具有物理意义的函数，同时估计可能未知的参数值。需要注意的是，这与哈密顿神经

网络 [5, 20] 不同，后者在要求实现准确预测之外，还施加了动力学的结构特征（辛性或其哈密顿性质）。

在将我们的工作置于当前文献背景下时，我们考虑了几类相似研究：第一类关注使用缺失或部分信息数据训练的黑箱模型；第二类关注使用可变采样率数据训练的黑箱模型；最后一类关注部分已知物理规律（灰箱模型）。

使用缺失或部分信息数据训练的黑箱模型。实验数据的测量结果通常无法以适合数据驱动预测/学习的格式获得。传感器可能不会以固定间隔运行，手动采样的数据也可能无法以固定间隔获得，并且某些数据点可能受到外部噪声的干扰。这会产生一个以可变频率采样的数据集。此外，不同传感器可能以不同的采样率运行；在每个采样点上，也可能只进行部分分析。这会产生一个具有部分观测的数据集——在每个采样点上，仅测量所有变量中的一个子集。
为了使这类训练数据适用于统计分析和机器学习，通常会进行预处理；处理方法可能简单到省略“异常”数据点 [36], 但更常见的是采用某种形式的插补。这可以是线性插值或“向前填充最后一次观测值”的方法 [17, 41], 也可能采用更稳健的平滑技术，例如多项式平滑或局部加权散点平滑（局部估计散点图平滑）[23], 还可以使用期望最大化等机器学习技术 [39], K近邻 [6, 54, 41, 36] 或人工神经网络 [54, 39]来进行平滑和插补。

先用插补填补缺失数据 再用清理后的数据集训练模型 的方法，往往会反过来强化插补值，仿佛它们是真实值：“如果你用花生酱填补奶酪上的洞，就不该假装自己有了更多奶酪！” [23]。可以通过同时进行插补和模型训练 来缓解这一问题：利用模型插补缺失数据，再用这些数据继续训练模型（但不计算插补值，即非真实值处的损失）；然后用改进后的模型再次进行插补，并不断迭代。

基于可变时间点采样的数据训练的黑箱模型。要使用在可变采样时间点获取的数据进行训练，我们需要一种能够自然支持任意时间步进的方法。最简单的监督学习方法是学习当前状态 $x(t)$ 与未来某一时刻 Δt 的状态 $x(t+\Delta t)$ 之间的直接映射，但这种方法要么要求训练数据按固定 Δt 间隔采样，要么要求将 Δt 本身也作为神经网络的输入（图 1(a)）。另一种方法（在我们看来也更加灵活）是认识到底层系统可以表示为连续时间的微分方程。因此，如果我们改为让神经网络学习微分方程组右端的形式，就可以利用成熟的数值积分方法，在任意所需的时间点生成预测（图 1(b)）。早在 20 世纪 90 年代，文献中就提出了这种方法 [49, 50] ，近期的研究则通过神经常微分方程对其进行了扩展和推广 [11]。这项关于神经常微分方程的近期研究使其能够兼容 Python 内置的稳健数值积分器，使用伴随方法而非展开来计算梯度，并且 t orchdiffeq [10] 和 *diffrax* [28] 等软件包也已公开发布。
神经常微分方程还提供了一种自然的方式，可同时进行插补和学习：插补由时间演化模型完成，并随着模型训练在每次迭代中动态变化。在这项工作中，我们通过在 PyTorch 中将神经网络明确模板化为一个四阶龙格-库塔（RK4）积分器，来“打开”神经常微分方程方法 [49] (图 2)。在我们的工作中，训练梯度通过展开计算，而不是通过伴随方法计算。

灰箱模型——利用已知部分物理学知识从数据中学习。文献中的多项先前工作涉及缺失数据特征的不同组合。Rubanova 等人使用神经常微分方程，在变量完全隐藏的数据上学习不规则采样时间序列数据 [51], , 但没有考虑部分观测。他们采用潜在表示，这使得灰箱建模变得困难，因为潜变量未必具有直接明了的物理解释。
Kidger 等人 [29] 将神经常微分方程扩展为神经控制微分方程，其中隐藏状态随时间连续演化。他们展示了神经控制微分方程方法在多种分类任务中的应用，但没有处理时间序列的系统辨识，也没有将其与混合灰箱方法结合。相反，这项应用代表了典型函数学习 RNN 应用的一种直接连续时间类比。

Buisson-Fenet 等人 [8] 展示了一种简单的方法，将神经常微分方程与先验知识相结合，从而形成一种用于系统辨识的特定混合灰箱方法；其中，物理知识以已知动力学方程的形式给出

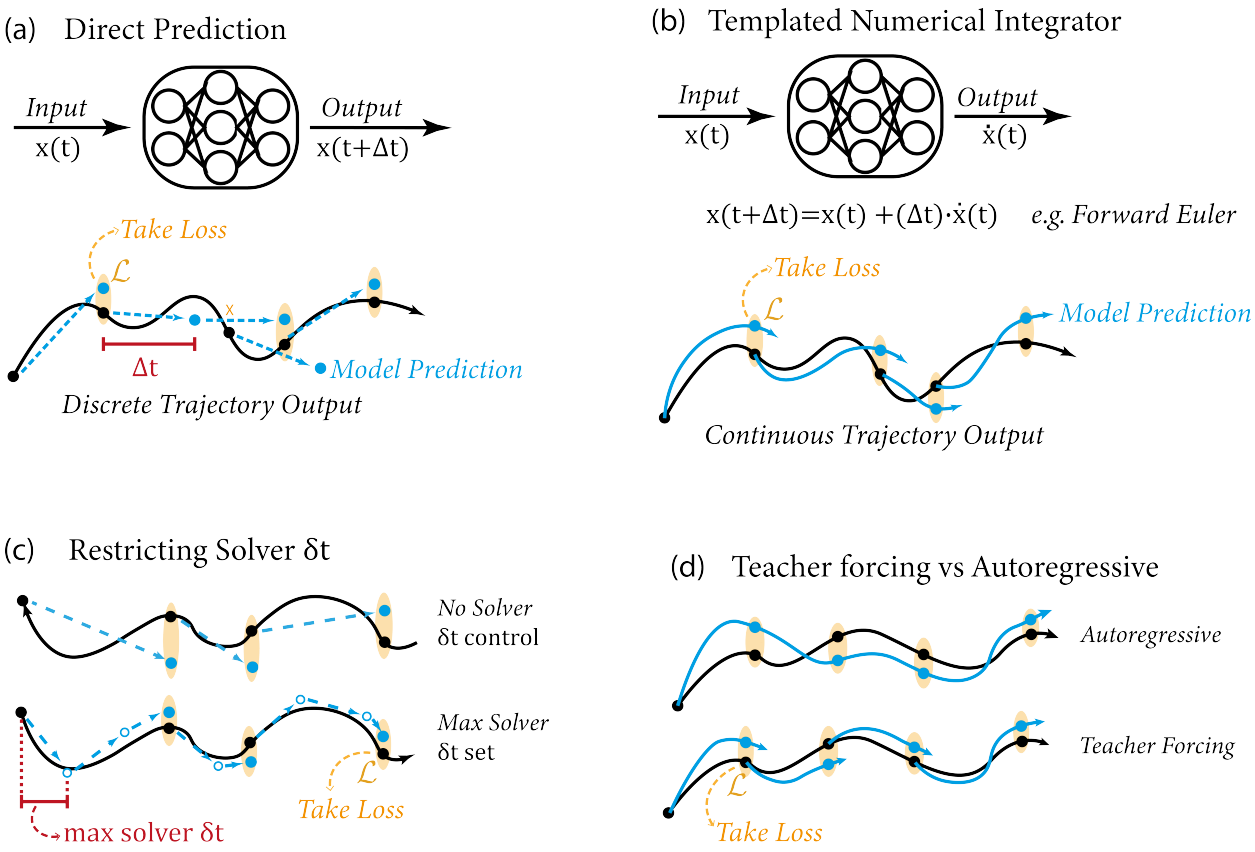


图 1：**不同循环神经网络类型的比较**。（a）直接预测方法通常会迫使神经网络采用固定的Δt，从而需要对恰好未在该 Δt采样的数据进行重采样、插值或插补。（b）将神经网络纳入带有 Δt 输入的数值积分器模板中，可以自然地实现对不同 Δt 值的预测。（c）限制求解器最大 δt 超参数后，如果数据Δt 较大，神经网络就必须采取多个步长。（d）我们同时研究教师强制和自回归方法：在教师强制中，有真实数据时将其用作输入；在自回归中，则将预测结果反馈给神经网络，用于下一次预测。在部分观测数据上进行训练期间，教师强制和自回归两者的要素都会被使用；而在推理阶段，则从已知的初始条件开始使用自回归。

，而神经网络则学习一个系统参数（未知频率）。除了通过融入先验信息处理一系列灰箱模型外， [8] 的一项关键贡献是基于（Kazantzis-Kravaris-Luenberger）KKL 观测器理论，使用同步动力系统来推断缺失的初始条件。这种方法解决了在包含隐变量的数据上进行训练的问题，但无法处理当前工作中的部分观测。另有研究提出使用核方法（而非神经网络）来学习动力系统，或预测长短期记忆网络潜在动力学中缺失的初始条件 [27]。

在我们的工作中，神经网络学习具有物理意义的函数，例如微生物生长速率或化学反应速率，同时还学习固定实验参数的未知值。 1

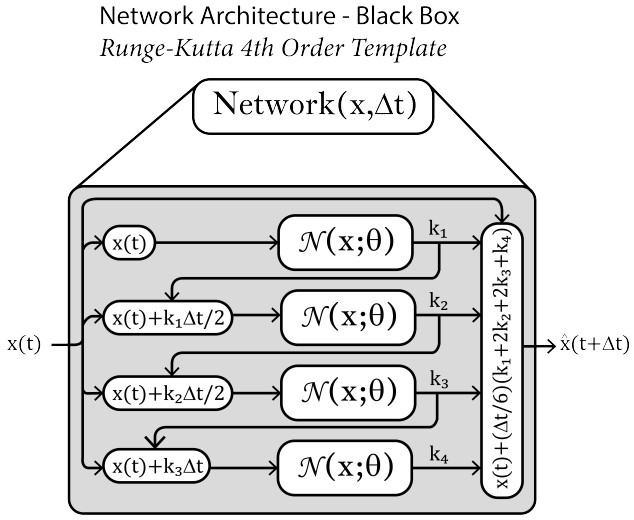
本文其余部分安排如下：在给出上述术语的定义后，我们首先概述支撑本方法及其算法的关键概念。我们首先在一个成熟的示例模型——自主搅拌釜反应器 [56] 上展示本方法，并逐步增加用于学习模型的训练数据的复杂度。随后，我们转向一个更复杂的系统——一个六维生物动力学模型，其中涉及三种共培养的微生物物种，它们呈现自主振荡 [3]。在这两个系统中，我们展示了如何利用灰箱建模来估计未知参数，同时学习未知动力学。最后，我们总结观测结果，并讨论本方法的适用范围及其优势与不足。

2 方法论

用于任意时间采样学习的常微分方程网络。为了让神经网络学习常微分方程系统的右端项（RHS），而不是直接学习离散时间下的输出，如果神经网络架构以成熟的数值积分器框架为模板（图 1(a)、图 1(b)），便会很方便。我们将这种神经网络与传统数值分析的“融合”称为常微分方程网络（文献中也称为神经常微分方程 [11]）。在本研究中，我们明确地以四阶龙格-库塔（RK4）（式 1，图 2）为模板来构建神经网络。

$$\hat{\dot{\mathbf{x}}} = \mathcal{N}(\mathbf{x}(t), \mathbf{p}; \theta) \tag{1a}$$

$$\begin{aligned}
\mathbf{k}_1 &= \mathcal{N}(\mathbf{x}(t), \mathbf{p}; \theta) \\
\mathbf{k}_2 &= \mathcal{N}(\mathbf{x}(t) + \mathbf{k}_1 \left(\frac{\Delta t}{2}\right), \mathbf{p}; \theta) \\
\mathbf{k}_3 &= \mathcal{N}(\mathbf{x}(t) + \mathbf{k}_2 \left(\frac{\Delta t}{2}\right), \mathbf{p}; \theta) \\
\mathbf{k}_4 &= \mathcal{N}(\mathbf{x}(t) + \mathbf{k}_3 \left(\Delta t\right), \mathbf{p}; \theta) \\
\hat{\mathbf{x}}(t + \Delta t) &= \mathbf{x}(t) + \left(\frac{\Delta t}{6}\right)(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)
\end{aligned} \tag{1b}$$



Fig图 2：采用四阶龙格-库塔数值积分器的黑盒神经网络模型 r

我们从数值分析中了解到，一般而言，在对已知方程进行积分时，随着步长减小，前向误差会降低，这推动了自适应步长积分方案的发展。同样可以证明，在根据假定为真实的轨迹数据发现方程时，随着步长减小，后向误差也会降低 [65]。因此，我们控制 RK4 积分器所采用的时间步长，以保持求解器的精度；这意味着可能需要通过神经网络进行多次迭代，才能获得下一个观测时间步的预测结果（图 1（c））。

我们还指出两种不同的训练方法：教师强制 和自回归（图 1(d)）：教师强制最初是在用于时间序列监督学习的循环网络背景下提出的 [61, 42]，其做法是在后续迭代中，以真实数据（教师信号）替代神经网络的输出（如果真实数据可用）。这种方法能加快模型在训练过程中的收敛，尤其是在训练的早期迭代阶段 [4]。相比之下，自回归训练会将神经网络的输出反馈给自身，作为下一次迭代的输入，从而生成完整时间轨迹 [11]。先前文献指出，在训练和模型预测中，兼顾这两种方法效果最佳 [4],，因此我们在训练方法中考虑了这一点（图 4）。

