基于傅里叶神经算子的时空建模与预测

Pratik Nag[1], Andrew Zammit-Mangion[1], Sumeetpal Singh[1], 和 Noel Cressie[1]

[1]伍伦贡大学数学与应用统计学院，澳大利亚

arXiv:2601.01813v1 [stat.ME] 5 Jan 2026

摘要

时空过程模型常用于对随空间和时间演变的动态物理现象和生物现象进行建模。这些现象可能表现出环境异质性和复杂的相互作用，难以通过高斯过程等传统统计过程模型加以捕捉。本文提出使用傅里叶神经算子（FNO）构建用于预测的统计时空动力模型。FNO 是一种灵活的函数映射，可高效近似可能未知的线性或非线性偏微分方程（PDE）的解算子。它利用输入样本及其对应的输出进行建模，因此无需明确了解底层偏微分方程。通过已知解的非线性偏微分方程模拟，我们将 FNO预测与最先进的统计时空预测方法进行比较。此外，利用大西洋海表温度数据和欧洲降水数据，我们展示了基于 FNO 的动态时空统计建模能够捕捉真实世界中复杂的时空依赖关系。通过一系列测试实例，我们证明 FNO-DST 的预测准确，且不确定性量化有效。

关键词： Burgers方程、格林函数、积分差分方程、非平稳性、偏微分方程。

1 引言

时空（ST）建模与数据分析通常是物理、环境和生物学研究的一部分。这类时空过程可能在空间和时间上都呈现复杂的依赖关系，因此难以建模。时空统计模型分为两

大类（Wikle 等人，2019年）。第一类是描述性时空模型，例如高斯过程（GP；例如，Cressie 和 Huang，1999年；Gneiting 和 Guttorp，2007；Cressie 和 Wikle，2011年；Banerjee 等人，2014），其中时间维度仅被视为一个额外维度，并以与空间维度相同的方式处理。这种方法往往面临计算可扩展性方面的挑战，也难以处理复杂的动力学。此外，这些描述性时空模型通常依赖于预先定义的时空协方差函数类别，从而限制了其灵活性。

第二类包括动态时空（DST）模型。这类模型通过指定连续时间步长中空间场之间及其内部的条件依赖关系来构建（例如，Cressie 和 Wikle，2011年，第 7 章）。这些模型为捕捉空间过程的时间演化提供了一个优雅的框架，因此在底层（物理、生物等）系统动力学复杂的应用中尤其有效。动态时空模型相比描述性时空模型具有明显优势，因为它们能够编码所建模的科学过程，因此非常适合预测任务。

一类被广泛用于建模时空动力系统的动态时空模型，是积分差分方程（IDE）模型（Wikle 和 Cressie，1999年；Wikle，2002）。这些模型通过积分算子来刻画未来时间步长的空间场与其当前状态之间的条件依赖关系。它们已成功应用于多个领域，例如生态学（例如，Neubert 等，1995）、气象学（例如，Xu 等，2005年；Calder 等，2011年）和流行病学（例如，Kot 和 Schaffer，1986）。

通常，动态时空模型假设存在一个线性算子，这简化了其数学表述。然而，在长时间范围内，线性假设可能并不适用，因为环境异质性和动态相互作用可能会导致强烈的

非线性依赖。解决这一问题的一种方法是采用基于二次非线性 IDE 的 DST 模型（Wikle 和 Hooten，2010 年；Wikle 和 Holan，2011年）。尽管这些模型更擅长捕捉复杂的非平稳行为，但在进行统计推断时，其计算效率较低。

为了降低拟合灵活的 DST 模型的计算成本，de B´ezenac et al. (2019年) 提出使用卷积神经网络（CNNs）从数据中学习动力学参数。他们的方法后来被Zammit-Mangion andWikle (2020年) 扩展到随机 IDE 模型。在这两项研究中，卷积神经网络都通过离线方式训练；训练完成后，参数估计及后续预测所需的计算成本都很低。该框架依赖于预先定义的控制线性偏微分方程（PDE），并利用其解构建 IDE。然而，许多现实世界应用涉及异质或高度非线性的动力学，无法通过线性偏微分方程得到很好的描述。

深度学习的最新进展彻底改变了研究人员对复杂系统进行建模的能力，尤其是那些由线性或非线性偏微分方程所支配的系统。其中一项进展是傅里叶神经算子（FNO；Li et al.，2020年），它是用于建模时空过程复杂物理动力学的基础模块。与显式求解偏微分方程的传统求解器不同，傅里叶神经算子利用训练数据，根据输入及其相应输出的样本，学习函数空间之间的映射。因此，它尤其适用于时空预测应用，因为这类应用的底层动力学可能并不为人所知。值得注意的是，与标准偏微分方程求解器相比，傅里叶神经算子的计算效率更高，因此是随机物理模型的一种颇具吸引力的替代方案。

本研究提出了一类通用的随机动态时空模型，利用傅里叶神经算子框架捕捉空间过程的时间演化；我们将其称为傅里叶神经算子-动态时空模型。本文通过模拟研究评估其预测性能

，其中包括 Burgers方程——一种非线性（非随机）偏微分方程，其预测结果可通过有限差分近似进行数值计算。随后，我们研究地球物理变量的预测。我们使用 FNO-DST 预测北大西洋的海表温度（SST）；研究表明，其动力学可通过对流-扩散偏微分方程很好地表示（de B´ezenac 等人，2019年）。在第二个应用中，我们使用 FNO-DST 预测欧洲某地区的降水，该地区的物理动力学复杂且未知。在这两项应用中，我们发现 FNO-DST模型能够捕捉复杂的时空相互作用，从而实现准确预测。

本文其余部分的结构如下：第2 节介绍傅里叶神经算子的建模与方法细节，并说明如何将其纳入FNO-DST模型。第3 节详细介绍模拟实验的设置，该实验比较FNO-DST时空预测与其他时空预测的性能，其中时空数据由Burgers方程生成，后者是非线性的。第4 节将FNO-DST方法应用于两个地球物理变量，并再次将其预测性能与其他方法进行比较。第5 节讨论主要发现及未来研究的潜在方向。本文附有在线补充材料。

2 FNO-DST模型

在第2.1节中，我们介绍积分微分方程模型以及一种采用神经算子的非线性推广。在第2.2节中，我们将介绍 FNO-DST 模型，作为第2.1节中所讨论的基于神经算子的模型的一种计算上易于处理的具体形式。第2.3节将介绍参数估计，具体来说是对 FNO-DST 模型的参数进行最大似然估计。

2.1 基于经典积分微分方程的动态时空模型

令 T $= [0,\ K]$ 表示连续时间域，令 D $\subset \mathbb{R}^d$ 表示空间域；在我们的示例中， d为 1 或 2。尽管许多针对我们关注的环境过程的模型都假设其在 D $\times$T 上连续演化，但为提高计算效率，通常会对其解进行离散化，从而得到空间过程的相关时间序列。我们在此采用相同的方法：令 $\{t_1,\ t_2,\ \ldots\ldots,\ t_T\}$为 T 中一组有限的时间点，间隔固定为 $t_{k+1}-t_k=\delta>0$，其中 $\delta \ll K$，且 $0 \leqslant t_1 \leqslant t_T \leqslant K$。对于由偏微分方程支配的时空过程，有时可以根据这些连续时间点处的积分解构造离散时间表示。作为示例，我们选取一个简单的线性平流-扩散偏微分方程：这里的 C(s t) 是在 D $\times [t_k,\ t_{k+h}]$上的实值连续时间空间过程，其演化遵循：

$$\frac{\partial \mathcal{C}(\mathbf{s},t)}{\partial t}+\gamma_1\sum_{i=1}^{d}\frac{\partial \mathcal{C}(\mathbf{s},t)}{\partial s_i}-\gamma_2\sum_{i=1}^{d}\frac{\partial^2 \mathcal{C}(\mathbf{s},t)}{\partial s_i^2}=0;\quad \mathbf{s}\in\mathcal{D},\quad t\in[t_k,t_{k+h}], \tag{1}$$

其中参数向量为 $\boldsymbol{\gamma}\equiv(\gamma_1、\ \gamma_2)^\top\in\mathbb{R}\times\mathbb{R}^+\equiv\Gamma$，且 h 是一个固定的正整数，使得终点为 $t_{k+h}\in$ T。对于固定且已知的 $\boldsymbol{\gamma}$、有界的 D，并且在 C(s t_k) 上满足适当的正则性条件（Brezis 和 Br´ezis，2011年；Evans，2022年）时，式（1）在终点 t_{k+h} 处的解涉及格林函数 $g(\cdot、\ \cdot)$，其形式为（Brauer，1967年；Pan，2009年）：

$$\mathcal{C}(\mathbf{s},t_{k+h})=\mathcal{G}_{\boldsymbol{\gamma}}[\mathcal{C}(\cdot,t_k)]\equiv\int_{\mathcal{D}}g(\mathbf{s}-\mathbf{u},h\delta;\boldsymbol{\gamma})\mathcal{C}(\mathbf{u},t_k)\mathrm{d}\mathbf{u};\quad \mathbf{s}\in\mathcal{D},k=1,2,\ldots,T-h. \tag{2}$$

在（2）中，我们用 Gγ 表示积分算子，它将输入函数 C$(\cdot,t_k)\in L^2$(D;$\mathbb{R}$) 映射为同一空间 L^2(D; $\mathbb{R}$)中的相应输出函数，其中 L^2(D; $\mathbb{R}$)是从 D 映射到 $\mathbb{R}$的平方可积函数空间。在（2）中，格林函数为 g(s $-$u $h\delta$; $\boldsymbol{\gamma}$) $=(4\pi\gamma_2 h\delta)^{-d/2}\exp\big[-\|\mathbf{s}-\mathbf{u}-\gamma_1 h\delta\,\mathbf{1}\|^2/(4\gamma_2 h\delta)\big]$,, 其推导见补充材料的 S.1 节。积分解（2）是空间过程的确定性线性

自回归过程，时间步长为 $h\delta$ ，起始空间场为 C(s t_k)。

现在，我们引入两个不确定性来源，将式（2）转化为随机过程。首先，在式（2）中加入空间相关随机项 η(s t_{k+h}),s $\in$D，以刻画所建模现象的变异性，这种变异性无法通过积分解加以解释。在本文中，假设空间过程 $\eta(\cdot, t_{k+h})$ and $\eta(\cdot, t_{k+h+k'})$ 在给定 $Y_k(\cdot,$ t_k)的条件下互不相关，其中 $k > 0$ 和 $k' < 0$ （它们在时间维度上的具体概率依赖关系将在式（25）中进一步说明）。此外，假设过程$\{\eta$(s $t_k)\}_k$ 关于空间索引 s 均为至少二阶可微，这与式（1）的解相同。其次，我们将式（2）中的参数 $\boldsymbol{\gamma}$ 建模为随机变量，并假设其与过程 $\eta(\cdot, \cdot)$ 完全独立，且在空间和时间维度上均如此。这些建模假设导出如下积分差分方程模型，

$$Y_{k+h}(\mathbf{s}) = \int_{\mathcal{D}} g(\mathbf{s}-\mathbf{u}, h\delta; \boldsymbol{\gamma}) Y_k(\mathbf{u}) \mathrm{d}\mathbf{u} + \eta(\mathbf{s}, t_{k+h}); \quad \mathbf{s} \in \mathcal{D},\ k = 1, 2, \ldots, T-h, \tag{3}$$

其中 $g(\cdot,\ \cdot)$ 是格林函数， $Y_k(\cdot) \equiv \mathcal{C}(\cdot,\ t_k)$ 对于 $k = 1$、.....、$T-h$以及 $Y_k(\cdot) \equiv Y(\cdot,$ t_k)是时间点 t_k处的随机过程。注意，给定 $\boldsymbol{\gamma}$后，式（3）中定义的 h 步过程在时间上是马尔可夫的。例如，当 $h = 1$时，给定 $\boldsymbol{\gamma}$后，过程$\{Y_k(\cdot)\}_k$ 在时间上是马尔可夫的。然而，我们将看到，从边缘分布来看，在给定当前状态的条件下，过程的过去与未来并不独立。

通常会采用 $Y_{k+h}(\cdot)$ 的有限维表示。这里，我们将$\mathcal{D}$离散为面积相同的基本面积单元(BAU)，面积为 Δ。令$\mathcal{D}^G \subset \mathcal{D}$ 表示 BAU 质心，并定义随机列向量$\mathbf{Y}_k \equiv (Y_k(\mathbf{s}) :\ \mathbf{s} \in \mathcal{D}^G)^\top$ 和 $\boldsymbol{\eta}_{k+h} \equiv (\eta(\mathbf{s}\ t_{k+h}) :\ \mathbf{s} \in \mathcal{D}^G)^\top$。将（3）中的积分近似为求和，可知（3）可表示为一个$|\mathcal{D}^G|$维向量自回归（VAR）过程，

$$\mathbf{Y}_{k+h} = \mathbf{G}_{\boldsymbol{\gamma}} \mathbf{Y}_k + \boldsymbol{\eta}_{k+h}; \quad k = 1, 2, \ldots, T-h, \tag{4}$$

其中 $\mathbf{G}_{\boldsymbol{\gamma}} \equiv (g(\mathbf{s} - \mathbf{u}\ h\delta;\ \boldsymbol{\gamma})\Delta : \mathbf{s},\mathbf{u} \in \mathbf{D}^G)$ 是一个 $|\mathbf{D}^G| \times |\mathbf{D}^G|$ 矩阵，并且回顾一下，$\boldsymbol{\gamma}$，因此 $\mathbf{G}\gamma$ 是随机的。现在假设 $\boldsymbol{\eta}_{k+h}$ 是一个均值为零的 $|\mathbf{D}^G|$维高斯向量，且 $\text{var}(\boldsymbol{\eta}_{k+h} \mid \mathbf{Y}_k) = \boldsymbol{\Sigma}_{k+h}(\mathbf{Y}_k; \boldsymbol{\alpha})$ 参数为 $\boldsymbol{\alpha}$。此外，假设对于 $k = (h+1),\ldots,(T-h)$，$\text{cov}(\boldsymbol{\eta}_k, \boldsymbol{\eta}_{k+h} \mid \mathbf{Y}_k) = \mathbf{0}$。那么，

$$\mathbf{Y}_{k+h} \mid \mathbf{Y}_k, \boldsymbol{\gamma} \sim \text{Gau}\left(\mathbf{G}_{\boldsymbol{\gamma}}\mathbf{Y}_k, \boldsymbol{\Sigma}_{k+h}(\mathbf{Y}_k; \boldsymbol{\alpha})\right); \quad k = 1, 2, \ldots, T-h. \tag{5}$$

