基于神经算子的连续张量函数表示及其应用

Ruoyang Su, Xi-Le Z hao, *IEEE* 高级会员, Sheng Liu, Wei-Hao Wu, Yisi Luo, and Michael K. Ng

arXiv:2603.01812v1 [cs.CV] 2 Mar 2026

摘要——**近年来，连续张量函数受到越来越多的关注，因为它们能够统一表示网格上的数据以及网格之外的数据。然而，由于 mode-n product 本质上是离散且线性的，当前连续张量函数表示的潜力仍未得到充分发挥。为打破这一瓶颈，我们提出基于神经算子的 mode-n operators，作为离散且线性的 mode-$n$product 的连续非线性替代方案。所提出的 mode-n operator 不再将离散核心张量映射为离散目标张量，而是直接将连续核心张量函数映射为连续目标张量函数，从而为真实世界数据提供真正的连续表示，并减轻离散化伪影。借助连续非线性的 mode-n operators，我们提出一种基于神经算子的连续张量函数表示（简称 NO-CTR）。与经典的离散张量表示和连续张量函数表示相比，该方法能够更忠实地表示复杂的真实世界数据。从理论上讲，我们还证明了任意连续张量函数都可以由 NO-CTR 近似。为检验 NO-CTR 的能力，我们提出一种基于 NO-CTR 的多维数据补全模型。针对规则网格上的各类数据（多光谱图像和彩色视频）、具有不同分辨率的网格上的数据（Sentinel-2 图像）以及网格之外的数据（点云）所开展的大量实验，均证明了 NO-CTR 的优越性。**

索引词—**连续张量函数，张量分解，神经算子，多维数据补全**

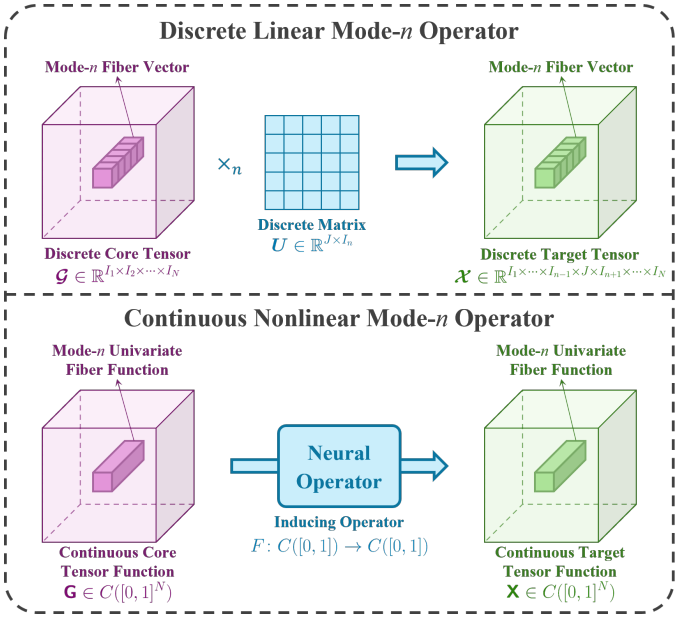


图 1. 离散线性模-n 算子与所提出的连续非线性模-n算子。离散线性模-n 算子作用于离散核心张量$\mathcal{G}$的模-n 纤维向量，并将其映射为离散目标张量$\mathcal{X}$的模-n 纤维向量。连续非线性模-n算子作用于连续核心张量函数$\mathbf{G}$的模-n单变量纤维函数，并将其映射为连续目标张量函数$\mathbf{X}$的模-n 单变量纤维函数。

I. 引言

MULTI-DIMENSIONAL 数据 [1]–[4] 是现代领域的基础，并广泛应用于各种场景，包括遥感成像 [5],[6], 目标检测 [7]–[9],和信号处理 [10]–[12]。有效表示并捕获这些多维数据的内在结构，对于分类 [13], [14], 压缩 [15], [16], 和恢复 [17]–[20]等任务至关重要。简单地将这些多维数据矩阵 [21]–[23]展平或向量化，往往无法保留高阶交互关系，从而导致性能和可解释性较差。因此，开发有效的方法来表示和处理多维数据仍然是一项根本性挑战。

作为向量和矩阵向更高维度的自然扩展，张量已成为表示多维数据的基础且常见的结构。传统张量方法主要侧重于低秩分解。CANDECOMP/PARAFAC（CP）分解 [24], [25]旨在将张量分解为多个秩一分量之和。Tucker分解 [26] 着重于将张量分解为一个核心张量，并沿每个模态与矩阵相乘。张量奇异值分解（t-SVD）[27]–[29] 旨在将三阶张量分解为一个 f-对角张量与两个正交张量的 t-积。张量列车（TT）分解 [30] 将一个 N阶张量分解为一系列矩阵、$N-2$ 个三阶张量以及另一个矩阵。张量环（TR）分解 [31] 的特点是将一个N阶张量分解为 N 个以循环方式相乘的三阶张量。

连续张量函数的出现彻底改变了多维数据表示。通过将坐标映射到相应的数据值，连续张量函数摆脱了对固定网格的依赖，并将网格上的数据与网格之外的数据无缝统一

(通讯作者：*Xi-Le Zhao)*Ruoyang Su、Xi-Le Zhao、Sheng Liu 和 Wei-Hao Wu 均就职于中国成都 611731 的电子科技大学数学科学学院（电子邮箱：202511110508@std.uestc.edu.cn；xlzhao122003@163.com；liusheng16@163.com；weihaowu99@163.com）。Yisi Luo 就职于中国西安 710049 的西安交通大学数学与统计学院（电子邮箱：yisiluo1221@foxmail.com）。Michael K. Ng 就职于中国香港的香港浸会大学数学系（电子邮箱：michael-ng@hkbu.edu.hk）。

到单一的灵活框架中，为多维数据提供了更加灵活的表示方式。借鉴传统张量分解的思想，研究人员提出了一系列连续张量函数表示。Imaizumi 等人 [32]提出了平滑 Tucker 分解（STD），该方法利用少量基函数逼近观测张量，并通过 Tucker分解对其进行分解。Fang 等人 [33] 提出了函数贝叶斯 Tucker 分解（FunBaT）。他们将连续索引数据视为 Tucker 核心张量与一组潜在函数之间的交互，并使用高斯过程作为函数先验来建模潜在函数。此外，Luo 等人 [34] 提出了一种由多层感知机参数化的低秩张量函数表示（LRTFR），凭借强大的表示能力，可以连续表示网格之外的数据。尽管取得了这一突破，现有的连续张量函数表示仍受到一个根本性限制：从离散核心张量到离散目标张量的映射（模-n乘积）本质上是离散且线性的，因此无法充分捕捉真实世界数据的复杂结构，使连续张量函数的潜力在很大程度上仍未得到释放。

为突破这一瓶颈，本文提出一种连续、非线性的 mode-n 积替代方案，即连续、非线性的 mode-n 算子。为实现这一目标，我们首次将神经算子 [35]–[37]引入连续张量函数表示领域，以真正捕捉真实世界数据的复杂结构。具体而言，我们利用神经算子，将连续核心张量函数的 mode-n 单变量函数直接映射为连续目标张量函数中对应的函数。随后，我们提出一种以神经算子为基础的连续张量函数表示（NO-CTR），将复杂的真实世界数据忠实地表示为连续核心张量函数与一系列连续、非线性的 mode-n 算子的复合。图 1 对比了以往离散、线性的 mode-n 算子与本文提出的连续、非线性的 mode-n算子。得益于连续、非线性的 mode-n算子强大的非线性关系捕捉能力，本文提出的 NO-CTR 能够更忠实地表示复杂的真实世界数据。NO-CTR 不仅释放了连续张量函数的潜力，也弥合了神经算子与张量表示之间的差距。

本文的主要贡献如下：

(1) 为突破离散线性模式-n乘积的局限，我们提出以神经算子为基础的模式-n 算子作为连续、非线性的替代方案，它能够为真实世界数据提供真正连续的表示，并减轻离散化伪影。

(2) 借助连续、非线性的模式-n算子，我们提出了神经算子驱动的连续张量函数表示（NO-CTR），能够忠实地表示复杂的真实世界数据。

（3）在理论上，我们证明了 NO-CTR 可以逼近任意连续张量函数。此外，为了检验 NO-CTR 的能力，我们提出了一种基于 NO-CTR 的多维数据补全模型。

（4）我们在规则网格上的各类数据（多光谱图像和彩色视频）、不同分辨率网格上的数据（Sentinel-2 图像）以及网格之外的数据（点云）上进行了大量实验，结果证明了 NO-CTR 的优越性。

II. 相关工作

A. 连续张量函数表示

最流行的连续张量函数表示之一是隐式神经表示（INR），其目标是利用深度神经网络学习多维数据关于坐标的连续张量函数。它最初用于三维形状表示，例如 [38], [39]。此后，它也被广泛用于表示各种多维数据。Sitzmann 等人[40]提出在隐式神经表示中使用周期激活函数，并证明这些网络（称为 SIREN）非常适合表示复杂的自然信号及其导数。Tancik 等人 [41]提出了一种为特定问题选择傅里叶特征的方法，大幅提升了多层感知机（MLP）在计算机视觉和图形学领域相关低维回归任务上的性能。Saragadam 等人 [42]开发了一种新的、高精度且稳健的 INR：小波隐式神经表示（WIRE），这是一种基于复数 Gabor 小波激活函数的新型 INR。Shi 等人 [43]提出了一种傅里叶重参数化方法，通过学习固定傅里叶基的系数矩阵来构成多层感知机的权重。近期，许多研究人员在这一领域开展了大量工作 [44]–[49]。INR 为多维数据提供了一种新颖的表示方法，但没有考虑多维数据的内在结构以及各因子之间的相互作用。

B. 神经算子

近年来，神经算子在科学计算中受到了广泛关注。在数学中，神经算子是从一个函数到另一个函数的映射 [50], [51],，具有非线性结构和强大的能力。它们在求解偏微分方程方面展现出巨大潜力，通常通过学习从参数函数到解函数的映射 [52]。Anandkumar 等人 [35]通过组合非线性激活函数和一类积分算子，构建了无穷维映射的逼近，其中核积分由图网络上的消息传递计算。Lu 等人 [36] 提出了一种通用深度学习框架，即深度算子网络（DeepONet），用于学习各种连续且非线性的算子。Li 等人 [37] 通过直接在傅里叶空间中参数化积分核，构建了一种新的神经算子，从而实现富有表现力且高效的架构。总之，神经算子是通常具有非线性结构的强大工具，能够灵活地从一个函数映射到另一个函数。

III. 符号与预备知识

在本文中，标量、向量、矩阵和（离散）张量分别用 x、 $\boldsymbol{x}$、 $\boldsymbol{X}$和$\boldsymbol{\mathcal{X}}$表示。而$\boldsymbol{\mathcal{X}}(i_1, i_2, \cdots, i_N)$表示 N阶离散张量$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$的第$(i_1, i_2, \cdots, i_N)$个元素。

一个 N 阶连续张量函数 [1]，（记作 $\mathbf{X}$）是一个具有稠密且完备的定义域的 N 元连续函数，其输入是一个具有 N 维的坐标，输出则是目标数据中对应的值。不失一般性，本文将坐标归一化到 $[0,1],$，因此 $\mathbf{X} \in C([0,1]^N)$。

纤维是矩阵的行和列在高阶情形下的对应概念。纤维定义为固定除一个索引 [54] 之外的所有索引。对于离散张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$，模式-$n$ 纤维向量记为

$$\boldsymbol{\mathcal{X}}(i_1, i_2, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_N) \in \mathbb{R}^{I_n}.$$

类似地，对于连续张量函数 $\mathbf{X} \in C([0,1]^N)$，模式-$n$ 单变量纤维函数记为

$$\mathbf{X}(y_1, y_2, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N) \in C([0,1]).$$

定义 *1*（模式-n 展开 *[54]*）：给定离散张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$，模式-$n$ 展开记为 $\boldsymbol{X}_{(n)}$，其将模式-n纤维向量排列为所得矩阵的列，定义如下：

$$\boldsymbol{X}_{(n)}(i_n, j) = \boldsymbol{\mathcal{X}}(i_1, i_2, \cdots, i_n, \cdots, i_N),$$

where $j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m$, and $n = 1, 2, \cdots, N$. The inverse operation of mode-n unfolding is denoted as fold_n, i.e. $\boldsymbol{\mathcal{X}} = \text{fold}_n(\boldsymbol{X}_{(n)})$.