利用已知部分物理知识进行数据驱动学习的灰箱建模。 我们的目标是开发并实现一种基于神经网络的训练方法，使其能够与已知的“白箱”物理知识耦合，并能从采样时间不固定且仅包含部分观测的训练数据集中学习。这两项要求都可以通过将训练问题表述为学习常微分方程系统的右端项来满足。通常，已有的物理知识以空间和时间上的微分方程形式表示（如化学或微生物动力学、对流、扩散等）。以微分方程的形式表示动力学，能够在神经网络模型训练中施加恰当的归纳偏置，从而捕捉这些动力系统的连续性[30]；同时，它也使神经网络训练能够自然地融入现有的部分物理知识。这些方法可以扩展到偏微分方程，甚至随机偏微分方程 [14, 43], ，但这超出了本文的讨论范围。

在简单的黑箱模型中，神经网络将全权负责学习支配系统动力学的规律：

$$\hat{\dot{\mathbf{x}}} = \mathcal{N}_{BB}(\mathbf{x};\theta) \tag{2a}$$

先前已有的物理知识可能并不完整和/或并不完美，但可以纳入模型框架，使神经网络只需学习未知参数/动力学，或可能学习对已知但不完美的动力学的修正。此类纳入可以采用修正形式（加法形式（式 2b）或乘法形式（式 2c）），也可以采用函数复合形式（式 2d）。

$$\hat{\dot{\mathbf{x}}} = f_{WB}(\mathbf{x};\mathbf{p}) + \mathcal{N}_{BB_{Add}}(\mathbf{x};\theta) \tag{2b}$$

$$\hat{\dot{\mathbf{x}}} = f_{WB}(\mathbf{x};\mathbf{p}) \cdot \mathcal{N}_{BB_{Mult}}(\mathbf{x};\theta) \tag{2c}$$

$$\hat{\dot{\mathbf{x}}} = f_{WB}(\mathbf{x}, \mathcal{N}_{BB_{Func}}(\mathbf{x});\theta) \tag{2d}$$

我们也可以将修正型灰箱与函数复合型灰箱结合起来（式 3，图 3）。

$$\phi = f_\phi(\mathbf{x};\mathbf{p_f},\theta) = \begin{cases} f_{WB}(\mathbf{x};\mathbf{p_f}) + \mathcal{N}_{BB_{Add}}(\mathbf{x};\theta) & \text{Additive} \\ f_{WB}(\mathbf{x};\mathbf{p_f}) \cdot \mathcal{N}_{BB_{Mult}}(\mathbf{x};\theta) & \text{Multiplicative} \end{cases} \tag{3a}$$

$$\hat{\dot{\mathbf{x}}} = g_{WB}(\mathbf{x},\phi;\mathbf{p_g}) \tag{3b}$$

其中， f_ϕ 是本构关系，或用于描述动力学速率或微生物速率的函数；对于这些函数，我们可能有先验假设 f_{WB}。神经网络会学习对 f_{WB}中任何缺陷或不完整之处进行修正；如果没有先验 假设，则会学习整个 f_ϕ 。 ϕ 和 x 是另一个白箱模型 g_{WB} 的输入，该模型严格施加不可违背的物理定律，例如守恒定律，或我们希望施加的其他全局定律，例如反应速率与反应热之间的耦合。请注意，此处白箱模型的参数$\mathbf{p_f}$和$\mathbf{p_g}$ 既可以先验固定；也可以在数据足以支持辨识时，作为额外训练参数纳入模型，与神经网络参数 θ同时拟合。

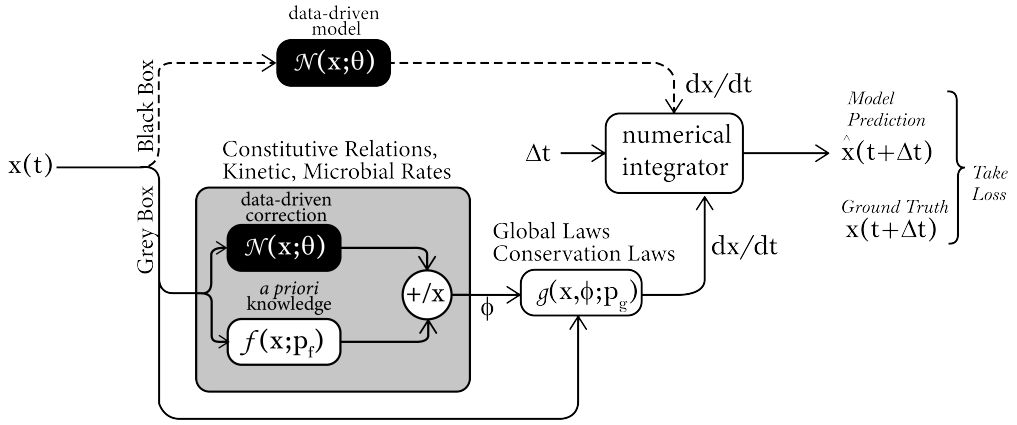


图 3：**黑箱和灰箱模型的网络架构**。在黑箱模型中，学习右端项的全部任务都由神经网络承担。在灰箱模型中，则通过本构项或已知的微生物和反应速率，以及全局定律和守恒定律，融入物理信息。

在我们的方法中，我们专注于函数复合，其中神经网络负责学习未知的微生物或动力学速率，而其架构由施加已知物理耦合的白箱结构组成

（例如反应速率与反应热生成之间的关系，或微生物生长与底物消耗/产物形成之间的关系（公式4）。

$$\phi = \mathcal{N}_{BB}(x;\theta) \tag{4a}$$

$$\hat{\dot{x}} = g_{WB}(x,\phi;p_g) \tag{4b}$$

训练算法与批处理。 训练算法总结于图 4。在每条训练轨迹中，会标出**关键**时间点（计算损失的位置）和**求解器**时间点（为确保模板化数值积分器的稳定性和精确度而纳入），并通过沿时间方向逐步遍历每条轨迹，利用当前的向量场估计生成预测。每条训练轨迹的关键时间点和求解器时间点可能各不相同，

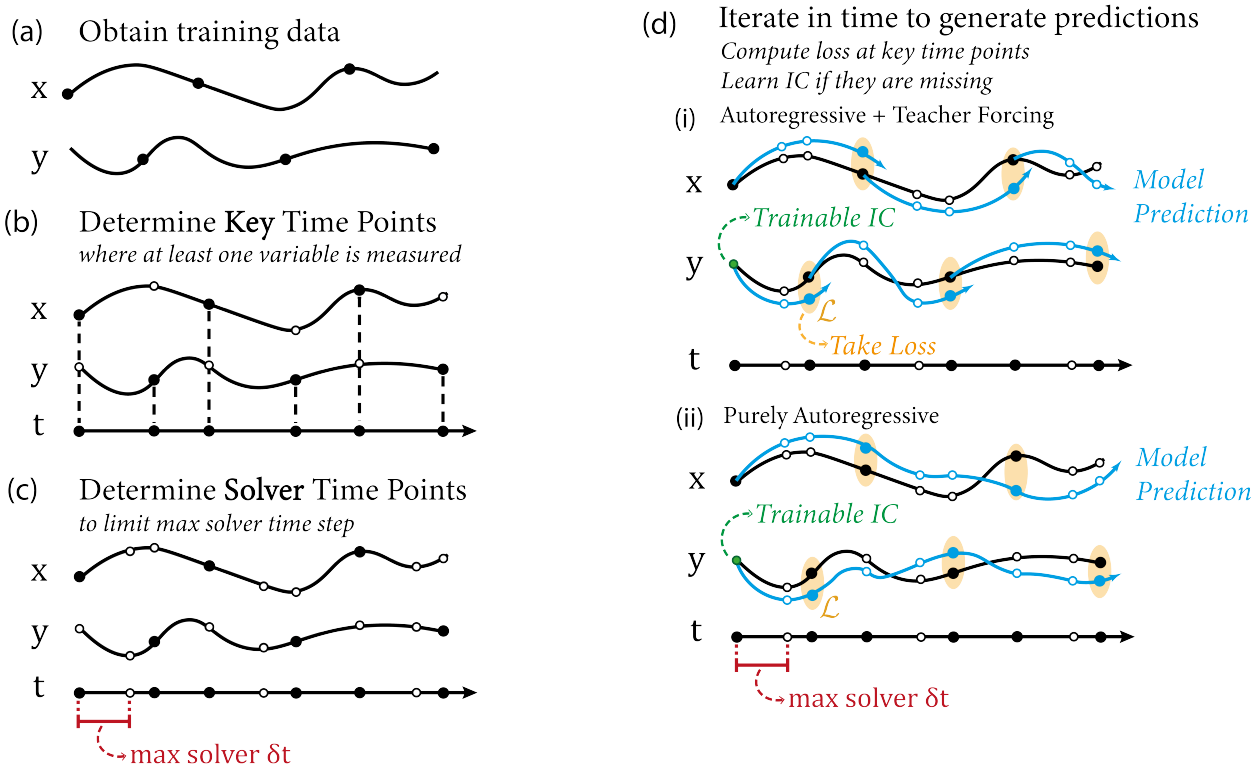


图 4：**训练算法与批处理。** (a) 系统地记录每条训练数据轨迹。(b) 确定每条轨迹的**关键**时间点；关键时间点是至少测量了一个变量的时间点，因此会在这些时间点计算损失。(c) 确定**求解器**时间点。这些时间点不计算损失，纳入它们是为了确保数值积分器的稳定性和精确度。(d) 通过沿时间方向积分遍历每条轨迹来生成预测；(i) 当某个特征有真实数据时，使用教师强制；若没有，则自回归地生成预测。本文在模型训练期间采用此方法；(ii) 模型预测完全以自回归方式生成，并从给定的初始条件开始。本文在推理期间采用此方法。训练时，在每条轨迹的关键时间点计算损失，通过反向传播计算梯度，并执行梯度下降来训练神经网络。

长度也可能各不相同。为了在同一个训练小批次中纳入多条轨迹，数据输入中会同时包含一系列评估时间点，并通过填充使所有数据序列和时间序列长度一致。这样便可以同时使用多个数据轨迹组成的小批次进行训练。

损失函数和误差指标。损失函数采用均方误差（MSE），计算模型预测轨迹与真实数据在有数据可用的时间点上的差异：

$$L = \frac{1}{N_{\text{data}}} \sum_{i \in \text{trajectories}} \sum_{j \in \text{times}_i} \sum_{k \in \text{channels}(i,j)} (\hat{y}_i^{(k)}(t_j) - y_i^{(k)}(t_j))^2. \tag{5a}$$

$$N_{\text{data}} = \sum_{i \in \text{trajectories}} \sum_{j \in \text{times}_i} ||\text{channels}(i,j)|| \tag{5b}$$

此处，channels(i, j) 表示在时间 j观测到的轨迹 i 中的通道索引集合。有关用于评估模型性能的指标的详情，请参阅 A.2 节。

3 结果

我们的第一个示例采用了经典化学工程文献中的非等温连续搅拌釜式反应器模型（ [56] URP连续搅拌釜式反应器模型）的模拟数据。该模型是一个二维常微分方程系统（式 Eq6），

$$\frac{dx_1}{dt} = -x_1 + \text{Da} \cdot (1 - x_1) \cdot \exp(x_2) = f_1(x_1, x_2) \tag{6a}$$

$$\frac{dx_2}{dt} = -x_2 + \text{B} \cdot \text{Da} \cdot (1 - x_1) \cdot \exp(x_2) - \beta \cdot x_2 = f_2(x_1, x_2) \tag{6b}$$

其中，x_1 为无量纲浓度（转化率），x_2 为无量纲温度，t 为无量纲时间。该系统由固定参数 $B = 11$ 和 $\beta = 3.0$ 以及一个可变参数 Da （Damköhler数）确定，该参数的采样范围为 Da $\in$ (0.2,0.5)。在此 B 和 β取值下，模型关于 Da 有两个 Hopf分岔点：分别位于 Da $= 0.280275$ 和 Da $= 0.419548$ 处（图 5）。在范围 $0.280275 \leq$ Da ≤ 0.419548内可观察到稳定周期振荡行为，而在此范围之外则只有一个稳定稳态。训练数据的示例轨迹见图 5。

这里的神经网络架构是一个多层感知器，其结构如下：(a) 包含三个神经元并采用 SiLU 激活函数的输入层；(b) 两个隐藏层，每层包含 64 个神经元并采用 SiLU 激活函数；以及 (c) 输出层，在黑箱表述中包含两个神经元，在灰箱表述中包含一个神经元，并采用线性激活函数。

在“黑箱”表述中，不假设系统具备任何先验知识，因此神经网络必须学习完整的动力学（公式 7）及其对 Da的依赖关系。

$$\hat{\mathbf{x}} = \mathcal{N}(\mathbf{x}, \text{Da}; \theta) \quad \text{where } \mathbf{x} = \{x_1, x_2\} \tag{7}$$

在灰箱表述中，系统动力学的部分内容先验已知，而其他部分未知（公式 8）。这里假设反应动力学未知，但总体质量与能量守恒定律的形式，以及反应速率与反应热生成（反应热，该量可独立测量）之间的耦合关系，均为已知。

$$\begin{aligned} \hat{\phi} &= \mathcal{N}(\mathbf{x}, Da; \theta) \\ \hat{x_1} &= -x_1 + \hat{\phi} \\ \hat{x_2} &= -x_2 + \text{B} \cdot \hat{\phi} - \beta \cdot x_2 \end{aligned} \tag{8a}$$