在本文中，我们主要关注根据截至并包括 t_k时刻的信息，对$\mathbf{Y}_{k+h}$ 在 t_{k+h} 时的过程进行预测。仅根据$\mathbf{Y}_k$ 在 t_k 时的信息得到的预测分布为

$$p(\mathbf{Y}_{k+h} \mid \mathbf{Y}_k) = \int_\Gamma p(\mathbf{Y}_{k+h} \mid \mathbf{Y}_k, \boldsymbol{\gamma}) p(\boldsymbol{\gamma} \mid \mathbf{Y}_k) \mathrm{d}\boldsymbol{\gamma}. \tag{6}$$

基于（6）给出的预测分布进行预测可能价值不大，因为仅凭 $\boldsymbol{\gamma}$ 时的 $\mathbf{Y}_k$ ，关于动力学参数的信息 k十分有限。也就是说，（ $p(\boldsymbol{\gamma} \mid \mathbf{Y}_k) \approx p(\boldsymbol{\gamma})$ （6）中。

直观而言，可以通过纳入截至时间滞后 τ的近期历史信息来学习动力学参数:

$$p(\mathbf{Y}_{k+h} \mid \mathbf{Y}_k, \ldots, \mathbf{Y}_{k-\tau}) = \int_\Gamma p(\mathbf{Y}_{k+h} \mid \mathbf{Y}_k, \boldsymbol{\gamma}) p(\boldsymbol{\gamma} \mid \mathbf{Y}_k, \ldots, \mathbf{Y}_{k-\tau}) \mathrm{d}\boldsymbol{\gamma}, \tag{7}$$

其中，现在利用 $\boldsymbol{\gamma}$ 的条件分布来收紧预测分布。现在，随着 τ 增大，可以预期 $\boldsymbol{\gamma}$ 的条件分布会将其概率集中在真实参数附近，我们将其记为 $\boldsymbol{\gamma}^o$；也就是说，由（7）可得，$p(\mathbf{Y}_{k+h} \mid \mathbf{Y}_k,\ \ldots\ldots,\ \mathbf{Y}_{k-\tau}) \approx p(\mathbf{Y}_{k+h} \mid \mathbf{Y}_k,\ \boldsymbol{\gamma}^o)$当 τ较大时。这番简要讨论表明，在缺乏对支配系统的动力学参数的了解时，开发时空预测模型必须使用包含过去信息的 $\tau > 0$ 取值，即使潜在时空模型在时间维度上满足条件马尔可夫性。

有几种方法可以将这种历史依赖性纳入时间动态变化的

具有隐含动力学参数的时空模型。方法之一是将（3）中的空间卷积替换为按时间求和的 $(\tau+1)$ 空间卷积，从而得到

$$Y_{k+h}(\mathbf{s}) = \sum_{j=0}^{\tau} \int_{\mathcal{D}} g_j(\mathbf{s}-\mathbf{u}, h\delta; \boldsymbol{\theta}') Y_{k-j}(\mathbf{u}) \mathrm{d}\mathbf{u} + \eta(\mathbf{s}, t_{k+h}); \quad \mathbf{s} \in \mathcal{D}, \quad k = \tau+1, \ldots, T-h, \tag{8}$$

其中 $\{g_j(\cdot、\ h\delta;\ \boldsymbol{\theta}')\}_{j=0}^{\tau}$ 是由 $\boldsymbol{\theta}'$参数化的核。式（8）在$\tau = 0$ 且 $g_0(\cdot、\ \cdot) = g(\cdot、\ \cdot)$时，简化为式（3）。与推导式（4）时一样，我们可以将 $\mathcal{D}$ 离散化为 $\mathcal{D}^G$，从而得到一个向量自回归过程：

$$\mathbf{Y}_{k+h} = \sum_{j=0}^{\tau} \mathbf{G}_{j,\boldsymbol{\theta}'} \mathbf{Y}_{k-j} + \boldsymbol{\eta}_{k+h}; \quad k = \tau+1, \ldots, T-h, \tag{9}$$

其中 $\mathbf{G}_{j、\ \boldsymbol{\theta}'} \equiv (g_j(\mathbf{s}\ -\mathbf{u}\ h\delta;\ \boldsymbol{\theta}')\Delta:\ \mathbf{s}、\ \mathbf{u}\ \in \mathrm{D}^G)$ 是一个 $|\mathrm{D}^G| \times |\mathrm{D}^G|$ 矩阵。

另一种纳入历史依赖性的方法，是使式（8）中的线性性得到放宽，即令式（3）中的核依赖于状态；这将得到如下形式的状态依赖VAR过程（Cressie and Wikle，2011年，第7.3.2节；Zammit-Mangion and Wikle，2020年）：

$$\mathbf{Y}_{k+h} = \mathbf{G}_{\boldsymbol{\theta}''}(\mathbf{Y}_k, \ldots, \mathbf{Y}_{k-\tau}) \mathbf{Y}_k + \boldsymbol{\eta}_{k+h}; \quad k = \tau+1, \ldots, T-h. \tag{10}$$

尽管这两种方法都能构建灵活的时空模型类别，但在实际实现中，需要对（9）和（10）中的矩阵施加结构和限制，并采用恒定的规则网格划分。拟合这些模型也需要大量计算。接下来讨论的基于傅里叶神经算子（FNO）的模型，能够让建模者避免其中若干缺点。具体而言，这类模型对于连续算子具有通用逼近性质（Kovachki等人，2023年）；可以轻松适配 $\mathcal{D}$ 上不同的规则网格划分；并且得益于快速傅里叶变换，拟合和预测速度极快。

2.2 FNO-DST 模型的高效计算

鉴于上一节关于隐藏动力学的讨论，对于底层动力学未预先指定的时空模型，一般而言应包含马尔可夫阶数大于 1 的模型类别。这里我们考虑一个灵活的算子，它将时空过程 $Y(\cdot, \cdot)$ 在 $L^2(\mathbf{D} \times [t_{k-\tau},\ t_k]$ 上的样本路径映射到 $L^2(\mathbf{D} \times \{t_{k+h}\};\ \mathbb{R})$ 对任意 $k \in \{\tau+1,\ \ldots\ldots,\ T-h\}$。请注意，输入的函数空间使我们能够将历史依赖性纳入算子，在这里是通过混合时间卷积来实现的。具体而言，下面定义的 FNO 的核心是一个时空卷积，其形式为

$$v(\mathbf{s}, t_{k+h}) = \int_{t_{k-\tau}}^{t_k} \int_{\mathcal{D}} g(\mathbf{s}-\mathbf{u}, t_{k+h}-r; \boldsymbol{\gamma}) v(\mathbf{u}, r)\, \mathrm{d}\mathbf{u}\, \mathrm{d}r, \tag{11}$$

其中 $v \in L^2(\mathcal{D} \times \mathcal{T};\ \mathbb{R})$ 且 $g(\cdot,\ \cdot;\ \boldsymbol{\gamma})$ 是由 $\boldsymbol{\gamma}$ 参数化的格林函数。由（11）可知，该算子是卷积，因此是时间不变的。因此，通过平移 $v(\cdot,\ \cdot)$ 的时间坐标，积分（11）中的时间区间可以选为 $[0,\ \tau\delta]$:

$$v(\mathbf{s}, t_{k+h}) = \int_{0}^{\tau\delta} \int_{\mathcal{D}} g(\mathbf{s}-\mathbf{u}, (h+\tau)\delta - q; \boldsymbol{\gamma}) v_0^{(k)}(\mathbf{u}, q)\, \mathrm{d}\mathbf{u}\, \mathrm{d}q, \tag{12}$$

其中，经过时间平移的时空过程 $v_0^{(k)}(\mathbf{u}\ t)$ 定义为

$$v_0^{(k)}(\mathbf{u}, t) \equiv v(\mathbf{u}, t + t_{k-\tau}).$$

于是 $v(\mathbf{s}\ t_{k+h}) \equiv \mathcal{G}_0[v_0^{(k)}]$，其中 $\mathcal{G}_0[\cdot]$ 是由（ k）定义的时间不变算子。由此，无损一般性地，我们可以将（11）中的时变算子表示为作用于时空过程平移版本上的时间不变算子，这相当于隐含地假设底层物理定律在时间上不变。

现在定义 $\mathsf{G}_0[\cdot] \equiv \mathsf{G}_{\boldsymbol{\theta}}[\cdot]$，其中 $\boldsymbol{\theta}$ 是用于参数化傅里叶神经算子的参数。傅里叶神经算子对 $\mathsf{G}_{\boldsymbol{\theta}}$采用通用的（非线性）算子模型，该模型具有通用逼近性质

如 Kovachki等人(2023年):

$$\mathcal{G}_{\boldsymbol{\theta}}[Y_0^{(k)}] = \mathcal{Q} \circ \sigma_{\mathrm{a}}(\mathcal{W}_L + \mathcal{K}_L) \circ \cdots \circ \sigma_{\mathrm{a}}(\mathcal{W}_1 + \mathcal{K}_1) \circ \mathcal{P}[Y_0^{(k)}];\ Y_0^{(k)} \in L^2(\mathcal{D} \times [0,\ \tau\delta];\ \mathbb{R}) \tag{13}$$

其中 $Y_0^{(k)}(\cdot,\ t) \equiv Y(\cdot,\ t + t_{k-\tau})$ 对于 $t \in [0,\ \tau\delta]$。方程（13）由三个主要组成部分构成，下面将详细介绍。

- **提升算子 P：** 算子 $\mathrm{P}[Y_0^{(k)}]$ 是将 $Y_0^{(k)}$ 的图逐点映射到高维空间 $\mathrm{D}_v \subseteq \mathbb{R}^{d_v}$ 中的映射，其中 $d_v > d + 2$。具体而言，对于$\mathbf{x} \equiv (\mathbf{s}^\top,\ t)^\top \in \mathrm{D} \times [0,\ \tau\delta]$，令 $\mathbf{x}_Y^{(k)} \equiv (\mathbf{x}^\top,\ Y_0^{(k)}(\mathbf{s}\ t))^\top$，它是一个 $(d+2)$ 维列向量。在我们的实现中，

$$\mathbf{v}_0^{(k)}(\mathbf{x}) \equiv \mathcal{P}[Y_0^{(k)}](\mathbf{x}) = \mathbf{H}_{\mathcal{P}}\mathbf{x}_Y^{(k)} + \mathbf{b}_{\mathcal{P}};\quad \mathbf{x} \in \mathcal{D} \times [0, \tau\delta], \tag{14}$$

其中，$\mathbf{H}_{\mathcal{P}}$ 是一个 $d_v \times (d+2)$ 矩阵，$\mathbf{b}_{\mathcal{P}}$ 是一个 d_v 维向量。我们将算子 $\mathrm{P}[\,\cdot\,]$ 中的参数收集到向量 $\boldsymbol{\phi}_{\mathcal{P}} \equiv \left(\mathrm{vec}(\mathbf{H}_{\mathcal{P}})^\top,\ \mathbf{b}_{\mathcal{P}}^\top\right)^\top$ 中。

- **核积分算子 K_l 和局部线性算子 W_l：** 对于层 l =1、...、L，算子 $\mathrm{K}_l[\,\cdot\,]$ 和 $\mathrm{W}_l[\,\cdot\,]$ 是神经算子的核心组成部分。具体而言，$\mathrm{W}_l[\,\cdot\,]$ 是对 d_v 维函数 $\mathbf{v}_{l-1}^{(k)}(\cdot)$进行的如下线性变换，

$$\mathcal{W}_l[\mathbf{v}_{l-1}^{(k)}](\mathbf{x}) = \mathbf{A}_{\mathcal{W},l}\,\mathbf{v}_{l-1}^{(k)}(\mathbf{x}) + \mathbf{b}_{\mathcal{W},l};\quad \mathbf{x} \in \mathcal{D} \times [0, \tau\delta], \tag{15}$$

其中，$\mathbf{A}_{\mathcal{W},\,l}$ 是一个 $d_v \times d_v$ 矩阵，$\mathbf{b}_{\mathcal{W},\,l}$ 是一个 d_v维向量。我们将$\mathcal{W}_l[\,\cdot\,]$的参数收集到向量$\boldsymbol{\phi}_{\mathcal{W},\,l} \equiv \left(\mathrm{vec}(\mathbf{A}_{\mathcal{W},\,l})^\top,\ \mathbf{b}_{\mathcal{W},\,l}^\top\right)^\top$。积分算子$\mathcal{K}_l[\,\cdot\,]$是（12）中特例的多元版本，定义如下：

$$\mathcal{K}_l[\mathbf{v}_{l-1}^{(k)}](\mathbf{x}) = \int_{\mathcal{D}\times[0,\tau\delta]} \mathbf{G}_l(\mathbf{x} - \mathbf{x}';\boldsymbol{\phi}_{\mathcal{K},l})\,\mathbf{v}_{l-1}^{(k)}(\mathbf{x}')\,\mathrm{d}\mathbf{x}';\quad \mathbf{x} \in \mathcal{D} \times [0, \tau\delta], \tag{16}$$

其中 $\mathbf{G}_l(\cdot;\ \boldsymbol{\phi}_{\mathcal{K},\,l})$ 是一个由核组成的 $d_v \times d_v$ 矩阵，其参数收集在向量 r

$\phi_{\mathcal{K},\,l}$。最后，第 l层的输出记为$\mathbf{v}_l^{(k)}$，它是一个 d_v维函数，

$$\mathbf{v}_l^{(k)}(\mathbf{x}) \equiv \sigma_{\mathrm{a}}\left(\mathcal{W}_l[\mathbf{v}_{l-1}^{(k)}](\mathbf{x}) + \mathcal{K}_l[\mathbf{v}_{l-1}^{(k)}](\mathbf{x})\right); \quad l = 1, \ldots, L, \quad \mathbf{x} \in \mathcal{D} \times [0, \tau\delta], \tag{17}$$

其中，回顾一下，对于某个固定的 k，$\sigma_{\mathrm{a}}(\cdot)$ 是一个逐点激活函数，使算子呈现非线性。在我们的例子中，我们对 $\sigma_{\mathrm{a}}(\cdot)$使用修正线性单元（ReLU），它返回输入向量，将负分量设为零，并保持正分量不变。

- **投影算子 Q：** 该算子将最终层的 d_v维输出投影回目标输出空间。我们通过以下方式实现该算子：

$$\mathcal{Q}[\mathbf{v}_L^{(k)}](\mathbf{x}) = \mathbf{h}_{\mathcal{Q}}^{\top}\,\mathbf{v}_L^{(k)}(\mathbf{x}) + b_{\mathcal{Q}}; \quad \mathbf{x} \in \mathcal{D} \times [0, \tau\delta], \tag{18}$$

其中，$\mathbf{h}_{\mathcal{Q}}$ 是一个 d_v维向量，而 $b_{\mathcal{Q}}$ 是一个标量。我们将算子 $\mathrm{Q}[\,\cdot\,]$ 中的参数收集到向量 $\boldsymbol{\phi}_{\mathcal{Q}} \equiv (\mathbf{h}_{\mathcal{Q}}^{\top},\ b_{\mathcal{Q}})^{\top}$中。

回顾一下，在适当的正则条件下，由（13）给出的神经算子具有通用逼近性质。傅里叶神经算子（FNO）同样如此，其中算子 $\mathcal{K}_l[\,\cdot\,]$ 表示为傅里叶-逆傅里叶对。卷积的傅里叶变换是傅里叶变换的乘积，因此，

$$\mathcal{K}_l\left[\mathbf{v}_{l-1}^{(k)}\right](\mathbf{x}) = \mathcal{F}^{-1}\big[\mathcal{F}[\mathbf{G}_l(\cdot;\boldsymbol{\phi}_{\mathcal{K},l})]\mathcal{F}[\mathbf{v}_{l-1}^{(k)}]\big](\mathbf{x}), \tag{19}$$

其中傅里叶算子和逆傅里叶算子为

$$U(\boldsymbol{\omega}) \equiv \mathcal{F}[u](\boldsymbol{\omega}) = \int_{\mathcal{D}\times[0,\tau\delta]} u(\mathbf{x})\exp(-\iota\boldsymbol{\omega}^{\top}\mathbf{x})\mathrm{d}\mathbf{x}; \quad \boldsymbol{\omega} \in \mathbb{R}^{d+1}, \tag{20}$$

$$\mathcal{F}^{-1}[U](\mathbf{x}) = \tfrac{1}{(2\pi)^{d+1}}\int_{\mathbb{R}^{d+1}} U(\boldsymbol{\omega})\exp(\iota\boldsymbol{\omega}^{\top}\mathbf{x})\mathrm{d}\boldsymbol{\omega}; \quad \mathbf{x} \in \mathcal{D} \times [0, \tau\delta], \tag{21}$$

其中 $\boldsymbol{\omega}$ 表示 s　时空角频率和 $u(\cdot)$ 一个 L^2(维函数 $\times$ $[0,\ \tau\delta]$; $\mathbb{R}$).

