arXiv:2304.03894v2 [math.NA] 9 Feb 2024

面向物理系统持续学习的多保真方法

Amanda Howard、Yucheng Fu 和 Panos Stinis先进计算、数学与数据部 太平洋西北国家实验室 华盛顿州里奇兰 99354
amanda.howard@pnnl.gov

2024年2月13日

摘要

我们提出了一种基于多保真度深度神经网络的新型持续学习方法。该方法学习先前训练模型的输出与当前训练数据集上模型期望输出之间的相关性，从而限制灾难性遗忘。单独使用时，多保真度持续学习方法在多个数据集上均表现出稳健的效果，能够限制遗忘。此外，我们还展示了多保真度方法可以与现有的持续学习方法相结合，包括回放和记忆感知突触，从而进一步限制灾难性遗忘。所提出的持续学习方法尤其适用于物理问题，因为在这些问题中，每个区域的数据都满足相同的物理定律；它也适用于物理信息神经网络，因为在这些情况下，我们预计先前模型的输出与当前训练区域上的模型输出之间存在较强的相关性。

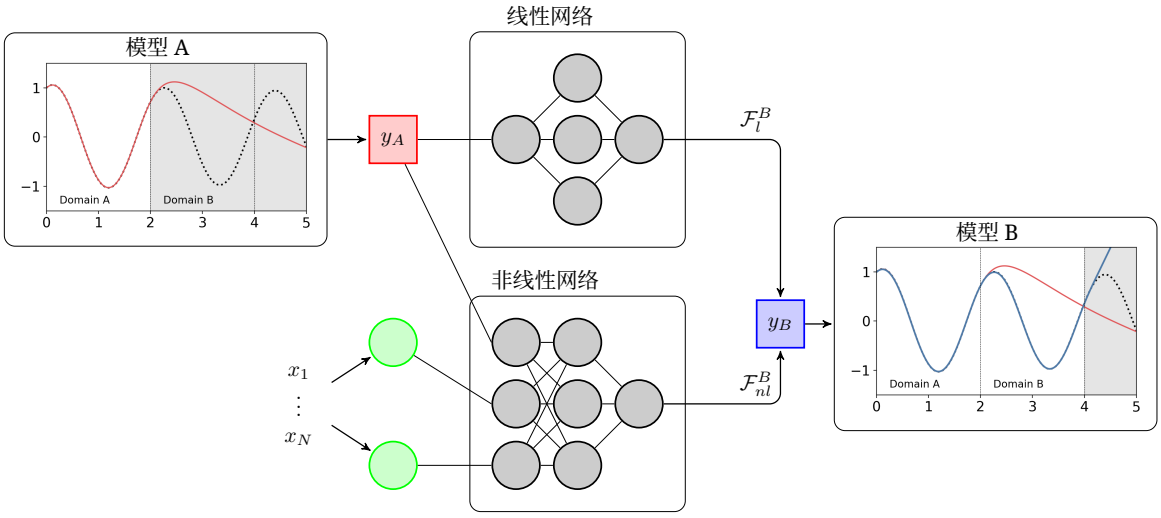


图 1：图形摘要

1 引言

在许多现实世界的机器学习应用中，数据会按顺序接收，或以离散数据集的形式接收。当这些数据用作训练数据时，系统相关的新信息要求对给定的神经网络进行彻底重新训练。许多

近期研究着重探讨如何将新接收的训练数据纳入机器学习模型，而无需使用完整数据集重新训练，也不会遗忘先前学到的模型。这一过程称为持续学习 [1]。持续学习的一个关键目标是限制灾难性遗忘，即突然且彻底地遗忘先前训练所用的数据。

人们提出了许多用于限制持续学习中遗忘现象的方法。在回放（复习）方法中，训练后续模型时会使用先前训练区域的训练集子集，因此该方法可以通过在先前区域上重新评估来限制遗忘 [2]。然而，回放需要访问先前使用过的训练数据集。这不仅要求为大型数据集提供大容量存储，还需要能够实际访问先前的数据集。但是，数据隐私可能会限制对先前数据集的访问，因此回放可能并非可行选项。回放的替代方法是正则化方法，即使用正则化器为神经网络中的每个参数分配权重，以表示该参数的重要性。随后施加惩罚，以防止权重最大的参数发生变化。人们已经提出了多种计算重要性权重的方法。其中最主要的选择包括突触智能 [3], 弹性权重固化（EWC） [4], 和记忆感知突触（MAS）[5]。后续研究表明，MAS 在多种使用场景中的表现名列前茅，并且对超参数的选择更加稳健，因此本文采用 MAS [6, 7]。最后，第三类持续学习方法包括采用任务专用模块 [8],集成 [9], 适配器 [10], 基于储备池计算的架构 [11],慢速—快速权重 [12, 13]等方法。

近年来，针对物理系统的科学机器学习方法一直是研究热点，例如流体力学和流变学 [14, 15, 16],、超材料开发 [17, 18, 19, 20], 、高速流动 [21, 22, 23],以及电力系统[24], 。具体而言，物理信息神经网络（PINN） [25, 26, 27] 可以通过自动微分准确表示微分算子，从而无需显式生成网格即可求解偏微分方程。针对 PINN 的持续学习研究仍然有限。作为初步尝试，虽然可以在整个区域上训练 PINN，因为数据采集和隐私问题并不适用，但已有许多系统被发现无法在整个目标时间区域上训练 PINN。例如，即使是本文采用的简单示例——摆和 Allen-Cahn 方程，也无法通过 PINN 在较长时间范围内完成训练。近期研究探索了改进这类系统中 PINN 训练效果的方法，包括神经切线核的应用 [28], ，但仍有许多工作有待开展。据我们所知，与 PINN 持续学习最相关的工作是 [29], 中的向后兼容物理信息神经网络，以及 [30]中的增量物理信息神经网络（iPINNs）。向后兼容物理信息神经网络在一系列 [31]时间区域上训练 N PINN，并在每个新区域中要求当前 PINN 的输出满足当前区域内的 PINN 损失函数，同时要求前一个模型的输出在所有先前区域上满足该损失函数。我们指出，本文采用的回放方法与这项工作不同，无论是在单保真度还是多保真度情形下都是如此，因为我们要求第 N 个神经网络满足所有先前区域中的残差，而不是前一个模型的输出。在 iPINNs 中，PINN 通过为每个方程设置一个子网络来满足一系列不同的方程，而不是在较长时间范围内处理同一个方程。

我们将在第 2 节介绍多保真持续学习方法。随后，我们将在第 3 节展示该方法在物理信息问题上的性能，并在第 4 节展示其在数据驱动问题上的性能。

2 多保真持续学习方法

我们假设有一个区域 Ω，并将其划分为 N 子域 $\Omega = \cup_{i=0}^{N}\Omega_i$。我们将在每个子区域 Ω_i上学习序列模型，目标是使第 i个模型能够在区域 $\cup_{j=0}^{i}\Omega j$上提供准确预测。也就是说，第 i个模型不会遗忘在训练中使用的早期区域上学到的信息。我们将重点关注物理系统的应用，其中我们可以获得数据或了解系统所遵循的物理定律。本节将首先简要介绍物理信息神经网络（PINN），然后讨论多保真持续学习方法（MFCL），最后介绍我们用于限制灾难性遗忘的方法。

2.1 物理信息神经网络

本节将简要介绍单保真度和多保真物理信息神经网络（PINN）。这类方法在 [28]中首次提出，并已针对许多相关应用进行了深入研究 [32, 14]。在这些应用中，PINN 通常用于求解初边值问题。

$$\mathbf{s}_t + \mathcal{O}_{\mathbf{x}}[\mathbf{s}] = \mathbf{0},\ \mathbf{x} \in \Omega, t \in [0, T] \tag{1}$$
$$\mathbf{s}(\mathbf{x}, t) = \mathbf{g}(\mathbf{x}, t)\ \mathbf{x} \in \partial\Omega, t \in [0, T] \tag{2}$$
$$\mathbf{s}(\mathbf{x}, 0) = \mathbf{u}(\mathbf{x})\ \mathbf{x} \in \Omega \tag{3}$$

其中，$\Omega \in \mathbb{R}^N$ 是一个边界为 $\partial\Omega$ 的开有界区域，$\mathbf{g}$ 和$\mathbf{u}$ 是已知函数，$\mathbf{x}$ 和 t 分别为空间和时间坐标。 $\mathcal{O}_{\mathbf{x}}$ 是关于$\mathbf{x}$的一般微分算子。我们希望通过一个深度神经网络（或一系列深度神经网络）来近似$\mathbf{s}(\mathbf{x}\ t)$，其参数为 γ，记作$\mathbf{s}^{\gamma}(\mathbf{x}\ t)$。神经网络通过最小化损失函数进行训练

$$\mathcal{L}(\gamma) = \lambda_{bc}\mathcal{L}_{bc}(\gamma) + \lambda_{ic}\mathcal{L}_{ic}(\gamma) + \lambda_r\mathcal{L}_r(\gamma) + \lambda_{data}\mathcal{L}_{data}(\gamma) \tag{4}$$

其中，下标 bc、ic、r 和 data 分别表示与边界条件、初始条件、残差以及任何给定数据相对应的项。我们将 N_{bc}、 N_{ic} 和 N_r 分别设为边界、初始和残差数据点的批次大小，并将训练数据表示为 $\{(\mathbf{x}^i_{bc}, t^i_{bc})$、$\mathbf{g}(\mathbf{x}^i_{bc}, t^i_{bc})\}_{i=0}^{N_{bc}}$、 $\{(\mathbf{x}^i_{ic})$、$\mathbf{u}(\mathbf{x}^i_{bc})\}_{i=0}^{N_{ic}}$ 以及 $\{(\mathbf{x}^i_r, t^i_r)\}_{i=0}^{N_r}$。边界和初始配点分别在各自的区域内均匀随机采样。 N_r 残差点的选取将在第 ?? 节中讨论。如果有表示解 $\mathbf{s}$ 的数据可用，我们还可以考虑额外的数据集 $\{(\mathbf{x}^i_{data}, t^i_{data})$、$\mathbf{s}(\mathbf{x}^i_{data}, t^i_{data})\}_{i=0}^{N_{data}}$。该项用于纳入基于数据的训练，我们将在第 4 节中介绍。