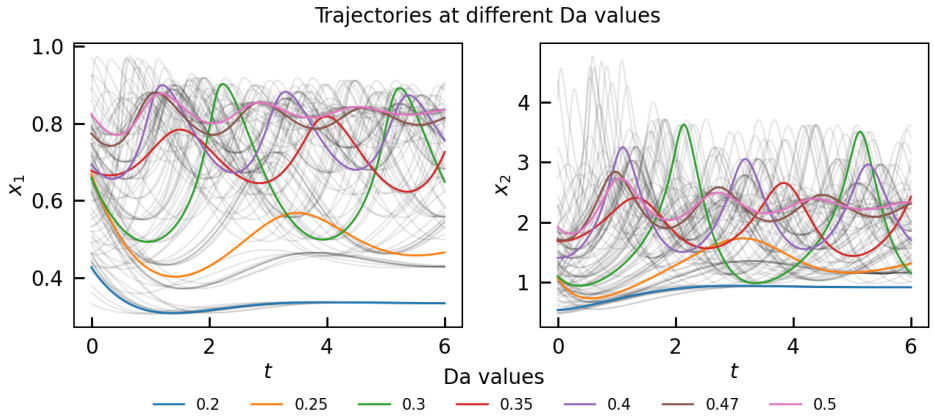
ˆ这里 ϕ(**x** Da) 是反应动力学的神经网络近似（公式 8b）。

$$\phi = f_\phi(\mathbf{x}, \text{Da}) = \text{Da} \cdot (1 - x_1) \cdot \exp(x_2) \tag{8b}$$

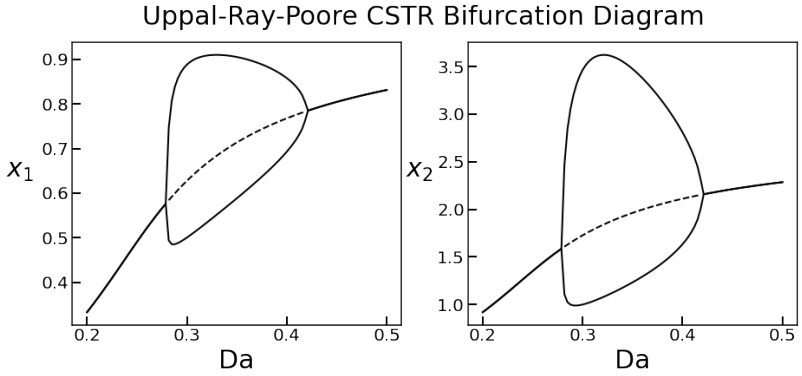
这里 B 和 β 是可独立测量的实验参数，假定先验 已知并“硬编码”在神经网络中；如果它们恰好不 为人所知，当然也可以将它们视为额外的 训练参数，与 Da和神经网络权重一起，在人工神经网络训练过程中进行拟合。

为简单起见，除非另有说明， $\mathcal{N}$ 表示经过训练以预测$\hat{\mathbf{x}}$的整体网络，所有灰盒方面均包含在这一表示中。

$$\hat{\mathbf{x}} = \mathcal{N}(\mathbf{x}, \text{Da}; \theta) \tag{8c}$$



（a）不同 Da值下的训练数据轨迹



(b) $Da = 0.280275$ 和 $Da = 0.419548$处的分岔点。实线：稳定稳态或 稳定振荡的最大值/最小值；虚线：不稳定稳态

图 5：URP CSTR 的训练数据可视化

Case	A	B	C	D	E	F	URP BB
数据 $< \Delta t >$	0.1	0.5	0.5	0.5	0.5（x_1） 0.55(x_2)	0.5	0.5
求解器 求解器最大值 δt	0.1	0.5	0.1	0.1	0.1	0.1	0.1
任意的 数据 Δt	-	-	-	✓	-	✓	✓
部分 观测值	-	-	-	-	✓	✓	✓
初始条件 给定值	✓	✓	✓	✓	✓	✓	-
解 误差（$\mathcal{L}_2$）	$(9.68 \pm 1.71) \times 10^{-3}$	$(1.49 \pm 0.12) \times 10^{-2}$	$(8.69 \pm 1.18) \times 10^{-3}$	$(4.23 \pm 0.43) \times 10^{-3}$	$(6.09 \pm 0.56) \times 10^{-3}$	$(8.91 \pm 1.65) \times 10^{-3}$	$(7.92 \pm 1.36) \times 10^{-3}$
RHS 误差（$\mathcal{L}_2$）	$(2.72 \pm 0.36) \times 10^{-3}$	$(9.71 \pm 0.76) \times 10^{-3}$	$(5.95 \pm 1.21) \times 10^{-3}$	$(3.50 \pm 0.34) \times 10^{-3}$	$(3.66 \pm 0.46) \times 10^{-3}$	$(7.12 \pm 0.81) \times 10^{-3}$	$(7.69 \pm 0.11) \times 10^{-3}$

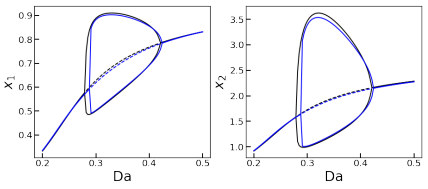
表1：URP CSTR 训练的指标汇总表。此表展示了模型在表示复杂度不断增加的数据上的模型性能。误差为10次训练运行的均值和标准差。有关误差指标的详细信息，请参阅 A.2 节。

在我们的演示中，我们逐步增加所使用训练数据的复杂性。所有模型结果总结于表1和表2中。案例A（图6a）是基准案例，其中训练数据轨迹在时间上进行密集采样，具有完整观测和规则时间采样。对于这个简单的基准案例，我们观察到黑箱训练网络能够很好地复现分岔图。案例B（图6b）和案例C（图6c）展示了控制求解器最大值的重要性 δt。在案例B中，对于使用嵌入式RK4积分器进行稳定且精确的积分而言，求解器最大值 $\delta t = 0.5$ 过大，而模型无法复现 δt附近的Hopf分岔点。

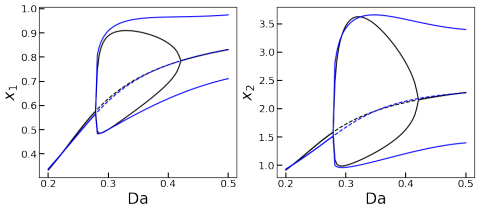
Case	可学习的动力学函数	可学习的实验值参数	解误差（$\mathcal{L}_2$）	RHS 误差（$\mathcal{L}_2$）	动力学函数误差（$\mathcal{L}_2$）	实验值参数误差（$\mathcal{L}_2$）
URP BB	-	-	$(7.92 \pm 1.36) \times 10^{-3}$	$(7.69 \pm 0.11) \times 10^{-3}$	-	-
URP GB1	✓	-	$(9.91 \pm 1.26) \times 10^{-3}$	$(4.43 \pm 0.51) \times 10^{-3}$	$(5.31 \pm 0.62) \times 10^{-3}$	-
URP GB2	✓	✓	$(1.22 \pm 0.17) \times 10^{-2}$	$(6.49 \pm 2.08) \times 10^{-3}$	$(7.77 \pm 2.25) \times 10^{-3}$	$(4.45 \pm 2.28) \times 10^{-2}$

表 2：URP CSTR 数据的指标汇总表。本表展示了黑箱模型和灰箱模型的模型性能。所有模型均使用任意数据 Δt、部分观测值，且未给定初始条件。误差为 10 次训练运行的均值和标准差。有关误差指标的详细信息，请参阅 A.2 节。

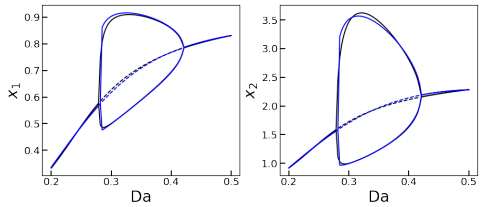
在嵌入式RK4积分器中，模型无法复现 $Da = 0.42$附近的Hopf分岔点。将案例C中的求解器最大值 δ 减小到0.1即可解决这一问题。在案例D和案例E中，我们分别引入任意时间采样和部分观测；案例F则同时存在这两种数据病态。在这三个案例中，我们都观察到良好的训练结果，解误差和右端项误差都很低。最后，在案例 URP BB（图 6d）中，我们令初始条件向量可训练，从而能够利用初始条件未给定或在第一个采样时间并未测量所有变量的数据进行训练。



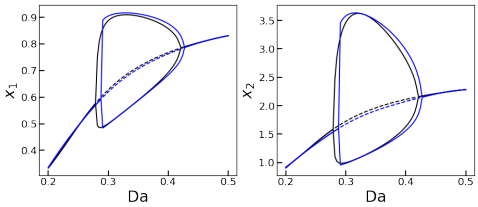
(a) 案例A：基准案例：理想数据（精细采样、完整观测、固定时间采样）



(b) 案例 B：无求解器时间步长控制的训练



(c) 案例C：带求解器时间步长控制的训练



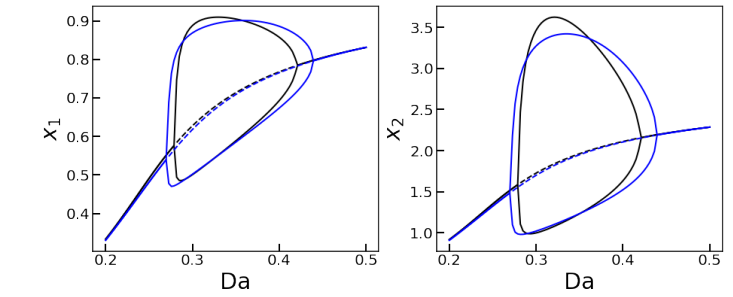
(d) 案例 URP_BB：最终模型（求解器时间步长控制、部分观测、可变时间步长采样、可训练初始条件

图 6：分岔图预测

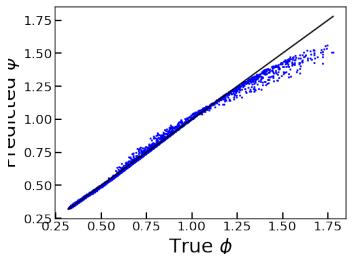
在案例 URP GB1 和 URP GB2 中（图 7、表 2），我们将物理信息“灰箱”引入模型。在这两个案例中，模型不是直接学习变量 x_1 和 x_2 的动态演化，而是学习动力学函数 f_ϕ；x_1 和 x_2 的演化则通过质量与能量衡算，以及反应动力学与能量学的耦合进行重构。在案例 URP GB1 中，耦合常数 B 和 β 是已知的；而在案例 URP GB2 中，它们是未知的，并作为模型的额外可训练参数引入。在这两个案例中，数据都存在部分观测、任意时间采样以及未给定初始条件的问题。在案例 URP GB2 中（图 7），我们观察到分岔图得到了良好复现，动力学函数 f_ϕ 得到了准确预测，并进一步转化为对 x_1 和 x_2 的准确右端项预测；同时，耦合常数 B 和 β 也得到了良好预测。

与“黑箱”相比，采用“灰箱”进行训练的一大优势是能够保留物理直觉；因此，我们期望它能够将模型外推到训练数据以外的参数空间。这一点在 URP GB1 案例（图 8）中得到了证明；该模型是在 B 和 β 的固定已知值下训练的，并利用神经网络学习动力学反应项。这样一来，模型便可外推至远超训练范围的 B 和 β 值，从而预测 B 和 β 的分岔图；尽管模型仅在一组 参数值下训练过，这些预测仍与真实值极为接近。

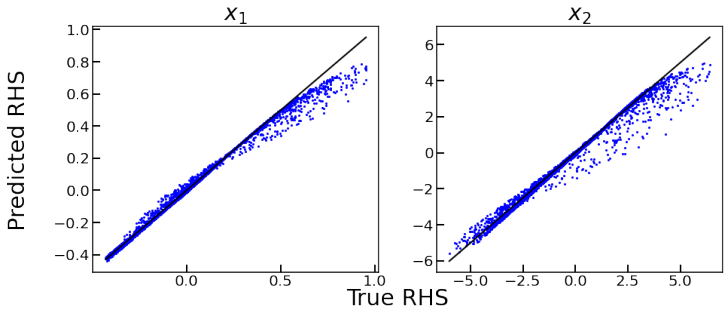
作为第二个说明性示例，我们使用了一个经典生化模型中的数据，该模型涉及三种微生物物种在共培养中的共存生长，并通过各种化学底物和辅因子 [3]（B&F 模型）相互耦合。这是一个六维耦合常微分方程系统，详见公式 9，参数值列于表 3 中。训练



