arXiv:2301.02152v1 [cs.LG] 5 Jan 2023

L-HYDRA：多头物理信息神经网络

ZONGREN ZOU* 和GEORGE EM KARNIADAKIS†

摘要。 我们将多头神经网络（MH-NNs）引入物理信息机器学习。这是一类神经网络（NNs），其所有非线性隐藏层构成主体，多个线性输出层则构成多头。因此，我们构建了多头物理信息神经网络（MH-PINNs），将其作为强大的工具，用于科学机器学习（SciML）中的多任务学习（MTL）、生成建模和少样本学习，以解决各种问题。MH-PINNs通过共享主体（作为基函数）以及头部共享的分布，将多个函数/任务联系起来。前者通过使用MH-PINNs求解多个任务来实现，每个头部独立对应一个任务；后者则通过使用归一化流（NFs）进行密度估计和生成建模来实现。为此，我们的方法分为两个阶段，两个阶段均可借助标准的神经网络深度学习工具来处理，因此易于在实践中实现。MH-PINNs可用于多种目的，例如近似随机过程、协同求解多个任务、为元学习和迁移学习等下游少样本学习任务提供有价值的先验知识、学习具有代表性的基函数，以及进行不确定性量化。我们在五个基准测试中展示了MH-PINNs的有效性，并探究了其在回归分析中进行协同学习的可能性。我们将开源代码命名为“勒拿九头蛇”（L-HYDRA），因为这种神话生物拥有许多头部，能够执行多个重要任务，这与我们提出的方法相似。

关键词。物理信息神经网络、元学习、多任务学习、迁移学习、生成模型、正态化流、随机问题

数学学科分类代码。34F05, 62M45, 65L99, 65M99, 65N99

1. 引言。近年来，跨任务学习在深度学习领域备受关注，也是科学机器学习（SciML）中一个新兴的研究主题，因为若干类科学问题具有相似性，和/或因共同的物理规律而内在相关。直观而言，如果任务彼此相似，例如在近似随机过程 [44], 、学习常微分方程/偏微分方程（ODEs/PDEs）的解算子 [28], 以及求解参数化偏微分方程[42, 19, 4], 等情境中，那么在建模、算法设计和/或求解过程中将它们关联起来可能会有所裨益。在这方面，过去几年迅速发展的机器学习求解器，相较于传统数值求解器具有更高的灵活性和更大的潜力。该领域已取得显著进展，包括用于求解常微分方程/偏微分方程的元学习 [30, 27, 34, 6], 、物理信息神经网络（PINNs）的迁移学习 [3, 7], 、求解偏微分方程时应对领域偏移的迁移学习 [14], 、面向物理信息神经网络的多任务学习 [40], ，以及用于求解随机微分方程（SDEs）的生成方法 [44, 46, 15]。近年来，算子学习 [28, 24] 备受关注；这种方法直接学习算子映射，随后以单次学习的形式将其用于其他任务。

多头神经网络（MH-NNs）非常适合各种跨任务学习场景。它们最初作为硬参数共享神经网络（NNs）的一员，为深度多任务学习（MTL）而提出 [5], ，其中多个任务（记为 $\mathcal{T}_k, k = 1, ..., M$， M为任务总数）会被同时解决。使用多头神经网络开展多任务学习的总体目标多种多样：实现

*布朗大学应用数学系，美国罗得岛州普罗维登斯 02912（zon- gren zou@brown.edu）。

†通讯作者。布朗大学应用数学系，美国罗得岛州普罗维登斯 02912（george karniadakis@brown.edu）。

所有任务更好的性能，为下游任务学习良好且有用的表示，和/或借助辅助任务促进主要任务的学习。此外，尽管多头神经网络最初是为解决多个任务而设计的，但近年来也被广泛用于元学习。例如， [41],分析了多任务学习与元学习之间的联系，并讨论了用于多头神经网络的元学习算法； [25], 表明，以多任务学习方式训练的多头神经网络在元学习中也能进行任务特定的适应； [37] 认为，模型无关元学习 [10], 这一知名的元学习算法之所以有效，可能是因为成功学习了良好的表示，而不是学会了适应，并使用多头神经网络研究神经网络在快速任务适应中的具体贡献。总的来说，文献普遍认为，多头神经网络在用于解决既有任务时，能够提炼有用的共享信息，并将其存储在自身的主体和头部中。

本文开发了用于物理信息机器学习的多头神经网络（MH-NNs） [17],，提出了多头物理信息神经网络（MH-PINNs），并进一步探讨其在多任务学习、生成建模和元学习中的适用性与能力。如图 1 所示，MH-PINN 构建于传统的多头神经网络之上，由两个主要部分组成：主体 和多个头部，每个头部都连接到一个特定的常微分方程/偏微分方程任务。在不同应用场景中，多头神经网络采用了多种架构拆分策略；例如，对于某些计算机视觉问题，神经网络会拆分为由卷积层构成的主体，以及接在主体之后、充当头部的全连接层。然而，本文选择最简单的策略，即主体由所有非线性层构成，而头部是最后一个线性层，理由有二：（1）头部的维度较低，从而能够快速进行密度估计（见下一节）；（2）主体自然地提供了一组基函数。

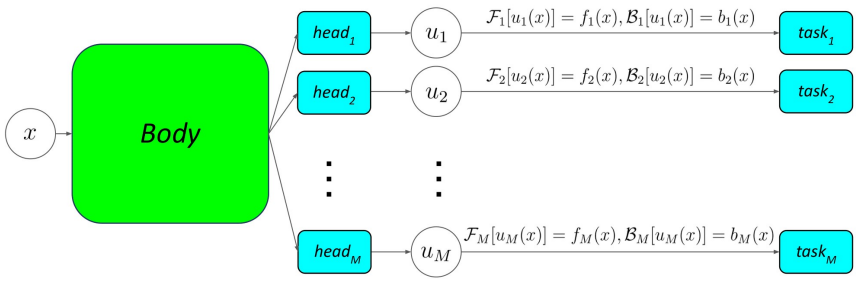


FIG. 1. *Schematic view of the structure of multi-head physics-informed neural networks (MH-PINNs) with M different heads, which are built upon conventional multi-head neural networks. The shared layers are often referred to as body and the task-specific layer as head. Generally, u_k , $k = 1,\ldots,$ M represent M solutions to M different ODEs/PDEs, formulated in Eq.* (2.1) *whichmay differ in source terms f_k, boundary/initial condition terms b_k, or differential operator $\mathcal{F}_k$.*

本研究的创新之处和主要贡献如下：

1. 我们提出了一种新的物理信息生成方法，利用 MH-PINNs 从数据和物理规律中学习随机过程。2. 我们提出了一种利用 MH-PINNs 解决具有不确定性量化的物理信息小样本回归问题的新方法。3. 我们研究并展示了在回归问题中，利用多头神经网络进行多任务学习和协同学习的有效性。

本文组织如下。第 2 节介绍问题的表述、

MH-PINNs 的细节以及总体方法，包括如何将 MH-PINNs 用于多任务学习、生成建模，以及带不确定性量化（UQ）的下游少样本物理信息学习。第 3 节讨论与本文密切相关的现有研究，并从概念层面进行比较。第 4 节使用五个基准测试检验 MH-PINNs，每个基准测试对应一种或多种学习目的，例如多任务学习和生成建模。第 5 节通过函数逼近示例研究多任务学习和协同学习。我们在第 6 节总结全文。实验详情，例如神经网络架构和训练策略，可参见附录 A 和 B，以及 L-HYDRA 在 GitHub 上的开源代码；论文被接收后，这些代码将予以发布。

2. 方法论。 我们假设存在一系列任务 $\{\mathcal{T}_k\}_{k=1}^M$，每个任务都与数据 $\mathcal{D}_k, k = 1, ..., M$相关。本文主要关注科学计算和常微分方程/偏微分方程，因此我们进一步假设 $\{\mathcal{T}_k\}_{k=1}^M$ 是物理信息回归问题 [17]。

考虑以下形式的偏微分方程：

$$\mathcal{F}_k[u_k(x)] = f_k(x), x \in \Omega_k, \tag{2.1a}$$
$$\mathcal{B}_k[u_k(x)] = b_k(x), x \in \partial\Omega_k, \tag{2.1b}$$

其中，k 表示任务的索引，$k = 1, ..., M$，x 表示 D_x 维的一般时空坐标，Ω_k 为有界域，f_k 和 u_k 分别为该偏微分方程的D_u维源项和解，$\mathcal{F}_k$ 为一般微分算子，$\mathcal{B}_k$ 为一般边界/初始条件算子，b_k 为边界/初始条件项。为简便起见，本文假设所有任务的定义域和边界/初始条件算子（分别记为 Ω 和 $\mathcal{B}$）均相同，而解 u_k 则是任务特定的。任务 $\mathcal{T}_k$ 描述为根据数据 $\mathcal{D}_k$ 和方程（2.1）近似 u_k和/或 f_k和/或 $\mathcal{F}_k$和/或 b_k。

传统数值求解器通常独立处理 $\{\mathcal{T}_k\}_{k=1}^M$，而不会利用或迁移不同任务之间的知识。物理信息神经网络方法 [38] 旨在使用神经网络独立求解常微分方程/偏微分方程，但这会产生 M 彼此不相关的结果。与此不同，本文将 $\{\mathcal{T}_k\}_{k=1}^M$ 视为一个整体，并通过 MH-PINN 将它们联系起来。其架构如图 1 所示，能够在解 u_k的预测中强制共享基函数。除了信息丰富的表示/主体之外，我们还通过假设 MH-PINN 中对应的头部（记为 $\{H_k\}_{k=1}^M$）是一个随机变量的样本，来建立 $\{\mathcal{T}_k\}_{k=1}^M$ 之间的联系；该随机变量的概率密度函数（PDF）未知，分别记为 H 和 $p(H)$。共享主体与 H 的生成模型立即构成了解 u的生成模型；此外，将 u 代入方程（2.1），并通过自动微分 [1], ，还可得到源项 f 和边界/初始项 b 的生成器。由此，我们无缝地构建出一种用于近似随机过程的生成方法。

如 [30], 所述，u、f 和 b的生成器能够在物理信息贝叶斯推断 [43, 26] 以及科学机器学习的不确定性量化 [47, 36], 中提供信息先验分布；该信息先验能够弥补观测数据不足的问题，从而在只有少量含噪测量值时，也能应对物理信息学习问题。本文将此类问题推广到确定性情形，此时数据无噪，方法和结果均为确定性的，并将其称为少样本物理信息学习。其基本思路是将通过把 $\{\mathcal{T}_k\}_{k=1}^M$ 与多头物理信息神经网络相结合而学到的先验知识应用于新任务（记为$\tilde{\mathcal{T}}$），这些任务的数据不足 $\mathcal{D}$，旨在获得准确的 和可信的 预测。