各个损失项由均方误差给出，

$$\mathcal{L}_{bc}(\gamma) = \frac{1}{N_{bc}}\sum_{i=0}^{N_{bc}} \left|\mathbf{s}_\gamma(\mathbf{x}^i_{bc}, t^i_{bc}) - \mathbf{g}(\mathbf{x}^i_{bc}, t^i_{bc})\right|^2 \tag{5}$$

$$\mathcal{L}_{ic}(\gamma) = \frac{1}{N_{ic}}\sum_{i=0}^{N_{ic}} \left|\mathbf{s}_\gamma(\mathbf{x}^i_{ic}, 0) - \mathbf{u}(\mathbf{x}^i_{ic})\right|^2 \tag{6}$$

$$\mathcal{L}_r(\gamma) = \frac{1}{N_r}\sum_{i=0}^{N_r} \left|\mathbf{r}_\gamma(\mathbf{x}^i_r, t^i_r)\right|^2 \tag{7}$$

$$\mathcal{L}_{data}(\gamma) = \frac{1}{N_{data}}\sum_{i=0}^{N_{data}} \left|\mathbf{s}_\gamma(\mathbf{x}^i_{data}, t^i_{data}) - \mathbf{s}(\mathbf{x}^i_{data}, t^i_{data})\right|^2 \tag{8}$$

其中

$$\mathbf{r}_\gamma(\mathbf{x}, t) = \frac{\partial}{\partial t}\mathbf{s}_\gamma(\mathbf{x}, t) + \mathcal{O}_{\mathbf{x}}[\mathbf{s}_\gamma(\mathbf{x}^i_{data}, t^i_{data})]. \tag{9}$$

加权参数 λ_{bc}、 λ_{ic}、 λ_r和 λ_{data} 由用户在训练前选定。

本文采用的多保真物理信息神经网络受 [33]的启发。我们假设有一个低保真模型，其形式为深度神经网络，以较低精度近似给定的数据集或微分算子。我们希望训练另外两个神经网络，以学习低保真度近似与高保真近似或高保真数据之间的线性和非线性相关性。我们将这两个神经网络分别记为 $\mathcal{NN}_l$（用于线性相关性）和 $\mathcal{NN}_{nl}$（用于非线性相关性）。输出为$\mathbf{s}^{\gamma}(\mathbf{x}\ t) = \mathcal{NN}_{nl}(\mathbf{x}\ t;\ \gamma) + \mathcal{NN}_l(\mathbf{x}\ t;\ \gamma)$，其中 γ是线性网络和非线性网络的所有可训练参数。损失函数还包含一个附加项，

$$\mathcal{L}_{MF}(\gamma) = \lambda_{bc}\mathcal{L}_{bc}(\gamma) + \lambda_{ic}\mathcal{L}_{ic}(\gamma) + \lambda_r\mathcal{L}_r(\gamma) + \lambda_{data}\mathcal{L}_{data}(\gamma) + \lambda\sum(\gamma_{nl,ij})^2, \tag{10}$$

其中 $\{\gamma_{nl,ij}\}$ 是非线性网络 $\mathcal{NN}_{nl}$的所有权重和偏置的集合。$\mathcal{NN}_l$ 中不使用激活函数，以便学习前一次预测与高保真模型之间的线性相关性。

2.2 多保真度持续学习

在 MFCL 方法中，我们利用先前区域上已训练模型与当前区域上预期模型之间的相关性。具体而言，我们将先前模型 $\mathcal{NN}_{i-1}$用作区域 Ω_i的低保真模型。然后，我们学习 $\mathcal{NN}_{i-1}$在区域 Ω_i 上与该区域给定的数据或物理规律之间的相关性。通过学习线性项和非线性项的一般组合，我们能够捕捉复杂的相关性。由于该方法只学习先前模型与新模型之间的相关性，因此通常可以在每个子区域中使用规模较小的神经网络。该流程需要两个初始步骤:

1. 在 Ω_1，上训练一个（单保真度）深度神经网络或物理信息神经网络，记为 $\mathcal{NN}^*(\mathbf{x}\ t;\ \gamma^*)$。该神经网络将近似单一区域内的解。2. 在 Ω_1，上训练一个多保真度深度神经网络或物理信息神经网络，并将单保真度模型 $\mathcal{NN}^*(\mathbf{x}\ t;\ \gamma^*)$ 作为低保真度近似输入。这个初始多保真度神经网络记为 $\mathcal{NN}_1(\mathbf{x}\ t;\ \gamma_1)$

随后，对于每个新增区域 Ω_i，我们在 Ω_i 上训练一个多保真度深度神经网络或物理信息神经网络，记为 $\mathcal{NN}_i(\mathbf{x}\ t;\ \gamma_i)$，并将先前的多保真度模型 $\mathcal{NN}_{i-1}(\mathbf{x}\ t;\ \gamma_{i-1})$ 作为低保真度近似输入。目标是使 $\mathcal{NN}_i(\mathbf{x}\ t;\ \gamma_i)$ 能够在 $\cup_{j=1}^{i}\Omega_i$ 上提供准确的解，即使训练该多保真度神经网络 $\mathcal{NN}_i$ 时未使用来自 $\Omega j,\ j < i$ 的数据。图 2 给出了该方法的示意图。

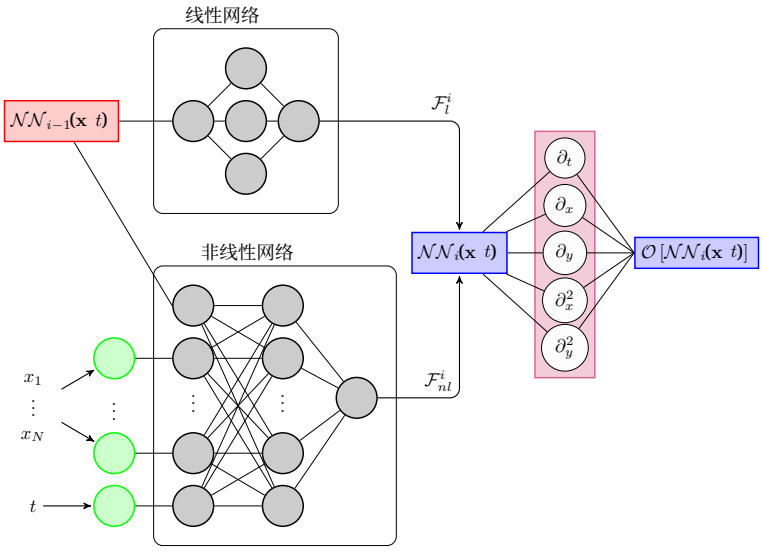


图 2：MF-CL 方法在区域 Ω_i上的示意图。先前训练的神经网络的输出，$\mathcal{NN}_{i-1}(\mathbf{x}\ t;\ \gamma_{i-1})$，被用作线性和非线性子网络在点 $(\mathbf{x}\ t)\ \in\Omega_i$、$\mathbf{x}\ \in\mathbb{R}^N$ 处的输入。输出神经网络是线性和非线性子网络的总和。

正如我们将要展示的，与单独进行单保真度训练相比，MF-CL 方法能以更少的遗忘获得更准确的结果。不过，该方法还可通过一些已有方法加以改进，这些方法既可减少持续学习中的遗忘，也可用于选择训练物理信息神经网络的配点。下面将讨论这些方法。

2.3 记忆感知突触

记忆感知突触（MAS）是一种持续学习方法，它通过为神经网络中的每个神经元分配重要性权重，来尝试限制持续学习中的遗忘。随后，在损失函数中加入惩罚项，以防止训练下一个网络时重要权重的值出现较大偏差。通过衡量神经网络输出 $\mathcal{NN}_n$ 对网络参数 [5] 变化的敏感程度，可以确定重要性权重。对于神经网络中的每个权重和偏置 γ_{ij}，我们计算其重要性权重参数

$$\Omega_{ij}^n = \frac{1}{N}\sum_{k=1}^{N}\left\|\frac{\partial\left(\ell_2^2\mathcal{NN}_n(x_k;\gamma)\right)}{\partial\gamma_{ij}}\right\| \tag{11}$$

其中，ℓ_2^2 表示神经网络 $\mathcal{NN}_n$ 在 x_k处的输出的 ℓ_2 范数平方。于是，式（10）中的损失函数修改为：

$$\begin{aligned}\mathcal{L}_{MF,MAS}(\gamma^n) =&\lambda_{bc}\mathcal{L}_{bc}(\gamma^n) + \lambda_{ic}\mathcal{L}_{ic}(\gamma^n) + \lambda_r\mathcal{L}_r(\gamma^n) + \lambda_{data}\mathcal{L}_{data}(\gamma^n) + \lambda\sum_{i,j}(\gamma_{nl,ij}^n)^2\\ &+\lambda_{MAS}\sum_{i,j}\Omega_{ij}^{n-1}\left(\gamma_{ij}^n-\gamma_{ij}^{n-1}\right)^2\end{aligned} \tag{12}$$

将 MAS 应用于多保真神经网络时，我们分别计算 MAS 项：

$$\Omega_{ij}^{n,nl} = \frac{1}{N}\sum_{k=1}^{N}\left\|\frac{\partial\left(\ell_2^2\mathcal{NN}_n^{nl}(x_k;\gamma)\right)}{\partial\gamma_{ij}^{nl}}\right\|,\quad \Omega_{ij}^{n,l} = \frac{1}{N}\sum_{k=1}^{N}\left\|\frac{\partial\left(\ell_2^2\mathcal{NN}_n^{l}(x_k;\gamma)\right)}{\partial\gamma_{ij}^{l}}\right\| \tag{13}$$