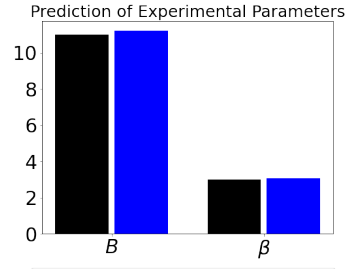
(a) 分岔图



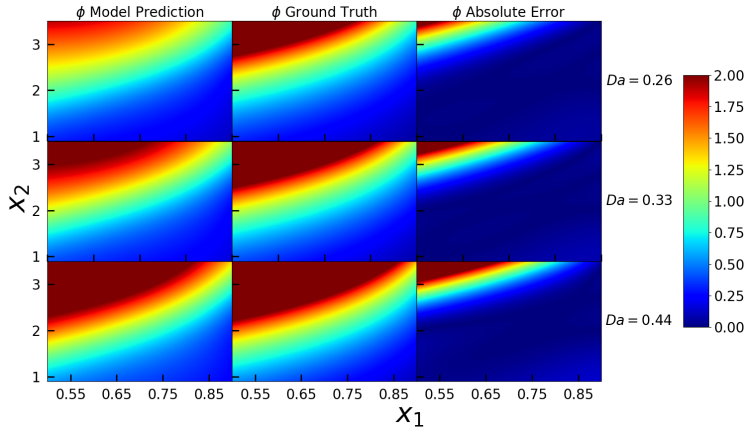
(b) 预测
动力学项



(c) 右端项的预测



(d) 预测
实验参数

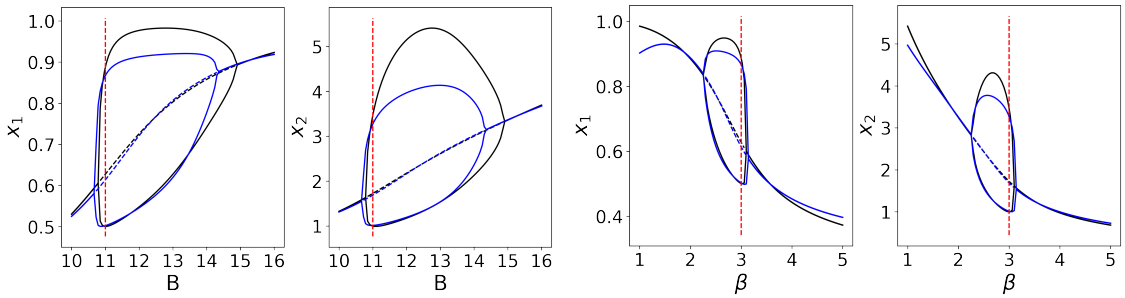


（e）对于各种 Da 值，学习得到的 ϕ 和预测的 ϕ 的热图

图 7：训练　得到的模型性能，案例 URP GB2（带有可训练实验参数的灰箱）

数据如图9所示。

$$
\begin{aligned}
\frac{dx}{dt} &= -\alpha x + \mu_1 x - \mu_{c_1} x && \text{Host Population} \\
\frac{dy}{dt} &= -\alpha y + \mu_2 y - \mu_{c_2} y && \text{Commensal Population 1} \\
\frac{dz}{dt} &= -\alpha z + \mu_3 z - \mu_{c_3} z && \text{Commensal Population 2} \\
\frac{du}{dt} &= \alpha(u_f - u) - \mu_1 x && \text{Host Substrate} \\
\frac{dv}{dt} &= -\alpha v + \omega\mu_1 x - \mu_2 y - \sigma\mu_3 z && \text{Growth Factor} \\
\frac{dg}{dt} &= -\alpha g + \rho\mu_2 y + \eta\mu_3 z && \text{Inhibition Factor,}
\end{aligned}
\tag{9a}
$$



(a) B 的分岔图　　(b) β 的分岔图

图 8：URP GB1 案例（灰箱模型，实验参数固定且已知）：外推至新的参数值。红色虚线表示训练数据的参数值；其他所有参数空间均为超出训练数据范围的外推区域。

其中

$$\begin{aligned}
\mu_1 = f_{\mu_1}(u,g) &= \frac{u}{(1+u)(1+g)} && \text{Host Population Growth Rate} \\
\mu_2 = f_{\mu_2}(v) &= \frac{\phi_1 v}{1+v} && \text{Commensal Population 1 Growth Rate} \\
\mu_3 = f_{\mu_3}(v) &= \frac{\phi_2 v}{\sigma+v} && \text{Commensal Population 2 Growth Rate.}
\end{aligned} \tag{9b}$$

宿主 x 以底物 u 为养分生长，并分泌生长因子 v。共生种群 y 和 z 消耗生长因子v，并释放抑制因子 g，该因子会抑制宿主 x的生长。 f_{μ_i} 是各物种的生长速率函数， μ_{ci} 是一个常数参数，用于规定各生物量物种的维持速率。

参数	值
α	$1/7.3$
u_f	250.0
ω	9.7
σ	10.0
ρ	0.13138686
η	1.29166
ϕ_1	0.2941176
ϕ_2	0.367647
μ_{c_1}	0.367647
μ_{c_2}	0.117647
μ_{c_3}	0.1617647

表 3：Baltzis 与 Frederickson（B&F）模型的参数值

神经网络架构为多层感知器，包括：(a) 一个具有 6 个神经元并采用 SiLU 激活函数的输入层，(b) 两个隐藏层，每层各有 32 个神经元并采用 SiLU 激活函数，以及 (c) 一个具有 6 个神经元（黑箱表述）或 3 个神经元（灰箱表述）并采用线性激活函数的输出层。在“黑箱”表述中，假定对系统没有先验知识，因此神经网络必须学习整个系统（方程 10）

$$\hat{\dot{\mathbf{x}}} = \mathcal{N}(\mathbf{x};\theta) \quad \text{where } \mathbf{x} = \{x, y, z, u, v, g\}. \tag{10}$$

在灰箱表述中，系统动力学的某些部分是先验已知的；其他部分则未知（公式 11）。这里假设微生物生长速率未知，但总体质量和能量守恒定律，以及生长速率与底物、促生长因子/抑制因子的生成和利用之间的耦合关系是已知的：

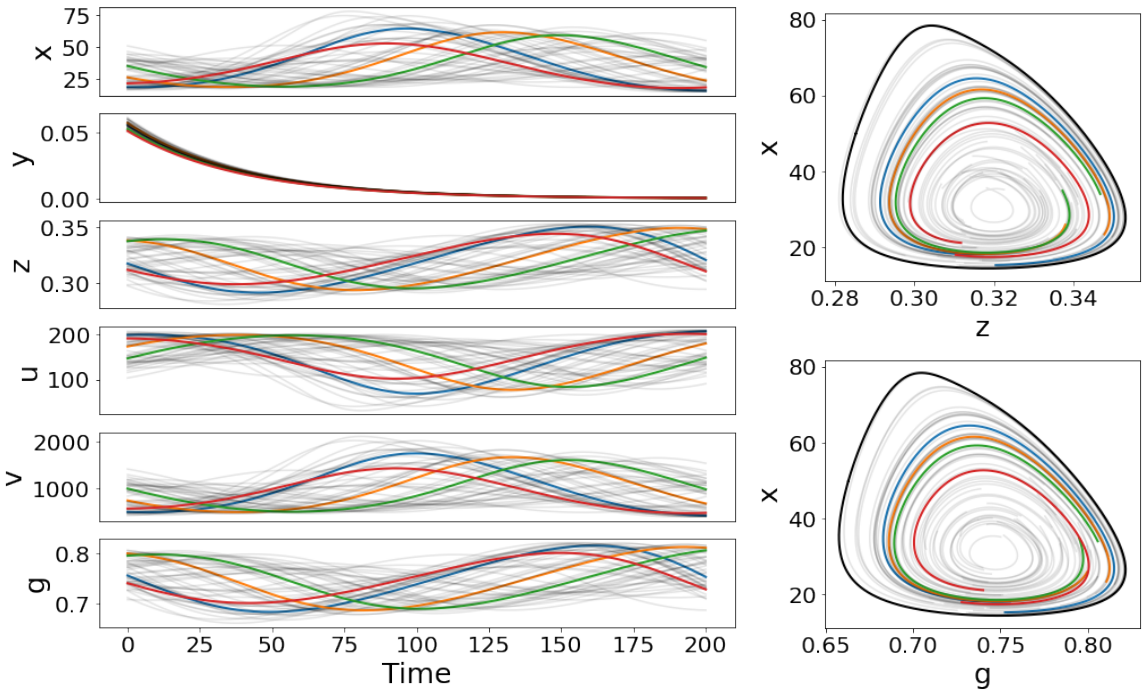


图9：B&F训练数据的瞬态过程和极限环（200条轨迹，其中4条重点标出）。黑色曲线表示给定的稳定极限环。

以及生长速率与底物的耦合，以及生长/抑制因子的生成和利用均已知：

$$
\begin{aligned}
\{\hat{\mu}_{1,x}, \hat{\mu}_{2,y}, \hat{\mu}_{3,z}\} &= \mathcal{N}(\mathbf{x}; \theta) \\
\hat{\dot{x}} &= -\alpha x + \hat{\mu}_{1,x} - \mu_{c_1} x \\
\hat{\dot{y}} &= -\alpha y + \hat{\mu}_{2,y} - \mu_{c_2} y \\
\hat{\dot{z}} &= -\alpha z + \hat{\mu}_{3,z} - \mu_{c_3} z \\
\hat{\dot{u}} &= \alpha(u_f - u) - \hat{\mu}_{1,x} \\
\hat{\dot{v}} &= -\alpha v + \omega \hat{\mu}_{1,x} - \hat{\mu}_{2,y} - \sigma \hat{\mu}_{3,z} \\
\hat{\dot{g}} &= -\alpha g + \rho \hat{\mu}_{2,y} + \eta \hat{\mu}_{3,z}
\end{aligned}
\tag{11a}
$$

其中，$\{\hat{\mu}_{1,x}, \hat{\mu}_{2,y}, \hat{\mu}_{3,z}\}$ 是神经网络对微生物生长速率（公式 9b）的近似，并乘以相应微生物的生物量浓度；$\{\omega, \sigma, \rho, \eta\}$ 是实验参数，这些参数要么是先验已知的，要么作为额外的 训练参数纳入，在人工神经网络训练过程中拟合。为简化表达，除非另有说明，令 $\mathcal{N}$ 表示用于预测 $\hat{\dot{\mathbf{x}}}$ 的整体网络，所有灰盒方面均包含在此表示中。

$$
\hat{\dot{\mathbf{x}}} = \mathcal{N}(\mathbf{x}; \theta) \tag{11b}
$$

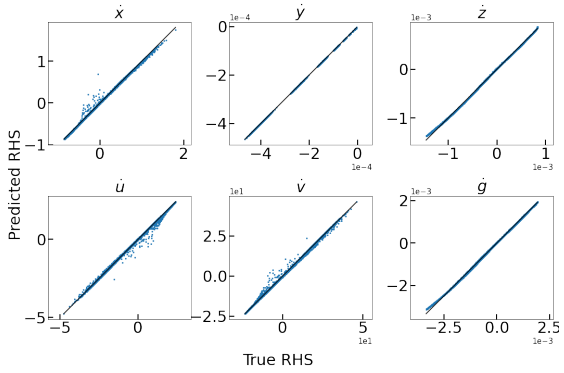
现在我们展示　　通过 URP连续搅拌釜式反应器模型开发并阐释的训练方法，应用于 th　　e

Case	可学习的动力学函数	可学习的 Exp. 参数	解 误差来源 真实 LC ($\mathcal{L}_2$)	RHS 误差 ($\mathcal{L}_2$)	动力学函数 误差 ($\mathcal{L}_2$)	Exp. 参数 误差 ($\mathcal{L}_2$)
BF BB	✗	✗	$(3.25 \pm 0.43) \times 10^{-3}$	$(2.12 \pm 0.45) \times 10^{-3}$	-	-
BF GB1	✓	✗	$(5.16 \pm 0.68) \times 10^{-3}$	$(5.43 \pm 0.80) \times 10^{-3}$	$(7.66 \pm 1.14) \times 10^{-4}$	-
BF GB2	✓	✓	$(6.10 \pm 2.72) \times 10^{-3}$	$(7.50 \pm 3.2) \times 10^{-3}$	$(1.09 \pm 0.49) \times 10^{-3}$	$(7.51 \pm 2.1) \times 10^{-2}$

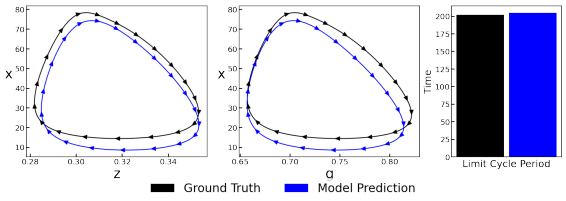
表 4：B&F 数据的指标汇总表。本表展示了黑箱模型和灰箱模型的模型性能。所有模型均使用任意数据 Δt，仅有部分观测，且未给定初始条件。有关指标的详情，请参阅 A.2 节。

B&F 模型；度量　　初始条件汇总于表 4 中。所有案例都存在任意时间采样这一数据缺陷　　，

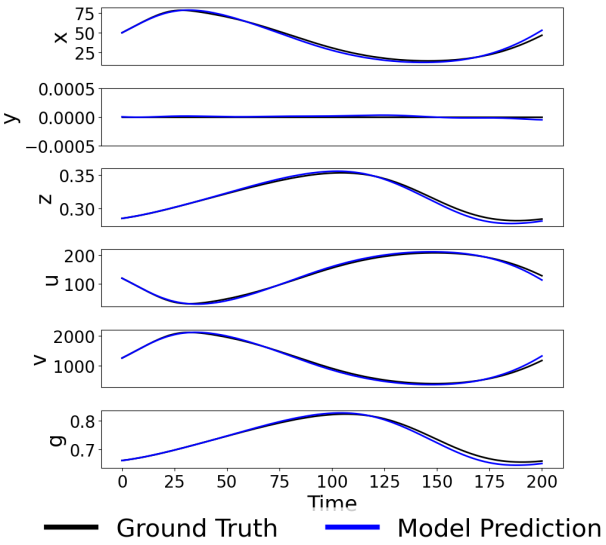
部分观测值且未给定初始条件。案例 BF BB（图 10）是黑箱案例，我们观察到对全部 6 个变量的右端项均能进行良好预测，同时还能准确重现稳态极限环。在案例中