学习框架的示意图如图2所示，具体细节将在下文说明。

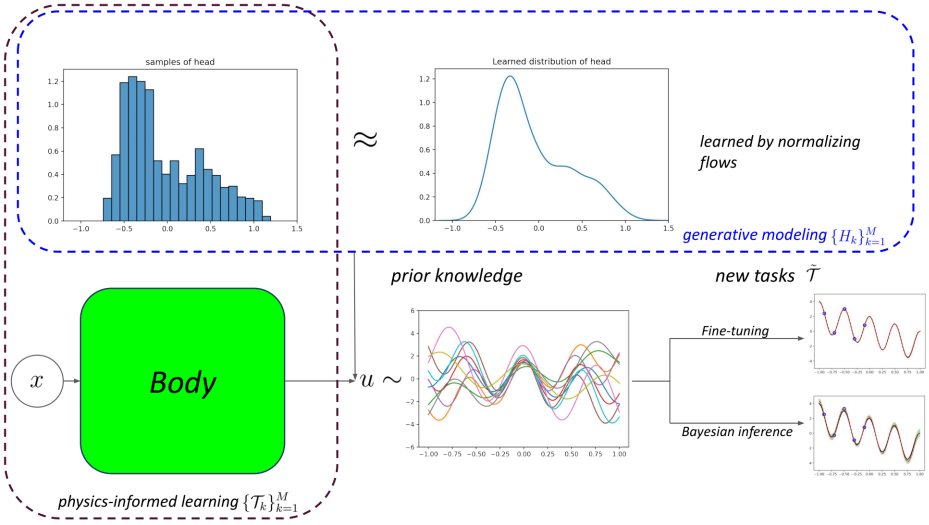


FIG. 2。 *Schematic view of the learning framework and the proposed method. Three general types of learning are addressed: physics-informed learning , generative modeling , and few-shot learning. The physics-informed learning is performed with MH-PINNs; the generative modeling is done afterwards by density estimate over the head via normalizing flows (NFs); in the end the few-shot physics-informed learning is accomplished with prior knowledge obtained from previous tasks via either fine-tuning with the learned regularization or Bayesian inference with the learned prior distribution.*

The body represents the set of basis functions learned from solving $\{\mathcal{T}_k\}_{k=1}^M$ with MH-PINNs, and the density of the head, estimated from its samples using NFs, acts as the regularization, the prior distribution or the generator together with the body depending on the usage of MH-PINNs in applications.

2.1 多头物理信息神经网络（MH-PINNs）。在考虑使用神经网络进行多任务学习（MTL）时，硬参数共享是最常用的方法，而多头神经网络（MH-NNs）作为其最简单的实例，也被频繁采用 [5, 39]。如前文所述，一个 MH-PINN 由一个主体和多个头部组成。我们将 Φ 称为主体，将 H_k 称为 $\mathcal{T}_k$的头部。注意，此处 Φ： $\mathbb{R}^{D_x} \to \mathbb{R}^L$ 是一个由参数为 θ的神经网络参数化的函数，而 $H_k \in \mathbb{R}^{L+1}$ 是一个向量，其中 L 是主体最后一层的神经元数量。定义$H_k = [h_k^0, h_k^1, ..., h_k^L]^T$、 $\Phi(x) = [\phi^1(x) ....., \phi^L(x)]^T$，其中 ϕ： $\mathbb{R}^{D_x} \to \mathbb{R}$，则 $\mathcal{T}_k$ 中解的代理模型可改写为 $\hat{u}_k(x) = h_k^0 + \sum_{l=1}^L h_k^l \phi^l(x) \quad \forall x \in \Omega$。相应地，近似的源项和边界项/初始项由方程（2.1）导出。在 MTL 框架中，给定数据 $\{\mathcal{D}_k\}_{k=1}^M$ 和物理方程（2.1），损失函数 $\mathcal{L}$ 可表述如下：

$$\mathcal{L}(\{\mathcal{D}_k\}_{k=1}^M; \theta, \{H_k\}_{k=1}^M) = \frac{1}{M} \sum_{k=1}^M \mathcal{L}_k(\mathcal{D}_k; \theta, H_k), \tag{2.2}$$

其中， $\mathcal{L}_k$ 表示物理信息神经网络中的常见损失函数。通常，$\mathcal{T}_k$的数据表示为 $\mathcal{D}_k = \{\mathcal{D}_k^f, \mathcal{D}_k^b, \mathcal{D}_k^u\}$，其中 $\mathcal{D}_k^f = \{x_k^i, f_k^i\}_{i=1}^{N_k^f}$、 $\mathcal{D}_k^b = \{x_k^i, b_k^i\}_{i=1}^{N_k^b}$和

$\mathcal{D}_k^u = \{x_k^i, u_k^i\}_{i=1}^{N_k^u}$、以及 $\mathcal{L}_k$ 如下：

$$\begin{aligned}\mathcal{L}_k(\mathcal{D}_k;\theta,H_k) =& \frac{w_k^f}{N_k^f}\sum_{i=1}^{N_k^f}||\mathcal{F}_k(\hat{u}_k(x_k^i)) - f_k^i||^2 + \frac{w_k^b}{N_k^b}\sum_{i=1}^{N_k^b}||\mathcal{B}(\hat{u}_k(x_k^i)) - b_k^i||^2 \\ &+ \frac{w_k^u}{N_k^u}\sum_{i=1}^{N_k^u}||\hat{u}_k(x_k^i) - u_k^i||^2 + \mathcal{R}(\theta,H_k),\end{aligned} \tag{2.3}$$

其中，$||\cdot||$ 表示适当选取的范数，$\mathcal{R}(\cdot)$是针对神经网络参数的正则化方法，N_k^f、N_k^b、N_k^u分别表示 f_k、b_k、u_k的数据点数量，而w_k^f、w_k^b、w_k^u则是用于平衡损失函数中不同项的权重。

2.2. 生成建模与归一化流（NFs）。如前所述，MH-PINNs通过以下两个假设将 $\{\mathcal{T}_k\}_{k=1}^M$ 联系起来：(1) 解u_k、$k=1$、..... M共享同一组基函数 Φ；(2) 相应的系数是同一个随机变量 H的样本。在 [7]中，Φ 被用作从 $\{\mathcal{T}_k\}_{k=1}^M$ 向下游物理信息学习任务传递先验知识的载体。在本文中，我们通过利用头部中的信息对其进行扩展：使用归一化流（NFs）从其样本 $\{H_k\}_{k=1}^M$估计 H 的概率密度函数和生成器。有关NFs的综述，感兴趣的读者可参阅 [32, 22] ；有关一些流行NFs的发展，可参阅[9, 33, 20] 。

与其他常用的生成模型相比，例如生成对抗网络（GANs） [13], 变分自编码器（VAEs） [21], 或扩散模型 [16], ，我们选择正规化流（NFs），因为它既是密度估计器，也是生成器。前者能够为下游的小样本物理信息学习任务提供适当的正则化，后者则可用于构建物理信息生成方法，以近似随机过程。值得注意的是，在以往关于物理信息生成方法的研究中 [44, 46, 15], ，神经网络（NNs）通过 u_k和/或 f_k和/或 b_k的测量值进行训练。相比之下，我们的模型通过头部的样本进行学习，而该头部是在第一步中通过多任务学习（MTL）获得的。这种学习策略带来了两项显著优势：（1）能够灵活处理非结构化数据，例如任务间测量值不一致的情况；（2）通过将物理信息学习与生成建模解耦，使训练更简单且易于控制。

2.3. 下游任务中利用的先验知识。本文将介绍如何利用存储于 MH-PINNs 中的先验知识，用于下游˜少样本物理信息学习任务 $\mathcal{T}$。该任务的定义方式与上游训练中的所有其他任务相同，但测量值要少得多。MH-PINNs 和 NFs 的训练会得到一个主体 Φ、头部样本 $\{H_k\}_{k=1}^M$，以及头部的估计概率密度函数˜ ˜ˆ$p(H) \approx p(H)$。在使用 $\mathcal{D}$求解 $\mathcal{T}$时，我们固定主体 Φ ，并寻找能够最好地解释数据 $\mathcal{D}$ 以及方程（2.1）中的物理规律的头部˜ ˜H 。本文考虑无噪声数据和含噪数据两种情况：对于无噪声数据，在新任务中对头部进行常规神经网络训练，以提供确定性预测，此时头部的学习到的概率密度函数ˆ$p(H)$作为损失函数中的正则化项；对于含噪数据，还会对头部进行贝叶斯推断，其中ˆ$p(H)$表示先验分布。具体细节将在下文介绍。

2.3.1. 优化中的正则化。少样本学习中的数据有限，往往会导致过拟合和/或插值/外推性能不佳。在这方面，根据头部的概率密度函数对其进行正则化，能够防止过拟合并

为提升插值/外推性能提供额外的先验知识。该优化问题可表示为

$$\tilde{H}^* = \arg\min_{\tilde{H}} \mathcal{L}^*(\tilde{\mathcal{D}}; \tilde{H}), \text{where } \mathcal{L}^*(\tilde{\mathcal{D}}; \tilde{H}) = \mathcal{L}(\tilde{\mathcal{D}}; \tilde{H}) - \alpha \log p(\tilde{H}) \approx \mathcal{L}(\tilde{\mathcal{D}}; \tilde{H}) - \alpha \log \hat{p}(\tilde{H}), \tag{2.4}$$

其中，$\mathcal{L}$ 是针对数据 $\mathcal{D}$ 和参数 $\tilde{H}$ 的物理信息学习中的正则损失函数，$\alpha \geq 0$ 是用于调节正则化效应的系数。本文通过梯度下降求解问题（2.4）。

2.3.2. 贝叶斯推断中的先验分布。与通过求解优化问题（2.4）获得的点估计相反，$\mathcal{T}$ 中头部的后验分布 $\tilde{}$ 通过贝叶斯推断获得。与 [30, 47], 类似，H 的后验分布建立如下：

$$p(\tilde{H}|\tilde{\mathcal{D}}) \propto p(\tilde{\mathcal{D}}|\tilde{H})p(\tilde{H}) \approx p(\tilde{\mathcal{D}}|\tilde{H})\hat{p}(\tilde{H}), \tag{2.5}$$

其中，$p(H|\mathcal{D})$ 是后验分布，$p(\mathcal{D}|H)$ 是似然分布，$\tilde{}$ 通常假设其在 $\mathcal{D}$ 的所有测量值上服从独立高斯分布，而 $\hat{p}$ 是通过正规化流得到的头部的估计概率密度函数。分布（2.5）的不可解析性要求采用近似方法，其中马尔可夫链蒙特卡罗方法（例如哈密顿蒙特卡洛（HMC））[31], 通常能够提供最准确的估计。此外，H 的维度相对较低，也使得可以使用拉普拉斯近似（LA）[18], ；本文将其作为 HMC 的替代方法。