其中，nl 表示非线性网络，l 表示线性网络。这样可以分别确定权重在计算线性项和非线性项时的重要性，而不是确定两个网络输出之和整体的重要性。线性部分和非线性部分使用相同的参数 λ_{MAS}。

2.4 回放

在回放过程中，每次迭代都会从先前训练过的区域 $\cup_{i=1}^{n-1}\Omega_i$ 中选取若干点，并在这些点上评估残差损失 $\mathcal{L}_r(\gamma^n)$。这样，多保真度训练仍能在先前训练过的区域内满足 PDE。对于物理信息神经网络，回放方法只需了解 $\cup_{i=1}^{n-1}\Omega_i$的几何形状，而无需知道模型在该区域上的输出值。

2.5 迁移学习

在本文的所有情形中，每个后续神经网络 $\mathcal{NN}_i, i \geq 2$ 的可训练参数值均由前一个神经网络 $\mathcal{NN}_{i-1}$ 的可训练参数最终值初始化。用符号表示，即 $\gamma_i^0 = \gamma_{i-1}$。这种方法可以加快训练速度，因为神经网络无需随机初始化。我们注意到，先前的一些研究发现，随机初始化每个后续神经网络可以减少遗忘 [34], ；这一方案留待未来研究探索。

3 物理信息训练

本节举例说明在物理信息神经网络无法训练的情况下，如何应用多保真持续学习。我们展示了随时间进行持续学习可以提高 PINN 在长时间积分问题中的训练精度，因为单个 PINN 并不足以解决这类问题。训练所用的所有超参数均列于附录 8。

3.1 摆的动力学

本节中，我们考虑由 [29] 阻尼的重力摆。该系统由关于$t \in [0, T]$的常微分方程描述

$$\frac{ds_1}{dt} = s_2, \tag{14}$$

$$\frac{ds_2}{dt} = -\frac{b}{m}s_2 - \frac{g}{L}\sin(s_1). \tag{15}$$

初始条件由 $s_1(0) = s_2(0) = 1$ 给出。我们取 $m = L = 1$、 $b = 0.05$ 和 $g = 9.81$，并取$T = 10$。

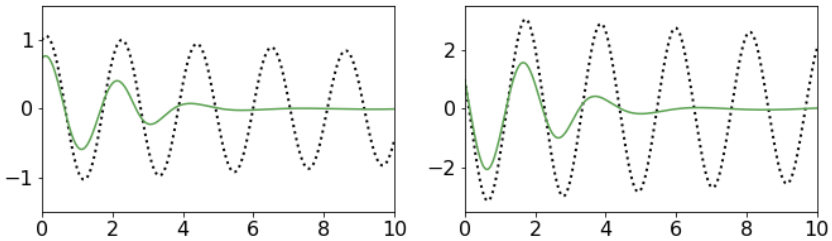


图 3：训练单个物理信息神经网络以满足方程 14 和 15（实线）所得结果，与 s_1 (左) 和 s_2 (右) 的精确解（虚线）进行比较。结果迅速衰减至零，且学习得到的解与精确解吻合不佳。

我们首先考虑图 3 中在 $t \in [0, 10]$ 训练的单个物理信息神经网络。解迅速趋于零，表明即使对于这个简单系统，单个物理信息神经网络也无法捕捉其长期动力学。 [29] 中也展示了类似结果。值得注意的是，近期已有一些进展，旨在改进物理信息神经网络在长时间积分问题中的训练 [29, 35, 36]。本节将探讨如何通过将时间区域划分为子区域，利用持续学习获得长时间范围内的精确解。

我们将区域划分为五个子域 $\Omega_i = [2(i-1)\ 2i]$，并在每个区域上分别采用传统的单保真度持续学习和 MF-CL，以及结合回放和 MAS 的 SF-CL 与 MF-CL 方法进行训练。对于每个

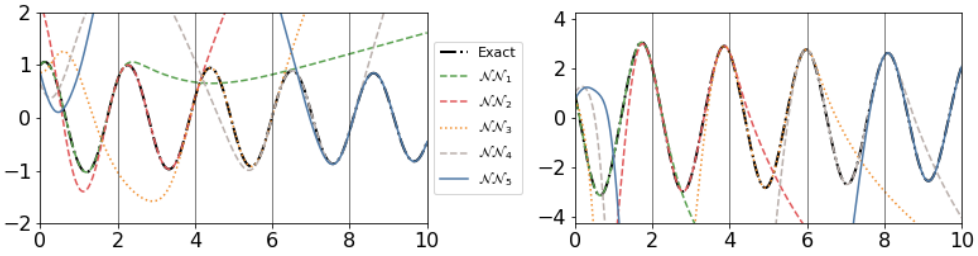
	单保真度	多保真度
单独应用	16.774	2.147
回放	0.041	0.079
MAS	1.390	1.146

表 1：摆问题在全域上的最终输出 $\mathcal{NN}_5$ 的均方根误差。对于 MAS 情况，神经网络针对 λ_{MAS} 的六个取值进行训练，并在上表中展示均方根误差最低的情况。回放实验在每个隐藏层中设置 $N = 100$ 个神经元；有关每个隐藏层神经元数量不同的情况，请参见表 2。

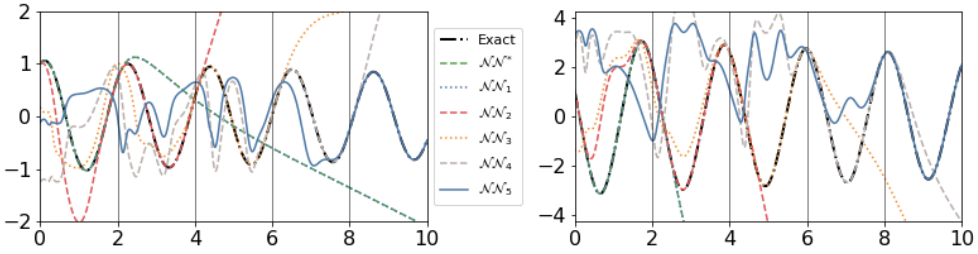
情况下，我们计算 t 全域 $\Omega = [0, 10]$ 上最终输出 $\mathcal{NN}_5$ 的均方根误差（RMSE），其中

$$RMSE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[\mathcal{NN}_5(t_j) - \mathbf{s}(t_j)]^2}, \tag{16}$$

其中 s 表示精确解。如果遗忘程度有限，最终解在全域上的均方根误差应较小。



(a) 单保真度



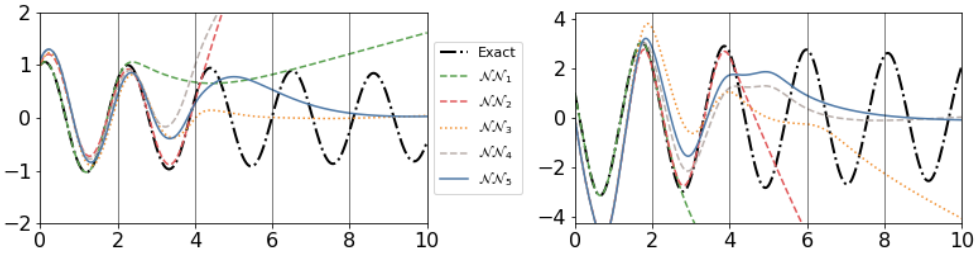
(b) 多保真度

图 4：仅使用单保真度 (a) 和多保真度 (b) 进行训练以满足方程 14 和 15 的结果，并与 s_1（左）和 s_2（右）的精确解（点划线）进行比较。尤为重要的是最终神经网络 $\mathcal{NN}_5$（蓝色实线），它是在 $\Omega_5 = [8, 10]$上训练的。尽管 (b) 中的多保真度结果存在显著误差，但它们仍显著优于 (a) 中的单保真度结果。在单保真训练中，每个神经网络 $\mathcal{NN}_i$ 仅在子区域 Ω_i内准确，而在 Ω_i 之外进行外推则面临很大困难。

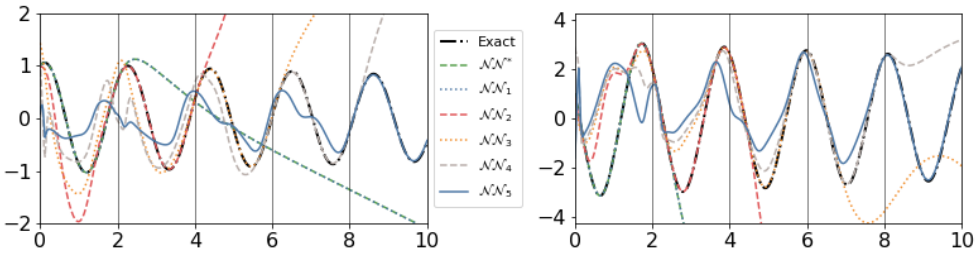
从表 1 可以明显看出，回放在两种情况下都表现最佳，并且显著优于其他任何方法。单独应用 SF 的情况均方根误差较大，这并不意外，因为它没有采用任何防止遗忘的技术。该情况如图 4a 所示。

图 5 展示了针对所考虑的各组超参数所取得的最佳 MAS 结果，其中 $\lambda_{MAS} = 100$ 对应单保真度，$\lambda_{MAS} = 0.001$ 对应多保真度。鉴于均方根误差较小，这并不意外：采用 MAS 的多保真度训练优于单保真训练。

如图 6 所示，SF-回放的表现似乎优于 MF-回放。不过，值得关注的是表 2 中均方根误差随网络规模变化的情况。MF-回放对网络规模的变化具有鲁棒性，而单保真度情况只有采用非常特定的架构时，才能达到较小的均方根误差。