定义：方程（13），以及式（$\mathcal{K}_l[\cdot]$中将其表示为一对傅里叶变换与逆傅里叶变换的表达式（19），共同定义了傅里叶神经算子（FNO），它将 L^2(维 $\times$ $[0,\ \tau\delta]$; $\mathbb{R}$)中的函数映射为 L^2(维 $\times$ $\{(h+\tau)\delta\}$; $\mathbb{R}$)中的函数。

快速计算是任何神经算子的关键，在傅里叶神经算子中，我们通过使用库利-图基快速傅里叶变换（快速傅里叶变换）（Cooley 和 Tukey，1965）来实现这一点。具体而言，我们不使用（20）和（21），而是从离散傅里叶变换开始：

$$U(\boldsymbol{\kappa}) = \sum_{\mathbf{x}\in\mathcal{D}^G\times\mathcal{T}_\tau^G} u(\mathbf{x}) \exp\left(-2\pi\iota \sum_{j=1}^{d+1} \frac{\kappa_j x_j}{m_j}\right); \quad \boldsymbol{\kappa}\in\Omega, \tag{22}$$

$$u(\mathbf{x}) = \frac{1}{m_1\cdots m_{d+1}} \sum_{\boldsymbol{\kappa}\in\Omega} U(\boldsymbol{\kappa}) \exp\left(2\pi\iota \sum_{j=1}^{d+1} \frac{\kappa_j x_j}{m_j}\right); \quad \mathbf{x}\in\mathcal{D}^G\times\mathcal{T}_\tau^G, \tag{23}$$

其中，$\mathbf{T}_\tau^G \equiv \{0,\ \delta,\ \ldots\ldots,\ \tau\delta\}$; m_j是第 j个时空坐标中的网格间距数，因此 $m_{d+1}=\tau$; $\Omega \equiv \times_{j=1}^{d+1}(0,\ \ldots\ldots,\ m_j-1)$。随后，为了提高计算效率，我们将 Ω 缩减为 $\tilde{\Omega} \equiv \times_{j=1}^{d+1}(0,\ \ldots\ldots,\ \tilde{m}_j-1)$，其中 $\tilde{m}_j \ll m_j,\ \ j=1,\ \ldots d+1$。因此，傅里叶变换的乘积可归结为 $O(\prod_{j=1}^{d+1}\tilde{m}_j)$ 个维度为 d_v的矩阵-向量乘法，并且可以非常快速地计算；参见补充材料中的S.1.3节，其中简要介绍了快速傅里叶变换算法。

在我们的傅里叶神经算子实现中，我们不会为 $\mathbf{G}_l(\cdot;\boldsymbol{\phi}_{\mathcal{K},l})$ 在方程（19）中指定参数化形式后再求其离散傅里叶变换。相反，我们将（19）中的 $\mathbf{F}[\mathbf{G}_l]$ 定义为傅里叶域中的复数谱权重张量，并使其具有共轭对称性（因为信号被视为实值信号）。我们用 $\mathbf{F}_{\mathcal{K},l} \in \mathbb{C}^{d_v\times d_v\times\tilde{m}_1\times\cdots\times\tilde{m}_{d+1}}$ 表示这些张量，它们位于复数构成的高维空间中，是可训练参数；我们将它们收集到向量$\boldsymbol{\phi}_{\mathcal{K},l} \equiv \mathbf{vec}(\mathbf{F}_{\mathcal{K},l})$中。它们赋予傅里叶神经算子所需的灵活性，使其能够表示复杂的时空动力学。

令 $\boldsymbol{\theta} = \left(\boldsymbol{\phi}_{\mathcal{P}}^\top,\ \boldsymbol{\phi}_{\mathcal{W},1}^\top,\ \boldsymbol{\phi}_{\mathcal{K},1}^\top,\ \ldots\ldots,\ \boldsymbol{\phi}_{\mathcal{W},L}^\top,\ \boldsymbol{\phi}_{\mathcal{K},L}^\top,\ \boldsymbol{\phi}_{\mathcal{Q}}^\top\right)^\top$ 表示与式（13）中的神经算子相关联的可训练参数集合。令 $\mathbf{G}(\cdot;\ \boldsymbol{\theta})$ 表示采用离散傅里叶变换（22）实现的神经算子，它也便于进行有限维参数化

。现在，对于 $k = \tau + 1, \ldots, T - h$，定义向量值随机时间序列 $\mathbf{Y}_{(k-\tau):k} \equiv \{\mathbf{Y}_{k-\tau}, \ldots, \mathbf{Y}_k\}$，其中 $|\mathbf{D}^G|$ 维向量 $\mathbf{Y}_k$ 在 $\mathbf{D}^G$ 处取值为 $Y_k(\cdot)$。于是，我们可以将离散化域上式（13）中定义的算子输出改写为 $\mathbf{G}(\mathbf{Y}_{(k-\tau):k};\ \boldsymbol{\theta})$。关于这些离散化张量的更多细节，请参见补充材料的 S.1.2 节。算子 $\mathbf{G}_{\boldsymbol{\theta}}[\cdot]$ in 的离散化求值 $\mathbf{G}(\cdot\,;\ \boldsymbol{\theta})$ 是一个 $|\mathbf{D}^G|$ 维向量，它按如下方式刻画潜在复杂物理过程的时空动力学：

(13)

$$\mathbf{Y}_{k+h} = \mathbf{G}(\mathbf{Y}_{(k-\tau):k};\, \boldsymbol{\theta}) + \boldsymbol{\eta}_{k+h}; \quad k = \tau + 1, \ldots, T - h. \tag{24}$$

这应与公式（10）进行比较，其中算子 $\mathbf{G}_{\boldsymbol{\theta}''}(\mathbf{Y}_k$、……、$\mathbf{Y}_{k-\tau})\mathbf{Y}_k$ 受到相对严格的约束。由于公式（24）中定义的过程模型是动态时空（DST）模型，我们将其称为 *FNO-DST* 模型。

现在来看加性项 $\{\boldsymbol{\eta}_{\tau+1+h}$、……、$\boldsymbol{\eta}_T\}$。我们使用具有空间和时间变化标准差的平方指数协方差函数，对条件协方差矩阵 $\boldsymbol{\Sigma}_{k+h}(\mathbf{Y}_k; \boldsymbol{\alpha})$ 的 $\boldsymbol{\eta}_{k+h}$ 进行建模，表示为

$$\operatorname{cov}\left(\eta(\mathbf{s}, t_{k+h}), \eta(\mathbf{s}', t_{k+h}) \mid \mathbf{Y}_k\right) = \sigma_{k+h}(\mathbf{s}; \mathbf{Y}_k)\, \sigma_{k+h}(\mathbf{s}'; \mathbf{Y}_k)\, \exp\left(-\frac{\|\mathbf{s} - \mathbf{s}'\|^2}{2\alpha_{\mathrm{r}}^2}\right), \tag{25}$$

对于 $\mathbf{s}$、$\mathbf{s}' \in \mathbf{D}^G.$，我们通过将时间 t_{k+h} 处的标准差参数建模为过程$\mathbf{Y}_k$的函数来处理异方差性。具体而言，令

$$\left(\sigma_{k+h}(\mathbf{s}_i; \mathbf{Y}_k) : \mathbf{s}_i \in \mathcal{D}^G\right)^\top = f_{\mathrm{NN}}(\mathbf{Y}_k; \boldsymbol{\alpha}_{\mathrm{NN}}), \tag{26}$$

其中，$f_{\mathrm{NN}}(\cdot)$ 是一个浅层前馈神经网络，以$\mathbf{Y}_k$ 为输入，并输出空间网格上各点的标准差向量。这指定了 $\boldsymbol{\Sigma}_{k+h}(\mathbf{Y}_k; \boldsymbol{\alpha})$，其中$\boldsymbol{\alpha} \equiv (\mathrm{vec}(\boldsymbol{\alpha}_{\mathrm{NN}})^\top, \alpha_{\mathrm{r}})^\top$ 是 FNO-DST 模型中的其他可训练参数。

2.3 似然计算、训练与预测

考虑在 D $\times [0, K]$ 上由时空过程模型生成的 N 个独立实现，例如通过随机初始条件或动力学参数生成。假设对于 $n = 1, \ldots, N$，我们在时间点 $\{t_1, \ldots, t_T\}$ 观测每个过程，时间步长为 δ，从而得到离散时间序列 $\{\mathbf{Y}_1^{(n)}, \ldots, \mathbf{Y}_T^{(n)}\}$。在给定 τ 观测历史的条件独立性假设下，未知参数 $(\boldsymbol{\theta}^\top, \boldsymbol{\alpha}^\top)^\top$ 的似然函数可写为

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \prod_{n=1}^{N} \prod_{k=\tau+1}^{T-h} p\big(\mathbf{Y}_{k+h}^{(n)} \mid \mathbf{Y}_{(k-\tau):k}^{(n)}, \boldsymbol{\theta}, \boldsymbol{\alpha}\big) = \prod_{n=1}^{N} \prod_{k=\tau+1}^{T-h} \mathcal{L}_k^{(n)}(\boldsymbol{\theta}, \boldsymbol{\alpha}), \tag{27}$$

其中 $\mathbf{Y}^{(n)}_{(k-\tau):k} \equiv \{\mathbf{Y}_{k-\tau}^{(n)}$、……、$\mathbf{Y}_k^{(n)}\}$。根据式（24）给出的高斯向量自回归模型，

$$\mathrm{Y}_{k+h}^{(n)} \mid \mathrm{Y}_{(k-\tau)}^{(n)}{:}_k, \boldsymbol{\theta}, \boldsymbol{\alpha} \sim \mathrm{Gau}\Big(\mathrm{G}(\mathrm{Y}_{(k-\tau)}^{(n)}{:}_k;\ \boldsymbol{\theta}), \boldsymbol{\Sigma}_{k+h}^{(n)}(\mathrm{Y}_k;\ \boldsymbol{\alpha})\Big);\ n = 1, \ldots, N, k = \tau+1, \ldots, T-h, \tag{28}$$

其中 $\boldsymbol{\Sigma}_{k+h}^{(n)}(\mathrm{Y}_k; \boldsymbol{\alpha})$ 是 $\boldsymbol{\eta}_{k+h}^{(n)}$ 的协方差矩阵，其参数由 $\boldsymbol{\alpha}$表示。我们使用带有 Adam 学习率调度方案的标准优化算法最大化 $\mathrm{L}(\boldsymbol{\theta}, \boldsymbol{\alpha})$ （Kingma，^2014）。令 $(\boldsymbol{\theta}, \widehat{\boldsymbol{\alpha}})$ 表示使 L 最大化的估计参数。将这些估计值代入方程（28），即可得到重复样本 i的预测分布：

$$\mathrm{Y}_{k+h}^{(n)} \mid \mathrm{Y}_{(k-\tau)}^{(n)}{:}_k, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\alpha}} \sim \text{高斯}(\mathrm{G}(\mathrm{Y}_{(k-\tau)}^{(n)}{:}_k;\ \widehat{\boldsymbol{\theta}}), \boldsymbol{\Sigma}_{k+h}^{(n)}(\mathrm{Y}_k;\ \widehat{\boldsymbol{\alpha}}));\ n = 1, \ldots, N, k = \tau+1, \ldots, T-h, \tag{29}$$

其中 $\mathrm{G}(\mathrm{Y}_{(k-\tau)}^{(n)}{:}_k;\ \widehat{\boldsymbol{\theta}})$是估计的预测均值， $\boldsymbol{\Sigma}_{k+h}^{(n)}(\mathrm{Y}_k;\ \widehat{\boldsymbol{\alpha}})$是相应的估计预测协方差矩阵。

3 仿真实验

在本节中，我们使用 Burgers方程的二维（1-D 空间 $\times$ 时间）模拟，对比分析所提出的 FNO-DST方法与当前最先进方法的预测性能。Burgers方程是一种非线性偏微分方程，其解可以

通过数值方法求得。比较所用的性能指标包括均方预测误差（MSPE）、预测区间覆盖概率（PICP）和平均预测区间宽度（MPIW）；有关 PICP 和 MPIW 的更多详情，请参阅 Lakshminarayanan 等人(2017) 以及补充材料 S.2.2节。在仿真实验中，我们将FNO-DST模型的预测与以下方法的预测进行比较：

- 卷积长短期记忆 (ConvLSTM) 神经网络模型（Shi 等人，2015），该模型在神经网络内部的 LSTM 模块中采用卷积转换。为训练该模型，我们将$\mathbf{G}(\cdot\,;\ \boldsymbol{\theta})$ 中的24替换为 ConvLSTM 块。