3. 相关工作。近年来，深度神经网络已被广泛用于求解常微分方程/偏微分方程、随机微分方程以及算子学习。虽然未被明确称为 MH-PINN，但 MH-NN 首先由 [7], 用于求解常微分方程/偏微分方程：该方法先在多个相似任务上对 MH-PINN 进行预训练，随后舍弃头部、保留主体，并将其迁移至新任务；对于线性常微分方程/偏微分方程，采用最小二乘估计；对于非线性问题，则通过梯度下降进行微调。[7], 提出了一种用于线性问题的单次迁移学习算法，但与本文的工作不同，该研究并未讨论 MH-NN 的其他潜在用途，例如多任务学习和生成建模。此外，[7] 仅关注在充足的干净数据下实现快速、确定且准确的预测；而本文还研究了 MH-NN 在小样本物理信息学习中的适用性，在这种情形下，数据不足和/或含有噪声，并通过不确定性量化处理这些情况。值得注意的是，[45],也将 MH-NN 用作多输出神经网络，但该研究侧重于求解单个任务并获取不确定性。

科学计算背景下的生成建模近期也得到了研究，并且已有少数尝试将深度生成神经网络应用于科学机器学习问题，其中大多数尝试集中于近似随机过程和求解随机微分方程。我们提出了一种新的物理信息生成方法，作为现有方法的替代方案，该方法使用多头物理信息神经网络，并对其在近似随机过程方面的表现进行了测试。在这方面，我们的方法在功能上与现有方法相同，但在技术上有所不同。以往所有方法都通过端到端学习策略，将物理信息学习和生成建模这两种不同类型的学习结合起来，以实现物理信息生成建模；当任一类型的学习变得更加复杂时，这可能会给实际实施和使用带来问题。另一方面，我们的方法从一个全新的角度处理这一问题，即

通过解耦这两者：首先执行物理信息学习，随后学习生成器。为此，我们的方法分为两个步骤；借助两个领域中成熟的算法，该方法在实施的灵活性和简便性方面都具有优势。

4. 结果。在本节中，我们使用五个基准测试来检验所提出的方法。第一个是用于教学演示的函数回归，旨在展示该方法的基本适用性和能力，并说明在下游任务中纳入头部的分布，以及获得带有或不带不确定性的结果的重要性。第二个示例是一个非线性常微分方程系统，我们在其中检验该方法通过微分算子逼近随机过程的能力，比较不同的正规化流，最终将该方法与另一种知名的物理信息生成模型——物理信息生成对抗网络（PI-GANs）[44] 在生成建模方面进行比较。第三个示例是一维非线性反应扩散方程，第四个是二维非线性 Allen-Cahn 方程，第五个则是具有 20 个维度的二维随机亥姆霍兹方程。除非另有说明，在所有示例中，$\{\mathcal{D}_k\}_{k=1}^M$ 的数据均无噪声且对任务而言充足，而下游任务中的 $\tilde{\mathcal{D}}$ 则并不充足，因此这些下游任务属于少样本类型。此外，除第一个示例外，贝叶斯推断的结果均通过使用 HMC 得出；预测均值（记为 μ）和预测标准差（记为 σ）则根据函数或未知参数的后验样本计算得出。本文将预测不确定性定义为 2σ。

4.1. 函数逼近。我们首先考虑一个仅使用数据而不涉及物理信息的函数回归问题。这是方程（2.1）的一个退化实例，其中 $\mathcal{F}_k$固定为恒等算子，不包含 $\mathcal{B}$ 和 b_k，且 $u_k = f_k$ 是任务特定的。在此情况下，$\mathcal{D}_k$和 $\tilde{\mathcal{D}}$分别表示为 $\{(x_k^i,\ f_k^i)\}_{i=1}^{N_k}$和 $\{(x^i,\ f^i)\}_{i=1}^{N}$，而 $\mathcal{T}_k$和 $\mathcal{T}$分别被定义为从 $\mathcal{D}_k$ 和 $\mathcal{D}$逼近函数 f_k 和 f。本例中的随机函数 f 定义如下：

$$
\begin{aligned}
&f(x) = A\cos(\omega x) + 2\beta x, x \in [-1, 1],\\
&A \sim U[1, 3), \omega \sim U[2\pi, 4\pi), \mathcal{P}(\beta = \pm 1) = 0.5,
\end{aligned}
\tag{4.1}
$$

其中，U 表示均匀分布，$\mathcal{P}(\Xi)$定义为事件 Ξ发生的概率。我们的目标是利用多头神经网络和正规化流从数据中逼近 f，并同时求解下游少样本回归任务 $\mathcal{T}$；在这些任务中，两个函数 $2\cos(2\pi x)+2x$ 和 $2\cos(4\pi x)-2x$分别根据在 $[-0.9, -0.1]$ 上均匀分布的 4 个和 5 个等距测量值进行回归。

对于 MH-NNs 和 NFs 的训练，根据方程 (4.1) 对 1，000 f 进行采样，每个样本都构成一个回归任务，其中包含在$[-1, 1]$ 上等距采样的 40 个测量值作为数据。用于训练的 f 样本展示于图 3(a)。少样本回归任务中同时考虑无噪声和含噪数据两种情况。如第 2.3 节所述，前者通过梯度下降微调头部来求解，后者则通过使用 HMC 和 LA 估计后验分布（方程 (2.5)）来求解。假设噪声 ε 为尺度为 0。2 的独立加性高斯噪声，即 $\varepsilon \sim N(0,\ 0。2_2)$。在下游少样本回归任务中，我们将本方法与另外两种方法进行比较：来自 [7], 的迁移学习（TL）方法，该方法仅迁移主体 Φ；以及常规神经网络方法，该方法不使用先验知识，并训练神经网络的所有参数。

图 3 展示了近似 f 并解决下游任务的结果。如图 3(a) 所示，我们的方法能够很好地近似随机函数 f

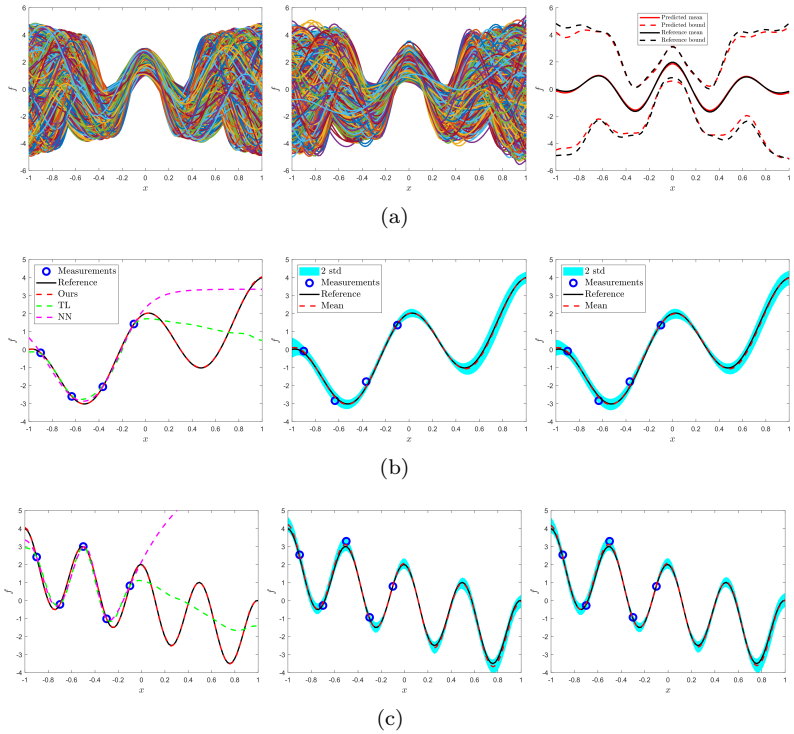


FIG. 3. *Results for approximating the stochastic function defined in Eq.* (4.1) *and solving the downstream few-shot regression tasks. (a) Left:* 1 000 *samples generated from the exact distribution; middle:* 1 000 *samples generated from the learned generator; right: statistics computed from* 1 000 *samples in which we refer to the interval of mean* ± 2 *standard deviations as bound. (b)/(c) Results for the downstream tasks. Left: results for noiseless cases using our method and the transfer learning method in [7], middle: results for noisy cases using our method with regular NN output, and right: results for the same using our method with LA for posterior estimate.*

，展示了多头神经网络在生成建模方面的能力。在使用无噪声数据解决下游任务时，迁移学习方法和常规神经网络方法中都施加了 L_2 正则化，以防止在仅有 4 或 5 个测量值可用时出现过拟合。如图 3(b) 和 (c) 所示，我们的方法能够生成准确的预测，在两个任务中的表现都显著优于另外两种方法，尤其是在没有测量值的区域。通过将我们的方法与神经网络方法进行比较可以看出，f 的先验知识是从 $\{\mathcal{T}_k\}_{k=1}^M$ 中学习得到的，并成功迁移到了新的下游任务中。通过将我们的方法与迁移学习方法进行比较可以看出，先验知识同时存储在主体和头部（以及头部的分布）中。对于含噪情形可以看出，对于两个任务和两种后验估计方法，预测都是准确且可信的：预测均值与参考值一致，误差被预测不确定性所界定。值得注意的是，预测不确定性在区间 $[0,1]$ 内不会增大，即使没有测量值，也会呈现周期性模式。这是因为在贝叶斯推断中，头部被施加了由多头神经网络和正规化流学习得到的富有信息量的先验。

此前考虑的下游任务中的目标函数均选为分布内函数。即使数据不足，也能很好地对其进行回归，这主要是因为它们

属于生成器训练所依据的函数空间。然而，当下游任务中的函数属于分布外（OOD）时，即使数据充足，我们的方法也无法给出良好的预测，如图4所示。此处，目标函数选为 $2\cos(4.5\pi) + x$，其中 ω 和 β 均为分布外。虽然预测出了波动，但与参考值并不吻合。从图4还可以进一步看出，数据充足时，从头开始训练的神经网络明显优于我们的方法。这表明，对于分布外函数，我们越依赖学习到的正则化（由方程 (2.4) 中的 α 值表示），预测就越容易出错。

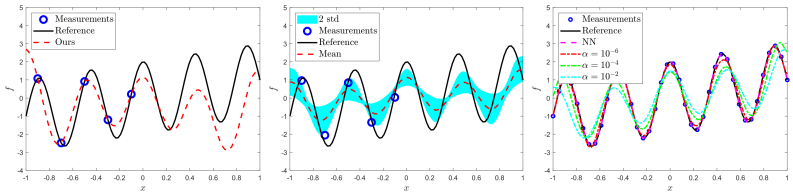


FIG. 4. *Results for regression on an out-of-distribution function. Left: few-shot regression with clean data using our approach; middle: few-shot regression with noisy data using our approach with HMC for posterior estimate; right: regression with sufficient clean data using regular NN method and our approach with different regularization terms α in Eq. 2.4.*

4.2. 非线性常微分方程系统。 在此示例中，我们考虑以下常微分方程系统 [28]，，它描述了受外力作用的摆的运动：

$$\begin{aligned} \frac{du_1}{dt} &= u_2, \\ \frac{du_2}{dt} &= -\lambda \sin(u_1) + f(t), \end{aligned} \tag{4.2}$$