IV. 重新审视模-n乘积

以往的张量表示通常将张量表示为一组因子。因子之间的运算或连接在很大程度上决定了表示能力。本节将从一个全新的视角——算子的视角，重新审视这些表示。

模-n 乘积广泛应用于张量表示中，其定义如下：

定义 *2*（模-n 乘积 *[54]*）：给定一个 N 阶离散张量 $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 和一个矩阵 $\boldsymbol{U} \in \mathbb{R}^{J \times I_n}$，模 -$n$ 乘积 $\boldsymbol{\mathcal{G}} \times_n \boldsymbol{U} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$定义为

$$\begin{aligned} &\boldsymbol{\mathcal{G}} \times_n \boldsymbol{U}(i_1, i_2, \cdots, i_{n-1}, j, i_{n+1}, \cdots, i_N) \\ &= \sum_{i_n=1}^{I_n} \boldsymbol{\mathcal{G}}(i_1, i_2, \cdots, i_{n-1}, i_n, i_{n+1}, \cdots, i_N) \boldsymbol{U}(j, i_n), \end{aligned}$$

也可以等价地表示为

$$\boldsymbol{\mathcal{G}} \times_n \boldsymbol{U} = \text{fold}_n(\boldsymbol{U}\boldsymbol{G}_{(n)}),$$

其中 $n = 1,\ 2,\ \cdots,\ N$。

[1]在以往文献中，这一概念还有其他名称，例如连续索引张量 [33], 函数张量 [53], 张量函数 [34]。

为了更好地理解模式-n 乘积的作用，我们从算子的角度重新解释它：

定义 *3*（离散线性模态n 算子）：给定一个 N 阶离散张量 $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 和一个矩阵$\boldsymbol{U} \in \mathbb{R}^{J \times I_n}$，由其诱导的离散线性模态-$n$ 算子

$$\mathscr{U}^{[n]} : \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N} \to \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$$

由 $\boldsymbol{U}$ 诱导的定义为

$$\begin{aligned} &\mathscr{U}^{[n]}(\boldsymbol{\mathcal{G}})(i_1, i_2, \cdots, i_{n-1}, j, i_{n+1}, \cdots, i_N) \\ &= (\boldsymbol{U}\boldsymbol{\mathcal{G}}(i_1, i_2, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_N))(j), \end{aligned} \tag{1}$$

其中 $n = 1,\ 2,\ \cdots,\ N$。

结合定义 2，不难看出，离散线性模态-n 算子等价于模态-n乘积，即

$$\mathscr{U}^{[n]}(\boldsymbol{\mathcal{G}}) = \boldsymbol{\mathcal{G}} \times_n \boldsymbol{U}.$$

如图 1 所示，离散线性模态-n 算子 $\mathscr{U}^{[n]}$ 作用的基本单元是模态-n 纤维向量

$$\boldsymbol{\mathcal{G}}(i_1, i_2, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_N) \in \mathbb{R}^{I_n}$$

离散张量$\boldsymbol{\mathcal{G}}$的模态-n纤维向量，随后被映射为模态-n纤维向量

$$\mathscr{U}^{[n]}(\boldsymbol{\mathcal{G}})(i_1, i_2, \cdots, i_{n-1}, :, i_{n+1}, \cdots, i_N) \in \mathbb{R}^{J}$$

离散目标张量 $\mathscr{U}^{[n]}(\boldsymbol{\mathcal{G}})$。因此，在模-$n$ 乘积中，从 $\boldsymbol{\mathcal{G}}$ 到 $\mathscr{U}^{[n]}(\boldsymbol{\mathcal{G}})$ 的映射本质上是离散且线性的。

从算子的角度出发，我们从这一视角回顾先前的表示方法 Tucker分解 [26] 和 LRTFR [34]。它们都将离散目标张量表示为一个离散核心张量与一组离散且线性的模-n 算子的复合。在数学上，两者均可重写为

$$\boldsymbol{\mathcal{X}} = \mathscr{U}^{[N]} \circ \cdots \circ \mathscr{U}^{[2]} \circ \mathscr{U}^{[1]}(\boldsymbol{\mathcal{G}}), \tag{2}$$

其中，$\boldsymbol{\mathcal{G}}$ 是离散核心张量，$\mathscr{U}^{[n]}$ 是由因子矩阵 $\boldsymbol{U}_n(n = 1, 2, \cdots, N)$诱导的离散且线性的模-$n$ 算子。在Tucker分解 [26]中，$\{\boldsymbol{U}_n\}_{n=1}^{N}$是普通矩阵；而在LRTFR [34]中，$\{\boldsymbol{U}_n\}_{n=1}^{N}$ 是从由隐式神经表示实现的函数中采样得到的矩阵。尽管核心张量或因子矩阵可以由神经网络生成，但从离散核心张量 $\boldsymbol{\mathcal{G}}$ 到离散目标张量 $\boldsymbol{\mathcal{X}}$ 的映射本质上是线性的，这限制了这些表示方法的能力。此外，真实世界数据通常较为复杂，且包含大量非线性关系，仅通过线性映射也难以准确捕捉这些关系。

V. 基于神经算子的连续张量函数表示

在本节中，我们将提出连续的非线性模态-n算子。基于此，我们将提出神经算子基础的连续张量函数表征（NO-CTR），并证明任何连续张量函数都可以通过NO-CTR 进行近似。最后，我们将提出一种基于NO-CTR 的多维数据补全模型。

A. 连续与非线性 *Mode-*n算子

为了解决离散线性模态-n算子的上述局限性，本小节将提出连续的非线性模态-n 算子。

在第四节中，我们从算子的角度重新审视了 mode-n 积。从这一视角来看，离散线性 mode-n 算子作用的基本单元是 mode-n 纤维向量。这一视角启发我们为连续张量函数定义新的交互方式。具体而言，所提出的连续非线性 mode-n 算子直接作用于连续张量函数的 mode-n 单变量纤维函数，同时保留了离散线性 mode-n 算子的这一关键性质。下面给出连续非线性 mode-n 算子的定义。

定义 *4*（连续与非线性 *Mode-*n 算子）：给定一个 N 阶连续张量函数$\mathbf{G} \in C([0,1]^N)$以及一个连续算子 F: $C([0,1]) \to C([0,1])$，连续非线性 mode-n 算子

$$\mathscr{F}^{\langle n\rangle}: C([0,1]^N) \to C([0,1]^N)$$

由 F 诱导，定义为

$$\begin{aligned}&\mathscr{F}^{\langle n\rangle}(\mathbf{G})(y_1, y_2, \cdots, y_N)\\ &= F\big(\mathbf{G}(y_1, y_2, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N)\big)(y_n),\end{aligned} \tag{3}$$

其中 $n = 1, 2, \cdots, N$。为避免混淆，我们将 F 称为诱导算子。

在公式（3）中，我们从连续张量函数$\mathbf{G}$ 出发，而不是离散张量$\boldsymbol{\mathcal{G}}$。如图1所示，连续非线性模-n 算子 $\mathscr{F}^{\langle n\rangle}$ 作用的基本单元是连续张量函数的模-n 单变量纤维函数

$$\mathbf{G}(y_1, y_2, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N) \in C([0,1])$$

，随后这些函数被映射为连续目标张量函数$\mathbf{G}$的模-n 单变量纤维函数

$$F\big(\mathbf{G}(y_1, y_2, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N)\big) \in C([0,1])$$

。此外，诱导算子 F通常是非线性的，因此连续非线性模-n 算子是从$\mathbf{G}$ 到$\mathscr{F}^{\langle n\rangle}(\mathbf{G})$的非线性映射。

最后，我们通过实验结果进一步展示连续非线性运算的优势。图2展示了采样率为10%时，采用离散线性模-n 算子（LRTFR [34]）和连续非线性模-n 算子（提出的NO-CTR）所得的视觉及定量恢复结果。在定量结果方面，提出的NO-CTR在各种多维数据上均优于LRTFR。在视觉结果方面，提出的NO-CTR能够更准确地恢复细节，例如衣物上的条纹和青蛙的眼睛。这表明了所提出的连续非线性模-n算子的优越性。

B. 连续张量函数表示

借助连续且非线性的模态-n 算子，本小节将提出一种神经算子驱动的连续张量函数表示（NO-CTR）。下面给出其数学定义。

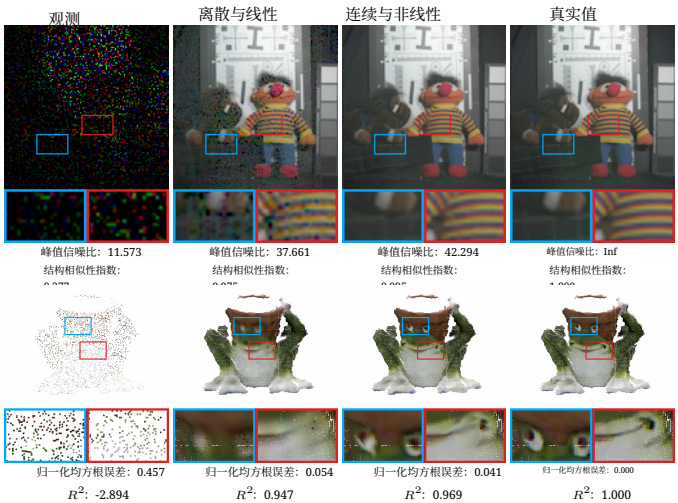


图2. 在采样率为10%时，多光谱图像玩具（上）和点云青蛙（下）上，采用离散线性模态-n 算子（LRTFR [34]）以及连续非线性模态-n 算子（所提出的NO-CTR）得到的视觉和定量恢复结果。

定义*5*（*NO-CTR*）：对于一个 N阶连续张量函数$\mathbf{X} \in C([0,1]^N)$，其神经算子驱动的连续张量函数表示（简称 NO-CTR）可表述为

$$\mathbf{X} = \mathscr{F}_N^{\langle N\rangle} \circ \cdots \circ \mathscr{F}_2^{\langle 2\rangle} \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G}), \tag{4}$$

其中，$\mathbf{G} \in C([0,1]^N)$表示连续核心张量函数，而 $\mathscr{F}_n^{\langle n\rangle}$表示由 $F_n(n = 1、2、\cdots、N)$诱导的连续非线性模态-n 算子。

在公式（2）中，先前的表示方法Tucker分解 [26]和 LRTFR [34] 将离散目标张量$\boldsymbol{\mathcal{X}}$ 表示为由一组离散线性模态-n 算子$\{\mathscr{U}^{[n]}\}_{n=1}^N$复合而成的离散核心张量$\boldsymbol{\mathcal{G}}$ 。类似地，在公式（4）中，所提出的NO-CTR将复杂真实世界数据 $\mathbf{X}$ 表示为由一系列连续非线性模态-n 算子$\{\mathscr{F}^{\langle n\rangle}\}_{n=1}^N$复合而成的连续核心张量函数$\mathbf{G}$ 。得益于连续模态-n算子捕获非线性关系的强大能力，NO-CTR能够更加忠实地表示复杂真实世界数据。