（a）右端项预测



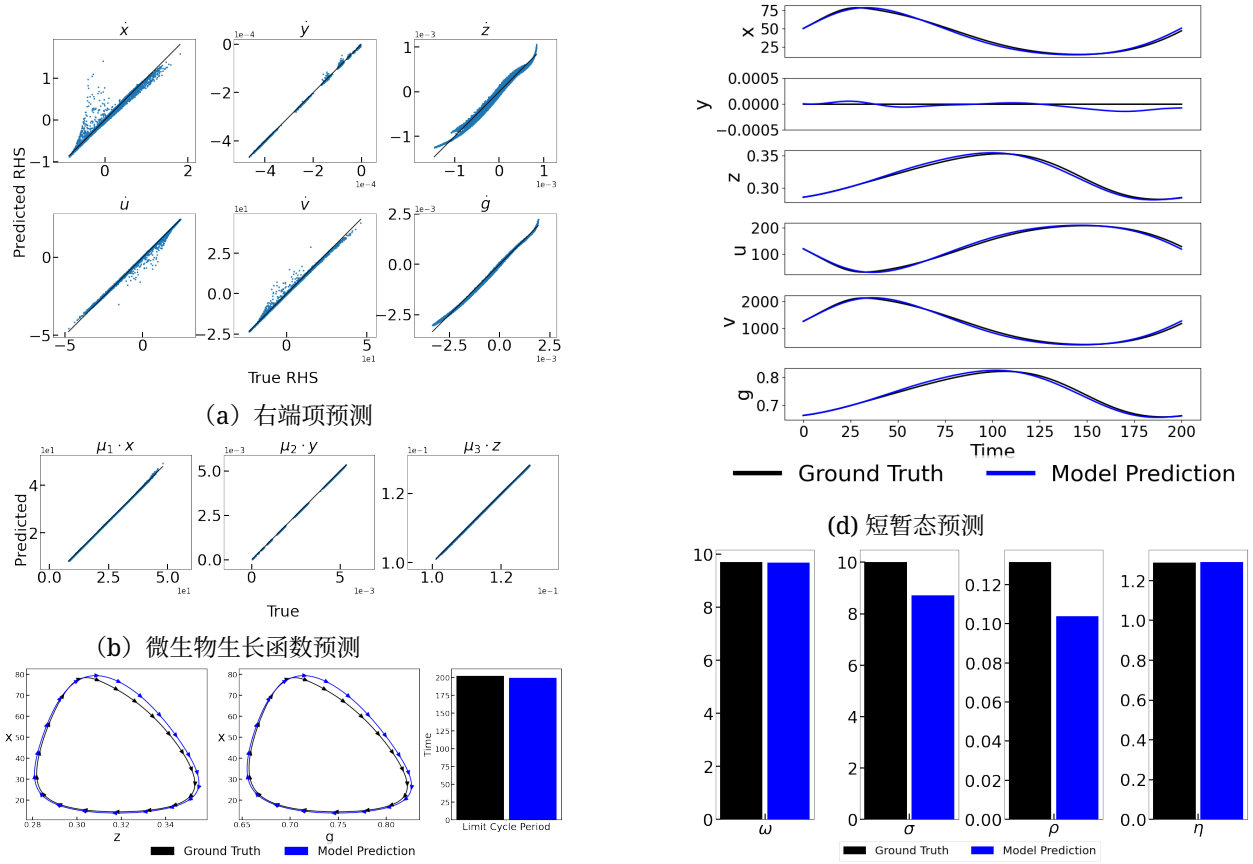
（b）极限环预测



（c）短暂态预测

图 10：BF BB 案例的训练模型性能

在案例 BF GB1 和案例 BF GB2（图 11）中，我们引入了物理信息灰箱，其中模型学习物种 x的 y、 z 微生物生长函数，并使用带有耦合参数 ω、σ、ρ、η 的质量衡算来规定全部 6 个变量的整体演化。在案例 BF GB1 中，这些耦合参数是已知的；而在案例 BF GB2 中，它们是未知的，并作为附加参数引入，在模型训练期间进行学习。对于案例 BF GB2（图 11），模型能够准确预测微生物生长速率和耦合参数；这使得对全部 6 个变量右端项的预测十分准确，并且能够良好地再现极限环。



（a）右端项预测

（b）微生物生长函数预测

（c）极限环预测

(d) 短暂态预测

(e) 实验参数预测

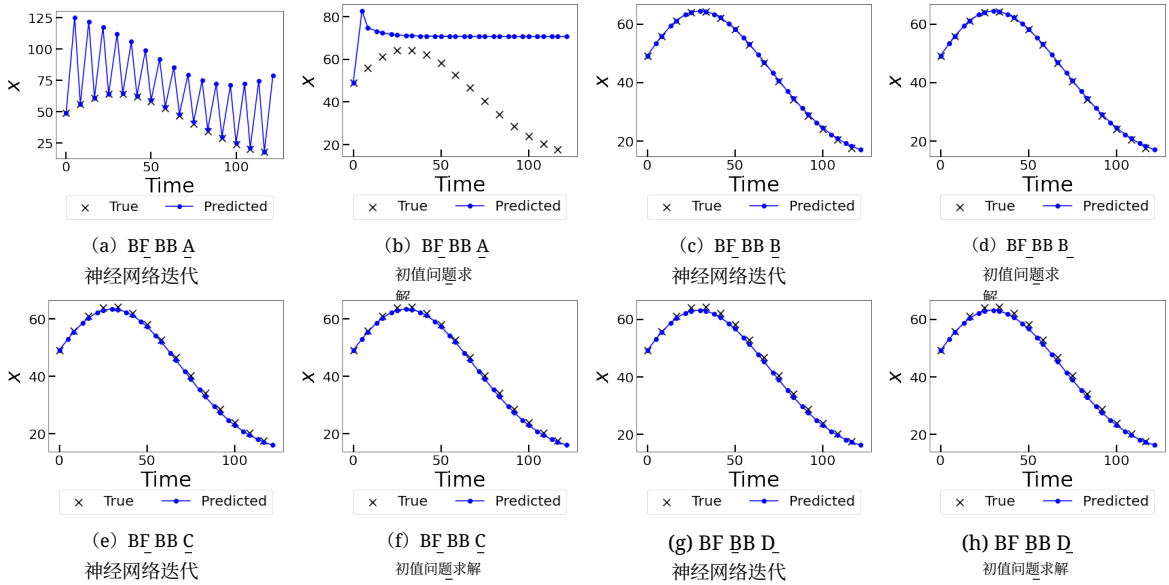
图 11：案例 BF GB2 的训练模型性能

4 讨论与结论

在这项工作的过程中，我们观察到模型训练中的一些异常现象，认为它们非常值得报告和讨论。

“共振”效应。我们观察到，在处理固定速率时间采样数据时，模型权重的初始化方式可能会导致训练成功拟合数据，却收敛到一个“明显错误”的局部最小值。模型没有学习底层常微分方程系统的右端项，而是学到了一个“共振型”单位时间映射：它不再以平滑曲线插值数据点，而是通过两个斜率迥异的积分步（图 12 (a)、(b)），从每个数据点推进到下一个数据点；这可以描述为积分步与数据采样步之间的 2:1 共振。

在上面的示例中，数据每隔 $\Delta t = 8.333$ 采样一次。对于 BF BB A 案例，Δt 被 RK4 数值积分器一致地划分为长度为（$5 + 3.333$）的步长。使用“糟糕的”（随机）权重初始化时，模型训练会“硬编码”这个时间步长，实质上学到的是单位时间映射，而没有显式学习常微分方程的右端项；当以训练期间相同的时间步长将数据输入网络进行迭代时，我们会观察到一种典型的锯齿状“共振”风格输出。我们探索了各种缓解技术，以防止收敛到这样的局部最小值。一种方法是尝试更好地初始化模型权重（此处仅将输出层的权重缩小 100 倍（BF BB B 案例），其他层的初始化保持不变）。其次，我们随机化求解器的 δt 步长，使得 $\Delta t = 8.333$不是始终采用长度为（$5+3.333$）的步长，而是被划分成长度随机的时间“块”，且每块长度均小于最大 δt 5（BF BB C 案例）。第三，我们对数据采样本身进行随机化，使用均值为 $< \Delta t = 8.333 >$的伽马分布，这样求解器的δt 值也自然实现了随机化。这三种方法都使模型得以摆脱糟糕的局部最小值，甚至完全消除了它。



（a）BF_BB_A 神经网络迭代　（b）BF_BB_A 初值问题求解　（c）BF_BB_B 神经网络迭代　（d）BF_BB_B 初值问题求解

（e）BF_BB_C 神经网络迭代　（f）BF_BB_C 初值问题求解　(g) BF_BB_D 神经网络迭代　(h) BF_BB_D 初值问题求解

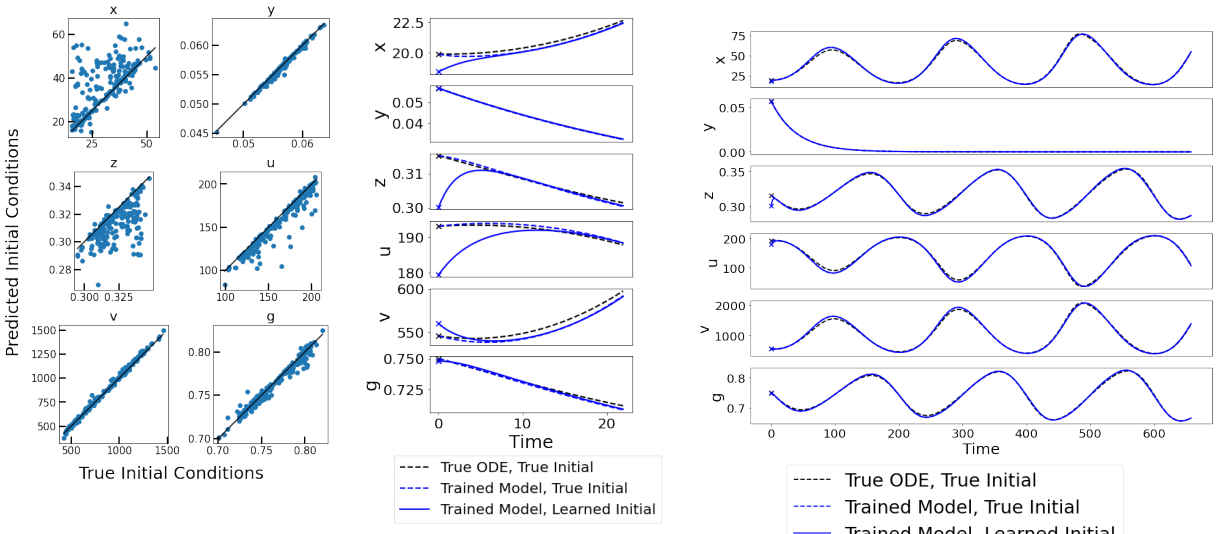
图 12：每个案例中最佳模型在一条具有代表性的训练轨迹上的性能。图中分别展示了：(a) 递归神经网络迭代，其采用 BF BB A 训练中使用的时间步长；以及 (b) scipy.integrate.solve ivp 求解器，该求解器使用可变时间步进（因此有助于更好地测试对所学习右端项进行积分的准确性）。

Case	初始缩放	随机化 Data Δt	随机化求解器 δt	解误差来源 真实 LC ($\mathcal{L}_2$)	RHS 误差 ($\mathcal{L}_2$)
BF_BB_A	$\times 100^{0}$	✗	✗	$(4.33 \pm 2.58) \times 10^{-1}$	8.47 ± 1.63
BF_BB_B	$\times 100^{-1}$	✗	✗	$(4.13 \pm 0.53) \times 10^{-3}$	$(2.89 \pm 0.92) \times 10^{-3}$
BF_BB_C	$\times 100^{0}$	✗	✓	$(8.40 \pm 3.52) \times 10^{-2}$	1.65 ± 0.68
BF_BB_D	$\times 100^{0}$	✓	✓（因为的数据）	$(1.14 \pm 0.17) \times 10^{-2}$	$(2.88 \pm 0.68) \times 10^{-2}$
BF_BB_E	$\times 100^{-1}$	✓	✓（因为的数据）	$(4.18 \pm 0.71) \times 10^{-3}$	$(3.10 \pm 0.56) \times 10^{-3}$

表5：指标总和 B&F 数据汇总表。此表展示了共振效应。请参阅 A.2 节以了解 r 有关指标的详细信息。

初始条件的学习。我们观察到，如果初始条件不属于训练集且需要推断，模型就**无法**始终正确地推断出初始条件（图 13）。考虑多时间尺度奇异摄动动力学问题（快慢系统、时间尺度分离较大的系统）中发生的情况，可以解释这一现象（并加以缓解）（图 14，式 12）。在这类问题中（我们的 6 个常微分方程的线性化特征值相差一个或多个数量级（表 6）），初始条件会迅速（在快时间尺度上）收敛到“慢流形”，长期动力学则在慢时间尺度上沿其演化。这与化学动力学中著名的准稳态近似（QSSA）或 Bodenstein近似的论述相同。这意味着，无穷多个初始条件会迅速（在快时间尺度上）到达慢流形上的（实际上）同一状态值，经历这段短暂的初始积分阶段后，便停留在慢流形上并沿其演化 [16, 58, 1, 63]。

在相关文献中 [2], , 这一问题通过寻找一个已经大致位于慢流形上的初始条件来缓解：假设我们关注的是位于慢流形上、在所需时间点 Δt 之前的某个初始条件 Δt ，而该时间点早于我们的第一个采样数据点。“技巧”在于构造这样一个“成熟”或“培育”的初始条件：首先估计一个更早的初始条件（例如 10 Δt 更早，此时它可以位于慢流形的快纤维丛上的任意位置），然后让其演化 $9\Delta t$。这样，较早初始条件中的“坏”成分就能“成熟”（“培育”），从而在所需时间点 Δt 之前，为我们提供一个位于慢流形上的初始条件的良好估计。