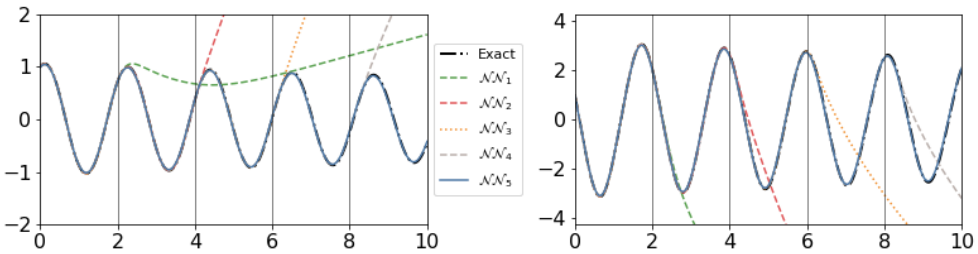


(a) 单保真度

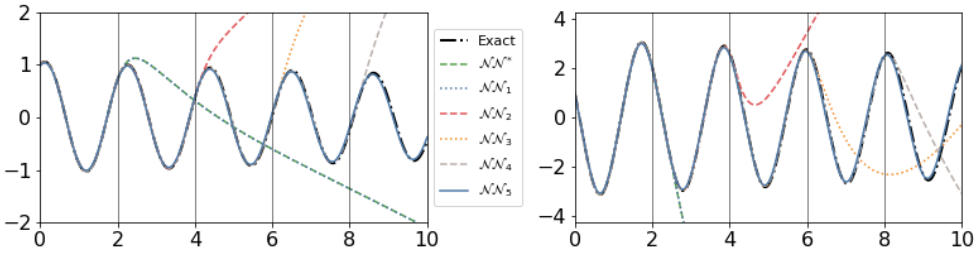


(b) 多保真度

图 5：使用 MAS 训练单保真度 (a) 和多保真度 (b)，使其满足公式 14 和 15；结果与 s_1 (左) 和 s_2 (右) 的精确解（点划线）进行了比较。特别重要的是最终的神经网络，$\mathcal{NN}_5$(蓝色实线)，它在 $\Omega_5 = [8, 10]$ 上进行训练。此处绘制的模拟结果在所有测试的超参数组中，在 Ω 上的均方根误差最小，为$\mathcal{NN}_5$。在单保真度情形下，MAS 似乎对训练施加了过于严格的限制，导致后续神经网络 $\mathcal{NN}_i$ 在各自的区域 Ω_i 上不再准确。对于多保真度训练，解在全域更广的范围内保持准确，且与单独进行多保真度训练相比，均方根误差有所降低。



(a) 单保真度



（b）多保真度

图 6：使用回放训练单保真度（a）和多保真度（b）以满足公式 14 和 15 的结果，并与 s_1（左）和 s_2（右）的精确解（点划线）进行比较。两种情况下都几乎没有遗忘。

N	单保真度	多保真度
25	3.130	0.044
50	0.107	0.027
100	0.041	0.079
150	0.055	0.054
200	0.091	0.038

表2：摆问题在全域上的最终输出 $\mathcal{NN}_5$ 的均方根误差。单保真度情形包含五个隐藏层，每层有 N 个神经元。多保真度情形中，每个非线性神经网络包含五个隐藏层，每层有 N 个神经元。多保真度线性神经网络包含一个隐藏层，其中有20个神经元。

3.2 Allen-Cahn方程

Allen-Cahn方程表示为

$$u_t - c_1^2 u_{xx} + 5u^3 - 5u = 0, \quad t \in (0, 1], x \in [-1, 1] \tag{17}$$
$$u(x, 0) = x^2 \cos(\pi x), \quad x \in [-1, 1] \tag{18}$$
$$u(x, t) = u(-x, t), \quad t \in [0, 1], x = -1, x = 1 \tag{19}$$
$$u_x(x, t) = u_x(-x, t), \quad t \in [0, 1], x = -1, x = 1 \tag{20}$$

我们采用 $c_1^2 = 0.0001$。Allen-Cahn方程通过直接应用 [37, 38],求解时，对于物理信息神经网络而言是出了名的困难（见图7）。通过对物理信息神经网络进行修改，已经能够成功求解Allen-Cahn方程，包括使用离散Runge-Kutta神经网络 [28], 对配点进行自适应采样 [37],以及采用向后兼容的物理信息神经网络 [30]。在本节中，我们展示了如何通过应用多保真度持续学习框架，准确学习Allen-Cahn方程的解。

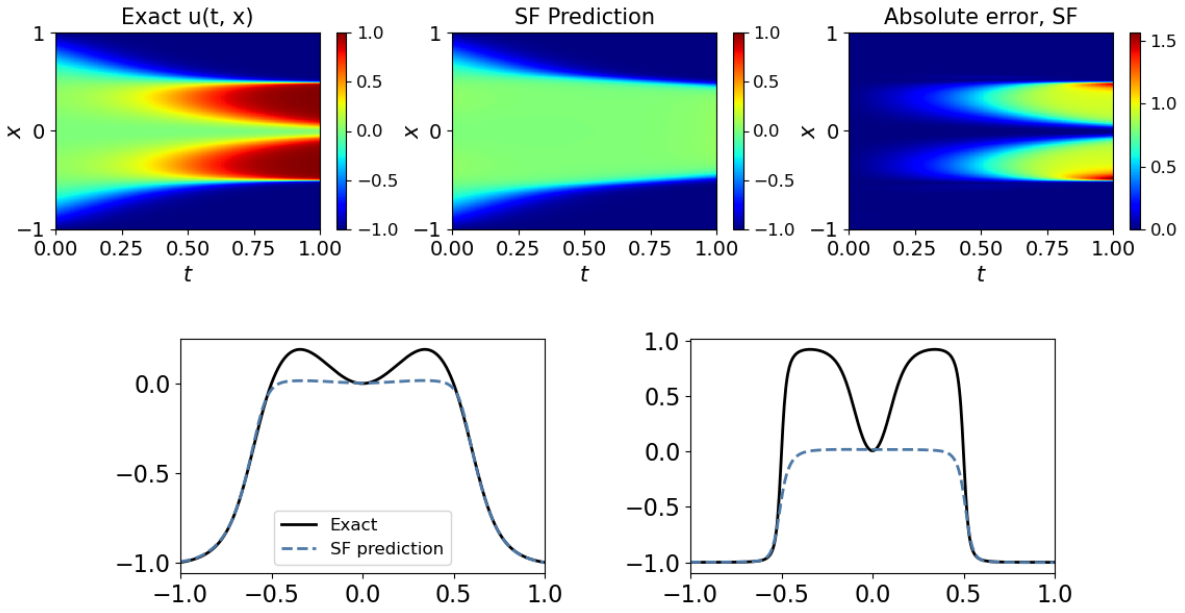


图7：训练单个物理信息神经网络求解Allen-Cahn方程的结果。底部图像分别取自$t = 0.25$ （左）和 $t = 0.75$ （右）。尽管该物理信息神经网络在约0.3之前训练良好，但随着 t的增加，解的质量逐渐下降。

我们将区域划分为四个子区域 $\Omega_i = [2(i-1)\ 2i]$，并报告 $\mathcal{NN}_4$ 在全域 Ω 上的相对均方根误差。如图 8 所示，单独训练多保真度方法和单保真度方法时，两者的相对均方根误差大致相同。MAS 和回放均能改善结果，分别见图 9 和图 10。结果汇总于表 3。

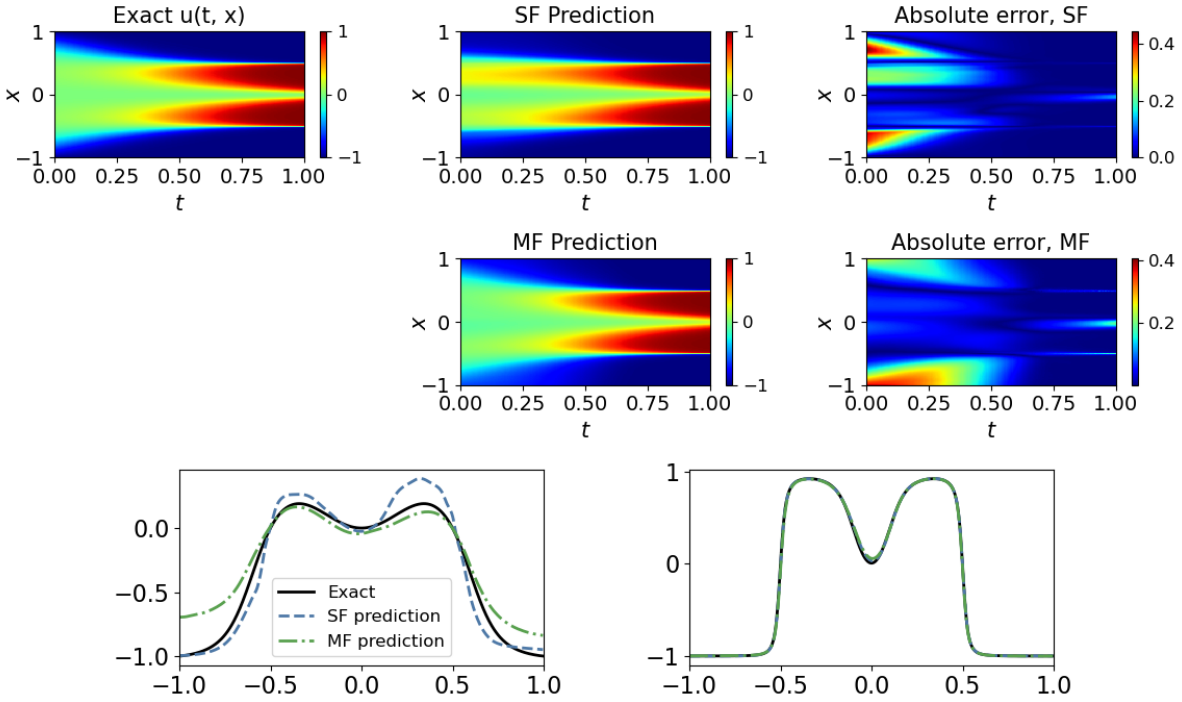


图 8: $\mathcal{NN}_4$仅使用单保真度和多保真度物理信息神经网络训练得到的 Allen-Cahn方程结果。底部图分别取自 $t = 0.25$ （左）和 $t = 0.75$ （右）。多保真度结果的误差约为单保真度结果的一半。