C. 实现方法

在公式（4）中，连续核心张量函数$\mathbf{G}$ 以及连续非线性模态-n 算子 $\{\mathscr{F}^{\langle n\rangle}\}_{n=1}^N$ 是 NO-CTR 的关键组成部分。本小节将讨论具体的实现方法。

1）连续核心张量函数：连续核心张量函数 $\mathbf{G} \in C([0,1]^N)$本质上是在 $[0,1]^N$ 上定义的多变量连续函数。在实践中，神经网络非常适合表示这类连续函数，而且神经网络的选择十分多样。本文选择广受欢迎的隐式神经表示 SIREN [40] 来实现连续核心张量函数 $\mathbf{G}$：它以坐标 $\boldsymbol{y} \in [0,1]^N$ 作为输入，并输出对应的值 $\mathbf{G}(\boldsymbol{y}) \in \mathbb{R}$。我们还将在第VII-C节讨论连续核心张量函数的其他实现方法。

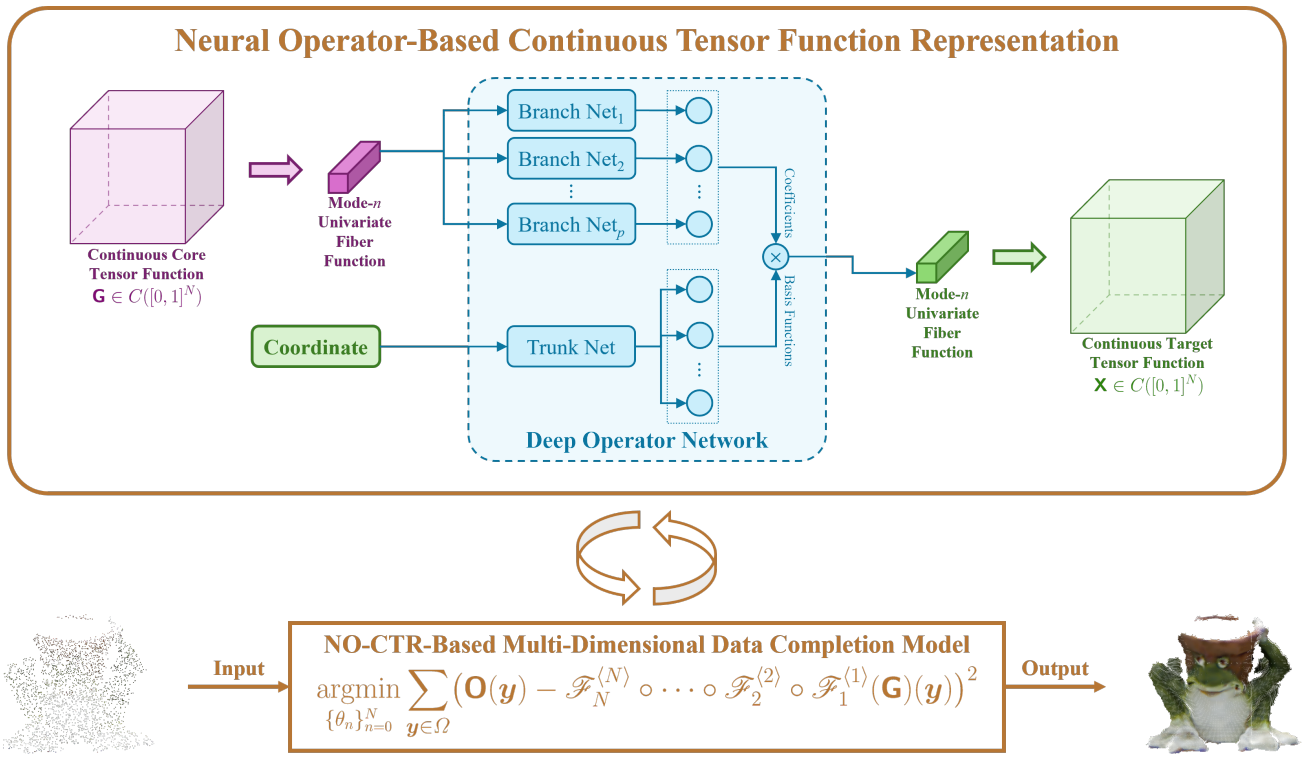


图 3. NO-CTR 及其对应的多维数据补全模型流程图。我们首先巧妙地利用神经算子构建连续非线性的 mode-n 算子。具体而言，连续非线性 mode-n 算子作用的基本单元是连续核心张量函数 $\mathbf{G}$ 的 mode-n 单变量纤维函数，随后这些函数被映射为连续目标张量函数 $\mathbf{X}$ 的 mode-n 单变量纤维函数。借助连续非线性的 mode-n 算子，我们提出了基于神经算子的连续张量函数表示（NO-CTR），它能够将连续目标张量函数 $\mathbf{X}$ 忠实地表示为连续核心张量函数 $\mathbf{G}$ 和一系列连续非线性的 mode-n 算子 $\{\mathscr{F}^{\langle n\rangle}\}_{n=1}^N$。为检验 NO-CTR 的能力，我们提出了一个基于 NO-CTR 的多维数据补全模型。

2）连续非线性模态-n算子：在数学中，任何函数算子都可用于实现连续非线性模态-n 算子。本文巧妙地利用了一种新兴且强大的工具：神经算子，为我们提供了正确且高效的选择。神经算子在科学计算中发挥着重要作用[35]–[37]。本质上，神经算子是从一个函数到另一个函数的映射 [50], [51]。这使我们能够直接将神经算子视为诱导算子，开创了神经算子在张量表示领域的首次应用。

在各种神经算子中，我们选择深度算子网络（DeepONet）[34]来实现连续非线性的 mode-n 算子，并将在第 VII-B 节讨论神经算子的其他架构。DeepONet 由主干网络和分支网络组成。具体而言，对于诱导算子 F_n: $C([0,1]) \to C([0,1])$,

$$F_n(v)(y) = \sum_{p=1}^{P_n} b_p^{(n)}\left(v(z_1), v(z_2), \cdots, v(z_{m_n})\right) t_p^{(n)}(y),$$

其中，$v \in C([0,1])$ 是输入单变量函数，$y \in [0,1]$ 是一个坐标。$\boldsymbol{t}^{(n)}$ 是一个称为主干网络的向量值函数，$\{t_p^{(n)}\}_{p=1}^P$ 是 $\boldsymbol{t}$的 P 个分量，可以从物理角度将其解释为输出空间的基函数。$\{b_p^{(n)}\}_{p=1}^{P_n}$ 是称为分支网络的标量值函数，用于编码输入函数的离散采样值在固定传感器点 $\{z_i\}_{i=1}^{m_n}$ （称为传感器）处的函数 v ，提供线性组合所需的系数。

D. 通用逼近定理

为从理论上说明上述实现的合理性，我们提出并证明了 NO-CTR 的稠密性，这表明任意连续张量函数都可以由 NO-CTR 逼近，并且可以视为一个通用逼近定理。

定理 *1*：对于任意连续张量函数 $\mathbf{X}$ $\in C([0,1]^N)$, $\forall \varepsilon > 0$，存在一个全连接神经网络

$$\mathbf{G}\colon [0,1]^N \to \mathbb{R}$$

以及 N DeepONets [36]

$$F_n\colon C([0,1]) \to C([0,1]), \quad (n = 1, 2, \cdots, N)$$

使得

$$\|\mathbf{X} - \mathscr{F}_N^{\langle N\rangle} \circ \cdots \circ \mathscr{F}_2^{\langle 2\rangle} \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| < \varepsilon,$$

其中 $\mathscr{F}_n$ 是由 F_n 诱导的连续非线性模态-n 算子。

证明见附录。

E. 多维数据补全模型

为考察 NO-CTR 的能力，我们提出了一种基于 NO-CTR 的多维数据补全模型，旨在从不完整的观测中估计缺失元素。

在数学上，该模型可表述为

$$\underset{\{\theta_n\}_{n=0}^{N}}{\operatorname{argmin}} \sum_{\boldsymbol{y}\in\Omega} \left(\mathbf{O}(\boldsymbol{y}) - \mathscr{F}_N^{\langle N\rangle} \circ \cdots \circ \mathscr{F}_2^{\langle 2\rangle} \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})(\boldsymbol{y})\right)^2, \quad (5)$$

其中 $\mathbf{O} \in C([0,1]^N)$ 是观测的连续张量函数， $\boldsymbol{y}$ 是一个坐标， $\Omega \subseteq [0,1]^N$ 是由所有观测元素的坐标组成的集合， $\mathbf{G} \in C([0,1]^N)$ 是连续核心张量函数， $\mathscr{F}_n$ 是由 F_n 诱导的连续非线性模态-n 算子， θ_0 表示连续核心张量函数 $\mathbf{G}$ 的参数，而θ_n 表示诱导算子 F_n $(n = 1, 2, \cdots, N)$ 的参数。

VI. 实验

为全面评估 NO-CTR 的性能，我们针对规则网格上的各种数据（多光谱图像和彩色视频）、不同分辨率网格上的数据（Sentinel-2 图像）以及网格之外的数据（点云）开展了大量实验。在这些实验中，通过以四种不同的比例（5%、10%、15% 和 20%）随机采样原始真实值来生成观测数据。每项实验中，所有方法都仅使用一个观测作为训练数据。

对于网格上的数据，对比方法包括传统张量方法（TR-ALS [55]）和连续表示方法（SIREN [40], MFN [44], FR-INR [43]和 LRTFR [34]）；对于网格之外的数据，由于传统张量方法无法处理这类数据，对比方法仅包括连续表示方法（SIREN [40],MFN [44], FR-INR [43] 和 LRTFR [34]）。在实现方面，TR-ALS [55] 使用 MATLAB 编写，其他方法则基于 Anaconda 环境中的 PyTorch 库实现。

所有基线方法的超参数配置均严格遵循其原始出版物或作者提供的官方源代码中的设置。对于所提出的 NO-CTR，超参数则根据相关文献 [34], [40]并结合大量经验调试进行调整。此外，考虑到多维数据的结构以及计算能力的限制，本文实验中将沿空间模态的所有连续非线性模态-n算子均设为恒等算子。在训练循环中，每 100 次迭代通过比较重建结果与真实值进行定量评估，并选取性能最佳的检查点作为最终结果。

为进行客观、定量的比较，采用四种广泛使用的指标来评估重建质量：峰值信噪比（PSNR）、结构相似性（SSIM） [56], 归一化均方根误差（NRMSE）以及决定系数（R^2）作为质量指标。通常，更高的 PSNR、SSIM 和 R^2 值，以及更低的 NRMSE 值，表明重建质量更高。

A. 多光谱图像

首先，我们在规则网格上的传统经典多维数据——多光谱图像（MSI）上进行实验。我们使用的数据库 [57] 涵盖种类繁多的真实世界材料。每幅多维图像均调整为 128×128 像素，包含 31 个波段。

表 I 列出了多光谱图像补全的定量恢复结果。在所有实验设置下，所提出的 NO-CTR 均展现出优异性能：对于所有多光谱图像和所有采样率，其峰值信噪比和结构相似性指数均高于所有对比方法。在低采样率下，这一优势同样明显；随着采样率提高，优势进一步扩大。图 4 展示了采样率为 10% 时多光谱图像补全的视觉恢复结果。所提出的 NO-CTR 在恢复细节和纹理方面表现出色。与其他方法相比，NO-CTR 的恢复结果更接近真实值。

B. 彩色视频

随后，我们在规则网格上的高阶多维数据——彩色视频上进行实验。彩色视频由按时间顺序排列的帧序列构成。每段彩色视频均调整为 144×176 像素，包含三个颜色通道（红绿蓝），由连续的 30 帧组成。

