分 类 号: O241.82　　**单位代码**: 10183
研究生学号: 2023311025　　**密　　级**: 公　开

吉 林 大 学

博士学位论文

神经常微分方程的逼近理论
及其在算子学习中的应用

Neural Ordinary Differential Equations: Approximation Theory and Its Applications in Operator Learning

作者姓名: 李兹谦
专　　业: 运筹学与控制论
研究方向: 深度学习与科学计算
指导教师: 汤涛　教授
培养单位: 吉林大学数学学院

2026 年 05 月

神经常微分方程的逼近理论
及其在算子学习中的应用

Neural Ordinary Differential Equations: Approximation Theory and Its Applications in Operator Learning

作者姓名: 李兹谦

专业名称: 运筹学与控制论

指导教师: 汤涛 教授

学位类别: 理学博士

论文答辩日期: 2026 年 5 月 18 日

授予学位日期: 年 月 日

答辩委员会组成:

	姓名	职称	工作单位
主席	徐英祥	教授	东北师范大学
委员	邹永魁	教授	吉林大学
	李正光	教授	吉林大学
	贾继伟	教授	吉林大学
	杨 茂	教授	东北电力大学
	王瑞姝	副教授	吉林大学

吉林大学博士学位论文原创性声明

本人郑重声明: 所呈交的博士学位论文, 是本人在指导教师的指导下, 独立进行研究工作所取得的成果. 除文中已经注明引用的内容外, 本论文不包含任何其他个人或集体已经发表或撰写过的作品成果. 对本文的研究做出重要贡献的个人和集体, 均已在文中以明确方式标明. 本人完全意识到本声明的法律结果由本人承担.

学位论文作者签名: 李兹谦

日期: 2026 年 5 月 18 日

关于学位论文使用授权的声明

本人完全了解吉林大学有关保留、使用学位论文的规定，同意吉林大学保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权吉林大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文.

(保密论文在解密后应遵守此规定)

论文级别：□ 硕士 ■ 博士

学科专业：运筹学与控制论

论文题目：神经常微分方程的逼近理论及其在算子学习中的应用

作者签名：李兹谦 指导教师签名：

2026 年 05 月 18 日

作者联系地址(邮编)：长春市吉林大学前卫南区数学研究所(130012)

作者联系电话：+86 13855141583

指导教师对学位论文/实践成果的学术评语

<table>
<tr><td>学位论文/实践成果题目：</td><td colspan="5">神经常微分方程的逼近理论及其在算子学习中的应用</td></tr>
<tr><td>作者姓名：</td><td colspan="2">李兹谦</td><td colspan="2">专业：</td><td>运筹学与控制论</td></tr>
<tr><td>导师姓名：</td><td>汤涛</td><td>职称：</td><td>教授</td><td>所在单位：</td><td>数学学院</td></tr>
<tr><td colspan="6">对学位论文/实践成果的工作过程介绍和学术评语：

李兹谦同学在攻读博士学位期间，认真钻研本学科及相关交叉领域的基础理论与研究方法，在本人指导下独立完成了博士学位论文的各项研究工作。论文选题具有较强的理论意义和应用价值，研究过程安排合理，技术路线清晰，工作开展扎实，体现了较好的科研素养和独立研究能力。
该论文围绕神经常微分方程的逼近理论及其在算子学习中的应用展开研究，针对相关模型在复杂度控制、动力系统逼近及偏微分方程算子学习中的若干关键问题，进行了较为系统的理论分析与数值研究。论文提出的相关方法具有较强创新性，研究结果表明所构建模型在理论分析和算法应用方面均取得了较好的效果。全文结构完整，层次清楚，论证较严谨，文字表达规范，达到博士学位论文的要求。
总体来看，该生已具备较扎实的理论基础和较强的科研能力。本人认为该论文已达到理学博士学位论文水平，同意提交评审，并建议参加博士学位论文答辩。

签名：
2026 年 3 月 31 日</td></tr>
</table>

指导小组对学位论文/实践成果的学术评语

学位论文/实践成果题目：	神经常微分方程的逼近理论及其在算子学习中的应用		
申请人姓名：	李兹谦	专业：	运筹学与控制论

小组成员	职称	所在单位
汤涛	教授	吉林大学
张然	教授	吉林大学
张与彪	教授	吉林大学
贾继伟	教授	吉林大学

对学位论文/实践成果的工作过程介绍和学术评语:

一、 论文工作过程介绍

该生在攻读博士学位期间治学严谨，刻苦钻研，尊师重道。在论文工作期间，该生全身心投入科研，迎难而上。论文不仅展现了该生在逼近论、微分方程数值解等计算数学领域坚实宽广的理论功底，更体现了其在科学计算与机器学习交叉领域极强的创新与工程实践能力。整个论文工作过程规范，学术态度端正，逻辑严密，工作量饱满。指导小组认为，该生已系统深入地掌握了本学科的专门知识，具备了独立从事高水平学术研究工作的能力。

二、 论文的学术评语

该博士学位论文面向数据驱动的复杂动力系统建模与偏微分方程高效求解等重大前沿难题，针对现有神经常微分方程（NODEs）及算子学习模型（如DeepONets）中存在的参数复杂度高、缺乏物理先验、难以进行长期时间外推等痛点，开展了系统且深入的创新性研究。论文取得了具有国际先进水平的学术成果，主要贡献如下：

- 提出了半自治神经常微分方程（SA-NODEs）并建立了完善的逼近理论： 针对经典NODEs参数冗余的问题，创新性地提出了参数不随时间变化的SA-NODEs模型。运用Pinkus定理与自举论证，严格证明了其对动力系统的定性万能逼近性质；并在Barron空间下推导出了逼近误差的严格定量收敛速率 $O(1/P)$。进一步，利用前推算子将理论推广至传输方程，首次在Wasserstein-1距离下证明了一致收敛速率，在渐近意义下为神经网络克服维数灾难提供了坚实的数学保障。
- 构建了面向偏微分方程算子学习的NODE-ONet框架： 首次提出将物理编码的神经常微分方程（Physics-encoded NODEs）作为动力学演化代理，完美融入“编码器-解码器”算子学习架构中。论文在理论层面为一般的编码器-解码器网络建立了统一的误差分析框架；在方法层面，通过时空变量的解耦及底层偏微分方程物理先验的注入，大幅降低了模型复杂度，并显著提升了模型超越训练时间域的外推预测与多输入泛化能力。
- 开展了详实且极具说服力的大规模数值实验： 将所提算法框架成功应用于常微分系统、传输方程以及复杂偏微分方程。实验充分证明，SA-NODEs与NODE-ONet在降低模型自由度、提升计算效率、增强时间外推稳定性方面均显著优于经典的NODEs与DeepONets等主流基准模型。

该学位论文立意新颖，文献综述详实，数学推导严密无误，算法实验丰富且极具代表性，行文流畅，格式规范。指导小组一致认为，这是一篇具有极高学术水平的优秀博士学位论文，表明李兹谦同学已完全达到了教育部关于理学博士学位的水平要求。指导小组一致同意该生通过学位论文内部审查，建议提交盲审及答辩。

张与彪 汤涛 贾继伟

签名：

2026 年 4 月 10 日

答辩决议书

<table>
<tr><td>论文题目：</td><td colspan="5">神经常微分方程的逼近理论及其在算子学习中的应用</td></tr>
<tr><td>作者姓名：</td><td>李兹谦</td><td>专业：</td><td>运筹学与控制论</td><td>学院：</td><td>数学学院</td></tr>
<tr><td>答辩委员会</td><td>姓名</td><td colspan="2">职称</td><td colspan="2">工作单位</td><td>是否博导</td></tr>
<tr><td>主席</td><td>徐英祥</td><td colspan="2">教授</td><td colspan="2">东北师范大学</td><td>是</td></tr>
<tr><td rowspan="5">委员</td><td>邹永魁</td><td colspan="2">教授</td><td colspan="2">吉林大学</td><td>是</td></tr>
<tr><td>李正光</td><td colspan="2">教授</td><td colspan="2">吉林大学</td><td>是</td></tr>
<tr><td>贾继伟</td><td colspan="2">教授</td><td colspan="2">吉林大学</td><td>是</td></tr>
<tr><td>杨茂</td><td colspan="2">教授</td><td colspan="2">东北电力大学</td><td>是</td></tr>
<tr><td>王瑞姝</td><td colspan="2">副教授</td><td colspan="2">吉林大学</td><td>是</td></tr>
</table>

答辩委员会对论文及答辩情况的评语：

李兹谦同学的博士学位论文主要面向数据驱动的动力系统建模与偏微分方程高效求解问题，针对现有神经常微分方程及算子学习模型中存在的参数复杂度高、缺乏物理先验、难以长期外推等关键问题，开展了系统且深入的创新性研究。选题属国际前沿。论文的主要内容包括：

(1) 提出了半自治神经常微分方程 (SA-NODEs)，并建立了其完整的逼近理论。该模型采用不随时间变化的常值参数，对动力系统在长时间区间上进行逼近；进一步将该结果推广至传输方程，在 Wasserstein-1 距离下给出一致收敛估计。

(2)将偏微分方程的内在结构 (如双线性耦合、非线性反应、加性源项等) 显式融入网络架构，提出了基于神经常微分方程的偏微分方程算子学习框架(NODE-ONet)。

论文写作规范，结构合理，层次分明，理论分析深入，数据详实，结论正确。论文工作表明，李兹谦同学已掌握了本学科坚实宽广的基础理论和系统深入的专门知识，具有独立从事科学研究的能力。

在答辩过程中，李兹谦同学能够准确回答评委提出的问题。经答辩委员会讨论，认为该论文达到博士学位论文标准，是一篇优秀的博士学位论文，建议授予理学博士学位。

注:本页编入学位论文。

摘要

偏微分方程与动力系统是刻画复杂物理与工程现象的基础数学工具. 深度学习在逼近非线性动力系统以及构建偏微分方程代理模型方面展现出巨大的应用潜力. 然而, 现有方法在控制模型复杂度, 捕获长期时间动态以及融合底层物理规律等方面仍面临挑战. 针对上述问题, 本文围绕神经常微分方程 (Neural Ordinary Differential Equations, NODEs) 展开研究, 从基础的动力系统非线性逼近理论到面向偏微分方程的算子学习算法进行了系统的探讨. 本文的主要研究内容与贡献可归纳为以下两个部分:

针对动力系统的逼近问题, 本文提出了一种参数量相较于 NODEs 较少的半自治神经常微分方程 (Semi-Autonumous NODEs) 框架. 在理论层面, 系统研究了 SA-NODEs 针对动力系统的逼近性质. 在有限时间区间与一般性假设下, 证明了当网络宽度趋于无穷大时, 模型收敛于原动力系统. 进一步地, 在附加的正则性条件下, 借助 Barron 空间的定量逼近理论, 明确了逼近误差随网络宽度变化的严格收敛速率. 基于该常微分方程的逼近结果, 本文推导并证明了利用神经传输方程逼近真实传输方程的理论收敛率. 数值实验部分涵盖了多类常微分方程系统与传输方程, 验证了 SA-NODEs 在捕获动力学演化特征方面的有效性. 对比结果表明, SA-NODEs 在降低模型架构复杂度的同时, 具备良好的数值表现.

针对偏微分方程的算子学习问题, 本文提出了一种深度神经常微分方程算子网络 (NODE Operator Network, NODE-ONet) 框架. 现有算子学习方法通常缺乏对偏微分方程固有领域知识的利用, 导致其在刻画时间动态规律及超越训练时间域的外推泛化方面存在局限. NODE-ONet 框架采用编码器-解码器架构以缓解上述问题, 具体包含三个核心组件: 对输入函数进行空间离散化的编码器, 捕获潜在时间动态的神经常微分方程, 以及在物理空间内重建数值解的解码器. 在

理论方面, 本文对一般的编码器-解码器架构进行了统一的误差分析. 在算法设计方面, 提出了物理编码的神经常微分方程, 将底层偏微分方程特定的数学结构融入网络中. 该设计在相较于 DeepONet 算法降低模型复杂度的同时, 改善了计算效率与泛化能力. 在一维非线性反应-扩散方程与二维 Navier-Stokes 方程上的数值模拟表明, 该方法具备较高的数值精度与超越训练时间框架的预测能力. 此外, 框架支持灵活接入多类编码器与解码器, 并在结构相关的偏微分方程族中具备泛化能力, 提供了一种可扩展的物理编码计算工具.

综上所述, 本文在理论层面建立了半自治神经常微分方程的逼近性质与收敛性估计, 在应用层面构建了融合物理结构先验的高效算子学习框架. 相关研究为复杂动力系统的降维建模与偏微分方程的高效数值求解提供了理论支撑与算法参考.

关键词: 神经常微分方程, 算子学习, 误差分析, 机器学习, 科学计算.

Abstract

Partial differential equations (PDEs) and dynamical systems are fundamental mathematical tools for characterizing complex physical and engineering phenomena. Deep learning has demonstrated significant potential in approximating nonlinear dynamical systems and constructing surrogate models for PDEs. However, existing methods still face challenges in controlling model complexity, capturing long-term temporal dynamics, and incorporating underlying physical laws. To address these issues, this thesis centers on neural ordinary differential equations (NODEs), conducting a systematic investigation ranging from the foundational nonlinear approximation theory of dynamical systems to operator learning algorithms for PDEs. The main research contents and contributions of this thesis can be summarized into the following two parts:

Targeting the approximation of dynamical systems, this thesis proposes a semi-autonomous neural ordinary differential equation (SA-NODE) framework, which employs fewer parameters compared to vanilla NODEs. Theoretically, we systematically investigate the approximation properties of SA-NODEs for dynamical systems. Under a finite-time horizon and general assumptions, we establish an asymptotic approximation result, demonstrating that the approximation error vanishes as the width of the neural network approaches infinity. Furthermore, under additional regularity conditions, we specify the strict convergence rate of the approximation error with respect to the width of the neural network by utilizing quantitative approximation results in the Barron space. Based on this ODE approximation result, we derive and prove the theoretical convergence rate for approximating true transport equations using their neural counterparts. The numerical experiments cover various ODE systems and transport equations, validat-

ing the effectiveness of SA-NODEs in capturing dynamical evolution characteristics. Comparative results indicate that SA-NODEs achieve favorable numerical performance while significantly reducing the architectural complexity of the model.

Addressing the operator learning problem for PDEs, this thesis introduces a deep neural ODE operator network (NODE-ONet) framework. Existing operator learning approaches often overlook the domain knowledge inherent in the underlying PDEs, leading to limitations in characterizing temporal dynamics and extrapolating beyond the training time frames. To alleviate these issues, the NODE-ONet framework adopts an encoder-decoder architecture comprising three core components: an encoder that spatially discretizes input functions, a neural ODE that captures latent temporal dynamics, and a decoder that reconstructs the numerical solutions in the physical space. Theoretically, we provide a unified error analysis for general encoder-decoder architectures. Computationally, we propose physics-encoded neural ODEs to incorporate the specific mathematical structures of the underlying PDEs into the network. This design improves computational efficiency and generalization capacity while reducing model complexity compared to the DeepONet algorithm. Numerical simulations on 1D nonlinear reaction-diffusion equations and 2D Navier-Stokes equations demonstrate that the proposed method possesses high numerical accuracy and predictive capabilities beyond the training time frame. Additionally, the framework supports the flexible integration of diverse encoders and decoders, and exhibits generalization capabilities across related PDE families, providing a scalable, physics-encoded computational tool.

In summary, this thesis establishes the approximation properties and convergence estimates for semi-autonomous neural ODEs at the theoretical level, and constructs an efficient operator learning framework incorporating physical structural priors at the application level. This research provides theoretical foundations and algorithmic references for the reduced-order modeling of complex dynamical systems and the efficient numerical

solution of PDEs.

Keywords: Neural ordinary differential equations, operator learning, error analysis, machine learning, scientific computing.

目 录

第 1 章 绪论

1.1 研究现状

1.1.1 神经网络的逼近理论

神经网络 (Neural Networks) 是一类由仿射变换与非线性激活函数复合而成的参数化函数族, 其基本思想是通过多层结构逐步提取数据中的有效特征, 从而实现对复杂非线性映射的表示与逼近. 从数学上看, 神经网络可视为一类具有高度非线性表达能力的函数逼近工具. 在本文中, 为了便于后续与神经常微分方程的结构进行比较, 我们首先考虑最基本的浅层神经网络 (即单隐层神经网络), 并将其写为

$$f_{NN}(\boldsymbol{x}) = \sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i \boldsymbol{x} + B_i), \tag{1.1}$$

其中, P 表示隐藏层中神经元的个数, $\boldsymbol{x} \in \mathbb{R}^d$ 为输入向量, $W_i \in \mathbb{R}^d$, $A_i \in \mathbb{R}^{d\times d}$ 和 $B_i \in \mathbb{R}^d$ 分别为网络的可学习参数, 符号 $\circ$ 表示 Hadamard 乘积. 向量函数 $\boldsymbol{\sigma} : \mathbb{R}^d \to \mathbb{R}^d$ 由标量激活函数 σ 在各分量上逐点作用而得, 其中 σ 可以取 Sigmoid, ReLU, ReLU^k 等经典激活函数. 关于本文所用符号的更精确定义, 见第 2.1 节.

神经网络之所以受到广泛关注, 根本原因在于其在众多实际问题中表现出了强大的建模能力. 特别是包含多个隐藏层的深度神经网络 (Deep Neural Networks, DNNs), 已在图像识别 [1], 自然语言处理 [2] 以及科学计算 [3] 等领域取得显著成功. 这些应用表明, 神经网络不仅能够有效处理高维数据, 还能够通过层级表征自动提取复杂结构中的关键信息, 从而在诸多高度非线性的学习任务中展现出优异性能.

然而, 从严格的数学分析角度来看, 神经网络之所以能够在如此广泛的场景中发挥作用, 其基础仍在于逼近理论. 换言之, 无论是监督学习中的回归与分类, 还是科学机器学习中对函数, 泛函乃至解算子的学习, 都首先依赖于神经网络是否具备足够强的表达能力, 以逼近目标映射. 因此, 神经网络逼近理论不仅是理解其表示能力的核心工具, 也是分析其泛化能力, 训练复杂度

与模型设计原则的重要理论起点.

在这一背景下, 神经网络逼近理论的核心命题便是所谓的万能逼近定理 (Universal Approximation Theorem). 该定理指出, 在适当的激活函数假设下, 只要隐藏层神经元数量 P 足够大, 浅层神经网络便能够以任意给定精度逼近某一函数空间中的任意目标函数. 从历史发展看, 相关思想可追溯至 Wiener 的 Tauberian 定理 [4, 定理 II]. 1989 年, Cybenko 在紧集上的连续函数空间中证明了 Sigmoid 型浅层网络的万能逼近性质 [5]. 随后, Hornik 等人在 $\mathbb{L}^p$ 框架下将这一结论推广到更一般的多层前馈网络 [6]. 对于包括 ReLU 在内的更广泛激活函数类别, Leshno 等人进一步证明了非多项式激活函数所对应的统一万能逼近结论 [7]. 关于这一方向的系统综述, 可参见文献 [8].

上述万能逼近结果本质上是定性的, 它回答的是"能否逼近"的问题, 但并未说明 "以多快的速度逼近" 以及 "该速度是否会随维数急剧恶化". 针对这一问题, Barron 的经典工作 [9] 在谱 Barron 空间中建立了浅层神经网络的定量逼近理论, 给出了与网络宽度相关的误差估计, 并揭示出在特定函数类上, 神经网络逼近能够在一定意义下摆脱传统网格型方法所面临的维数灾难. 近年来, 围绕 Barron 空间, ReLU 激活函数以及更高阶 Sobolev 意义下的逼近速率, 已有大量进一步发展, 例如文献 [10-14]. 关于神经网络定量逼近理论的整体进展, 我们也推荐参考综述文献 [15].

1.1.2 神经常微分方程

从更广的视角来看, 神经网络的逼近理论不仅适用于静态函数逼近, 也为连续时间动力系统中的学习模型提供了理论启发. 特别是, 当网络层数趋于连续极限时, 深度神经网络可以与微分方程框架建立紧密联系, 从而导向近年来受到广泛关注的神经常微分方程 (Neural Ordinary Differential Equations, NODEs) [16]. 正是在这一意义下, NODE 可被视为神经网络模型在连续时间方向上的自然推广, 而其表达能力与逼近性质也因此与经典神经网络理论保持着深刻联系. 这一模型源于如下观察: 残差神经网络 [17] (ResNets) 可以被视为连续动力系统的离散逼近, 其数值格式为:

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k) + \boldsymbol{x}_k,$$

其中 x_k 代表离散时域中第 k 个时间步的轨迹. 经典的 NODE 模型通过一个由神经网络参数化的常微分方程, 描述绝对连续的状态轨迹 $\boldsymbol{x} = \boldsymbol{x}(t)$:

$[0,T]\to\mathbb{R}^d$ 的演化:

$$\begin{cases}\dot{\boldsymbol{x}}=\displaystyle\sum_{i=1}^{P}W_i(t)\circ\boldsymbol{\sigma}(A_i(t)\boldsymbol{x}+B_i(t)),\\ \boldsymbol{x}(0)=x_0.\end{cases}\tag{1.2}$$

在全文中, 我们将上述 NODE 形式称为经典 NODE (Vanilla NODE). 其中, $A_i\in\mathbb{L}^\infty([0,T];\mathbb{R}^{d\times d})$, $W_i\in\mathbb{L}^\infty([0,T];\mathbb{R}^d)$, $B_i\in\mathbb{L}^\infty([0,T];\mathbb{R}^d)$ $(i=1,\dots,P)$ 为 NODE 的参数. 基于 NODE 作为 ResNet 形式极限的思想, P 可理解为由 $t\in[0,T]$ 参数化的网络中, 每个 "无穷薄" 层内包含的神经元数量.

NODE 是一种非常灵活的模型, 经过训练后甚至能够对非结构化或粗糙的数据集进行插值, 尤其是在这些数据具有时间依赖性的情况下. 然而, 为了量化当前构建的模型精度, 通常有理由假设这些数据仅仅是某种底层物理规律的体现, 而该规律可抽象为如下形式的动力系统:

$$\begin{cases}\dot{\boldsymbol{z}}=f(t,\boldsymbol{z}),\\ \boldsymbol{z}(0)=z_0.\end{cases}\tag{1.3}$$

随后, 通过衡量学习模型相对于预期动力学的偏差来评估其准确性. 此类 ODE 系统在大量应用中出现, 例如力学中的哈密顿 (Hamiltonian) 系统, 非平稳偏微分方程的半离散化 (如使用有限元方法; 详情参见文献 [18, 第 8.6.1 节]) 等. 此外, 含时场 (time-dependent field) 的存在也使我们能够将外部源考虑在内. 因此, ODE 系统的逼近可以被视为一个基准问题 (benchmark problem); 开发能够高效执行该任务的学习架构显得至关重要, 而这正是本文第二章的研究目标.

NODEs 属于更广泛的数据驱动系统学习与识别技术框架. 与其他最先进的范式相比, NODEs 的突出特点在于其完全数据驱动: 它既不需要预先引入候选函数字典 (例如 SINDy 或基于 Koopman 算子的方法 [19]), 也不需要关于系统物理性质的先验知识 (例如 PINNs [20]). NODEs 的连续时间建模能力使其在需要平滑插值和处理不规则采样数据的任务中具有显著优势, 例如时间序列分析 [21] 和分类 [22].

当已知底层模型的先验信息时, NODEs 的灵活性允许我们相应地定制系统 (1.2) 的结构. 这正是快速发展的保结构学习 (Structure-Preserving

Learning) 领域的核心: 通过结构约束将期望的物理属性强制引入 NODE 中. 例如, 如文献 [23, 第 2.2.2 节] 所建议的, 如果已知驱动动力学的守恒定律, 人们可以采用哈密顿神经网络 [24] 或拉格朗日 (Lagrange) 神经网络 [25] 来构建 (1.2) 中具有物理意义的右端项. 又如在近期研究 [26] 中, 作者通过选择特定的结构来保证 NODE 的长时间稳定性. 该方向的其他工作还包括 [27-28], 在这些文章中, 作者采取了相反的思路, 即从相关的 NODE 出发构建保结构的神经网络.

从理论的角度来看, NODEs 最显著的吸引力之一是其微分结构使其非常适合运用分析和最优控制技术进行研究, 其首要目标是为诸如 ResNets 等经典机器学习算法的行为提供正式的理论依据. 近年来, 文献中涌现了大量沿着这一方向展开的工作. 关于此类方程的能控性 (controllability), 例如文献 [29-30] 以及 [31]: 这些工作深入分析了不同类型的 NODE 在精确意义和近似意义上逼近目标轮廓并将输入驱动至最终目标点的能力, 并进一步探讨了控制变量 W_i, A_i, B_i 的范数与逼近精度之间的关系, 以及 NODE 深度和宽度之间的关系 [32]. 所有这些论述普遍依赖于一个基础性属性: 系数 $W_i(t), A_i(t), B_i(t)$ 对时间的依赖性. 这种属性有效地允许模型动态改变正在受到 NODEs 影响的状态空间区域, 从而仅将必要的输入移动到预期的目标.

对 NODEs 的理论研究不仅局限于能控性领域. 部分文献探讨了将 NODE 作为 ResNet 极限的本质形式化工作 [33-34]; 以及关于此类方程长期行为和其逼近性质对最终时间 T 的依赖性的研究 [35]. 该领域的另一个显著贡献是 Osher 等人的工作 [36]: 该工作证明了与 NODEs 对应的传输方程的万能逼近性质 (Universal Approximation Property, UAP), 证明了连续性方程的解可以通过具有分段常数训练权重的 NODEs 进行逼近, 从而达到任意程度的相近.

1.1.3 神经网络求解偏微分方程与算子学习

偏微分方程 (PDEs) 是刻画物理学, 工程学, 生物学及经济学等领域中复杂动态系统的基础数学工具. 在这些系统中, 诸如温度, 压力, 波幅或种群密度等物理量随空间与时间不断演化. 获取偏微分方程的解析解往往极其困难, 尤以非线性或高维问题为甚. 因此, 借助数值方法求取其近似解已成为不可或缺的手段. 传统的偏微分方程数值解法, 例如有限元法 (Finite

Element Method, FEM), 有限差分法 (Finite Difference Method, FDM), 有限体积法 (Finite Volumn Method, FVM) 及谱方法等, 通过将连续的时空域离散化为计算网格, 从而将偏微分方程转化为易于处理的代数系统. 这些方法建立在严密的数学理论基础之上, 能够提供具备严格收敛性与高精度保证的数值解, 尤其在处理线性偏微分方程与适定问题时表现卓越. 此外, 此类方法在中低维 (不超过三维) 空间中极为高效, 且与之配套的线性系统求解器与预条件子均已发展得相当完备. 因此, 凭借在各类偏微分方程应用中所展现出的高保真度与可靠性, 传统数值方法具有不可替代的地位. 然而, 在面对高维空间与复杂区域上的偏微分方程时, 传统方法依然面临着严峻的挑战. 在科学与工程计算中, 诸如反问题求解与最优控制等前沿课题, 往往要求针对不同的参数配置对偏微分方程进行成百上千次的反复求解 [37-40]. 针对这类多查询任务, 依赖于网格离散化的传统数值求解器会反复生成大规模代数系统, 导致计算开销呈指数级增长, 从而在多查询场景下面临巨大困难.

近年来, 基于深度学习的偏微分方程求解方法引起了学术界的广泛关注, 参见 [41-43, 20, 44] 及其参考文献. 得益于深度神经网络卓越的万能逼近能力 [5, 45-46] 与出色的泛化性能 [47-48], 这些数据驱动方法极大提升了处理复杂多尺度偏微分方程系统的可行性, 并在各大科学与工程领域得到了成功应用, 详见文献 [49-50, 3, 51-52, 39]. 较之于传统数值格式, 深度学习方法通常具备无网格特性且易于实现, 在求解定义于复杂几何区域或高维空间上的各类偏微分方程时展现出极高的灵活性. 代表性的深度学习方法, 如深度 Ritz 方法 [41], 深度 Galerkin 方法 [44] 以及物理信息神经网络 (PINNs) [53, 20], 均利用神经网络直接参数化偏微分方程的解, 通过优化特定的损失函数来获取数值近似.

尽管此类方法在诸多实际应用中取得了令人瞩目的成果, 但其本质上仍是为单一特定的偏微分方程实例量身定制的. 换言之, 当偏微分方程的关键参数 (如初始条件, 边界条件, 源项或物理系数) 发生改变时, 必须从头训练一个新的神经网络. 这一过程的计算代价可能极为高昂, 导致其难以胜任对动态输入数据进行实时预测的任务. 为突破这一瓶颈, 近期文献中提出了一系列算子学习 (Operator Learning) 方法, 参见例如 [54-55, 42-43].

算子学习的核心思想是利用神经网络直接逼近偏微分方程的解算子, 即建立从偏微分方程无限维参数空间 (如源项, 初值条件) 到解空间的非线性映

射. 一旦这种神经解算子训练完毕, 我们便获得了一个极其高效的神经代理模型. 此时, 针对任何新的参数输入, 仅需一次神经网络的前向传播即可瞬间输出相应的偏微分方程解. 现阶段, 用于求解偏微分方程的代表性算子学习框架包括深度算子网络 (DeepONets) [43], MIONet [56], 物理信息 DeepONets [57], 傅里叶神经算子 (FNO) [42], 图神经算子 [58], 随机特征模型 [59], PCA-Net [54], 拉普拉斯神经算子 [60] 以及上下文算子网络 [61].

尽管在追求极致精度的单次求解任务中, 传统数值求解器或许仍占据优势, 但算子学习方法不仅能达到工程应用所需的令人满意的精度, 更在数值效率与多任务泛化能力上展现出传统方法难以企及的优势. 因此, 算子学习正被广泛应用于构建偏微分方程的高效代理模型, 对于那些计算成本高昂且需要频繁调用的重复性模拟任务而言, 更是具有无与伦比的吸引力, 参见 [62-68, 37-38, 40].

在诸多算子学习方法中, 深受算子万能逼近定理 [69, 43] 启发并采用编码器-解码器 (Encoder-Decoder) 架构的 DeepONets 备受瞩目. 该模型已在多种复杂应用场景中展现出卓越性能, 如电流体动力学对流 [70], 多尺度气泡生长动力学 [71] 以及主动脉夹层模拟 [72]. 关于 DeepONets 更广泛的工程应用, 理论剖析与数值研究, 可进一步参阅 [73-75]. 为了在不同数学设定下更有效地学习偏微分方程解算子, 学界还开发了多种 DeepONets 变体, 如贝叶斯 DeepONet [76], 结合本征正交分解的 POD-DeepONet [75] 以及融合物理方程残差的物理信息 DeepONets [57].

从网络结构上看, DeepONet 主要由分支网络 (branch net) 与主干网络 (trunk net) 两部分构成. 其中, 分支网络以输入函数在若干固定传感点上的离散采样值为输入, 用于提取并编码输入函数的信息; 主干网络则以输出函数的评价位置为输入, 用于表征该位置处的响应特征. 二者输出的隐特征随后通过内积或线性组合的方式进行融合, 从而得到对应位置处的算子预测值. 因而, DeepONet 本质上可理解为一种 “由分支网络生成系数, 由主干网络生成与坐标相关的基函数” 的编码器–解码器型算子学习框架 [43, 75].

然而必须指出的是, 上述基于 DeepONets 的框架通常将时间变量与空间变量同等对待, 并在整个时空域上以一次性全局的方式对神经网络进行训练. 这种时空耦合的处理方式极易导致神经网络在优化过程中遭遇病态或梯度传播困难, 进而引发数值精度的显著下降, 参见文献 [77-78]. 此外, 尽

管 DeepONets 具有广泛的通用性, 但其主干网络往往采用常规的神经网络架构 (如全连接神经网络 (FCNNs), 卷积神经网络 (CNNs) 等), 从而不可避免地忽略了底层偏微分方程所蕴含的特定领域知识, 例如系统固有的动力学演化结构或特定物理参数的作用机制. 这种对先验物理知识的忽略在很大程度上削弱了其学习解算子时的计算效率. 尤为致命的是, DeepONets 在刻画含时偏微分方程的时间演化规律方面存在内在局限性, 这使其在超越训练时间区间进行外推预测时极易失效. 另一方面, 正如文献 [56] 所指出的, 原始 DeepONets [43] 在处理包含多个独立输入函数的算子逼近任务时, 其精度与表征能力略显不足. 为弥补这一缺陷, 文献 [56] 提出了多输入算子网络 (MIONet), 显著提升了多输入场景下的数值精度. 遗憾的是, MIONet 的网络规模与模型复杂度会随着输入函数数量的增加而急剧膨胀, 且未能从根本上解决超越训练时间框架的预测难题.

这些局限性强烈促使我们探索并构建一种全新的算子学习架构, 该架构必须同时满足以下三个关键准则: (i) 将时间视作连续变量, 而非仅在固定的训练网格节点上进行刻画; (ii) 能够显式地编码并保留底层偏微分方程的动力学演化结构; (iii) 在向训练时间范围之外进行长时间外推预测时, 能够保持高度的稳定性与保真度. 而这正是本文第三章的研究动机.

1.2 主要结果

1.2.1 半自治神经常微分方程的逼近理论及其应用

将 NODE 视为 ResNet 的连续极限时, 让系数随时间变化是一个自然的选择. 然而, 这一选择会显著增加模型的复杂度: 在实际应用中, 每一个时间步通常需要对应一层网络, 这导致参数数量随时间步数呈线性增长. 因此, 一个自然的问题是: 是否能够在保留关键动力学特征 (这些特征在具体应用中往往至关重要) 的同时, 降低模型的复杂度?

此外, 现有关于 NODE 的大量文献主要关注于优化系数 $W_i(t)$, $A_i(t)$ 和 $B_i(t)$, 旨在将 $t = 0$ 时的初始点分布 (对应输入层) 驱动至 $t = T$ 时的目标分布 (对应输出层), 却较少关注在整个时间区间 $[0, T]$ 上对完整轨迹的跟踪; 近期的工作 [79] 是少有的例外. 然而, 基于建模的直觉, 我们期望 NODE 不仅能够匹配初始与终端状态, 更应具备逼近整段轨迹的能力. 受这些问题的启

发, 本文聚焦于如下一种特殊形式的 NODE:

$$\begin{cases} \dot{\boldsymbol{x}} = \sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 \boldsymbol{x} + A_i^2 t + B_i), \\ \boldsymbol{x}(0) = x_0, \end{cases} \tag{1.4}$$

其中 $P \in \mathbb{N}_+$ 是网络宽度, 且 $W_i \in \mathbb{R}^d$, $A_i^1 \in \mathbb{R}^{d\times d}$, $A_i^2 \in \mathbb{R}^d$, $B_i \in \mathbb{R}^d$ $(i = 1, \ldots, P)$ 为神经网络的参数. 需要注意的是, 此时的参数完全不随时间变化; 实际上, t 仅作为激活函数内部的乘性因子出现. 基于这一结构特征, 我们将其命名为半自治神经常微分方程 (Semi-Autonomous NODEs), 简称 SA-NODEs.

这种特定的结构选择并非随意为之. 其出发点源于 Pinkus 的经典万能逼近结果 [8]: 对于紧集上的连续向量场 $f(t, \boldsymbol{z})$, 可以利用如下形式的浅层 (单隐层) 神经网络在 $\mathbb{L}^\infty$ 范数下实现任意精度的逼近:

$$f_\Theta(t, \boldsymbol{z}) = \sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 \boldsymbol{z} + A_i^2 t + B_i),$$

其中 $\Theta = (W_i, A_i^1, A_i^2, B_i)_{i=1}^P$. 其完整的定理表述见定理 2.4. 以此为起点, 自然引出了我们的第一个主要结果 (定理 2.1), 即 SA-NODEs 对动力系统的万能逼近性质. 直观而言, 该结果可以通过结合 Pinkus 定理与 Grönwall 估计来获得. 但这一推导过程并非平凡: 由于 Pinkus 定理仅在紧集上成立, 为了能够应用 Grönwall 估计, 必须准确地识别出一个合适的紧集, 使得该紧集能够同时包络由逼近向量场 f_Θ 产生的所有 SA-NODE 解轨迹.