	单保真度	多保真度
无CL	0.126	0.128
回放	0.087	0.084
MAS	0.146	0.056

表 3：Allen-Cahn方程在全域上的最终输出 $\mathcal{NN}_4$的相对均方根误差。对于 MAS 的情况，神经网络针对 λ_{MAS}的七个取值进行训练，上表展示了均方根误差最低的情况。

4 数据驱动训练

4.1 电池

在这种情况下，如果添加额外的数据集，事先无法明确判断它属于哪个子区域。因此，最终模型必须能够在整个区域内准确预测当前情况，同时避免遗忘。

测试中选用钒氧化还原液流电池（VRFB）系统来生成数据集。图 11 左侧的图像展示了 VRFB 的典型配置，该系统由电极、集流体和隔膜组成。负极侧和正极侧各有一个储罐，分别用于储存 V^{2+}/V^{3+} 和 V^{4+}/V^{5+}氧化还原电对。我们采用 MFCL方法，通过给定的充电电压曲线识别施加的充电电流。为生成 VRFB 充电曲线数据集，我们使用了一种计算效率很高的二维解析模型[39, 40]。该模型充分解析了电池单体内活性物质传输、电化学反应动力学和流体动力学之间的耦合物理过程，从而忠实地表征 VRFB系统。关于该模型及其参数的更多详情见 [39]。图 11 右侧的图展示了五种选定电流水平下的典型充电曲线。对于给定的充电电流，电池电压（E）在不同荷电状态（SOC）下计算，以形成作为输入数据的充电曲线。产生该充电曲线的施加充电电流 I 是我们要预测的输出量。

我们根据充电电流将数据集划分为五组，并在单保真度情形下分别使用和不使用 MAS 进行训练。这些子区域为 $\Omega_1 = [0.1, 2)$、$\Omega_2 = [2, 4)$、$\Omega_3 = [4, 6)$、$\Omega_4 = [6, 8)$ 和 $\Omega_5 = [8, 9]$。误差通过以下方式计算：

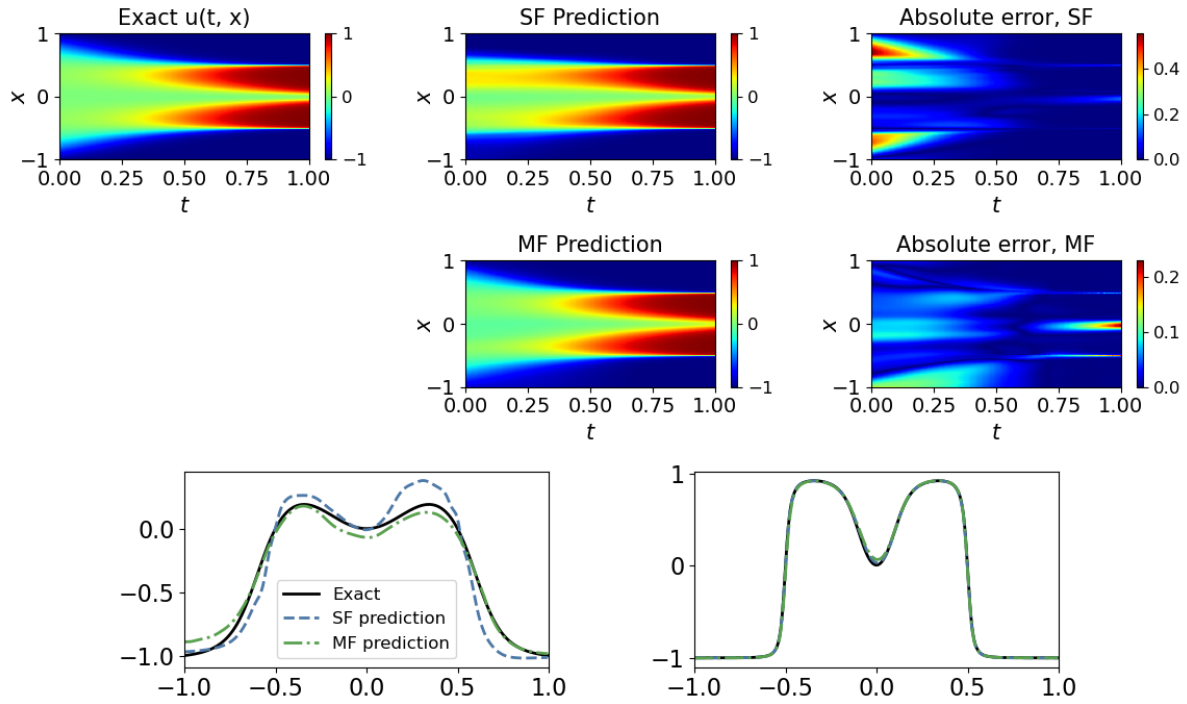


图9： $\mathcal{NN}_4$ 使用 MAS 对 Allen-Cahn方程进行单保真度训练和多保真度物理信息神经网络训练的结果。底部图像分别取自 $t = 0.25$ （左）和 $t = 0.75$ （右）。这些结果是所有考察的超参数组合中 MAS 的最佳结果。多保真度结果的误差约为单保真度结果的四分之一。

计算 $\mathcal{NN}_5$ 在从 $\Omega = \cup_{i=1}^{5}\Omega_i$ 中选取的测试集上的输出的均方根误差。我们测试了两种网络架构：一种较宽的神经网络，每个网络包含两个隐藏层，每层有 80 个神经元；另一种较深且较窄的神经网络，包含三个隐藏层，每层有 40 个神经元。首先，我们分别单独使用单保真度和多保真度方法进行训练，结果见图 12。多保真持续学习的结果表明，与单保真度持续学习相比，其遗忘更少。

接下来，我们考察添加 MAS 的影响。我们比较了采用和不采用 MAS 缩放的窄型和宽型神经网络，共四种情况。多保真度 MAS 的结果显示出显著改善，见图13。在图14中，我们比较了不同 MAS超参数 λ_{MAS}下的性能。我们发现，单保真度方法的性能较为稳健，因为它对 λ_{MAS}. 的取值不敏感。然而，其准确度并不高。另一方面，对于大多数 λ_{MAS}. 的取值，多保真方法的准确度都显著高于单保真度方法。总体而言，多保真度结果显著优于单保真度结果。

4.2 能源消耗

为了提供数据驱动的持续学习的第二个示例，我们考虑来自 [41] 的城市级每日能源消耗数据集。该数据集包含纽约、萨克拉门托和洛杉矶三个大都市区的每日能源使用量，以及每日天气数据。使用三年的数据作为测试集，另有一年作为测试集。

能源使用量很大程度上取决于天气：较暖月份会使用空调，冬季月份则会供暖。因此，为持续学习训练提供不同任务时，我们按季度划分三年的训练数据。任务1的训练数据为1月至3月，任务2为4月至6月，任务3为7月至9月，任务4为10月至12月。所有任务的测试集均用于预测2018年7月至2019年6月的能源使用量。图15展示了按任务划分的训练数据和测试数据。

我们训练采用和不采用 MAS 的单保真度神经网络和多保真度神经网络。我们考虑了一系列 $\lambda \in [0.001, 100]$。我们还将其与不采用持续学习训练神经网络的情况进行比较。在这种情况下，单保真度深度神经网络接收来自全部四个任务的所有 训练数据，以尝试预测 2018 年 7 月至 2019 年 6 月的能耗。该情况作为

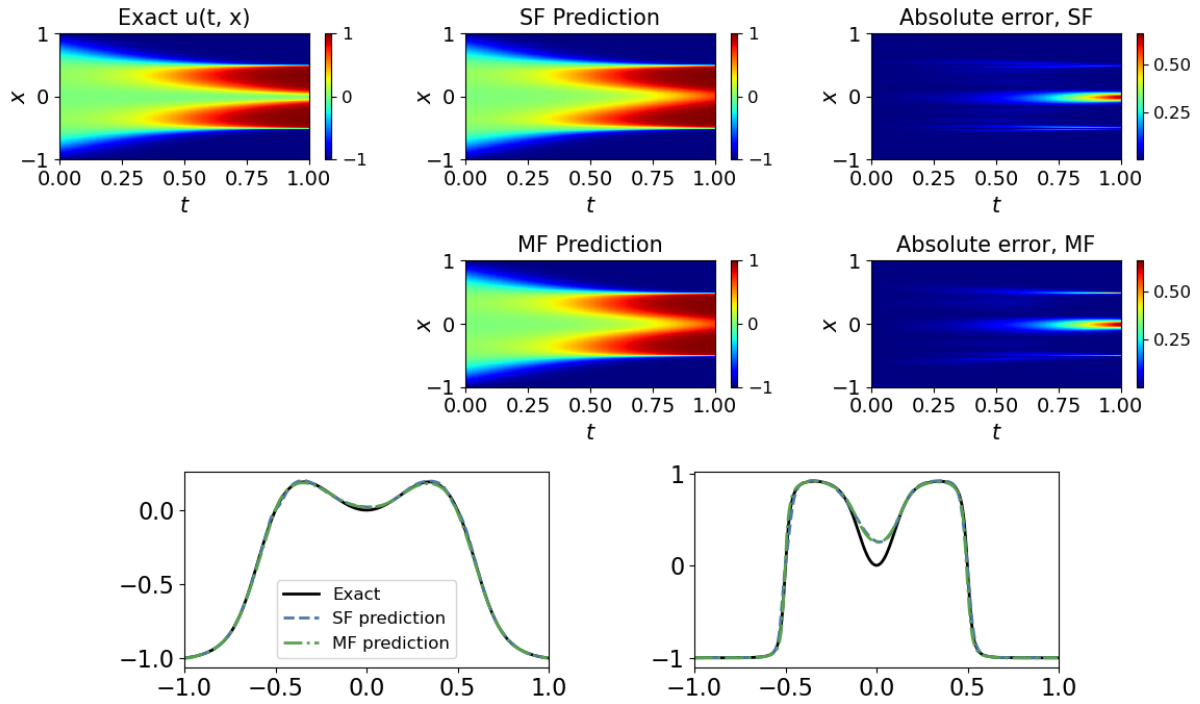


图10：$\mathcal{NN}_4$使用回放训练单保真度和多保真度物理信息神经网络训练在 Allen-Cahn方程上的结果。底部图像分别取自 $t = 0.25$ （左）和 $t = 0.75$ （右）。