- 时空深度克里金 (STDK) 模型（Nag 等人，2023年），该模型通过神经网络学习嵌入的时空位置 $(\mathbf{s}^\top, t)^\top$，对于 $\mathbf{s} \in \mathcal{D}^G$ 以及 $t \in \{t_1, \ldots\ldots, t_T\}$与底层过程 $Y(\mathbf{s}\ t)$之间的关系，从而进行时空预测。Nag 等人（2023）表明，对于复杂时空过程的预测，这种方法优于另一种基于高斯回归和 Vecchia 近似的方法（Katzfuss 和 Guinness, 2021）。

- 持续性 模型是一种简单模型，它将当前序列最后一个时间点的空间观测值用作对应位置未来时间点的预测值。这种方法可视为一个低门槛基准，其他模型的预测应当优于该基准。

可复现的代码可从 https://github.com/pratiknag/FNO-DSTM-Code获取。

3.1 二维实现

我们考虑二维（1-D 空间 $\times$ 时间）非稳态 Burgers方程（Hopf, 1950），这是一种非线性偏微分方程，已应用于各种领域，例如

对黏性流体的一维流动进行建模。Burgers方程的解可以通过数值方法求得，并可作为金标准，用来比较上述四种模型的预测。描述时空场 $\{u(s,\ t)\colon\ s \in [0,\ 1],\ t \in [0,\ 1]\}$ (即 D $= [0,\ 1]$ 和 T $= [0,\ 1]$) 的演化的 Burgers方程在此写作：

$$\frac{\partial u(s,t)}{\partial t} + \frac{1}{2}\frac{\partial u^2(s,t)}{\partial s} = \gamma \frac{\partial^2 u(s,t)}{\partial s^2};\quad (s,t) \in (0,1) \times (0,1), \tag{30}$$

初始条件为 $u(s、\ 0) = a_0(s)$，其中 $s \in [0、\ 1]$；周期性边界条件为 $u(0、\ t) = u(1、\ t)$，其中 $t \in [0、\ 1]$。这里，$\gamma \in [0、\ 1]$ 是黏度参数。为了在给定 γ 和 $a_0(\cdot)$的情况下求解（30），我们采用欧拉数值方法（Biswas等，2013年），将偏微分方程离散到细网格上，具体如下：我们用步长为 $\Delta t = 1/10000$, 的前向差分替代时间导数，用步长为 $\Delta s = 1/2^{11}$, 的后向差分替代空间一阶导数，并用相同步长的对称二阶差分替代空间二阶导数。也就是说，（30）的离散化结果为：

$$\frac{u(s_i,\tilde{t}_{j+1}) - u(s_i,\tilde{t}_j)}{\Delta t} + u(s_i,\tilde{t}_j)\frac{u(s_i,\tilde{t}_j) - u(s_{i-1},\tilde{t}_j)}{\Delta s} = \gamma \frac{u(s_{i+1},\tilde{t}_j) - 2u(s_i,\tilde{t}_j) + u(s_{i-1},\tilde{t}_j)}{(\Delta s)^2}, \tag{31}$$

其中 $s_0 = \tilde{t}_0 = 0$，周期性边界条件为 $u(s_0,\ \tilde{t}j) = u(s_{2^{11}},\ \tilde{t}_j)$，以及均匀网格$(s_i,\ \tilde{t}_j) = (i/2^{11},,j/10000)$ 取 $i = 1、\ldots\ldots、2_{11}$ 和 $j = 1、\ldots\ldots、$ 10000。因此，Burgers方程离散后，通过重新排列（31）中的各项，可以改写为一步超前预测，如下所示：

$$\begin{aligned} u(s_i,\tilde{t}_{j+1}) = u(s_i,\tilde{t}_j) - \tfrac{\Delta t}{\Delta s}u(s_i,\tilde{t}_j)\left(u(s_i,\tilde{t}_j) - u(s_{i-1},\tilde{t}_j)\right) + \\ \gamma \tfrac{\Delta t}{(\Delta s)^2}\left(u(s_{i+1},\tilde{t}_j) - 2u(s_i,\tilde{t}_j) + u(s_{i-1},\tilde{t}_j)\right), \end{aligned} \tag{32}$$

˜ ˜其中，在 t_{j+1} 处的预测基于当前时间的局部空间信息 t_j。（注意，我们使用 $\tilde{t}_j$ 表示模拟器时间点，以区别于 FNO-DST 时间点

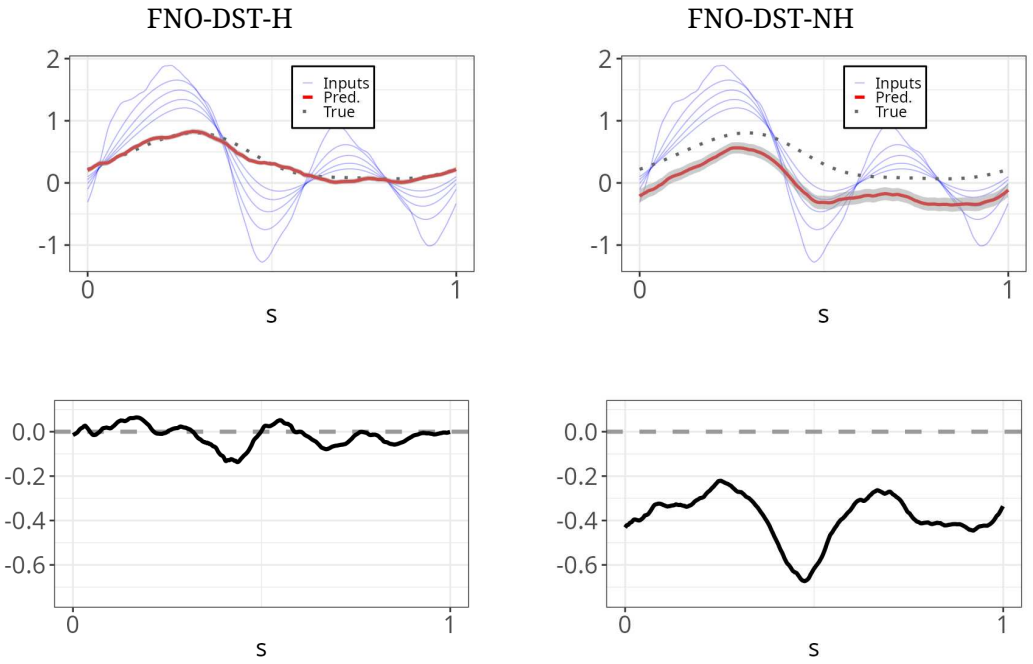
t_j）。由于我们采用周期性边界条件，因此我们 $a_0(\cdot)$ 在单位圆上进行一次模拟，模拟过程来自均值为零、协方差由弦长 Mat´ern 协方差函数（Guinness and Fuentes，2016）得到的环绕高斯过程，方差为 1。0，长度尺度为 1。0，光滑度为 2。0。我们考虑两种模拟设置：一种具有随机的 $\gamma \equiv \gamma^{(1)} \sim$Uniform$(0.05,\ 0.7)$，另一种具有固定的 $\gamma \equiv \gamma^{(2)} = 0.4$。

在这项模拟研究中，我们考虑以下形式的两种不同神经算子：

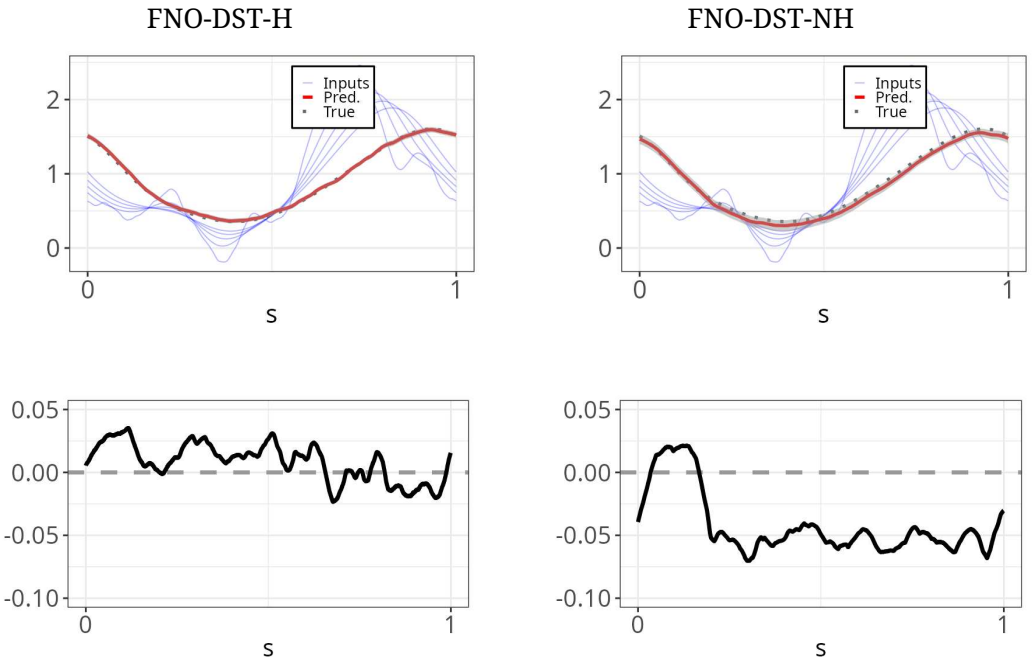
$$\mathcal{G}_{\boldsymbol{\theta}} : L^2([0,1]^2 \times [0, \tau\delta]; \mathbb{R}) \mapsto L^2([0,1]^2 \times \{(\tau + h)\delta\}; \mathbb{R}).$$

我们将第一个神经算子命名为 FNO-DST-H（带“历史”的 FNO-DST），并令其 $\tau = 4$。我们将第二个神经算子命名为 FNO-DST-NH（不带“历史”的 FNO-DST），并令其 $\tau = 0$。对于这两个算子，我们令 $\delta = 0.1$，并令 $h = 5$（提前 5 步预测）。我们考虑这两个神经算子，以验证我们在第 2.1节中的直觉：当动力学变化时（对于 $\gamma^{(1)}$的情况），需要以过去轨迹为条件，才能提升预测的准确性，尽管（31）表明其符合马尔可夫模型。

对于每种神经算子和模拟设置，我们在空间域 $\mathbf{D}^G = \{i/2^{11}$ 上模拟 1000 个独立样本：$i = 0, \ldots, 2^{11}\}$，时间步长为 $\Delta_t = 1/10000$，并使用（32）。神经算子采用更粗的时间步长 $\delta = 0.1$。此处，我们令 $\{t_1, \ldots, t_T\} = \{0.1, 0.2, \ldots, 1.0\}$，并根据截至 $t_5 = 0.5$ 的信息，在 $t_{10} = 1$ 评估预测性能。注意，$\mathbf{D}^G$ 定义了单位区间的均匀离散化，其中包含 2^{11} 个等距点；如补充材料 S.1.2 节所述，这样设置是为了便于进行高效的快速傅里叶变换计算。我们使用时间和空间离散化构造向量 $\{\mathbf{Y}_k : \ k = 1, \ldots, T\}$，以优化（27）中定义的对数似然。对于每种神经算子和模拟设置，我们使用 $N = 950$ 个样本进行训练；其余 $N_0 = 50$ 个样本用于测试。



(a) 随机 $\gamma = \gamma^{(1)}$ 对应的预测场（顶行）和误差（底行）。



(b) 固定 $\gamma = \gamma^{(2)}$ 对应的预测场（顶行）和误差（底行）。

图1：分别展示 FNO-DST-H 和 FNO-DST-NH 模型在 $T = 10$ 的预测中，$N_0 = 50$ 个测试实例中的第一个测试实例。(a)：$\gamma = \gamma^{(1)}$；(b)：$\gamma = \gamma^{(2)}$。(a) 和 (b) 的上方面板展示了当前时刻及此前四个时间步（蓝线）、$T = 10$ 的真实空间场（黑色虚线），以及带有 95% 预测区间（灰色阴影区域）的预测（红线）。

表 1：第$\gamma^{(1)}$ 节模拟实验中比较随机 $\gamma^{(2)}$ 和固定 $\gamma^{(2)}$ 情况下 FNO-DST-H 与 FNO-DST-NH 的性能指标3.1。

	模型	均方预测误差	预测区间覆盖概率	平均预测区间宽度
$\gamma = \gamma^{(1)}$	FNO-DST-NH	0.381	0.122	0.202
	FNO-DST-H	0.001	0.97	0.088
$\gamma = \gamma^{(2)}$	FNO-DST-NH	0.001	0.97	0.122
	FNO-DST-H	0.0007	0.98	0.065

3.2 模拟结果

图 1 展示了随机 $\gamma = \gamma^{(1)}$ 条件下第一个测试实例，以及固定 $\gamma = \gamma^{(2)}$条件下第一个测试实例中 FNO-DST-H 与 FNO-DST-NH 的比较。补充材料 S.2.2 节还提供了另外两个测试实例的同类比较，其模式与第一个测试实例一致。

表 1使用预测性能指标 MSPE、PICP 和 MPIW，对 FNO-DST-H 与 FNO-DST-NH 进行数值比较。这些指标是在 $t = 1$条件下，对重复试验、空间和预测时刻取平均后得到的（参见补充材料 S.2.1 节）。从比较表和图中可以看出，当 γ 由随机 $\gamma^{(1)}$给定时，训练期间输入当前及过去 $\tau = 4$ 步的数据，使 FNO-DST-H 能够更有效地学习底层控制动力学，并对空间场 $u(\cdot$生成优于 1) FNO-DST-NH 的提前5步预测。对于 γ 由固定 $\gamma^{(2)}$给定的情况，使用单个当前值进行训练，与使用当前值和过去四个值进行训练所得结果相当。这支持了我们在第2.1节提出的观点：当底层偏微分方程参数可变时，预测空间场应纳入时间历史，而不是仅依赖单个快照。

作为本节的总结，我们将 FNO-DST-H 模型与本节引言中针对模拟设置介绍的三个竞争

模型进行比较 $\gamma = \gamma^{(1)}$。比较结果以可视化形式展示于图2 中，并使用表2中的均方预测误差（MSPE）、预测区间覆盖概率（PICP）和平均预测区间宽度（MPIW）等预测性能指标进行评估。结果表明，FNO-DST-H 能够生成有效的预测（ConvLSTM 和 STDK 也同样如此），但其准确性更高，而且其 MPIW 缩小了一个数量级。