（a）初始条件预测　（b）瞬态（放大）　（c）瞬态

图 13：案例 BF GB2 的初始条件预测。尽管学习到的初始条件并不正确，但从学习到的初始条件（实蓝线）开始积分的训练好的人工神经网络，会迅速与从真实初始条件（虚蓝线）开始积分的轨迹相吻合；在长期动力学中，两者实际上难以区分。这是由于初始启动过程中的时间尺度分离（快-慢特性）所致。

$$\begin{aligned} \dot{x} &= f(x, y) \\ \varepsilon \dot{y} &= g(x, y) \end{aligned} \tag{12}$$

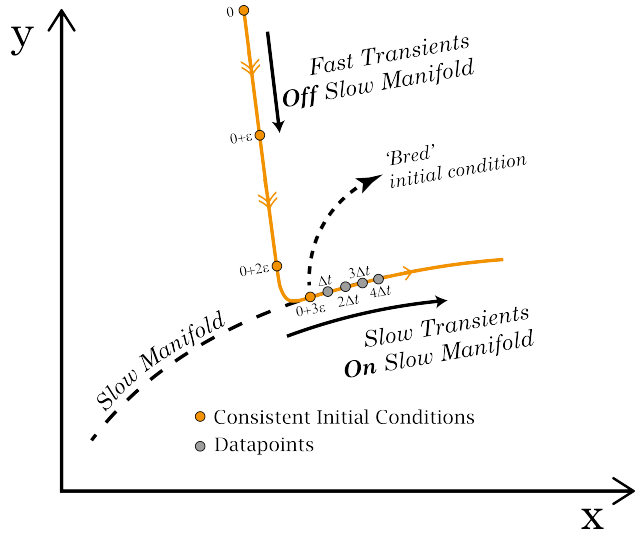


图 14：如奇异摄动问题（方程 12）所示，快、慢时间尺度的分离可能导致多个初始条件与同一个首次测量数据点实际上一致，因为这些初始条件会先在快流形上快速演化，然后收敛到慢流形上。理想情况下，我们希望初始条件经过“培育”直到位于慢流形上，因为这是“最有可能”的候选条件，但这很难施加。

总结与展望。在本文中，我们提出了一种基于神经网络的辨识架构，该架构受传统数值分析启发并以其为基础，用于处理一系列时间序列数据病态问题：不均匀采样率、缺失数据（包括初始条件），以及未知或部分已知物理学。

特征值	真实常微分方程 真实初始条件	训练后的人工神经网络 真实初始条件	训练后的人工神经网络 学习得到的初始条件
1	$-0.20506 - 0.020221j$	-0.51397	-0.51752
2	$-0.20506 + 0.020221j$	-0.44970	-0.44258
3	-0.20145	-0.20145	-0.20145
4	-0.025494	-0.026535	-0.026341
5	$0.002033 - 0.03847j$	$-0.000690 - 0.03661j$	$-0.004094 - 0.03534j$
6	$0.002033 + 0.03847j$	$-0.000690 + 0.03661j$	$-0.004094 + 0.03534j$

表 6：真实常微分方程与训练模型在真实初始条件和学习得到的初始条件下的雅可比矩阵特征值。与主导特征值（红色）相关的快速动力学会在 $t = 2 - 2.3$ 的时间尺度内迅速衰减，并快速收敛到与蓝色特征值相关且与真实系统一致的较慢流形。所使用的训练数据以平均 $\Delta t = 8$ 进行采样，因此在模型训练期间无法学习（或未鼓励学习）更快的动力学。

我们将系统辨识过程表述为学习常微分方程系统右端项的过程，该过程可以与数值积分方法相结合。在每次前向传递中，我们沿时间方向遍历每条训练轨迹：在有真实数据时使用教师强制，在没有真实数据时使用自回归迭代，从而支持基于部分观测进行训练。我们展示了如何将这一架构与白盒先验知识相结合，使神经网络学习具有物理可解释性的函数，例如微生物生长速率或化学反应速率，以及实验参数；全局定律，例如反应速率与反应热之间的耦合，或微生物生长速率、底物消耗与产物形成之间的关系，都可以轻松地系统性“硬编码”到该架构中。

我们预计，这种方法（尤其是其“灰盒”部分）将使其适用于工业应用（例如 CHO细胞培养生物制造，这正是我们目前正在探索的方向）。在技术和计算方面，一个自然而然的后续任务是高效且易于使用地实现并行批处理；更广泛地说，则是持续利用并行化来加速计算。我们还预计将致力于为该方法引入自适应机制，其思路类似于传统初值求解器中的自适应时间步长选择。将这项工作与不确定性量化结合起来是一项重要的研究方向，也是当前应用机器学习研究的课题，无论是针对参数不确定性还是预测不确定性。值得注意的是，已识别的动力学右端项（称为逆修正常微分方程 或 IMDEs）与真实动力学之间的误差——即所谓的逆修正误差分析——似乎是数值分析中一个新兴的分支，正是由我们在此呈现的这类工作催生的 [65, 64]

作者贡献 **s.** IGK负责研究规划、研究监督并撰写了稿件。SM、TB和TQ执行了 d研究并撰写了稿件 稿件。JLA和MB参与了模型相关问题的研究及稿件撰写。

资助致谢。 IGK 和 TB 的工作得到美国能源部（DOE）（资助编号 SA22-0052-S001）的部分支持；SM 和 JLA 得到 DOE（资助编号 DE- SC0019363）的支持。IGK 和 TB 的工作还得到美国空军科学研究办公室（AFOSR）（资助编号 FA9550-21-0317）的部分支持；此外，AMBIC 也为 MB 的工作提供了部分支持。SM 还获得了新加坡科学、技术与研究局（A*STAR）提供的国家科学奖学金（NSS）的支持。

利益冲突。 作者声明不存在利益冲突。

参考文献

[1] Zagaris Antonios、Christophe Vandekerckhove、C. William Gear、Tasso J. Kaper 和 Loannis G. Kevrekidis。约束运行方案稳定性及其稳定化：通过无方程投影到慢流形。Discrete and Continuous Dynamical Systems，32(8)：2759–2803，2012年。[2] Magdalena A. Balmaseda、Oscar J. Alves、Alberto Arribas、Toshiyuki Awaji、David W. Behringer、Nicolas Ferry、Yosuke Fujii、Tong Lee、Michele Rienecker、Tony Rosati 和 Detlef Stammer。季节性预报的海洋初始化。Oceanography，22(SPL.ISS. 3)：154–159，2009年。[3] B. C. Baltzis 和 A Frederickson。在非捕食者-猎物系统中竞争可再生资源的两个微生物种群的共存。Bulletin of Mathematical Biology，46(1)：155–174，1984年1月。[4] Nikzad Benny Toomarian 和 Jacob Barhen。利用伴随函数和教师强制学习轨迹。Neural Networks，5(3)：473–484，1992年1月。[5] Tom Bertalan、Felix Dietrich、Igor Mezi´c 和 Ioannis G. Kevrekidis。从数据中学习哈密顿系统。Chaos，29(12)，2019年12月。[6] Dimitris Bertsimas、Agni Orfanoudaki 和 Colin Pawlowski。时间序列中临床协变量的插补。Machine Learning，110(1)：185–248，2021年1月。[7] Jake Bouvrie 和 Boumediene Hamzi。用于近似非线性系统的核方法。SIAM Journal on Control and Optimization，55(4)：2460–2492，2017年8月。[8] Mona Buisson-Fenet、Valery Morgenthaler、Sebastian Trimpe 和 Florent Di Meglio。利用部分观测和神经常微分方程学习动力学的辨识模型。技术报告，2022年。[9] L. Chen、O. Bernard、G. Bastin 和 P. Angelov。利用神经网络对生物技术过程进行混合建模。Control Engineering Practice，8(7)：821–827，2000年7月。[10] Ricky T.Q. Chen。torchdiffeq，2018年。[11] Ricky T.Q. Chen、Yulia Rubanova、Jesse Bettencourt 和 David Duvenaud。神经常微分方程。技术报告，2018年。[12] S. Chen、S. A. Billings 和 P. M. Grant。利用神经网络进行非线性系统辨识。International Journal of Control，51(6)：1191–1214，1990年。[13] Richard D. De Veaux、Rod Bain 和 Lyle H. Ungar。用于环境过程控制的混合神经网络模型：（1998 年 Hunter 讲座）。Environmetrics，10(3)：225–236，1999年5月。[14] Felix Dietrich、Alexei Makeev、George Kevrekidis、Nikolaos Evangelou、Tom Bertalan、Sebastian Reich 和 Ioannis G. Kevrekidis。从微观模拟中学习有效的随机微分方程：将随机数值方法与深度学习相联系。技术报告，2021年。[15]Robert M Farber、Alan S Lapedes、Ramiro Rico-Mart´ınez 和 Ioannis G Kevrekidis。连续时间动力系统的辨识：基于神经网络的算法及并行实现，1993年6月。[16] C. W. Gear、T. J. Kaper、I. G. Kevrekidis 和 A. Zagaris。投影到慢流形：奇异摄动系统与遗留代码。SIAM Journal on Applied Dynamical Systems，4(3)：711–732，2005年8月。[17] Andrew Gelman 和 Jennifer Hill。使用回归和多层次/层次模型进行数据分析。剑桥大学出版社，2006年12月。[18] R. Gonz´alez-Garc´ıa、R. Rico-Mart´ınez 和 I. G. Kevrekidis。分布参数系统的辨识：一种基于神经网络的方法。技术报告 SUPPL.1，1998年。[19] N. J. Gordon、D. J. Salmond 和 A. F.M. Smith。非线性/非高斯贝叶斯状态估计的新方法。IEE Proceedings，Part F：Radar and Signal Processing，140(2)：107–113，1993年。[20] Sam Greydanus、Misko Dzamba 和 Jason Yosinski。哈密顿神经网络。技术报告，2019年。[21] Tobias Hagge、Panos Stinis、Enoch Yeung 和 Alexandre M. Tartakovsky。将具有未知本构关系的微分方程求解转化为循环神经网络。2017年10月。[22] Boumediene Hamzi 和 Houman Owhadi。从数据中学习动力系统：一种简单的交叉验证视角，第一部分：参数化核流。Physica D: Nonlinear Phenomena，421：132817，2021年7月。[23] James Honaker 和 Gary King。如何处理时间序列横截面数据中的缺失值。American Journal of Political Science，54(2)：561–581，2010年4月。[24] J. L. Hudson、M. Kube、R. A. Adomaitis、I. G. Kevrekidis、A. S. Lapedes 和 R. M. Farber。非线性信号处理与系统辨识：其在电化学反应时间序列中的应用。Chemical Engineering Science，45(8)：2075–2081，1990年1月。

[25] R. E. Kalman。线性滤波与预测问题的新方法。《流体工程学杂志》、*ASME*汇刊，82(1):35–45，3 1960年。[26] Felix P. Kemeth、Sergio Alonso、Blas Echebarria、Ted Moldenhawer、Carsten Beta 和 Ioannis G. Kevrekidis。振幅方程的黑箱与灰箱学习：相场系统中的应用。技术报告，2022年。[27] Felix P. Kemeth、Tom Bertalan、Nikolaos Evangelou、Tianqi Cui、Saurabh Malani 和 Ioannis G. Kevrekidis。通过流形学习初始化LSTM内部状态。*Chaos*，31(9):093111，9 2021年。[28] Patrick Kidger。关于神经微分方程。2 2022年。[29] Patrick Kidger、James Morrill、James Foster 和 Terry Lyons。面向不规则时间序列的神经控制微分方程。技术报告，2020年。[30] Aditi S. Krishnapriyan、Alejandro F. Queiruga、N. Benjamin Erichson 和 Michael W. Mahoney。为连续物理学习连续模型。2 2022年。[31] John G. Kuschewski、Stanislaw H. ˙Zak 和 Stefen Hui。前馈神经网络在动态系统辨识与控制中的应用。*IEEETransactions on Control Systems Technology*，1(1):37–49，1993年。[32] Seungjoon Lee、Yorgos M. Psarellis、Constantinos I. Siettos 和 Ioannis G. Kevrekidis。基于主体的蒙特卡罗模拟数据学习黑箱与灰箱趋化偏微分方程/闭合关系。技术报告，2022年。[33] Lennart Ljung。扩展卡尔曼滤波器作为线性系统参数估计器的渐近行为。*IEEE Transactions on Automatic Control*，24(1):36–50，1979年。[34] Robert J. Lovelett、Jos´e L. Avalos 和 Ioannis G. Kevrekidis。部分观测与守恒定律：生物技术与光遗传学中的灰箱建模。*Industrial and Engineering ChemistryResearch*，59(6):2611–2620，2020年。[35] 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。*Nature Machine Intelligence*，3(3):218–229，10 2021年。[36] Gideon A. Lyngdoh、Mohd Zaki、N. M.Anoop Krishnan 和 Sumanta Das。通过缺失数据插补和可解释机器学习实现混凝土强度预测。*Cement and ConcreteComposites*，128:104414，4 2022年。[37] Philipp Menesklou、Tabea Sinn、Hermann Nirschl 和 Marco Gleiss。通过将数值过程模型与神经网络耦合，对卧螺离心机进行灰箱建模。*Minerals*，11(7):755，7 2021年。[38] Oliver Nelles。非线性系统辨识：从经典方法到神经网络和模糊模型。2001年。[39] Fulufhelo V. Nelwamondo、Shakir Mohamed 和 Tshilidzi Marwala。缺失数据：神经网络与期望最大化技术的比较。*Current Science*，93(11):1514–1521，2007年。[40] R. Oliveira。结合第一性原理建模与人工神经网络：通用框架。载于*Computers and Chemical Engineering*，第28卷，第755–766页。Pergamon，5 2004年。[41] Luke Oluwaseye、Joel * A1、Wesley Doorsamy 和 Sena Paul。机器学习缺失数据处理技术综述。*International Journalof Innovative Technologyand Interdisciplinary Scienceswww.IJITIS.org*，5(3):971–1005，9 2022年。[42] Fernando J. Pineda。神经计算的动力学与架构。*Journal of Complexity*，4(3):216–245，9 1988年。[43] Yorgos M. Psarellis、Seungjoon Lee、Tapomoy Bhattacharjee、Sujit S. Datta、Juan M. Bello-Rivas 和 Ioannis G. Kevrekidis。通过机器学习数据驱动发现细菌趋化迁移。8 2022年。[44] Dimitris C. Psichogios 和 Lyle H. Ungar。用于过程建模的神经网络与第一性原理混合方法。*AIChEJournal*，38(10):1499–1511，10 1992年。[45] S. Purwar、I. N. Kar 和 A. N. Jha。使用神经网络进行非线性系统辨识，2007年。[46] 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与逆问题的深度学习框架。*Journal ofComputational Physics*，378:686–707，2019年。[47] Maziar Raissi、Paris Perdikaris 和 George Em Karniadakis。物理信息深度学习（第二部分）：非线性偏微分方程的数据驱动发现。11 2017年。