初始条件为 $u_1(0) = u_2(0) = 0$。在方程（4.2）中，f 是外力，λ 是一个常数。为了展示并研究我们的方法在生成建模中的能力，这里我们首先考虑 f 是高斯过程且 $\lambda = 1$ 已知的情形，这称为正问题。与以往直接用神经网络的输出近似随机过程的研究 [44, 46, 15], 不同，在此示例中，我们将微分算子置于神经网络之后，并将随机过程 f 近似为方程（2.1）中的源项。我们还在反问题上测试了我们的方法，其中方程（4.2）中 λ 的值在$\{\mathcal{T}_k\}_{k=1}^M$ 和 $\tilde{\mathcal{T}}$中未知。正问题对应于方程（2.1），其中 $\mathcal{F}_k$，b_k 对所有任务都相同，而 u_k，f_k 则是任务特定的；反问题则对应于方程（2.1），其中 b_k 相同，而 u_k，f_k以及微分算子 F_k 则不同，这是任务特定的 λ所导致的。

4.2.1. 正问题。我们首先假设方程 (4.2) 中的 $\lambda = 1$ 已知，且有关于 f 的数据，即 $\mathcal{D}_k = \{(x_k^i, f_k^i)\}_{i=1}^{N_k}$, $k = 1, \ldots\ldots, M$。如前所述，我们使用 MH-PINN 一次性求解 $\{\mathcal{T}_k\}_{k=1}^M$，然后使用正规化流学习头部的分布。因此，我们得到 f 和 u 的生成器。在这种情况下，假设 f 是具有平方核函数的高斯过程：

$$f(t) \sim GP(0, K), t \in [0, 1], K(x, x') = \exp(-\frac{|x - x'|^2}{2l^2}), \tag{4.3}$$

其中，相关长度 l 设为 0.1,0.25,0.5。如第 2.2 节所述，过去十年间，人们为生成建模和

密度估计开发了多种正规化流。作为生成器使用时，多头物理信息神经网络与所有正规化流兼容。为此，我们比较三种常用的正规化流：RealNVP [9], MAF [33] 和 IAF [20]，最终将多头物理信息神经网络（结合正规化流）与 PI-GAN [44] 进行比较，以近似方程 (4.3) 中定义的、具有不同相关长度的高斯过程。

对于多头物理信息神经网络、正规化流以及物理信息生成对抗网络的训练，按照方程 (4.3) 对 2，000 f 进行采样，每个样本都构成一个物理信息回归任务，其中包含在 $[0,1]$ 上等距采样的 65 个 f 测量值作为数据。注意，方程 (4.2) 中的常微分方程系统可以改写为更简单的形式，如下所示：

$$u_{tt} = -\lambda \sin(u) + f(t), t \in [0,1], \tag{4.4}$$

初始条件为 $u(0) = u_t(0) = 0$。因此，我们选择使用方程 (4.4) 构建物理信息学习的损失函数。

比较结果见图 5 和表 1。从图 5 所示近似高斯过程的谱分析可以看出，采用 MAF 和 IAF 的多头物理信息神经网络与物理信息生成对抗网络表现相当，而采用 RealNVP 的多头物理信息神经网络略逊一筹。如表 1 所示，采用 MAF 和 RealNVP 的多头物理信息神经网络的计算成本显著低于物理信息生成对抗网络，而采用 IAF 的多头物理信息神经网络则比物理信息生成对抗网络更耗费计算资源。值得注意的是，在这个例子中，我们还记录了使用不同生成器进行采样的计算成本。如表 1 所示，物理信息生成对抗网络生成样本的速度显著更快。这是因为，与正规化流不同，生成对抗网络通常只需要相对浅层的神经网络，而正规化流需要深层架构才能保持表达能力。在三种正规化流中，IAF 的采样速度最快，MAF 最慢；而训练时的情况恰好相反。这与这两种正规化流的特性一致：MAF 的前向传播较慢，用于生成样本；逆向传播较快，用于计算密度；IAF 则正好相反。尽管 MAF 的采样速度较慢，但考虑到其训练速度快且性能良好，本文其余所有示例中，我们均采用 MAF 作为多头物理信息神经网络的密度估计器和生成器。

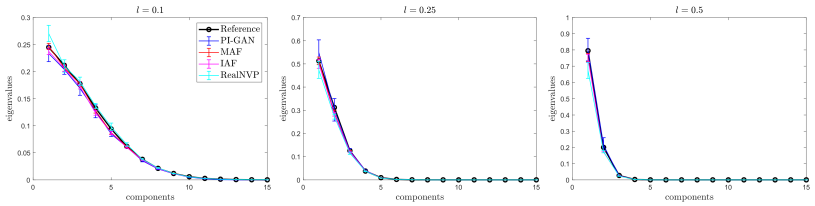


FIG. 5。 *Approximating Gaussian processes as the source term in Eq.* (4.2) *using different models: spectra of the correlation structure for the learned generators for different correlation , , , , lengths l. The covariance matrix is constructed using* 10 000 *generated samples and eigen-values are averaged over 10 generators trained independently.*

4.2.2. 反问题。接下来，我们假设方程（4.2）中的 λ 未知，并且除了 f 之外，还可以获得 u 的一些测量值，即 $\mathcal{D}_k = \{\{x_k^i, f_k^i\}_{i=1}^{N_k^f}, \{x_k^i, u_k^i\}_{i=1}^{N_k^u}\}$。首先使用多头物理信息神经网络，根据数据 $\mathcal{D}_k$ 和物理规律推断 u_k 以及 λ_k，随后使用正规化流从 H_k 和 λ_k 的样本中学习。为此，生成模型针对 u、f 和 λ 的联合分布。这里，我们假设 f 遵循高斯过程的截断 Karhunen–Loève（KL）展开（取前 5 项），该高斯过程采用平方核函数，相关长度为 0。1；对于每个任务 $\mathcal{T}_k$,

	MAF	IAF	RealNVP	PI-GAN
阶段 1	134s	134s	134s	N/A
阶段 2	252s	3939s	245s	N/A
总计	386s	4073秒	379s	3243秒
采样	1.98×10^{-1}秒	1.48×10^{-2}秒	1.50×10^{-2}s	2.29×10^{-3}s

TABLE 1 *Computational time for different models to approximate Gaussian process with correlation length $l = 0.1$. The MH-PINN method is a two-step method and hence its computation is decomposed into two parts: training MH-PINNs referred to as phase 1 and training NFs referred to as phase 2. Sampling time is defined to be the average time needed to generate* 10 000 *samples of* u*.*,

$\lambda_k = \frac{1}{2}\exp(\int_{[0,1]} f_k^2(t)dt)$。对于下游任务 $\tilde{\mathcal{T}}$，目标是根据 u 和 f 的数据不足，推断 u 和 λ。

对于 MH-PINN 和正规化流的训练，我们生成了 2,000 个 f 样本，并在图6(a)中展示。对于每项任务，我们假设有 33 个 f_k 测量值和 9 个 u_k测量值，可在 $[0,1]$, 上等间距获取，并在神经网络建模中对初始条件进行硬编码。对于下游任务，我们假设有 1 个 u 随机测量值和 8 个 f 随机测量值可用，同时也对初始条件进行硬编码。对于含噪测量的情况，我们假设噪声 ε_f 和 ε_u 服从加性高斯分布，其中f测量值的噪声尺度为 0.05， u测量值的噪声尺度为 0.005，即 $\varepsilon_f \sim N(0, 0.05_2)$ 和 $\varepsilon_u \sim N(0, 0.005_2)$。对于每项任务 $\mathcal{T}_k$，通过求解方程 (4.2) 生成 u_k的参考解和干净数据，并使用 Matlab $ode45$及相应的 f_k 和 λ_k。

结果如图 6和表2所示，由此可见，我们的方法能够很好地将随机过程近似为源项，并在数据有限的下游任务中，为 u， f 以及 λ 生成准确且可信的预测，无论是在无噪情形还是含噪情形下。如图所示，由于数据不足，物理信息神经网络方法对 u 和 λ 的估计均不可接受；而我们的方法通过结合来自 $\{\mathcal{T}_k\}_{k=1}^M$ 的先验知识与多头物理信息神经网络，精度要高得多。

	PINN	MH-PINN
λ	0.8440	2.5428
误差（%）	63.99	1.21

TABLE 2*Estimate of* λ *and* L_2 *relative error of* u *for the downstream inverse problem on Eq.* (4.2) *with clean data, using our approach and the regular PINN method. The reference value for* λ *is* 2.3609*.*

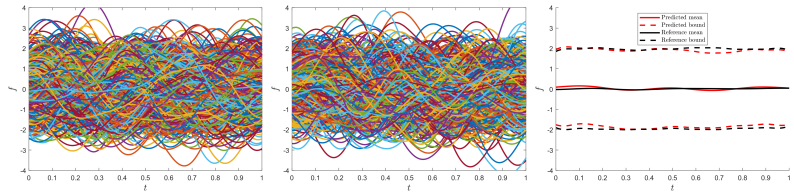
4.3. 一维非线性反应扩散方程。现在，我们在一维非线性时变反应扩散方程上测试我们的方法，该方程通常称为 Fisher方程 [2]:

$$u_t = Du_{xx} + ku(1-u), t \in [0,1], x \in [-1,1], \tag{4.5}$$

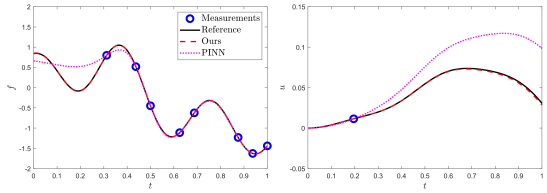
$$u(t,-1) = u(t,1) = 0, t \in [0,1], \tag{4.6}$$

$$u(0,x) = u_0(x), x \in [-1,1], \tag{4.7}$$

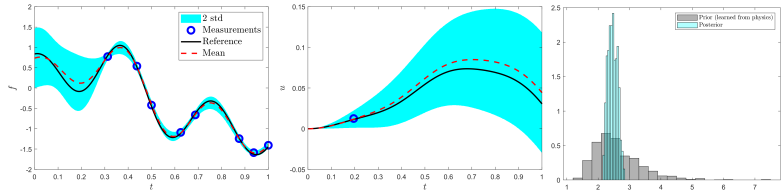
其中 $D = 0.1, k = 0.1$， $u_0(x)$ 是初始条件函数。在此示例中，我们假设初始条件函数是一个随机过程，其分布如下:



(a)



(b)



(c)

FIG. 6. *Results for the inverse problem of the ODE system* 4 2 *with initial conditions hard-*(∘), , *encoded in NN modeling. (a) Left:* 1 000 *samples of* f *generated from the exact distribution; middle:* 1 , 000 *samples of* f *generated from the learned generator; right: statistics computed from samples.* , , , *The bound is defined as the same as in the caption of Fig. 3. (b) Predicted* f *and* u *using PINNs and our approach for the downstream inverse problem with noiseless data. (c) Predicted* f u *and* λ *with uncertainties using our approach* , *for the downstream inverse problem with noisy data. The, predicted mean and standard deviation of* λ *is* 2.4663 *and* 0.1501 *while the reference value is* 2.3609.