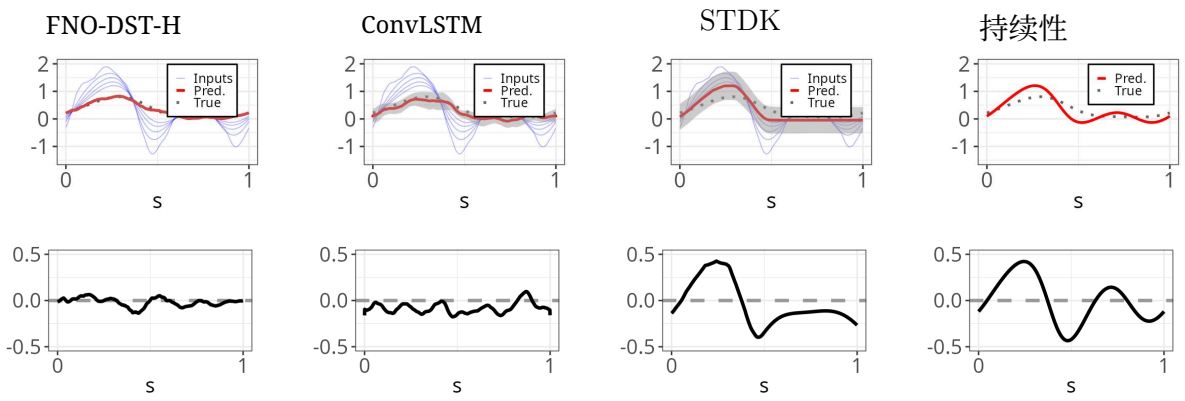


图2：第一个测试实例（共 $N_0 = 50$ 个测试实例）中，各模型的预测场（按模型排列的顶行）及相应的预测误差（按模型排列的底行）。该可视化同时展示了在随机 $\gamma = \gamma^{(1)}$的模拟设置下的预测质量 $T = 10$ 以及空间上的误差分布。在顶部面板中，展示了当前时刻和此前四个时间步（蓝线）、 $T = 10$ 处的真实空间场（黑色点线），以及带有95%预测区间（灰色阴影区域）的预测结果（红线）。

表2：模拟设置下各竞争模型的性能指标 $\gamma = \gamma^{(1)}$ 和FNO-DST-H，详见第3.1节。

模型	均方预测误差	预测区间覆盖概率	平均预测区间宽度
FNO-DST-H	0.001	0.97	0.088
ConvLSTM	0.054	0.98	0.574
STDK	0.075	0.99	1.013
持续性	0.089	-	-

4 FNO-DST 在地球物理应用中的预测

本节对 FNO-DST 应用于两个地球物理数据集的预测进行比较分析。我们还将这些预测与另一种动力预测方法进行比较，该方法基于第 2.1 节讨论的、受物理原理启发的积分差分方程。线性积分差分方程的神经网络版本（CNN-IDE；Zammit-Mangion 和 Wikle，2020年）在预期存在此类动力学的情况下（如海表温度预测）具有竞争力，但在对底层动力学知之甚少时，表现则较差。

4.1 海表温度数据的应用

本研究使用由 NEMO（海洋欧洲建模核心）模拟框架生成的海表温度（SST）数据（Madec 和 NEMO系统团队，2024年）。NEMO 是一个先进的建模系统，旨在进行海洋模拟，采用受物理原理启发的方程对区域和全球海洋环流进行建模。与气候再分析相似，该系统还利用历史观测生成海洋状态的同化表示，确保其数据与实际海表温度测量数据高度吻合。

我们使用 NEMO 海表温度数据，该数据取自 GLOBAL ANALYSIS FORECAST PHY 001 _ _ _ _ _024产品，由哥白尼海洋环境监测服务 (CMEMS) 提供。我们将该来源的数据汇总到 0.5 度的经纬度网格上。按照Zammit-Mangion 和 Wikle (2020年) 的做法，我们将海表温度的分析与预测集中在北大西洋。该区域划分为 19 个不同的区域，每个区域覆盖一个标准化的 64×64 网格 $\mathcal{D}^G \equiv \left\{ \left(\frac{i-1}{63}, \frac{j-1}{63} \right) : i, j \in \{1, \ldots, 64\} \right\}$。

— —

在本研究中，我们重点利用FNO-DST框架对算子进行建模

第2.2 中定义的映射

$$\mathcal{G}_{\boldsymbol{\theta}} : L^2([0,1]^2 \times [0, \tau\delta]; \mathbb{R}) \mapsto L^2([0,1]^2 \times \{(\tau + h)\delta\}; \mathbb{R}).$$

为了训练我们的模型，我们使用该数据产品中可获得的最初4000天数据，涵盖2006年12月27日至2017年12月16日期间，并将19个区域视为重复样本。最后10天的数据留作测试。因此，共有$4000 \times 19 = 76000$ 张图像用于训练。图3 （参考Zammit-Mangion和Wikle的图表，2020年）展示了这19个区域的划分，

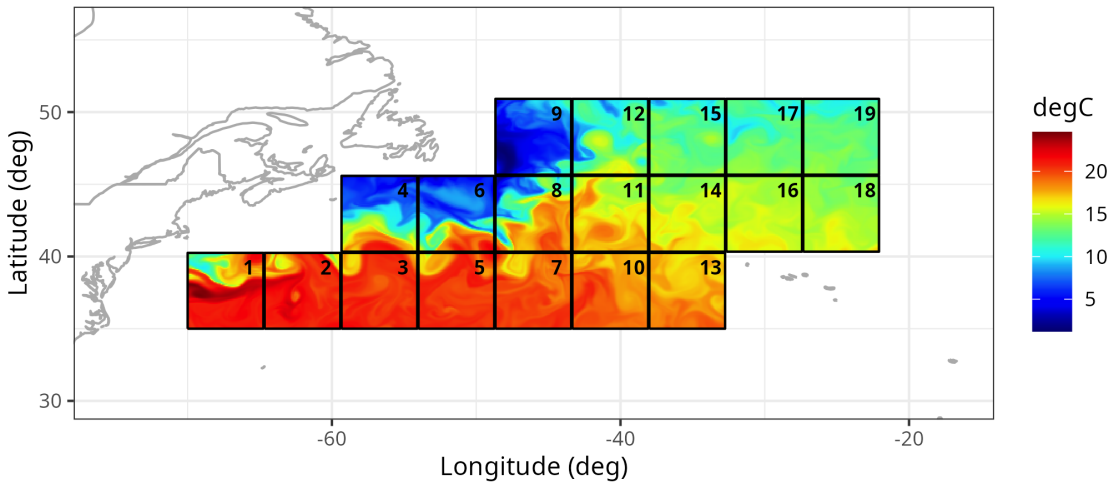


图3：2006年12月27日的北大西洋海表温度（海表温度）NEMO数据，其中19个区域以数字编号，并用黑色方框划定。

以2006年12月27日的海表温度数据为例（这是该数据产品中的第一天）。将数据划分为19个区域，主要是为了提高计算效率；不过，这并不会显著影响该方法，因为每个区域内的主导动力学仍可用于根据近期过去预测未来。我们取 $\tau = 2$ （近期过去）、 $h = 3$ （提前3步的预测），并将时间维度离散化为日分辨率。因此，我们根据当天及之前两天的数据，向前预测3天。

图4 对比了 FNO-DST模型和 CNN-IDE模型提前3天的预测。众所周知，海表温度数据集具有平流-扩散

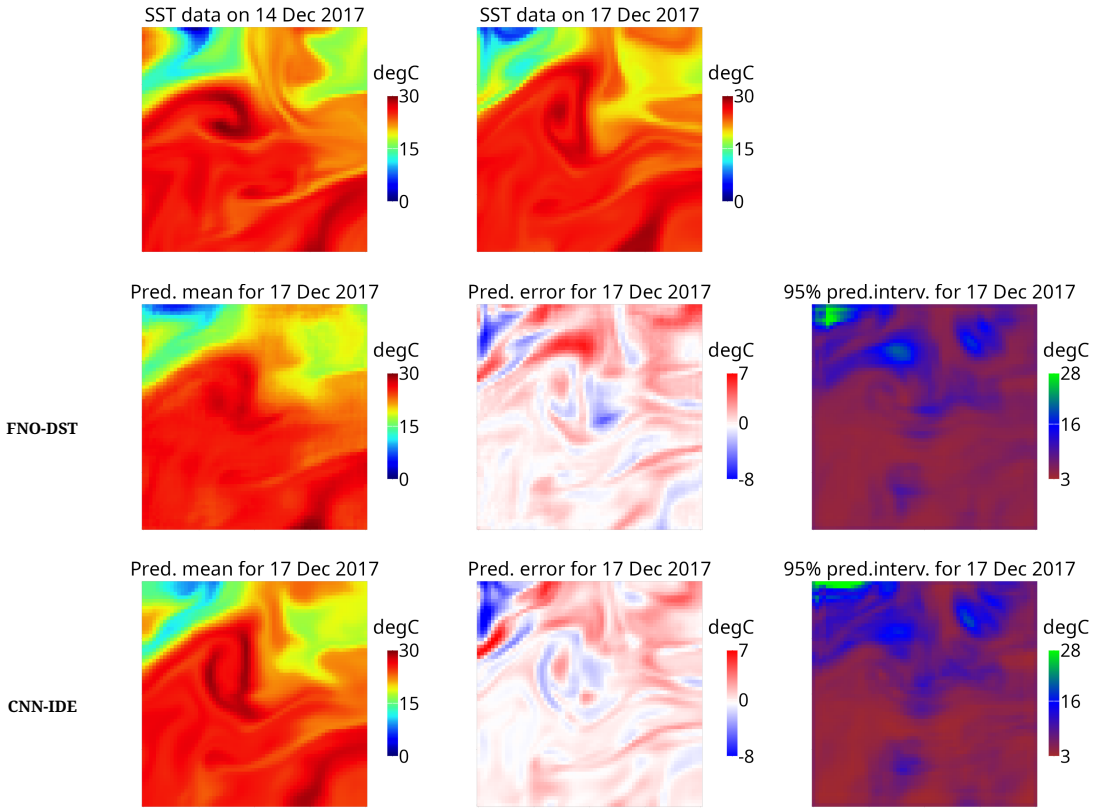


图4：北大西洋区域3中 FNO-DST 与 CNN-IDE 预测的可视化比较。基于2017年12月12日、13日和14日的数据，对2017年12月17日的海表温度进行了预测。顶行显示2017年12月14日的数据及其随后3天演变至2017年12月17日的情况。中间行（FNO-DST）和底行（CNN-IDE）显示2017年12月17日的预测结果（预测均值、预测误差和逐像素95%预测区间宽度）。

特性，CNN-IDE模型通过先验物理知识有效地将其纳入模型。表 3 采用了表 1 和 2（第3节）中使用的相同指标，展示了更全面的比较，其中还包括模拟中比较过的另外三种预测方法（ConvLSTM、STDK 和持续性）。尽管 CNN-IDE对海表温度的预测表现最佳，FNO-DST的预测仍然表现良好，即使模型未明确纳入地球物理知识。这表明，该模型在动力学规律知之甚少的应用场景中具有预测潜力，例如下一小节中的日降水数据。

表3：第2.1节和3节所述竞争模型的海表温度预测性能指标，取19个区域以及2017年12月17日至2017年12月26日期间所有提前3天的预测的平均值。（FNO-DST模型包含从当前时间到前两个时间点的“历史”数据。）

模型	均方预测误差	预测区间覆盖概率	平均预测区间宽度
FNO-DST	1.56	0.95	5.77
CNN-IDE	1.34	0.95	5.65
ConvLSTM	1.66	0.96	5.85
STDK	2.27	0.93	3.98
持续性	2.45	-	-

4.2 降水数据的应用

第二个应用使用从西欧和中欧各地气象站收集的日降水数据（Toreti，2014）。该数据集包含全面的日降水观测记录，时间跨度为2014年1月1日至2024年12月31日。图5 展示了气象站的空间分布，其中红色标出的子区域被选作本研究的预测区域。为预处理数据，我们对每个气象站的日数据应用10日移动平均，以减轻短期波动并填补数据中的小段缺失。随后，对每个时间序列进行标准化，以确保不同空间位置和时间点的尺度一致，并保存样本均值和样本标准差，以便将预测结果还原至原始尺度。最后，采用Nag 等人 (2023) 的 Deep-Kriging 方法进行时空插值，在64×64 标准化网格 $\mathcal{D}^G \equiv \left\{\left(\frac{i-1}{63}, \frac{j-1}{63}\right) : i, j \in \{1, \ldots\ldots, 64\}\right\}$ 上生成降水场，该网格覆盖图5中标出的子区域，即 $[15^\circ\text{E}\ 25^\circ\text{E}] \times [45^\circ\text{N}\ 55^\circ\text{N}]$，适用于*2014*年*1*月*1*日至*2024*年*12*月*31*日（含首尾日期）的所有日期。有关数据集、其预处理以及

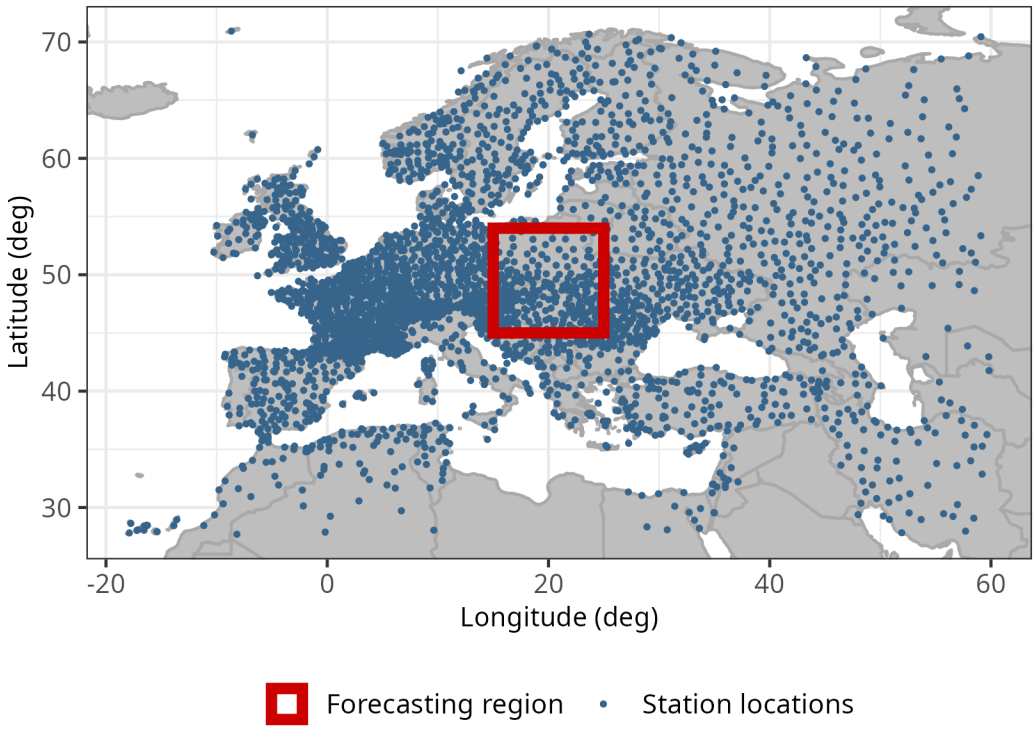


图5：西欧和中欧气象站位置的分布。红色标出的区域是专门为预测而选定的。

插值框架的更多详细信息见Nag (2025)。为训练模型，我们使用网格化空间数据集的前3200天，时间跨度为2014年1月1日至2023年11月5日。剩余的421天，即2023年11月6日至2024年12月31日，留作测试。与第4.1节一样，我们取 $\tau = 2$ (近期过去)， $h = 3$ (提前3步预测)。请注意，与有19个重复样本的海表温度应用不同，此处只有一个重复样本，因此训练使用3200幅图像。