表 II 列出了彩色视频补全的定量恢复结果。在所有彩色视频和所有采样率下，所提出的 NO-CTR 始终优于所有竞争方法，并在每种实验设置中都取得了最高的峰值信噪比和结构相似性指数值。这种优势在高采样率下尤为明显，在低采样率下也依然保持领先。图 5 展示了采样率为 10% 时彩色视频补全的视觉恢复结果。所提出的 NO-CTR 能够有效恢复动态场景的视觉细节。对于这些彩色视频，所提出的 NO-CTR 通过恢复更清晰的纹理和更锐利的边缘，优于竞争方法，从而更好地保留了彩色视频的视觉信息。

C. 哨兵-*2* 图像

接下来，我们在具有不同空间分辨率的网格上，对多维数据——哨兵-2 图像——开展实验。哨兵-2 是一项欧洲任务，采用宽幅、高分辨率多光谱成像。该任务配备了光学仪器载荷，可采集 13 个光谱波段，包括空间分辨率为 10 m 的 4 个波段、20 m 的 6 个波段以及 60 m 的 3 个波段。相应图像分别裁剪为 300×300 像素、 150×150 像素和50×50 像素。

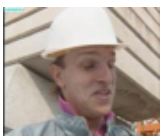
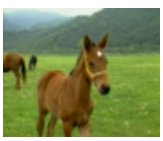
表 III 列出了 Sentinel-2 图像补全的定量恢复结果。所提出的 NO-CTR 方法在三个数据集和所有评估采样率下均表现出色，几乎全面优于其他方法

表 I 多光谱图像补全的定量恢复结果

采样率		5%			10%			15%			20%		
MSI	方法	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差
	观测	15.532	0.045	0.274	15.769	0.078	0.266	16.023	0.112	0.259	16.277	0.148	0.251
	TR-ALS	26.918	0.672	0.074	30.129	0.841	0.051	31.514	0.879	0.043	31.977	0.890	0.041
	SIREN	29.737	0.842	0.053	34.377	0.946	0.031	37.860	0.982	0.021	41.742	0.992	0.013
	MFN	28.507	0.806	0.061	31.843	0.907	0.042	36.741	0.975	0.024	40.022	0.989	0.016
	FR-INR	32.099	0.918	0.041	37.273	0.977	0.022	41.133	0.991	0.014	43.701	0.995	0.011
	LRTFR	30.631	0.859	0.048	34.300	0.946	0.032	37.817	0.977	0.021	40.461	0.988	0.016
布料	NO-CTR	**33.906**	**0.943**	**0.033**	**39.188**	**0.985**	**0.018**	**42.793**	**0.994**	**0.012**	**45.579**	**0.997**	**0.009**
	观测	16.269	0.372	0.168	16.497	0.403	0.163	16.744	0.436	0.159	17.010	0.468	0.154
	TR-ALS	27.399	0.729	0.047	31.348	0.873	0.030	34.373	0.921	0.021	35.974	0.933	0.017
	SIREN	33.706	0.925	0.023	38.681	0.977	0.013	41.702	0.989	0.009	46.041	0.995	0.005
	MFN	29.564	0.823	0.036	35.092	0.943	0.019	39.800	0.977	0.011	43.464	0.989	0.007
	FR-INR	34.713	0.941	0.020	39.783	0.981	0.011	43.309	0.991	0.007	46.686	0.994	0.005
	LRTFR	33.619	0.930	0.023	36.879	0.965	0.016	40.100	0.984	0.011	43.249	0.991	0.008
花卉	NO-CTR	**37.121**	**0.981**	**0.015**	**42.591**	**0.994**	**0.008**	**46.456**	**0.997**	**0.005**	**49.717**	**0.999**	**0.004**
	观测	16.574	0.105	0.258	16.810	0.141	0.251	17.059	0.177	0.244	17.328	0.214	0.237
	TR-ALS	29.634	0.789	0.057	33.219	0.892	0.038	35.558	0.928	0.029	36.321	0.938	0.027
	SIREN	34.747	0.918	0.032	39.731	0.973	0.018	43.023	0.986	0.012	46.164	0.994	0.009
	MFN	32.563	0.863	0.041	37.103	0.943	0.024	41.731	0.980	0.014	44.520	0.990	0.010
	FR-INR	35.374	0.934	0.030	40.407	0.977	0.017	43.850	0.989	0.011	46.953	0.994	0.008
	LRTFR	34.597	0.925	0.032	38.807	0.967	0.020	41.587	0.982	0.014	44.254	0.989	0.011
绘画	NO-CTR	**38.147**	**0.972**	**0.022**	**42.971**	**0.990**	**0.012**	**46.456**	**0.995**	**0.008**	**49.157**	**0.998**	**0.006**
	观测	11.332	0.241	0.278	11.573	0.277	0.270	11.826	0.314	0.263	12.085	0.351	0.255
	TR-ALS	25.245	0.746	0.056	29.733	0.881	0.033	33.762	0.932	0.021	35.000	0.943	0.018
	SIREN	31.916	0.915	0.026	37.259	0.974	0.014	42.048	0.988	0.008	45.306	0.995	0.006
	MFN	28.157	0.802	0.040	34.213	0.922	0.020	38.931	0.969	0.012	42.759	0.987	0.007
	FR-INR	32.818	0.919	0.023	39.103	0.975	0.011	43.359	0.989	0.007	46.464	0.996	0.005
	LRTFR	31.836	0.928	0.026	37.661	0.975	0.013	40.447	0.986	0.010	43.099	0.991	0.007
Toy	NO-CTR	**35.156**	**0.980**	**0.018**	**42.294**	**0.995**	**0.008**	**46.493**	**0.998**	**0.005**	**48.878**	**0.999**	**0.004**

最佳 和次佳结果以高亮显示。

表 II 彩色视频补全的定量恢复结果

采样率		5%			10%			15%			20%		
彩色视频	方法	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差
	观测	6.947	0.015	0.436	7.180	0.032	0.425	7.434	0.056	0.412	7.684	0.081	0.401
	TR-ALS	16.595	0.509	0.144	19.223	0.697	0.106	21.363	0.803	0.083	22.811	0.850	0.070
	SIREN	18.877	0.690	0.110	21.631	0.825	0.080	23.401	0.875	0.066	24.950	0.914	0.055
	MFN	17.209	0.583	0.134	19.180	0.704	0.107	21.312	0.807	0.083	23.153	0.873	0.068
	FR-INR	19.337	0.723	0.105	21.763	0.826	0.079	23.519	0.880	0.065	24.737	0.904	0.056
	LRTFR	18.340	0.661	0.118	20.705	0.777	0.090	22.437	0.850	0.073	23.423	0.870	0.065
Bird	NO-CTR	**19.643**	**0.765**	**0.101**	**22.030**	**0.860**	**0.077**	**24.026**	**0.908**	**0.061**	**25.382**	**0.933**	**0.052**
	观测	3.895	0.007	0.676	4.128	0.013	0.658	4.375	0.019	0.639	4.641	0.025	0.620
	TR-ALS	22.867	0.692	0.076	25.656	0.803	0.055	27.461	0.852	0.045	28.990	0.890	0.038
	SIREN	24.450	0.805	0.063	27.119	0.879	0.047	29.072	0.908	0.037	30.715	0.931	0.031
	MFN	22.931	0.717	0.076	25.043	0.787	0.059	26.719	0.848	0.049	29.456	0.900	0.036
	FR-INR	24.955	0.816	0.060	27.665	0.878	0.044	29.711	0.910	0.035	31.421	0.935	0.028
	LRTFR	24.712	0.789	0.062	27.267	0.862	0.046	28.818	0.895	0.038	29.853	0.914	0.034
Foreman	NO-CTR	**26.759**	**0.891**	**0.049**	**29.719**	**0.940**	**0.035**	**31.841**	**0.957**	**0.027**	**32.940**	**0.967**	**0.024**
	观测	7.219	0.011	0.425	7.451	0.025	0.414	7.699	0.043	0.402	7.978	0.063	0.389
	TR-ALS	23.944	0.882	0.062	26.024	0.918	0.049	28.047	0.946	0.039	29.172	0.954	0.034
	SIREN	25.705	0.911	0.051	27.739	0.942	0.040	29.193	0.957	0.034	30.444	0.964	0.029
	MFN	24.984	0.899	0.055	26.897	0.927	0.044	28.632	0.944	0.036	30.012	0.964	0.031
	FR-INR	26.384	0.928	0.047	28.326	0.946	0.037	29.831	0.958	0.031	31.117	0.970	0.027
	LRTFR	26.766	0.932	0.045	28.622	0.954	0.036	30.228	0.966	0.030	30.983	0.971	0.028
Horse	NO-CTR	**27.459**	**0.949**	**0.041**	**29.504**	**0.966**	**0.033**	**30.849**	**0.976**	**0.028**	**32.075**	**0.981**	**0.024**