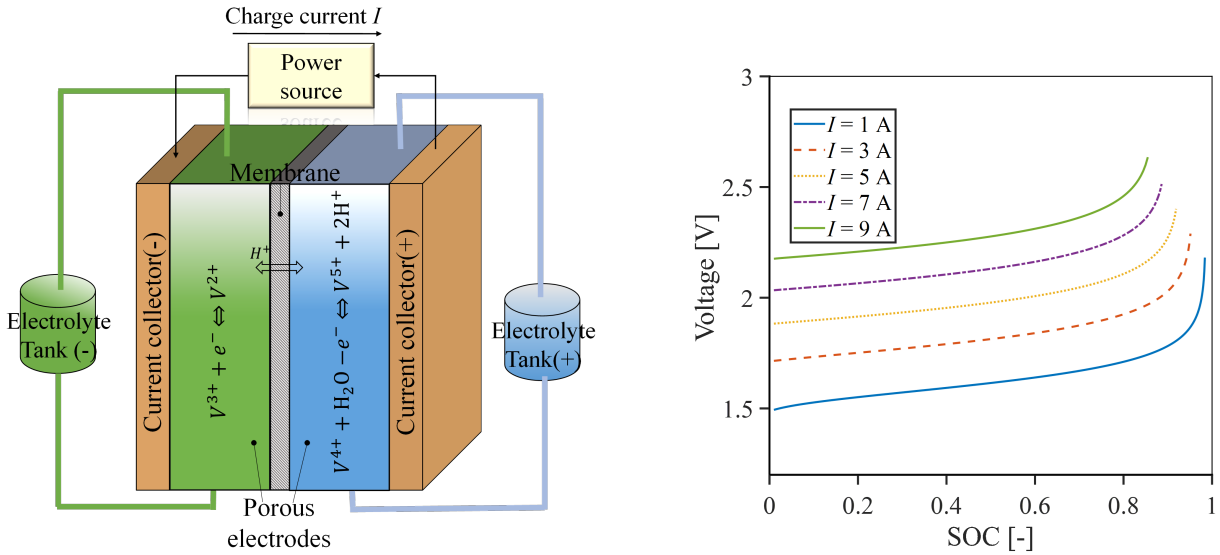
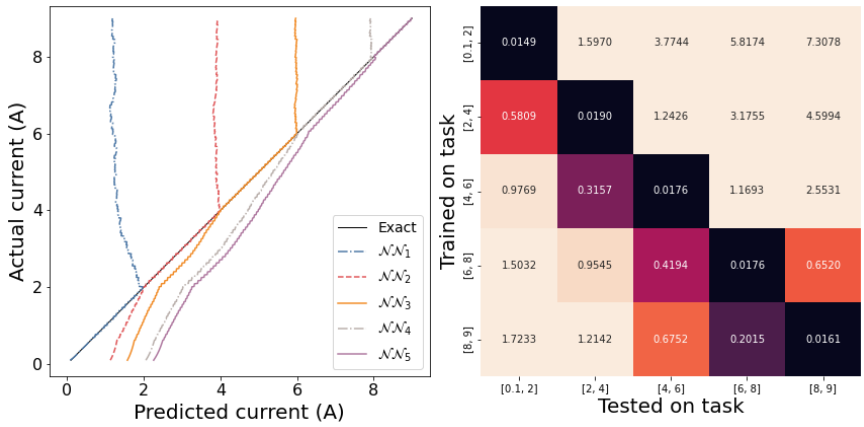
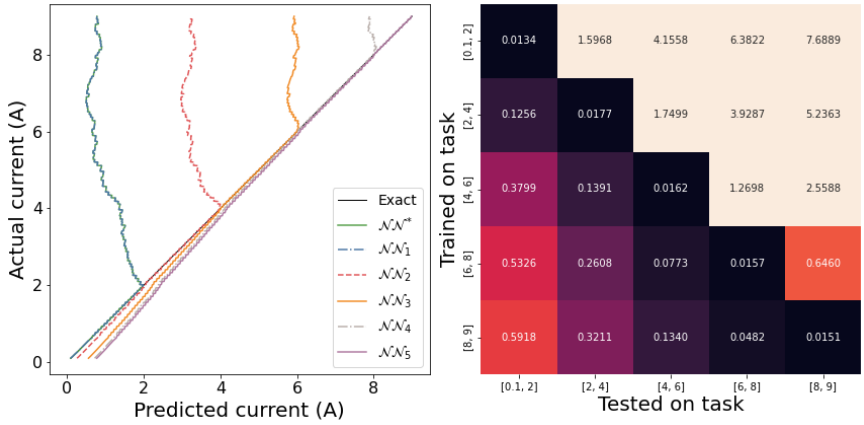


图11：用于电池数据生成的 VRFB系统（左）。不同充电电流下的充电曲线分布样本（右）。

我们使用持续学习时，对模型可预期误差水平的合理基准。结果见表4。我们注意到，在所有情况下，多保真度持续学习方法的表现都优于单保真度持续学习。正如图16所示，加入 MAS 确实改善了结果。持续学习方法的表现确实不如不采用持续学习的情况，这符合预期，因为它们无法同时访问全部训练数据。图17给出了对测试的 λ_{MAS} 的每个取值计算的均方根误差比较。我们注意到



(a) 单保真度



(b) 多保真度

图 12：电池测试案例中，仅采用单保真度 (a) 和多保真度 (b) 训练的结果。左列显示各任务在所有任务上的网络输出，右列显示各任务在其他每个任务上测试得到的均方根误差。本图结果采用窄型架构，包含三个隐藏层，每层有 40 个神经元。

City	无CL	单保真度	多保真度	单保真度，MAS	多保真度，MAS
纽约	16.6289	95.8546	36.8522	38.2728	30.5356
萨克拉门托	2.6952	8.9388	4.4678	5.4253	4.7614
洛杉矶	5.5859	15.1774	9.8254	8.0617	7.0310

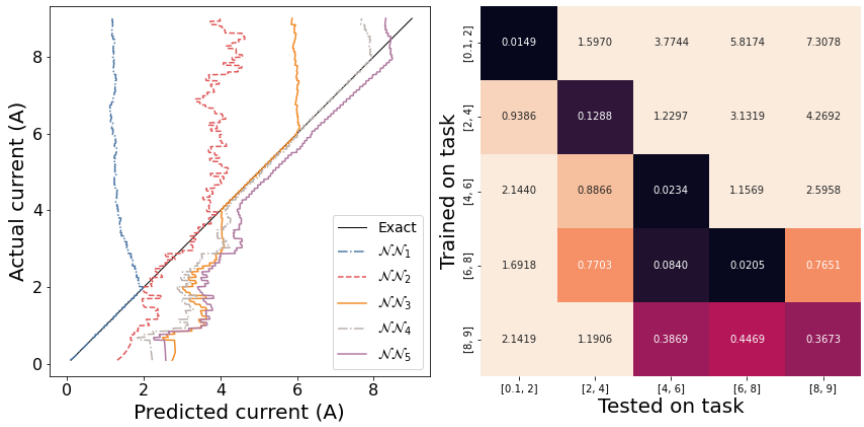
表4：能源消耗案例中，在完整测试区域上的最终输出 $\mathcal{NN}_4$ 的均方根误差（吉瓦时）。对于 MAS 案例，神经网络针对 λ_{MAS} 的七个取值进行训练，上表展示了均方根误差最低的案例。

总体而言，与采用 MAS 的单保真训练相比，采用 MAS 的多保真方法更加稳健，在一系列 λ_{MAS}.中均能获得更小的均方根误差

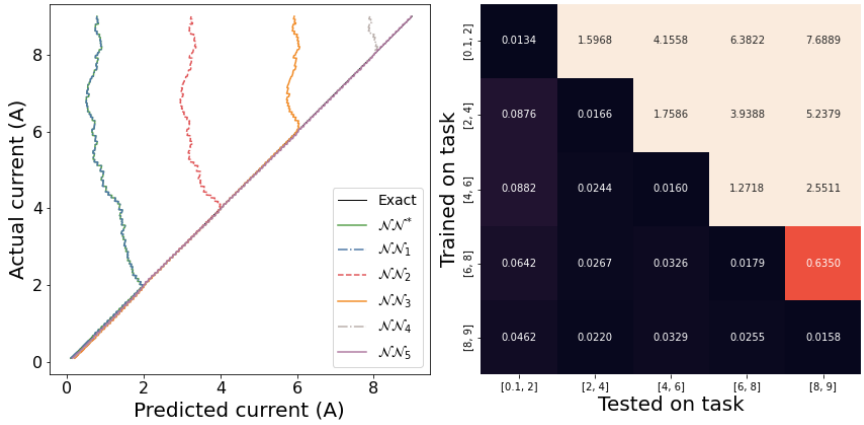
.