图6对 FNO-DST模型和 CNN-IDE模型提前3天的预测进行了可视化比较。表4采用了与表3相同的指标和预测方法。结果清楚地表明，在所有评估指标上，FNO-DST对降水的预测均优于其他模型的预测。对于降水应用，相关动力学的理解相对有限，CNN-IDE的性能排名第三。如图6所示，FNO-DST模型能够预测出东西向

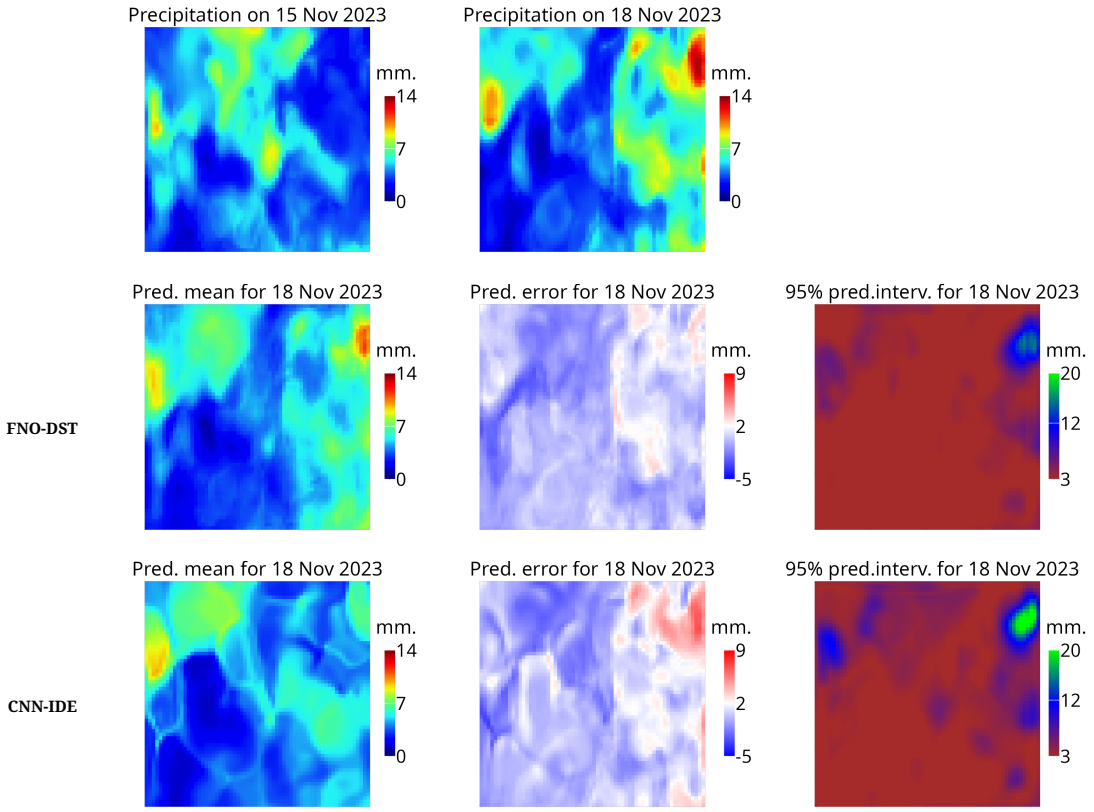


图6：对图5所示高亮区域内降水数据的 FNO-DST 和 CNN-IDE 预测进行可视化比较。2023年11月18日的预测基于2023年11月13日、2023年11月14日和2023年11月15日的数据生成。顶行展示2023年11月15日的数据及其未来3天至2023年11月18日的演变。中间行（FNO-DST）和底行（CNN-IDE）展示预测结果（预测均值、预测误差和逐像素95%预测区间宽度）。

降水锋面，与2023年11月18日实际出现的情况相似。另一方面，CNN- IDE模型未能再现这一特征。第3节中的模拟结果以及本节的应用凸显了FNO-DST预测在不同场景下的推断能力和计算效率。

表4：第3节和第4节所述不同竞争模型的性能指标，取2023年11月6日至2024年12月31日期间所有空间位置及所有日期的提前3步预测的平均值。

模型	均方预测误差	预测区间覆盖概率	平均预测区间宽度
FNO-DST	0.32	0.96	1.95
CNN-IDE	0.46	0.96	2.43
ConvLSTM	0.38	0.94	2.23
STDK	0.82	0.92	1.97
持续性	0.84	-	-

5 结论

在本研究中，我们提出了一种动态时空（DST）建模框架，利用傅里叶神经算子（FNO）以高效的计算方式预测完整的空间场。离散化的偏微分方程模型和积分微分方程模型往往难以处理许多环境系统和物理系统中固有的非线性相互作用与复杂动力学，这会影响其预测性能。我们的方法利用傅里叶神经算子通过快速、可并行化的计算，学习并近似空间场随时间变化的映射，而无需明确了解底层动力机制。

我们通过模拟研究和现实世界应用相结合的方式，评估了第2.2节中提出的FNO-DST模型的预测性能。在基于Burgers方程（非线性偏微分方程）的仿真实验中，FNO-DST模型展现出出色的预测能力，能够捕捉支配该系统的非线性动力学。在真实数据应用中，我们将该模型应用于两种不同的时空地球物理场景：预测北部海域的海表温度（SST）

大西洋海表温度，以及预测欧洲某一地区的降水。在海表温度预测应用中，尽管此前已开发出基于偏微分方程的物理模型，FNO-DST模型的表现几乎与基于偏微分方程的模型不相上下。在降水预测应用中，由于底层动力学要复杂得多且尚未得到充分表征，FNO-DST模型仍能学习这些动力学，并根据多项性能指标提供更优的预测结果。我们的研究结果表明，基于FNO的框架能够适应各种时空系统，即使其动力学未知或知之甚少。

除了预测能力之外，基于FNO的框架的一项关键优势是其计算效率和可扩展性，无论是在训练期间还是训练之后都是如此。这为实时预测应用带来了显著的计算优势，尤其是在高维域中；在这些领域，传统统计模型和机理模型可能会变得计算成本过高，或者传统模型可能为了提高计算效率而过度简化。同样重要的是，通过将FNO嵌入统计模型中，该框架能够自然地实现不确定性量化，而这对于决策至关重要。

未来研究仍有几个颇具前景的方向。首先，将物理约束或守恒定律直接纳入傅里叶神经算子框架，将进一步提升预测能力，同时增强模型可解释性。其次，构建FNO-DST模型的贝叶斯层次统计表述，将参数不确定性纳入考量，从而改进不确定性量化。最后，尽管当前工作聚焦于空间格点，但将该方法论应用于不规则空间域、非欧几里得空间或多变量时空过程，是拓展FNO-DST方法预测适用范围的重要方向。

致谢

本研究得到美国空军科学研究办公室FA2386-23-1-4100号项目的支持。Nag和Zammit-Mangion还感谢JRC（意大利）提供的差旅支持，使其得以访问该研究中心；此次访问促成了第4.2节所 प्रस्तुत的应用案例研究。

附录：符号说明

D： 连续空间域。

T： 连续时间域。

s： 空间坐标，$\mathbf{s} \in \mathrm{D}$。

t**：** 时间坐标，$t \in \mathrm{T}$。

$\mathrm{C}(\cdot,\ \cdot)$**：** 基于偏微分方程的确定性实值连续时间空间过程。$\boldsymbol{\gamma}$**：** 与底层偏微分方程相关的参数。$g(\cdot,\ \cdot)$，$g_j(\cdot,\ \cdot)$**：** 不同情形下的格林函数和核。$\eta(\cdot,\ \cdot)$**：** 空间相关随机创新项。$Y(\cdot,\ \cdot)$**：** 所建模的时空随机过程。$Y_k(\cdot) \equiv Y(\cdot,\ t_k)$**：** 在时间点 t_k 处评估的随机（空间）过程。

$\{Y_k(\cdot)\}_k \equiv \{Y(\mathbf{s}\ t_k) :\ \mathbf{s}\ \in \mathrm{D}\}_{k=1}^{T}$**：** 定义在域 D 上的空间过程时间序列。

D^G**：** D 的基本分析单元离散化。

$\boldsymbol{\eta}_{k+h}$：　$\eta(\cdot,\ t_{k+h})$ 在 D^G 上评估。

$\boldsymbol{\Sigma}_{k+h}(\mathbf{Y}_k;\ \boldsymbol{\alpha})$：与 $\boldsymbol{\eta}_{k+h}$ 相关的协方差矩阵，它是 $\mathbf{Y}_k$ 的函数，并由 $\boldsymbol{\alpha}$ 参数化。

$\mathbf{Y}_k$：　$Y_k(\cdot)$在 D^G 上的取值。

$\mathrm{G}_{\boldsymbol{\gamma}}$、$\mathrm{G}_{j,\ \boldsymbol{\theta}'}$、$\mathrm{G}_{\boldsymbol{\theta}''}$：分别表示式（3）、（8）和（10）中算子的 BAU离散化格林函数矩阵，其参数分别为 $\boldsymbol{\gamma}$、$\boldsymbol{\theta}'$ 和 $\boldsymbol{\theta}''$。

τ：用于构建条件历史的过去滞后。

h：用于 h 步超前预测的预测时域。

$\mathbf{x} \equiv (\mathbf{s}^\top,\ t)^\top$：时空坐标，$\mathbf{x} \in \mathrm{D} \times \mathrm{T}$。

$\mathrm{G}_{\boldsymbol{\theta}}$：由 $\boldsymbol{\theta}$参数化的 FNO算子。

$\mathcal{P},\ \mathcal{K},\ \mathcal{W},\ \mathcal{Q}$：分别如式（14）、（16）、（15）和（18）所定义的提升算子、核积分算子、局部线性算子和投影算子。

$\mathcal{F},\ \mathcal{F}^{-1}$：分别如式（20）和（21）所定义的傅里叶算子和逆傅里叶算子。

$\mathbf{G}(\cdot;\ \boldsymbol{\theta})$：表示算子 $\mathrm{G}_{\boldsymbol{\theta}}$ 的 BAU离散化输出的向量。

$\mathbf{Y}^{(k-\tau):k}$：向量值随机时间序列 $\{\mathbf{Y}_{k-\tau}$、……、$\mathbf{Y}_k\}$。

$\{\mathbf{Y}_k^{(n)}\}_k$：表示第 i 个独立重复样本的 BAU离散化时间序列的向量。

$\boldsymbol{\Sigma}_{k+h}^{(n)}(\mathbf{Y}_k^{(n)};\ \boldsymbol{\alpha})$：与$\mathbf{Y}_{k+h}^{(n)}$相关的协方差矩阵。

$\mathrm{T}_\tau^G \equiv \{0,\ \delta,\ \ldots\ldots,\ \tau\delta\}$：将区间 $[0,\ \tau\delta]$时间离散化为 $\tau+1$ 个等间距点，步长为 δ。

$L(\boldsymbol{\theta},\ \boldsymbol{\alpha})$：参数 $\boldsymbol{\theta}$ 和 $\boldsymbol{\alpha}$ 的似然函数，其中 $\boldsymbol{\alpha}$ 是描述创新空间场空间依赖的参数向量。

参考文献

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). 空间数据的层次建模与分析. Chapman and Hall, CRC Press, Boca Raton, 佛罗里达州。

Biswas, B. N., Chatterjee, S., Mukherjee, S., and Pal, S. (2013). 关于欧拉方法的讨论：综述。*Electronic Journal of Mathematical Analysis and Applications*, 1(2):2090–2792。

Brauer, F. (1967年). 奇异常微分算子的格林函数。*CanadianJournal of Mathematics*, 19:571–582。

Brezis, H. and Br´ezis, H. (2011年). 泛函分析、*Sobolev*空间与偏微分方程. 第2卷，施普林格，纽约。参见第8章。

Calder, C. A., Berrett, C., Shi, T., Xiao, N., and Munroe, D. K. (2011年). 使用卫星数据和大气输送模型输出对气溶胶的时空动力学进行建模。*Journal of Agricultural, Biological, and Environmental Statistics*, 16:495–512。

Cooley, J. 和 Tukey, J.（1965）。用于机器计算复傅里叶级数的算法。*Mathematics of Computation*，19(90):297–301。

Cressie, N. 和 Huang, H.-C.（1999年）。不可分离的时空平稳协方差函数类别。*Journal of the American Statistical Association*，94(448):1330–1340。

Cressie, N. 和 Wikle, C. K.（2011年）。时空数据统计学。Wiley，Hoboken，NJ。

de B´ezenac, E.、Pajot, A. 和 Gallinari, P.（2019年）。面向物理过程的深度学习：融入先验科学知识。《统计力学杂志：理论与实验》，2019年(12)：124009。

Evans，L. C.（2022年）。偏微分方程，第*5*章。第19卷，美国数学会，普罗维登斯。

Gneiting，T. 和 Guttorp，P.（2007年）。连续参数时空模型。环境计量学，18：412–429。

Guinness，J. 和 Fuentes，M.（2016年）。球面上的各向同性协方差函数：一些性质和建模考量。《多元分析杂志》，143：143–152。

Hopf，E.（1950年）。偏微分方程 $u_t + uu_x = \mu u_{xx}$。《纯粹与应用数学通讯》，3（3）：201–230。

Katzfuss，M. 和 Guinness，J.（2021年）。高斯过程 Vecchia 近似的一般框架。《统计科学》，36（1）：124–141。

Kingma，D. P.（2014年）。Adam：一种随机优化方法。*arXiv* 预印本 *arXiv:1412.6980*。

Kot, M. 和 Schaffer, W. M.（1986）。离散时间生长扩散模型。《数学生物科学》，80：109–136。

Kovachki, N.、Li, Z.、Liu, B.、Azizzadenesheli, K.、Bhattacharya, K.、Stuart, A. 和 Anand- kumar, A.（2023 年）。神经算子：学习函数空间之间的映射及其在偏微分方程中的应用。《机器学习研究杂志》，24(89)：1–97。