最佳和次佳结果已突出显示。

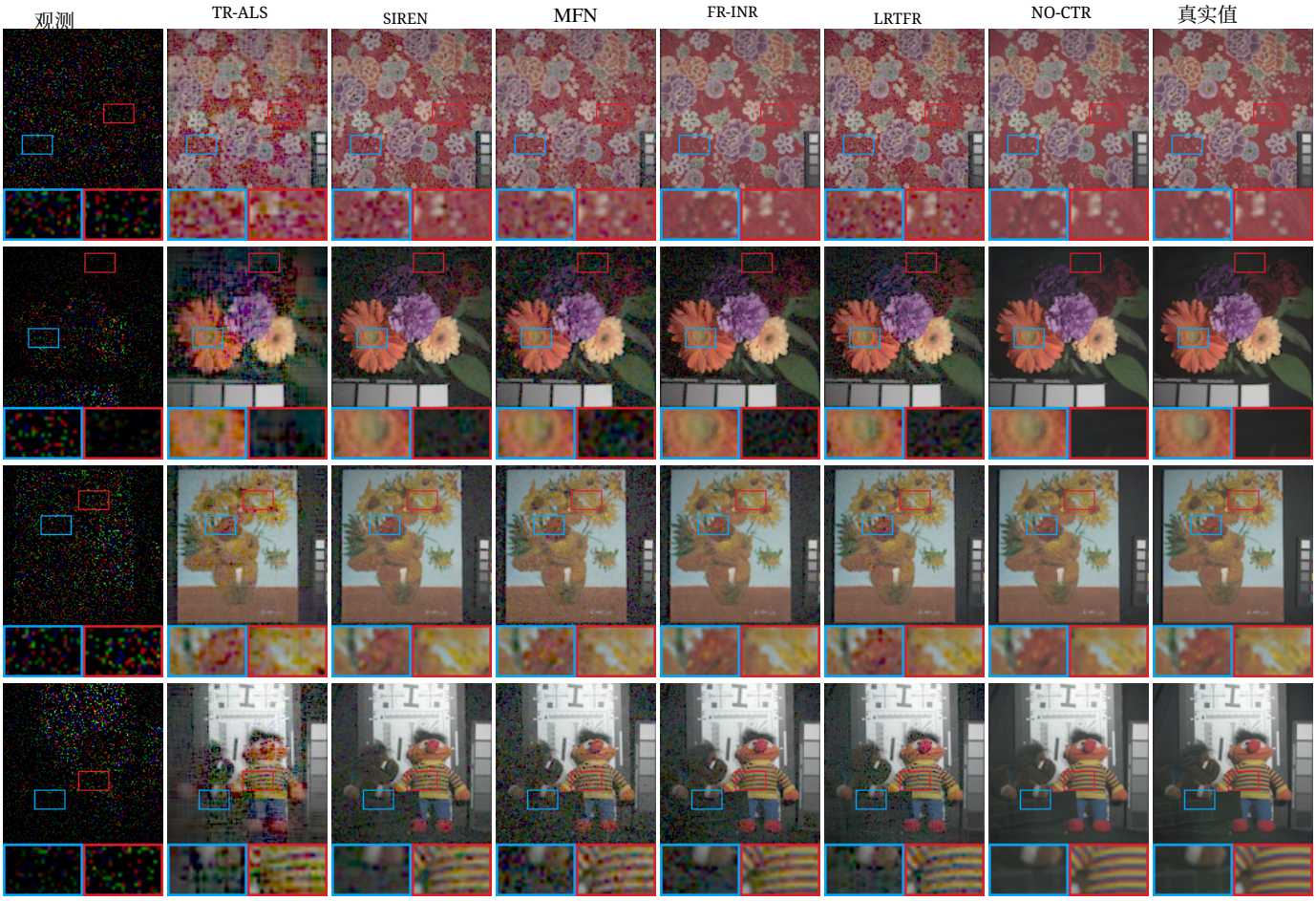


图 4. 采样率为 10% 时多光谱图像补全的视觉恢复结果。

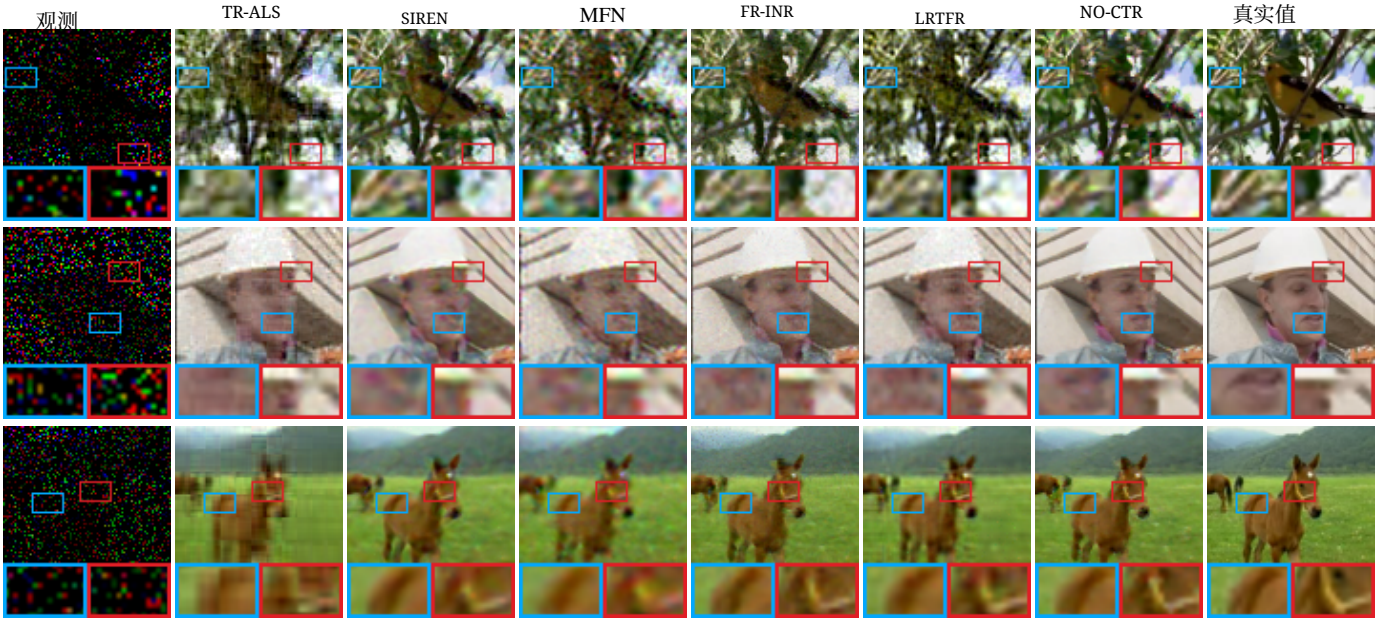


图 5. 采样率为 10% 时彩色视频补全的视觉恢复结果。

表 III 哨兵-2号图像补全的定量恢复结果

采样率		5%			10%			15%			20%		
数据集	方法	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差	PSNR	SSIM	归一化均方根误差
	观测	9.512	0.015	0.454	9.746	0.022	0.443	9.998	0.027	0.430	10.267	0.031	0.417
	TR-ALS	25.993	0.508	0.082	29.151	0.567	0.071	30.638	0.688	0.059	32.071	0.705	0.057
	SIREN	32.382	0.715	0.056	33.999	0.770	0.049	35.197	0.834	0.038	36.280	0.862	0.036
	MFN	32.202	0.734	0.058	33.814	0.800	0.046	34.967	0.848	0.039	35.904	0.871	0.034
	FR-INR	**33.243**	0.777	0.050	34.673	0.833	0.041	35.288	0.868	0.035	36.499	0.890	0.031
	LRTFR	30.534	0.706	0.060	32.039	0.809	0.043	33.240	0.861	0.036	34.227	0.882	0.031
T30UYV	NO-CTR	33.206	**0.822**	**0.045**	**35.231**	**0.873**	**0.037**	**36.389**	**0.900**	**0.032**	**37.077**	**0.916**	**0.029**
	观测	8.587	0.014	0.485	8.819	0.019	0.472	9.078	0.024	0.458	9.340	0.028	0.445
	TR-ALS	22.935	0.461	0.101	27.835	0.608	0.061	30.139	0.696	0.051	31.615	0.691	0.045
	SIREN	32.530	0.698	0.052	33.839	0.790	0.039	34.834	0.821	0.036	36.026	0.858	0.027
	MFN	31.518	0.726	0.053	33.345	0.796	0.041	34.575	0.844	0.036	35.604	0.878	0.030
	FR-INR	32.974	0.792	0.043	34.507	0.849	0.036	35.381	0.874	0.031	36.299	0.891	0.028
	LRTFR	30.111	0.745	0.047	31.740	0.824	0.037	33.043	0.859	0.033	33.971	0.884	0.029
T32TQR	NO-CTR	**33.068**	**0.829**	**0.040**	**35.100**	**0.886**	**0.031**	**36.287**	**0.911**	**0.027**	**36.971**	**0.928**	**0.024**
	观测	8.883	0.015	0.479	9.118	0.021	0.467	9.357	0.026	0.454	9.633	0.032	0.440
	TR-ALS	24.291	0.509	0.085	27.084	0.573	0.061	29.234	0.660	0.053	31.625	0.707	0.044
	SIREN	32.547	0.765	0.043	34.807	0.833	0.033	36.176	0.867	0.027	37.280	0.882	0.023
	MFN	31.761	0.769	0.050	33.921	0.839	0.038	34.898	0.869	0.033	36.537	0.901	0.027
	FR-INR	33.329	0.820	0.043	34.955	0.871	0.033	35.812	0.897	0.027	36.937	0.908	0.024
	LRTFR	30.691	0.797	0.044	32.433	0.846	0.036	33.681	0.884	0.030	34.611	0.909	0.026
T33TUL	NO-CTR	**33.673**	**0.879**	**0.036**	**35.892**	**0.920**	**0.028**	**37.136**	**0.938**	**0.024**	**38.100**	**0.946**	**0.021**

最佳和次佳结果已突出显示。

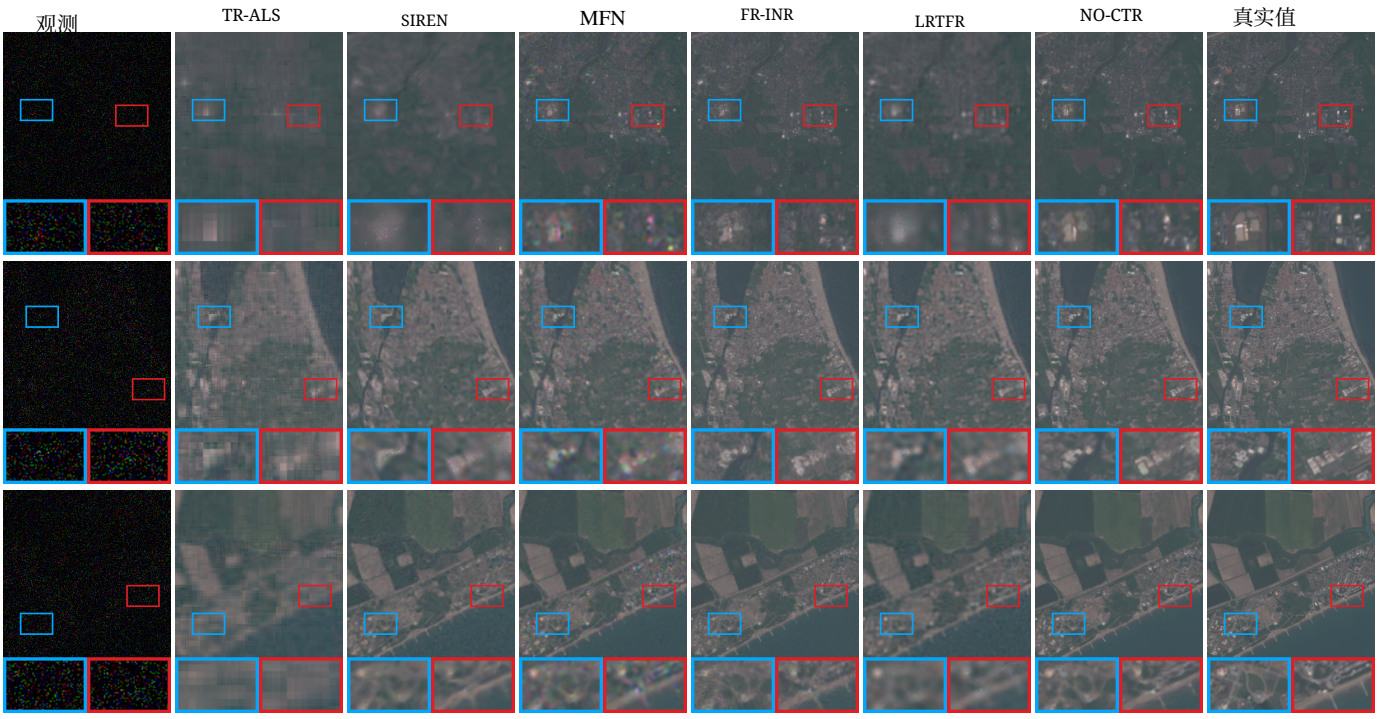


图 6. 采样率为 10% 时哨兵-2号图像补全的视觉恢复结果。

表 IV 点云补全的定量恢复结果

采样率		5%		10%		15%		20%	
点云	方法	归一化均方根误差	R^2	归一化均方根误差	R^2	归一化均方根误差	R^2	归一化均方根误差	R^2
 *Frog*	观测	0.470	-3.120	0.457	-2.894	0.443	-2.662	0.432	-2.484
	SIREN	0.054	0.947	0.044	0.964	0.039	0.972	0.031	0.983
	MFN	0.060	0.936	0.051	0.953	0.047	0.959	0.038	0.974
	FR-INR	0.057	0.940	0.047	0.959	0.043	0.966	0.033	0.980
	LRTFR	0.062	0.930	0.054	0.947	0.050	0.955	0.040	0.971
	NO-CTR	**0.051**	**0.953**	**0.041**	**0.969**	**0.034**	**0.980**	**0.028**	**0.985**
 马里奥	观测	0.435	-1.939	0.423	-1.788	0.412	-1.637	0.398	-1.469
	SIREN	0.088	0.878	0.063	0.938	0.058	0.948	0.044	0.970
	MFN	0.096	0.855	0.069	0.925	0.067	0.931	0.052	0.957
	FR-INR	0.091	0.871	0.064	0.936	0.062	0.940	0.045	0.968
	LRTFR	0.099	0.846	0.077	0.906	0.074	0.915	0.056	0.951
	NO-CTR	**0.083**	**0.892**	**0.061**	**0.942**	**0.055**	**0.952**	**0.043**	**0.971**
 *Rabbit*	观测	0.465	-8.562	0.451	-8.003	0.439	-7.546	0.427	-7.077
	SIREN	0.082	0.709	0.069	0.795	0.056	0.863	0.045	0.914
	MFN	0.088	0.669	0.071	0.781	0.060	0.840	0.048	0.903
	FR-INR	0.095	0.604	0.070	0.783	0.058	0.851	0.046	0.908
	LRTFR	0.088	0.671	0.075	0.756	0.064	0.818	0.054	0.874
	NO-CTR	**0.077**	**0.740**	**0.062**	**0.831**	**0.047**	**0.906**	**0.040**	**0.930**