除定理 2.1 之外, 本文其他的核心理论贡献亦沿袭了相似的思路, 它们均源自浅层神经网络在适当条件下的万能逼近性质. 我们将这些贡献详细概括如下:

1. 前文提及的定理 2.1 确立了 SA-NODEs 对形如 (1.3) 的动力系统的万能逼近性质. 在仅假设向量场 f 关于时间连续且关于空间一致 Lipschitz (见假设 2.1) 的条件下, 我们证明: 对于任意给定的容差 $\varepsilon > 0$ 以及任意的初始数据紧集 $K \subset \mathbb{R}^d$, 均存在网络宽度 $P \geq 1$ 及参数 W_i, A_i^1, A_i^2, B_i, 使得对于所有 $z_0 \in K$, 原系统的轨迹均可在 $\mathbb{L}^\infty(0, T)$ 意义下, 被具有相同初值的 SA-NODE 轨迹逼近, 且误差不超过 ε. 需要强调的是, 该结论

并非仅关注系统的初始与终端状态, 而是涵盖了整段连续轨迹; 这可被视为对文献 [22] 中万能逼近结果在轨迹层面上的推广.

2. 我们的第二个结果 (即定理 2.2) 给出了 SA-NODEs 逼近速率的上界, 并明确了该速率与网络宽度 P 之间的定量关系. 为此, 我们引入了额外的正则性假设, 即假设 f 属于局部 Sobolev 空间 $\mathcal{H}^k_{\text{loc}}$; 即对任意紧集 $X \subseteq \mathbb{R}^d \times [0,T]$, 函数在 X 上的限制属于 Sobolev 空间 $\mathcal{H}^k(X)$. 同时假设 $k > (d+1)/2 + 2$ (见假设 2.2). 在该设定下, 设 $\boldsymbol{z}_{z_0}$ 与 $\boldsymbol{x}_{z_0}$ 分别表示真实动力系统 (1.3) 与 SA-NODE (1.4) 从同一初始点 z_0 出发的解, 我们建立如下误差估计:

$$\sup_{(z_0,t)\in K\times[0,T]} \|\boldsymbol{z}_{z_0}(t) - \boldsymbol{x}_{z_0}(t)\| \leq \frac{C_{T,K,f}}{\sqrt{P}}, \tag{1.5}$$

其中常数 $C_{T,K,f}$ 独立于 P. 与使用有限元法等经典插值手段相比, 当向量场具备足够的平滑性时, SA-NODE 方法在关于神经元数量的收敛速率上摆脱了维数灾难 (curse of dimensionality) 的影响 (详见注记 2.3).

3. 基于前述成果, 定理 2.3 确立了传输方程 (2.7) (其解记为 ρ) 被其对应的神经网络模型 (2.8) (其解记为 ρ_Θ) 逼近时的收敛速率:

$$\sup_{t\in[0,T]} \mathbb{W}_1(\rho(t,\cdot), \rho_\Theta(t,\cdot)) \leq \frac{C_{T,f,\rho_0}}{\sqrt{P}}, \tag{1.6}$$

其中 ρ_0 为传输方程的初始分布, C_{T,f,ρ_0} 为独立于 P 的常数, 而 $\mathbb{W}_1(\cdot,\cdot)$ 表示 Wasserstein-1 距离 (详见第2.2节, 定义2.1). 需要指出的是, 该结论改进了文献 [80] 中的发现 (该文献主要关注终端时刻分布 $\rho(T,\cdot)$ 的逼近); 同时也弥补了文献 [36] 的不足, 后者虽在 $\mathbb{W}_2$ 意义下给出了传输方程的万能逼近结果, 但缺乏对收敛速率的精确刻画.

4. 最后, 我们呈现了一系列数值实验, 并对 SA-NODEs 的性能进行了详尽的分析. 首先, 在第 2.4 节中, 我们借助经典的最优控制技术, 阐明了我们的主要理论结果与 SA-NODEs 训练过程之间的内在联系. 随后, 我们深入考察了此类方程的逼近能力, 并将其与经典 NODEs (Vanilla NODEs) 进行了比较.

   在固定每层神经元数量 (即宽度 P) 以确保对比公平的前提下, 我们观察到 SA-NODEs 在多个方面均优于经典 NODEs: (1) SA-NODEs 涉及

的参数量显著减少, 从而大幅缩短了训练时间并降低了存储需求; (2) 相对于训练轮数 (epochs), 其收敛速度更快; (3) 即使在较小规模的数据集下, SA-NODEs 依然能够取得精确的逼近结果; (4) 在逼近 ODE 和传输方程时, SA-NODEs 展现出比经典 NODEs 更优越的稳定性.

在此, 我们概述主要结果证明的核心思想, 完整的详细论证将在第 2.3 节中给出.

1. **定性收敛 (Qualitative convergence)**. 正如前文所述, SA-NODE 的结构衍生自对原始 ODE 向量场 f 的 Pinkus 逼近. 由于神经网络的逼近在整个空间上并非一致的, 因此必须确定一个合适的紧集: 对于足够小的 ε, 该紧集能够同时约束所有用于构建原始 ODE 的 ε-逼近的 SA-NODEs 解轨迹. 结合初始条件集 K 的紧致性, 以及在引理 2.1 中通过自举论证 (bootstrapping argument) 获得的先验界 (a priori bound), 我们确定了该紧集. 随后, 在该紧集上用神经网络逼近替换 f, 并应用 Grönwall 不等式, 即可推导出 SA-NODE 解的定性收敛结果.

2. **定量收敛 (Quantitative convergence)**. 收敛速率 (1.5) 源于浅层神经网络 (设神经元数量为 P) 在 $\mathbb{L}^\infty$ 范数下对紧集上向量场 f 所实现的 $\mathcal{O}(1/\sqrt{P})$ 逼近. 这种逼近对于 Barron 空间 (2.9) 中的函数是成立的; 参见引理 2.2 以及 [81, 定理 2]. 此外, 我们证明了所使用神经网络的 Lipschitz 常数独立于 P. 在引理 2.3 中, 我们证明了当 $k \geq (d+1)/2$ 时, 局部 Sobolev 空间 $\mathcal{H}^k_{\text{loc}}$ 能够连续嵌入到 Barron 空间中. 因此, 当 f 位于该 Sobolev 空间时, 即可确保存在如推论 2.1 所述的, 具有对 Lipschitz 常数一致控制的神经网络逼近. 这种一致界限使我们能够确定一个合适的紧集, SA-NODEs 的所有轨迹都将保持在该紧集内, 从而能够应用 Grönwall 类型的论证来导出估计式 (1.5).

3. **传输方程 (Transport equation)**. 传输方程的收敛估计 (1.6) 可通过将界限 (1.5) 应用于其特征线 ODE, 并结合叠加原理 (superposition principle) 以及 Wasserstein-1 距离的定义推导得出. 此外, 如注记 2.7 所述, 如果向量场在 $\mathcal{W}^{1,\infty}$ 范数下被逼近, 那么在 $\mathbb{L}^p$ 范数下也会成立类似的收敛速率.

1.2.2 神经常微分方程在算子学习中的应用

基于 NODE 的逼近理论, 我们建立了一套基于 NODE 的算子学习框架. 为了清晰地阐述本文的核心思想, 我们考虑如下一类偏微分方程模型:

$$\begin{cases} \partial_t u(t,x) + \mathcal{L}[a](u)(t,x) = f(t,x) & \forall (t,x) \in [0,T] \times \Omega, \\ u(0,x) = u_0(x) & \forall x \in \Omega, \\ \mathcal{B}u(t,x) = u_b(t,x) & \forall (t,x) \in [0,T] \times \partial\Omega. \end{cases} \tag{1.7}$$

式中, $T > 0$, $\Omega \subset \mathbb{R}^d$ 为具有合适光滑条件边界 $\partial\Omega$ 的有界区域, $\mathcal{L}$ 表示以 $a : [0,T] \times \Omega \to \mathbb{R}$ 为底层参数的微分算子 (例如, $\mathcal{L}[a](u)(t,x) = -\nabla \cdot (a(t,x))\nabla u(t,x)$), $f : [0,T] \times \Omega \to \mathbb{R}$ 为源项, $u_0 : \Omega \to \mathbb{R}$ 为初始条件, $\mathcal{B}$ 表示用以施加任意的 Dirichlet, Neumann, Robin 或周期边界条件的边界算子, 而 $u_b : [0,T] \times \partial\Omega \to \mathbb{R}$ 为边值.

令 $v \subset \{f, a, u_0, u_b\}$ 表示方程 (1.7) 中的一个参数集合. 我们假设, 对于任意相容的 v, 系统 (1.7) 在适当的函数空间内均存在唯一的经典解 u. 算子学习的核心目标是利用一个带有可训练参数 θ 的基于神经网络的泛函 Ψ_θ 来逼近真实的解算子 $\Psi^\dagger$, 即实现从参数 v 到解 u 的映射:

$$\Psi_\theta \approx \Psi^\dagger : v \mapsto u.$$

在本文中, 我们在深度神经常微分方程算子网络 (NODE-ONet) 这一新颖框架下对 Ψ_θ 进行构造. NODE-ONet 采用经典的编码器-解码器 (Encoder-Decoder) 架构, 具体包含以下三个核心组件:

1. **编码 (Encoding):** 借助空间离散化格式 (例如, 计算网格上的逐点求值, 抑或是基于有限元或傅里叶基函数的展开), 将输入参数集合 v 嵌入至潜在空间 (latent space) 中. 经过编码后, 原偏微分方程 (1.7) 的系统可由一组降维后的状态变量来表征, 我们将其称为潜在变量 (latent variables), 其动力学行为随时间持续演化.

2. **NODE 代理模型 (NODE surrogate):** 我们开发了物理编码的 NODE, 采用显式依赖于时间的参数 (例如, 关于 t 的多项式形式) 来逼近潜在变量的动力学演化, 从而大幅降低了传统 NODE 的参数复杂度 [16]. 此外, 物理编码 NODE 的网络架构被精心设计以融入底层偏微分方程的

内在结构性质, 例如方程 (1.7) 中 a 与 u 之间的非线性依赖关系, 以及 f 与 u 之间的耦合关系. 系统中其他已知参数 (即方程 (1.7) 中的集合 $\{f, a, u_0, u_b\} \setminus v$) 的物理效应同样被无缝集成到该 NODE 的设计之中.

3. **解码 (Decoding):** 在成功学习潜在变量的动力学规律之后, 依靠一个仅依赖于空间域的解码器, 将 NODE 的输出在物理空间中重建为偏微分方程的最终解 u.

必须指出的是, NODE-ONet 框架对时间变量与空间变量采取了解耦分离的处理方式. 这种分离策略与求解含时偏微分方程的传统数值方法具有高度的一致性, 后者通常采用时间顺序步进的格式逐步推进求解过程, 而非在整个时空域上进行全局性的隐式求解. 因此, 经过训练且仅依赖于空间域的解码器, 具备极强的泛化能力, 能够直接迁移至具有相似结构的各类偏微分方程算子学习任务中, 这一点将在第 3.4 节中得到充分验证. 此外尤为值得注意的是, 尽管 NODE-ONet 最初是为演化型偏微分方程量身定制的, 但它同样能够极其自然地推广至稳态问题中, 详见注记 3.1, 推论 3.1 以及第 3.2 节的深入探讨.

本文引入的 NODE-ONet 提供了一种用于偏微分方程算子学习的全新框架, 并在理论分析与计算实践层面均取得了实质性的突破. 在理论层面, 我们在定理 3.1 中为一般的编码器-解码器架构建立了严格的误差估计, 这为未来逐案严格证明各类 NODE-ONet 的收敛性奠定了坚实的理论基石. 在计算实践层面, 我们创新性地提出了物理编码的 NODE, 它是赋予 NODE-ONet 极高计算效率, 鲁棒性及广泛适用性的核心引擎, 这将在第 3.4 节的详尽数值实验中得到全面展现.

这种物理编码的设计赋予了模型两项关键的独特能力. 首先, 它使得模型能够深刻捕捉底层偏微分方程的内在动力学模式, 从而实现对超越训练时间框架的系统演化过程的高保真预测. 其次, 它能够无缝且自然地扩展至多输入函数的情形, 且完全不会增加神经网络的参数复杂度. 这种卓越的可扩展性使得 NODE-ONet 能够灵活适配各种复杂的输入配置, 同时保持极致的计算效率. 值得强调的是, 物理编码 NODE 是深度根植于底层偏微分方程的特定数学结构而设计的, 而其神经网络则完全基于偏微分方程高精度数值解生成的数据进行离线训练. 由此可见, 物理编码的 NODE-ONet 在基于偏微分方程的领域先验知识与纯数据驱动范式之间, 架构起了一座极具潜力的协

同桥梁. 这种深度的混合机理不仅汲取了偏微分方程严格的数学可解释性与绝对可靠性, 同时也完美融合了深度神经网络的强大表征灵活性与泛化能力.

1.3 本文的主要结构

本文的整体章节结构与内容安排如下:

第二章探讨了半自治神经常微分方程 (SA-NODE) 的理论逼近性质. 本章不仅详细给出了 SA-NODE 逼近常微分方程与传输方程的严格定理证明, 还通过详尽的数值模拟充分验证了上述理论分析的可靠性与有效性.

第三章阐述了深度神经常微分方程算子网络 (NODE-ONet) 框架. 首先, 我们引入了一般的编码器-解码器 (Encoder-Decoder) 网络架构并对其进行了严密的误差分析. 随后, 本章从核心设计理念, 网络架构构造, 模型训练策略以及与代表性算法 DeepONet 的深度对比等多个维度, 对 NODE-ONet 算法进行了全面剖析. 最后, 借助一维扩散-反应方程与二维 Navier-Stokes 方程的数值实验, 充分验证了该模型在求解偏微分方程时的可靠性, 高效性与鲁棒性.

第四章对全文的核心研究成果与主要贡献进行了系统性的总结, 并对该领域未来可能的研究方向与潜在的拓展工作进行了的展望.

第 2 章 半自治神经常微分方程的逼近理论与应用

本章围绕半自治神经常微分方程 (SA-NODE) 的理论基础, 逼近性质及其应用展开. 首先, 在第 2.1 节中, 我们介绍全文所需的基本记号, 函数空间与激活函数设定, 并给出 SA-NODE 的模型结构及其适定性, 为后续分析奠定基础. 接着, 在第 2.2 节中, 我们系统建立 SA-NODE 的逼近理论: 先证明其对非自治常微分方程解轨道的万能逼近性质, 再在额外 Sobolev 正则性假设下给出关于网络宽度的定量逼近速率, 并进一步将该结果推广到与动力系统相关的传输方程, 在 Wasserstein-1 距离意义下得到一致收敛估计. 随后, 在第 2.3 节中, 我们给出上述主要结果的详细证明, 其中通过 Barron 空间逼近, Sobolev 空间嵌入以及流映射与推前测度之间的联系, 构成本章理论分析的核心技术路线. 在此基础上, 第 2.4 节进一步从优化与控制的角度讨论 SA-NODE 的训练策略, 将参数学习表述为最优控制问题, 并借助伴随方法推导目标泛函的梯度公式, 同时说明其在离散数据与传输方程情形下的实现方式. 最后, 第 2.5 节通过常微分方程与传输方程的数值实验, 验证 SA-NODE 在逼近精度, 训练效率, 模型复杂度及稳定性方面的表现, 并与经典 NODE 模型进行对比, 从而说明 SA-NODE 的理论价值与实际潜力.

2.1 预备知识

令 $n, d \in \mathbb{N}_+$. 对任意 $x \in \mathbb{R}^n$ 与 $p \in \mathbb{N}_+$, 记 $\|x\|_{\ell^p}$ 为 x 的 ℓ^p-范数. 为简便起见, 记 $\|x\|$ 为 x 的欧几里得范数 (即 ℓ^2-范数). 对于 $x, y \in \mathbb{R}^n$, 其内积与 Hadamard 乘积分别记为 $\langle x, y\rangle$ 和 $x \circ y$, 即

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i, \quad x \circ y = (x_1 y_1, \ldots, x_n y_n).$$

在本文后续内容中, 除非另有说明, 我们将激活函数 σ 固定为 ReLU 函数, 并用 $\boldsymbol{\sigma}$ 表示其 d 维向量值形式:

$$\sigma(x) = \max\{x, 0\}, \quad \forall x \in \mathbb{R}; \quad \boldsymbol{\sigma}(\boldsymbol{x}) = (\sigma(x_1), \ldots, \sigma(x_d)), \quad \forall \boldsymbol{x} \in \mathbb{R}^d,$$

其中 $\boldsymbol{x}=(x_1,\dots,x_d)$. 设 $\Omega\subseteq\mathbb{R}^n$ 为闭集. 记 $\mathcal{H}^k(\Omega)$ 为 Sobolev 空间 (参见 [82, 定义 3.2, $p=2$], 其中 $k\in\mathbb{N}_+$), 并记 $\mathcal{C}(\Omega)$ 为 Ω 上的连续函数空间, 二者均赋予其标准范数. 对于任意向量值函数 $F\in\mathcal{H}^k(\Omega;\mathbb{R}^d)$ 与 $G\in\mathcal{C}(\Omega;\mathbb{R}^d)$, 其范数定义为

$$\|F\|_{\mathcal{H}^k(\Omega;\mathbb{R}^d)}:=\sqrt{\sum_{i=1}^{d}\|F_i\|^2_{\mathcal{H}^k(\Omega)}},\qquad \|G\|_{\mathcal{C}(\Omega;\mathbb{R}^d)}:=\sup_{x\in\Omega}\|G(x)\|,$$

其中 F_i 表示 F 的第 i 个分量. 在不致引起混淆的情况下, 为简便起见我们将简写为 $\|F\|_{\mathcal{H}^k(\Omega)}$.

让我们考虑一个常微分方程, 其向量场从 $\mathbb{R}^{d+1}$ (d 维空间变量与一维时间变量) 映射到 $\mathbb{R}^d$. 我们的目标是利用向量值的浅层神经网络 (见推论 2.1) 来逼近该向量场. 由此引出了如下动力系统, 我们称之为半自治神经常微分方程 (SA-NODE):

$$\begin{cases}\dot{\boldsymbol{x}}=\displaystyle\sum_{i=1}^{P}W_i\circ\boldsymbol{\sigma}(A_i^1\boldsymbol{x}+A_i^2t+B_i), & t\in(0,T),\\ \boldsymbol{x}(0)=x_0,\end{cases}\tag{2.1}$$

其中 $P\in\mathbb{N}_+$ (从神经网络角度，表示网络宽度), 且 $W_i\in\mathbb{R}^d$, $A_i^1\in\mathbb{R}^{d\times d}$, $A_i^2\in\mathbb{R}^d$, $B_i\in\mathbb{R}^d(i=1,\dots,P)$ 为 SA-NODE 的参数. 因此, SA-NODE 的参数总数 (即自由度, Degree of Freedom, DoF) 为 $Pd(d+3)$.

设 $\Theta=(W_i,A_i^1,A_i^2,B_i)_{i=1}^P$. 为方便起见, 我们将方程 (2.1) 的右端项记为 $f_\Theta(x,t)$. 易证 f_Θ 满足关于 x 的全局 Lipschitz 连续性:

$$\|f_\Theta(x,t)-f_\Theta(y,t)\|\ \le\ L_\Theta\|x-y\|,\qquad\forall\,x,y\in\mathbb{R}^d,\ \forall\,t\ge0,\tag{2.2}$$

其中 Lipschitz 常数 L_Θ 由下式给出:

$$L_\Theta=\left(\sum_{j=1}^{d}\left(\sum_{i=1}^{P}|(W_i)_j|\,\|(A_i^1)_j\|\right)^2\right)^{1/2}.\tag{2.3}$$

此处, $(W_i)_j$ 表示权重向量 W_i 的第 j 个分量, $(A_i^1)_j$ 表示矩阵 A_i^1 的第 j 行.

因此, 根据 Cauchy-Lipschitz 定理 (即 Picard-Lindelöf 定理) 可知: 对于任意参数 Θ 和任意初始点 $\boldsymbol{x}_0$, 系统 (2.1) 在 $t\ge0$ 上存在唯一解.

2.2 半自治神经常微分方程的逼近理论

对于固定的 $T > 0$, 我们考虑如下非自治 ODE 系统: 其向量场为 $f\colon \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$, 初始状态为 $\boldsymbol{z}_0 \in \mathbb{R}^d$,

$$\begin{cases} \dot{\boldsymbol{z}} = f(\boldsymbol{z}, t), \ t \in (0,T), \\ \boldsymbol{z}(0) = z_0. \end{cases} \tag{2.4}$$

为保证系统 (2.4) 解的存在性与唯一性, 我们引入如下假设.

假设 2.1. 函数 $f\colon \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ 关于 t 连续, 且存在常数 $L > 0$ 使得

$$\|f(x,t) - f(y,t)\| \le L\|x - y\|, \quad \forall (x,y) \in \mathbb{R}^d, \forall t \in [0,T].$$

我们的第一个主要结果刻画了 SA-NODEs 对 ODE 系统的万能逼近性质.

定理 2.1. 设假设 *2.1* 成立. 对任意紧集 $K \subseteq \mathbb{R}^d$ 与任意 $\varepsilon > 0$, 存在常数 $P_{\varepsilon,T,K,f}$ 使得: 对任意 $P \ge P_{\varepsilon,T,K,f}$, 可取参数 $(W_i, A_i^1, A_i^2, B_i) \in \mathbb{R}^d \times \mathbb{R}^{d\times d} \times \mathbb{R}^d \times \mathbb{R}^d$ *(*其中 $i = 1, \dots, P$*)* 使得

$$\|\boldsymbol{z}_{z_0}(\cdot) - \boldsymbol{x}_{z_0}(\cdot)\|_{\mathcal{C}([0,T];\mathbb{R}^d)} \le \varepsilon, \quad \forall z_0 \in K,$$

其中 $\boldsymbol{z}_{z_0}(\cdot)$ *(*相应地, $\boldsymbol{x}_{z_0}(\cdot)$*)* 表示在时间区间 $[0,T]$ 上, 以 z_0 为初始状态的系统 (2.4) *(*相应地, (2.1)*)* 的解.

需要强调的是, 上述定理中的最优参数选择与 $z_0 \in K$ 的具体取值无关; 这也说明我们所实现的是对动力系统本身的学习, 而非仅仅对单一轨迹的拟合.

注 2.1. 当系统 (2.4) 为自治系统时, 定理 *2.1* 可在完全相同的形式下应用于更简单的自治 *NODE:*

$$\begin{cases} \dot{\boldsymbol{x}} = \displaystyle\sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 \boldsymbol{x} + B_i), \\ \boldsymbol{x}(0) = x_0, \end{cases}$$

该系统可通过令 $A_i^2 = 0$ 得到. 在本文中, 我们有意不预设数据是由自治还是非自治系统产生; 我们认为这更符合真实实验情形, 因为实际数据往往夹杂着微小的随时间变化的误差. 当然, 若已知系统 (2.4) 的结构信息, 则可直接采用自治 *NODE*.

我们的第二个结果给出了 SA-NODEs 关于网络宽度 P 对 ODE 系统的逼近速率上界 (见定理 2.2). 为此, 我们需要对向量场 f 的正则性作进一步假设. 设 X 为 $\mathbb{R}^d \times [0,T]$ 的任意子集. 局部 Sobolev 空间 $\mathcal{H}^k_{\text{loc}}(\mathbb{R}^d \times [0,T])$ 定义为: 对任意紧集 $X \subseteq \mathbb{R}^d \times [0,T]$, 函数在 X 上的限制属于 $\mathcal{H}^k(X)$.

假设 2.2. 存在 $k > (d+1)/2 + 2$ 使得 $f \in \mathcal{H}^k_{\text{loc}}(\mathbb{R}^d \times [0,T]; \mathbb{R}^d)$.

定理 2.2. 设假设 *2.1*-*2.2* 成立. 固定任意紧集 $K \subseteq \mathbb{R}^d$, 则对任意 $P \geq 3$, 存在参数 $(W_i, A_i^1, A_i^2, B_i) \in \mathbb{R}^d \times \mathbb{R}^{d\times d} \times \mathbb{R}^d \times \mathbb{R}^d (i = 1, \ldots, P)$, 使得

$$\|\boldsymbol{z}_{z_0}(\cdot) - \boldsymbol{x}_{z_0}(\cdot)\|_{\mathcal{C}([0,T];\mathbb{R}^d)} \leq \frac{C_{T,K,f}}{\sqrt{P}}, \quad \forall z_0 \in K, \tag{2.5}$$

其中常数 $C_{T,K,f}$ 与 P 无关, 且 $\boldsymbol{z}_{z_0}(\cdot)$(相应地, $\boldsymbol{x}_{z_0}(\cdot)$) 表示在区间 $[0,T]$ 上, 以 z_0 为初始状态的系统 (2.4)(相应地, (2.1)) 的解.

注 2.2. 定理 *2.1* 与定理 *2.2* 分别刻画了 *SA-NODEs* 逼近性质的不同侧面: 前者给出定性结论, 后者则利用神经元数量 P 对逼近精度进行了定量刻画. 除额外的正则性要求外, 我们在得到估计式 (2.5) 时所作的主要让步在于: 该界限必须对 K 中的所有初始值均成立.

注 2.3 (与有限元逼近的比较)**.** 我们将定理 *2.2* 的逼近结果与使用 P_1 有限元方法 *(FEM)* 插值向量场 f 的结果进行对比. 设假设 *2.2* 成立. 由 *Sobolev* 嵌入定理可得

$$f \in \mathcal{W}^{2,\infty}_{loc}(\mathbb{R}^{d+1}).$$

因此, 对任意具有 *Lipschitz* 边界的紧区域 $\Omega \subset \mathbb{R}^{d+1}$ 及其网格尺度为 h 的正则网格 Ω_h, 由 *[83, 定理 3.1.6]* 可知, 在有限元空间中存在逼近 f_h 使得

$$\|f - f_h\|_{\mathbb{L}^\infty(\Omega)} \leq C\|f\|_{\mathcal{W}^{2,\infty}(\Omega)} h^2,$$

其中常数 C 仅依赖于区域 Ω.

当固定基函数数量 P 时, 在规则网格上取 $h \sim P^{-1/(d+1)}$ 的 P_1-*FEM* 插值可得到误差阶 $\|f - f_h\|_{\mathbb{L}^\infty(\Omega)} = \mathcal{O}(P^{-2/(d+1)})$. 该复杂度随维数 d 快速恶化, 充分体现了经典的维数灾难; 即便对于高度光滑的函数 *(*例如 $f \in \mathcal{C}_c^\infty(\mathbb{R}^{d+1})$*)*, 这一现象也依然存在.

相比之下, 定理 2.2 表明在假设 2.2 下, *SA-NODE* 的 $\mathbb{L}^\infty$ 误差关于神经元数 P 的阶为 $\mathcal{O}(P^{-1/2})$. 尽管与 $P^{-1/2}$ 相伴的前置因子会随维数 d 呈指数增长 *(*见注记 2.4*)*, 但就 P 本身而言, 其收敛阶并不随维数退化. 因此, 从渐近角度来看, 当固定 $d \geq 4$ 且 P 足够大时, 神经网络逼近的误差衰减速度快于经典 *FEM* 的收敛速率 $\mathcal{O}(P^{-2/(d+1)})$. 这说明在高维情形下, 尽管常数项中仍存在维数灾难的影响, 但基于神经网络的模型在渐近意义下依然具备优势.

最后我们指出, 与 *FEM* 不同, 神经网络参数的训练通常对应于一个非凸优化问题. 然而在实际应用中, 这些参数完全可以通过随机梯度下降 *(SGD)* 算法进行高效学习, 详见第 2.4 节.

注 2.4 (常数的显式表达)**.** 我们在如下设定下给出定理 2.2 中常数 $C_{T,K,f}$ 的一个显式上界*:* 设 $f \in \mathcal{H}^{d/2+3}_{\text{loc}}$, 并且对于某个 $L > 0$, 向量场 f 关于空间变量一致 *L-Lipschitz*. 定义

$$\mathcal{F}_{L,d} := \left\{ f \in \mathcal{H}^{d/2+3}_{\text{loc}}(\mathbb{R}^{d+1}; \mathbb{R}^d) \;\middle|\; f(\cdot, t) \text{ 对所有 } t \text{ 在 } x \text{ 上为 } L\text{-Lipschitz} \right\}.$$

为简化表述, 固定初始值区域 $K = [-1, 1]^d$ 以及时间区间 $T \geq 1$. 则对于任意 $f \in \mathcal{F}_{L,d}$, 系统 (2.4) 的可达集 *(*包含时间变量*)* 包含于

$$\Omega_{L,T,d} := [-Te^{LT}, Te^{LT}]^{d+1}.$$

存在一个仅依赖于维数 d 的常数 $C_d > 0$, 使得对于每个 $f \in \mathcal{F}_{L,d}$, 定理 2.2 中的常数 $C_{T,K,f}$ 满足

$$C_{T,K,f} \leq C_d\, T \left\|f\right\|_{\mathcal{H}^{\frac{d}{2}+3}(\Omega_{L,T,d})} \exp\left(\tfrac{5}{2}LT + \sqrt{d}\, L + C_d\, e^{\frac{3}{2}LT} \left\|f\right\|_{\mathcal{H}^{\frac{d}{2}+3}(\Omega_{L,T,d})}\right). \tag{2.6}$$

其详细证明见第 2.3.3 节. 下面对式 (2.6) 中的依赖关系作简要说明*:*

- *(*维数依赖*)* 因子 C_d 来自于超立方体 $[-1, 1]^d$ 上的 *Barron* 型逼近常数, 其一般不存在简单的闭式表达. 总体而言, $C_{T,K,f}$ 对 d 呈指数依赖, 因

此在逼近速率 (1.5) 的分子中出现了维数灾难效应. 然而, 正如注记 2.3 所讨论的, 网络误差关于 P 的阶为 $P^{-1/2}$, 而经典 P_1-*FEM* 的误差阶为 $P^{-2/(d+1)}$. 因此, 当固定 $d \geq 4$ 且 P 足够大时, 网络逼近在渐近意义下更优.

- *(*时间依赖*)* 对于固定的向量场 $f \in \mathcal{F}_{L,d}$, 可以观察到常数 $C_{T,K,f}$ 随时间呈超指数级增长. 这是由于可达域随 T 呈指数扩张, 从而在该域上对 f 的逼近误差也会随着域尺度的放大而增加; 再结合 *Grönwall* 不等式, 最终导致了总体的双重指数增长. 若 *ODE* 系统具有 *Lyapunov* 稳定性, 则可达集可被一致控制, 从而可能获得更好的时间方向增长估计. 在实践中, 可以通过模型预测控制 *(MPC)* 策略来缓解这种误差爆炸 [84]; 参见注记 2.6.

- *(*函数范数依赖*)* 在固定的时间区间 T 下, 常数 $C_{T,K,f}$ 对 f 在可达域上的 *Sobolev* 范数呈指数依赖. 其原因在于所学习到的 *(SA-NODE)* 向量场的 *Lipschitz* 常数会随该范数增长, 并被带入到 *Grönwall* 指数项中. 若使用更高阶的 *ReLU* 激活函数来更好地逼近 f 的导数, 则有望获得更紧致的常数界限 *(*参见 *[14,* 定理 *3])*.

将定理 2.2 应用于与系统 (2.4) 相关联的传输方程 (2.7), 即可推导出我们的第三个主要结果 (定理 2.3): 关于方程 (2.7) 的解被其对应的神经网络模型 (2.8) 逼近时的一致收敛速率. 传输方程如下:

$$\begin{cases} \partial_t \rho + \operatorname{div}_x(f(x,t)\,\rho) = 0, \quad (x,t) \in \mathbb{R}^d \times (0,T), \\ \rho(\cdot,0) = \rho_0 \in \mathcal{M}(\mathbb{R}^d), \end{cases} \tag{2.7}$$

其中主要未知量为 ρ: $\mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}$, $\mathcal{M}(\mathbb{R}^d)$ 表示符号测度空间. 类似地, 与 (2.1) 相对应的传输方程 (即所谓神经传输方程 [80]) 为:

$$\begin{cases} \partial_t \rho + \operatorname{div}_x \left(\left(\sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 x + A_i^2 t + B_i) \right) \rho \right) = 0, \quad (x,t) \in \mathbb{R}^d \times [0,T], \\ \rho(\cdot,0) = \rho_0 \in \mathcal{M}(\mathbb{R}^d). \end{cases} \tag{2.8}$$

NODE 与传输方程之间的联系并非新近现象, 在归一化流 (Normalizing Flows) 理论中亦自然出现 [85-86]. 特别地, 文献 [80] 研究了在终端时刻对 (2.7) 的分布由方程 (2.8) 逼近的问题. 在下述定理中, 我们将该结果推广到了整个时间区间上的一致逼近. 回顾概率测度 Wasserstein-1 距离的定义如下 [87, 定义 6.1].

定义 2.1 (Wasserstein-1 距离)**.** 记

$$\mathcal{P}_1(\mathbb{R}^d) := \left\{ \mu \in \mathcal{P}(\mathbb{R}^d) : \int_{\mathbb{R}^d} \|x\| \, \mathrm{d}\mu(x) < \infty \right\},$$

即 $\mathbb{R}^d$ 上所有具有有限一阶矩的概率测度所构成的集合. 对于任意 $\mu, \nu \in \mathcal{P}_1(\mathbb{R}^d)$, 定义

$$\Pi(\mu, \nu) := \left\{ \pi \in \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) : (\mathrm{pr}_1)_{\#}\pi = \mu, \ (\mathrm{pr}_2)_{\#}\pi = \nu \right\},$$

其中 $\mathrm{pr}_1, \mathrm{pr}_2 : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ 分别表示到第一, 第二个分量的投影映射, $(\cdot)_{\#}$ 表示推前测度. 则 μ 与 ν 之间的 *Wasserstein-1* 距离定义为

$$\mathbb{W}_1(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\| \, \mathrm{d}\pi(x, y).$$

该距离刻画了在所有以 μ 和 ν 为边缘分布的耦合测度中, 将单位质量从 x 输运到 y 所需平均代价 $\|x - y\|$ 的最小可能值.

假设 2.3. 初始数据 ρ_0 为具有紧支撑的概率测度.

定理 2.3. 设假设 *2.1*-*2.3* 成立. 则对任意 $P \geq 3$, 存在参数 $\Theta = \{(W_i, A_i^1, A_i^2, B_i)\}_{i=1}^P$ 使得

$$\sup_{t \in [0,T]} \mathbb{W}_1(\rho(\cdot, t), \rho_\Theta(\cdot, t)) \leq \frac{C_{T,f,\rho_0}}{\sqrt{P}},$$

其中 C_{T,f,ρ_0} 为独立于 P 的常数, $\mathbb{W}_1(\cdot, \cdot)$ 为 *Wasserstein-1* 距离, 且 $\rho(\cdot, t)$*(*相应地, $\rho_\Theta(\cdot, t)$*)* 表示在时刻 $t \in [0, T]$ 时方程 (2.7)*(*相应地, (2.8)*)* 的解.