[48] R. Rico-Martinez、J. S. Anderson 和 I. G. Kevrekidis。连续时间非线性信号处理：一种用于灰箱辨识的基于神经网络的方法。载于*Neural Networks for Signal Processing - Proceedings oftheIEEE Workshop*，第596–605页。IEEE，1994年。[49] R. Rico-Martinez、K. Krischer、M. C. Kube、J. L. Hudson 和 I. G. Kevrekidis。铜电溶解数据的离散时间与连续时间非线性信号处理。*Chemical Engineering Communications*，118(1):25–48，1992年。[50] Ramiro Rico-Martinez 和 Ioannis G. Kevrekidis。非线性系统的连续时间建模：一种基于神经网络的方法。载于*1993 IEEE International Conference on Neural Networks*，第1522–1525页。IEEE出版，1993年。[51] Yulia Rubanova、Ricky T.Q. Chen 和 David Duvenaud。用于不规则采样时间序列的潜在常微分方程。技术报告，2019年。[52] Saeed Saadatnejad、Mohammadhosein Oveisi 和 Matin Hashemi。基于LSTM的心电图分类：用于个人可穿戴设备的连续监测。*IEEEJournal of Biomedical and Health Informatics*，24(2):515–523，2 2020年。[53] Guanya Shi、Xichen Shi、Michael O'Connell、Rose Yu、Kamyar Azizzadenesheli、Animashree Anandkumar、Yisong Yue 和 Soon Jo Chung。神经网络着陆器：使用学习得到的动力学实现稳定的无人机着陆控制。载于*Proceedings - IEEE International Conference on Robotics and Automation*，第2019-May卷，第9784–9790页。电气电子工程师学会，5 2019年。[54] Esther Lydia Silva-Ram´ırez、Rafael Pino-Mej´ıas 和 Manuel L´opez-Coello。使用多层感知器进行单次插补，以及结合多层感知器和K近邻进行多重插补，以处理单调模式。*Applied Soft Computing*，29:65–74，4 2015年。[55] Michael L. Thompson 和 Mark A. Kramer。利用先验知识和神经网络对化学过程进行建模。*AIChE Journal*，40(8):1328–1340，8 1994年。[56] A. Uppal、W. H. Ray 和 A. B. Poore。连续搅拌釜反应器的动态行为。*ChemicalEngineeringScience*，29(4):967–985，4 1974年。[57] H. J.L. Van Can、H. A.B. Te Braake、C. Hellinga、K. C.A.M. Luyben 和 J. J. Heijnen。基于宏观衡算中神经网络的生物过程高效模型开发策略。*Biotechnology and Bioengineering*，54(6):549–566，6 1997年。[58] Christophe Vandekerckhove、Ioannis Kevrekidis 和 Dirk Roose。用于在慢流形上初始化约束运行方案的高效Newton-Krylov实现。*Journal of Scientific Computing*，39(2):167–188，5 2009年。[59] E. A. Wan 和 R. Van Der Merwe。用于非线性估计的无迹卡尔曼滤波器。载于*IEEE 2000 Adap-tiveSystemsfor SignalProcessing, Communications, and Control Symposium, AS-SPCC 2000*，第153–158页。电气电子工程师学会，2000年。[60] Yu Wang。使用LSTM神经网络进行动态系统辨识的新概念。载于*Proceedings of theAmerican Control Conference*，第5324–5329页。电气电子工程师学会，6 2017年。[61] Ronald J. Williams 和 David Zipser。持续运行的全循环神经网络学习算法。技术报告2，1989年。.[62] Ozal Yildirim。基于深度双向LSTM网络模型的新型小波序列，用于心电信号分类。*Computers in Biology and Medicine*，96:189–202，5 2018年。[63] Antonios Zagaris、C. William Gear、Tasso J. Kaper 和 Yannis G. Kevrekidis。无方程投影到慢流形的准确性与收敛性分析。*Mathematical Modelling and Numerical Analysis*，43(4):757–784，2009年。[64] Aiqing Zhu、Tom Bertalan、Beibei Zhu、Yifa Tang 和 Ioannis G. Kevrekidis。隐式模板化常微分方程网络的实现与（逆修改）误差分析。3 2023年。[65] Aiqing Zhu、Pengzhan Jin、Beibei Zhu 和 Yifa Tang。关于神经常微分方程中的数值积分。第*39*届国际机器学习会议，162(17–23 Jul):27527–27547，6 2022年。

补充材料 A

A.1 补充结果

A.1.1 URP模型

Case	Data $< \Delta t >$	求解器 max δt	任意的 数据 Δt	部分 观测值	初始 条件 给定值	解 误差	RHS 误差
A	0.1	0.1	✗	✗	✓	$\mathcal{L}_1$: $(1.31 \pm 0.24) \times 10^{-2}$ $\mathcal{L}_2$: $(9.68 \pm 1.71) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.58 \pm 0.27) \times 10^{-2}$	$\mathcal{L}_1$: $(3.66 \pm 0.52) \times 10^{-3}$ $\mathcal{L}_2$: $(2.72 \pm 0.36) \times 10^{-3}$ $\mathcal{L}_\infty$: $(4.56 \pm 0.57) \times 10^{-3}$
B	0.5	0.5	✗	✗	✓	$\mathcal{L}_1$: $(1.97 \pm 0.17) \times 10^{-2}$ $\mathcal{L}_2$: $(1.49 \pm 0.12) \times 10^{-2}$ $\mathcal{L}_\infty$: $(2.58 \pm 0.22) \times 10^{-2}$	$\mathcal{L}_1$: $(1.32 \pm 0.10) \times 10^{-2}$ $\mathcal{L}_2$: $(9.71 \pm 0.76) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.61 \pm 0.13) \times 10^{-2}$
C	0.5	0.1	✗	✗	✓	$\mathcal{L}_1$: $(1.14 \pm 0.15) \times 10^{-2}$ $\mathcal{L}_2$: $(8.69 \pm 1.18) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.50 \pm 0.24) \times 10^{-2}$	$\mathcal{L}_1$: $(7.90 \pm 1.64) \times 10^{-3}$ $\mathcal{L}_2$: $(5.95 \pm 1.21) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.03 \pm 0.21) \times 10^{-2}$
D	0.5	0.1	✓	✗	✓	$\mathcal{L}_1$: $(5.56 \pm 0.60) \times 10^{-3}$ $\mathcal{L}_2$: $(4.23 \pm 0.43) \times 10^{-3}$ $\mathcal{L}_\infty$: $(7.44 \pm 0.80) \times 10^{-3}$	$\mathcal{L}_1$: $(4.67 \pm 0.44) \times 10^{-3}$ $\mathcal{L}_2$: $(3.50 \pm 0.34) \times 10^{-3}$ $\mathcal{L}_\infty$: $(5.97 \pm 0.62) \times 10^{-3}$
E	0.5 (x_1) 0.55 (x_2)	0.1	✗	✓	✓	$\mathcal{L}_1$: $(8.08 \pm 0.79) \times 10^{-3}$ $\mathcal{L}_2$: $(6.09 \pm 0.56) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.04 \pm 0.09) \times 10^{-2}$	$\mathcal{L}_1$: $(4.91 \pm 0.63) \times 10^{-3}$ $\mathcal{L}_2$: $(3.66 \pm 0.46) \times 10^{-3}$ $\mathcal{L}_\infty$: $(6.27 \pm 0.75) \times 10^{-3}$
F	0.5	0.1	✓	✓	✓	$\mathcal{L}_1$: $(1.16 \pm 0.22) \times 10^{-2}$ $\mathcal{L}_2$: $(8.91 \pm 1.65) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.59 \pm 0.30) \times 10^{-2}$	$\mathcal{L}_1$: $(9.22 \pm 1.05) \times 10^{-3}$ $\mathcal{L}_2$: $(7.12 \pm 0.81) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.27 \pm 0.15) \times 10^{-2}$
URP BB	0.5	0.1	✓	✓	✗	$\mathcal{L}_1$: $(9.91 \pm 1.71) \times 10^{-3}$ $\mathcal{L}_2$: $(7.92 \pm 1.36) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.45 \pm 0.26) \times 10^{-2}$	$\mathcal{L}_1$: $(9.99 \pm 1.41) \times 10^{-3}$ $\mathcal{L}_2$: $(7.69 \pm 0.11) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.37 \pm 0.18) \times 10^{-2}$

表 A1：指标汇总表（续）：URP CSTR 数据。本表展示了模型在表示复杂度不断增加的数据上的模型性能 g 复杂度 形式。有关指标的详细信息，请参阅 A.2 节。

Case	可学习的 动力学 函数	可学习的 实验值 参数	解 误差	RHS 误差	动力学 函数 误差	实验值 参数 误差
URP BB	✗	✗	$\mathcal{L}_1$: $(9.91 \pm 1.71) \times 10^{-3}$ $\mathcal{L}_2$: $(7.92 \pm 1.36) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.45 \pm 0.26) \times 10^{-2}$	$\mathcal{L}_1$: $(9.99 \pm 1.41) \times 10^{-3}$ $\mathcal{L}_2$: $(7.69 \pm 0.11) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.37 \pm 0.18) \times 10^{-2}$	-	-
URP GB1	✓	✗	$\mathcal{L}_1$: $(1.37 \pm 0.18) \times 10^{-2}$ $\mathcal{L}_2$: $(9.91 \pm 1.26) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.53 \pm 0.18) \times 10^{-2}$	$\mathcal{L}_1$: $(6.24 \pm 0.72) \times 10^{-3}$ $\mathcal{L}_2$: $(4.43 \pm 0.51) \times 10^{-3}$ $\mathcal{L}_\infty$: $(6.87 \pm 0.80) \times 10^{-3}$	$(5.31 \pm 0.62) \times 10^{-3}$	-
URP GB2	✓	✓	$\mathcal{L}_1$: $(1.68 \pm 0.24) \times 10^{-2}$ $\mathcal{L}_2$: $(1.22 \pm 0.17) \times 10^{-2}$ $\mathcal{L}_\infty$: $(1.90 \pm 0.26) \times 10^{-2}$	$\mathcal{L}_1$: $(8.97 \pm 2.87) \times 10^{-3}$ $\mathcal{L}_2$: $(6.49 \pm 2.08) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.04 \pm 0.34) \times 10^{-2}$	$(7.77 \pm 2.25) \times 10^{-3}$	$(4.45 \pm 2.28) \times 10^{-2}$

表 A2：URP CSTR 数据的指标汇总表。此表展示了黑箱和灰箱模型的模型性能。所有模型均使用任意数据 Δt、部分观测值，且未给定初始条件。有关指标的详细信息，请参阅 A.2 节。

A.1.2 B&F 模型

Case	可学习 动力学 函数	可学习的 Exp. 参数	解 误差 From 真实 LC	RHS 误差	动力学 函数 误差	Exp. 参数 误差
BF BB	✗	✗	$\mathcal{L}_1$: $(6.26 \pm 0.81) \times 10^{-3}$ $\mathcal{L}_2$: $(3.25 \pm 0.43) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.36 \pm 0.17) \times 10^{-2}$	$\mathcal{L}_1$: $(4.00 \pm 0.77) \times 10^{-3}$ $\mathcal{L}_2$: $(2.12 \pm 0.45) \times 10^{-3}$ $\mathcal{L}_\infty$: $(7.13 \pm 1.28) \times 10^{-3}$	-	-
BF GB1	✓	✗	$\mathcal{L}_1$: $(1.04 \pm 0.15) \times 10^{-2}$ $\mathcal{L}_2$: $(5.16 \pm 0.68) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.90 \pm 0.24) \times 10^{-2}$	$\mathcal{L}_1$: $(9.94 \pm 1.41) \times 10^{-3}$ $\mathcal{L}_2$: $(5.43 \pm 0.80) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.68 \pm 0.23) \times 10^{-2}$	$\mathcal{L}_1$: $(1.02 \pm 0.14) \times 10^{-3}$ $\mathcal{L}_2$: $(7.66 \pm 1.14) \times 10^{-4}$ $\mathcal{L}_\infty$: $(1.24 \pm 0.17) \times 10^{-3}$	-
BF GB2	✓	✓	$\mathcal{L}_1$: $(1.22 \pm 0.57) \times 10^{-2}$ $\mathcal{L}_2$: $(6.10 \pm 2.72) \times 10^{-3}$ $\mathcal{L}_\infty$: $(2.33 \pm 1.09) \times 10^{-2}$	$\mathcal{L}_1$: $(1.37 \pm 0.60) \times 10^{-2}$ $\mathcal{L}_2$: $(7.50 \pm 3.2) \times 10^{-3}$ $\mathcal{L}_\infty$: $(2.47 \pm 1.48) \times 10^{-2}$	$\mathcal{L}_1$: $(1.48 \pm 0.71) \times 10^{-3}$ $\mathcal{L}_2$: $(1.09 \pm 0.49) \times 10^{-3}$ $\mathcal{L}_\infty$: $(1.79 \pm 1.17) \times 10^{-3}$	$(7.51 \pm 2.1) \times 10^{-2}$