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). 简单且可扩展的预测

不确定性估计方法：使用深度集成。神经信息处理系统进展，30:1–12。

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Stuart, A., Bhattacharya, K. 和 Anand- kumar, A.（2020年）。用于参数偏微分方程的多极图神经算子。神经信息处理系统进展，33:6755–6766。

Madec, G. 和 NEMO系统团队（2024年）。NEMO海洋引擎参考手册。https://doi.org/10.5281/zenodo.1464816。

Nag, P.（2025年）。PyTorch 中的时空深度克里金法：降水数据插值与概率预报的补充应用。*arXiv* 预印本 *arXiv:2509.12708*。

Nag, P., Sun, Y. 和 Reich, B. J.（2023年）。用于插值与概率预报的时空深度克里金法。空间统计学，57:100773。

Neubert, M. G., Kot, M., and Lewis, M. A.（1995）。离散时间捕食者-猎物模型中的扩散与图案形成。理论种群生物学，48(1):7–43。

Pan, E.（2009）。格林函数。第13–46页，施普林格，纽约。

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C.（2015）。卷积长短期记忆网络：一种用于降水临近预报的机器学习方法。神经信息处理系统进展，28:1–9。

Toreti, A.（2014）。欧洲网格化农业气象数据。 http://data.europa.eu/89h/jrc-marsop4-7-weather_obs_grid_2019。下载于2025年1月24日。

Wikle, C. K.（2002年）。用于非高斯时空过程的基于核的谱模型。统计建模，2(4):299–314。

Wikle, C. K. 和 Cressie, N.（1999年）。时空卡尔曼滤波的降维方法。*Biometrika*, 86(4)：815–829。

Wikle, C. K. 和 Holan, S. H.（2011年）。多项式非线性时空积分-差分方程模型。时间序列分析期刊，32(4)：339–350。

Wikle, C. K. 和 Hooten, M. B.（2010 年）。用于时空动力学模型的通用基于科学的框架。*Test*，19：417–451。

Wikle, C. K.、Zammit-Mangion, A. 和 Cressie, N.（2019年）。《*R*时空统计学》。CRC Press，Boca Raton，佛罗里达州。

Xu, K.、Wikle, C. K. 和 Fox, N. I.（2005年）。用于天气雷达反射率临近预报的基于核的时空动力学模型。*Journal of the American StatisticalAssociation*，100(472)：1133–1144。

Zammit-Mangion, A. 和 Wikle, C. K.（2020年）。用于时空预测的深度积分差分方程模型。空间统计学，37:100408。

补充材料：使用傅里叶神经算子进行时空建模与预测

Pratik Nag[1], Andrew Zammit-Mangion[1], Sumeetpal Singh[1], 和 Noel Cressie[1]

[1]伍伦贡大学数学与应用统计学院，澳大利亚

此处提供有关方法论（格林函数）、离散化、计算（快速傅里叶变换）以及仿真实验（性能指标、额外图表）的更多细节。

S.1 更多技术细节

S.1.1 正文中公式（2）的格林函数推导

设 $\mathcal{C}(\mathbf{s}, t)$ 的傅里叶变换为 $\mathcal{F}[\mathcal{C}](\cdot,\cdot)$，其中

$$\mathcal{F}[\mathcal{C}](\boldsymbol{\kappa}, t) \equiv \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t) = \int_{\mathbb{R}^d} e^{-\iota \boldsymbol{\kappa}^\top \mathbf{s}} \mathcal{C}(\mathbf{s}, t)\, \mathrm{d}\mathbf{s},$$

及其逆

$$\mathcal{F}^{-1}[\widehat{\mathcal{C}}](\mathbf{s}, t) \equiv \mathcal{C}(\mathbf{s}, t) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{\iota \boldsymbol{\kappa}^\top \mathbf{s}} \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t)\, \mathrm{d}\boldsymbol{\kappa},$$

其中，ι 是复平面中的虚数，且 $\boldsymbol{\kappa} = (\kappa_1, \ldots\ldots, \kappa_d)^\top$。对正文中的公式 (1) 进行傅里叶变换，并利用线性性，

$$\frac{\partial}{\partial t}\mathcal{F}[\mathcal{C}](\boldsymbol{\kappa}, t) + \gamma_1 \sum_{i=1}^{d} \mathcal{F}\Big[\frac{\partial \mathcal{C}}{\partial s_i}\Big] - \gamma_2 \sum_{i=1}^{d} \mathcal{F}\Big[\frac{\partial^2 \mathcal{C}}{\partial s_i^2}\Big] = 0, \quad \text{for } (\boldsymbol{\kappa}, t) \in \mathbb{R}^d \times [t_k, t_k + \tau]. \quad \text{(S.1.1)}$$

对于每个 $i = 1, \ldots\ldots, d$,

$$\mathcal{F}\Big[\frac{\partial \mathcal{C}}{\partial s_i}\Big](\boldsymbol{\kappa}, t) = \int_{\mathbb{R}^d} e^{-\iota \boldsymbol{\kappa}^\top \mathbf{s}} \frac{\partial \mathcal{C}}{\partial s_i}(\mathbf{s}, t)\, \mathrm{d}\mathbf{s}.$$

对 s_i 进行分部积分可得,

$$\mathcal{F}\Big[\frac{\partial \mathcal{C}}{\partial s_i}\Big](\boldsymbol{\kappa}, t) = \iota \kappa_i \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t), \qquad \mathcal{F}\Big[\frac{\partial^2 \mathcal{C}}{\partial s_i^2}\Big](\boldsymbol{\kappa}, t) = -\kappa_i^2 \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t).$$

请注意，分部积分步骤利用了边界项在 $|s_i| \to \infty$ 时趋于零这一事实。在温和假设下，这一点成立：对于每个固定的 t，$C(\cdot\ t)$ 和 $(\partial C/\partial s_i)(\cdot\ t)$ 属于 $L^1(\mathbb{R}^d;\ \mathbb{R})$，这尤其意味着当 $\|s$ $\| \to \infty$ 时，$C(s\ t) \to 0$（Brezis 和 Br´ezis，2011年；Evans，2022年）。

将这些表达式代入方程（S.1.1），得到

$$\frac{\partial}{\partial t}\widehat{\mathcal{C}}(\boldsymbol{\kappa}, t) + \gamma_1 \sum_{i=1}^{d} \iota \kappa_i \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t) + \gamma_2 \sum_{i=1}^{d} \kappa_i^2 \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t) = 0,$$

或者等价地，

$$\frac{\partial \widehat{\mathcal{C}}}{\partial t}(\boldsymbol{\kappa}, t) = -\left(\iota \gamma_1 \mathbf{1}^\top \boldsymbol{\kappa} + \gamma_2 \|\boldsymbol{\kappa}\|^2\right) \widehat{\mathcal{C}}(\boldsymbol{\kappa}, t),$$

其中 $\mathbf{1}$ d 1 $du/dt =$ 是由 s 构成的 -维向量。$-au$ $u\ t \equiv \widehat{\boldsymbol{\kappa}}\ t$ 我们将上面的方程简记为，其中 ( ) C(,)，且 $a \equiv \iota\gamma_1 \mathbf{1}^\top \boldsymbol{\kappa} + \gamma_2 \|\boldsymbol{\kappa}\|^2$，对于固定的 。于是，

$$u(t) \propto e^{-at}.$$

在起始时 ^ ^值 $t = t_k$ 位于区间 $[t_k, t_k + \tau]$中时，我们写作 $C(\boldsymbol{\kappa}, t_k) \equiv C_0(\boldsymbol{\kappa})$。因此

$$u(t_k + \tau) = e^{-a\tau}\, \widehat{\mathcal{C}}_0(\boldsymbol{\kappa}),$$

^或者用 $C(\boldsymbol{\kappa}, \cdot)$ 表示，

$$\widehat{\mathcal{C}}(\boldsymbol{\kappa}, t_k + \tau) = \exp\Big\{ -(\iota\gamma_1 \mathbf{1}^\top \boldsymbol{\kappa} + \gamma_2 \|\boldsymbol{\kappa}\|^2)\tau \Big\}\, \widehat{\mathcal{C}}_0(\boldsymbol{\kappa}).$$

其傅里叶逆变换为，

$$\mathcal{C}(\mathbf{s}, t_k + \tau) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{\iota \boldsymbol{\kappa}^\top \mathbf{s}}\, \widehat{\mathcal{C}}_0(\boldsymbol{\kappa})\, \exp\Big\{ -(\iota\gamma_1 \mathbf{1}^\top \boldsymbol{\kappa} + \gamma_2 \|\boldsymbol{\kappa}\|^2)\tau \Big\}\, \mathrm{d}\boldsymbol{\kappa}.$$

利用 $\widehat{C}_0(\boldsymbol{\kappa}) = \int_{\mathbb{R}^d} e^{-\iota \boldsymbol{\kappa}^\top \mathbf{u}} C(\mathbf{u}\ t_k)\, \mathbf{du}$，并应用富比尼定理，可得

$$\mathcal{C}(\mathbf{s}, t_k + \tau) = \int_{\mathbb{R}^d} \left[\frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \exp\Big\{ \iota \boldsymbol{\kappa}^\top (\mathbf{s} - \mathbf{u}) - \iota\gamma_1 \tau \mathbf{1}^\top \boldsymbol{\kappa} - \gamma_2 \tau \|\boldsymbol{\kappa}\|^2 \Big\}\, \mathrm{d}\boldsymbol{\kappa} \right] \mathcal{C}(\mathbf{u}, t_k)\, \mathrm{d}\mathbf{u}.$$

因此，对于 $\boldsymbol{\gamma} = (\gamma_1, \gamma_2)^\top$，其中 $\gamma_2 > 0$，格林函数为

$$g(\mathbf{r}, \tau; \boldsymbol{\gamma}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} \exp\Big\{ \iota \boldsymbol{\kappa}^\top \mathbf{r} - \iota\gamma_1 \tau \mathbf{1}^\top \boldsymbol{\kappa} - \gamma_2 \tau \|\boldsymbol{\kappa}\|^2 \Big\} \mathrm{d}\boldsymbol{\kappa},$$

其为实 值函数，并可通过标准高斯积分计算为 :

$$g(\mathbf{r}, \tau; \boldsymbol{\gamma}) = \frac{1}{(4\pi\gamma_2\tau)^{d/2}} \exp\Big(-\frac{\|\mathbf{r} - \tau\gamma_1 \mathbf{1}\|^2}{4\gamma_2 \tau} \Big), \quad \tau > 0.$$

最后，将 $\tau = h\delta$ 代入即可得到正文中公式（2）的所需结果。

S.1.2 关于傅里叶神经算子输入离散化的更多细节

按照正文第2.2节中的记号，在 $\mathbf{D}^G \times \mathbf{T}^G$ 上对 $k \in \{\tau+1, \ldots\ldots, T-h\}$ 求值时，输入场 $Y_0^{(k)}(\cdot)$（其中 $\mathbf{T}^G \equiv \{0, \delta, \ldots\ldots, \tau\delta\}$）是一个大小为 $m_1 \times \cdots \times m_{d+1}$。类似地，在 $\mathbf{D}^G \times \mathbf{T}^G$ 上求值的第 l 层的输出 $\mathbf{v}_l(\cdot)$ 是一个大小为 $d_v \times m_1 \times \cdots \times m_{d+1}$ 的张量，其中 $l = 1, \ldots, L$。这种张量表示方式有利于在图形处理单元和张量处理单元上进行高效计算。具体而言，局部操作 $\mathbf{P}[\cdot], \mathbf{Q}[\cdot]$, 和 $\mathbf{W}_l[\cdot]$,（其中 $l = 1, \ldots, L$）可简化为多次批量乘法和加法，并且能够针对 $\mathbf{D}^G \times \mathbf{T}^G$ 中的每个点以易并行的方式执行。

S.1.3 离散傅里叶变换的计算

正文第2.2节中的张量表示方式使我们能够利用离散傅里叶变换（DFT）。方程（21）和（22）的计算复杂度为$\mathcal{O}((\prod_{j=1}^{d+1} \tilde{m}_j)^2)$，其中 $\tilde{m}_j$ 在正文方程（22）下方定义。为加快计算，实际应用中我们使用库利–图基快速傅里叶变换（FFT）算法（Cooley 和 Tukey, 1965），令 $\tilde{m}_j = 2^{q_j}$, $j = 1, \ldots, d+1$，并递归地计算偶数和奇数索引处的傅里叶变换。快速傅里叶变换的计算复杂度为 $\mathcal{O}((\prod_{j=1}^{d+1} \tilde{m}_j)\log(\prod_{j=1}^{d+1} \tilde{m}_j))$，比朴素 DFT 快得多。同样，库利–图基快速傅里叶变换算法也可用于计算方程（20）中定义的逆离散傅里叶变换（IDFT），其速度远快于朴素 IDFT。我们使用 `pytorch`软件包为应用实现多维快速傅里叶变换。

S.2 进一步的模拟实验细节

S.2.1 性能指标

我们使用在 N_0 个测试实例上评估的指标来衡量每个模型的预测性能，即从训练中排除的空间数据时间序列。我们通过将所有模型的预测与相应空间位置处的真实过程值进行比较，计算这些指标。我们考察的诊断指标包括均方预测误差（MSPE）、名义覆盖率为 95% 的预测区间的预测区间覆盖概率（PICP），以及平均预测区间宽度（MPIW）。这些指标定义如下：

$$\mathrm{MSPE} = \frac{1}{N_0}\frac{1}{|\mathcal{D}^G|}\frac{1}{T-h-\tau}\sum_{j=1}^{N_0}\sum_{\mathbf{s}\in\mathcal{D}^G}\sum_{k=\tau+1}^{T-h}(Y^{(j)}(\mathbf{s}, t_{k+h}) - \widehat{Y}^{(j)}(\mathbf{s}, t_{k+h}))^2,$$