注 2.5 (更锐利的 Sobolev 指数)**.** 假设 *2.2* 中出现的 *Sobolev* 正则性指数 $(d+1)/2+2$ 源于引理 *2.3* 所建立的 *Sobolev* 空间到 *Barron* 空间的连续嵌入. 需要指出的是, 文献 *[88,* 定理 *1]* 基于 *Radon* 变换技术证明了该嵌入结果的一个更锐利的版本. 在应用该精化结论后, 假设 *2.2* 中的正则性要求可被放宽为 $k \geq (d+1)/2 + 3/2$; 在此改进下, 定理 *2.2* 与 *2.3* 的结论依然保持不变.

注 **2.6** (模型预测控制 (MPC) 视角)**.** 如注记 *2.4* 所述, 误差在理论上会随着时间快速爆炸. 因此, 即便选取非常大的网络宽度 P 以确保初始逼近误差很小, 该误差仍会随 T 呈超指数级增长; 传输方程的情形亦是如此. 一种可行的实践策略是采用模型预测控制视角: 与其训练单个 *SA-NODE* 去覆盖整个区间 $[0,T]$, 不如在长度为 τ 的连续短时间窗上不断更新或微调网络参数. 该策略可视为 *SA-NODE* ($\tau = T$) 与经典 *NODE* ($\tau \to 0$) 之间的一种折中方案. 探索如何选择最优的时间步长 τ 仍是值得进一步研究的重要方向.

注 **2.7** ($\mathbb{L}^p$-范数下的逼近)**.** 定理 *2.3* 在 *Wasserstein* 意义下给出了逼近误差. 当初始分布具有 $\mathbb{L}^p$ 密度时, 解 ρ 属于 $\mathcal{C}([0,T];\mathbb{L}^p(\mathbb{R}^d))$. 此外, 当向量场 f 在 $\mathcal{W}^{1,\infty}$-范数下被神经网络逼近时 (即同时控制函数及其梯度), 我们可以利用经典的能量方法在 $\mathbb{L}^p$ 意义下估计逼近误差 [6]. 在这种情形下, 可以进一步应用文献 [14, 定理 3] 中更强的逼近结果.

2.3 主要结果的证明

本节给出本章主要结果的证明.

2.3.1 定理 2.1 的证明

证明基于文献 [8] 中 Pinkus 给出的如下万能逼近结果. 该结果将 Cybenko 的经典定理 [5] 推广到了非多项式激活函数. 为便于读者查阅, 我们在此给出与本文情形相适配的表述.

定理 2.4 ([8])**.** 取任意紧集 $X \subseteq \mathbb{R}^{d+1}$. 设 σ 为非多项式的连续函数. 则对任意 $g \in \mathcal{C}(X;\mathbb{R}^d)$ 与任意 $\varepsilon > 0$, 存在参数 $(W_i, A_i, B_i) \in \mathbb{R}^d \times \mathbb{R}^{(d+1)\times d} \times \mathbb{R}^d (i = 1,\dots,P)$, 使得记

$$f_\Theta(x) = \sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i x + B_i), \quad \forall x \in X,$$

则有

$$\|g - f_\Theta\|_{\mathcal{C}(X;\mathbb{R}^d)} \le \varepsilon.$$

我们还需要如下关于 SA-NODE (2.1) 解的先验界引理.

引理 2.1 (先验界). 设假设 *2.1* 成立. 对任意 $t \in [0,T]$, 定义

$$K_t := \left\{ x \in \mathbb{R}^d \,\middle|\, \|x\| \le \sup_{z\in K} \left(\|z\| + t + \int_0^t \|f(0,s)\| ds \right) \exp(Lt) \right\}.$$

若 $f_1 \in \mathcal{C}(\mathbb{R}^d \times [0,T];\, \mathbb{R}^d)$ 关于 x 局部 *Lipschitz,* 且满足 $\|f_1 - f\|_{\mathbb{L}^\infty(K_T\times[0,T];\,\mathbb{R}^d)} \le 1$, 并且 $\boldsymbol{y}$ 满足

$$\dot{\boldsymbol{y}} = f_1(\boldsymbol{y}, t), \quad \boldsymbol{y}(0) = z_0 \in K,$$

则对任意 $t \in [0,T]$ 都有 $\boldsymbol{y}(t) \in K_t$.

证明: 证明可由标准的自举原理 (bootstrap principle) 得到 (参见 [89, 命题 1.21]). 对任意 $t \in [0,T]$, 记 $\mathbf{H}(t)$ 为如下假设: 对任意 $s \in [0,t]$, $\|f_1(\boldsymbol{y}(s),s) - f(\boldsymbol{y}(s),s)\| \le 1$; 记 $\mathbf{C}(t)$ 为如下结论: 对任意 $s \in [0,t]$, 有 $\boldsymbol{y}(s) \in K_s$.

首先, $\mathbf{H}(0)$ 显然成立. 其次, 由 Grönwall 不等式可得: $\mathbf{H}(t)$ 推出 $\mathbf{C}(t)$. 另一方面, 由 f_1 的假设与 K_t 的定义可知: 若 $\mathbf{C}(t)$ 成立, 则存在 t 的一个邻域, 使得该邻域内任意 $t' \in [0,T]$ 都有 $\mathbf{H}(t')$ 成立. 由于 K_t 为紧集且随 t 连续变化, 命题 $\mathbf{C}(t)$ 关于 t 具有闭性. 据此可由 [89, 命题 1.21] 的自举论证 (bootstrapping argument) 推出结论. □

下面证明定理 2.1. 固定任意 $0 < \varepsilon < 1$. 将定理 2.4 应用于引理 2.1 中定义的紧集 K_T 上的向量场 f, 可找到 P 以及 $W_i \in \mathbb{R}^d$, $A_i = (A_i^1, A_i^2) \in \mathbb{R}^{(d+1)\times d}$, $B_i \in \mathbb{R}^d$, 使得

$$f_\Theta(x,t) = \sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 x + A_i^2 t + B_i)$$

在 $\mathbb{L}^\infty(K_T \times [0,T];\, \mathbb{R}^d)$ 范数下以误差 ε 逼近 f. 由于 $\varepsilon < 1$, 由引理 2.1 可知对任意 $t \in [0,T]$ 都有 $\boldsymbol{x}_{z_0}(t) \in K_T$. 因此, 利用 f 关于空间变量的一致 Lipschitz 连续性, 有

$$\begin{aligned}
\|\boldsymbol{z}_{z_0}(t) - \boldsymbol{x}_{z_0}(t)\| &= \left\| z_0 + \int_0^t f(\boldsymbol{z}_{z_0}(s), s) ds - z_0 - \int_0^t f_\Theta(\boldsymbol{x}_{z_0}(s), s) ds \right\| \\
&\le \int_0^t \|f(\boldsymbol{z}_{z_0}(s), s) - f(\boldsymbol{x}_{z_0}(s), s) + f(\boldsymbol{x}_{z_0}(s), s) - f_\Theta(\boldsymbol{x}_{z_0}(s), s)\|\, ds \\
&\le L \int_0^t \|\boldsymbol{z}_{z_0}(s) - \boldsymbol{x}_{z_0}(s)\| ds + \varepsilon t,
\end{aligned}$$

该不等式对任意 $t \le T$ 成立. 再次应用 Grönwall 引理得到

$$\|\boldsymbol{z}_{z_0} - \boldsymbol{x}_{z_0}\|_{\mathbb{L}^\infty([0,T];\mathbb{R}^d)} \le \varepsilon T e^{LT}.$$

将 ε 重新记号 (吸收常数因子) 即可得到结论.

2.3.2 Barron 空间中的逼近速率

固定任意紧集 $X \subseteq \mathbb{R}^n (n \in \mathbb{N}_+)$. 回顾 [10, 公式 1] 中关于 X 上 Barron 空间的定义:

$$\mathcal{S}_{\mathrm{B}}(X) \coloneqq \Big\{ f \in \mathcal{C}(X) \,\Big|\, \exists \mu \in \mathcal{P}(\mathbb{R}^{n+2}) \\ \text{s.t. } f(x) = \int_{\mathbb{R}^{n+2}} w\sigma(\langle a, x\rangle + b) d\mu(w,a,b),\ \forall x \in X \Big\}, \tag{2.9}$$

其中 $\mathcal{P}(\mathbb{R}^{n+2})$ 表示 $\mathbb{R}^{n+2}$ 上所有 Borel 概率测度的集合.

下面回顾一个重要结果: 它刻画了一类属于 Barron 空间的函数, 并给出了浅层神经网络在一致范数下的统一逼近速率. 该引理可视为对 [81, 定理 2] 的一个改进.

引理 2.2. 令 $X = [-1,1]^n$. 设 $f \in \mathcal{C}(X)$ 存在某个延拓 $\bar{f} \in \mathbb{L}^1(\mathbb{R}^n)$, 且其 *Fourier* 变换满足

$$v_{f,2} \coloneqq \int_{\mathbb{R}^n} \|\xi\|_{\ell^1}^2 \left|\mathcal{F}(\bar{f})(\xi)\right| d\xi < \infty. \tag{2.10}$$

则 $f \in \mathcal{S}_{\mathrm{B}}(X)$. 此外, 对任意整数 $P \ge 3$, 存在 $(w_i, a_i, b_i) \in \mathbb{R}^{n+2} (i = 1, \dots, P)$ 使得

$$\Big\| f - \sum_{i=1}^{P} w_i\, \sigma\big(\langle a_i, \cdot\rangle + b_i\big) \Big\|_{\mathcal{C}(X)} \le \frac{C_n\, v_{f,2}}{\sqrt{P}}, \quad \text{且}$$

$$\mathrm{Lip}\left(\sum_{i=1}^{P} w_i\, \sigma\big(\langle a_i, \cdot\rangle + b_i\big) \right) \le \|\nabla f(0)\| + 2\, v_{f,2},$$

其中 $C_n > 0$ 仅依赖于维数 n.

证明: 对任意 $P \ge 1$, [81, 定理 2] 给出了参数

$$w_i \in \big[-2v_{f,2}/P,\ 2v_{f,2}/P\big], \quad \|a_i\|_1 = 1, \quad b_i \in [-1,1],$$

使得

$$\Big\| f(x) - \Big(f(0) + \langle \nabla f(0), x \rangle + \sum_{i=1}^{P} w_i \, \sigma(\langle a_i, x \rangle + b_i) \Big) \Big\|_{\mathcal{C}(X)} \;\le\; \frac{C_n}{\sqrt{P}},$$

其中 $C_n > 0$ 仅依赖于 n. 注意到仿射项可由两个 ReLU 神经元表示:

$$f(0) + \langle \nabla f(0), x \rangle = \sigma\big(\langle \nabla f(0), x \rangle + f(0)\big) - \sigma\big(\langle -\nabla f(0), x \rangle - f(0)\big),$$

因此当 $P \ge 3$ 时即可得到所述误差界. 最后, 由于 $\|a_i\|_2 \le \|a_i\|_1 = 1$, 网络的 Lipschitz 常数满足

$$\|\nabla f(0)\| \;+\; \sum_{i=1}^{P} |w_i| \, \|a_i\|_2 \;\le\; \|\nabla f(0)\| + 2\, v_{f,2}.$$

结论得证. □

满足 (2.10) 的函数类在文献中常被称为 Fourier-Lebesgue 空间. 下面我们证明: 当光滑度参数 k 足够大时, Sobolev 空间 $\mathcal{H}^k(X)$ 连续嵌入到该 Fourier-Lebesgue 空间中, 从而也包含于 $\mathcal{S}_{\mathrm{B}}(X)$ 内. 关于该嵌入的更锐利版本, 可参见 [88] (基于 Radon 变换; 见注记 2.5).

引理 2.3. 令 $X = [-1,1]^n$. 对任意 $f \in \mathcal{H}^k(X)$, 若 $k > n/2 + 2$, 则有

$$v_{f,2} \le C_{n,k} \, \|f\|_{\mathcal{H}^k(X)},$$

其中 $v_{f,2}$ 定义于 (2.10), 常数 $C_{n,k} > 0$ 仅依赖于 (n,k).

证明: 由于 $X = [-1,1]^n$ 满足强局部 Lipschitz 条件 (见 [82, 定义 4.9]), 且 $f \in \mathcal{H}^k(X)$ 并满足 $k > n/2 + 2$, 由 [82, 定理 4.12] 可得 $f \in \mathcal{C}^2(X)$. 此外, 由 [82, 定理 5.24], 存在延拓 $\bar{f} \in \mathcal{H}^k(\mathbb{R}^n)$ 使得 $\bar{f}|_X = f$. 记 $\mathcal{F}(\bar{f})$ 为 $\bar{f}$ 的 Fourier 变换. 由 Cauchy-Schwarz 不等式,

$$\begin{aligned}
\int_{\mathbb{R}^n} \|\xi\|^2 |\mathcal{F}(\bar{f})(\xi)| d\xi &\le \int_{\mathbb{R}^n} \big(1 + \|\xi\|^2\big) \, |\mathcal{F}(\bar{f})(\xi)| d\xi \\
&\le \left(\int_{\mathbb{R}^n} \big(1 + \|\xi\|^2\big)^{2-k} \, d\xi \right)^{\frac{1}{2}} \left(\int_{\mathbb{R}^n} \big(1 + \|\xi\|^2\big)^{k} \, |\mathcal{F}(\bar{f})(\xi)|^2 d\xi \right)^{\frac{1}{2}} \\
&= \pi^{n/4} \, \frac{\Gamma(k - 2 - n/2)}{\Gamma(k-2)} \, \|\bar{f}\|_{\mathcal{H}^k(\mathbb{R}^n)},
\end{aligned}$$

其中 $\Gamma(\cdot)$ 为 Gamma 函数. 由于延拓算子 $E:\mathcal{H}^k(X)\to\mathcal{H}^k(\mathbb{R}^n)$ 有界, 且其算子范数仅依赖于 n,k, 再结合任意 $\xi\in\mathbb{R}^n$ 都满足 $\|\xi\|_{\ell^1}\le\sqrt{n}\,\|\xi\|$, 即可得到所需估计. □

回顾符号 $\circ$ 表示 Hadamard 乘积, 且 $\boldsymbol{\sigma}\colon\mathbb{R}^d\to\mathbb{R}^d$ 表示逐分量 ReLU 激活. 结合以上两个引理可得如下推论.

推论 2.1. 固定任意 $m\in\mathbb{N}$, 并令 $X_m=[-m,m]^n$. 设 $F\in\mathcal{H}^k(X_m;\mathbb{R}^d)$ 且 $k>n/2+2$. 则对任意 $P\ge 3$, 存在 $(W_i,A_i,B_i)\in\mathbb{R}^d\times\mathbb{R}^{d\times n}\times\mathbb{R}^d(i=1,\dots,P)$, 使得

$$\left\|F(\cdot)-\sum_{i=1}^{P}W_i\circ\boldsymbol{\sigma}(A_i\cdot+B_i)\right\|_{\mathcal{C}(X_m)}\le\frac{C_{n,k,m}\|F\|_{\mathcal{H}^k(X_m)}}{\sqrt{P}},\quad\text{且}$$

$$\operatorname{Lip}\left(\sum_{i=1}^{P}W_i\circ\boldsymbol{\sigma}(A_i\cdot+B_i)\right)\le\|\nabla F(0)\|_{\mathrm{F}}+C_{n,k,m}\,\|F\|_{\mathcal{H}^k(X_m;\mathbb{R}^d)},$$

其中 $C_{n,k,m}>0$ 仅依赖于 (n,k,m), 且 $\|\nabla F(0)\|_{\mathrm{F}}$ 为 *Jacobian* 矩阵 $\nabla F(0)$ 的 *Frobenius* 范数.

证明: 固定任意 $i\in\{1,\dots,n\}$, 定义伸缩函数

$$\tilde{F}_i(x)\;=\;F_i(m\,x),\qquad x\in X=[-1,1]^n.$$

于是

$$\|\tilde{F}_i\|_{\mathcal{H}^k(X)}\;\le\;m^{\,k-n/2}\,\|F_i\|_{\mathcal{H}^k(X_m)}.$$

由引理 2.3, 存在常数 $C_{n,k}>0$ 使得

$$v_{\tilde{F}_i,2}\le C_{n,k}\,\|\tilde{F}_i\|_{\mathcal{H}^k(X)}\le C_{n,k}\,m^{\,k-n/2}\,\|F_i\|_{\mathcal{H}^k(X_m)}.$$

再由引理 2.2, 存在 $((w_j^i,a_j^i,b_j^i)\in\mathbb{R}^{n+2})(j=1,\dots,P)$ 以及仅依赖于维数的常数 $C_n>0$, 使得

$$\Big\|\,\tilde{F}_i(\cdot)\;-\;\sum_{j=1}^{P}w_j^i\,\sigma\big(\langle a_j^i,\,\cdot\,\rangle+b_j^i\big)\Big\|_{\mathcal{C}(X)}\;\le\;\frac{C_n\,v_{\tilde{F}_i,2}}{\sqrt{P}}\;\le\;\frac{C_n\,C_{n,k}\,m^{\,k-n/2}\,\|F_i\|_{\mathcal{H}^k(X_m)}}{\sqrt{P}},$$

$$\operatorname{Lip}\left(\sum_{j=1}^{P}w_j^i\,\sigma\big(\langle a_j^i,\,\cdot\,\rangle+b_j^i\big)\right)\le m\|\nabla F_i(0)\|+2C_{n,k}\,m^{\,k-n/2}\,\|F_i\|_{\mathcal{H}^k(X_m)}.$$

由 $\tilde{F}_i$ 的定义可得

$$\Big\| F_i(\cdot) \;-\; \sum_{j=1}^{P} w_j^i\,\sigma\big(\langle a_j^i/m,\,\cdot\,\rangle + b_j^i\big)\Big\|_{\mathcal{C}(X_m)} \;\le\; \frac{C_n\,C_{n,k}\,m^{k-n/2}\,\|F_i\|_{\mathcal{H}^k(X_m)}}{\sqrt{P}}.$$

并且

$$\mathrm{Lip}\left(\sum_{j=1}^{P} w_j^i\,\sigma\big(\langle a_j^i/m,\,\cdot\,\rangle + b_j^i\big)\right) \le \|\nabla F_i(0)\| + 2C_{n,k}\,m^{\,k-1-n/2}\,\|F_i\|_{\mathcal{H}^k(X_m)}.$$

最后, 由向量值函数范数与矩阵范数的定义即可推出所需估计. □

注 2.8 ($\mathbb{L}^\infty$-逼近速率)**.** 引理 2.2 与推论 2.1 在 $\mathbb{L}^\infty$ 范数下给出了浅层神经网络的万能逼近速率; 我们的主要技术取自 *[81]*. 我们也注意到, 在 *[90,* 命题 *1]* 与 *[14,* 定理 *3]* 中, 借助几何差异理论 *(geometric discrepancy theory)* 的深刻工具 *[91]*, 可以得到更优的速率 $P^{-1/2-3/2n}$*(*在 *SA-NODE* 情形下, $n = d+1$*)*, 并且可将网络的 *Lipschitz* 常数一致控制 *(*与 P 无关*)*. 因此, 定理 2.2 中的收敛速率同样可以改进为

$$P^{-\frac{1}{2}-\frac{3}{2(d+1)}},$$

其中 d 为动力系统的维数.

注 2.9 ($\mathbb{L}^2$-逼近速率)**.** *ReLU* 网络在 $\mathbb{L}^2$ 意义下的逼近速率 *(*同样为 $P^{-1/2}$ 量级*)* 可由 *Hölder* 不等式与前述 $\mathbb{L}^\infty$ 结果直接推出. 另一种证明思路是使用 *Maurey* 不等式 *[92,* 引理 *2](*亦可参见 *[10])*. 该方法对激活函数的要求更弱, 不局限于 *ReLU* 及其幂函数 *(*后者在 $\mathbb{L}^\infty$ 结果中使用*)*. 特别地, *[93,* 定理 *4]* 证明: 当激活函数 σ 二次弱可微且满足可积性条件

$$\int_{\mathbb{R}} \big|\sigma''(x)\big|\,(1+|x|)\,dx < \infty, \tag{2.11}$$

则在 $\mathbb{L}^2$ 范数下仍成立 $P^{-1/2}$ 的逼近速率. *Sigmoid* 函数满足 (2.11). 因此, 借助下一小节中平行的论证, 定理 2.2 亦可被改写为: 对所有满足 (2.11) 的 σ, 关于初值 z_0 的 $\mathbb{L}^2$ 误差同样具有 $P^{-1/2}$ 的收敛速率.

2.3.3 定理 2.2 的证明

证明分两步进行.

步骤 1 (对 f 的逼近). 在假设 2.1 下, 系统 (2.4) 的可达集

$$\Omega_T(K) = \left\{ \boldsymbol{z}_{z_0}(t) \;\middle|\; z_0 \in K,\ t \in [0,T] \right\}$$

为紧集. 取

$$m = T \max\left\{ 1,\ \sup_{z_0 \in K} \|z_0\|\, e^{LT} \right\}, \tag{2.12}$$

由 Grönwall 引理可得

$$X_m := [-m,m]^{d+1} \ \supseteq\ \Omega_T(K) \times [0,T].$$

由假设 2.2,

$$f|_{X_m} \in \mathcal{H}^k(X_m;\mathbb{R}^d), \quad \text{其中} k > (d+1)/2 + 2.$$

因此, 由推论 2.1, 对任意 $P \geq 3$, 存在参数 $\Theta = (W_i, A_i^1, A_i^2, B_i)_{i=1}^P$ 使得

$$\|f(\cdot,\cdot) - f_\Theta(\cdot,\cdot)\|_{\mathcal{C}(X_m;\mathbb{R}^d)} \leq \frac{C_{d,k,m}\|f\|_{\mathcal{H}^k(X_m)}}{\sqrt{P}}, \tag{2.13}$$

$$\mathrm{Lip}\left(f_\Theta(\cdot,\cdot)\right) \leq \|\nabla f(0,0)\|_{\mathrm{F}} + C_{d,k,m}\|f\|_{\mathcal{H}^k(X_m)}, \tag{2.14}$$

其中 $C_{d,k,m} > 0$ 仅依赖于 (d,k,m).

步骤 2 (误差分解与估计). 对任意 $(z_0,t) \in K \times [0,T]$, 由三角不等式,

$$\begin{aligned}
&\|\boldsymbol{z}_{z_0}(t) - \boldsymbol{x}_{z_0}(t)\| \\
&= \left\| \int_0^t f(\boldsymbol{z}_{z_0}(s),s) - f_\Theta(\boldsymbol{z}_{z_0}(s),s) + f_\Theta(\boldsymbol{z}_{z_0}(s),s) - f_\Theta(\boldsymbol{x}_{z_0}(s),s)\, ds \right\| \\
&\leq \underbrace{\int_0^t \|f(\boldsymbol{z}_{z_0}(s),s) - f_\Theta(\boldsymbol{z}_{z_0}(s),s)\|\, ds}_{=:\gamma_1} + \underbrace{\int_0^t \|f_\Theta(\boldsymbol{z}_{z_0}(s),s) - f_\Theta(\boldsymbol{x}_{z_0}(s),s)\|\, ds}_{=:\gamma_2}.
\end{aligned}$$

由于对所有 $s \in [0,T]$ 都有 $\boldsymbol{z}_{z_0}(s) \in \Omega_T$, 从而 $(\boldsymbol{z}_{z_0}(s),s) \in X_m$. 因此由 (2.13), 对任意 $t \in [0,T]$,

$$\gamma_1 \ \leq\ \underbrace{C_{d,k,m}\,\|f\|_{\mathcal{H}^k(X_m)}}_{=:C_1}\ \frac{t}{\sqrt{P}}.$$

另一方面, 由 (2.14),

$$\gamma_2 \leq \underbrace{\left(\|\nabla f(0,0)\|_{\mathrm{F}} + C_{d,k,m}\|f\|_{\mathcal{H}^k(X_m)}\right)}_{=:C_2} \int_0^t \|\boldsymbol{z}_{z_0}(s) - \boldsymbol{x}_{z_0}(s)\|\, ds.$$

注意到常数 C_1 与 C_2 仅依赖于 T, K 与 f: 其中 d 为状态维数, 且 m 由 (2.12) 给出, 是 T, K 以及 f 的 Lipschitz 常数的显式函数. 将上述不等式合并可得, 对所有 $(z_0, t) \in K \times [0, T]$,

$$\|\boldsymbol{z}_{z_0}(t) - \boldsymbol{x}_{z_0}(t)\| \;\le\; C_1 \frac{t}{\sqrt{P}} \;+\; C_2 \int_0^t \|\boldsymbol{z}_{z_0}(s) - \boldsymbol{x}_{z_0}(s)\| \,\mathrm{d}s.$$

对其应用 Grönwall 引理, 得到对任意 $z_0 \in K$,

$$\sup_{t\in[0,T]} \|\boldsymbol{z}_{z_0}(t) - \boldsymbol{x}_{z_0}(t)\| \le \frac{TC_1 e^{C_2 T}}{\sqrt{P}}.$$

从而定理 2.2 得证.

为完整起见, 我们给出注记 2.4 中显式常数估计的证明.

注记 2.4 的证明 在注记 2.4 的设定下, 上述证明中出现的常数 m 可具体取为

$$m = Te^{LT}.$$

因此相应的区域为

$$\Omega_{L,T,d} = X_m = [-m, m]^{d+1}.$$

由推论 2.1 的证明, 并利用 $f \in \mathcal{H}^{d/2+3}_{\mathrm{loc}}(\mathbb{R}^{d+1}; \mathbb{R}^d)$(即 $k = d/2+3$), 可将估计式 (2.13)–(2.14) 中的常数显式化:

$$\|f - f_\Theta\|_{\mathcal{C}(X_m;\mathbb{R}^d)} \le \frac{C_d\, m^{5/2}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})}}{\sqrt{P}},$$

$$\mathrm{Lip}\big(f_\Theta\big) \le \|\nabla f(0,0)\|_{\mathrm{F}} + C_d\, m^{3/2}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})},$$

其中 $C_d > 0$ 仅依赖于维数 d, 来源于引理 2.2 与引理 2.3 中常数的乘积 (取 $n = d+1, k = d/2+3$).

由于 f 在空间变量上为 L-Lipschitz, 故

$$\|\nabla f(0,0)\|_{\mathrm{F}} \le \sqrt{d}\, L.$$

将上述估计合并即可得到前述证明中 C_1 与 C_2 的显式形式:

$$C_1 = C_d\, m^{5/2}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})} = C_d\, e^{\frac{5LT}{2}}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})},$$

$$C_2 \le \sqrt{d}\, L + C_d\, m^{3/2}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})} = \sqrt{d}\, L + C_d\, e^{\frac{3LT}{2}}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})}.$$

因此

$$\begin{aligned} C_{T,K,f} &= TC_1 e^{C_2 T} \\ &\le C_d\, T\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})} \exp\!\left(\tfrac{5}{2}LT + \sqrt{d}\, L + C_d\, e^{\frac{3LT}{2}}\, \|f\|_{\mathcal{H}^{d/2+3}(\Omega_{L,T,d})}\right), \end{aligned}$$

即得到所需的显式上界 (2.6). □

2.3.4 定理 2.3 的证明

由假设 2.1 且 σ 为 ReLU, 可知

$$f, f_\Theta \in \mathbb{L}^1\left([0,T]; \mathcal{W}^{1,\infty}_{\mathrm{loc}}(\mathbb{R}^d;\mathbb{R}^d)\right), \quad \text{并且} \quad \frac{\|f\|}{1+\|x\|}, \frac{\|f_\Theta\|}{1+\|x\|} \in \mathbb{L}^1\left([0,T]; \mathbb{L}^\infty(\mathbb{R}^d)\right),$$

其中 $\mathcal{W}^{1,\infty}_{\mathrm{loc}}$ 表示局部 Sobolev 空间. 由 [94, 命题 4 及注记 7], 传输方程 (2.7) 与神经传输方程 (2.8) 的解可表示为

$$\rho(\cdot,t) = \phi_t \# \rho_0, \quad \rho_\Theta(\cdot,t) = \phi_{\Theta,t} \# \rho_0, \quad \forall t \in [0,T], \tag{2.15}$$

其中 # 表示前推算子, ϕ_t(相应地, $\phi_{\Theta,t}$) 为由方程 (2.4)(相应地, (2.1)) 生成的流映射: 将初始状态映到时刻 t 的解. 因此 $\rho(\cdot,t), \rho_\Theta(\cdot,t) \in \mathcal{P}(\mathbb{R}^d)$; 并且由于 $\mathrm{supp}(\rho_0)$ 为紧集, 结合 Grönwall 不等式可知它们在某个紧集内有界支撑. 因此可用 [87, 公式 6.3] 的对偶公式计算 Wasserstein-1 距离:

$$\mathbb{W}_1(\rho(\cdot,t), \rho_\Theta(\cdot,t)) = \sup_{\mathrm{Lip}(g)\le 1} \int_{\mathbb{R}^d} g(x)\, d\left(\rho(x,t) - \rho_\Theta(x,t)\right).$$

令 K 表示 ρ_0 的支撑集. 由 (2.15),

$$\begin{aligned} \mathbb{W}_1(\rho(\cdot,t), \rho_\Theta(\cdot,t)) &= \sup_{\mathrm{Lip}(g)\le 1} \int_K g(\phi_t(z)) - g(\phi_{\Theta,t}(z))\, d\rho_0(z) \\ &\le \int_K \|\phi_t(z) - \phi_{\Theta,t}(z)\|\, d\rho_0(z). \end{aligned}$$

对任意 $z \in K$, 由定理 2.2, 存在常数 $C_{T,K,f}$ 使得

$$\|\phi_t(z) - \phi_{\Theta,t}(z)\| \le \frac{C_{T,K,f}}{\sqrt{P}}.$$

代回即得结论.

2.4 半自治神经常微分方程的训练策略

本节旨在构建用于训练半自治神经常微分方程 (SA-NODE) (2.1) 参数的优化问题, 使其在时间区间 $[0,T]$ 上逼近给定的动力系统, 并且初始值取自某个紧集 K:

$$\begin{cases} \dot{\boldsymbol{z}}_{z_0} = f(\boldsymbol{z}_{z_0}, t), & t \in [0,T], \\ \boldsymbol{z}_{z_0}(0) = z_0, & z_0 \in K. \end{cases}$$

最直观的情形是: 向量场 $f(\boldsymbol{x},t)$ 在若干空间位置与时间采样点上是已知的. 在这种情况下, 可以使用浅层神经网络对 f 进行直接插值, 从而获得一个 SA-NODE.

然而在实际应用中, 我们通常无法直接获取向量场的样本. 更为常见的观测方式是: 传感器沿着由动力系统生成的流 $\boldsymbol{z}_{z_0}(t)$ 运动并记录其位置; 不同的初始状态 z_0 会产生不同的轨迹.

从这一视角来看, 训练任务本质上转化为一个反问题 (inverse problem): 其目标是从观测到的传感器轨迹中推断出底层的动力学规律. 为便于表述, 我们首先在连续数据设定 (即无限的传感器与连续的时间) 下探讨该训练问题. 这一设定自然引出了如 (2.16) 所示的最优控制形式. 在定理 2.5 中, 我们利用伴随变量 (adjoint variable) 推导了目标泛函的梯度; 该梯度表达式在实现基于梯度的优化方法中发挥着核心作用. 适用于有限训练数据集的离散化版本将在 (3.15) 中给出. 此外, 如注记 2.11 所述, 该训练框架亦可推广至传输方程的情形.

为了在连续数据情形下确定 SA-NODE (2.1) 的最优参数 $\Theta = (W, A^1, A^2, B)$, 我们考虑如下最优控制问题:

$$\begin{aligned} \inf_{\Theta} \quad & L(\Theta) = \int_0^T \int_K \left\| \boldsymbol{z}_{z_0}(t) - \boldsymbol{x}_{z_0}(t) \right\|^2 \mathrm{d}z_0 \, \mathrm{d}t + \lambda \, g(\Theta), \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}_{z_0}(t) = f_\Theta\big(\boldsymbol{x}_{z_0}(t), t\big), \ \boldsymbol{x}_{z_0}(0) = z_0, \ \forall\, z_0 \in K, \end{aligned} \tag{2.16}$$

其中 g 表示一般的正则化项, 其前带有一个正系数 λ, 且 f_Θ 为方程 (2.1) 的向量场. 尽管前文的逼近速率是在 $\mathbb{L}^\infty$-范数下建立的, 在此处我们仍选择 $\mathbb{L}^2$-残差作为保真项 (fidelity term). 这种选择在回归任务中属于标准做法, 并且更易于应用梯度下降算法进行优化.

关于正则化项 g 的选择, 我们提供以下几种可行方案. 首先, 参数 Θ 的 ℓ^p-范数是监督学习中最为经典的正则化形式. 其次, 利用 SA-NODE 的 Lipschitz 常数 (2.3) 能够有效地促进分布意义上的泛化能力, 相关讨论可参见 [95, 第 3 节]. 第三, 亦可采用与浅层神经网络相关的其他范数, 例如扩展的 Barron 范数, 变差范数 (variation norm) 以及 Radon-BV 半范数等; 关于它们之间等价性的讨论详见 [13].

将 $\boldsymbol{x}_{z_0}$ 视为关于 Θ 的隐函数, 依据经典的伴随方法 (adjoint method) [96, 第 261-265 页], 我们可以推导出损失函数 L 的梯度, 如下述定理所示.

定理 2.5. 对于任意 $(\Theta, x, t) \in \mathbb{R}^{2Pd(d+1)} \times \mathbb{R}^d \times [0,T]$, 设 $\tilde{f}(\Theta, x, t) = f_\Theta(x,t)$. 假设 g 是局部 *Lipschitz* 连续的. 则对于几乎所有的 Θ, 如下等式成立*:*

$$\nabla L(\Theta) = \int_0^T \int_K \frac{\partial \tilde{f}}{\partial \Theta}(\Theta, \boldsymbol{x}_{z_0}(t), t)^\top \boldsymbol{a}_{z_0}(t) dz_0 dt + \lambda \nabla g(\Theta),$$

其中 $\boldsymbol{x}_{z_0}$ 满足 *SA-NODE* (2.1), 且 $\boldsymbol{a}_{z_0}$ 满足伴随方程*:*

$$\begin{cases} -\dot{\boldsymbol{a}}_{z_0}(t) = \frac{\partial \tilde{f}}{\partial x}(\Theta, \boldsymbol{x}_{z_0}(t), t)^\top \boldsymbol{a}_{z_0}(t) + 2(\boldsymbol{x}_{z_0}(t) - \boldsymbol{z}_{z_0}(t)), & t \in [0,T], \\ \boldsymbol{a}_{z_0}(T) = 0, & z_0 \in K. \end{cases}$$

我们在此省略具体证明, 该结论可由 [96, 命题 1, 第 262 页] 直接推导得出. 对于固定的 z_0, 文献 [33, 定理 1] 亦证明了类似的结果. 定理 2.5 勾勒出了训练 SA-NODE 的一般流程: 即通过梯度下降算法来优化系数, 其中梯度正是通过求解该伴随方程来计算的.

注 2.10. 在本文的设定中, 激活函数 σ 为 *ReLU*. 因此, 定理 *2.5* 中的函数 $\tilde{f}$ 是局部 *Lipschitz* 连续的, 从而几乎处处关于 Θ 和 x 可导*;* 这意味着 ∇L 的上述表示公式对于几乎所有的 Θ 均成立. 在伴随方程中, 对于任意固定的 Θ, $\tilde{f}$ 关于 x 的 *Lipschitz* 连续性确保了对应向量场在 $\boldsymbol{a}_{z_0}$ 上具有一致有界的散度. 这保证了伴随方程的适定性 *(well-posedness)*.