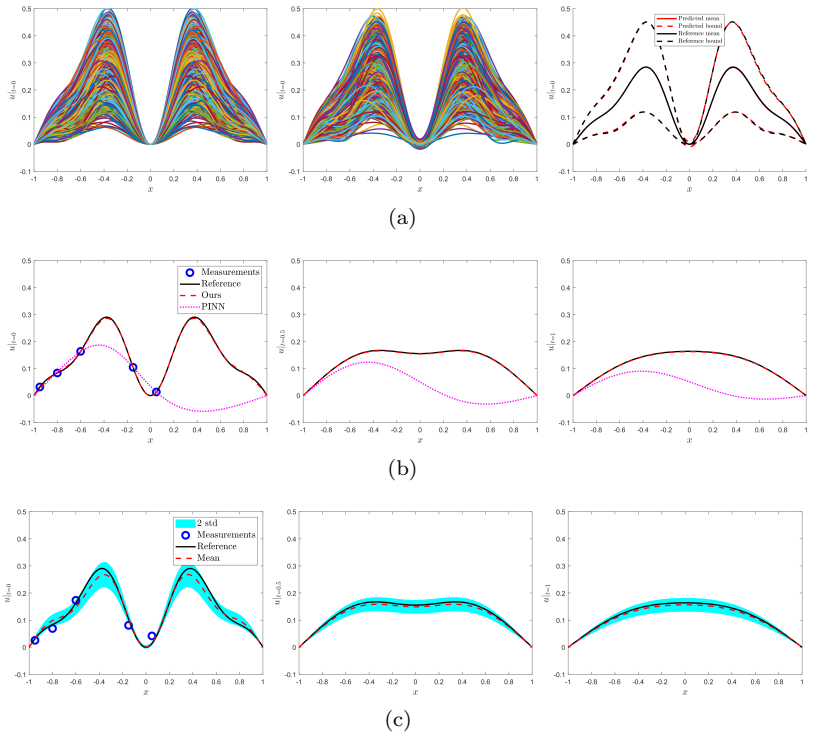
分布：

$$u_0(x) = \frac{(x^2-1)}{5}\sum_{j=1}^{5}\xi_j(\cos^2(jx)-1), x \in [-1,1], \tag{4.8}$$

其中， $\xi_j, j = 1,...,5$ 是独立同分布（i.i.d.）随机变量，服从 $[0,1)$ 上的均匀分布，即 $\xi_j \sim U[0,1)$。与前面的示例不同，随机性来自初始条件，而非源项。此示例对应于方程（2.1），其中 $\mathcal{F}_k$ 和 f_k 对所有任务均相同，而 u_k 和 b_k 则因任务而异。除了 u_0的测量值外，在 $\{\mathcal{T}_k\}_{k=1}^{M}$ 和 $\tilde{\mathcal{T}}$中还需要用于计算偏微分方程残差的点。因此，数据为 $\mathcal{D}_k = \{\{(t_k^i, x_k^i)\ 0\}_{i=1}^{N_k^f}, \{(0, x_k^i)\ b_k^i\}_{i=1}^{N_k^b}\}$.

在训练中，生成了 2，000 个 $u_0(x)$ 样本，如图7(a)所示；每个样本都构成一个物理信息回归任务，其中包含在 $[-1,1]$ 上等距采样的u_0 的 41 个测量值，作为初始条件的数据。此外，对于所有任务，均在时空域 $[0,1]\times[-1,1]$ 上使用均匀网格 21×41 来计算偏微分方程残差损失。对于下游任务 $\mathcal{T}$，可获得 u_0 的 5 个随机测量值，并采用相同的均匀网格。在 $\{\mathcal{T}_k\}_{k=1}^{M}$ 和 $\tilde{\mathcal{T}}$中，边界条件均被硬编码到神经网络建模中。对于含噪情形，

假设噪声 ε 是独立的加性高斯噪声，对于 u_0 的测量值和偏微分方程残差，其噪声尺度均为 0.02，即 $\varepsilon \sim N(0,\ 0.02_2)$。结果见图7和表3。可以看出，我们的方法能够根据数据和物理规律估计出随机过程的良好生成器，从而为下游少样本物理信息回归任务提供有信息量的先验知识。无论是在无噪声还是含噪情形下，预测都很准确；在含噪情形下，误差均受到预测不确定性的约束。表3所示的 L_2 u误差表明，我们的方法显著优于物理信息神经网络方法，从而证明了将先验知识引入相似任务求解的有效性。



(a)

(b)

(c)

FIG. 7. *Generator learning and few-shot physics-informed learning on 1-D time-dependent reaction-diffusion equation* 4 5 (.) *with boundary conditions hard-encoded in NN modeling. (a) Left:* , 1 0 · 000 *training samples of* u_0*; middle:* 1 000 *samples of* u *from the learned generator; right:* , (,) , , *statistics computed from samples. The bound is defined as the same as in the caption of Fig. 3. (b) Predicted* u *at* $t = 0$ 0.5 1 *using our approach and the PINN method with noiseless measurements.* , , *(c) Predicted mean and uncertainty of* u *at* $t = 0$ 0.5 1 *using our approach with HMC for posterior estimate with noisy measurements.* ,

	PINN	MH-PINN
误差 (%)	78.77	0.22

TABLE 3

L_2 *relative error of* u *for the downstream few-shot physics-informed learning task on Eq.*(4.5) *with clean data of* u_0 *using our approach and the PINN method.*

4.4. 二维非线性 Allen-Cahn 方程。 现在我们转向一个具有狄利克雷边界条件 [43] 的二维稳态非线性 Allen-Cahn 方程：

$$\lambda \Delta u + u(u^2 - 1) = f, x, y \in [0, 1], \tag{4.9}$$

$$u(x, 0) = u(x, 1) = u(0, y) = u(1, y) = 0, \tag{4.10}$$

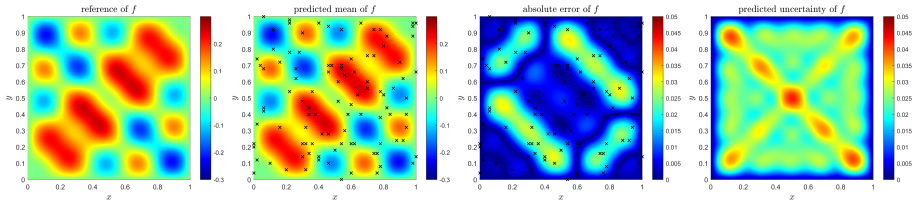
其中， $\lambda = 0.1$ 是一个常数， f 是源项。这里，我们对 f 施加一个分布，该分布由方程（4.9）以及解 u 的以下分布推导而来：

$$u(x, y) = \frac{1}{5} \sum_{j=1}^{5} \xi_j \frac{\sin(j\pi x) \sin(j\pi y)}{j^2 \pi^2}, x, y \in [0, 1], \tag{4.11}$$

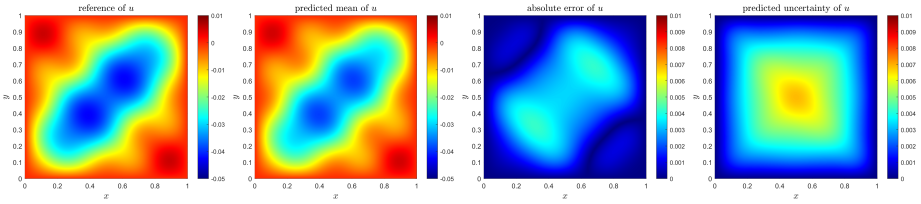
其中随机变量 ξ_j， $j = 1$，……，5 服从均匀分布，即 $\xi_j \sim U[0, 1)$ 。在此示例中，我们希望使用我们的方法，根据 f 的数据和方程（4.9）中的物理规律，学习 u 和 $f^{\sim}$ 的生成器，并利用它在 $^{\sim}$ 数据不足 $\mathcal{D}$ 的情况下解决下游任务 $\mathcal{T}$。此示例对应于方程（2.1），其中 $\mathcal{F}_k$， b_k 相同

在任务之间， f_k， u_k 具有任务特定性，且数据为 $\mathcal{D}_k = \{(x \quad {}^{i}_{k}, y^{k}) \; f_{\,{}^{i}_{k}}\}^{N^f}_{ki=1}$。

为训练 MH-PINN 和 NFs，我们从其分布中采样 2， 000 f 其中每个样本都在二维空间域 $[0, 1] \times [0, 1]$上使用 51×51 均匀网格进行求解。对于下游任务，假设可以获得均匀网格上的 100 个 f 随机测量值。噪声被假设为相互独立的加性高斯˜噪声，噪声尺度为 0。05。在 $\mathcal{T}_k$ 和 $\mathcal{T}$中，边界条件均以硬编码方式融入神经网络建模。结果以及测量值的位置展示于图 8 和表 4 中。与前述所有示例类似，我们的方法给出了准确且可信的预测，表明先验知识在确定性推断和贝叶斯推断中都得到了良好的学习与迁移。



(a)



(b)

FIG. 8。 *Results for few-shot physics-informed learning on the 2-D nonlinear Allen-Cahn equation Eq.* 4 9 (。) *with noisy measurements of* f. *Predicted mean* μ *and standard deviation* σ *are computed over* 1 000 *posterior samples from HMC. The absolute error is defined as the absolute value of difference between the reference and* μ. *Black crosses represent the locations of the measurements on* f.