最佳和次佳结果已突出显示。

所有实验设置。唯一的小例外是：在*T30UYV*上以5%采样率时，FR-INR取得了略高的峰值信噪比。值得注意的是，随着采样率提高，这一优势变得更加明显，表明其在哨兵-2号图像补全方面具有很强的能力。图6展示了采样率为10%时哨兵-2号图像补全的视觉恢复结果。所提出的NO-CTR在恢复地理细节方面表现出更优的能力。在重建卫星图像时，所提出的NO-CTR能够更准确地恢复城市结构和土地覆盖区域的纹理与边界，这对遥感分析至关重要。

D. 点云

最后，我们对网格之外的多维数据开展实验：点云，这是传统张量补全模型几乎难以实现的。点云广泛应用于计算机辅助设计、虚拟现实和自动驾驶等多个领域，用于表示物体或环境的形状或表面特征。每个点云是指三维空间中的一组点，其中每个点包含空间坐标和RGB值。在我们的实验中，每个点云包含约20000个点。我们分别随机选取5%、10%、15%或20%的点作为观测，并根据其余点的坐标恢复其RGB值。

表 IV 列出了点云补全的定量恢复结果。如 NR MSE 最低且在每种实验设置下 R^2 最高所示，所提出的 NO-CTR 方法在所有点云和所有采样率下始终优于所有竞争方法。值得注意的是，即使在低采样率和高采样率下，这种优势依然成立，证明了该方法在点云补全中的有效性。图 7 展示了采样率为 10% 时点云补全的视觉恢复结果。所提出的 NO-CTR 成功重建了这些三维模型的表面细节。由所提出的 NO-CTR 恢复的点云包含更准确的细节，尤其能捕捉其独特特征。

VII. 讨论

A. 连续与非线性模态-n算子的贡献

连续且非线性的mode-n 算子是NO-CTR的核心。为讨论连续且非线性mode-n算子的贡献，在本小节中，我们考虑不同的算子，包括离散且线性的mode-n 算子，以及连续且非线性的mode-n 算子。表V列出了采用不同算子进行多光谱图像补全的定量恢复结果。显然，连续且非线性的mode-n算子在提升NO-CTR的表示能力方面发挥了巨大作用，从而实现了更强的表示能力。与以往采用离散且线性mode-n 算子的方法相比，所提出的采用连续且非线性mode-n算子的NO-CTR展现出显著优势。

表V 不同模态-n算子下多光谱图像补全的定量恢复结果

采样率		10%		20%	
MSI	模态-n 算子	PSNR	SSIM	PSNR	SSIM
花卉	无算子	30.565	0.898	34.367	0.948
	离散与线性	39.175	0.984	43.119	0.984
	连续与非线性	**42.591**	**0.994**	**49.717**	**0.999**
Toy	无算子	26.190	0.835	29.087	0.895
	离散与线性	38.907	0.982	41.395	0.991
	连续与非线性	**42.294**	**0.995**	**48.878**	**0.999**

结果**最佳**的部分已突出显示。



图 7. 采样率为 10% 时点云补全的视觉恢复结果。

B. 神经算子的架构

在NO-CTR中，连续且非线性的mode-n 算子可以借助神经算子优雅地部署，这些神经算子将连续核心张量函数直接映射到连续目标张量函数。为讨论神经算子的影响，在本小节中，我们考虑两种流行的神经算子：傅里叶神经算子（FNO） [37]和深度算子网络（DeepONet） [36]。表 VI列出了采用不同神经算子进行多光谱图像补全的定量恢复结果。神经算子确实提升了NO-CTR的性能，而DeepONet因其灵活的架构取得了更好的性能。

表 VI 不同神经算子下多光谱图像补全的定量恢复结果

采样率		10%		20%	
MSI	架构	PSNR	SSIM	PSNR	SSIM
花卉	无算子	30.565	0.898	34.367	0.948
	FNO	40.940	0.992	47.446	0.997
	DeepONet	**42.591**	**0.994**	**49.717**	**0.999**
Toy	无算子	26.190	0.835	29.087	0.895
	FNO	41.167	0.994	46.076	0.998
	DeepONet	**42.294**	**0.995**	**48.878**	**0.999**

结果**最佳**。

C. 连续核心张量函数的影响 *tion*

连续核心张量函数也是 NO-CTR 的重要组成部分。本小节讨论连续核心张量函数的影响。我们考察了连续核心张量函数的不同表示方法，包括 LRTFR [34], 基于位置编码的多层感知机（简称 PE-MLP） [41], 和 SIREN [40]。表 VII 列出了采用不同连续核心张量函数表示方法进行多光谱图像补全的定量恢复结果。LRTFR 对连续核心张量函数施加低秩约束，这在一定程度上限制了其表示能力。SIREN 在此处表现最佳。

表 VII 连续核心张量函数不同表示方法的多光谱图像补全定量恢复结果

采样率		10%		20%	
MSI	Core	PSNR	SSIM	PSNR	SSIM
花卉	LRTFR	39.586	0.983	44.232	0.992
	PE-MLP	42.086	0.993	47.326	0.998
	SIREN	**42.591**	**0.994**	**49.717**	**0.999**
Toy	LRTFR	37.781	0.980	41.948	0.991
	PE-MLP	40.831	0.993	45.765	0.998
	SIREN	**42.294**	**0.995**	**48.878**	**0.999**

The**最佳**结果以高亮显示。

D. 可学习参数数量

在本小节中，我们比较不同连续张量函数表示方法的可学习参数数量，包括 SIREN [40], LRTFR [34],以及本文提出的 NO-CTR。可学习参数的数量是衡量模型复杂度的关键指标，也与训练成本和速度相关。所提出的 NO-CTR 参数数量最多，因为它包含一个连续核心张量函数，以及一系列基于神经算子的连续非线性模态-n算子。尽管参数更多，NO-CTR 的恢复性能仍显著更优。这表明增加参数是合理且有效的。

表 VIII 不同连续张量函数表示下多光谱图像补全的定量恢复结果与参数数量

采样率		10%		20%	
MSI	方法	峰值信噪比 结构相似性指数	参数	峰值信噪比 结构相似性指数	参数
花卉	SIREN	38.681 0.977	**16 711**	46.041 0.995	**47 671**
	LRTFR	36.879 0.965	130 191	43.249 0.991	331 023
	NO-CTR	**42.591 0.994**	507 341	**49.717 0.999**	472 721
Toy	SIREN	37.259 0.974	**16 711**	45.306 0.995	**47 671**
	LRTFR	37.661 0.975	130 191	43.099 0.991	595 231
	NO-CTR	**42.294 0.995**	597 791	**48.878 0.999**	599 291

结果**最佳**。

E. 超参数敏感性

在本小节中，我们讨论超参数敏感性，重点关注 NO-CTR 的核心：连续非线性模态-n算子。连续非线性模态-n 算子通过 DeepONets[36]优雅地实现。具体而言，对于公式（4）中的每个诱导算子，

$$F_n(v)(y) = \sum_{p=1}^{P_n} b_p^{(n)}\big(v(z_1), v(z_2), \cdots, v(z_{m_n})\big) t_p^{(n)}(y), \quad (6)$$

其中，$v \in C([0,1])$ 是输入函数，$y \in [0,1]$ 是坐标，$\{b_p^{(n)}\}_{p=1}^{P_n}$ 是称为分支网络的标量值函数，$\boldsymbol{t}^{(n)}$ 是称为主干网络的向量值函数，$\{t_p^{(n)}\}_{p=1}^{P_n}$ 是 $\boldsymbol{t}^{(n)}$的 P_n 个分量，$z_i \in [0,1]$ $(i = 1、2、\cdots、m_n)$ 是固定位置，称为传感器。这里我们考察 DeepONet 的两个超参数：传感器数量 m_n 和分支网络数量 P_n。

1）传感器数量：DeepONet 需要在足够多但有限的位置（即公式（6）中的 $\{z_i\}_{i=1}^{m_n}$ ）获取输入函数的值，这些位置称为“传感器”。传感器数量（即公式（6）中的 m_n ）直接决定了 DeepONet 输入的准确度和信息量。图 8 展示了不同传感器数量下的定量恢复结果。随着传感器数量增加，峰值信噪比和结构相似性指数均先大幅上升；随后趋于稳定，仅有轻微波动，这表明适当数量的传感器足以获得良好的恢复性能。

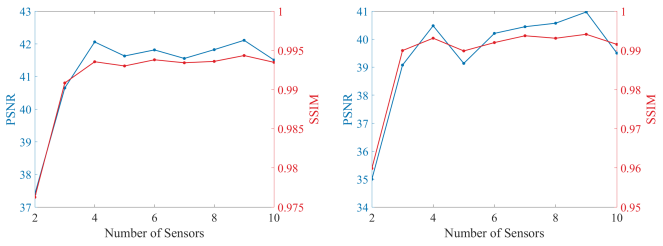


图 8. 采样率为 10% 时，多光谱图像花卉 （左）和玩具 （右）在不同传感器数量下的定量恢复结果。

2）分支网络数量：DeepONet 由一个主干网络和若干分支网络组成。这些分支网络主要用于处理输入函数。根据 [36]，使用大量分支网络效率低下，因此所有分支网络都合并为一个单独的神经网络，可以将其视为所有分支网络共享同一组参数。分支网络数量（即公式（6）中的 P_n ）是该神经网络的输出维度。图 9 展示了不同分支网络数量下的定量恢复结果。结果表明，随着分支数量增加，峰值信噪比和结构相似性指数最初都会上升，并达到一个最优范围。超过这一范围后，进一步增加分支网络会导致轻微波动。

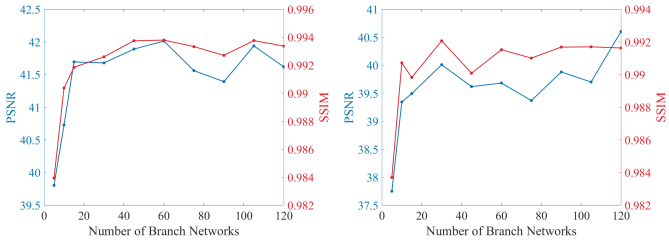


图 9. 采样率为 10% 时，多光谱图像花卉 （左）和玩具 （右）在不同分支网络数量下的定量恢复结果。

VIII. 结论

为充分发挥连续张量函数表示的潜力，本文首先提出了基于神经算子的 mode-n算子，将其作为一种连续且非线性的替代方案。该方案能够对真实世界数据进行真正的连续表示，并可减轻离散化伪影。借助连续且非线性的 mode-n 算子，我们提出了基于神经算子的连续张量函数表示（简称 NO-CTR）。与经典离散张量表示和连续张量函数表示相比，该方法能够更忠实地表示复杂的真实世界数据。从理论上讲，我们还证明了任意连续张量函数都可以由 NO-CTR 近似。为考察 NO-CTR 的能力，我们提出了基于 NO-CTR 的多维数据补全模型。在规则网格上的各类数据（多光谱图像和彩色视频）、不同分辨率网格上的数据（Sentinel-2 图像）以及网格之外的数据（点云）上进行的大量实验，证明了 NO-CTR 的优越性。