最后, 由于在实际应用中不可能处理连续无穷多个初始点, 我们必须对损失函数中的积分进行离散化. 假设训练数据集具有结构 $\{\boldsymbol{z}_k(t_l)\}$, 其中 $k = 1, 2, \ldots, N$; $l = 1, 2, \ldots, M$: 这里 $\boldsymbol{z}_k$ 表示 N 条轨迹 (对应 N 个初始位置)

中的第 k 条轨迹, 而 t_l 表示总共 M 个时间步中的第 l 步. 由此, 我们得到了 (2.16) 的有限维离散对应形式:

$$\hat{L}(\Theta) = \frac{1}{NM} \sum_{k=1}^{N} \sum_{l=1}^{M} \left(\boldsymbol{z}_k(t_l) - \boldsymbol{x}_k(t_l, \Theta)\right)^2 + \lambda\, g(\Theta). \tag{2.17}$$

此处, $\boldsymbol{x}_k(t_l; \Theta)$ 是模型对第 k 条轨迹在时刻 t_l 的预测值. 在这种离散背景下, $\hat{L}$ 的梯度可以采用与定理 2.5 相平行的方式进行计算: 此时反向传播 (backpropagation) 算法在数值实现中便扮演了伴随方程的角色.

针对传输方程的训练, 我们利用以下注记, 将其训练策略还原到 ODE 的训练框架中.

注 2.11 (传输方程的训练策略)**.** 为了训练神经传输方程 (2.8) 的参数以逼近原始传输方程 (2.7), 我们考虑与 (2.8) 相关的对应特征线系统 *(characteristic system)*:

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = \displaystyle\sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 \boldsymbol{x} + A_i^2 t + B_i), \\ \dfrac{d\rho}{dt} = -\mathrm{div}_x \left(\displaystyle\sum_{i=1}^{P} W_i \circ \boldsymbol{\sigma}(A_i^1 \boldsymbol{x} + A_i^2 t + B_i) \right) \rho, \end{cases} \tag{2.18}$$

其中 ρ 是沿轨迹 $\boldsymbol{x}(t)$ 的流体密度. 由于激活函数 $\boldsymbol{\sigma}$ 是显式已知的, 上述第二个方程等价于:

$$\frac{d\rho}{dt} = -\rho \left(\sum_{i=1}^{P} \left\langle W_i,\, \mathrm{diag}(A_i^1) \boldsymbol{\sigma}' \left(A_i^1 \boldsymbol{x} + A_i^2 t + B_i \right) \right\rangle \right),$$

其中 $\mathrm{diag}(A_i^1)$ 为 A_i^1 的对角部分. 事实上, 通过将我们的 *ODE* 框架仅应用于 (2.18) 的第一行 *(*该行控制着轨迹的位置演化*)*, 我们便能还原参数的求解过程, 并产生与 (3.15) 相同的损失函数.

然而, 在传输问题的设定中, 我们还可以利用 (2.18) 第二行所提供的关于沿轨迹密度的额外信息. 将针对密度的残差项一并纳入损失函数中, 通常会带来提升 *SA-NODE* 泛化能力的附带优势. 在后续的数值实验中, 我们采用的正是这种增强型的损失函数公式.

2.5 数值实验

本节给出若干数值实验结果, 以展示半自治神经常微分方程 (SA-NODEs) 在精确模拟常微分方程与传输方程方面的能力. 此外, 我们还将 SA-NODEs (2.1) 与经典神经常微分方程 (1.2) 的表现进行了对比, 从误差与模型复杂度两个维度验证了 SA-NODEs 在此类任务中的有效性与精度优势. 所有代码均使用 Python 语言并基于 PyTorch 深度学习框架实现. 全部实验均在一台图形工作站上完成: 该工作站配备两颗 24 核 Intel Xeon Platinum 8269CY 处理器, 一块 Nvidia RTX A6000 图形处理器, 512GB 内存, 并运行搭载 PyTorch 的 Ubuntu 20.04 操作系统.

2.5.1 ODE 的模拟

训练与评估数据集由若干批轨迹组成: 我们首先采用四阶 Runge-Kutta 方法在时间区间 $[0, 5]$ 上, 以时间步长 0.05 对精确系统进行数值求解, 从而得到 "精确系统" 的轨迹作为数据. 初始条件从网格上进行采样: 在 z_1 与 z_2 两个维度上均取区间 $[-2, 2]$, 采样步长为 0.2. 由此共得到 441 条轨迹, 其中仅随机选取一半 (即 220 条轨迹) 用于训练. 需要说明的是, 为了保证图像的清晰度, 在后续的图中我们仅绘制了其中的 100 条轨迹. 我们将展示: 即使在数据量相对有限的情况下, SA-NODE 依然能够准确捕捉底层动力系统的主要行为.

在后续的插图中, 红色曲线表示 NODE 对训练集轨迹的模拟结果, 绿色曲线表示 NODE 对测试集轨迹的模拟结果. 绿色曲线对于评估模型的泛化能力尤为关键, 它反映了模型对未见初始值所对应动力学的预测能力. 神经网络采用单隐层结构, 隐层神经元数取 $P = 1000$, 激活函数为 ReLU. 训练阶段使用 Adam 优化算法, 初始学习率设为 10^{-3}; 每训练 1000 轮将学习率衰减为原来的 0.8 倍, 总训练轮次为 10000 轮. 损失函数 (3.15) 中的权重参数 λ 取 10^{-4}, 正则项 g 取为 (2.3) 中由 Lipschitz 常数所定义的正则化.

图 2.1 汇总了实验结果: 左图为 SA-NODE 的模拟轨迹, 中图为精确系统的轨迹, 右图为误差的均值与标准差. 其中, 第 k 条轨迹在时刻 t 的误差定义为 $e_k(t) = \|\boldsymbol{z}_k(t) - \boldsymbol{x}_k(t)\|$. 在图 2.1 的右侧子图中, 红色 (相应地, 蓝色) 曲线表示训练集 (相应地, 测试集) 误差 e_k 的均值; 灰色阴影带则表示测试集误

差 e_k 的标准差范围.

例 1: 非线性自治 ODE

非线性 ODE 由于其行为的复杂性与多样性而极具挑战性. 与线性系统 (其解的结构相对可预测且更易于分析) 不同, 非线性系统可能会出现极限环, 混沌与分岔等现象, 这使得理论分析与数值逼近都变得更加困难. 我们考虑的非线性 ODE 示例为无阻尼单摆, 其动力学方程为

$$\begin{cases} \dot{z}_1 = z_2, \\ \dot{z}_2 = -\sin(z_1). \end{cases} \tag{2.19}$$

如图 2.1a 与 2.1b 所示, SA-NODE 能够较好地刻画该动力系统的行为, 但在较长的时间跨度下其精度会呈现逐渐下降的趋势. 我们推测这与系统的 "双重性质" 有关: 系统的轨迹随初始值的不同, 可能呈现出周期轨道或无界轨道. 同时我们也注意到, 较差的表现主要集中在测试集上, 这意味着即使面对如此复杂的系统, SA-NODE 在训练数据范围内依然保持了良好的模拟性质.

例 2: 非线性非自治 ODE

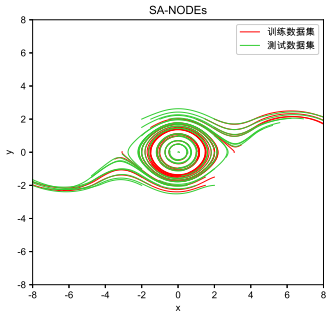
非线性非自治 ODE 可用于刻画多种复杂现象, 例如机械系统中的受迫振荡, 生态或生物系统中随时间变化的环境影响等. 这类 ODE 的求解与学习往往十分困难, 因为非线性与时间依赖性相互耦合, 极易导致分岔, 混沌以及对初始值的高度敏感等现象. 我们考虑如下非线性非自治系统:

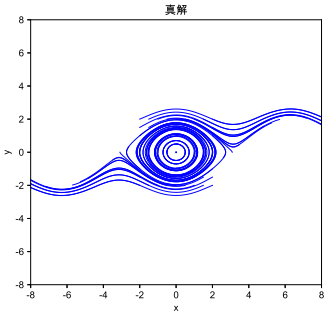
$$\begin{cases} \dot{z}_1 = z_2, \\ \dot{z}_2 = z_1 - z_1^3 + \delta\cos(\omega t). \end{cases} \tag{2.20}$$

该方程即著名的受迫 Duffing 方程, 常用于描述某些具有外部驱动的振子模型; 其中 δ 控制外部周期驱动项的强度, ω 为驱动力的角频率. 在接下来的实验中取 $\delta = 0.1, \omega = \pi$. 图 2.1c 表明 SA-NODE 能够很好地模拟该非线性非自治系统, 而图 2.1d 进一步展示了其极高的逼近精度.

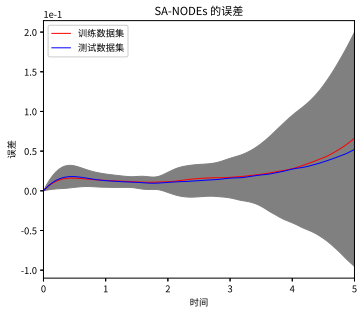
2.5.2 与经典 NODEs 的对比

本小节将比较经典 NODEs (1.2) 与 SA-NODEs (2.1) 的逼近表现. 比较主要围绕两个指标展开: 其一是模型的精度 (通过误差来衡量), 其二是模型

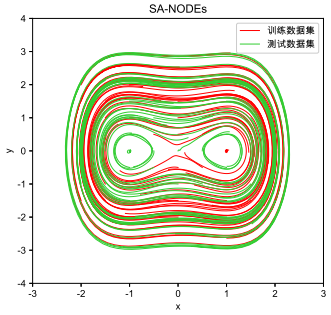


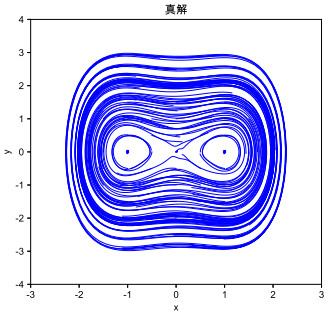


(a) 系统 (2.19) 的 SA-NODE 模拟与精确解对比.

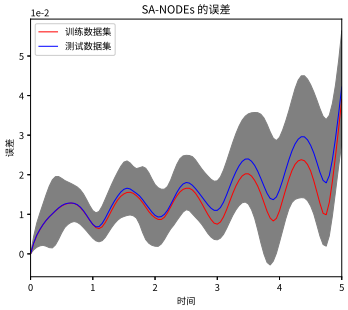


(b) 系统 (2.19) 的误差.





(c) 系统 (2.20) 的 SA-NODE 模拟与精确解对比.



(d) 系统 (2.20) 的误差.

图 2.1: ODE 系统的 SA-NODE 解, 精确解与误差汇总.

的复杂度 (以神经网络所需的参数数量来量化). 为保证比较的公平性, 我们对每个模型均训练了相同且足够大的轮数 (10^4 轮), 并采用了相同的学习率 (10^{-3}). 在该设定下, 唯一可调的 “瓶颈” 是每层使用的神经元数量 P. 此外, 两种方法在损失函数中均使用了所有 NODE 参数的 ℓ^1 范数作为正则项.

我们首先在图 2.2 与图 2.3 中分别给出了自治系统 (2.19) 与非自治系统 (2.20) 的数值比较结果. 其中, 图 2.2a 与图 2.3a 对比了经典 NODEs, SA-NODEs 与精确解的轨迹; 图 2.2b 与图 2.3b 则展示了测试误差随时间的演化过程. 可以清晰地观察到, SA-NODEs 在逼近精度与轨迹的光滑性方面整体表现更优.

为进一步提供量化的比较, 我们在表 2.1 中列出了不同网络规模下的误差与自由度. 其中, $e_{\max}$ 表示测试集中 “均值误差” 的最大值, e_T 表示终端时刻的均值误差. 回顾可知, P 为每层神经元数量, M 为时间离散步数, d 为问题的维数. 经典 NODEs 的自由度为 $(d+3)dMP$, 而 SA-NODEs 的自由度为 $(d+3)dP$. 由于 SA-NODEs 的参数数量与时间步数 M 完全无关, 因此当 M 较大时, 其模型复杂度得到了显著的降低.

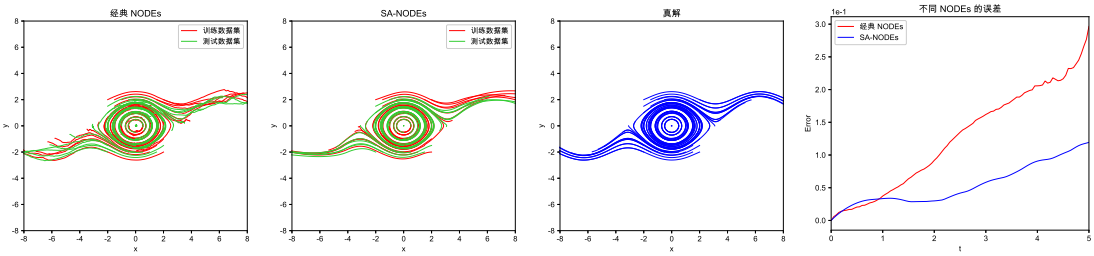
由表 2.1 可以看出: 在固定 P 时, SA-NODEs 的误差始终小于经典 NODEs, 并且其自由度大幅减少. 此外, 随着 P 的增大, 误差呈现单调下降的趋势, 这与定理 2.2 的理论结论高度一致.

进一步地, 我们还在不同训练轮数与不同训练集规模下评估了两类模型的表现. 在图 2.4 的左图中, 我们绘制了随训练轮数增加时两类模型的最大均值误差演化曲线, 结果表明 SA-NODEs 的收敛速度显著快于经典 NODEs. 在右图中, 我们给出了最大均值误差随训练集规模 (即轨迹条数) 变化的曲线: 通过改变网格步长 $\Delta x \in \{1.0, 0.5, 0.4, 0.2, 0.1\}$, 对应的训练轨迹数分别为 $12, 40, 60, 220$ 和 840. 结果表明: 相较于经典 NODEs, SA-NODEs 在利用更少轨迹数据的情况下即可达到收敛状态. 我们将综合比较的结论总结如下:

1. 在固定训练集规模的条件下, SA-NODEs 的训练收敛显著更快, 从而大大降低了计算成本.

2. 在小样本训练集的情形下, SA-NODEs 持续优于经典 NODEs, 在数据稀缺的场景中展现出明显的优势.

3. 当训练集规模与训练轮数均足够大时, 两类模型的最终性能趋于相近.

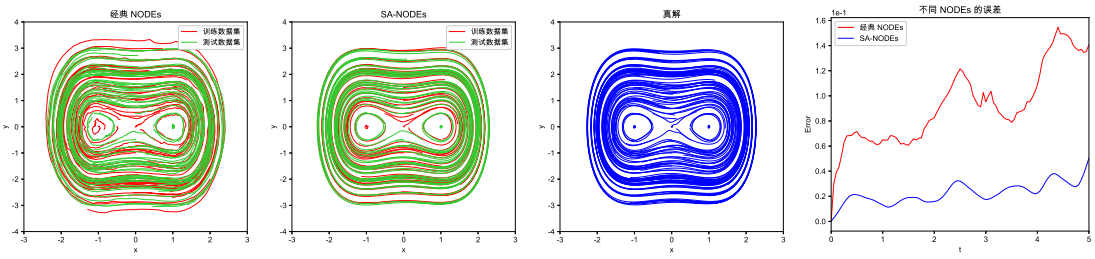


(a) 经典 NODEs, SA-NODEs 与精确解对比. (b) 测试误差.

图 2.2: 系统 (2.19) 上经典 NODEs 与 SA-NODEs 的解与误差对比.

2.5.3 传输方程的模拟

本小节将 SA-NODEs 应用于模拟传输方程的解, 从而验证定理 2.3 所探讨的逼近能力. 我们首先给出一个非自治传输方程的简单算例, 用以说明注记 2.11 中提及的训练策略. 在相同的方法框架下, 我们进一步考察了模型在 Doswell 锋生算例 [97] 中的逼近表现. 该训练方法的核心是将传输方程转



(a) 经典 NODEs, SA-NODEs 与精确解对比. (b) 测试误差.

图 2.3: 系统 (2.20) 上经典 NODEs 与 SA-NODEs 的解与误差对比.

P	神经常微分方程	自治情形			非自治情形		
		$e_{\max}$	e_T	自由度	$e_{\max}$	e_T	自由度
100	经典 NODEs	1.88e-01	1.88e-01	1e+06	1.17e+00	9.93e-02	1e+06
	SA-NODEs	9.78e-02	9.78e-02	1e+03	5.46e-02	5.46e-02	1e+03
500	经典 NODEs	1.69e-01	1.69e-01	5e+06	9.62e-02	9.62e-02	5e+06
	SA-NODEs	8.97e-02	8.97e-02	5e+03	3.61e-02	3.61e-02	5e+03
1000	经典 NODEs	1.52e-01	1.52e-01	1e+07	9.57e-02	9.57e-02	1e+07
	SA-NODEs	6.55e-02	6.55e-02	1e+04	3.44e-02	3.44e-02	1e+04

表 2.1: 自治与非自治 ODE 下, 经典 NODEs 与 SA-NODEs 的误差与自由度对比.

化为其对应的特征线 ODE 系统 (见注记 2.11), 这一转化过程需要计算激活函数的导数. 因此, 在本小节的实验中, 我们使用 Sigmoid 激活函数来代替 ReLU, 以确保其处处可微性. 得益于注记 2.9 的理论支撑, 基于 Sigmoid 的 SA-NODE 在整体意义上的万能逼近能力与基于 ReLU 的版本是相当的. 在以下的实验中, 我们设定每层神经元数量 $P = 200$. 学习率初始化为 10^{-3}, 并通过调度器进行动态调整: 每训练 10000 轮将学习率衰减为原来的 0.8 倍, 总训练轮次为 50000 轮.

例 3: 非自治传输方程

我们重点研究如下二维非自治传输方程:

$$\begin{cases} \partial_t \rho(x,y,t) + \operatorname{div}\left(\left(\dfrac{\sin(x)}{1+t^2}, \dfrac{\sin(y)}{1+t^2}\right)\rho(x,y,t)\right) = 0, & (x,y,t) \in \mathbb{R}^2 \times [0,T], \\ \rho(\cdot,0) = \rho_0 \in \mathcal{M}(\mathbb{R}^2). \end{cases} \tag{2.21}$$

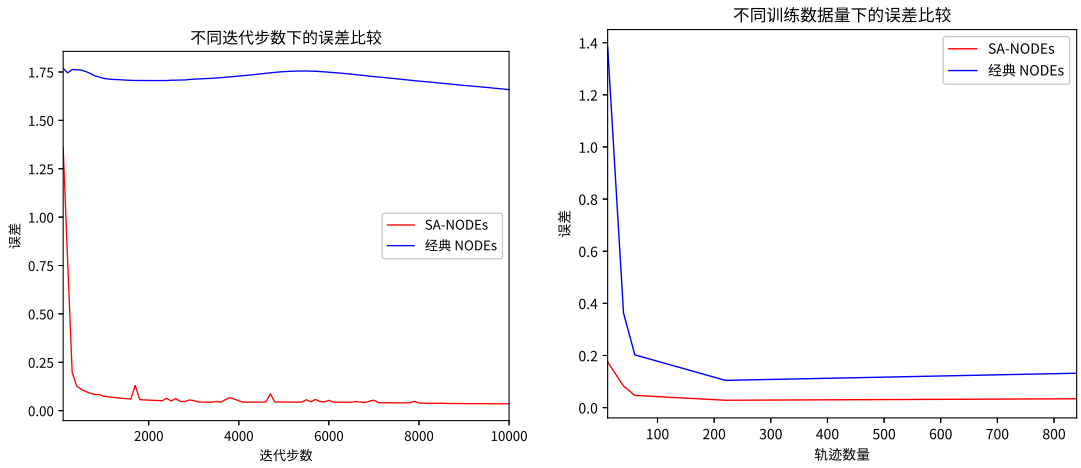


图 2.4: 系统 (2.20) 上经典 NODEs 与 SA-NODEs 的测试误差对比: (左) 训练集固定为 220 条轨迹, 训练轮次从 10 到 10^4 变化; (右) 训练轮次固定为 10^4, 训练集规模从 12 到 840 条轨迹变化.

根据注记 2.11 可知, 我们只需逼近 (2.21) 的如下特征线系统即可:

$$\begin{cases} \dfrac{dx}{dt} = \dfrac{\sin(x)}{1+t^2}, \\ \dfrac{dy}{dt} = \dfrac{\sin(y)}{1+t^2}, \\ \dfrac{d\rho}{dt} = -\rho \cdot \dfrac{\cos(x)+\cos(y)}{1+t^2}, \end{cases}$$

其中 $t \in [0,T]$. 在训练 SA-NODE 时, 我们从一个简单的初始分布中提取样本: 即在集合 $K=[-4,4]^2$ 上的均匀测度,

$$\rho_0^{\text{train}}(x,y) = 0.5, \quad (x,y) \in [-4,4]^2. \tag{2.22}$$

而在测试阶段, 为了客观评估模型的性能, 我们采用了不同的初始分布: 即在 K 上截断的高斯型测度,

$$\rho_0^{\text{test}}(x,y) = e^{-\frac{x^2+y^2}{4}}, \quad (x,y) \in [-4,4]^2. \tag{2.23}$$

记 ρ_Θ 与 ρ 分别为神经传输方程与真实传输方程的解, 二者均以相同的初始测度 (2.23) 进行初始化. 为了量化 ρ_Θ 的逼近效果, 我们对每个时间步 $t \in [0,5]$ 定义如下归一化测试误差:

$$e_{\text{test}}(t) = \|\bar{\rho}_\Theta(\cdot,t) - \bar{\rho}(\cdot,t)\|_{\mathbb{L}^1},$$

其中 $\bar{\rho}_\Theta(\cdot,t)=\frac{\rho_\Theta(\cdot,t)}{\|\rho(\cdot,0)\|_{\mathbb{L}^1}}$ 且 $\bar{\rho}(\cdot,t)=\frac{\rho(\cdot,t)}{\|\rho(\cdot,0)\|_{\mathbb{L}^1}}$. 由于初始测度具备正性与形式上的一致性, 结合传输方程的质量守恒性质, ρ_Θ 与 ρ 共享了相同的归一化因子. 在此处, 我们采用了 $\mathbb{L}^1$ 范数而非 Wasserstein-1 距离 $\mathbb{W}_1$(后者在定理 2.3 中被使用) 来度量误差, 原因有二:

1. 初始分布是绝对连续且具有紧支撑的, 因此解在任意时刻均保持在 $\mathbb{L}^1(\mathbb{R}^2)$ 空间内. 计算 $\mathbb{L}^1$ 距离远比计算 $\mathbb{W}_1$ 距离简单; 后者通常需要求解复杂的数值最优传输问题.

2. $\mathbb{W}_1$ 误差可以通过 $\mathbb{L}^1$ 误差给出上界:

$$\mathbb{W}_1\big(\bar{\rho}_\Theta(\cdot,t),\,\bar{\rho}(\cdot,t)\big)\;\le\;\frac{\operatorname{diam}(\Omega_t)}{2}\,\big\|\bar{\rho}_\Theta(\cdot,t)-\bar{\rho}(\cdot,t)\big\|_{\mathbb{L}^1},$$

   其中 $\operatorname{diam}(\Omega_t)$ 表示 $\bar{\rho}_\Theta(\cdot,t)$ 与 $\bar{\rho}(\cdot,t)$ 的共同支撑集的直径; 根据 Grönwall 引理可知, 该直径在有限时间内保持有界.

训练数据集的初始位置取自区域 $[-4,4]^2$ 上的网格, 采样步长为 0.2(共计 1681 条轨迹). 测试数据集则采用了更为致密的网格 $[-4,4]^2$, 步长为 0.1, 从而获得了 6561 个初始值及其对应的轨迹, 用以评估模型在整个状态空间上的泛化能力.

图 2.5 从上到下依次展示了: SA-NODE 的数值解, 经典 NODE 的数值解以及传输方程的精确解. 展示的区域为 $(x,y)\in[-4,4]^2$, 时间跨度上选取了 $t\in[0,5]$ 的 51 个等距时刻 (包含端点 0 与 T). 图 2.6 给出了相应的逼近误差: 左图展示了 SA-NODE 的训练与测试误差, 右图则对比了 SA-NODE 与经典 NODE 在测试集上的误差表现.

从图 2.5 中可以观察到, 两种神经网络模型均能较好地逼近真实的动力学过程. 在图 2.6 中, 测试误差始终保持在较低水平, 其量级未超过 10^{-1}. 右图清晰地显示: 尽管这两种模型在演化后期都会收敛到相近的精度, 但经典 NODE 的误差出现了明显的震荡; 这一现象充分突显了 SA-NODE 在稳定性方面的优势. 此外, 由于理论误差 (参见定理 2.2 与 2.3) 是以 “整个时间区间上的最大值” 来定义的, 因此在这一鲁棒性的意义下, SA-NODE 展现出了更为优异的逼近表现.

例 4: Doswell 锋生模型

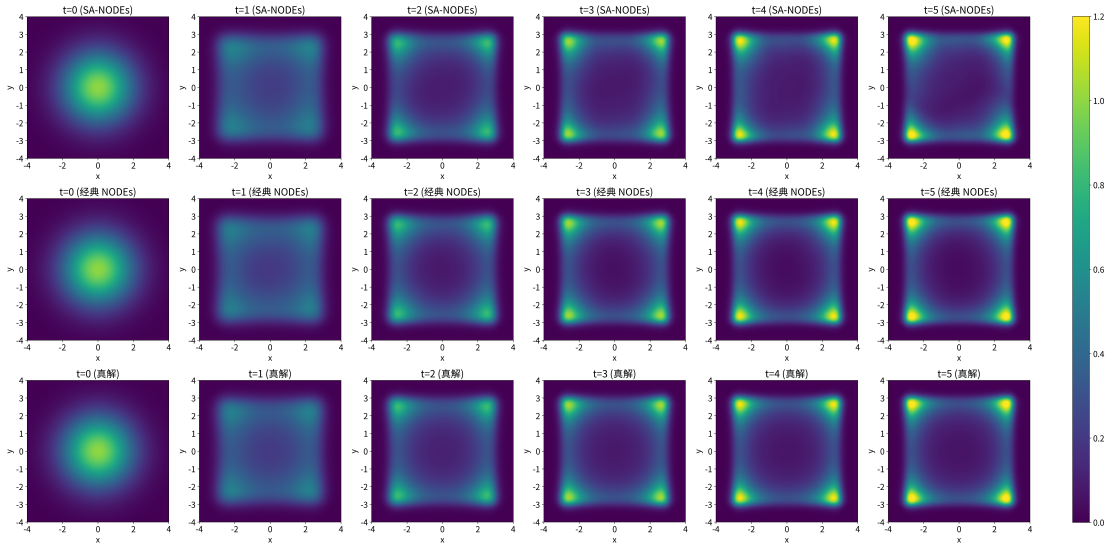


图 2.5: 在初值测度为 (2.23) 的条件下, 传输方程 (2.21) 的 SA-NODE 解, 经典 NODE 解与精确解对比.

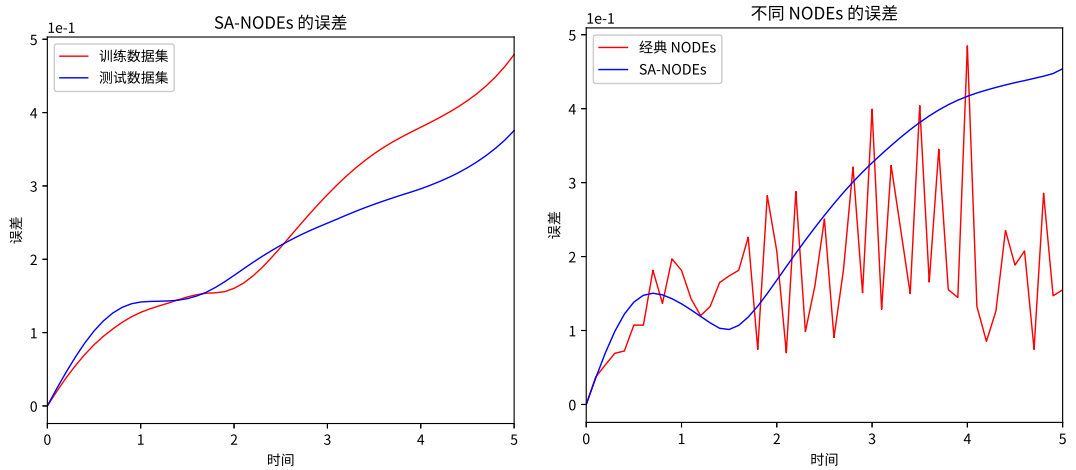


图 2.6: 传输方程 (2.21): SA-NODE 的训练与测试误差, 以及与经典 NODE 的测试误差对比.

下面我们考虑二维 Doswell 锋生方程 [97-98]. 该模型在气象动力学中被广泛用于描述水平温度梯度与锋面的产生和演化过程. 其控制方程为:

$$\begin{cases} \partial_t \rho(x,y,t) + \operatorname{div}\left((-yg(r(x,y)), xg(r(x,y)))\,\rho(x,y,t)\right) = 0, \quad (x,y,t) \in \mathbb{R}^2 \times [0,T], \\ \rho(\cdot,0) = \rho_0, \end{cases} \tag{2.24}$$

其中

$$g(r(x,y)) = \frac{1}{r(x,y)}\,\overline{v}\,\operatorname{sech}^2(r(x,y))\tanh(r(x,y)), \tag{2.25}$$

并且 $r(x,y) = \sqrt{x^2+y^2}$, 测度常数 $\overline{v} = 2.59807$. 训练与测试阶段的初值分别取为:

$$\rho_0^{\text{train}}(x,y) = \tanh(y), \quad \rho_0^{\text{test}}(x,y) = \tanh(10\,y).$$

在这种特定的初始值选择下, 方程 (2.24) 的精确解可以直接通过解析计算得出:

$$\nu(x,y,t) = \tanh\left(\frac{y\cos(g(r)t) - x\sin(g(r)t)}{\delta}\right),$$

其中, 当初始值为 ρ_0^{train} 时取 $\delta = 1$, 而当初始值为 ρ_0^{test} 时取 $\delta = 1/10$.

为了生成所需的训练集与测试集, 我们选取了空间区域 $K = [-5,5]^2$, 并在规则网格上对初始条件进行采样: 训练集的网格步长为 0.2 (共提取 2601 条轨迹), 测试集的网格步长为 0.1 (共提取 10201 条轨迹). 时间离散化方案与上一实验保持一致.

图 2.7 给出了在测试初始值条件下, SA-NODE 数值解, 经典 NODE 数值解与精确解的对比, 可以看到它们在整个时间区间 $[0,4]$ 上几乎完全重合. 相应的误差曲线 (定义同例 3) 展示在图 2.8 中, 模型误差始终维持在 10^{-2} 的极低水平. 同样地, 右侧的对比图深刻揭示了经典 NODE 的不稳定性: 其误差出现了明显的震荡. 这一结果进一步证实了 SA-NODE 的逼近在鲁棒性与稳定性方面具有显著的优势.

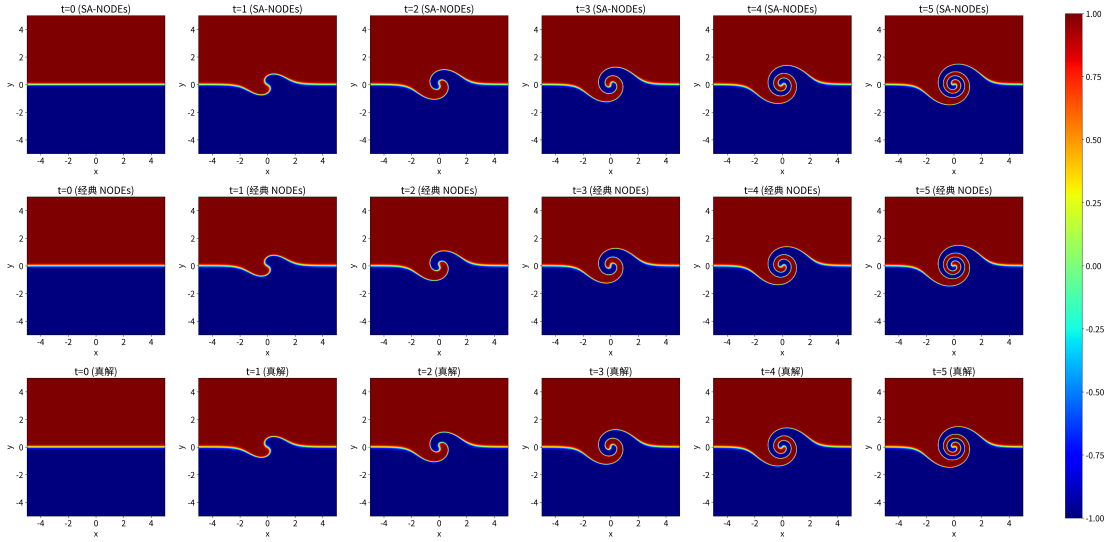


图 2.7: 在测试初始值下, 传输方程 (2.24) 的 SA-NODE 解, 经典 NODE 解与精确解的对比.

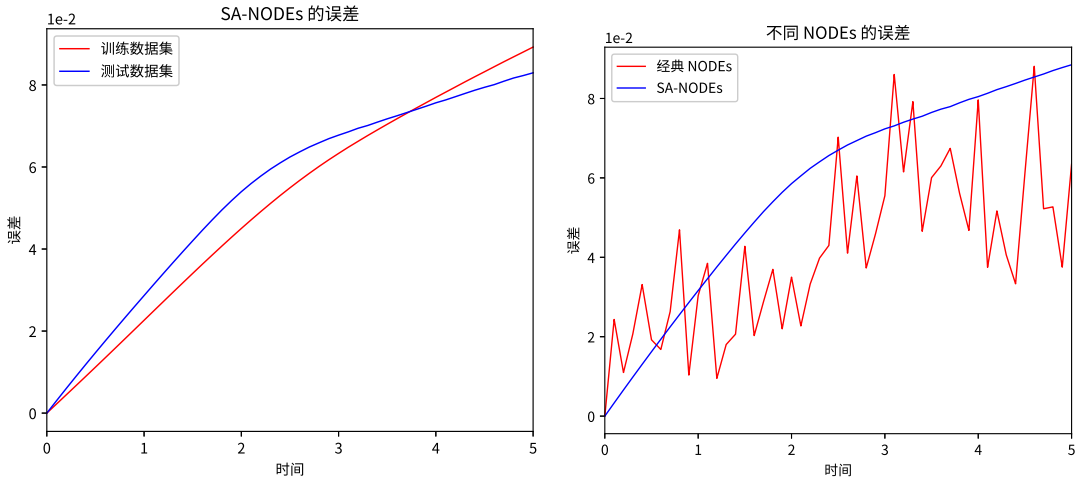


图 2.8: 传输方程 (2.24): SA-NODE 的训练与测试误差, 以及与经典 NODE 的测试误差对比.

第 3 章 基于神经常微分方程的算子学习

本章围绕基于神经常微分方程的算子学习展开理论建构与方法研究. 具体而言, 第 3.1 节首先在一般编码器-解码器网络的抽象框架下, 讨论其基本架构, 说明性示例及统一误差估计, 从而为后续模型分析提供理论准备. 在此基础上, 第 3.2 节进一步提出 NODE-ONet 框架, 并从稳态情形, 含时演化情形, 训练机制以及与 DeepONets 的比较等方面, 系统说明其结构设计与方法特征. 随后, 第 3.3 节结合非线性反应-扩散方程与 Navier-Stokes 方程, 构造融入物理先验信息的 NODE 模型, 以增强网络对偏微分方程演化规律的表达能力. 最后, 第 3.4 节通过一系列数值实验, 对所提出方法的逼近精度, 计算效率, 外推能力及其相对于基准模型的性能表现进行系统验证.