	PINN	MH-PINN
误差（%）	12.82	0.30

TABLE 4

L_2 relative error of u for the downstream few-shot physics-informed learning task on Eq.(4.9) *with clean data of f, using our approach and the PINN method.*

4.5. 二维随机亥姆霍兹方程。本文测试的最后一个示例是具有随机源项和狄利克雷边界条件 [35]的二维亥姆霍兹方程：

$$(\lambda^2 - \nabla^2)u = f, x, y \in [0, 2\pi], \tag{4.12}$$

$$u(x, 0) = u(x, 2\pi) = u(0, y) = u(2\pi, y) = 0, \tag{4.13}$$

其中， λ^2 是亥姆霍兹常数， f 定义如下：

$$f(x, y) = \frac{2}{d}\{\sum_{i=1}^{d/4} \xi_i \sin(ix) + \xi_{i+d} \cos(ix) + \xi_{i+2d} \sin(iy) + \xi_{i+3d} \cos(iy)\}, \tag{4.14}$$

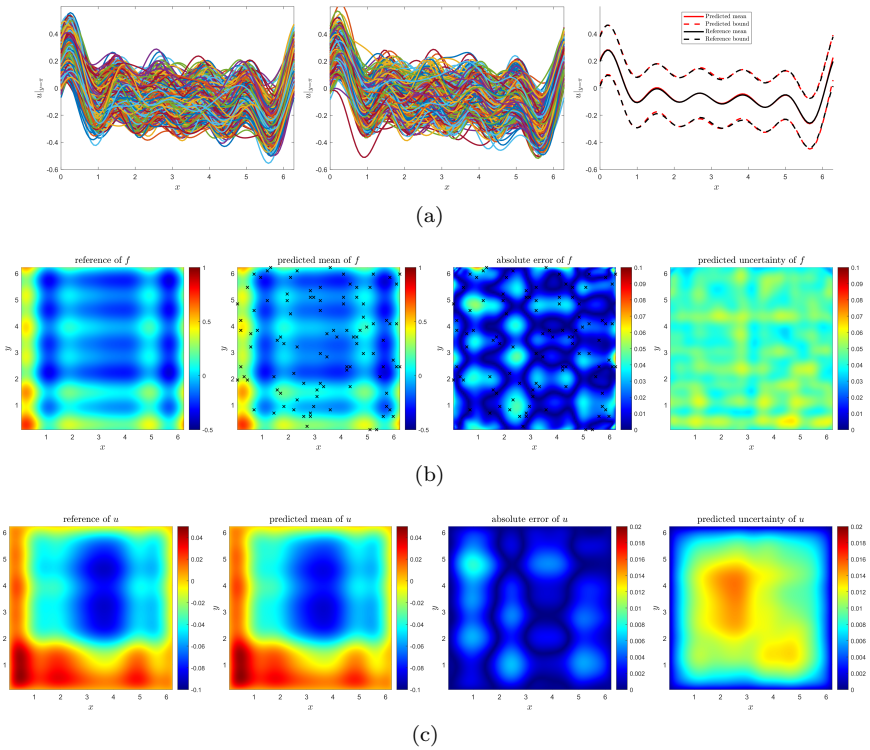
其中， $\xi_j, j = 1, ..., d$ 是服从均匀分布 $U[0, 1)$ 的独立同分布随机变量， d 表示随机性的维度。为便于演示，本文考虑 $d = 20$ 的情况，这与 [35] 中 $d = 100$的情况不同。

我们研究的第一个案例是 $\lambda^2 = 1$ 已知时的正问题。此设置对应于方程（2.1），其中 $\mathcal{F}_k$， b_k 在各任务间共享，而 u_k， f_k 则因任务而异。接下来，我们研究 λ未知时的反问题，此时可获得关于 u 和 f的数据；这对应于方程（2.1），其中只有 b_k 相同，而 u_k, f_k以及算子 $\mathcal{F}_k$ 因任务而异。在这两种情况下，下游任务均定义为与 $\{\mathcal{T}_k\}_{k=1}^M$ 相同，但测量值更少。

对于正问题和反问题，均从其分布中抽取 $10,000$ f，因此使用多头物理信息神经网络求解了 $10,000$ 个任务，并在神经网络建模中对边界条件进行硬编码。图9(a)展示了 f 一个切片的样本。对于正问题， $\mathcal{D}_k$ 仅包含源项 f_k的测量值，即 $\mathcal{D}_k = \{\{(x_k^i, y_k^i)\ f_k^i\}_{i=1}^{N_k^f}\}$；而对于反问题， $\mathcal{D}_k$还包含待求解的解 u_k的测量值：$\mathcal{D}_k = \{\{(x_k^i, y_k^i)\ f_k^i\}_{i=1}^{N_k^f}, \{(x_k^i, y_k^i)\ u_k^i\}_{i=1}^{N_k^u}\}$。正问题的训练中，通过在二维空间域 $(0, 2\pi) \times (0, 2\pi)$ 上使用 50×50 均匀网格（不包括边界）来求解 f 的每个样本。对于反问题，使用相同的 $10,000$ 个 f 样本，但这次通过 21×21 均匀网格进行求解。此外，对于每个任务 $\mathcal{T}_k$，还可获得在 6×6均匀网格上对 u_k 的测量值。参考解以及 u 的测量值通过使用五点格式的有限差分法求解方程（ 4.12 ）生成，其中 $\lambda_k^2 = \int_{[0,2\pi]^2} f_k^2(x, y)dxdy$。对于下游任务，正问题可获得 f 的 100 个随机测量值；反问题则可获得 f 的 50 个随机测量值和 u的 10 个随机测量值。假设噪声为相互独立的加性高斯噪声，噪声尺度为 0.05。

结果见表5和表6，以及图9和图10。如图所示，在正问题与逆问题中，学习到的生成器均能高质量地生成 f 样本，同时为下游任务提供有价值的先验知识。对于采用贝叶斯推断和不确定性量化的含噪情形，预测均值与参考值一致，且绝对误差大多处于预测不确定性的范围内。我们的方法在少样本物理信息学习中的有效性

以及其对确定性优化和贝叶斯推断的适用性，已在前述五个示例中得到持续验证。



(a)

(b)

(c)

Generator learning and few-shot physics-informed learning on the stochastic Helmholtz
FIG. 9. *equation* 4 12 . *(a) Left:* 1 000 *training samples of a slice of* f *at* $y = \pi$*; middle:* 1

(.),, 000 *samples of a slice of* f *at* $y = \pi$ *from the learned generator; right: statistics computed from samples. (b)/(c) Results for the downstream forward problem with* 100 *random noisy measurements on* f *using our ,,,, approach with HMC. From left to right are reference predicted mean absolute error and predicted uncertainty of* f/u*. Black crosses represent the locations of the measurements of* f*.*

	PINN	MH-PINN
误差 (%)	21.14	1.12

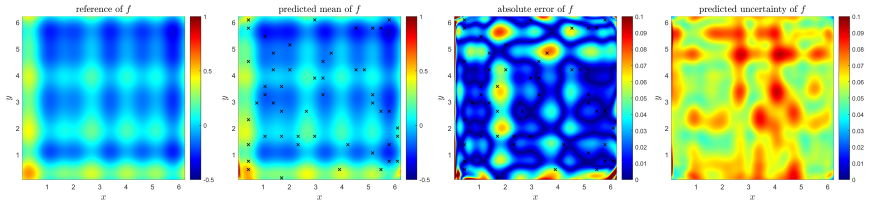
TABLE 5

L_2 *relative error of* u *for the downstream forward problem on Eq.* (4.12) *with clean data of* f*, using our approach and the PINN method.*

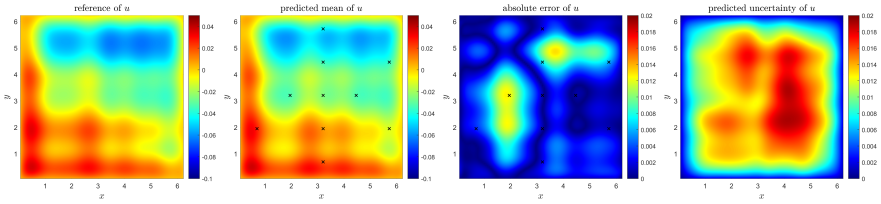
	PINN	MH-PINN
λ	1.9328	1.0170
误差 (%)	59.92	2.58

TABLE 6*Estimate of* λ *and* L_2 *relative error of* u *for the downstream inverse problem on Eq.* (4 12 *with* .) *clean data. The reference value of* λ *is* 1.0042.

5. 使用多头神经网络进行多任务学习。迄今为止，我们主要关注将 MH-NNs 与 NFs 结合起来估计随机生成器



(a)



(b)

FIG. 10. *Results for the downstream inverse problem on the stochastic Helmholtz equation 4.12 (with 50 random noisy measurements of f and 10 random noisy measurements of u., λ is estimated as 1.0785 ± 0.0307 in the format of predicted mean $\pm$ predicted standard deviation while the reference value is 1.0042. (a)/(b) From left to right are reference, predicted mean, absolute error, and predicted uncertainty of f/u. Black crosses represent locations of the measurements of f or u.*

并从 $\{\mathcal{T}_k\}_{k=1}^M$ 中学习有用的先验知识。具体做法是，先以 MTL 方式训练 MH-NNs，然后训练 NFs 来估计头部的概率密度函数。直观而言，MH-NNs 经 MTL 训练后捕获共享信息的能力，是生成建模和少样本学习取得成功的关键。对于使用 MH-PINNs 的物理信息多任务学习，我们同时求解常微分方程/偏微分方程；假设这些解共享同一组基函数，便可得到这组系数的样本，从而实现生成建模，继而进行少样本学习，这正是本文所提方法的核心所在。然而，对所有解施加相同基函数所带来的成本和/或益处，尚未得到明确讨论。一方面，共享主体将各任务的训练联系起来；若任务在某些方面相似，这可能会有所帮助。另一方面，强制所有解共享相同的基函数，在它们表现不同时也可能有害。具体而言，对于数据和物理信息都足够充分的任务，强迫它们与所有其他任务共享同一主体，可能会起到负向正则化的作用；在这些特定任务上，单任务学习（STL）的预测准确率可能优于 MTL。本节通过重新探讨第 4.1 节中的简单函数逼近示例，研究使用 MH-NNs 进行 MTL 的效果，并给出初步结果和分析，希望能为今后更严谨的研究提供有用的信息和见解。

5.1. 基函数学习与协同学习。如前所述，在 MTL 中学习到的基函数的质量和行为，对于生成式建模、学习任务之间的关系以及任务的代表性信息至关重要$\{\mathcal{T}_k\}_{k=1}^M$。我们从数值示例中始终观察到，多头神经网络中头部的初始化对 MTL 的平均准确率、基函数的学习以及头部的分布有很大影响。这里，我们测试了三种初始化策略：标准差为 0.05 的随机正态初始化方法，简称RN

(0.05)、Glorot 均匀初始化方法 [12] ，简称GU，以及标准差为 1 的随机正态初始化方法，简称$RN\ (1)$。在下游少样本学习任务中，我们对头部进行微调，但不使用学习到的概率密度函数；这实际上是 [7], 中的迁移学习方法，该方法排除了头部的分布所包含的信息，预测准确率完全取决于基函数中包含的先验知识水平。

如图 11 所示，RN (0.05) 产生的信息性最低的基函数，其行为主要受神经网络的双曲正切激活函数支配。使用迁移学习方法进行下游少样本学习任务，进一步验证了这一点。此外，正如表 8 所示，它在多任务学习中的平均预测准确率也是最差的。GN 和 RN (1) 的表现相似。一些基函数的图似乎表明，RN (1) 产生的基函数更好，其行为与图 11(b) 所示的函数族更为相似。然而，如图 11(c) 所示，这并不一定意味着下游任务具有更丰富的先验知识。

实证结果表明，与另外两种初始化策略相比，采用 RN (0.05) 的多头神经网络既无法实现准确的多任务学习，也无法在基函数上实现协同学习。然而，我们注意到，在生成建模中，它在准确率和收敛速度方面表现显著更好。如图 11(d) 所示，学习到的生成器生成的样本质量更高。我们一再发现，使用相对较小的值初始化头部，通常能使正规化流更容易、更快速地训练，并能准确地学习生成模型。我们推测，这是因为在多任务学习中，当头部以较小的值初始化时，多头神经网络倾向于将代表性信息和信息性信息包含在头部中；而当头部以相对较大的值初始化时，则倾向于将这些信息包含在基函数中。

	RN（0.05）	GU	RN (1)
误差 (%)	0.8373 ± 0.2341	0.1907 ± 0.0690	0.3131 ± 0.0937

TABLE 7

L_2 relative errors from MTL for 1 000 tasks using different initialization methods. The ,,,,, errors are displayed in the format of mean ± standard deviation computed over all tasks.