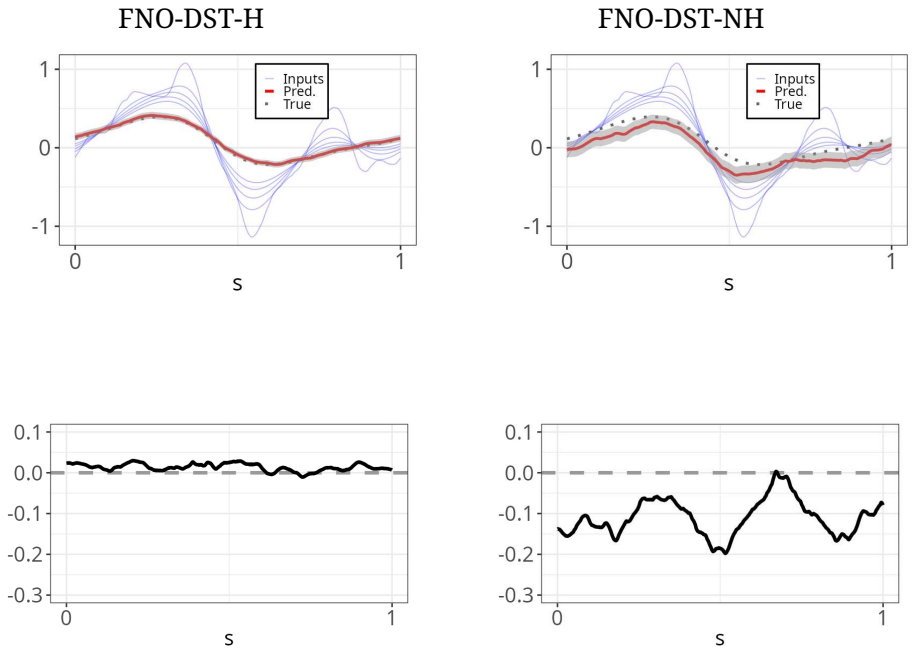
$$\mathrm{PICP} = \frac{1}{N_0}\frac{1}{|\mathcal{D}^G|}\frac{1}{T-h-\tau}\sum_{j=1}^{N_0}\sum_{\mathbf{s}\in\mathcal{D}^G}\sum_{k=\tau+1}^{T-h}\mathbb{1}\{Y^{(j)}(\mathbf{s}, t_{k+h}) \in [L^{(j)}(\mathbf{s}, t_{k+h}), U^{(j)}(\mathbf{s}, t_{k+h})]\},$$

$$\mathrm{MPIW} = \frac{1}{N_0}\frac{1}{|\mathcal{D}^G|}\frac{1}{T-h-\tau}\sum_{j=1}^{N_0}\sum_{\mathbf{s}\in\mathcal{D}^G}\sum_{k=\tau+1}^{T-h}[U^{(j)}(\mathbf{s}, t_{k+h}) - L^{(j)}(\mathbf{s}, t_{k+h})],$$

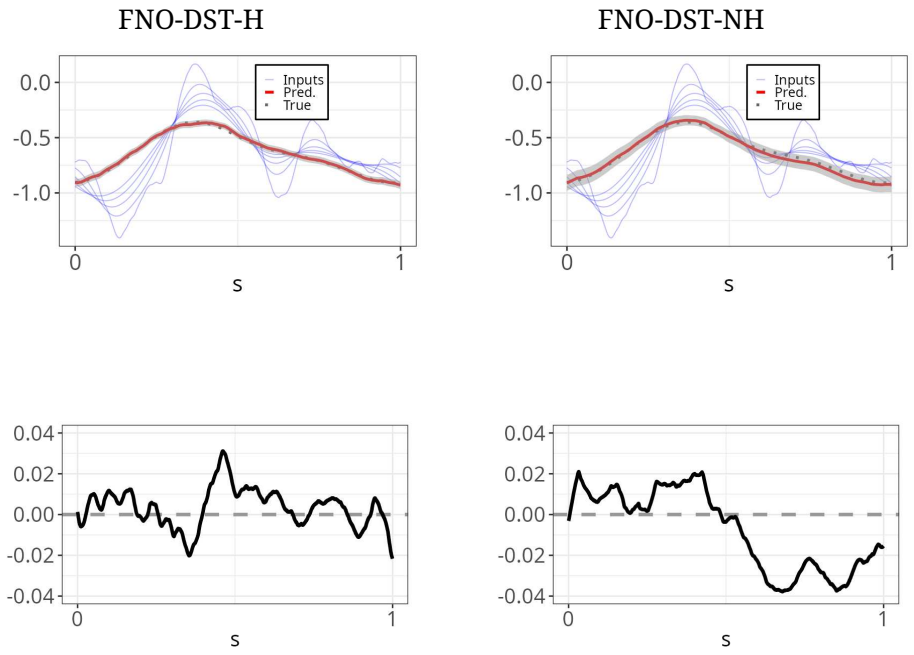
其中，$L^{(j)}(\mathbf{s}\ t_{k+h})$ 和 $U^{(j)}(\mathbf{s}\ t_{k+h})$ 分别是 $Y_{(j)}(\mathbf{s}\ t+h)$ 的 95% 预测区间的下限和上限。

S.2.2 第3节的其他测试实例

本节展示了与正文第3节对应的补充图，说明了另外两个测试实例的结果。



（a）随机 $\gamma = \gamma^{(1)}$的预测场（顶行）和误差（底行）。



（b）固定 $\gamma = \gamma^{(2)}$的预测场（顶行）和误差（底行）。

图 S.1：分别展示 FNO-DST-H 和 FNO-DST-NH 模型在 $T = 10$ 对 $N_0 = 50$ 个测试实例进行预测时的第二个测试实例。(a)：$\gamma = \gamma^{(1)}$，(b)：$\gamma = \gamma^{(2)}$。(a) 和 (b) 的上方图中展示了当前时刻和此前四个时刻（蓝线）、$T = 10$ 处的真实空间场（黑色点线），以及带有 95% 预测区间（灰色阴影区域）的预测结果（红线）。

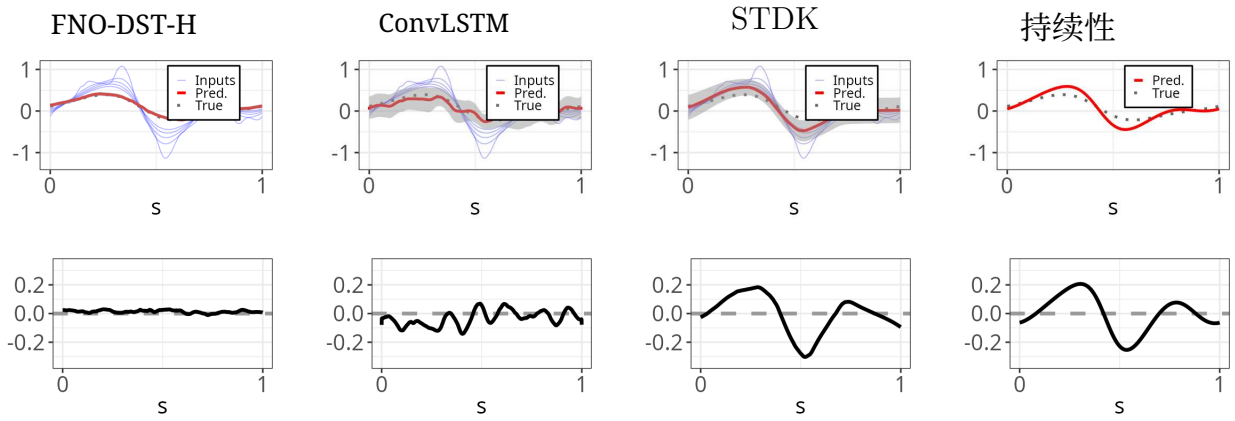
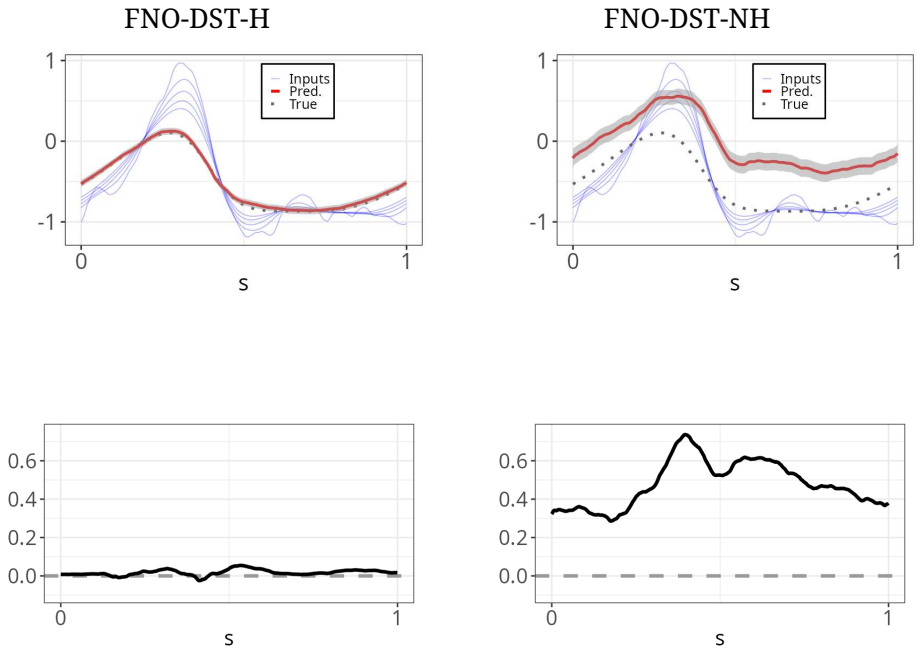
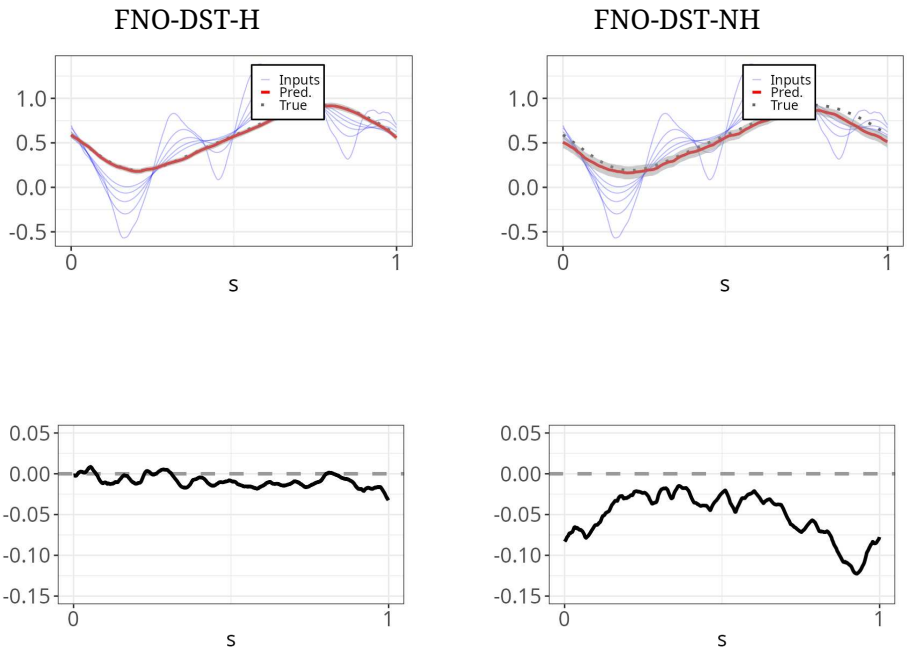


图 S.2：第二个测试实例的预测场（按模型排列的顶行）及相应的预测误差（按模型排列的底行），该实例是 $N_0 = 50$ 个测试实例之一。此可视化展示了模拟设置下随机 $\gamma = \gamma^{(1)}$时的预测质量（ $T = 10$ ）以及空间上的误差分布。顶行图中展示了当前时刻和此前四个时刻的空间场（蓝线）、 $T = 10$ 时刻的真实空间场（黑色虚线），以及预测结果（红线）和 95% 预测区间（灰色阴影区域）。



（a）随机 $\gamma = \gamma^{(1)}$的预测场（顶行）和误差（底行）。



（b）固定 $\gamma = \gamma^{(2)}$的预测场（顶行）和误差（底行）。

图 S.3：分别展示 FNO-DST-H 和 FNO-DST-NH 模型在 $T = 10$ 对第三个测试实例的预测，共 $N_0 = 50$ 个测试实例。(a)：$\gamma = \gamma^{(1)}$；(b)：$\gamma = \gamma^{(2)}$。(a) 和 (b) 的上方图中展示了当前及此前四个时间步（蓝线）、$T = 10$ 时的真实空间场（黑色点线），以及带有 95% 预测区间（灰色阴影区域）的预测结果（红线）。

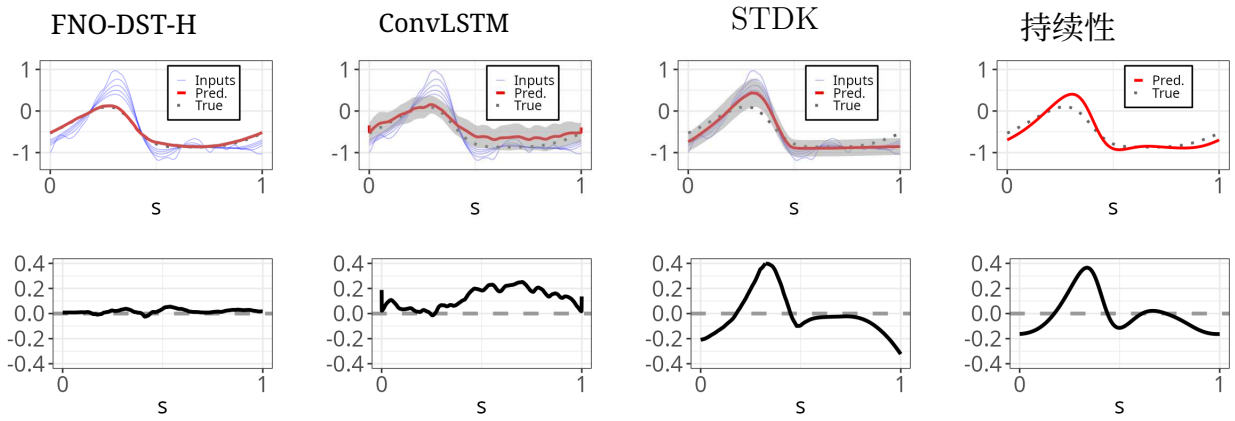


图 S.4：第三个测试实例（共 $N_0 = 50$ 个测试实例）的预测场（按模型排列于顶行）及其对应的预测误差（按模型排列于底行）。此可视化展示了在随机 $\gamma = \gamma^{(1)}$ 的模拟设置下，$T = 10$ 的预测质量以及误差在空间上的分布。在顶部各图中，展示了当前及前四个时间步（蓝线）、$T = 10$ 处的真实空间场（黑色虚线），以及带有 95% 预测区间（灰色阴影区域）的预测结果（红线）。

参考文献

Brezis, H. 和 Br´ezis, H.（2011年）。泛函分析、*Sobolev*空间与偏微分方程。第 2 卷，施普林格，纽约。参见第 8 章。

Cooley, J. 和 Tukey, J.（1965）。一种用于计算机计算复傅里叶级数的算法。*Mathematics of Computation*，19(90)：297–301。Evans, L. C.（2022年）。偏微分方程，第 *5* 章。第 19 卷，美国数学会，普罗维登斯。