参考文献

[1] J. Hou、F. Zhang、H. Qiu、J. Wang、Y. Wang 和 D. Meng，“基于二值测量的稳健低管秩张量恢复”，*IEEE*模式分析与机器智能汇刊，第 44 卷，第 8 期，第 4355–4373 页，2022年。[2] X. Chen 和 L. Sun，“用于多维时间序列预测的贝叶斯时间因子分解”，*IEEE*模式分析与机器智能汇刊，第 44 卷，第 9 期，第 4659–4673 页，2022年。[3] H. Wang、J. Peng、W. Qin、J. Wang 和 D. Meng，“融合低秩性与平滑性的保证张量恢复”，*IEEE*模式分析与机器智能汇刊，第 45 卷，第 9 期，第 10990–11007 页，2023年。[4] C. He、Y. Xu、Z. Wu、S. Zheng 和 Z. Wei，“多维视觉数据修复：揭示变换后高阶张量奇异值中的全局差异”，*IEEE*图像处理汇刊，第 33 卷，第 6409–6424 页，2024年。

[5] W. He、Q. Yao、C. Li、N. Yokoya、Q. Zhao、H. Zhang 和 L. Zhang，“非局部与全局相结合：一种用于高光谱图像修复的迭代范式”，IEEE模式分析与机器智能汇刊，第 44 卷，第 4 期，第 2089–2107 页，2022年。[6] T.-H. Zhang、J.-L. Zhao、S. Fang、Z. Li 和 M.-G. Gong，“用于高光谱图像修复的全模态增强张量列车秩最小化”，IEEE地球科学与遥感汇刊，第 62 卷，第 1–13 页，2024年。[7] S. Li、Z. Zhou、M. Zhao、J. Yang、W. Guo、Y. Lv、L. Kou、H. Wang 和 Y. Gu，“卫星视频多任务基准数据集：目标检测、跟踪与分割”，IEEE地球科学与遥感汇刊，第 61 卷，第 1–21 页，2023年。[8] K. Wang、M. Zhou、Q. Lin、G. Niu 和 X. Zhang，“用于三维目标检测的几何引导点生成”，IEEE信号处理快报，第 32 卷，第 136–140 页，2025年。[9] B.-Z. Fu、T.-Z. Huang、X.-L. Zhao、W.-H. Wu 和 H. Peng，“ProTD：用于高光谱异常检测的提示式张量分解”，IEEE地球科学与遥感汇刊，第 63 卷，第 1–13 页，2025年。[10] X. Zhang 和 M. K. Ng，“基于泊松观测的低秩张量补全”，IEEE模式分析与机器智能汇刊，第 44 卷，第 8 期，第 4239–4251 页，2022年。[11] J. Jiang、Y. Xu、H. Xu、G. Shen 和 J. Zheng，“通过加权混合图拉普拉斯实现多维视觉数据补全”，信号处理，第 216 卷，文章编号 109305，2024年。[12] R. Su、X.-L. Zhao、W.-H. Wu、S. Liu 和 J. He，“用于多维信号恢复的深度全连接张量网络分解”，信号处理，第 233 卷，文章编号 109903，2025年。[13] M. M. Salut 和 D. V. Anderson，“用于含噪高光谱图像分类的随机张量鲁棒主成分分析”，IEEE地球科学与遥感快报，第 20 卷，第 1–5 页，2023年。[14] R. Zhang、Z. Rong 和 Z. Dong，“CAMGnet：通过多维跨域特征动态融合处理单变量连续时间序列信号的自学习分类模型”，神经计算，第 657 卷，文章编号 131686，2025年。[15] C. Zhu、G. Lu、B. He、R. Xie 和 L. Song，“用于多视图视频压缩的隐式-显式集成表示”，IEEE图像处理汇刊，第 34 卷，第 1106–1118 页，2025年。[16] C. Shi、K. Shi、Z. Zeng 和 F. Zhu，“由多维协同边缘信息引导的动态特征增强网络，用于遥感图像压缩”，知识型系统，第 310 卷，文章编号 112996，2025年。[17] Q. Xie、Q. Zhao、D. Meng 和 Z. Xu，“基于克罗内克基表示的张量稀疏性及其在张量恢复中的应用”，IEEE模式分析与机器智能汇刊，第 40 卷，第 8 期，第 1888–1902 页，2018年。[18] F. Zhang、J. Wang、W. Wang 和 C. Xu，“利用先验子空间信息进行低管秩加稀疏张量恢复”，IEEE模式分析与机器智能汇刊，第 43 卷，第 10 期，第 3492–3507 页，2021年。[19] J.-Y. Xie、H. Zhang、X.-L. Zhao 和 Y.-S. Luo，“IRTF：一种用于不规则多维数据恢复的新型张量因子分解”，知识型系统，第 330 卷，文章编号 114372，2025年。[20] L. Xu、L. Cheng、N. Wong 和 Y.-C. Wu，“是否折叠：用于视觉数据补全的图正则化张量列车”，IEEE模式分析与机器智能汇刊，第 1–18 页，2025年。[21] E. J. Cand`es 和 B. Recht，“通过凸优化实现精确矩阵补全”，计算数学基础，第 9 卷，第 717–772 页，2009年。[22] J.-F. Cai、E. J. Cand`es 和 Z. Shen，“用于矩阵补全的奇异值阈值算法”，SIAM优化学报，第 20 卷，第 4 期，第 1956–1982 页，2010年。[23] M. Fornasier、H. Rauhut 和 R. Ward，“通过迭代重加权最小二乘最小化实现低秩矩阵恢复”，SIAM优化学报，第 21 卷，第 4 期，第 1614–1640 页，2011年。[24] J. D. Carroll 和 J.-J. Chang，“通过 n 路推广的“Eckart-Young”分解分析多维尺度分析中的个体差异”，心理计量学，第 35 卷，第 283–319 页，1970年。[25] R. A. Harshman，“PARAFAC过程的基础：用于“解释性”多模态因子分析的模型与条件”，加州大学洛杉矶分校语音学工作论文，第 16 卷，第 1–84 页，1970年。[26] L. R. Tucker，“关于三模态因子分析的一些数学注记”，心理计量学，第 31 卷，第 279–311 页，1966年。[27] M. E. Kilmer、C. D. Martin 和 L. Perrone，“矩阵奇异值分解的三阶推广：表示为三阶张量的乘积”，塔夫茨大学计算机科学系，美国马萨诸塞州梅德福，技术报告 TR-2008-4，2008年。

[28] M. E. Kilmer和C. D. Martin，《三阶张量的分解策略》，《线性代数及其应用》，第435卷，第3期，第641–658页，2011年，特刊：献给 Pete Stewart，以庆祝其70岁生日。[29] M. E. Kilmer、K. Braman、N. Hao和R. C. Hoover，《作为矩阵算子的三阶张量：一个用于成像的理论与计算框架》，《SIAM矩阵分析与应用杂志》，第34卷，第1期，第148–172页，2013年。[30] I. V. Oseledets，《张量列车分解》，《SIAM科学计算杂志》，第33卷，第5期，第2295–2317页，2011年。[31] Q. Zhao、G. Zhou、S. Xie、L. Zhang和A. Cichocki，《张量环分解》，2016年。 [在线]。网址：https://arxiv.org/abs/1606. 05535[32] M. Imaizumi和K. Hayashi，《具有平滑性的张量分解》，载于《第34届国际机器学习会议论文集》，《机器学习研究论文集》系列，D. Precup和Y. W. Teh编，第70卷。PMLR，2017年8月6–11日，第1597–1606页。[33] S. Fang、X. Yu、Z. Wang、S. Li、M. Kirby和S. Zhe，《连续指标张量数据的函数型贝叶斯塔克分解》，载于《第十二届国际学习表征会议》，2024年。[34] Y. Luo、X. Zhao、Z. Li、M. K. Ng和 D. Meng，《多维数据恢复的低秩张量函数表示》，《IEEE模式分析与机器智能汇刊》，第46卷，第5期，第3351–3369页，2024年。[35] A. Anandkumar、K. Azizzadenesheli、K. Bhattacharya、N. Kovachki、Z. Li、B. Liu和A. Stuart，《神经算子：用于偏微分方程的图核网络》，载于2020年ICLR深度神经模型与微分方程融合研讨会，2019年。[36] L. Lu、P. Jin、G. Pang、Z. Zhang和G. E. Karniadakis，《基于算子通用逼近定理，通过DeepONet学习非线性算子》，《自然·机器智能》，第3卷，229，2021年218页。–。[37] Z. Li、N. B. Kovachki、K. Azizzadenesheli、B. Liu、K. Bhattacharya、A. Stuart和A. Anandkumar，《用于参数化偏微分方程的傅里叶神经算子》，载于《国际学习表征会议》，2021年。[38] J. J. Park、P. Florence、J. Straub、R. Newcombe和S. Lovegrove，《DeepSDF：用于形状表示的连续有符号距离函数学习》，载于2019年IEEE/CVF计算机视觉与模式识别会议（CVPR），2019年，第165–174页。[39] L. Mescheder、M. Oechsle、M. Niemeyer、S. Nowozin和A. Geiger，《占用网络：在函数空间中学习三维重建》，载于2019年IEEE/CVF计算机视觉与模式识别会议（CVPR），2019年，第4455–4465页。[40] V. Sitzmann、J. Martel、A. Bergman、D. Lindell和G. Wetzstein，《具有周期激活函数的隐式神经表示》，载于《神经信息处理系统进展》，H. Larochelle、M. Ranzato、R. Hadsell、M. Balcan和H. Lin编，第33卷。Curran Associates, Inc.，2020年，第7462–7473页。[41] M. Tancik、P. Srinivasan、B. Mildenhall、S. Fridovich-Keil、N. Raghavan、U. Singhal、R. Ramamoorthi、J. Barron和R. Ng，《傅里叶特征使网络能够学习低维域中的高频函数》，载于《神经信息处理系统进展》，H. Larochelle、M. Ranzato、R. Hadsell、M. Balcan和H. Lin编，第33卷。Curran Associates, Inc.，2020年，第7537–7547页。[42] V. Saragadam、D. LeJeune、J. Tan、G. Balakrishnan、A. Veeraraghavan和R. G. Baraniuk，《WIRE：小波隐式神经表示》，载于2023年IEEE/CVF计算机视觉与模式识别会议（CVPR），2023年，第18 507–18 516页。[43] K. Shi、X. Zhou和S. Gu，《通过傅里叶重参数化训练改进隐式神经表示》，载于2024年IEEE/CVF计算机视觉与模式识别会议（CVPR），2024年，第25 985–25 994页。[44] R. Fathony、A. K. Sahu、D. Willmott和J. Z. Kolter，《乘性滤波器网络》，载于《国际学习表征会议》，2021年。[45] Y. Chen、S. Liu和X. Wang，《利用局部隐式图像函数学习连续图像表示》，载于2021年IEEE/CVF计算机视觉与模式识别会议（CVPR），2021年，第8624–8634页。[46] G. Y¨uce、G. Ortiz-Jim´enez、B. Besbinar和P. Frossard，《隐式神经表示的结构化字典视角》，载于2022年IEEE/CVF计算机视觉与模式识别会议（CVPR），2022年，第19 206–19 216页。[47] V. Saragadam、J. Tan、G. Balakrishnan、R. G. Baraniuk和A. Veer-araghavan，“Miner：多尺度隐式神经表示，”in计算机