3.1 一般的编码器-解码器网络及其误差分析

本节首先回顾算子学习领域中被广泛采用的编码器-解码器 (Encoder-Decoder) 网络架构 [54, 99, 43]. 在此基础上, 我们针对此类网络构建了一套通用的误差分析框架. 该分析框架为后续章节中专门用于求解偏微分方程的深度 NODE-ONets 框架的架构设计与理论证明奠定了坚实的理论基础.

3.1.1 编码器-解码器网络的架构

编码器-解码器网络旨在逼近定义于无限维输入空间 $\mathcal{V}$ 与输出空间 $\mathcal{U}$ 之间的目标算子 $\Psi^\dagger : \mathcal{V} \to \mathcal{U}$. 为此, 我们首先引入两个适当的潜在空间 (latent spaces), 分别记为 $\mathcal{V}_h$ 与 $\mathcal{U}_h$. 相较于原始的无限维空间 $\mathcal{V}$ 与 $\mathcal{U}$, 这两个潜在空间通常具备更为简单的数学结构, 且往往为有限维空间. 随后, 我们选取编码器 $E_{\mathcal{V}}$ 与解码器 $D_{\mathcal{U}}$:

$$E_{\mathcal{V}} : \mathcal{V} \to \mathcal{V}_h \quad \text{以及} \quad D_{\mathcal{U}} : \mathcal{U}_h \to \mathcal{U},$$

这二者既可以是预先给定的, 也可以是由神经网络进行参数化的. 接下来, 我们进一步设计解码器 $D_{\mathcal{V}}$ 与编码器 $E_{\mathcal{U}}$:

$$D_{\mathcal{V}} : \mathcal{V}_h \to \mathcal{V} \quad \text{以及} \quad E_{\mathcal{U}} : \mathcal{U} \to \mathcal{U}_h,$$

使得复合映射

$$D_{\mathcal{U}} \circ E_{\mathcal{U}} \quad \text{与} \quad D_{\mathcal{V}} \circ E_{\mathcal{V}}$$

能够分别逼近空间 $\mathcal{U}$ 与 $\mathcal{V}$ 上的恒等映射 (针对线性情形的详细说明, 参见假设 3.1 (1)-(2)). 上述编码器与解码器对自然地导出了底层无限维算子 $\Psi^\dagger$ 在潜在空间中的一种编码表示, 从而在潜在空间 $\mathcal{V}_h$ 与 $\mathcal{U}_h$ 之间诱导出一个映射函数:

$$\psi : \mathcal{V}_h \to \mathcal{U}_h, \quad \psi(\zeta) = E_{\mathcal{U}} \circ \Psi^\dagger \circ D_{\mathcal{V}}(\zeta), \quad \forall \zeta \in \mathcal{V}_h,$$

该结构的示意图参见图 3.1 的右侧. 于是, 在形式上我们可得:

$$D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}} \approx \Psi^\dagger.$$

利用带有可训练参数 θ 的神经网络 $\psi_\theta \colon \mathcal{V}_h \to \mathcal{U}_h$ 来逼近 ψ, 我们便构造出了如下形式的编码器-解码器网络 $\Psi_\theta \colon \mathcal{V} \to \mathcal{U}$:

$$\Psi_\theta := D_{\mathcal{U}} \circ \psi_\theta \circ E_{\mathcal{V}}, \tag{3.1}$$

该网络最终被用于逼近目标算子 $\Psi^\dagger$, 如图 3.1 的左侧所示. 目前文献中具有代表性的编码器-解码器网络架构包括 PCA-Net [54], 积分自编码器网络 [100], DeepONets [43], MIONet [56] 及其各类扩展变体 [101-103, 57, 104] 等.

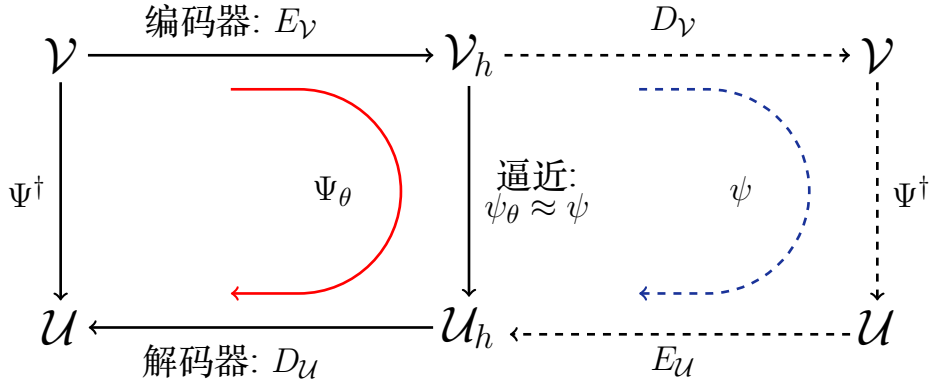


图 3.1: 无限维空间 $\mathcal{V}$ 和 $\mathcal{U}$ 之间映射的潜在结构.

3.1.2 一个说明性的示例

在此, 我们通过一个具体的示例来演示如何将 NODEs 完美融入至编码器-解码器架构中, 从而学习偏微分方程的解算子. 关于该框架的更一般性表

述将在第 3.2 节中详述. 考虑一维空间 ($d = 1$) 下的偏微分方程 (1.7). 对于给定的固定系数场 $a(t,x)$ 与非线性反应项 R, 其空间微分算子具有如下形式:

$$\mathcal{L}[a](u)(t,x) = -\nabla \cdot \big(a(t,x)\nabla u(t,x)\big) + R\big(u(t,x)\big).$$

假设边界条件 u_b 保持固定, 而初始条件作为该偏微分方程的待定输入参数, 记作 $v = u_0 \in \mathcal{C}(\Omega)$. 本示例的核心目标在于学习从初始条件到解的映射算子:

$$\Psi^{\dagger}_{\text{initial}} : u_0 \mapsto u.$$

首先, 我们将潜在空间分别设定为有限维欧几里得空间 $\mathcal{V}_h = \mathbb{R}^{N_x}$ 以及函数空间 $\mathcal{U}_h = \mathcal{C}([0,T];\mathbb{R}^{N_x})$, 其中网格节点数 $N_x \in \mathbb{N}_+$. 编码与解码步骤分别借助均匀有限差分网格上的逐点求值与 P_1 有限元基函数的空间插值来实现. 具体而言, 设 $\{x_i\}_{i=1}^{N_x}$ 为空间区域 Ω 上的均匀剖分网格, α_i 为以节点 x_i 为中心的 P_1 有限元基函数. 由此构建的编码器-解码器架构具体设定如下:

$$\textbf{网络架构}\begin{cases} \text{编码器 } E_{\mathcal{V}}\colon\ u_0 \ \mapsto\ U_{0,h} = \big(u_0(x_i)\big)_{i=1}^{N_x} \in \mathbb{R}^{N_x}, \\ \text{NODE 代理模型 } \psi_\theta\colon U_{0,h} \mapsto U_\theta,\ \text{其中} \begin{cases} \dot{U}_\theta(t) = \mathcal{NN}_\theta(U_\theta(t), t), \\ U_\theta(0) = U_{0,h}, \end{cases} \\ \text{解码器 } D_{\mathcal{U}}\colon\ u(t,x)\ =\ \displaystyle\sum_{i=1}^{N_x} \big(U_\theta(t)\big)_i\, \alpha_i(x). \end{cases}$$

式中, $\mathcal{NN}_\theta$ 表示一个由 θ 进行参数化且其参数不显式依赖于时间 t 的神经网络函数. 这一关键特性构成了本方法与公式 (1.2) 中经典 NODE 标准形式的根本区别. 为了将偏微分方程特定的物理演化信息物理地编码至网络中, 我们对 $\mathcal{NN}_\theta$ 的内部结构进行了精心设计, 其具体的表达形式将在后文的方程 (3.17) 与 (3.20) 中予以详述.

在模型训练阶段, 我们利用经典的有限差分法 (FDM) 生成高保真度的数据集, 并据此对相关的损失函数进行优化求解. 设 N_t 为有限差分法中的时间离散步数, t_j ($j = 1,\dots,N_t$) 为相应的时间离散节点. 类似地, 设 N_v 为输入函数样本的总个数, 而 u_0^k ($k = 1,\dots,N_v$) 则表示第 k 个样本对应的初始条件

输入. 整个训练阶段的具体设定可归纳如下:

$$\textbf{训练方案:}\begin{cases}\text{数据集:}\begin{cases}\text{特征 (Features): } U_{0,h}^{k}\in\mathbb{R}^{N_x},\ \text{通过对初始条件} u_0^k \text{ 进行编码获得;}\\ \text{标签 (Labels): } U_{h}^{k}\in\mathbb{R}^{N_x\times N_t},\ \text{通过对 (1.7) 有限差分求解获得;}\end{cases}\\ \text{损失函数: } L(\theta)=\dfrac{1}{N_vN_xN_t}\displaystyle\sum_{k=1}^{N_v}\sum_{i=1}^{N_x}\sum_{j=1}^{N_t}\left|\left(U_\theta^k(t_j)\right)_i-\left(U_h^k(t_j)\right)_i\right|^2+\mathcal{R}(\theta).\end{cases}$$

此处, U_θ^k 表示以 $U_{0,h}^k$ 为初始条件的神经网络代理模型所预测的解轨迹, 而 $\mathcal{R}(\theta)$ 则是用于防止过拟合的常规正则化惩罚项. 该训练过程完全在离线 (Offline) 状态下进行. 一旦优化算法成功收敛并求得最优的网络参数 θ^*, 针对任意全新给定初始条件 u_0 的偏微分方程 (1.7), 我们只需直接复用上述固化的网络架构进行一次前向推理 (Online inference), 即可极其高效地获取其对应的数值解.

3.1.3 编码器-解码器网络的误差分析

本小节旨在对第 3.1.1 节中提出的一般编码器-解码器网络展开严格的误差估计, 即对逼近误差 $\Psi_\theta-\Psi^\dagger$ 进行有界性分析. 为此, 我们首先针对相关的函数空间, 编码器与解码器架构以及目标映射 $\Psi^\dagger$ 作出如下理论假设.

假设 **3.1.** 设 $\mathcal{V}, \mathcal{U}, \mathcal{V}_h$ 以及 $\mathcal{U}_h$ 均为 *Banach* 空间. 我们假设*:*

1. *(*线性*, **Linearity**)* 编码器 $E_{\mathcal{V}}\colon \mathcal{V}\to\mathcal{V}_h$ 与解码器 $D_{\mathcal{U}}\colon \mathcal{U}_h\to\mathcal{U}$ 均为有界线性算子, 且广义逆始终存在唯一 *(*参见文献 *[105]* 中的定义 *1.38).*

2. *(*广义逆*, **Generalized inversion**)* 解码器 $D_{\mathcal{V}}\colon \mathcal{V}_h\to\mathcal{V}$ 与编码器 $E_{\mathcal{U}}\colon \mathcal{U}\to\mathcal{U}_h$ 分别为 $E_{\mathcal{V}}$ 与 $D_{\mathcal{U}}$ 的广义逆算子, 其数学意义在于满足如下恒等关系 *(*参见文献 *[105]* 中的定义 *1.38* 与等式 *(1.7)):*

$$D_{\mathcal{V}}\circ E_{\mathcal{V}}\circ D_{\mathcal{V}}=D_{\mathcal{V}},\qquad D_{\mathcal{U}}\circ E_{\mathcal{U}}\circ D_{\mathcal{U}}=D_{\mathcal{U}}.$$

3. *(*连续性*, **Continuity**)* 目标算子 $\Psi^\dagger\colon \mathcal{V}\to\mathcal{U}$ 满足 β*-Hölder* 连续性, 即存在某个指数 $\beta\in(0,1]$, 使得其 *Hölder* 常数为 $L_{\Psi^\dagger}$.

4. *(*万能逼近性质*, **Universal approximation property**)* 定义算子 $\psi\colon \mathcal{V}_h\to\mathcal{U}_h$ 为*:*

$$\psi(\zeta)=E_{\mathcal{U}}\circ\Psi^\dagger\circ D_{\mathcal{V}}(\zeta).$$

假设对于潜在空间 $\mathcal{V}_h$ 中的任意紧子集 $\mathcal{K} \subset \mathcal{V}_h$ 以及任意给定的逼近容限 $\epsilon > 0$, 均存在一个具备适当网络架构与参数 θ 的神经网络泛函 $\psi_\theta \colon \mathcal{V}_h \to \mathcal{U}_h$, 使得如下一致逼近条件成立:

$$\|\psi_\theta(v) - \psi(v)\|_{\mathcal{U}_h} \leq \epsilon, \qquad \forall v \in \mathcal{K}.$$

进而, 我们定义该编码-解码格式中两个关键的一致性误差 (Consistency errors):

$$d_1(v) \coloneqq \|D_{\mathcal{V}} \circ E_{\mathcal{V}}(v) - v\|_{\mathcal{V}}, \qquad \forall v \in \mathcal{V}, \tag{3.2}$$

$$d_2(u) \coloneqq \|D_{\mathcal{U}} \circ E_{\mathcal{U}}(u) - u\|_{\mathcal{U}}, \qquad \forall u \in \mathcal{U}. \tag{3.3}$$

在陈述主要的误差估计结论之前, 我们首先给出一个满足上述所有假设的说明性示例 (详见下述注记). 应当指出, 在对输入参数 v 与物理真实解 u 施加额外的正则性条件后, 上述一致性误差将随着潜在空间离散度的细化而自然趋于零.

注 3.1. 考虑定义在 d 维平坦环面 $\mathbb{T}^d$ 上的稳态反应-扩散方程:

$$-\Delta u + c(x)\, u = f(x), \quad x \in \mathbb{T}^d,$$

假设系数 $c \in \mathcal{C}^{0,\alpha}(\mathbb{T}^d)$ 是严格正的, 且源项 $f \in \mathcal{C}^{0,\alpha}(\mathbb{T}^d)$, 其中 $\alpha \in (0, 1]$ [1]. 依据经典的 *Schauder* 先验估计 *[106, 定理 4.3.2]*, 并结合椭圆型方程的 *Sobolev* 估计 *[107, 第 5.1 章, 定理 1(ii)]* 以及 *Morrey* 不等式 *[82, 第二部分, 定理 4.12]*, 该方程存在唯一经典解 u 且满足如下有界性:

$$u \in \mathcal{C}^{2,\alpha}(\mathbb{T}^d), \qquad \|u\|_{\mathcal{C}^{2,\alpha}} \leq C_1 \|f\|_{\mathcal{C}^{0,\alpha}}, \qquad \|u\|_{\mathcal{C}^1(\mathbb{T}^d)} \leq C_2 \|f\|_{\mathcal{C}(\mathbb{T}^d)},$$

[1] 对于任意的整数 $k \in \mathbb{Z}_+$ 与指数 $\alpha \in (0, 1]$, Hölder 函数空间 $\mathcal{C}^{k,\alpha}(\mathbb{T}^d)$ 严格定义为

$$\mathcal{C}^{k,\alpha}(\mathbb{T}^d) \coloneqq \left\{ u \in \mathcal{C}^k(\mathbb{T}^d) \,:\, \|u\|_{\mathcal{C}^{k,\alpha}} < \infty \right\},$$

其对应的 Hölder 范数由下式给出:

$$\|u\|_{\mathcal{C}^{k,\alpha}} \coloneqq \sum_{\|\beta\|_{\ell^1} \leq k} \|\partial^\beta u\|_{L^\infty(\mathbb{T}^d)} + \sum_{\|\beta\|_{\ell^1} = k} \sup_{\substack{x, y \in \mathbb{T}^d \\ x \neq y}} \frac{\left|\partial^\beta u(x) - \partial^\beta u(y)\right|}{\|x - y\|^\alpha}.$$

此处, 记号 $\partial^\beta u$ 表示函数 u 关于多重指标 $\beta \in \mathbb{Z}_+^d$ 的偏导数.

其中常数 C_1 与 C_2 完全独立于源项 f. 因此, 解算子映射 $\Psi^\dagger\colon f \mapsto u$ 从 $\mathcal{C}^{0,\alpha}(\mathbb{T}^d)$ 到 $\mathcal{C}^{2,\alpha}(\mathbb{T}^d)$, 以及从 $\mathcal{C}(\mathbb{T}^d)$ 到 $\mathcal{C}^1(\mathbb{T}^d)$ 的映射均是 *Lipschitz* 连续的 *(*因而自然满足 *Hölder* 连续性条件*)*. 接着, 我们考察如下构建的编码器-解码器网络设定*:*

- 令 $\mathcal{U} = \mathcal{V} = \mathcal{C}(\mathbb{T}^d)$, 并在极其精细的网格尺寸 $h = 1/N \ll 1$ 下设定潜在空间 $\mathcal{U}_h = \mathcal{V}_h = \mathbb{R}^{N^d}$*;*
- 利用有限差分模板以及在步长为 h 的均匀网格上构造的 Q_1 有限元基函数插值, 来分别定义编码器 $E_{\mathcal{V}}$ 与解码器 $D_{\mathcal{U}}$*;*
- 令 $D_{\mathcal{V}} = D_{\mathcal{U}}$ 以及 $E_{\mathcal{U}} = E_{\mathcal{V}}$, 我们可以严格验证假设 *3.1(2)* 中的广义逆条件得到充分满足 *(*严格的数学证明参见引理 *3.1* 与注记 *3.2)*.

在此构造体系下, 由解算子 $\Psi^\dagger$ 的 *Lipschitz* 连续性以及编码器与解码器算子的有界连续性可直接推导出, 复合算子

$$\psi(v) = E_{\mathcal{U}} \circ \Psi^\dagger \circ D_{\mathcal{V}}(v)$$

亦是连续的. 依据经典的神经网络万能逼近定理 *(*例如 *[5])*, 该连续性充分保证了非线性算子 ψ 可以在 $\mathcal{V}_h$ 的任意紧子集上被浅层神经网络一致逼近 [2], 从而严谨地验证了假设 *3.1(4)*. 最终, 对应于该编码器与解码器架构的一致性误差满足如下渐近估计*:*

$$\sup_{\substack{v\in\mathcal{C}^{0,\alpha}(\mathbb{T}^d)\\ \|v\|_{\mathcal{C}^{0,\alpha}}\le 1}} d_1(v) \le Ch^\alpha, \qquad \sup_{\substack{u\in\mathcal{C}^{1}(\mathbb{T}^d)\\ \|u\|_{\mathcal{C}^{1}}\le 1}} d_2(u) \le Ch,$$

其中 $C > 0$ 为独立于网格尺寸 h 的常数. 特别地, 此处的一致性误差以 h^α 的收敛速率趋于零, 从而在网格细化 *(*$h \to 0$*)* 的极限情形下变得任意小.

针对一般编码器-解码器网络架构的严格误差估计结果在如下核心定理中予以阐明.

[2]从输入空间 $\mathbb{R}^n$ 映射到输出空间 $\mathbb{R}^m$ 的浅层神经网络具有如下代数形式:

$$f_{\text{shallow}}(x;\,\theta) := W_2\,\sigma(W_1 x + b),$$

其中 σ 表示逐分量作用的非线性激活函数, 网络参数集 $\theta = (W_1, W_2, b)$ 满足权重矩阵 $W_1 \in \mathbb{R}^{P\times n}$, $W_2 \in \mathbb{R}^{m\times P}$, 以及偏置向量 $b \in \mathbb{R}^P$. 此处, 整数 $P \in \mathbb{N}$ 表征隐藏层中的神经元宽度. 该神经网络架构的万能逼近性质是在神经元宽度 P 趋于无穷大的渐近意义下理解的.

定理 3.1. 假设条件 *3.1* 成立. 那么, 对于潜在空间中的任意紧子集 $\mathcal{K} \subset \mathcal{V}_h$ 以及任意给定的逼近容限 $\epsilon > 0$, 必然存在一个具备适当网络架构与参数 θ 的神经网络泛函 $\psi_\theta \colon \mathcal{V}_h \to \mathcal{U}_h$, 使得对于任意满足编码条件 $E_{\mathcal{V}}(v) \in \mathcal{K}$ 的输入 $v \in \mathcal{V}$, 如下总体误差界恒成立:

$$\begin{aligned} \|\Psi_\theta(v) - \Psi^\dagger(v)\|_{\mathcal{U}} \leq & \underbrace{L_{\Psi^\dagger}\, d_1(v)^\beta}_{\mathcal{V}\text{ 空间中的编码-解码误差}} + \underbrace{d_2\big(\Psi^\dagger \circ D_{\mathcal{V}} \circ E_{\mathcal{V}}(v)\big)}_{\mathcal{U}\text{ 空间中的编码-解码误差}} \\ & + \underbrace{\|D_{\mathcal{U}}\|\,\epsilon}_{\text{神经网络的非线性逼近误差}}, \end{aligned} \tag{3.4}$$

其中代理模型 Ψ_θ 由等式 (3.1) 所定义, 且一致性误差函数 d_1 与 d_2 由 (3.2)–(3.3) 给出.

证明: 给定任意紧子集 $\mathcal{K} \subset \mathcal{V}_h$ 与误差容限 $\epsilon > 0$. 选取任意函数 $v \in \mathcal{V}$ 使得其空间编码满足 $E_{\mathcal{V}}(v) \in \mathcal{K}$. 我们首先将总逼近误差

$$\Psi_\theta(v) - \Psi^\dagger(v) = D_{\mathcal{U}} \circ \psi_\theta \circ E_{\mathcal{V}}(v) - \Psi^\dagger(v)$$

进行如下形式的代数分解:

$$\Psi_\theta(v) - \Psi^\dagger(v) := \gamma_1(v) + \gamma_2(v), \tag{3.5}$$

其中误差项定义为:

$$\gamma_1(v) := D_{\mathcal{U}} \circ \psi_\theta \circ E_{\mathcal{V}}(v) - D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}}(v), \quad \gamma_2(v) := D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}}(v) - \Psi^\dagger(v).$$

依据假设 3.1(4) 中的万能逼近性质, 神经网络自身诱导的逼近误差可被严格控制为:

$$\|\gamma_1(v)\| \leq \|D_{\mathcal{U}}\|\,\epsilon. \tag{3.6}$$

随后, 引入如下辅助变量:

$$v_1 \coloneqq D_{\mathcal{V}} \circ E_{\mathcal{V}}(v) \quad \text{与} \quad u_1 \coloneqq D_{\mathcal{U}} \circ E_{\mathcal{U}} \circ \Psi^\dagger(v_1).$$

鉴于 v_1 与 u_1 分别严格隶属于解码器算子 $D_{\mathcal{V}}$ 与 $D_{\mathcal{U}}$ 的值域, 假设 3.1(2) 中的广义逆性质直接蕴含了如下恒等关系:

$$D_{\mathcal{V}} \circ E_{\mathcal{V}}(v_1) = v_1 \quad \text{与} \quad D_{\mathcal{U}} \circ E_{\mathcal{U}}(u_1) = u_1.$$

我们进而将误差分量 $\gamma_2(v)$ 进一步展开并拆解为两项:

$$\gamma_2(v) = D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}}(v) - \Psi^\dagger(v) = \gamma_{2,1}(v) + \gamma_{2,2}(v), \tag{3.7}$$

其中

$$\gamma_{2,1}(v) := D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}}(v) - D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}}(v_1) - \Big(\Psi^\dagger(v) - \Psi^\dagger(v_1)\Big),$$
$$\gamma_{2,2}(v) := D_{\mathcal{U}} \circ \psi \circ E_{\mathcal{V}}(v_1) - u_1 - \Big(\Psi^\dagger(v_1) - u_1\Big).$$

结合映射 ψ 的算子定义以及核心恒等式 $D_{\mathcal{V}} \circ E_{\mathcal{V}}(v_1) = v_1$, 我们通过三角不等式可得:

$$\|\gamma_{2,1}(v)\|_{\mathcal{U}} = \Big\|\Psi^\dagger(v) - \Psi^\dagger(v_1)\Big\|_{\mathcal{U}} \le L_{\Psi^\dagger}\, d_1(v)^\beta, \tag{3.8}$$

其中的不等号放缩得益于假设 3.1(3) 中目标算子的 Hölder 连续性. 同理, 根据 ψ 的构造机制与辅助变量 u_1 的代数定义, 易知:

$$\|\gamma_{2,2}(v)\|_{\mathcal{U}} = \Big\|\Psi^\dagger(v_1) - D_{\mathcal{U}} \circ E_{\mathcal{U}}\big(\Psi^\dagger(v_1)\big)\Big\|_{\mathcal{U}}.$$

回顾等式 (3.3) 中关于一致性误差 d_2 的统一定义, 我们进一步推导出:

$$\|\gamma_{2,2}(v)\|_{\mathcal{U}} = d_2\Big(\Psi^\dagger(v_1)\Big) = d_2\Big(\Psi^\dagger \circ D_{\mathcal{V}} \circ E_{\mathcal{V}}(v)\Big). \tag{3.9}$$

综合上述关系式 (3.5) 至 (3.9), 即可直接导出并完成定理 3.1 的理论证明. □

推论 3.1. *在注记 3.1 所述的偏微分方程数学设定下, 假定解的正则性指数 $\alpha \in (0,1)$ 保持固定, 并给定有限差分的离散化网格步长 $h = 1/N$ (其中 $N \in \mathbb{N}_+$). 那么, 必然存在一个网络宽度足够大且参数 θ 经过适当优化的浅层神经网络模型 $\psi_\theta \colon \mathbb{R}^{N^d} \to \mathbb{R}^{N^d}$, 使得对于任意满足如下正则性条件的源项输入 f:*

$$f \in C^{0,\alpha}(\mathbb{T}^d), \qquad \|f\|_{C^{0,\alpha}(\mathbb{T}^d)} \le 1,$$

以下关于解算子逼近的误差上界恒成立:

$$\big\|\Psi_\theta(f) - \Psi^\dagger(f)\big\|_{\mathcal{C}(\mathbb{T}^d)} \le C\, h^\alpha,$$

其中尺度常数 $C > 0$ 完全独立于网格步长 h 与源项输入 f.

证明: 在如下的放缩证明过程中, 记号 $\lesssim$ 表示左侧项小于或等于右侧项乘以一个既不依赖于 h 也不依赖于 f 的普适正常数. 首先回顾定理 3.1 中的主要误差估计拆解式 (3.4).

针对等式右侧的第一项 $L_{\Psi^\dagger}\, d_1(v)^\beta$, 利用解算子 $\Psi^\dagger$ 的 Lipschitz 连续性 (即取连续性指数 $\beta = 1$), 并代入注记 3.1 中针对一致性误差 d_1 的控制边界, 可得:

$$L_{\Psi^\dagger}\, d_1(f)^\beta \;\lesssim\; h^\alpha.$$

针对误差控制的第二项 $d_2\big(\Psi^\dagger \circ D_{\mathcal{V}} \circ E_{\mathcal{V}}(f)\big)$, 回顾注记 3.1 中对编码器-解码器复合算子的显式定义, 可推导出如下范数控制:

$$\|D_{\mathcal{V}} \circ E_{\mathcal{V}}(f)\|_{\mathcal{C}(\mathbb{T}^d)} \;\lesssim\; \|f\|_{\mathcal{C}(\mathbb{T}^d)} \;\lesssim\; \|f\|_{\mathcal{C}^{0,\alpha}(\mathbb{T}^d)}.$$

借助 Sobolev 空间理论中的经典椭圆正则性估计 [107, 第 5.1 章, 定理 1 (ii)] 并结合 Morrey 嵌入不等式 [82, 第二部分, 定理 4.12], 我们能够获得更高阶的导数控制:

$$\|\Psi^\dagger \circ D_{\mathcal{V}} \circ E_{\mathcal{V}}(f)\|_{\mathcal{C}^1(\mathbb{T}^d)} \;\lesssim \|D_{\mathcal{V}} \circ E_{\mathcal{V}}(f)\|_{\mathcal{C}(\mathbb{T}^d)} \;\lesssim\; \|f\|_{\mathcal{C}^{0,\alpha}(\mathbb{T}^d)}.$$

由此, 将注记 3.1 中关于误差 d_2 的结论代入, 即可得到第二个一致性误差的严格上界:

$$d_2\big(\Psi^\dagger \circ D_{\mathcal{V}} \circ E_{\mathcal{V}}(f)\big) \;\lesssim\; h.$$

至于最后的神经网络逼近误差项 $\|D_{\mathcal{U}}\|\,\epsilon$, 首先观察到解码器算子 $D_{\mathcal{U}}$ 的算子范数严格为 1 且始终有界. 此外, 考虑到连续函数 f 本身的有界性以及编码器算子 $E_{\mathcal{V}}$ 的连续映射性质, 编码后的潜在向量 $E_{\mathcal{V}}(f)$ 必在欧几里得空间 $\mathbb{R}^{N^d}$ 的某个固定紧集内部取值. 受益于多层前馈网络的万能逼近定理 [5], 限制在该紧致区域上的复合非线性映射 $E_{\mathcal{U}} \circ \Psi^\dagger \circ D_{\mathcal{V}}$ 完全能够被浅层神经网络以任意指定的精度所逼近. 令此处的逼近容限参数 $\epsilon = h^\alpha$, 则可直接得出结论:

$$\|D_{\mathcal{U}}\|\,\epsilon \;\lesssim\; h^\alpha.$$

最终, 依据三角不等式将上述三项的放缩上界予以合并, 即可得出目标定理中所声明的渐近误差估计. □

针对稳态偏微分方程的算子学习, 注记 3.1 与推论 3.1 为编码器-解码器网络的理论构建与误差剖析提供了一套数学规范框架. 然而, 将此类静态的分析框架直接推广至非稳态 (演化型) 偏微分方程时, 其数学处理显得更为微妙与复杂. 其中的核心挑战之一在于时间变量的离散化策略: 传统的时空全离散格式通常需要极其谨慎地施加时间步长约束以保证数值稳定性 (例如严苛的 CFL 条件). 作为一种具备优良理论性质的替代思路, 研究者往往倾向于采用时间连续的空间半离散格式. 然而, 这种处理方式不可避免地将低维潜在空间转化为了无限维的函数空间, 进而要求我们在无限维测度设定的背景下, 寻求并建立更高阶的泛函万能逼近理论. 本文的第二章深入探讨了利用神经常微分方程逼近连续动力系统的数学理论保证, 这一成果为分析由常微分方程所驱动的函数空间映射提供了关键的理论分析视角. 依托上述理论进展, 我们在第 3.2 节中正式面向非稳态偏微分方程构建了新颖的 NODE-ONet 框架. 尽管对 NODE-ONet 框架进行精细的收敛性误差分析错综复杂地依赖于具体偏微分方程特有的算子结构, 且必然涉及极其繁冗的技术推导细节, 但其整体的误差拆解逻辑依然遵循定理 3.1 所确立的完备范式. 鉴于对网络逼近误差进行绝对严谨的偏微分方程特异性边界分析已超出本文目前的探讨范围, 该前沿课题将被留作未来工作的重点研究方向. 基于此, 本章后续的重心将完全聚焦于 NODE-ONet 算法架构的物理编码创新设计, 以及其在求解各类复杂偏微分方程时的大规模基准数值验证.

3.2 NODE-ONet 框架体系

本节在上一节所构建的一般编码器-解码器网络架构的框架下, 正式引入深度神经常微分方程算子网络 (NODE-ONet) 框架. 该计算框架旨在精准逼近从偏微分方程 (1.7) 的参数空间到其物理真实解空间 u 的非线性算子映射.

3.2.1 稳态情形的启发

在深入探讨一般演化偏微分方程 (1.7) 之前, 我们首先考察其所对应的稳态数学模型:

$$\begin{cases} \mathcal{L}[a](u)(x) = f(x), & \forall x \in \Omega, \\ \mathcal{B}u(x) = u_b(x), & \forall x \in \partial\Omega, \end{cases} \tag{3.10}$$

以期为后续 NODE-ONet 的整体架构设计提供直观且具有启发性的视角. 在此稳态模型中, 微分算子参数 a 与源项 f 仅依赖于空间变量. 为简化并统一理论分析过程, 假设有界开区域 $\Omega \subset \mathbb{R}^d$ 为紧集, 且边界条件 u_b 固定. 因此, 该设定下的核心输入参数自然退化为 a 与 f. 令 $\mathcal{C}(\Omega)$ 表示定义在 Ω 上的连续函数空间. 假设输入参数 $a, f \in \mathcal{E} \subset \mathcal{C}(\Omega)$, 且对于任意给定的 $a, f \in \mathcal{E}$, 稳态方程 (3.10) 均适定且存在唯一的经典解. 相应地, 我们假定参数空间 $\mathcal{V}$ 与解空间 $\mathcal{U}$ 均为 $\mathcal{C}(\Omega)$ 的闭子空间. 如第 3.1.1 节所阐述, 构建编码器-解码器网络的关键步骤包括: 选取适当的潜在空间 $\mathcal{V}_h$ 与 $\mathcal{U}_h$, 设计用以逼近潜在映射的神经网络 ψ_θ, 以及严格定义编码器 $E_{\mathcal{V}}$ 与解码器 $D_{\mathcal{U}}$. 首先, 我们将两端的潜在空间分别设定为维度为 $d_{\mathcal{V}}$ 与 $d_{\mathcal{U}}$ 的有限维欧几里得空间:

$$\mathcal{V}_h = \mathbb{R}^{d_{\mathcal{V}}}, \quad \mathcal{U}_h = \mathbb{R}^{d_{\mathcal{U}}}.$$

令 ψ_θ 为满足假设 3.1(4) 逼近条件的神经网络. 于是, 针对该稳态情形的编码器与解码器可显式定义如下:

(S1). **稳态编码 (空间离散化):** 给定任意输入函数 $v \in \mathcal{C}(\Omega)$, 其稳态编码器定义为一映射:

$$E_{\mathcal{V}}^s : \mathcal{C}(\Omega) \to \mathbb{R}^{d_{\mathcal{V}}}, \quad v \mapsto \big(L_\ell(v)\big)_{\ell=1}^{d_{\mathcal{V}}},$$

其中 L_ℓ 代表 $\mathcal{C}(\Omega)$ 上的有界线性算子. 依据 Riesz 表示定理, 每一个有界线性泛函 L_ℓ 均必然具有如下的积分表示形式:

$$L_\ell(v) = \int_\Omega v \, d\mu_\ell, \quad \text{对应于某个测度} \mu_\ell \in \mathcal{M}(\Omega),$$

其中 $\mathcal{M}(\Omega)$ 表示支撑集包含于 Ω 内的 Radon 测度空间. 尤为值得注意的是, 若选取 Dirac 测度 $\mu_\ell = \delta_{x_\ell}$, 该积分算子便自然退化为在空间节点 x_ℓ 处对函数 v 的逐点求值 (Pointwise evaluation).

(S2). **稳态解码 (解重建):** 给定核心神经网络 ψ_θ 预测输出的潜在向量 $(\psi_j)_{j=1}^{d_{\mathcal{U}}} \in \mathbb{R}^{d_{\mathcal{U}}}$, 其稳态解码器 $D_{\mathcal{U}}^s$ 采用如下的基函数展开插值形式:

$$D_{\mathcal{U}}^s : \mathbb{R}^{d_{\mathcal{U}}} \to \mathcal{C}(\Omega), \quad (\psi_j)_{j=1}^{d_{\mathcal{U}}} \mapsto \sum_{j=1}^{d_{\mathcal{U}}} \alpha_j(x;\theta_\alpha)\, \psi_j, \quad \text{其中 } x \in \Omega.$$

在此处, 函数集合

$$\boldsymbol{\alpha}(x) := \{\alpha_j(x;\theta_\alpha)\}_{j=1}^{d_{\mathcal{U}}} \in \mathbb{R}^{d_{\mathcal{U}}} \tag{3.11}$$

既可以代表一组预先设定的全局空间基函数 (如有限元基或傅里叶基) 在空间点 x 处的具体取值, 亦可视为由权重参数 $\theta_\alpha \in \mathbb{R}^{p_\alpha}$ 驱动的解码神经网络 $\mathcal{N}_{\theta_\alpha} : \Omega \to \mathbb{R}^{d_\mathcal{U}}$ 的多通道输出结果.