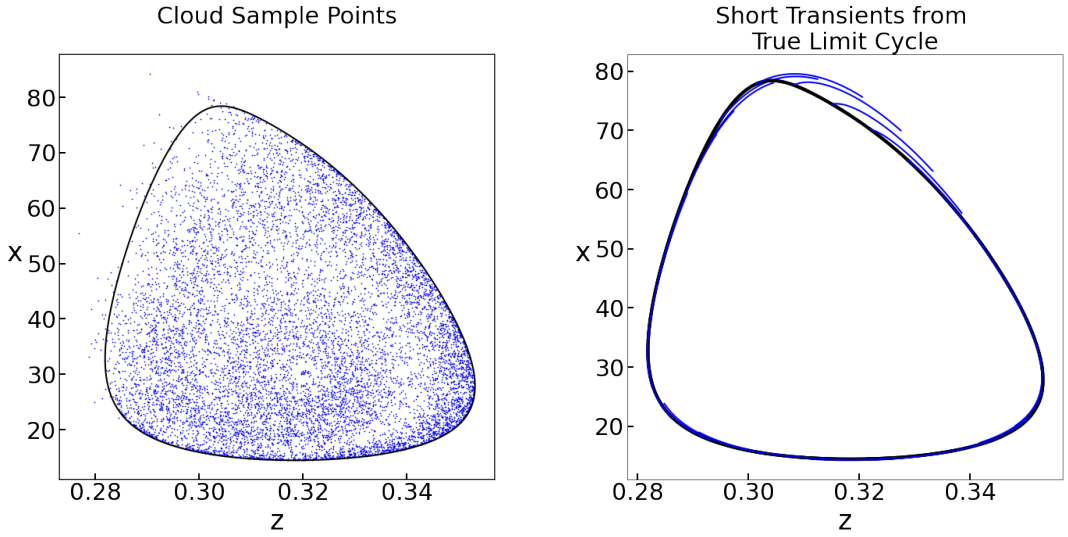
表 A3：B&F 数据的指标汇总表。本表展示了黑箱模型和灰箱模型的模型性能。所有模型均使用任意数据 Δt，仅有部分观测，且未给定初始条件。误差为 10 次训练运行的均值和标准差。误差指标的详情请参阅 A.2 节。

Case	初始 缩放	随机化 Data Δt	随机化 求解器 δt	解 误差 From 真实 LC	RHS 误差
BF BB A	$\times 100^{0}$	✗	✗	$\mathcal{L}_1:\ (8.02 \pm 4.02)\ \times 10^{-1}$ $\mathcal{L}_2:\ (4.33 \pm 2.58)\ \times 10^{-1}$ $\mathcal{L}_\infty:\ 2.03 \pm 1.43$	$\mathcal{L}_1:\ 14.6 \pm 2.56$ $\mathcal{L}_2:\ 8.47 \pm 1.63$ $\mathcal{L}_\infty:\ 38.3 \pm 9.43$
BF BB B	$\times 100^{-1}$	✗	✗	$\mathcal{L}_1:\ (7.84 \pm 0.92)\ \times 10^{-3}$ $\mathcal{L}_2:\ (4.13 \pm 0.53)\ \times 10^{-3}$ $\mathcal{L}_\infty:\ (1.72 \pm 0.24)\ \times 10^{-2}$	$\mathcal{L}_1:\ (5.29 \pm 1.50)\ \times 10^{-3}$ $\mathcal{L}_2:\ (2.89 \pm 0.92)\ \times 10^{-3}$ $\mathcal{L}_\infty:\ (9.23 \pm 2.49)\ \times 10^{-3}$
BF BB C	$\times 100^{0}$	✗	✓	$\mathcal{L}_1:\ (1.63 \pm 0.70)\ \times 10^{-1}$ $\mathcal{L}_2:\ (8.40 \pm 3.52)\ \times 10^{-2}$ $\mathcal{L}_\infty:\ (3.50 \pm 1.43)\ \times 10^{-1}$	$\mathcal{L}_1:\ 2.77 \pm 1.22$ $\mathcal{L}_2:\ 1.65 \pm 0.68$ $\mathcal{L}_\infty:\ 7.59 \pm 3.38$
BF BB D	$\times 100^{0}$	✓	✓ (因为 数据)	$\mathcal{L}_1:\ (2.29 \pm 0.39)\ \times 10^{-2}$ $\mathcal{L}_2:\ (1.14 \pm 0.17)\ \times 10^{-2}$ $\mathcal{L}_\infty:\ (4.46 \pm 0.66)\ \times 10^{-2}$	$\mathcal{L}_1:\ (4.57 \pm 1.00)\ \times 10^{-2}$ $\mathcal{L}_2:\ (2.88 \pm 0.68)\ \times 10^{-2}$ $\mathcal{L}_\infty:\ (1.21 \pm 0.31)\ \times 10^{-1}$
BF BB E	$\times 100^{-1}$	✓	✓ (因为 的数据)	$\mathcal{L}_1: (7.99 \pm 1.33)\ \times 10^{-3}$ $\mathcal{L}_2: (4.18 \pm 0.71)\ \times 10^{-3}$ $\mathcal{L}_\infty: (1.74 \pm 0.30)\ \times 10^{-2}$	$\mathcal{L}_1: (5.69 \pm 1.01)\ \times 10^{-3}$ $\mathcal{L}_2: (3.10 \pm 0.56)\ \times 10^{-3}$ $\mathcal{L}_\infty:\ (1.02 \pm 0.18)\ \times 10^{-2}$

表 A4：B&F 数据的指标汇总表。该表展示了共振效应。指标详情请参见 A.2 节。

A.2 使用的指标

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{p}) && True\ \ ODE \\ \hat{\dot{\mathbf{x}}} &= \begin{cases} \mathcal{N}(\mathbf{x}, \mathbf{p}; \theta) & Learned\ \ ODE\ \ (Black\ \ Box) \\ g(\mathbf{x}, \mathbf{p}, \mathcal{N}(\mathbf{x}, \mathbf{p}; \theta)) & Learned\ \ ODE\ \ (Gray\ \ Box) \end{cases} \end{aligned} \tag{A1}$$



(a) A.4 和 A.5 节的采样　　(b) A.3 节的短期瞬态过程

图 A1：(a) 通过随机化初始条件、进行时间积分并舍弃早期瞬态过程，从 1000 条不同轨迹中进行再采样，获得采样点。(b) 以沿真实极限环均匀分布的 50 个点作为初始条件进行积分，得到短期瞬态过程，积分时长为 $t = 20$

A.3 短期预测误差

若稳态解稳定，则令$\mathbf{x}_j^{LC} = \{x_i^{LC}\}_j$为稳态解；若稳态解不稳定，则令其为极限环上的一个点。令 N_v 为变量的数量， N_n 为测试点的数量， T为短期瞬态的时间周期。

从$\mathbf{x}_j^{LC}$开始，对真实的常微分方程和人工神经网络学习得到的常微分方程进行积分，时间为$t=T$。

$$\begin{aligned}\mathbf{x}_j(t=0) &= \mathbf{x}_j^{LC},\ \ 1 \le j \le N_n \\ \mathbf{y}_j &= \int_0^T f(\mathbf{x},\mathbf{p})dt,\ \ 1 \le j \le N_n \\ \hat{\mathbf{y}}_j &= \begin{cases} \int_0^T \mathcal{N}(\mathbf{x},\mathbf{p};\theta)dt,\ \ 1 \le j \le N_n & (Black\ \ Box) \\ \int_0^T g(\mathbf{x},\mathbf{p},\mathcal{N}(\mathbf{x},\mathbf{p};\theta))dt,\ \ 1 \le j \le N_n & (Gray\ \ Box) \end{cases}\end{aligned} \tag{A2}$$

$\mathbf{y}_j$ 和$\hat{}\mathbf{y}_j$ 分别是真实矩阵和预测矩阵。这些矩阵通过第 A.7 节中定义的各种范数汇总为一个误差指标。

A.4 右端项误差

$$\begin{aligned}\mathbf{y}_j &= f(\mathbf{x},\mathbf{p}),\ \ 1 \le j \le N_n \\ \hat{\mathbf{y}}_j &= \begin{cases} \mathcal{N}(\mathbf{x},\mathbf{p};\theta),\ \ 1 \le j \le N_n & (Black\ \ Box) \\ g(\mathbf{x},\mathbf{p},\mathcal{N}(\mathbf{x},\mathbf{p};\theta)),\ \ 1 \le j \le N_n & (Gray\ \ Box) \end{cases}\end{aligned} \tag{A3}$$

$\mathbf{y}_j$ 和$\hat{}\mathbf{y}_j$ 分别是真实矩阵和预测矩阵。这些矩阵通过第 A.7 节中定义的各种范数汇总为一个误差指标。

A.5 灰箱函数误差

对于灰箱模型，神经网络学习完整常微分方程函数的一个子集。

$$\begin{aligned}\dot{\mathbf{x}} &= f(\mathbf{x},\mathbf{p}) = g(\mathbf{x},\mathbf{p},\phi(\mathbf{x},\mathbf{p})) \quad True\ \ ODE \\ \hat{\dot{\mathbf{x}}} &= g(\mathbf{x},\mathbf{p},\hat{\phi}(\mathbf{x},\mathbf{p})) \qquad\qquad\ \ Learned\ \ ODE\end{aligned} \tag{A4}$$

$$\begin{aligned}\mathbf{y}_j &= \phi(\mathbf{x},\mathbf{p}),\ \ 1 \le j \le N_n \\ \hat{\mathbf{y}}_j &= \hat{\phi}(\mathbf{x},\mathbf{p}) = \mathcal{N}(\mathbf{x},\mathbf{p};\theta),\ \ 1 \le j \le N_n\end{aligned} \tag{A5}$$

$\mathbf{y}_j$ 和 $\hat{}\mathbf{y}_j$ 分别为真实矩阵和预测矩阵。这些矩阵通过第 A.7 节中定义的各种范数汇总为一个误差指标。

A.6 实验值 参数误差

某些灰箱模型包含具有物理意义和实验意义的附加参数，这些参数会在人工神经网络训练期间进行拟合。对于这些参数，使用相对误差。

设 $\boldsymbol{\kappa} = \{\kappa\}_k$ 为 e 实验参数，其中 $\hat{}\boldsymbol{\kappa} = \{\hat{\kappa}\}_k$ 为人工神经网络预测结果，N_P 为可学习的 e ers.

实验值参数

$$\mathcal{L}(\boldsymbol{\kappa},\hat{\boldsymbol{\kappa}}) = \frac{1}{N_P}\sum_{k=1}^{N_P}\left|\frac{\kappa_k - \hat{\kappa}_k}{\kappa_k}\right| \tag{A6}$$

A.7 范数

误差矩阵$\mathbf{y}_j$ 和$\hat{}\mathbf{y}_j$ 首先针对每个变量使用最小-最大归一化进行归一化。$\hat{}\mathbf{y}_j$ 使用$\mathbf{y}_j$的最小值/最大值进行缩放。

$$\begin{aligned}\tilde{\mathbf{y}}_j &= \frac{\mathbf{y}_j - \min(\mathbf{y})}{\max(\mathbf{y}) - \min(\mathbf{y})} \\ \tilde{\hat{\mathbf{y}}}_j &= \frac{\mathbf{y}_j - \min(\hat{\mathbf{y}})}{\max(\mathbf{y}) - \min(\mathbf{y})}\end{aligned} \tag{A7}$$

这些归一化误差矩阵使用各种范数凝练为单一度量：

$$\begin{aligned}
\mathcal{L}_1(\tilde{\mathbf{y}}_j, \tilde{\hat{\mathbf{y}}}_j) &= \frac{1}{N_n N_V} \sum_{j=1}^{N_n} \sum_{i=1}^{N_V} |\tilde{y}_{i,j} - \tilde{\hat{y}}_{i,j}| \\
\mathcal{L}_2(\tilde{\mathbf{y}}_j, \tilde{\hat{\mathbf{y}}}_j) &= \frac{1}{N_n N_V} \sum_{j=1}^{N_n} \sqrt{\sum_{i=}^{N_V} (\tilde{y}_{i,j} - \tilde{\hat{y}}_{i,j})^2} \\
\mathcal{L}_\infty(\tilde{\mathbf{y}}_j, \tilde{\hat{\mathbf{y}}}_j) &= \frac{1}{N_n} \sum_{j=1}^{N_n} \max_i |\tilde{y}_{i,j} - \tilde{\hat{y}}_{i,j}|
\end{aligned} \tag{A8}$$