5 讨论与未来工作

我们介绍了一种基于多保真度深度神经网络的新型持续学习方法。该方法的前提是，先前训练的模型的输出与模型在当前训练数据集上的期望输出之间存在相关性。发现并利用这些相关性可以减少灾难性遗忘。单独使用时，这种多保真持续学习方法已在多个数据集上展现出稳健性，并有效减少了遗忘



(a) 单保真度



(b) 多保真度

图 13：电池测试案例中，采用 MAS 进行单保真度训练 (a) 和多保真度训练 (b) 的结果。单保真度案例难以实现准确训练，而多保真度的遗忘则非常有限。左列展示了每个任务在所有任务上的网络输出，右列展示了每个任务在其他各任务上测试时的均方根误差。所示结果代表测试过的 MAS 超参数中效果最佳的输出。对于单保真度案例，结果来自使用 $\lambda_{MAS} = 100$ 的窄神经网络；对于多保真度案例，结果来自使用 $\lambda_{MAS} = 0.001$ 的宽神经网络。

适用于基于物理信息和数据驱动训练的示例。此外，它还可以与现有的持续学习方法相结合，包括回放和记忆感知突触（MAS），以进一步减少灾难性遗忘。

所提出的持续学习方法尤其适用于这样的物理问题：不同区域的数据遵循相同的物理定律；也适用于物理信息神经网络，因为在这些情况下，我们预期先前模型的输出与当前训练区域中的模型高度相关。该方法利用不同区域数据之间的相关性，而不是针对每个区域都从头开始训练，因此可以使用规模更小的神经网络在新区域中继续学习。具体而言，其训练准确率对新区域所用神经网络的规模变化更为稳健。这有助于降低训练和推理的计算开销。该方法尤其适用于隐私顾虑可能限制对先前数据集访问的情形。此外，它还可以为联邦学习带来新的可能性，使人们能够设计新的算法，以分布式方式处理传感器数据。这些主题目前仍在研究中，相关结果将在未来的出版物中报告。

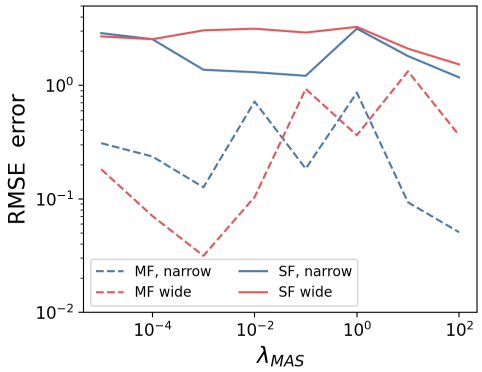


图14：氧化还原液流电池测试案例中使用MAS的均方根误差比较。与单保真度测试案例相比，几乎所有多保真度测试案例的均方根误差都更低。

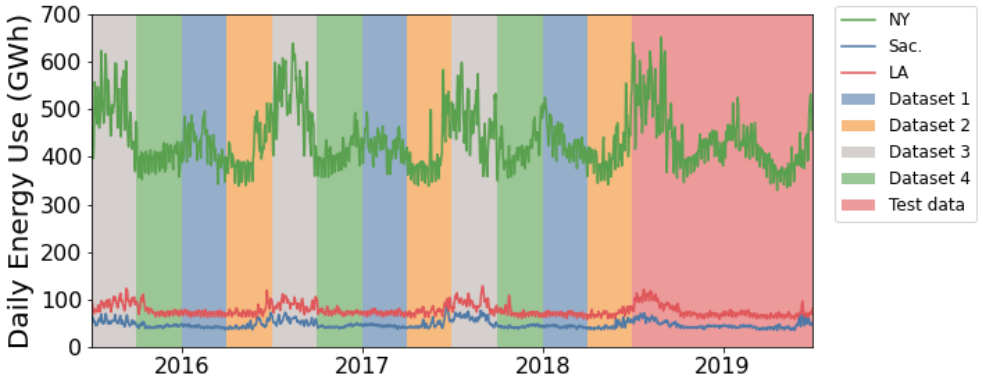


图15：能源消耗示例中所用数据集的示意图。

6 数据可用性

复现这些结果所需的所有代码和数据都将在 https://github.com/pnnl/ Multifidelity_continual_learning/ 上公开。

7 致谢

本研究得到了太平洋西北国家实验室储能材料倡议（ESMI）实验室指导的研究与开发（LDRD）计划的支持。计算工作使用了太平洋西北国家实验室的PNNL机构计算资源。PNNL是由Battelle纪念研究所在美国能源部（DOE）的支持下运营的多项目国家实验室，合同编号为DE-AC05-76RL01830。

参考文献

[1] German I Parisi、Ronald Kemker、Jose L Part、Christopher Kanan 和 Stefan Wermter。基于神经网络的持续终身学习：综述。神经网络，113:54–71，2019。[2] Eli Verwimp、Matthias De Lange 和 Tinne Tuytelaars。复习揭示的真相：在持续学习中重访样本的局限与优点。载于 *IEEE/CVF* 国际计算机视觉会议论文集，第9385–9394页，2021。[3] Friedemann Zenke、Ben Poole 和 Surya Ganguli。通过突触智能实现持续学习。载于国际机器学习会议，第3987–3995页。PMLR，2017。

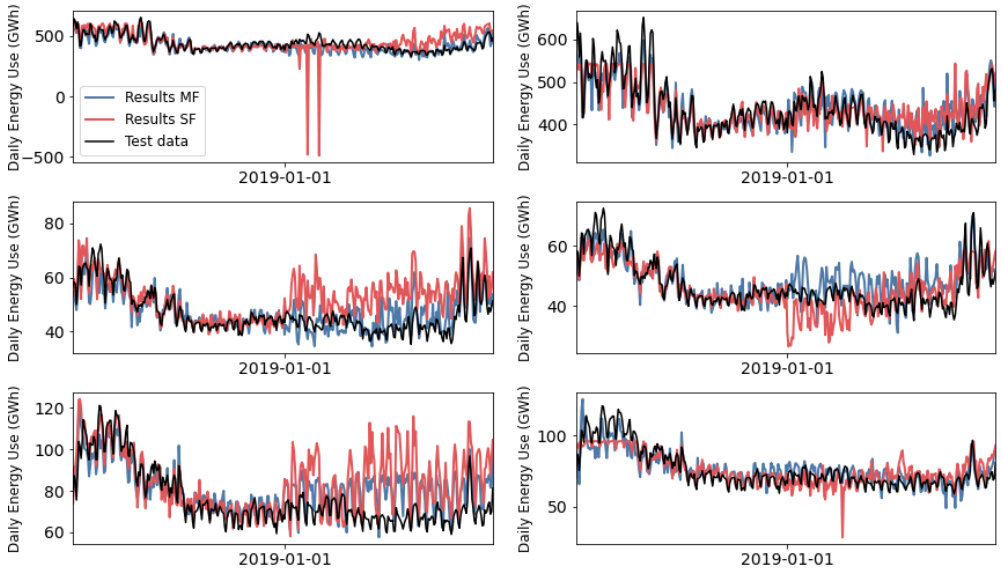


图16：结　　果：能源消耗问题的结果。（左）未使用 MAS 的结果。（右）使用 MAS 的结果　　.

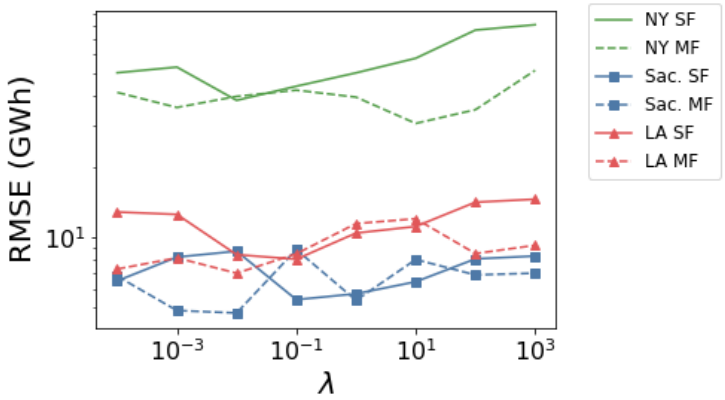


图17：比　　较能源消耗问题中使用 MAS 训练所得的均方根误差。

[4] James Kirkpatrick、Razvan Pascanu、Neil Rabinowitz、Joel Veness、Guillaume Desjardins、Andrei A Rusu、Kieran Milan、John Quan、Tiago Ramalho、Agnieszka Grabska-Barwinska 等。克服神经网络中的灾难性遗忘。美国国家科学院院刊，114(13):3521–3526，2017。[5] Rahaf Aljundi、Francesca Babiloni、Mohamed Elhoseiny、Marcus Rohrbach 和 Tinne Tuytelaars。记忆感知突触：学习该遗忘什么（以及不该遗忘什么）。载于欧洲计算机视觉会议（*ECCV*）论文集，第139–154页，2018。[6] Matthias De Lange、Rahaf Aljundi、Marc Masana、Sarah Parisot、Xu Jia、Aleš Leonardis、Gregory Slabaugh 和 Tinne Tuytelaars。持续学习综述：在分类任务中对抗遗忘。*IEEE*模式分析与机器智能汇刊，44(7):3366–3385，2021。[7] Yen-Chang Hsu、Yen-Cheng Liu、Anita Ramasamy 和 Zsolt Kira。重新评估持续学习场景：分类方法及采用强基线的理由。*arXiv*预印本 *arXiv:1810.12488*，2018。[8] Andrei A Rusu、Neil C Rabinowitz、Guillaume Desjardins、Hubert Soyer、James Kirkpatrick、Koray Kavukcuoglu、Razvan Pascanu 和 Raia Hadsell。渐进式神经网络。*arXiv*预印本 *arXiv:1606.04671*，2016。[9] Yeming Wen、Dustin Tran 和 Jimmy Ba。Batchensemble：一种高效集成与终身学习的替代方法。*arXiv*预印本 *arXiv:2002.06715*，2020。