5.2. 多任务学习（MTL）与单任务学习（STL）。如前所述，采用多头神经网络（MH-NNs）的多任务学习不一定能实现协同学习，也不一定能提高所有任务的平均准确率。这里，我们再次使用第 4.1 节中的函数逼近示例，考察采用多头神经网络进行多任务学习相较于单任务学习的有效性。我们首先考虑数据充足的情况。为此，我们从 1,000 个训练样本中随机选取 100 个，每个样本都由一个独立训练的神经网络进行逼近，并将结果与多头神经网络在预测准确率方面进行比较。请注意，在这种情况下，多头神经网络像之前一样使用 1,000 个函数进行训练，并在选定的 100 个函数上测试；而具有相同架构的单头神经网络则直接使用 100 个函数进行训练。结果见表 8，实证结果表明，在某些情况下，例如采用随机正态初始化方法时，多任务学习的表现不如单任务学习。

我们考虑的第二种情况是：某些任务的数据充足，而其他任务的数据不足。为此，我们将1，000个任务平均分为两个子集。对于前500个任务，假设我们在 $[-1, 1]$ 上只有10个随机采样的测量值；对于另外500个任务，假设我们在 $[-1, 1]$上有完整的40个等距分布的测量值。我们使用多头神经网络进行多任务学习，一次性处理这1，000个回归任务，并将所有任务视为同等重要。

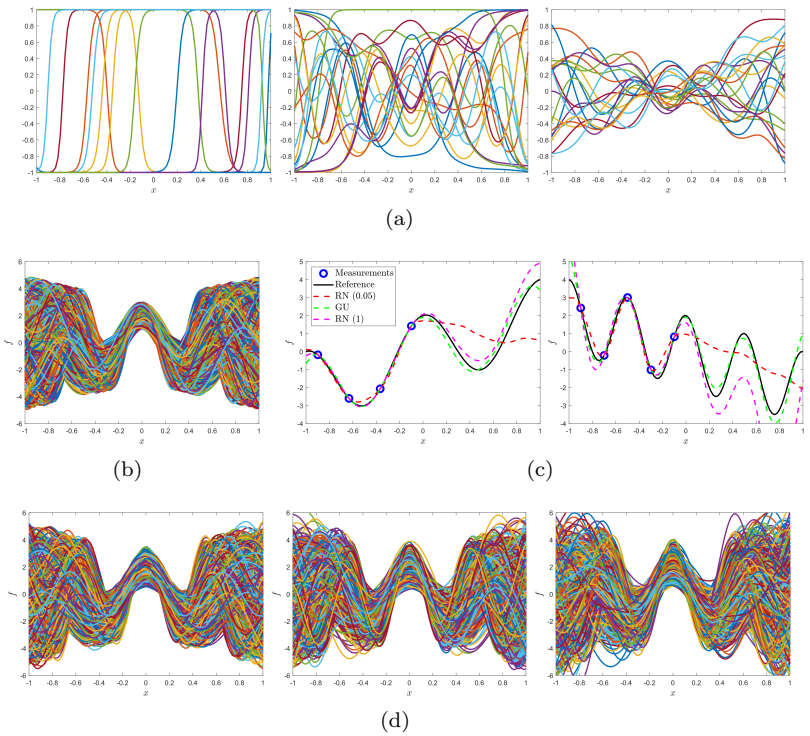


FIG. 11。 *The effect of different initialization methods of the head, in basis functions learning, few-shot learning trained, for approximating* 1 000 *f generated from Eq.* 4 1 *using from left to right RN (*0.05*) and generator learning. (a) Samples of* 20 *basis functions from MH-NNs, (。) GU and RN (*1*) initialization methods. (b)* 1 000 *training samples of f. (c) Results for two downstream ,few-shot regression tasks, using TL method without regularization informed by the learned PDF, as opposite to the proposed approach. (d) Results for generator learning using from left to right RN (*0.05*) GU and RN (*1*) initialization methods.*

	RN（0.05）	GU	RN (1)	STL
误差 (%)	0.7575 ± 0.2477	0.1362 ± 0.0259	0.3664 ± 0.1031	0.2102 ± 0.0794

TABLE 8

L_2 relative errors of f from MTL with MH-NNs and STL with NNs on 100 *tasks. Different , , , initialization methods are used for the heads in MH-NNs. The errors are displayed in the format of mean ± standard deviation computed over all* 100 *tasks.*

结果见表9和图12。可以看出，与单任务学习相比，多任务学习提高了数据不足任务的预测准确率，为协同学习提供了实证证据。另外，有趣的是，RN (1)初始化方法虽然生成的生成模型效果最差，却在三种方法中表现最佳。这与我们先前关于多头神经网络基函数学习的推测一致：以较大值初始化的头部往往会促使基函数编码具有代表性且信息性的信息。

6. 讨论。我们开发了用于物理信息机器学习的多头神经网络（MH-NNs），并提出了一种新方法——多头物理信息神经网络（MH-PINNs），该方法已在 L-HYDRA 代码中实现。本研究主要关注多头神经网络和多头物理信息神经网络在各种科学机器学习问题中的应用，

	RN (0.05)	GU	RN (1)
误差 (%)	63.60 ± 24.08	40.49 ± 20.49	16.91 ± 11.08

TABLE 9

L_2 relative errors of f from MTL with MH-NNs on 500 tasks equipped with insufficient data. , , , The errors are displayed in the format of mean ± standard deviation computed over all 500 tasks.

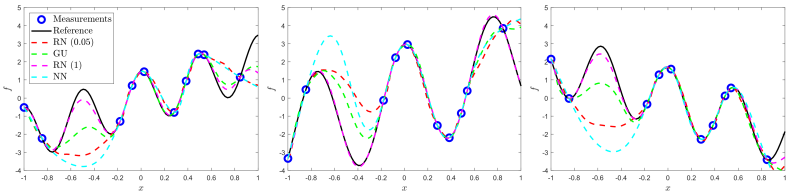


FIG. 12。 *Results for 3 tasks with insufficient data from MTL with MH-NNs using different , , initialization methods over the head and from STL with NNs with the same architecture. We note that tasks with sufficient data and tasks with insufficient data are treated equally in MTL.*

包括多任务学习（MTL）、随机过程逼近和小样本回归学习。我们首先在方程（2.1）中对问题进行了表述，介绍了 MH-PINNs 的架构设计，并提出了一种借助归一化流（NFs）将 MH-NNs 和 MH-PINNs 转换为生成模型的方法，以进行密度估计和生成建模。随后，我们研究了 MH-PINNs 在求解常微分方程/偏微分方程（ODEs/PDEs）以及逼近随机过程方面的适用性和能力。最后，我们对 MH-NNs 在协同学习中的应用进行了初步探索和实证研究，并考察了采用 MH-NNs 进行多任务学习的潜在收益与成本。

本文有多种用途：提出了一种用于通过多任务学习求解常微分方程/偏微分方程的神经网络方法；提供了一种逼近随机过程的新方法；提出了一种解决少样本物理信息学习问题的方法，这类问题常见于元学习和迁移学习的背景下；系统研究了将多头神经网络应用于科学计算问题；并首次提供了协同学习的实验证据。

然而，关于多头神经网络，我们还有几个尚未解决的主要问题，其中之一是多头神经网络，或更广泛地说，硬参数共享神经网络在逼近复杂随机过程时的表达能力。直观而言，如果两个函数的行为截然不同，强制它们共享相同的基函数会对逼近精度产生不利影响。第二个问题是多任务学习中损失函数不同项之间的平衡问题。文献 [29]表明，在单任务学习中训练的物理信息神经网络，已经深受损失函数中不同项前权重的影响，例如数据损失、边界条件损失和偏微分方程残差损失。训练 MH-PINN时，这一问题可能更加复杂，因为在多任务学习中，损失函数通常定义为任务特定损失的加权和。最后一个主要问题是用于协同学习的MH-PINN。本文仅研究了函数逼近中的一个示例，并给出了实验证据。未来应沿着理论和计算两个方向，进一步探索MH-PINN协同学习的原理。

致谢。 我们谨此感谢华中科技大学的孟旭辉教授所给予的宝贵讨论。本研究获得以下资助：美国海军研究办公室（ONR）范内瓦·布什教职奖学金（GEK）（N00014-22-

1-2795）；美国能源部先进科学计算研究计划通过“面向地球与嵌入式系统的可扩展、高效加速因果推理算子、图与脉冲”（SEA-CROGS）项目提供的资助（DE-SC0023191）；以及 MURI/AFOSR 项目（FA9550-20-1-0358）的资助。

参考文献

[1] M. Abadi, A. Agarwal, P. Barham, and et al., *TensorFlow: Large-scale machine learning on heterogeneous systems* 2015 , , https://www.tensorflow.org/。软件可从 tensorflow.org 获取。[2] M. J. Ablowitz and A. Zeppetella, *Explicit solutions of fisher's equation for a special wave speed* 41 1979 835 840 ,《数学生物学通报》，（），第 – 页。[3] B. Bahmani and W. Sun, *Training multi-objective/multi-task collocation physics-informed neural network with student/teachers transfer learnings* 2107 11496, , arXiv 预印本 arXiv:。（2021）。[4] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model reduction and neural networks for parametric PDEs* 2005 03180, 2020 , arXiv 预印本 arXiv:。（）。[5] R. Caruana, *Multitask Learning: A Knowledge-Based Source of Inductive Bias*,载于国际机器学习会议，1993。[6] X. Chen, J. Duan, and G. E. Karniadakis, *Learning and meta-learning of stochastic advection–diffusion–reaction systems from sparse measurements* ,《欧洲应用数学杂志》，32（2021），第397–420页。[7] S. Desai, M. Mattheakis, H. Joy, P. Protopapas, and S. Roberts, *One-shot transfer learning of physics-informed neural networks* 2110 11286, 2021 , arXiv 预印本 arXiv:。（）。[8] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton,A. Alemi M. Hoffman and R. A. Saurous *TensorFlow Distributions* ,, , , arXiv 预印本 arXiv:1711.10604, （2017）。[9] L. Dinh, J. Sohl-Dickstein, and S. Bengio, *Density Estimation Using Real NVP*, arXiv 预印本 arXiv:1605.08803, （2016）。[10] C. Finn, P. Abbeel, and S. Levine,*Model-agnostic meta-learning for fast adaptation of deep networks* 2017, 1126 1135 ,载于国际机器学习会议，PMLR，第 – 页。[11] M. Germain, K. Gregor, I. Murray, and H. Larochelle, *MADE: Masked autoencoder fordistribution estimation*,载于国际机器学习会议，PMLR，2015，第881–889页。[12] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neuralnetworks*,载于第十三届国际人工智能与统计学会议论文集，JMLR研讨会与会议论文集，2010，第249–256页。[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio *Generative adversarial networks* ,, , ,《美国计算机学会通讯》，63（2020），第139–144页。[14] S. Goswami, K. Kontolati, M. D. Shields, and G. E. Karniadakis, *Deep transfer operator learning for partial differential equations under conditional shift*,《自然·机器智能》，4（2022），第1155–1164页。[15] L. Guo, H. Wu, and T. Zhou, *Normalizing field flows: Solving forward and inverse stochastic differential equations using physics-informed flow models*,《计算物理学杂志》，461（2022），第111202页。[16] J. Ho, A. Jain, and P. Abbeel, *Denoising diffusion probabilistic models*,《神经信息处理系统进展》，33（2020），第6840–6851页。[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang,*Physics-informed machine learning* 3 2021 422 440 ,《自然·物理评论》，（），第 – 页。[18] R. E. Kass, *The validity of posterior expansions based on Laplace's method*,《统计学与计量经济学中的贝叶斯和似然方法》，（1990），第473–488页。[19] Y. Khoo, J. Lu, and L. Ying, *Solving parametric PDE problems with artificial neural networks* 32 2021 421 435 ,《欧洲应用数学杂志》，（），第 – 页。[20] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling,*Improved variational inference with inverse autoregressive flow*,《神经信息处理系统进展》，29（2016）。[21] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, arXiv 预印本 arXiv:1312.6114, （2013）。[22] I. Kobyzev, S. J. Prince, and M. A. Brubaker, *Normalizing flows: An introduction and re-view of current methods*, IEEE《模式分析与机器智能汇刊》，