引理 3.1. 设 $\mathcal{V}$ 与 $\mathcal{U}$ 为连续空间 $C(\Omega)$ 的非空闭子空间. 如上文所定义的编码器 $E^s_\mathcal{V}$ 与解码器 $D^s_\mathcal{U}$ 必然满足假设 *3.1(1)* 中的线性有界性要求. 此外, 满足假设 *3.1(2)* 要求的广义逆算子 $D^s_\mathcal{V}$ 与 $E^s_\mathcal{U}$ 可被显式地构造如下:

$$D^s_\mathcal{V}(z) = \sum_{k=1}^{d_\mathcal{V}} z_k f_k, \quad \forall\, z \in \mathbb{R}^{d_\mathcal{V}}, \quad \text{其中满足: } \int_\Omega f_k \, d\mu_\ell = \begin{cases} 1, & \text{若} k = \ell; \\ 0, & \text{若} k \neq \ell. \end{cases}$$

并且相应地有

$$E^s_\mathcal{U}(u) = \left(\int_\Omega u \, dm_i \right)_{i=1}^{d_\mathcal{U}}, \ \forall\, u \in \mathcal{U}, \quad \text{其中满足: } \int_\Omega \alpha_j \, dm_i = \begin{cases} 1, & \text{若} i = j; \\ 0, & \text{若} i \neq j. \end{cases}$$

证明: 上述恒等关系及正交性质可通过直接的代数计算与积分验证予以严谨证明. □

上述引理严格确立了标准有限差分网格求值算子与 Q_1 有限元基插值算子之间所具备的伪逆性质 (详见下述注记). 这一深层关联也从侧面印证了注记 3.1 中所给出的稳态椭圆方程示例在理论体系上的合法性与有效性.

注 3.2 (均匀有限差分网格与 Q_1 有限元基)**.** 设 Ω 为 $\mathbb{R}^d$ 空间中的规则超立方体, 并令离散化网格集为 $\Omega_h = \{x_i\}_{i=1,\dots,N^d}$, 其对应于空间步长为 h 的均匀有限差分剖分. 选取潜在维度 $d_\mathcal{V} = d_\mathcal{U} = N^d$. 此时, 基于网格节点的点值迹算子与基于 Q_1 有限元的基函数插值算子, 极其自然地构成了一对满足引理 *3.1* 要求的编码器-解码器算子对. 具体而言, 沿用引理 *3.1* 中的数学记号, 我们可作如下具体定义:

$$\mu_i = m_i = \delta_{x_i}, \qquad \alpha_i(x) = f_i(x) = \prod_{j=1}^{d} \phi_{i,j}(x), \qquad \forall\, i = 1, \dots, N^d, \quad \forall\, x \in \Omega,$$

其中 $\phi_{i,j}$ 表示在坐标轴方向 e_j 上以网格点 x_i 为对称中心, 且局部支撑区间长度为 $2h$ 的一维标准 P_1 帽函数 *(Hat function)*. 依据引理 *3.1*, 借助如下至关

重要的离散化点值恒等式:

$$\int_{\Omega} \alpha_i \, d\,\mu_j = \alpha_i(x_j) = \begin{cases} 1, & \text{若 } i = j; \\ 0, & \text{其他.} \end{cases}$$

我们可以推断, 假设 *3.1(2)* 对于由此实例构造而出的算子配对 $(E^s_{\mathcal{V}}, D^s_{\mathcal{V}})$ 与 $(E^s_{\mathcal{U}}, D^s_{\mathcal{U}})$ 是绝对成立的.

3.2.2 含时演化偏微分方程情形

现在, 我们在一般编码器-解码器架构所确立的范式下, 正式引出针对含时演化方程 (1.7) 所量身打造的 NODE-ONet 框架. 具体而言, 我们将系统地阐明如何科学选取匹配的潜在时空域空间 $\mathcal{V}_h$ 与 $\mathcal{U}_h$, 如何构造基于 NODE 诱导的神经代理模型 ψ_θ, 以及对时空编码器 $E_{\mathcal{V}}$ 与解码器 $D_{\mathcal{U}}$ 给出严格的数学刻画. 为表述严谨起见, 假设空间域 Ω 为紧集, 边界数据 u_b 固定, 且偏微分方程参数满足 $a, f \in \mathcal{K} \subset \mathcal{C}([0,T]; \mathcal{C}(\Omega))$, 初始条件 $u_0 \in \mathcal{C}(\Omega)$. 首先, 深受稳态情形下潜在空间构造的启发, 我们对含时系统的潜在空间作如下定义:

$$\mathcal{V}_h = \mathcal{C}([0,T]; \mathbb{R}^{d_{\mathcal{V}}}) \quad \text{与} \quad \mathcal{U}_h = \mathcal{C}([0,T]; \mathbb{R}^{d_{\mathcal{U}}}),$$

其中 $d_{\mathcal{V}}$ 与 $d_{\mathcal{U}}$ 均为给定的正整数常数. 随后, 我们提出由以下三大核心组件级联构成的 NODE-ONet 计算框架:

(NS1). **时空编码 (纯空间离散化):** 给定随时空演化的输入函数 $v \in \mathcal{C}([0,T]; \mathcal{C}(\Omega))$, 通过在每一时间切片 $t \in [0,T]$ 处对空间场 $v(t) \in \mathcal{C}(\Omega)$ 独立应用前述的稳态编码器 $E^s_{\mathcal{V}}$, 我们便自然地获得了适用于含时演化情形的算子编码器:

$$E_{\mathcal{V}} : \mathcal{C}([0,T]; \mathcal{C}(\Omega)) \to \mathcal{C}([0,T]; \mathbb{R}^{d_{\mathcal{V}}}),\ v \mapsto \boldsymbol{v} = (v_\ell)_{\ell=1}^{d_{\mathcal{V}}},$$

其中其各个分量具体展开为:

$$v_\ell(t) = (E^s_{\mathcal{V}}(v(t,\cdot)))_\ell = L_\ell(v(t,\cdot)), \quad \text{对于一切} t \in [0,T] \text{ 且} \ell = 1, \ldots, d_{\mathcal{V}}.$$

(NS2). **NODE 代理模型 (动力学逼近):** 给定连续编码后的特征轨迹 $\boldsymbol{v} \in \mathcal{C}([0,T]; \mathbb{R}^{d_{\mathcal{V}}})$, 令高维潜在轨迹 $\boldsymbol{\psi} \in \mathcal{C}([0,T]; \mathbb{R}^{d_{\mathcal{U}}})$ 严格遵循如下的神经

常微分方程 (NODE) 系统:

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = \mathcal{N}_{\theta_\psi}\big(\boldsymbol{\psi}(t),\, \mathcal{P}_v \boldsymbol{v}(t),\, t\big), & \text{对于 } t \in [0,T], \\ \boldsymbol{\psi}(0) = \mathcal{P}_u\, E_{\mathcal{U}}^{s}(u_0) \in \mathbb{R}^{d_{\mathcal{U}}}. \end{cases} \tag{3.12}$$

在此表达式中, $\mathcal{N}_{\theta_\psi} : \mathbb{R}^{d_{\mathcal{U}}} \times \mathbb{R}^{d_{\mathcal{U}}} \times \mathbb{R}_+ \to \mathbb{R}^{d_{\mathcal{U}}}$ 代表一个由可训练权值 $\theta_\psi \subset \mathbb{R}^{p_\psi}$ 完全参数化的非线性神经网络, 而 $\mathcal{P}_v$, $\mathcal{P}_u \in \mathbb{R}^{d_{\mathcal{U}} \times d_{\mathcal{V}}}$ 则为网络中可优化的降维或升维投影变换矩阵 (更为深入的物理编码设计原理请参见第 3.3 节). 基于此, 我们可以抽象出一个由参数簇 θ_Ψ, $\mathcal{P}_u$ 与 $\mathcal{P}_v$ 联合参数化的连续时间 NODE 算子, 记作:

$$\mathrm{NODE}\,(\theta_\Psi, \mathcal{P}_u, \mathcal{P}_v) : \mathcal{C}([0,T];\, \mathbb{R}^{d_{\mathcal{V}}}) \to \mathcal{C}([0,T];\, \mathbb{R}^{d_{\mathcal{U}}}), \quad \boldsymbol{v} \mapsto \boldsymbol{\psi},$$

并依靠该 NODE 模型所诱导出的神经网络泛函 $\psi_\theta := \mathrm{NODE}\,(\theta_\Psi, \mathcal{P}_u, \mathcal{P}_v)$ 来逼近未知的真实映射算子 $\psi = E_{\mathcal{U}} \circ \Psi^\dagger \circ D_{\mathcal{V}}$.

(NS3). **时空解码 (物理场重建):** 在获取了 NODE 诱导模型的预测输出轨迹 $\boldsymbol{\psi} := (\psi_j)_{j=1}^{d_{\mathcal{U}}} \in \mathcal{C}([0,T];\, \mathbb{R}^{d_{\mathcal{U}}})$ 后, 我们定义物理场的解码器 $D_{\mathcal{U}}$ 如下:

$$D_{\mathcal{U}} :\, \mathcal{C}([0,T];\, \mathbb{R}^{d_{\mathcal{U}}}) \to \mathcal{C}([0,T];\, \mathcal{C}(\Omega)),\ \boldsymbol{\psi} \mapsto \Psi,$$

其在每个时空节点上的重构值显式计算为:

$$D_{\mathcal{U}}(\boldsymbol{\psi})(t,x) = \sum_{j=1}^{d_{\mathcal{U}}} \alpha_j(x;\theta_\alpha)\, \psi_j(t) \quad \text{对于任意的} (t,x) \in [0,T] \times \Omega,$$

其中 $\{\alpha_j\}_{j=1}^{d_{\mathcal{U}}}$ 恰为在前述方程 (3.11) 中所定义的空间基函数族.

综合上述所有的核心组件, 针对任意的输入场 $v \in \mathcal{V} \subseteq \mathcal{C}([0,T];\, \mathcal{C}(\Omega))$ 以及时空域内的任意节点 $(t,x) \in [0,T] \times \Omega$, 我们所提出的 NODE-ONet 计算框架可被统一地表述为:

$$\Psi_{\text{NODE-ONet}}(v;\theta)(t,x) := \sum_{j=1}^{d_{\mathcal{U}}} \alpha_j(x;\theta_\alpha) \psi_j(t, \boldsymbol{v}; \theta_\psi, \mathcal{P}_u, \mathcal{P}_v),$$

其中空间编码 $\boldsymbol{v} = E_{\mathcal{V}}(v)$, 且参数集 $\theta = \{\theta_\psi, \theta_\alpha, \mathcal{P}_v, \mathcal{P}_u\}$ 囊括了网络中所有待优化的可训练参数, 其信息流架构与系统拓扑详见图 3.2.

紧接着, 我们通过如下两个注释, 进一步剖析编码器与解码器之间所内蕴的伪逆性质, 以及在 NODE-ONet 框架的设计环节 (NS2) 中引入 NODE 算子作为核心组件的深层物理机理与数学直觉.

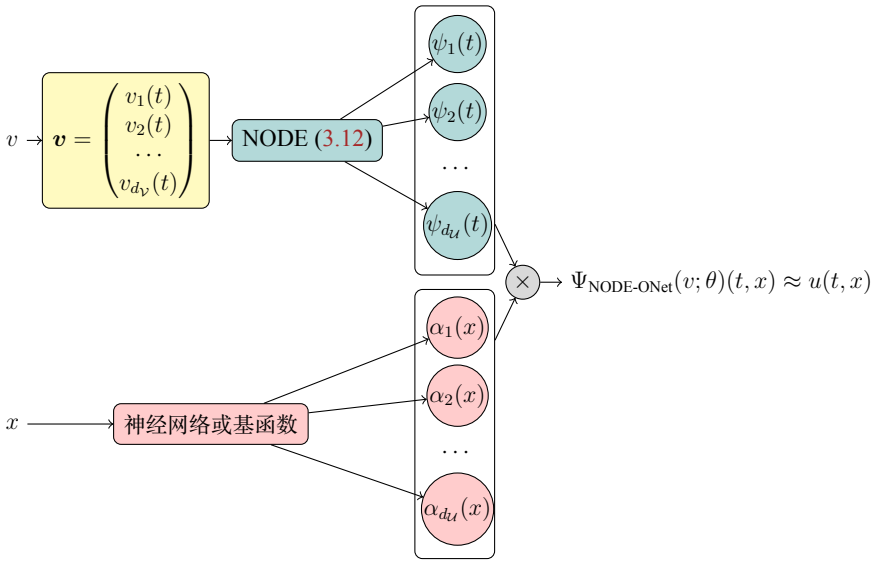


图 3.2: NODE-ONet 的通用计算架构图.

注 3.3. ***(广义伪逆性).*** 通过将引理 *3.1* 的结论推广并应用于上述稳态设定之中, 我们能够推导出: 此前在环节 *(NS1)* 与 *(NS3)* 中所构造的编码器 $E_\mathcal{V}$ 与解码器 $D_\mathcal{U}$, 均为假设 *3.1(2)* 下 $D_\mathcal{V}$ 与 $E_\mathcal{U}$ 的广义逆算子. 这些无限维泛函空间中的广义逆可被显式地刻画为:

$$
\begin{aligned}
D_\mathcal{V}(z)(t) &= D_\mathcal{V}^s(z(t)), && \forall\, t \in [0,T],\ \forall\, z \in \mathcal{C}\big([0,T];\mathbb{R}^{d_\mathcal{V}}\big),\\
E_\mathcal{U}(u)(t) &= E_\mathcal{U}^s(u(t,\cdot)), && \forall\, t \in [0,T],\ \forall\, u \in \mathcal{C}\big([0,T];\mathcal{C}(\Omega)\big),
\end{aligned}
$$

其中相关的静态逆算子 $D_\mathcal{V}^s$ 与 $E_\mathcal{U}^s$ 早在引理 *3.1* 中便已获得了严格定义.

注 3.4. ***(NODE 结构与半离散化格式的关联).*** 在框架环节 *(NS2)* 中所给出的由 *NODE* 系统诱导的神经网络代理 $\psi_\theta : \boldsymbol{v} \mapsto \boldsymbol{\psi}$, 其根本数学使命在于高逼真度地拟合极其复杂的复合映射算子 $\psi = E_\mathcal{U} \circ \Psi^\dagger \circ D_\mathcal{V}$. 然而, 相较于简单的静态稳态情形, 在含时演化系统中确立并证明 ψ_θ 的万能逼近性质往往面临着更为艰巨的理论挑战. 其内在困难在于: 映射 ψ 并非单调地将固定输入向量场的参数一致映射至某个简单的常微分方程 *(ODE)* 解轨迹. 尽管存在上述理论阻碍, 通过对原发偏微分方程系统 (1.7) 采用巧妙的空间半离散化格式 *(Semi-discrete schemes)*, 映射 ψ 依然能够获得理论上高精度的逼近解析表达. 为更直观地阐明这一关键点, 我们考察学习从强迫源项到真实解的系统算子 $\Psi_f : f \mapsto u$. 在该分析情境下, 固定输入 $v = f$, 进而我们将编码后的低维输入变量定义为 $f_h := E_\mathcal{V}(f) \in \mathcal{C}\big([0,T];\mathbb{R}^{d_\mathcal{V}}\big)$, 并为简便起见设定网络维度匹配

$d_{\mathcal{U}} = d_{\mathcal{V}}$. 如此一来, 复杂的复合映射轨迹 $\psi(f_h)$ 便能被原偏微分方程 (1.7) 的半离散截断版本的系统演化解 $u_h \in \mathcal{C}([0,T];\mathbb{R}^{d_{\mathcal{U}}})$ 在极高的数学保真度下予以逼近:

$$\frac{du_h}{dt} + \mathcal{L}_h[a](u_h) = f_h, \quad \forall t \in [0,T], \tag{3.13}$$

该方程组天然需配备适当的离散初始与边界条件, 且其中 $\mathcal{L}_h[a]$ 代表对原微分算子 $\mathcal{L}[a]$ 的空间高阶离散逼近. 值得注意的是, 理论轨迹 $\psi(f_h)$ 与半离散解 u_h 之间的残差恰恰完全等同于偏微分方程 (1.7) 进行空间投影离散时所不可避免的半离散化截断误差. 与涵盖时间与空间的全局全离散差分格式相比, 该空间半离散误差在量级上通常更为微小且易于进行严密的数学收敛性分析. 换言之, 在此框架下由 *NODE* 神经网络逼近所引入的非线性建模误差 ϵ *(*关于其一般性数学假设, 参见 *3.1(4))* 本质上已然包容并囊括了半离散化截断误差. 这种误差融合现象并不会对算子的整体学习构成任何实质性威胁, 因为误差项 ϵ 能够在数学上被无缝吸收合并进整体的编码-解码理论误差控制框架之中, 且绝不会削弱或改变总逼近误差的首阶渐近收敛行为. 针对这一整体性误差上界的极限严密分析与刻画, 也为本领域后续更深层次的理论工作指引了极具学术价值的研究方向.

3.2.3 NODE-ONet 的优化与模型训练

假设我们获取了包含具有丰富分布表征的不同输入函数特征 $\{v_i\}$ 以及空间采样节点 $\{x_j\}$ 的离线数据集 $\{v_i, x_j, \Psi^\dagger(v_i)(t,x_j)\}_{1\le i\le N_v, 1\le j\le N_x}$, 那么由代理模型 $\Psi_{\text{NODE-ONet}}$ 所诱导的连续时间均方损失泛函可显式表达为如下的连续积分形式:

$$\frac{1}{N_v N_x}\sum_{i=1}^{N_v}\sum_{j=1}^{N_x}\int_0^T \left\|\Psi_{\text{NODE-ONet}}(v_i;\theta)(t,x_j) - \Psi^\dagger(v_i)(t,x_j)\right\|_2^2 dt. \tag{3.14}$$

为了有效抑制并缓解深层神经网络固有的过拟合 (Overfitting) 风险, 我们在上述均方误差目标函数中额外施加了一个正则化惩罚泛函 $\mathcal{R}(\theta)$, 以此对 $\Psi_{\text{NODE-ONet}}$ 实施稳健的网络参数训练. 在具体的代码工程实现与数值计算中, 包含在等式 (3.14) 内部的连续时间积分项必须依靠数值求积算法 (Numerical quadrature schemes) 予以离散化逼近处理. 为此目的, 我们在演化时间域 $[0,T]$ 内部均匀或自适应地采样了一组离散时间网格节点

$\{t_k\}_{k=1}^{N_t} \subset [0,T]$, 随后调用高精度的常微分方程求解器引擎 (例如显式 Euler 差分法抑或高阶的 Runge-Kutta 算法) 来对潜在动力系统 NODE (3.12) 进行积分前推, 从而准确求得离散化代理预测集 $\{\Psi_{\text{NODE-ONet}}(v_i)(t_k,x_j)\}$, 其中足标变量满足界限 $1\le i\le N_v$, $1\le j\le N_x$, 且 $1\le k\le N_t$. 由此推导生成的适用于实机训练优化的最终离散损失函数可定格为:

$$\mathcal{L}(\theta)=\frac{1}{N_vN_xN_t}\sum_{i=1}^{N_v}\sum_{j=1}^{N_x}\sum_{k=1}^{N_t}\|\Psi_{\text{NODE-ONet}}(v_i)(t_k,x_j)-\Psi^{\dagger}(v_i)(t_k,x_j)\|_2^2+\lambda\mathcal{R}(\theta), \tag{3.15}$$

其中 $\lambda\ge 0$ 是用于平衡重构精度与模型复杂度的超参数 (Hyperparameter).

注 **3.5.** 值得指出是, 本文所构建的 *NODE-ONet* 框架完全可以通过更具物理直觉的物理信息机器学习 *(Physics-Informed Machine Learning)* 范式来赋予实现. 具体而言, 我们可以深度借鉴并融合物理信息神经网络 *(PINNs)* [20] 的惩罚机制思想: 我们可以抛弃对海量高精度标签数据的依赖, 转而通过在网络优化目标中直接最小化底层偏微分方程控制方程的物理残差项 *(Residuals)*, 而非仅仅盲目地通过单纯监督学习去最小化数据驱动的经验损失 (3.15).

注 **3.6.** 在标准的算子学习实验设定中, 用于充当训练集的输入样本函数 $\{v_i\}_{i=1}^{N_v}$ 往往是从某个预先人为指定的泛函分布空间中随机抽取而来的. 众所周知, 这种几乎纯粹依赖于数据驱动表征的机器学习模型 *(*这也同样涵盖了那些学习所得的偏微分方程神经算子*)* 其核心优势往往主要体现在数学意义上的函数插值 *(Interpolation)* 场景之中, 亦即当测试阶段的输入条件严格遵循或内含于训练分布域时, 模型将表现出无与伦比的最佳效能与精度 *(*相关讨论参见例如 *[108-110])*. 然而, 在不可预测的现实世界工程应用中, 模型往往被残酷地要求对外部分布 *(Out-of-Distribution, OOD)* 的极其极端的测试输入条件执行长期的外推演化 *(Extrapolation)* 预测, 而这种灾难性的分布偏移现象极易导致难以容忍的数值漂移误差乃至导致代理模型的全面崩溃与失效. 为了有效克服这一横亘在算子学习领域的共性外推瓶颈挑战, 并切实且显著地提升代理仿真模型的泛化可靠性与稳健度, 我们在未来完全能够将文献 *[110]* 等最新研究所开发的一系列前沿外推增强与鲁棒惩罚技术, 深度整合至当前的 *NODE-ONet* 基础框架体系之内.

3.2.4 与基准模型 DeepONets 的对比

纵观当前的算子学习学术领域, DeepONets [43] 无疑被公认并尊崇为最具代表性以及基准性的底层网络架构范式之一. 正如在前文中所推演, DeepONets 同样可以被非常自然地置于一般的编码器-解码器网络抽象框架体系内予以统一解释与理解, 相关的文献 [99] 亦从更为宏观的理论视角深刻强调并佐证了这一观点. 在本小节的论述中, 我们将首先在通用的编码器-解码器网络框架的宏观背景下, 对经典的 DeepONets 进行统一的数学公式化重构与表述, 随后将其与本文的核心成果 NODE-ONet 架构进行全方位且极具深度的对比剖析. 为了确保对比过程的直观性与公平性, 我们作出了一个合理的简化假设: 即假设 DeepONets 架构内部的特征提取分支网络 (Branch Net) 与时空评估主干网络 (Trunk Net) 均采用极其基础的浅层神经网络 (Shallow neural networks) 来予以实现. 这一极其巧妙的理论假设旨在摒弃过度繁冗的深层网络设计细节干扰, 使得我们能够更为直观, 清晰且透彻地从理论本质上剖析 DeepONets 相较于本文的 NODE-ONet 在核心参数特征复杂性以及底层计算推演成本上的根本性数学差异. 我们在此处统一沿用与上文第 3.2.2 节中所描述完全一致的符号设定体系. 于是, DeepONets 在处理含时演化偏微分方程时的编码-代理-解码信息流转流程可被精准重构且描述如下:

1. **DeepONets 中的时空编码 (传感器采样):** 给定一个随时空演化的动态输入函数 $v \in \mathcal{C}([0,T];\mathcal{C}(\Omega))$, 区别于仅在空间采样的策略, 该编码器被迫需要在 N_2 个离散的时间实例切片与 N_1 个固定的空间传感器探头节点上对 v 进行全局性的时空网格联合交叉采样. 这种全时空耦合的处理方式不可避免地导致输入特征的嵌入维度急剧膨胀至 $d_{\mathcal{V}} = N_1 N_2$, 其相应的编码映射算子则被强制定义为:

$$E_{\mathcal{V}}^{\mathrm{D}} : \mathcal{C}([0,T];\mathcal{C}(\Omega)) \to \mathbb{R}^{N_1 N_2}, \qquad v \mapsto \boldsymbol{v}^{\mathrm{D}} = (v(t_i, x_j))_{i,j}.$$

2. **DeepONets 中的神经代理模型 (分支网络 Branch Net):** 在接收了高维度的特征张量 $\boldsymbol{v}^{\mathrm{D}}$ 之后, 其发挥代理模型核心作用的是一个输出维度大小被限定为 $d_{\mathcal{U}}$ 的非线性神经网络单元:

$$\psi_j^{\mathrm{D}} = \sum_{k=1}^{P} w_j^k \, \sigma\big(\langle a_j^k, \boldsymbol{v}^{\mathrm{D}} \rangle + b_j^k\big), \qquad j = 1, \ldots, d_{\mathcal{U}},$$

其中 σ 表示非线性的逐层激活函数, 而整数 P 则表征了每一个独立输出坐标分量通道所对应隐层架构中极其庞大的神经元节点总数量.

3. **DeepONets 中的网络解码 (主干网络 Trunk Net):** 在接收并锁定 $\boldsymbol{\psi}^{\mathrm{D}} = (\psi_j^{\mathrm{D}})_{j=1}^{d_{\mathcal{U}}}$ 作为网络权重系数之后, 该主干解码器在数学上可被等效视为一个内部连接权重已被 $\boldsymbol{\psi}^{\mathrm{D}}$ 彻底固定死的时空耦合计算评估神经网络:

$$D_{\mathcal{U}}^{\mathrm{D}}\colon \mathbb{R}^{d_{\mathcal{U}}} \to \mathcal{C}([0,T];\mathcal{C}(\Omega)), \quad \boldsymbol{\psi}^{\mathrm{D}} \mapsto \Psi^{\mathrm{D}},$$
$$\Psi^{\mathrm{D}}(t,x) = \sum_{j=1}^{d_{\mathcal{U}}} \psi_j^{\mathrm{D}}\, \sigma\big(\langle \tilde{a}_j, (t,x)\rangle + \tilde{b}_j\big),$$

其中 $\tilde{a}_j \in \mathbb{R}^{1+d}$ 与 $\tilde{b}_j \in \mathbb{R}$ 为主干网络中待优化的基础可训练参数.

经过上述数学重构可以推断出, DeepONets 的逼近误差上限分析理论逻辑实际上同样可以非常自然地融入落入本文核心定理 3.1 所构建的泛函框架之中. 然而, DeepONet 模型架构与我们本文所创新性提出的 NODE-ONet 框架之间最为显著的区别, 在于二者在网络架构参数复杂性量级上的差距. 而正是这一底层的维度膨胀差异, 直接导致了两者在实机计算训练效率与长时间外推预测泛化表现上存在着巨大的鸿沟. 具体而言, DeepONets 框架在执行预测前, 被迫需要同时获取动态输入函数 v 在整个全局时空域网格内所有传感器探头节点上的精确点值. 这种时空未曾解耦的强行耦合读取机制不可避免地引发了一个极高维度的特征输入空间映射嵌入 (Embedding) $d_{\mathcal{V}} = N_1 N_2$, 从而直接致使前端核心分支网络 (Branch network) 内部需承载处理的总神经突触参数量以极其恐怖的量级飙升至: $P\,(2 + N_1 N_2)\, d_{\mathcal{U}}$. 此外, 通过简单的代数累加也很容易推导出尾端主干网络内部所蕴含的待优化参数量固定为 $(2+d)\, d_{\mathcal{U}}$. 因此, 宏观综合来看, 整个 DeepONets 代理模型的总体宏观参数复杂性阶数达到了:

$$\mathcal{O}\big(P\, d_{\mathcal{U}}\, N_1 N_2\big).$$

形成对比的是, 本文构建的 NODE-ONet 架构在其前置编码器模块中, 仅仅只使用了 N_1 个纯空间维度的离散探头节点来对物理空间域进行低维特征扫描提取与离散化, 亦即成功将输入特征张量维度压缩至 $d_{\mathcal{V}} = N_1$. 而其核心的时间动力学代理引擎 NODE 的输出特征隐空间维度依旧维持为 $d_{\mathcal{U}}$.

此时, 倘若我们公平地采用与 DeepONets 内部同等类型规格规模的简单浅层前馈神经网络来对时间导数向量场系统进行拟合逼近 (事实上这恰恰正是我们在前文研究所引入构建的半自治 SA-NODE 系统核心架构), 那么经严格测算我们所得出的 NODE-ONet 系统整体参数复杂性阶数仅为:

$$\mathcal{O}\big(P\, d_{\mathcal{U}}\,(d_{\mathcal{U}} + N_1)\big),$$

其中 P 同样作为统一衡量基准表示隐层特征空间的神经网络基底宽度. 在实际数值偏微分方程仿真应用中, 为了保障计算的稳定, 我们所选取设定的时空特征潜在空间维度 $d_{\mathcal{U}}$ 通常会远小于单维度空间网格的密集离散化剖分节点规模 N_1, 从而在乘积量级上远远呈指数级地小于全时空节点数乘积 $N_1 N_2$. 究其原因, 在于如果试图使用 DeepONets 精确且无误差地捕获并复现高频剧烈振荡的含时微分动态演化细节特征, 那么在时间轴上就必须被迫强行选取并设定足够极其庞大的细密离散时间探针节点数 N_2. 因此, 在面对相同计算难度的偏微分演化方程学习任务时, 相比于 DeepONets 极其冗余臃肿的全连接结构, NODE-ONet 在系统底层模型架构的特征维数复杂性层面上, 完美地实现了数量级规模的大幅降低.

这种在底层神经网络架构维度与维数复杂性上的降低与简化, 为该算子模型的实际应用催生带来了一系列优势: (1) 模型的大规模并行离线端训练收敛过程变得更为快速且具有高效性; (2) 高度精简且参数稀疏优化的轻量级模型架构, 极大程度上增强了网络模型自身抵抗规避训练过拟合 (Overfitting) 的能力; 以及 (3) 所学习训练拟合得到的非线性偏微分演化方程解代理模型, 在其进行前向大跨度时间推演预测计算的稳定性 (Temporal stability) 方面获得了质的提升, 这深刻归功于该 ODE 时序网络机制从微观数学骨架设计的根本上规避了模型在连续时间连续变量推演过程中去表征表达出那些缺乏物理意义根据的高频虚假振荡噪音演化行为. 所有这些基于前沿网络架构改良所推演出的理论算法优势, 均将会在下文第 3.4 节中所给出的真实物理复杂数值仿真模拟实验测试结果数据中, 得到进一步的验证.

3.3 物理编码的神经常微分方程 (NODEs)

NODE-ONet 框架是一个具有高度灵活性的通用架构, 它并未对其内部编码器与解码器的具体实现形式施加严格的结构约束. 因此, 现有的多种

编码器与解码器模型, 例如文献 [101-104] 中所提出的各类方案, 均可直接整合至该框架体系中. 此外, 通过选取不同的 NODE 核心组件, 我们可以从该框架衍生出多种适用于求解偏微分方程的 NODE-ONet 变体. 不难看出, NODE 组件在决定整体网络的计算有效性与数值效率方面起着关键作用. 然而, 若将现有的通用 NODE 模型直接应用于算子学习任务, 往往会面临诸多数值上的挑战. 为了克服这些局限性, 本节提出了一种物理编码 (Physics-encoded) 的 NODE 设计思路, 并通过具体算例详细阐述其数学表达与算法优势. 物理编码的 NODE 包含两个核心设计要素. 首先, 其可训练参数被设计为显式依赖于时间或完全与时间无关, 这一策略在降低模型参数复杂度的同时, 有效增强了模型在时间域上的泛化能力. 其次, 我们利用底层偏微分方程固有的结构特征, 将领域先验知识显式地编码进 NODE 的网络架构之中. 为使讨论更为具体, 我们将分别针对非线性反应-扩散方程与 Navier-Stokes 方程展开详细论述.

3.3.1 非线性反应-扩散方程

首先, 我们将通用方程 (1.7) 具体化为如下形式的反应-扩散方程:

$$\begin{cases} \partial_t u(t,x) - \nabla \cdot (D(t,x)\nabla u(t,x)) + R(t,x)u^2(t,x) = f(t,x), & \forall (t,x) \in [0,T] \times \Omega, \\ u(0,x) = u_0(x), & \forall x \in \Omega, \\ u(t,x) = u_b(t,x), & \forall (t,x) \in [0,T] \times \partial\Omega, \end{cases} \tag{3.16}$$

其中 $D : [0,T] \times \Omega \to \mathbb{R}$ 表示扩散系数, $R : [0,T] \times \Omega \to \mathbb{R}$ 表示反应速率系数, $f : [0,T] \times \Omega \to \mathbb{R}$ 为源项, $u_0(x)$ 与 $u_b(t,x)$ 分别对应初始条件与边界条件. 我们选取集合 $v := \{D, R, f, u_0\}$ 作为学习方程 (3.16) 解算子的输入参数. 偏微分方程 (3.16) 耦合了扩散过程, 非线性反应机制以及外部源项, 因而展现出丰富的动力学行为并具有广泛的应用背景. 该方程的解析性质与数值求解难度在很大程度上取决于上述各项之间的相互作用. 特别地, 我们注意到解 u 对扩散系数 D 呈现双线性依赖关系, 而对反应系数 R 则呈现非线性依赖关系. 受方程 (3.16) 这些内在结构性质的启发, 我们构造了如下形式的物理编

码 NODE:

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = \sum_{i=1}^{P} \left\{ (W_i \circ [\mathcal{P}_r \boldsymbol{R}(t)] + V_i) \circ \sigma \left(A_i \circ [\mathcal{P}_D \boldsymbol{D}(t)] \circ \boldsymbol{\psi} + A_i^n(t) + B_i \right) + \mathcal{P}_f \boldsymbol{f}(t) \right\}, \\ \boldsymbol{\psi}(0) = \mathcal{P}_u \boldsymbol{u}_0 \in \mathbb{R}^{d_{\mathcal{U}}}. \end{cases} \tag{3.17}$$

式中, $\sigma : \mathbb{R}^{d_{\mathcal{U}}} \to \mathbb{R}^{d_{\mathcal{U}}}$ 为激活函数, $\circ$ 代表 Hadamard 积 (逐元素乘积). 对于 $i = 1, \ldots, P$, 参数 $W_i, V_i, A_i, B_i \in \mathbb{R}^{d_{\mathcal{U}}}$ 均为不依赖于时间的可训练网络权重, $\mathcal{P}_D, \mathcal{P}_r, \mathcal{P}_f, \mathcal{P}_u \in \mathbb{R}^{d_{\mathcal{U}} \times d_{\mathcal{V}}}$ 为可训练的投影矩阵, 而向量 $\boldsymbol{u}_0, \boldsymbol{D}(t), \boldsymbol{R}(t), \boldsymbol{f}(t) \in \mathbb{R}^{d_{\mathcal{V}}}$ 分别是通过对 $u_0(x)$, $D(t,x)$, $R(t,x)$ 以及 $f(t,x)$ 进行空间离散化所得到的特征向量. 此外, $A_i^n(t) \in \mathbb{R}^{d_{\mathcal{U}}}$ 被设定为关于时间 t 的 n 次多项式, 其通项形式如下:

$$A_i^n(t) = \boldsymbol{a}_i^n t^n + \boldsymbol{a}_i^{n-1} t^{n-1} + \cdots + \boldsymbol{a}_i^1 t + \boldsymbol{a}_i^0, \tag{3.18}$$

其中系数 $\boldsymbol{a}_i^j \in \mathbb{R}^{d_{\mathcal{U}}}$ $(1 \le j \le n, 1 \le i \le P)$ 为待定参数.