[10] Jonas Pfeiffer、Andreas Rücklé、Clifton Poth、Aishwarya Kamath、Ivan Vuli´c、Sebastian Ruder、Kyunghyun Cho 和 Iryna Gurevych。AdapterHub：用于适配 Transformer 的框架。arXiv 预印本 arXiv:2007.07779，2020。[11] Leonard Bereska 和 Efstratios Gavves。利用竞争式联邦储层计算对动力系统进行持续学习。载于《终身学习智能体会议》，第 335–350 页。PMLR，2022。[12] Tsendsuren Munkhdalai 和 Hong Yu。元网络。载于《国际机器学习会议》，第 2554–2563 页。PMLR，2017。[13] Max Vladymyrov、Andrey Zhmoginov 和 Mark Sandler。使用超Transformer进行持续少样本学习。arXiv 预印本 arXiv:2301.04584，2023。[14] George Em Karniadakis、Ioannis G Kevrekidis、Lu Lu、Paris Perdikaris、Sifan Wang 和 Liu Yang。物理信息机器学习。《自然综述：物理学》，3(6):422–440，2021。[15] Nathan Baker、Frank Alexander、Timo Bremer、Aric Hagberg、Yannis Kevrekidis、Habib Najm、Manish Parashar、Abani Patra、James Sethian、Stefan Wild 等。科学机器学习基础研究需求研讨会报告：人工智能的核心技术。技术报告，美国能源部科学办公室（SC），华盛顿特区（美国），2019。[16] 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。《科学计算杂志》，92(3):88，2022。[17] Xiaowei Jin、Shengze Cai、Hui Li 和 George Em Karniadakis。NSFnets（Navier–Stokes 流动网络）：用于不可压缩 Navier–Stokes 方程的物理信息神经网络。《计算物理学杂志》，426:109951，2021。[18] Maziar Raissi、Alireza Yazdani 和 George Em Karniadakis。隐流体力学：从流场可视化中学习速度场和压力场。《科学》，367(6481):1026–1030，2020。[19] Shengze Cai、Zhiping Mao、Zhicheng Wang、Minglang Yin 和 George Em Karniadakis。用于流体力学的物理信息神经网络（PINN）：综述。《力学学报》，37(12):1727–1738，2021。[20] Archis S Joglekar 和 Alexander G R Thomas。多尺度流体模拟中隐变量的机器学习。《机器学习：科学与技术》，4(3):035049，2023 年 9 月。[21] Dehao Liu 和 Yan Wang。多保真度物理约束神经网络及其在材料建模中的应用。《机械设计杂志》，141(12)，2019。[22] Yuyao Chen、Lu Lu、George Em Karniadakis 和 Luca Dal Negro。用于纳米光学和超材料逆问题的物理信息神经网络。《光学快报》，28(8):11618–11633，2020。[23] Zhiwei Fang 和 Justin Zhan。用于超材料设计的深度物理信息神经网络。《IEEE开放期刊》，8:24506–24513，2019。[24] Zhiping Mao、Ameya D Jagtap 和 George Em Karniadakis。用于高速流动的物理信息神经网络。《应用力学与工程中的计算方法》，360:112789，2020。[25] George S Misyris、Andreas Venzke 和 Spyros Chatzivasileiadis。用于电力系统的物理信息神经网络。载于 2020 年 IEEE 电力与能源学会大会（PESGM），第 1–5 页。IEEE，2020。[26] Bin Huang 和 Jianhui Wang。物理信息神经网络在电力系统中的应用综述。《IEEE电力系统汇刊》，38(1):572–588，2022。[27] Christian Moya 和 Guang Lin。DAE-PINN：一种用于模拟微分代数方程的物理信息神经网络模型及其在电力网络中的应用。《神经计算与应用》，35(5):3789–3804，2023。[28] Maziar Raissi、Paris Perdikaris 和 George 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逆问题的深度学习框架。《计算物理学杂志》，378:686–707，2019。[29] Sifan Wang 和 Paris Perdikaris。使用物理信息 DeepONets 对参数化演化方程进行长时间积分。《计算物理学杂志》，475:111855，2023。[30] Revanth Mattey 和 Susanta Ghosh。一种用于训练物理信息神经网络求解 Allen–Cahn 和 Cahn–Hilliard 方程的新型序贯方法。《应用力学与工程中的计算方法》，390:114474，2022。[31] Aleksandr Dekhovich、Marcel HF Sluiter、David MJ Tax 和 Miguel A Bessa。iPINNs：物理信息神经网络的增量学习。arXiv 预印本 arXiv:2304.04854，2023。

[32] Majid Rasht-Behesht、Christian Huber、Khemraj Shukla 和 George Em Karniadakis。用于波传播和全波形反演的物理信息神经网络（物理信息神经网络）。《地球物理研究杂志：固体地球》，127(5):e2021JB023120，2022。[33] Xuhui Meng 和 George Em Karniadakis。能够从多保真度数据中学习的复合神经网络：应用于函数逼近和偏微分方程逆问题。《计算物理学杂志》，401:109020，2020。[34] Frederik Benzing。持续学习的正则化方法统一框架。*arXiv* 预印本 *arXiv:2006.06357*，2020。[35] Sifan Wang、Xinling Yu 和 Paris Perdikaris。物理信息神经网络何时以及为何训练失败：神经切线核视角。《计算物理学杂志》，449:110768，2022。[36] Xuhui Meng、Zhen Li、Dongkun Zhang 和 George Em Karniadakis。并行物理信息神经网络：用于含时偏微分方程的并行时间积分物理信息神经网络。《应用力学与工程中的计算方法》，370:113250，2020。[37] Colby L Wight 和 Jia Zhao。使用自适应物理信息神经网络求解 Allen-Cahn 方程和 Cahn-Hilliard方程。*arXiv* 预印本 *arXiv:2007.04542*，2020。[38] Franz M Rohrhofer、Stefan Posch、Clemens Gößnitzer 和 Bernhard C Geiger。动力系统不动点在物理信息神经网络训练中的作用。*arXiv* 预印本 *arXiv:2203.13648*，2022。[39] Yunxiang Chen、Jie Bao、Zhijie Xu、Peiyuan Gao、Litao Yan、Soowhan Kim 和 Wei Wang。用于氧化还原液流电池评估与优化的二维解析单元模型。《电源杂志》，506:230192，2021。[40] Yunxiang Chen、Zhijie Xu、Chao Wang、Jie Bao、Brian Koeppel、Litao Yan、Peiyuan Gao 和 Wei Wang。氧化还原液流电池设计的解析建模。《电源杂志》，482:228817，2021。[41] Zhe Wang、Tianzhen Hong、Han Li 和 Mary Ann Piette。使用数据驱动模型预测城市尺度日用电量。《应用能源进展》，2:100025，2021。

8 附录

本节报告用于训练并得出上述结果的训练参数。

	Sec. 3.1	Sec. 3.2	Sec. 4.1	Sec. 4.2
SF $\mathcal{NN}_1$ learning rate	(1e-3, 2000, 0.95)	(1e-4, 2000, 0.99)	(1e-3, 2000, 0.9)	(1e-3, 2000, 0.99)
SF $\mathcal{NN}_i$, $i > 1$ learning rate	(1e-3, 2000, 0.95)	(1e-4, 2000, 0.99)	(1e-3, 2000, 0.95)	(1e-3, 2000, 0.99)
SF activation function	`swish`	`tanh`	`relu`	`tanh`
SF architecture	[1, 100, 100, 100, 100, 100, 2]	[2, 200, 200, 200, 200, 200, 1]	[784, 80, 80, 1] or [784, 40, 40, 40, 1]	[5, 100, 100, 100, 1]
SF batch size	100	500	10	100
SF boundary condition batch size	1	100	–	–
SF iterations	50,000	100,000	100,000	100,000
MF $\mathcal{NN}^*$ learning rate	(1e-3, 2000, 0.99)	(1e-4, 2000, 0.99)	(1e-3, 2000, 0.9)	(1e-3, 2000, 0.99)
MF $\mathcal{NN}^*$ activation function	`swish`	`tanhtanh`	`relu`	`tanh`
MF $\mathcal{NN}^*$ architecture	[1, 200, 200, 200, 1]	[2, 200, 200, 200, 200, 200, 1]	[784, 40, 40, 40, 1]	[5, 100, 100, 100, 1]
MF $\mathcal{NN}^*$ iterations	50,000	50,000	100,000	100,000
MF batch size	200	1000	10	100
MF boundary condition batch size	1	100	–	–
MF $\mathcal{NN}_i$ learning rate	(1e-3, 2000, 0.99)	(5e-4, 2000, 0.99)	(1e-3, 2000, 0.95)	(1e-3, 2000, 0.99)
MF $\mathcal{NN}_i$ activation function	`swish`	`tanh`	`relu`	`tanh`
MF $\mathcal{NN}_i$ nonlinear architecture	[3, 100, 100, 100, 100, 100, 2]	[3, 200, 200, 200, 200, 200, 1]	[785, 80, 80, 1] or [785, 40, 40, 40, 1]	[6, 100, 100, 100, 1]
MF $\mathcal{NN}_i$ linear architecture	[2, 20, 2]	[1, 20, 1]	[1, 1]	[1, 5, 1]
MF $\mathcal{NN}_1$ iterations	50,000	50,000	100,000	100,000
MF $\mathcal{NN}_i$, $i > 1$ iterations	100,000	100,000	100,000	100,000
MF MAS samples	1000	3000	400	1201
SF MAS samples	1000	3000	400	1201
No CL learning rate	(1e-3, 2000, 0.95)	(1e-4, 2000, 0.99)	–	(1e-3, 2000, 0.99)
No CL activation function	`swish`	`tanh`	–	`relu`
No CL architecture	[1, 200, 200, 200, 1]	[2, 200, 200, 200, 200, 200, 1]	–	[5, 40, 40, 40, 1]
No CL batch size	200	500	–	100
No CL boundary condition batch size	1	100	–	–
No CL iterations	50,000	100,000	–	100,000
λ_{bc}	–	1	–	–
λ_{ic}	1	100	–	–
λ_r	10	1	–	–
λ_{data}	0	0	1	1
λ	1e-4	1e-5	1e-4	1e-5

Table 5: Hyperparameters for training the results in this paper. SF refers to the single fidelity results. MF refers to the multifidelity results. For the multifidelity results, $\mathcal{NN}^*$ denotes the first single fidelity network in the multifidelity framework, and $\mathcal{NN}_i$ denotes the multifidelity neural networks. For the learning rate, the triplet (a, b, c) denotes the `exponential_decay` function in Jax with learning rate a, decay steps b, and decay rate c. `relu` is the rectified linear unit (ReLU) activation function