43（2020），第3964–3979页。[23] J. Lao, C. Suter I. Langmore C. Chimisov A. Saxena P. Sountsov D. Moore R. A. ,, , , , , , , , , Saurous M. D. Hoffman and J. V. Dillon *tfp. mcmc: Modern Markov chain Monte* , arXiv 预印本 *Carlo tools built for modern hardware* 2002 01184, 2020 arXiv:。（）。[24] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar , *Fourier neural operator for parametric partial differential equations*, arXiv 预印本 arXiv:2010.08895, （2020）。[25] Z. Lin, Z. Zhao, Z. Zhang, H. Baoxing, and J. Yuan, *To Learn Effective Features: Understanding the Task-Specific Adaptation of MAML* 2021 ,, https://openreview.net/forum? id=FPpZrRfz6Ss。[26] K. Linka, A. Schafer, X. Meng, Z. Zou, G. E. Karniadakis, and E. Kuhl, *Bayesian Physics-Informed Neural Networks for real-world nonlinear dynamical systems*, arXiv 预印本 arXiv:2205.08304, （2022）。[27] X. Liu, X. Zhang, W. Peng, W. Zhou, and W. Yao, *A novel meta-learning initializationmethod for physics-informed neural networks*,《神经计算与应用》，（2022），第1–24页。[28] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *Learning nonlinear operatorsvia DeepONet based on the universal approximation theorem of operators*,《自然·机器智能》，3（2021），第218–229页。[29] L. McClenny and U. Braga-Neto, *Self-adaptive physics-informed neural networks using a soft attention mechanism* 2009 04544, 2020 , arXiv 预印本 arXiv:。（）。[30] X. Meng, L. Yang, Z. Mao, J. del Águila Ferrandis, and G. E. Karniadakis, *Learningfunctional priors and posteriors from data and physics*,《计算物理学杂志》，457（2022），第111073页。[31] R. M. Neal et al., *MCMC using Hamiltonian dynamics*,《马尔可夫链蒙特卡罗手册》，2（2011），第2页。[32] G. Papamakarios, E. T. Nalisnick D. J. Rezende S. Mohamed and B. Lakshmi- ,, , , 《机器学习研 narayanan *Normalizing Flows for Probabilistic Modeling and Inference.* 究杂志》，22（2021），第1–64页。[33] G. Papamakarios, T. Pavlakou, and I. Murray, *Masked autoregressive flow for density estimation* 30 2017 ,《神经信息处理系统进展》，（）。[34] M. Penwarden, S. Zhe, A. Narayan, and R. M. Kirby, *Physics-informed neural net-works (PINNs) for parameterized PDEs: A metalearning approach*, arXiv 预印本 arXiv:2110.13361, （2021）。[35] P. Perdikaris, D. Venturi, and G. E. Karniadakis, *Multifidelity information fusion algo-rithms for high-dimensional systems and massive data sets*,《SIAM科学计算杂志》，38（2016），第B521–B538页。[36] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis, *Uncertainty Quantifica-tion in Scientific Machine Learning: Methods, Metrics, and Comparisons*, arXiv 预印本 arXiv:2201.07766, （2022）。[37] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals,*Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML* 1909 09157, 2019 , arXiv 预印本 arXiv:。（）。[38] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations* 378 2019 686 707 ,《计算物理学杂志》，（），第 – 页。[39] S. Ruder, *An overview of multi-task learning in deep neural networks*, arXiv 预印本 arXiv:1706.05098, （2017）。[40] P. Thanasutives, M. Numao, and K.-i. Fukui, *Adversarial multi-task learning enhanced physics-informed neural networks for solving partial differential equations*,载于2021年国际联合神经网络会议（IJCNN），IEEE，2021，第1–9页。[41] H. Wang, H. Zhao, and B. Li, *Bridging multi-task learning and meta-learning: Towards effi-cient training and effective adaptation*,载于国际机器学习会议，PMLR，2021，第10991–11002页。[42] S. Wang, H. Wang, and P. Perdikaris, *Learning the solution operator of parametric partial differential equations with physics-informed DeepONets*,《科学进展》，7（2021），第eabi8605页。[43] L. Yang, X. Meng, and G. E. Karniadakis, *B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data*,《计算物理学杂志》，425（2021），第109913页。[44] L. Yang, D. Zhang, and G. E. Karniadakis, *Physics-informed generative adversarial networks for stochastic differential equations*,《SIAM科学计算杂志》，42（2020），第A292–A317页。

[45] M. Yang and J. T. Foster, *Multi-output physics-informed neural networks for forward and inverse PDE problems with uncertainties*, 《计算机在应用力学与工程中的方法》, (2022), 第 115041 页。
[46] W. Zhong and H. Meidani,*tic differential equations* P-VAE: Physics-Informed Variational Auto-Encoder for stochas- 2203 11363, 2022 ,
arXiv 预印本 arXiv:。 （）。[47] Z. Zou, X. Meng, A. F. Psaros, and G. E. Karniadakis, *NeuralUQ: A comprehensive libraryfor uncertainty quantification in neural differential equations and operators*, arXiv 预印本 arXiv:2208.11866, (2022)。

附录 A. 神经网络架构和训练超参数的详细信息。对于第 4 节和第 5 节中的所有示例，MH-PINN 均采用全连接神经网络（前馈神经网络）实现，包含 3 个非线性隐藏层，每层配备 50 个神经元，并采用双曲正切激活函数。头部数量与相应示例中的任务数量相同：第 4.1 节中为 1，000 个，第 4.2、4.3 和 4.4 节中为 2，000 个，第 4.5 节中为 10，000 个。MH-PINN 主体中的权重采用 Glorot均匀初始化 [12] ，偏置则初始化为零；头部通过从标准差为 0.05 的正态随机分布中采样进行初始化，以加快正规化流的训练并提升学习到的生成器的性能。

除第 4.2 节中的正问题外，本文中的正规化流均采用 MAF [33] ，包含 10 个双射变换，即正规化流中的可逆映射。每个双射变换均为 MADE[11], ，即一种带有掩码全连接层的神经网络，包含两个非线性隐藏层，每层配备 100 个神经元，并使用 ReLU激活函数。第 4.2 节正问题中使用的 RealNVP [9] 和 IAF [20] 也包含 10 个双射变换，每个变换均为一种神经网络，包含两个非线性隐藏层，每层配备 100 个神经元，并使用 ReLU 激活函数。正规化流的实现主要遵循 TensorFlow Probability 库[8]中的说明。

第 4.2 节中实现的物理信息生成对抗网络 [44] 具有以下架构：判别器是一个前馈神经网络，包含 3 个非线性隐藏层，每层配备 128 个神经元，并使用 Leaky ReLU激活函数；以 t 为输入的生成器是一个前馈神经网络，包含 3 个非线性隐藏层，每层配备 50 个神经元，并使用双曲正切激活函数；另一个生成器以 50 维高斯随机变量为输入，该变量的均值为零，协方差矩阵为单位协方差矩阵；此生成器由一个前馈神经网络实现，包含 3 个非线性隐藏层，每层有 128 个神经元，并使用双曲正切激活函数。这 3 个前馈神经网络的输入维度分别为 65、1 和 50，输出维度分别为 1、50 和 50。

MH-PINNs采用全批量训练，并使用Adam优化器迭代50，000次。除第4.2节中的正问题外，NFs均采用小批量训练，批量大小为100，并使用Adam优化器训练1，000个周期。第4.2节正问题中的NFs则训练500个周期；此外，对RealNVP的参数施加 L_2 正则化，以获得更好的性能。对于所有NFs，为了实现稳定训练，对每个双射变换计算得到的尺度对数施加双曲正切函数，使尺度对数位于$(-1, 1)$内。PI-GANs采用小批量训练，批量大小为100，并使用Adam优化器迭代100，000次。此外，与 [44, 30], 物理信息Wasserstein生成对抗网络（PI-WGANs）相同，采用梯度惩罚，其系数设为0。1。训练过程中，每迭代一次先更新判别器5次，再更新生成器1次。除训练PI-GANs外，Adam优化器的学习率设为 10^{-3} ，其他超参数均采用默认值。训练PI-GANs时，判别器和生成器所用Adam优化器的学习率分别设为 10^{-4}、$\beta_1 = 0.5$ 和$\beta_2 = 0.9$ 。

MH-PINNs、NFs和PI-GANs的训练均在单块NVIDIA TITAN Xp GPU上完成。论文被接收后，L-HYDRA的TensorFlow实现代码及一些具有代表性的示例将发布在GitHub上。

附录 B. 贝叶斯推断的实施细节。 所有用于不确定性量化（UQ）的贝叶斯推断示例均采用哈密顿蒙特卡洛（HMC） [31] ，而拉普拉斯近似 [18] 仅用于第一个示例。本文采用自适应步长 [23] 的 HMC，初始步长设为 0.1 或 0.01，并进行调整以获得更高的接受率。预热样本数和后验样本数均设为 1,000。蛙跳方案的步数设为 30 或 50，同样经过调整以获得更高的接受率。本文使用用于科学机器学习中 UQ 的 NeuralUQ库 [47] ，作为在下游任务中进行物理信息贝叶斯推断的工具。正如 [30, 47], 所讨论的，HMC 的理想接受率约为 60%。本文发现，接受率在 50% 至 80% 之间的链均可接受。