值得指出的是, 偏微分方程 (3.16) 的核心结构特征——即 D 与 u 的双线性耦合, R 与 u 的非线性依赖, 以及源项 f 的加性结构——均在模型 (3.17) 中得到了显式的结构保留. 这种设计确保了神经网络模型与底层物理机制的一致性. 此种结构保留特性使得 NODE (3.17) 能够更有效地学习方程 (3.16) 的动力学演化规律, 并赋予了 NODE-ONet 在训练时间窗口之外进行外推预测的能力. 设 $\{t_k\}_{k=1}^{N_t} \subset [0, T]$ 为针对 (3.17) 进行数值积分的时间离散网格. 用于求解反应-扩散方程 (3.16) 的 NODE-ONet 整体架构如图 3.3 所示, 其中解码基函数 $\boldsymbol{\alpha} := \{\alpha_j(x)\}_{j=1}^{d_{\mathcal{U}}}$ 由神经网络 $\mathcal{N}_{\theta_\alpha}$ 输出生成.

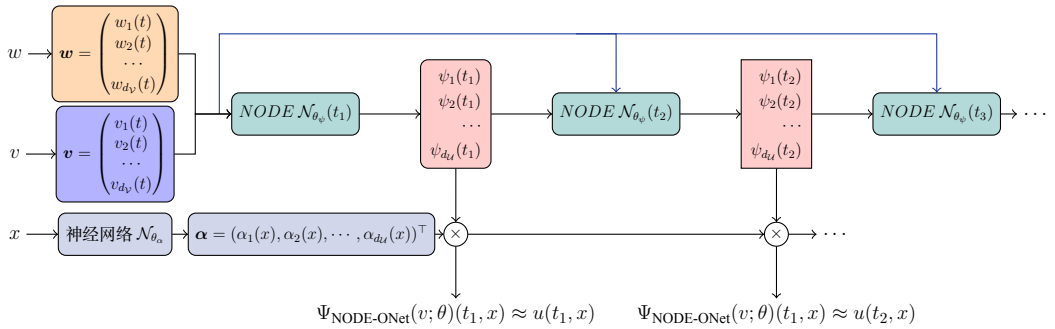


图 3.3: 用于学习方程 (3.16) 解算子 $\Psi^\dagger : v(t,x) \mapsto u(t,x)$ 的物理编码 NODE-ONet 架构图.

3.3.2 Navier-Stokes 方程

设空间区域 $\Omega = (0,1)^2$. 给定时间终端 $T > 0$, 考虑涡度-速度 (Vorticity-Velocity) 形式的不可压缩 Navier-Stokes 方程:

$$\begin{cases} \partial_t u(t,x) + \boldsymbol{V}(t,x) \cdot \nabla u(t,x) = \nu \Delta u(t,x) + f(t,x), & \forall (t,x) \in [0,T] \times \Omega, \\ u(t,x) = \nabla \times \boldsymbol{V}(t,x) \coloneqq \partial_{x_1} \boldsymbol{V}_2 - \partial_{x_2} \boldsymbol{V}_1, & \forall (t,x) \in [0,T] \times \Omega, \\ \nabla \cdot \boldsymbol{V}(t,x) = 0, & \forall (t,x) \in [0,T] \times \Omega, \\ u(0,x) = u_0(x), & \forall x \in \Omega, \end{cases} \tag{3.19}$$

并施加适当的边界条件. 在方程 (3.19) 中, $\boldsymbol{V}(t,x)$ 与 $u(t,x)$ 分别表示流体的速度场与涡度场; $\nu > 0$ 为运动粘性系数, $u_0(x)$ 为初始涡度分布, $f(t,x)$ 为外部强迫项. 利用算子 ∇ 与 $\nabla\times$ 的线性性质, 方程 (3.19) 中的对流项 $\boldsymbol{V}(t,x) \cdot \nabla u(t,x)$ 可被视作二次非线性项 $\mathcal{F}u \cdot \mathcal{G}u$, 其中 $\mathcal{F}$ 与 $\mathcal{G}$ 为待定的线性算子. 此外, 源项 f 以加性形式进入方程. 基于上述观察, 我们构建了如下形式的物理编码 NODE:

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = \displaystyle\sum_{i=1}^{P} \Big\{ W_i \circ \sigma\Big(A_i \circ \boldsymbol{\psi} + [C_i \circ \boldsymbol{\psi}] \circ [D_i \circ \boldsymbol{\psi}] + A_i^n(t) + B_i \Big) + \mathcal{P}_f \boldsymbol{f} \Big\}, \\ \boldsymbol{\psi}(0) = \mathcal{P}_u \boldsymbol{u}_0 \in \mathbb{R}^{d_{\mathcal{U}}}, \end{cases} \tag{3.20}$$

其中 $\sigma : \mathbb{R}^{d_{\mathcal{U}}} \to \mathbb{R}^{d_{\mathcal{U}}}$ 为激活函数, $\circ$ 表示 Hadamard 积. 对于 $i = 1, \ldots, P$, $W_i, A_i, B_i, C_i, D_i \in \mathbb{R}^{d_{\mathcal{U}}}$ 均为不依赖于时间的可训练参数, $\mathcal{P}_f, \mathcal{P}_u \in \mathbb{R}^{d_{\mathcal{U}} \times d_{\mathcal{V}}}$ 为可训练投影矩阵, $\boldsymbol{u}_0, \boldsymbol{f} \in \mathbb{R}^{d_{\mathcal{V}}}$ 分别是通过对 $u_0(x)$ 与 $f(x)$ 进行空间离散化所得到的特征向量. 此外, $A_i^n(t) \in \mathbb{R}^{d_{\mathcal{U}}}$ 同样由 (3.18) 定义. 模型 (3.20) 继承了模型 (3.17) 的优良数学性质. 其对应的 NODE-ONet 整体架构与图 3.3 类似, 故在此不再赘述. 得益于物理编码 NODE (3.17) 与 (3.20) 的引入, 相应的 NODE-ONets 显式地融合了特定物理系统的函数结构信息, 这一特性使其显著区别于传统的黑箱式算子学习方法. 这种融合策略不仅提升了计算效率, 同时也改善了模型的预测精度. 为了进一步展示物理编码 NODE 的实现细节, 并验证所构建的 NODE-ONets 在不同应用场景下的性能优势, 我们将在下一节呈现详细的数值实验结果.

3.4 数值模拟

本节旨在通过数值实验验证物理编码 NODE 及其衍生的 NODE-ONet 框架的有效性, 计算效率与适用性. 为此, 我们主要考察非线性反应-扩散方程 (3.16) 与 Navier-Stokes 方程 (3.19). 实验部分包含了与现有代表性算子学习方法的基准对比.

我们将方程 (3.16) 具体化为

$$\partial_t u(t,x) - \nabla\cdot(D(x)\nabla u(t,x)) + Ru^2(t,x) = f(x), \quad \forall (t,x)\in[0,1]\times[0,1], \tag{3.21}$$

并配以零初始条件与零边界条件. 该算例在偏微分方程算子学习的相关文献中已被广泛采用, 参见例如 [56, 43]. 遵循文献设定, 我们假设 D 与 f 与时间无关, 并取反应系数 $R=-0.01$. 本算例的测试目标包含三个方面. 首先, 在三种不同的输入配置 ($v=f(x)$, $v=D(x)$ 以及 $v=(D(x),f(x))$) 下测试物理编码 NODE (3.17) 学习解算子 $\Psi^\dagger : v(t,x)\mapsto u(t,x)$ 的性能. 其次, 通过与代表性算子网络 (包括 DeepONets [43] 及 MIONet [56]) 的对比, 分析 NODE-ONet 在数值误差与计算开销方面的表现. 最后, 检验模型的泛化能力与外推预测能力.

为了构建训练集, 我们首先生成高精度参考数据集. 在空间域上, 从高斯过程 $\mathcal{GP}(0,C)$ 中采样输入函数 $\{v_i\}_{i=1}^{N_{v_{\text{train}}}}$,

$$v_i \sim \mathcal{GP}(0,\,C)\,, \qquad C(x_1,x_2)=\exp\!\left(-\|x_1-x_2\|_2^2/(2l^2)\right), \tag{3.22}$$

其中 $l>0$ 是高斯协方差核的长度尺度; 较大的 l 对应更平滑的样本 v_i. 依据文献 [111] 的理论, 此类对输入函数的均匀采样可确保泛化误差随样本数量的增加而收敛, 进而在一定程度上缓解维数灾难. 对于每个 v_i, 我们在由 1001 个等距节点组成的细剖分空间网格上使用有限差分法求解 (3.21), 从而获得对应的高分辨率数值解 u_i. 为了在目标空间分辨率 N_x 下提取训练数据, 我们定义网格点 $\{x_j\}_{j=1}^{N_x}$ 并获取这些位置上的 u_i 值 (必要时进行插值), 从而生成数据元组 $(v_i, x_j, u_i(x_j))$. 集合 $\{(v_i,x_j,u_i(x_j))\}_{1\le i\le N_{v_{\text{train}}},\,1\le j\le N_x}$ 构成了用于学习解算子 $\Psi^\dagger : v\mapsto u$ 的训练数据集. 针对每个输入函数 v_i, 借鉴 DeepONets 的构造, 我们将编码器 $E_{\mathcal{V}}:\mathcal{V}\to[0,T]\times\mathbb{R}^{d_{\mathcal{V}}}$ 定义为: 对于任意 $t\in[0,T]$, $E_{\mathcal{V}}(v)=\boldsymbol{v}(t):=\{v_\ell(t)\}_{\ell=1}^{d_{\mathcal{V}}}\in\mathbb{R}^{d_{\mathcal{V}}}$, 其中 $v_\ell(t)=v(t,x_\ell)$, 且 $\{x_\ell\}_{\ell=1}^{d_{\mathcal{V}}}\subset\Omega$ 为一组固定的传感器采样点. 神经网络 $\mathcal{N}_{\theta_\alpha}$ 采用包含 2 个隐藏

层的全连接架构 (FCNN), 每层宽度为 $P = 100$, 激活函数选用 ReLU. 在方程 (3.17) 中设定 $A_i^n(t) = \boldsymbol{a}_i^1 t$. 所有的 NODE 模型均使用 ReLU 激活函数, 并采用具备 N_t 个时间步的显式欧拉方法进行离散求解. $\mathcal{N}_{\theta_\alpha}$ 与 NODE 的输出维度统一设定为 $d_{\mathcal{U}} = 50$. 除非另有说明, 我们采用学习率为 10^{-3} 的 ADAM 优化器, 对设定 $\lambda = 0$ 的损失函数 (3.15) 进行 1×10^5 个 epoch 的迭代优化以训练 NODE-ONet. $\mathcal{N}_{\theta_\alpha}$ 和 NODE 的参数通过 PyTorch 的默认机制进行初始化. 在精度测试阶段, 设定 $N_x = N_t = 100$, 并按照与训练集一致的采样流程, 随机生成 $N_{v_{\text{test}}}$ 个全新的输入函数 v. 本文采用如下两种指标来量化测试误差:

$$\begin{cases} \text{绝对误差} := \left(\dfrac{1}{N_{v_{\text{test}}} N_x N_t} \displaystyle\sum_{i=1}^{N_{v_{\text{test}}}} \sum_{j=1}^{N_x} \sum_{k=1}^{N_t} \| \Psi_{\text{NODE-ONet}}(v_i)(t_k, x_j) - \Psi^\dagger(v_i)(t_k, x_j) \|_2^2 \right)^{\frac{1}{2}}, \\ \text{相对误差} := \text{绝对误差} / \left(\dfrac{1}{N_{v_{\text{test}}} N_x N_t} \displaystyle\sum_{i=1}^{N_{v_{\text{test}}}} \sum_{j=1}^{N_x} \sum_{k=1}^{N_t} \| \Psi^\dagger(v_i)(t_k, x_j) \|_2^2 \right)^{\frac{1}{2}}. \end{cases} \tag{3.23}$$

接下来将给出利用物理编码 NODE (3.17) 求解方程 (3.21) 的具体实施过程, 并报告针对各类输入设定下的数值结果.

实验 1: 学习具备单一输入函数的解算子.

我们首先考察源项到解的算子逼近 $\Psi_f^\dagger : v = f \mapsto u$. 为此, 在方程 (3.21) 中固定 $D(t, x) = 0.01$, 输入函数 f 通过设定 $l = 0.5$ 的式 (3.22) 随机生成. 考虑到系数 D 与 R 均为常数, 物理编码 NODE (3.17) 可相应退化为如下形式:

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = \displaystyle\sum_{i=1}^{P} W_i \circ \sigma(A_i \circ \boldsymbol{\psi} + \boldsymbol{a}_i^1 t + B_i) + \mathcal{P}_f \boldsymbol{f}, \\ \boldsymbol{\psi}(0) = \mathbf{0} \in \mathbb{R}^{d_{\mathcal{U}}}. \end{cases} \tag{3.24}$$

我们设定 $d_{\mathcal{V}} = 20$, 并选取不同的 N_x, N_t 与 $N_{v_{\text{train}}}$ 参数组合来训练 NODE-ONet. 随后, 采样 $N_{v_{\text{test}}}$ 个新的输入函数 $f(x)$ 以评估学习得到的算子 Ψ_f^*. 表 3.1 汇总了不同参数设置下的测试绝对误差与相对误差. 图 3.4 展示了随机抽取的一个输入函数 $f(x)$ 所对应的数值预测结果.

为进行基准比较, 我们遵循文献 [43] 的设定实现了非堆叠 (Unstacked) DeepONet 模型. 特别地, 在 DeepONet 的训练阶段, 我们为每个 f 随机抽取 K 个时空节点作为主干网络的输入. 为保障公平比较, 本实验中所有算例训

	训练分辨率	可训练参数量	训练输入 f 数量	绝对误差	相对误差
NODE-ONet	$N_x = 10$ $N_t = 5$ $d_{\mathcal{V}} = 20$	27,550	100	4.248×10^{-3}	7.370×10^{-3}
NODE-ONet	$N_x = 100$ $N_t = 10$ $d_{\mathcal{V}} = 20$	27,550	500	1.368×10^{-3}	2.675×10^{-3}
DeepONet	$N_x = 100$ $N_t = 10$ $K = 50$ $d_{\mathcal{V}} = 100$	40,600	100	6.352×10^{-3}	1.230×10^{-2}
DeepONet	$N_x = 100$ $N_t = 100$ $K = 1,000$ $d_{\mathcal{V}} = 100$	40,600	500	1.313×10^{-3}	2.582×10^{-3}

表 3.1: NODE-ONet 与非堆叠 DeepONet 在学习方程 (3.21) 源项到解算子 $\Psi_f^\dagger : f(x) \mapsto u(t,x)$ 时的误差比较.

练步数均为 5×10^5; 测试分辨率均为 $N_x = 100$, $N_t = 100$; 测试输入 f 的数量均为 $10,000$. 误差结果已列于表 3.1 中, 针对同一随机输入函数 $f(x)$ 的数值结果如图 3.5 所示. 表 3.1 的数据表明, 在学习源项到解算子 Ψ_f 的任务中, NODE-ONet 在较粗的训练分辨率与较小的参数规模下即可达到相应的误差水平, 而 DeepONet 则需要相对更精细的离散化与更大的模型容量来获取相似的精度.

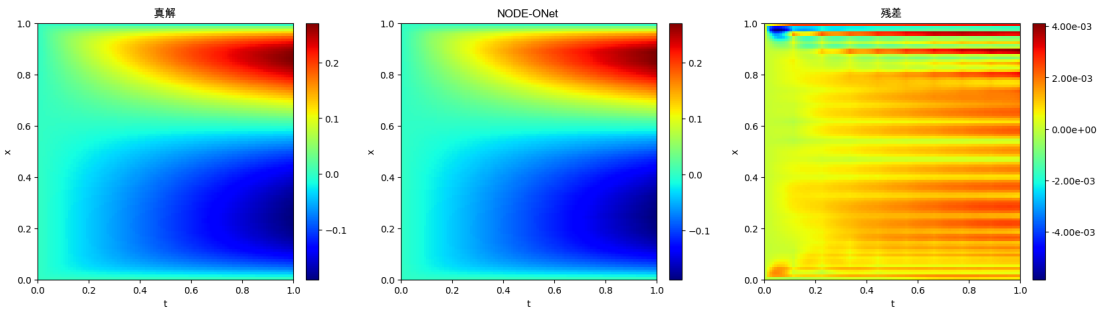


图 3.4: NODE-ONet 学习方程 (3.21) 源项到解算子 $\Psi_f^\dagger : f(x) \mapsto u(t,x)$ 针对随机 $f(x)$ 的测试结果. 训练参数: $N_x = 100, N_t = 10, N_{v_{\text{train}}} = 500$. 测试参数: $N_x = N_t = 100, N_{v_{\text{test}}} = 10,000$.

为了进一步考查模型的计算代价, 我们将 NODE-ONet 的性能与传统的基于网格的数值方法进行了对比. 基准参考解采用空间有限差分与时间 Crank–Nicolson 格式在细网格 ($N_x = 1000$, $N_t = 1000$) 上离散求解得到. 粗网格下的参考值由该高分辨率解均匀降采样生成. 作为比较基准, 我们考察了两种传统格式: (i) 结合显式 Euler 时间步的有限差分法, 以及 (ii) 结合隐式 Euler 格式的有限差分法. 对于 NODE-ONet, 我们使用在 $N_x^{\text{train}} = 100$ 且

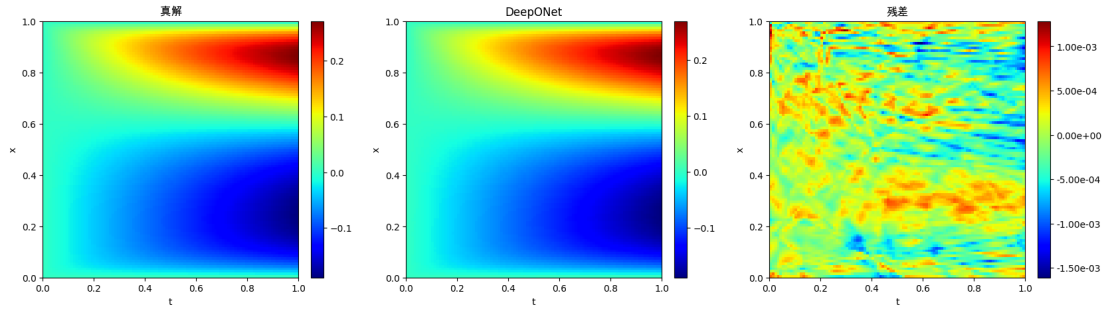


图 3.5: DeepONet 学习方程 (3.21) 源项到解算子 $\Psi_f^\dagger : f(x) \mapsto u(t,x)$ 针对随机 $f(x)$ 的测试结果. 训练: $N_x = 100, N_t = 100, N_{v_{\text{train}}} = 100$. 测试: $N_x = N_t = 100, N_{v_{\text{test}}} = 10,000$.

$N_t^{\text{train}} = 10$ 数据上预训练的模型直接进行推理, 无需二次训练. 表 3.2-3.4 报告了在不同 N_x 与 N_t 网格设定下, 各方法在相同硬件环境中的 CPU 运行时间以及相对于参考解的 L^2 误差. 结果均为 100 次独立实验的统计平均值. 观察可知, 在粗网格设定下 (如 $N_x = N_t = 10$), NODE-ONet 表现出了较低的绝对误差与较少的推理时间. 在较细的网格设定下 (如 $N_x = N_t = 100$), 显式 Euler 有限差分格式因违反 Courant-Friedrichs-Lewy (CFL) 稳定性条件而出现失稳现象, 此时 NODE-ONet 则保持了数值推演的稳定性.

N_x	N_t	显式 Euler 有限差分	隐式 Euler 有限差分	NODE-ONet
10	10	0.086	0.283	0.007
50	50	0.090	2.499	0.023
50	100	0.092	2.691	0.053
100	100	0.210	13.035	0.101

表 3.2: 求解 CPU 时间 (秒) 比较.

N_x	N_t	显式 Euler 有限差分	隐式 Euler 有限差分	NODE-ONet
10	10	8.597×10^{-3}	9.558×10^{-3}	2.797×10^{-3}
50	50	4.524×10^{-3}	5.234×10^{-3}	6.579×10^{-3}
50	100	2.643×10^{-3}	2.584×10^{-3}	7.053×10^{-3}
100	100	nan	2.932×10^{-3}	7.386×10^{-3}

表 3.3: 绝对 L^2 误差比较.

接下来考察扩散系数到解的算子逼近 $\Psi_D^\dagger : v = D \mapsto u$. 在方程 (3.21) 中设定 $f(x) = \sin(2\pi x)$, 并利用取 $l = 0.5$ 的式 (3.22) 生成输入函数 $D(x)$. 在此情形下, 物理编码 NODE (3.17) 转化为 (3.25) 的形式. 式 (3.25) 中的

N_x	N_t	显式 Euler 有限差分	隐式 Euler 有限差分	NODE-ONet
10	10	1.748×10^{-2}	1.943×10^{-2}	5.690×10^{-3}
50	50	8.848×10^{-3}	1.024×10^{-2}	1.289×10^{-2}
50	100	5.217×10^{-3}	5.103×10^{-3}	1.393×10^{-2}
100	100	nan	5.722×10^{-3}	1.442×10^{-2}

表 3.4: 相对 L^2 误差比较.

$\boldsymbol{f} := \{f(x_\ell)\}_{\ell=1}^{d_\mathcal{V}}$ 为已知函数 $f(x)$ 在离散点上的向量化表示.

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = \sum_{i=1}^{P} W_i \circ \sigma(A_i \circ [\mathcal{P}_D \boldsymbol{D}] \circ \boldsymbol{\psi} + \boldsymbol{a}_i^1 t + B_i) + \mathcal{P}_f \boldsymbol{f}, \\ \boldsymbol{\psi}(0) = \mathbf{0} \in \mathbb{R}^{d_\mathcal{U}}. \end{cases} \tag{3.25}$$

训练阶段参数设定为 $N_x = 100, N_t = 10, d_\mathcal{V} = 20$ 以及 $N_{v_{\text{train}}} = 1,000$. 测试阶段在网格 $N_x = N_t = 100$ 下对 $N_{v_{\text{test}}} = 10,000$ 个全新的输入函数 $D(x)$ 评估算子 Ψ_D^*. 最终测试的绝对误差和相对误差分别为 1.276×10^{-3} 与 3.919×10^{-3}, 随机输入函数 $D(x)$ 的预测表现如图 3.6 所示.

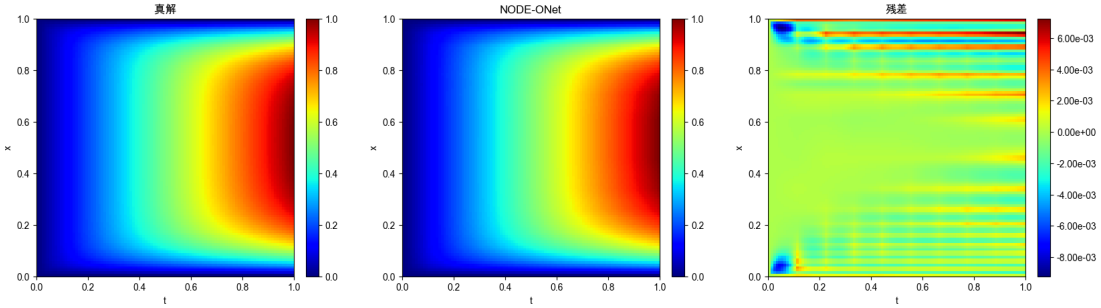


图 3.6: NODE-ONet 学习方程 (3.21) 扩散项到解算子 $\Psi_D^\dagger : D(x) \mapsto u(t,x)$ 针对随机 $D(x)$ 的测试结果.

• **实验 2: 学习具备多输入函数的解算子.** 现应用 NODE-ONet 求解具备双输入函数的算子映射 $\Psi_m^\dagger : v = \{D, f\} \mapsto u$. 依据文献 [56] 的设计, 令 $D(x) = 0.01(|g(x)| + 1)$, 其中辅助函数 $g(x)$ 与源项 $f(x)$ 均由取 $l = 0.2$ 的式 (3.22) 独立生成. 相应的物理编码 NODE 表达式如下:

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = \sum_{i=1}^{P} W_i \circ \sigma(A_i \circ [\mathcal{P}_D \boldsymbol{D}] \circ \boldsymbol{\psi} + \boldsymbol{a}_i^1 t + B_i) + \mathcal{P}_f \boldsymbol{f}, \\ \boldsymbol{\psi}(0) = \mathbf{0} \in \mathbb{R}^{d_\mathcal{U}}. \end{cases} \tag{3.26}$$

在 $d_{\mathcal{V}} = 20$ 且采用不同网格参数组合的条件下进行训练. 为考察模型能力, 将其与多输入算子学习基准网络 MIONet [56] 进行性能对比. 依照原文献设定实现的 MIONet 结构包含两个处理输入 D 和 f 的分支网络, 以及一个处理时空域 (t, x) 的主干网络. 为保障公平比较, 本实验中所有算例训练步数均为 1×10^5; 测试分辨率均为 $N_x = 100$, $N_t = 100$; 测试输入 $\{D, f\}$ 数量均为 $5,000$. 各模型的测试误差数据汇总于表 3.5, 一组随机输入对 $\{D(x), f(x)\}$ 的拟合结果分别见图 3.7 与 3.8. 在有限训练数据与粗分辨率 ($N_x = 50, N_t = 10, N_{v_{\text{train}}} = 100$) 的设定下, NODE-ONet 取得了相对较低的误差指标. 上述测试反映出模型在多输入算子学习场景下具备一定的计算可行性.

	训练分辨率	可训练参数量	训练输入 $\{D, f\}$ 数量	绝对误差	相对误差
NODE-ONet	$N_x = 50$ $N_t = 10$	28,550	100	2.362×10^{-2}	5.297×10^{-2}
NODE-ONet	$N_x = 100$ $N_t = 10$	28,550	1,000	4.626×10^{-3}	1.032×10^{-2}
MIONet	$N_x = 100$ $N_t = 100$	161,600	100	1.212×10^{-1}	2.661×10^{-1}
MIONet	$N_x = 100$ $N_t = 100$	161,600	1,000	9.491×10^{-3}	2.072×10^{-2}

表 3.5: NODE-ONet 与 MIONet 学习方程 (3.21) 双输入解算子 $\Psi_m^\dagger : \{D(x), f(x)\} \mapsto u(t, x)$ 的误差比较.

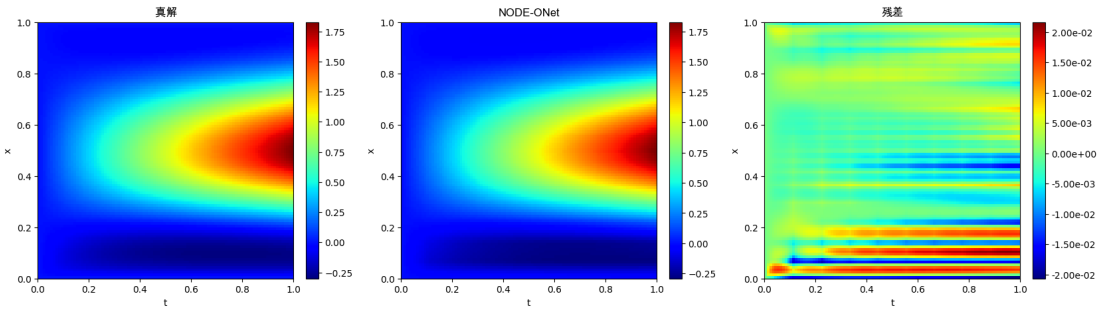


图 3.7: NODE-ONet 学习方程 (3.21) 双输入解算子 $\Psi_m^\dagger : \{D(x), f(x)\} \mapsto u(t, x)$ 针对随机输入对的测试结果. 训练: $N_x = 100, N_t = 10, N_{v_{\text{train}}} = 1000$. 测试: $N_x = N_t = 100, N_{v_{\text{test}}} = 5000$.

• **实验 3: 解码器 α 的重用与泛化.** 框架内的时空变量相互解耦, 使得预先训练好的空间解码器 $N_{\theta_\alpha^*}$ 有可能直接迁移至结构相近的其他偏微分方程学习任务中. 为考察此特性, 首先在原方程 (3.21) 设定 $D = 0.01, R = -0.01$ 的条件下, 训练 NODE-ONet 学习源项到解的算子 $\Psi_f^\dagger$. 利用 $l = 0.5$ 生成的 500 个

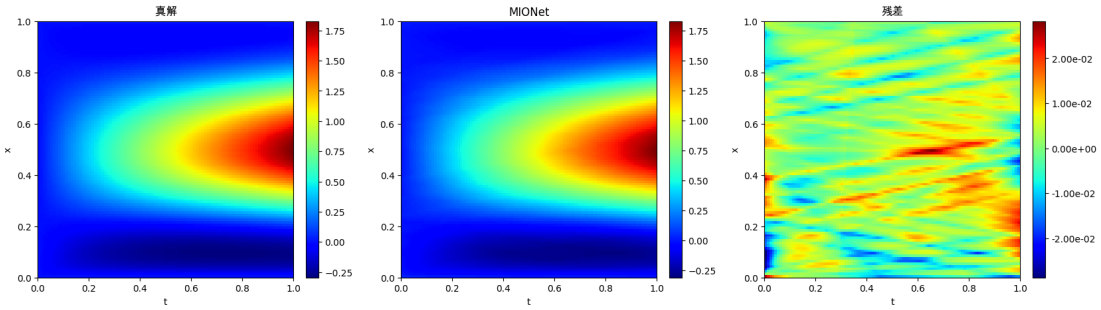


图 3.8: MIONet 学习方程 (3.21) 双输入解算子 $\Psi_m^\dagger : \{D(x), f(x)\} \mapsto u(t,x)$ 针对随机输入对的测试结果. 训练: $N_x = N_t = 100, N_{v_{\text{train}}} = 1000$. 测试: $N_x = N_t = 100, N_{v_{\text{test}}} = 5000$.

样本提取训练好的空间网络 $N_{\theta_\alpha^*}$. 随后, 设定新的物理参数 $D = 0.2, R = 0$, 使得偏微分方程退化为线性纯扩散方程. 以预训练的 $N_{\theta_\alpha^*}$ 固化为空间解码模块, 仅针对新算子优化 NODE 部分方程 (3.24). 利用 $l = 0.3$ 的分布重新采样 500 个样本用于重训练, 并在 10,000 个独立测试集上进行评估. 得到的测试绝对误差和相对误差分别为 1.836×10^{-3} 与 7.670×10^{-3}. 图 3.9 报告了对应的随机输入重建效果.

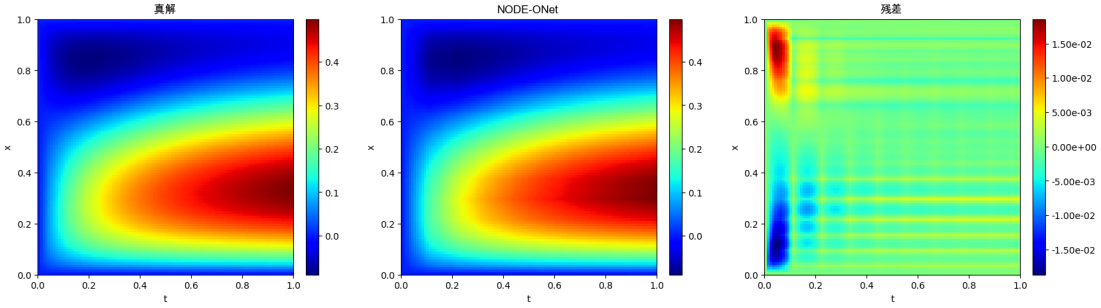


图 3.9: 固化预训练空间解码器 $\boldsymbol{\alpha}$ 学习方程 (3.21) 源项到解算子 $\Psi_f^\dagger : f(x) \mapsto u(t,x)$ 的测试结果.

• **实验 4: 外推时间域的预测演化.** 我们在扩展的时间区间 $t \in [0,2]$ 内测试已训练模型 (Ψ_f^* 与 Ψ_m^*) 的推演表现, 注意模型的原始训练域仅为 $t \in [0,1]$. 表 3.6 列举了各模型的外推测试误差. 图 3.10 – 3.13 直观对比了部分模型的外延推演过程. 结果显示, 在 $t \in [0,1]$ 的内插区间内, 几种模型均保持了较低的残差水平; 而在 $t \in [1,2]$ 的外推区间, 引入了物理结构约束的 NODE-ONet 所累计的预测偏离相对较缓.

• **实验 5: 空间基底选取的灵活性.** NODE-ONet 框架并未对空间解码过程施

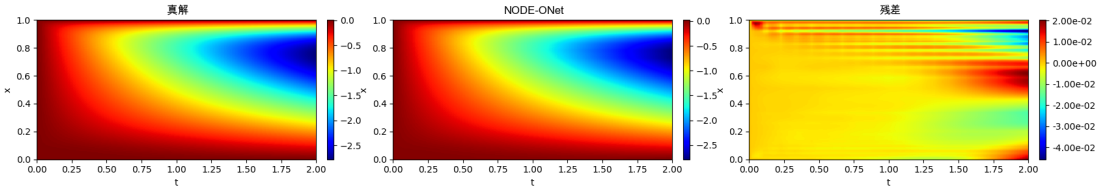


图 3.10: NODE-ONet 学习的源项到解算子 $\Psi_f^*: f(x) \mapsto u(t,x)$ 针对随机输入在扩展区间的外推结果. (测试时间域 $t \in [0,2]$; 训练时间域 $t \in [0,1]$)

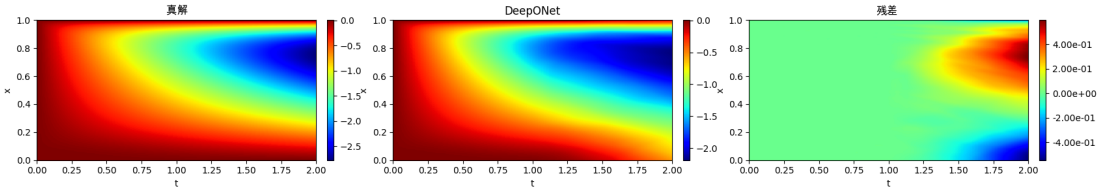


图 3.11: DeepONet 学习的源项到解算子 $\Psi_f^*: f(x) \mapsto u(t,x)$ 针对随机输入在扩展区间的外推结果. (测试时间域 $t \in [0,2]$; 训练时间域 $t \in [0,1]$).

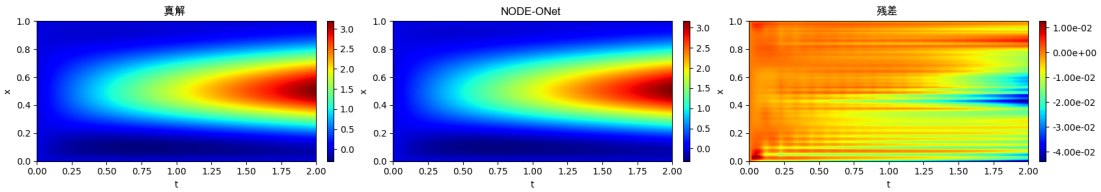


图 3.12: NODE-ONet 学习的双输入解算子 $\Psi_m^*: \{D(x), f(x)\} \mapsto u(t,x)$ 针对随机输入在扩展区间的外推结果. (测试时间域 $t \in [0,2]$; 训练时间域 $t \in [0,1]$).