视觉 – *ECCV 2022*, S. Avidan, G. Brostow, M. Ciss´e, G. M. Farinella, and T. Hassner, 编. Cham: Springer Nature S witzerland, 2022, pp. 318–333. [48] J. Li, X. Zhao, J. Wang, C. Wang, and M. Wang, “超像素引导的隐式神经表示用于多维数据,” in 计算机视觉 – *ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, 编. Cham: Springer Nature Switzerland, 2025, pp. 258–276. [49] T.-W. Zhou, X.-L. Zhao, W.-H. Wu, J.-L. Wang, and Y.-S. Luo, “频率感知隐式神经表示用于多维数据恢复,” 《*IEEE*视频技术电路与系统汇刊》, pp. 1–1, 2025. [50] N. Boull´e and A. Townsend, “算子学习数学指南,” in 数值分析与机器学习相遇, ser. 《数值分析手册》, S. Mishra and A. Townsend, 编. Elsevier, 2024, vol. 25, pp. 83–125. [51] N. B. Kovachki, S. Lanthaler, and A. M. Stuart, “算子学习：算法与分析,” in 数值分析与机器学习相遇, ser. 《数值分析手册》, S. Mishra and A. Townsend, 编. Elsevier, 2024, vol. 25, pp. 419–467. [52] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stu- art, and A. Anandkumar, “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》, vol. 24, no. 89, pp. 1–97, 2023. [53] S. K. Vemuri, T. B¨uchner, J. Niebling, and J. Denzler, “用于物理信息神经网络的函数张量分解,” in 模式识别, A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, and U. Pal, 编. Cham: Springer Nature Switzerland, 2025, pp. 32–46. [54] T. G. Kolda and B. W. Bader, “张量分解及其应用,” 《*SIAM*评论》, vol. 51, no. 3, pp. 455–500, 2009. [55] W. Wang, V. Aggarwal, and S. Aeron, “高效低秩张量环补全,” in *2017 IEEE*国际计算机视觉会议（*ICCV*）, 2017, pp. 5698–5706. [56] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “图像质量评估：从误差可见性到结构相似性,” *IEEE*图像处理汇刊, vol. 13, no. 4, pp. 600–612, 2004. [57] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar, “广义混合像素相机：拍摄后控制分辨率、动态范围和光谱,” 技术报告，2008年11月. [58] O. Calin, 通用逼近器. Cham: 施普林格国际出版, 2020, pp. 251–284. [59] T. Chen and H. Chen, “任意激活函数的神经网络对非线性算子的通用逼近及其在动力系统中的应用,” 《*IEEE*神经网络汇刊》, vol. 6, no. 4, pp. 911–917, 1995.

定理 1 的证明（附录）

引理 *1*（*[58]*）：设 σ 是任意连续判别函数。那么，形如以下形式的有限和

$$\sum_{k=1}^{K} a_k \sigma(\boldsymbol{w}_k^T \boldsymbol{y} + b_k), \quad \boldsymbol{w}_k \in \mathbb{R}^N,\ a_k, b_k \in \mathbb{R}$$

在 $C([0,1]^N)$ 中稠密。

引理 *2*（*[36], [59]*）：假设 W 是一个巴拿赫空间， $K_1 \subseteq W$ 和 $K_2 \subseteq \mathbb{R}^N$ 分别是 W 和 $\mathbb{R}^N$ 中的两个紧集， V 是 $C(K_1)$ 中的一个紧集， F 是一个连续算子，将 V 映射到 $C(K_2)$。那么，对任意 $\varepsilon > 0$，存在正整数 m 和 P，以及由深度神经网络实现的连续函数 $b_p \in C(\mathbb{R}^m)$ 和 $t_p \in C(K_2)$， $p = 1,\ 2,\ \cdots,\ P$， $\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots \boldsymbol{x}_m \in K_1$，使得

$$\left| F(v)(\boldsymbol{y}) - \sum_{p=1}^{P} b_p(v(\boldsymbol{x}_1), v(\boldsymbol{x}_2), \cdots, v(\boldsymbol{x}_m)) t_p(\boldsymbol{y}) \right| < \varepsilon$$

对所有 $v \in V$ 和 $\boldsymbol{y} \in K_2$ 均成立。

定理 *1* 的证明：根据引理 1，存在一个两层全连接神经网络

$$\mathbf{G}\colon [0,1]^N \to \mathbb{R},\ \boldsymbol{y} \mapsto \sum_{k=1}^{K} a_k \sigma(\boldsymbol{w}_k^T \boldsymbol{y} + b_k)$$

使得

$$\|\mathbf{G} - \mathbf{X}\| < \frac{\varepsilon}{2}. \tag{7}$$

注意到$\mathbf{X}$是定义在紧集 $[0,1]^N$上的连续函数，因此$\mathbf{X}$有界，且$\mathbf{G}$也有界。假设

$$\|\mathbf{G}\| \leqslant M. \tag{8}$$

根据引理 2，存在由DeepONet实现的 N 个神经算子

$$F_n\colon C([0,1]) \to C([0,1]),\ u(\cdot) \mapsto \sum_{p=1}^{P_n} b_p^{(n)}(u(x_1), u(x_2), \cdots, u(x_{m_n})) t_p^{(n)}(\cdot),$$

使得

$$|F_n(u)(y) - I(u)(y)| < \frac{\varepsilon}{(2^{N+1}-2)M}$$

对所有 $u \in C([0,1])$和 $y \in [0,1]$成立，因此

$$\begin{aligned}
\|F_n - I\| &= \sup_{\|u\|=1} \|(F_n - I)(u)\| \\
&= \sup_{\|u\|=1} \|F_n(u) - I(u)\| \\
&= \sup_{\|u\|=1} \sup_{y\in[0,1]} |(F_n(u) - I(u))(y)| \\
&= \sup_{\|u\|=1} \sup_{y\in[0,1]} |F_n(u)(y) - I(u)(y)| \\
&\leqslant \frac{\varepsilon}{(2^{N+1}-2)M},
\end{aligned}$$

其中， I表示恒等变换， $n = 1,\ 2,\ \cdots,\ N$。因此，

$$\begin{aligned}
\|\mathscr{F}_n^{\langle n\rangle} - \mathscr{I}^{\langle n\rangle}\| &= \sup_{\|\mathbf{X}\|=1} \|(\mathscr{F}_n^{\langle n\rangle} - \mathscr{I}^{\langle n\rangle})(\mathbf{X})\| \\
&= \sup_{\|\mathbf{X}\|=1} \sup_{\boldsymbol{y}\in[0,1]^N} |(\mathscr{F}_n^{\langle n\rangle} - \mathscr{I}^{\langle n\rangle})(\mathbf{X})(\boldsymbol{y})| \\
&= \sup_{\|\mathbf{X}\|=1} \sup_{\boldsymbol{y}\in[0,1]^N} |(F_n - I)(\mathbf{X}(y_1, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N))(y_n)| \\
&= \sup_{\|\mathbf{X}\|=1} \sup_{\bar{\boldsymbol{y}}\in[0,1]^{N-1}} \sup_{y_n\in[0,1]} |(F_n - I)(\mathbf{X}(y_1, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N))(y_n)| \\
&= \sup_{\|\mathbf{X}\|=1} \sup_{\bar{\boldsymbol{y}}\in[0,1]^{N-1}} \|(F_n - I)(\mathbf{X}(y_1, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N))\|
\end{aligned}$$

$$
\begin{aligned}
&\leqslant \sup_{\|\mathbf{X}\|=1} \sup_{\bar{\boldsymbol{y}}\in[0,1]^{N-1}} \|F_n - I\| \|\mathbf{X}(y_1, \cdots, y_{n-1}, \cdot, y_{n+1}, \cdots, y_N)\| \\
&= \|F_n - I\| \sup_{\|\mathbf{X}\|=1} \sup_{\bar{\boldsymbol{y}}\in[0,1]^{N-1}} \sup_{y_n\in[0,1]} |\mathbf{X}(y_1, \cdots, y_{n-1}, y_n, y_{n+1}, \cdots, y_N)| \\
&= \|F_n - I\| \sup_{\|\mathbf{X}\|=1} \|\mathbf{X}\| \\
&= \|F_n - I\| \\
&\leqslant \frac{\varepsilon}{(2^{N+1}-2)M},
\end{aligned} \tag{9}
$$

其中，$\boldsymbol{y} = (y_1,\ y_2,\ \cdots,\ y_N) \in [0,1]^N$，$\bar{\boldsymbol{y}} = (y_1,\ \cdots,\ y_{n-1},\ y_{n+1},\ \cdots,\ y_N) \in [0,1]^{N-1}$，$\mathscr{F}_n^{\langle n\rangle}$和 $\mathscr{I}^{\langle n\rangle}$ 分别是由 F_n 和 I诱导的连续非线性模态-n 算子。不失一般性，我们假设

$$
\|\mathscr{F}_n^{\langle n\rangle}\| \leqslant 2. \tag{10}
$$

结合式 (7)、(8)、(9) 和 (10)，我们得到

$$
\begin{aligned}
&\|\mathbf{X} - \mathscr{F}_N^{\langle N\rangle} \circ \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| \\
&\leqslant \|\mathscr{I}^{\langle N\rangle} \circ \mathscr{I}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{I}^{\langle 1\rangle}(\mathbf{X}) - \mathscr{I}^{\langle N\rangle} \circ \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| \\
&\quad + \|\mathscr{I}^{\langle N\rangle} \circ \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G}) - \mathscr{F}_N^{\langle N\rangle} \circ \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| \\
&= \|\mathscr{I}^{\langle N\rangle}(\mathscr{I}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{I}^{\langle 1\rangle}(\mathbf{X}) - \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G}))\| + \|(\mathscr{I}^{\langle N\rangle} - \mathscr{F}_N^{\langle N\rangle})(\mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G}))\| \\
&\leqslant \|\mathscr{I}^{\langle N\rangle}\| \|\mathscr{I}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{I}^{\langle 1\rangle}(\mathbf{X}) - \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| + \|\mathscr{I}^{\langle N\rangle} - \mathscr{F}_N^{\langle N\rangle}\| \|\mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| \\
&\leqslant \|\mathscr{I}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{I}^{\langle 1\rangle}(\mathbf{X}) - \mathscr{F}_{N-1}^{\langle N-1\rangle} \circ \cdots \circ \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| + \|\mathscr{I}^{\langle N\rangle} - \mathscr{F}_N^{\langle N\rangle}\| \|\mathscr{F}_{N-1}^{\langle N-1\rangle}\| \cdots \|\mathscr{F}_1^{\langle 1\rangle}\| \|\mathbf{G}\| \\
&\leqslant \cdots\cdots \\
&\leqslant \|\mathscr{I}^{\langle 1\rangle}(\mathbf{X}) - \mathscr{F}_1^{\langle 1\rangle}(\mathbf{G})\| + \sum_{k=1}^{N-1} \left(\|\mathscr{I}^{\langle k+1\rangle} - \mathscr{F}_{k+1}^{\langle k+1\rangle}\| \|\mathbf{G}\| \prod_{j=1}^{k} \|\mathscr{F}_j^{\langle j\rangle}\| \right) \\
&\leqslant \|\mathbf{X} - \mathbf{G}\| + \|\mathscr{I}^{\langle 1\rangle} - \mathscr{F}_1^{\langle 1\rangle}\| \|\mathbf{G}\| + \sum_{k=1}^{N-1} \left(\|\mathscr{I}^{\langle k+1\rangle} - \mathscr{F}_{k+1}^{\langle k+1\rangle}\| \|\mathbf{G}\| \prod_{j=1}^{k} \|\mathscr{F}_j^{\langle j\rangle}\| \right) \\
&< \frac{\varepsilon}{2} + \frac{\varepsilon}{(2^{N+1}-2)M} M + \sum_{k=1}^{N-1} \left(\frac{\varepsilon}{(2^{N+1}-2)M} M \cdot 2^k \right) \\
&= \varepsilon.
\end{aligned}
$$

□