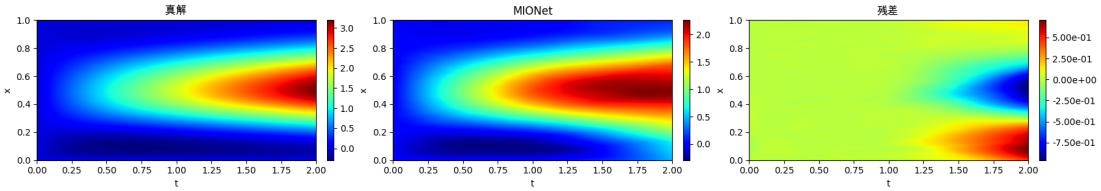


图 3.13: MIONet 学习的双输入解算子 $\Psi_m^*: \{D(x), f(x)\} \mapsto u(t,x)$ 针对随机输入在扩展区间的外推结果. (测试时间域 $t \in [0,2]$; 训练时间域 $t \in [0,1]$).

		训练输入函数数量	训练时间域	测试时间域	绝对误差	相对误差
Ψ_f^*	NODE-ONet	500	$t \in [0,1]$	$t \in [0,2]$	6.839×10^{-3}	7.113×10^{-3}
	DeepONet				2.302×10^{-1}	2.360×10^{-1}
Ψ_m^*	NODE-ONet	1,000	$t \in [0,1]$	$t \in [0,2]$	1.392×10^{-2}	1.732×10^{-2}
	MIONet				1.012×10^{-1}	1.251×10^{-1}

表 3.6: 方程 (3.21) 在扩展区间 $t \in [0,2]$ 内的预测误差. $\Psi_f^* : f(x) \mapsto u(t,x)$ 为单一源项解算子, $\Psi_m^* : \{D(x), f(x)\} \mapsto u(t,x)$ 为双输入解算子.

加严格的结构约束. 在此前的测试中, 空间解码算子 $\boldsymbol{\alpha}$ 由神经网络 $\mathcal{N}_{\theta_\alpha}$ 参数化生成. 此外, 根据第 3.2 节的定义, 亦可直接采用经典的解析基函数族替代网络输出. 作为验证, 我们将解码部分替换为截断的傅里叶基函数族, 在其余设定保持不变的条件下重新学习算子 $\Psi_f : f \mapsto u$. 该设置下的测试绝对误差与相对误差分别为 9.734×10^{-4} 和 1.908×10^{-3}. 图 3.14 展示了使用解析基底进行预测的结果. 上述测试反映了该计算框架在空间基底选取方面具有一定的灵活性.

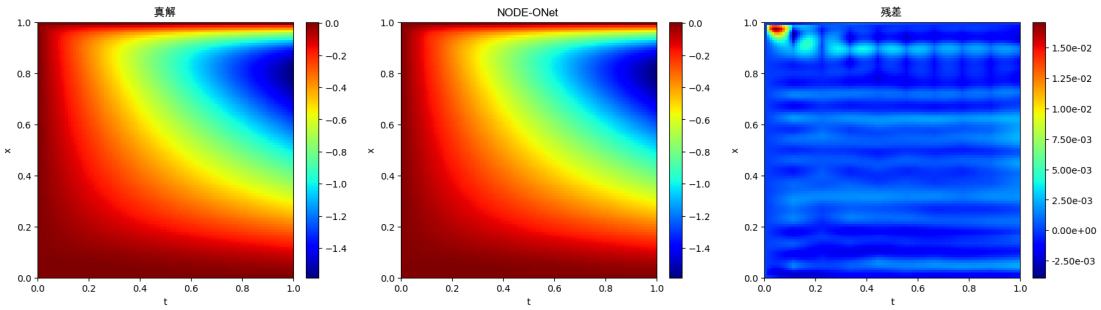


图 3.14: 结合傅里叶基函数的 NODE-ONet 学习方程 (3.21) 源项到解算子 $\Psi_f : f(x) \mapsto u(t,x)$ 的测试结果.

3.4.1 二维 Navier-Stokes 方程

本节通过测试二维 Navier-Stokes 方程 (3.19), 进一步验证 NODE-ONet 框架的计算可行性. 实验中, 设定运动粘性系数 $\nu = 0.001$, 并对涡度 u 施加周期边界条件. 依照文献 [42, 75] 的设定, 假设外力源项 f 不随时间变化. 基于此, 我们主要考察以下三种算子的学习任务:

- 初始条件到解的映射算子 $\Psi_i : u_0 \mapsto u$, 其中源项固定为 $f(x_1, x_2) = 0.1\sin(2\pi(x_1 + x_2)) + 0.1\cos(2\pi(x_1 + x_2))$;
- 源项到解的映射算子 $\Psi_f : f \mapsto u$, 其中初始条件固定为 $u_0(x_1, x_2) =$

$0.1\sin(2\pi(x_1+x_2))+0.1\cos(2\pi(x_1+x_2))$;

- 具备双输入函数的映射算子 $\Psi_m:\{u_0,f\}\mapsto u$.

上述算子学习任务的训练集由文献 [42] 提供的开源代码生成. 输入函数 u_0 与 f 从如下的高斯随机场中采样获得: $u_0\sim\mathcal{GP}(0,7^{3/2}(-\Delta+49I)^{-2.5})$ 且 $f\sim\mathcal{GP}(0,3^{3/2}(-\Delta+49I)^{-5})$. 在所有实验中, 空间解码网络 $\mathcal{N}_{\theta_\alpha}$ 均采用包含 4 个隐藏层的全连接架构 (FCNN), 单层宽度设为 $2,000$, 并使用 ReLU 激活函数. 在 NODE 模型 (3.20) 中, 设定 $A_i^n(t)=\boldsymbol{a}_i^1 t$ 以及网络宽度 $P=2,000$. 所有 NODE 模型同样使用 ReLU 激活函数, 并利用含 N_t 个时间步的显式欧拉格式进行积分演化. 网络 $\mathcal{N}_{\theta_\alpha}$ 与 NODE 的输出潜在维度统一设为 $d_{\mathcal{U}}=200$. 训练阶段, 我们依次采用 ADAM 与 L-BFGS 优化算法最小化经验损失函数 (3.15), 其中正则化项设定为 $\mathcal{R}(\theta)=\|\theta\|_1$ 且 $\lambda=10^{-5}$. 网络参数均采用 PyTorch 的默认机制进行初始化. 模型的测试精度依然通过式 (3.23) 中定义的绝对误差与相对误差进行量化. 具体的实验参数配置汇总于表 3.7.

	训练步数	学习率	训练分辨率	测试分辨率	$d_{\mathcal{V}}$	$d_{\mathcal{U}}$	N_v
Ψ_i	ADAM 5×10^5 LBFGS 100	10^{-4}	$N_x=50^2$ $N_t=10$	$N_x=100^2$ $N_t=100$	50^2	200	1000 (训练) 200 (测试)
Ψ_f	ADAM 5×10^5 LBFGS 100	10^{-4}	$N_x=50^2$ $N_t=10$	$N_x=100^2$ $N_t=100$	50^2	200	1000 (训练) 200 (测试)
Ψ_m	ADAM 5×10^5 LBFGS 100	10^{-4}	$N_x=50^2$ $N_t=20$	$N_x=100^2$ $N_t=100$	50^2	200	1000 (训练) 200 (测试)

表 3.7: 学习方程 (3.19) 解算子的实验参数设置.

	训练 时间域	测试 时间域	$[0,10]$ 内的 绝对测试误差	$[0,10]$ 内的 相对测试误差	$[0,20]$ 内的 绝对测试误差	$[0,20]$ 内的 相对测试误差
Ψ_i	$t\in[0,10]$	$t\in[0,20]$	1.396×10^{-2}	3.053×10^{-2}	5.860×10^{-2}	8.491×10^{-2}
Ψ_f	$t\in[0,10]$	$t\in[0,20]$	2.751×10^{-3}	3.180×10^{-2}	7.379×10^{-3}	7.167×10^{-2}
Ψ_m	$t\in[0,10]$	$t\in[0,20]$	1.320×10^{-2}	8.827×10^{-2}	1.208×10^{-2}	8.857×10^{-2}

表 3.8: 学习方程 (3.19) 解算子的测试与预测精度误差.

三个算子 Ψ_i, Ψ_f 与 Ψ_m 的数值测试结果分别汇总在表 3.8 以及图 3.15–3.17 中. 结果表明, NODE-ONet 能够学习方程 (3.19) 的非线性演化算子, 并在内插时间域 $t\in[0,10]$ 内输出稳定的数值结果. 此外, 在外推时间域 $t\in[10,20]$ 内的评估中, 模型亦保持了一定的预测精度. 上述测试数据初步验证了该计算框架在不同参数设定下求解含时流体偏微分方程的可行性.

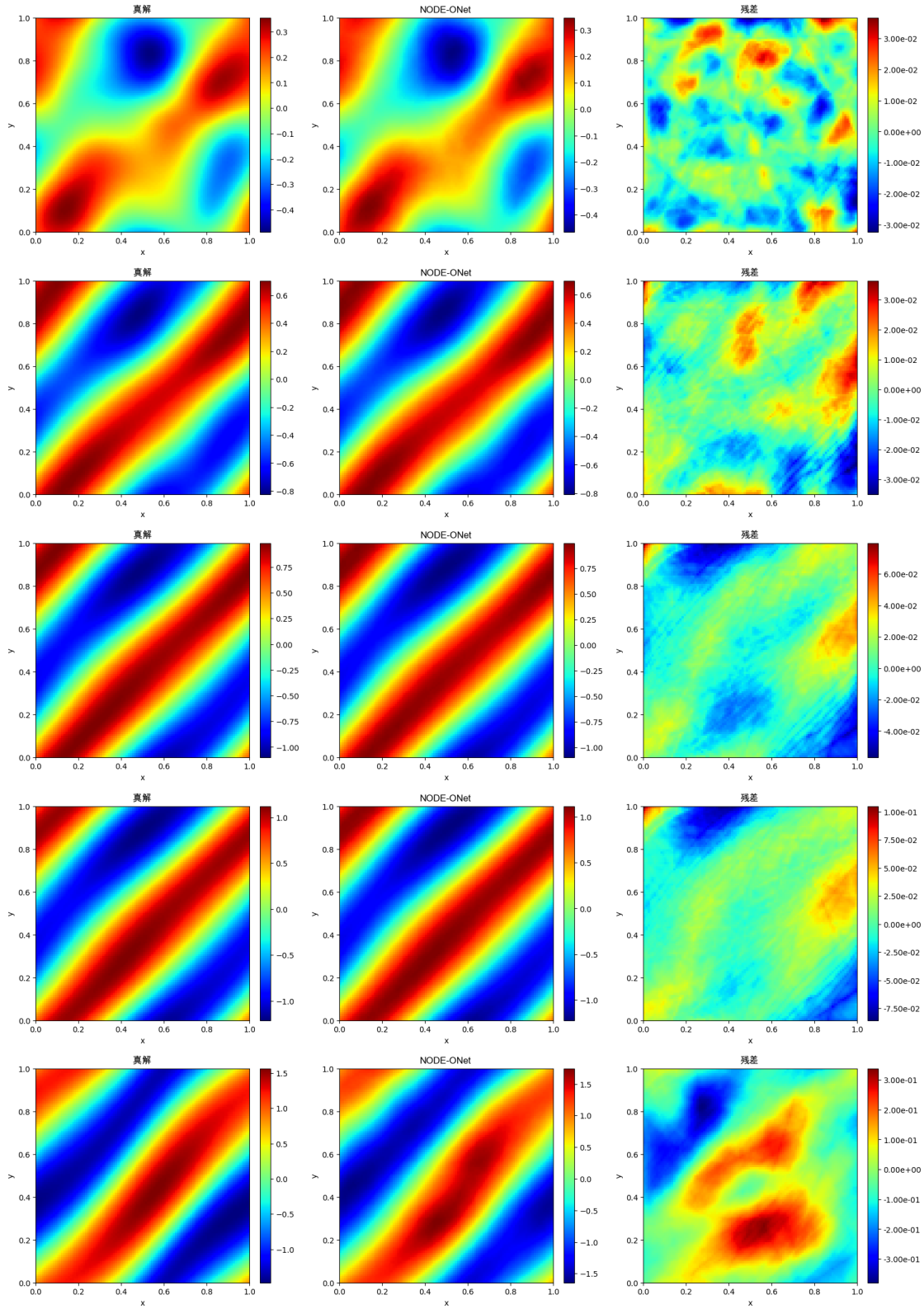


图 3.15: 学习方程 (3.19) 初始条件到解算子 $\Psi_i : u_0(x) \mapsto u(t,x)$ 针对随机输入函数的数值结果. 从上到下依次为: $t = 2, 6, 10$ (在内插时间域内测试), 以及 $t = 12, 20$ (超出训练时间域的外推预测).

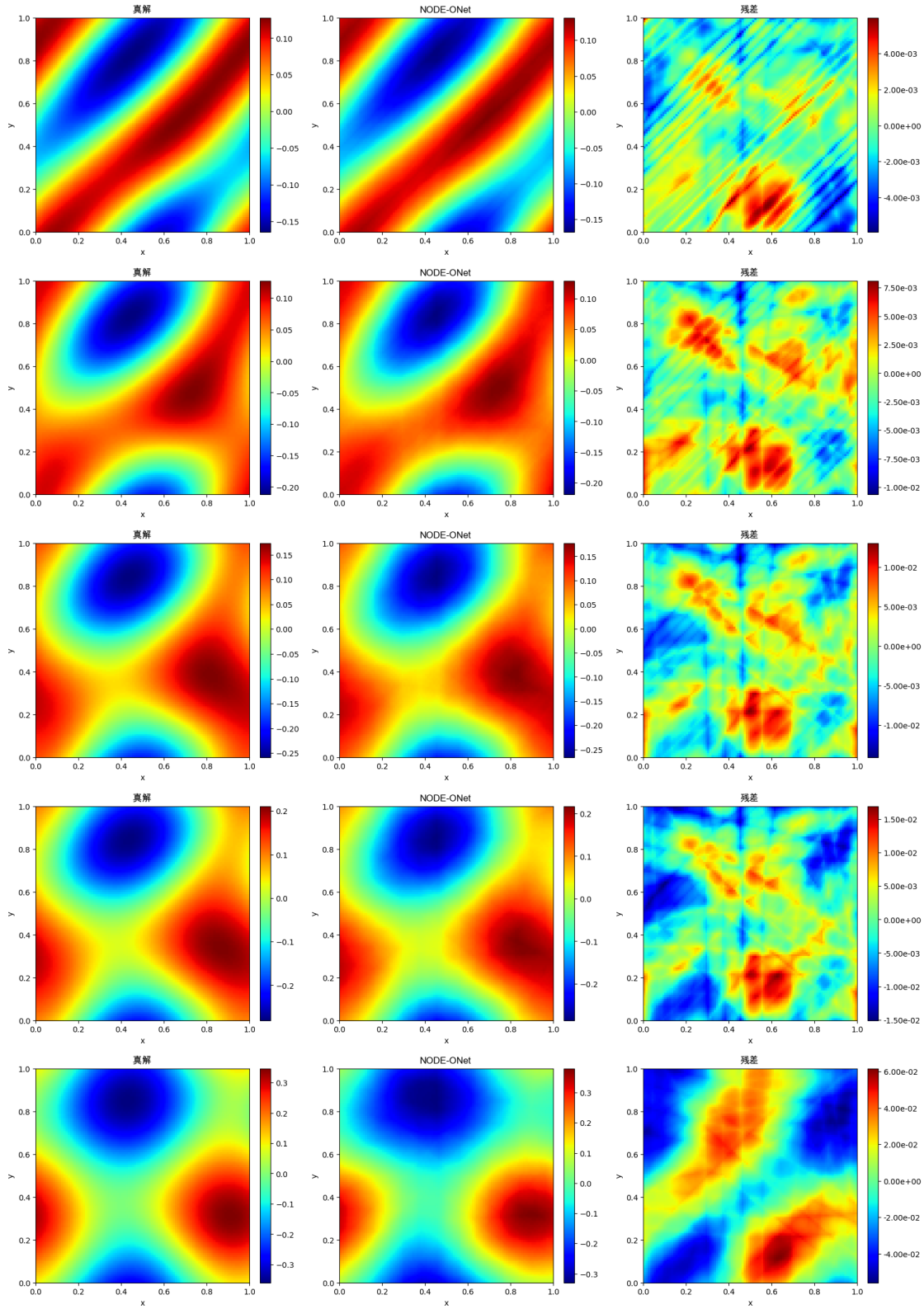


图 3.16: 学习方程 (3.19) 源项到解算子 $\Psi_f : f(x) \mapsto u(t,x)$ 针对随机输入函数的数值结果. 从上到下依次为: $t = 2, 6, 10$ (在内插时间域内测试), 以及 $t = 12, 20$ (超出训练时间域的外推预测).

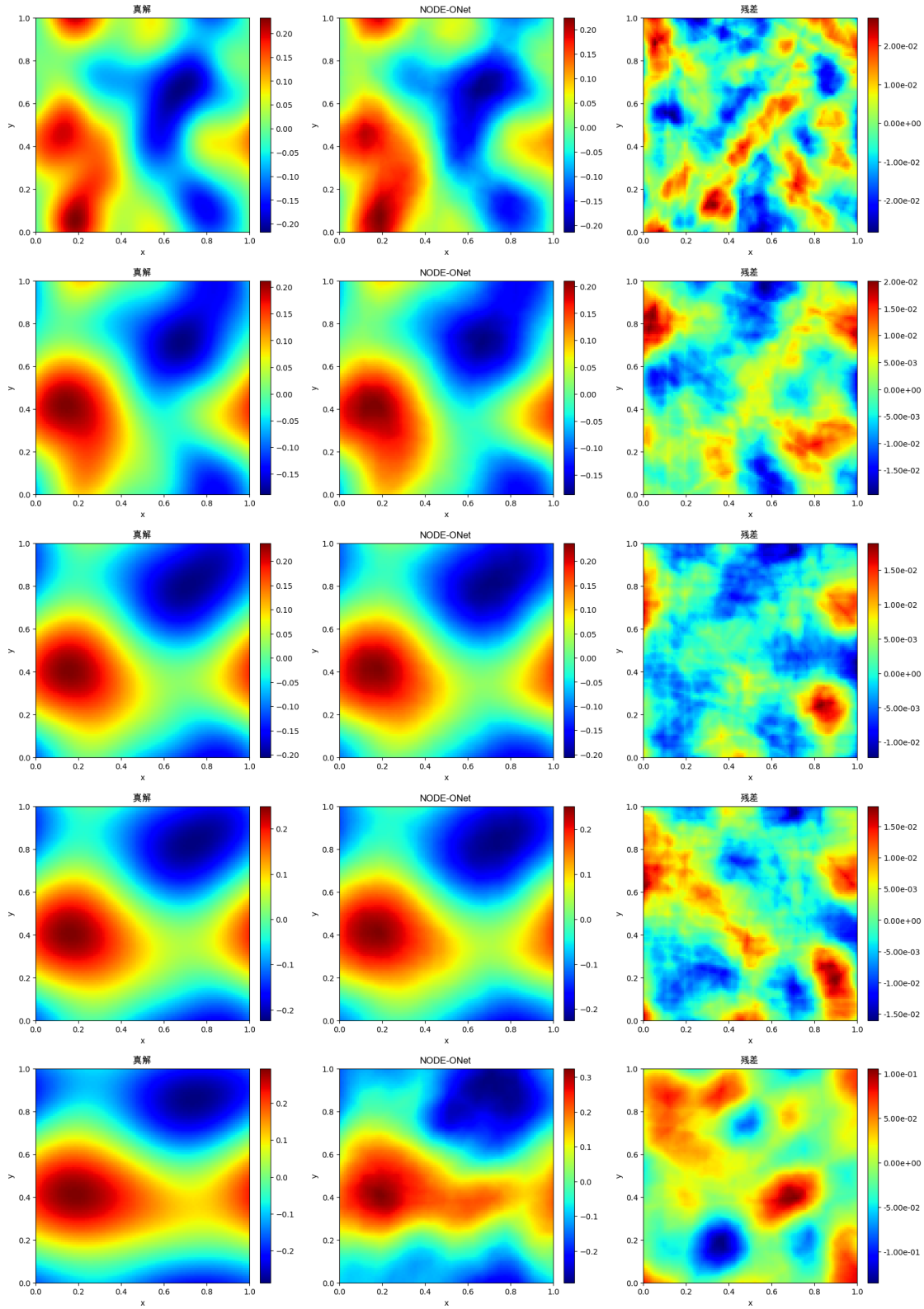


图 3.17: 学习方程 (3.19) 双输入解算子 $\Psi_m : \{u_0(x), f(x)\} \mapsto u(t,x)$ 针对随机输入对的数值结果. 从上到下依次为: $t = 2, 6, 10$ (在内插时间域内测试), 以及 $t = 12, 20$ (超出训练时间域的外推预测).

第 4 章 总结与展望

4.1 本文总结

本文围绕神经常微分方程 (NODE) 对动力系统的逼近理论与求解偏微分方程的算子学习展开了系统的理论分析与算法设计. 整体研究工作与主要贡献可归纳为以下两个部分:

第一部分为针对动力系统逼近的半自治神经常微分方程 (SA-NODEs). 这部分提出了 SA-NODEs 这一用于刻画与逼近动力系统的新型网络框架. 在理论层面, 我们确立了 SA-NODEs 的万能逼近性质与收敛速率, 从严格的数学意义上证明了其对复杂动力系统的逼近能力. 同时, 明确了 SA-NODEs 的训练过程可自然转化为一个最优控制问题, 即通过调节网络参数来重构生成数据的潜在动力学演化方程. 在数值计算层面, 我们在各类常微分方程以及传输方程上对该框架进行了测试. 实验结果表明, 依靠参数显式的时间依赖性设计, SA-NODEs 有效降低了模型复杂度. 相较于经典 NODEs, 该方法在所需参数量与训练轮次较少的情况下, 能够在精度与计算效率上保持稳定的数值表现, 并在训练数据规模受限时展现出一定的泛化鲁棒性.

第二部分为针对偏微分方程解算子学习的深度神经常微分方程算子网络 (NODE-ONet). 该框架通过将神经 ODEs 整合到编码器-解码器架构中, 实现了空间与时间变量的解耦, 这一设计理念与求解含时偏微分方程的传统时间步进数值方法相吻合. 其核心机制在于物理编码 NODEs 的设计, 能够将底层偏微分方程的内在结构性质显式编码至网络架构之中. 围绕该框架, 我们的主要贡献包括:

- **理论基础:** 建立了一般编码器-解码器网络的误差分析范式, 为无限维算子逼近提供了误差分析, 并为 NODE-ONets 的架构设计提供了理论指导.

- **物理编码机制:** 通过在可训练参数中引入显式的时间依赖, 并嵌入特定偏微分方程的领域先验知识 (如项间非线性依赖关系), 构造了低复杂度的物理编码 NODEs.

- **计算效率:** 数值实验表明, 在学习非线性反应-扩散方程与 Navier-Stokes 方程的算子映射 (尤其是多输入函数算子) 时, NODE-ONets 在数值精度与模型复杂度方面相较于 DeepONets 和 MIONet 等现有方法具备一定的计算优势.

- **外推与泛化能力:** 借助物理结构约束, 模型实现了超越训练时间域的稳定外推. 此外, 框架内部的时空解耦设计允许预训练的编码器/解码器在不重新训练的情况下迁移至结构相似的偏微分方程任务中.

4.2 未来展望

尽管本文在动力系统逼近与偏微分方程算子学习方面取得了一定的理论与计算进展, 但相关领域仍存在若干值得深入探索的开放性问题. 结合本文的两个主要部分, 未来的研究工作可围绕以下方向展开:

针对第一部分 (SA-NODEs) 的后续研究:

- **保结构动力系统的针对性逼近:** 针对具有特定物理约束的动力系统 (如梯度流, 哈密顿系统, 自治系统及具周期性轨道的方程), 现有的逼近理论仍有改进空间. 未来的工作可研究 SA-NODEs 能否内生地捕获这些特定的结构先验. 例如, 在哈密顿系统设定下, 可结合近期工作 [112] 的结论, 在概率意义上寻求更紧锐的逼近界限.

- **长期预测能力与理论边界:** 鉴于 SA-NODE 的系数不随时间演化, 其解轨道可自然延拓至数据可用时间 T 之外. 评估模型在 $t > T$ 阶段追踪真实物理演化的能力退化速率是一个重要问题. 这一任务与处理离散时间序列的回声状态网络 (ESNs) 存在联系 [113]. 借助于 Lyapunov 指数理论, 并参考 [114-115] 的分析框架, 未来有望建立 SA-NODE 在连续无限时间区间上的理论预测误差估计.

- **广义动力系统的降维建模:** SA-NODEs 的应用不局限于精确的 ODE 系统, 亦可用于复杂非确定性数据的插值拟合. 当观测数据源自受随机扰动的复杂物理系统时, 利用 SA-NODEs 重构出底层的确定性主导动力学成分, 或开发适应随机行为的混合型 SA-NODE 扩展架构, 是科学机器学习领域极具潜力的应用方向.

针对第二部分 (NODE-ONet) 的后续研究:

- **误差分析的深化:** 第 3.1.3 节建立的误差分析主要针对一般的编码器-解码器架构. 对包含连续时间积分的 NODE-ONet 框架开展针对特定偏微分方程的整体误差分析 (涵盖半离散截断误差与非线性网络逼近误差) 具有较高的数学技术难度, 仍需在未来的理论工作中予以完善 (参见注记 3.4).

- **物理编码架构的寻优准则:** 实验中所采用的物理编码 NODEs (如方程 (3.17)) 并非唯一构造. 例如, 针对多输入问题 (3.26), 一种等效的物理编码 NODE 可设计为:

$$\begin{cases} \dot{\boldsymbol{\psi}}(t) = -P_D \cdot \mathrm{Diag}(D) \cdot P_D^{\top} \cdot \boldsymbol{\psi} + \sum_{i=1}^{P} \left\{ W_i \odot \sigma\left(A_i \odot \boldsymbol{\psi} + \boldsymbol{a}_i^1 t + B_i\right) + \mathcal{P}_f \boldsymbol{f} \right\}, \\ \boldsymbol{\psi}(0) = \mathbf{0} \in \mathbb{R}^{d_U}, \end{cases}$$

  其中 $\mathrm{Diag}(\cdot) : \mathbb{R}^{d_V} \to \mathbb{R}^{d_V \times d_V}$ 为对角矩阵算子, $P_D \in \mathbb{R}^{d_U \times d_V}$. 该替代形式在模型复杂度与数值表现上与式 (3.26) 相当. 因此, 建立严格的数学准则以指导并确立针对特定偏微分方程的最优物理编码结构, 具有重要的理论价值.

- **复杂应用场景的扩展:** 首先, 将 NODE-ONet 框架扩展至偏微分方程最优控制与反问题求解是一个自然的研究方向. 此类问题通常涉及正向状态方程与反向伴随方程的耦合, 设计能够同时稳定捕获双向时间演化规律的 NODE 架构是该方向的核心挑战. 其次, 针对包含波传播特性的双曲型偏微分方程, 探索并引入二阶 NODEs 架构 (可借鉴文献 [116] 的相关思想) 亦是完善该算子学习框架的重要一环.

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]. Advances in Neural Information Processing Systems: Vol. 25. Curran Associates, Inc., 2012.

[2] VASWANI A, SHAZEER N, PARMAR N, USZKOREIT J, JONES L, GOMEZ A N, KAISER L U, POLOSUKHIN I. Attention is all you need[C]. Advances in Neural Information Processing Systems: Vol. 30. Curran Associates, Inc., 2017.

[3] KARNIADAKIS G E, KEVREKIDIS I G, LU L, PERDIKARIS P, WANG S, YANG L. Physics-informed machine learning[J]. Nature Reviews Physics, 2021, 3: 422-440.

[4] WIENER N. Tauberian theorems[J]. Ann. of Math. (2), 1932, 33(1): 1-100.

[5] CYBENKO G. Approximation by superpositions of a sigmoidal function[J]. Math. Control Signals Systems, 1989, 2(4): 303-314.

[6] HORNIK K. Approximation capabilities of multilayer feedforward networks[J]. Neural Networks, 1991, 4(2): 251-257.

[7] LESHNO M, LIN V Y, PINKUS A, SCHOCKEN S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function[J]. Neural Networks, 1993, 6(6): 861-867.

[8] PINKUS A. Approximation theory of the MLP model in neural networks[M]. Acta Numer.: Vol. 8 Acta numerica, 1999. Cambridge Univ. Press, Cambridge, 1999: 143-195.

[9] BARRON A R. Universal approximation bounds for superpositions of a sigmoidal function [J]. IEEE Trans. Inform. Theory, 1993, 39(3): 930-945.

[10] E W, MA C, WU L. The Barron space and the flow-induced function spaces for neural network models[J]. Constr. Approx., 2022, 55(1): 369-406.

[11] SIEGEL J W, XU J. Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks[J]. Found. Comput. Math., 2024, 24(2): 481-537.

[12] LI Y, LU S, MATHÉ P, PEREVERZEV S V. Two-layer networks with the ReLU^k activation function: Barron spaces and derivative approximation[J]. Numer. Math., 2024, 156(1): 319-344.

[13] LI Y, LU S. Function and derivative approximation by shallow neural networks[A/OL]. arXiv (2024). https://arxiv.org/abs/2407.05078.

[14] SIEGEL J W. Optimal Approximation of Zonoids and Uniform Approximation by Shallow Neural Networks[J]. Constr. Approx., 2025, 62(2): 441-469.

[15] DEVORE R, HANIN B, PETROVA G. Neural network approximation[J]. Acta Numer., 2021, 30: 327-444.

[16] CHEN R T Q, RUBANOVA Y, BETTENCOURT J, DUVENAUD D K. Neural ordinary differential equations[C]. Advances in Neural Information Processing Systems: Vol. 31. 2018.

[17] HE K, ZHANG X, REN S, SUN J. Deep residual learning for image recognition[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016.

[18] ALLAIRE G. Numerical mathematics and scientific computation: numerical analysis and optimization[M]. Oxford University Press, Oxford, 2007: xvi+455.

[19] MAUROY A, MEZIĆ I, SUSUKI Y. Lecture notes in control and information sciences: the Koopman operator in systems and control—concepts, methodologies and applications[M]. Springer, Cham, [2020] ©2020: xxiii+556.

[20] RAISSI M, PERDIKARIS P, KARNIADAKIS G E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations[J]. J. Comput. Phys., 2019, 378: 686-707.

[21] RUBANOVA Y, CHEN R T Q, DUVENAUD D K. Latent ordinary differential equations for irregularly-sampled time series[C]. Advances in Neural Information Processing Systems: Vol. 32. 2019.

[22] RUIZ-BALET D, ZUAZUA E. Neural ODE control for classification, approximation, and transport[J]. SIAM Rev., 2023, 65(3): 735-773.

[23] KIDGER P. On neural differential equations[A/OL]. arXiv (2022). https://arxiv.org/abs/2202.02435.

[24] GREYDANUS S, DZAMBA M, YOSINSKI J. Hamiltonian neural networks[C]. Advances in Neural Information Processing Systems: Vol. 32. 2019.

[25] CRANMER M, GREYDANUS S, HOYER S, BATTAGLIA P, SPERGEL D, HO S. Lagrangian neural networks[A/OL]. arXiv (2020). https://arxiv.org/abs/2003.04630.

[26] LOYA A A, SERINO D A, TANG Q. Structure-preserving neural ordinary differential equations for stiff systems[A/OL]. arXiv (2025). https://arxiv.org/abs/2503.01775.

[27] CELLEDONI E, MURARI D, OWREN B, SCHÖNLIEB C B, SHERRY F. Dynamical systems-based neural networks[J]. SIAM J. Sci. Comput., 2023, 45(6): A3071-A3094.

[28] MURARI D, CELLEDONI E, OWREN B, SCHÖNLIEB C B, SHERRY F. Structure preserving neural networks based on ODEs[C]. The Symbiosis of Deep Learning and Differential Equations II. 2022.

[29] ESTEVE-YAGÜE C, GESHKOVSKI B. Sparsity in long-time control of neural ODEs[J]. Systems Control Lett., 2023, 172: PaperNo.105452,14.

[30] E W. A proposal on machine learning via dynamical systems[J]. Commun. Math. Stat., 2017, 5(1): 1-11.

[31] AGRACHEV A, SARYCHEV A. Control on the manifolds of mappings with a view to the deep learning[J]. J. Dyn. Control Syst., 2022, 28(4): 989-1008.

[32] ÁLVAREZ LÓPEZ A, SLIMANE A H, ZUAZUA E. Interplay between depth and width for interpolation in neural ODEs[J]. Neural Networks, 2024, 180: 106640.

[33] MASSAROLI S, POLI M, PARK J, YAMASHITA A, ASAMA H. Dissecting neural odes [C]. Advances in Neural Information Processing Systems: Vol. 33. 2020: 3952-3963.

[34] SANDER M, ABLIN P, PEYRÉ G. Do residual neural networks discretize neural ordinary differential equations?[C]. Advances in Neural Information Processing Systems: Vol. 35. 2022: 36520-36532.

[35] GESHKOVSKI B, ZUAZUA E. Turnpike in optimal control of PDEs, ResNets, and beyond [J]. Acta Numer., 2022, 31: 135-263.

[36] ELAMVAZHUTHI K, GHARESIFARD B, BERTOZZI A L, OSHER S. Neural ODE control for trajectory approximation of continuity equation[J]. IEEE Control Syst. Lett., 2022, 6: 3152-3157.

[37] SONG Y, YUAN X, YUE H. Accelerated primal-dual methods with enlarged step sizes and operator learning for nonsmooth optimal control problems[A/OL]. arXiv (2023). https://arxiv.org/abs/2307.00296.

[38] SONG Y, YUAN X, YUE H, ZENG T. An operator learning approach to nonsmooth optimal control of nonlinear PDEs[A/OL]. arXiv (2024). https://arxiv.org/abs/2409.14417.

[39] TANYU D N, NING J, FREUDENBERG T, HEILENKÖTTER N, RADEMACHER A, IBEN U, MAASS P. Deep learning methods for partial differential equations and related parameter identification problems[J]. Inverse Problems, 2023, 39(10): PaperNo.103001,75.

[40] WANG S, BHOURI M A, PERDIKARIS P. Fast PDE-constrained optimization via self-supervised operator learning[A/OL]. arXiv (2021). https://arxiv.org/abs/2110.13297.

[41] E W, YU B. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems[J]. Commun. Math. Stat., 2018, 6(1): 1-12.

[42] LI Z, KOVACHKI N B, AZIZZADENESHELI K, LIU B, BHATTACHARYA K, STUART A, ANANDKUMAR A. Fourier neural operator for parametric partial differential equations [C]. International Conference on Learning Representations. 2021.

[43] LU L, JIN P, PANG G, ZHANG Z, KARNIADAKIS G E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators[J]. Nature Machine Intelligence, 2021, 3: 218-229.

[44] SIRIGNANO J, SPILIOPOULOS K. DGM: a deep learning algorithm for solving partial differential equations[J]. J. Comput. Phys., 2018, 375: 1339-1364.

[45] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2: 359-366.

[46] KIDGER P, LYONS T. Universal approximation with deep narrow networks[C]. ABERNETHY J, AGARWAL S. Proceedings of Machine Learning Research: Vol. 125 Proceedings of Thirty Third Conference on Learning Theory. PMLR, 2020: 2306-2327.

[47] KAWAGUCHI K, BENGIO Y, KAELBLING L. Generalization in deep learning[M]. Mathematical aspects of deep learning. Cambridge Univ. Press, Cambridge, 2023: 112-148.

[48] NEYSHABUR B, BHOJANAPALLI S, MCALLESTER D, SREBRO N. Exploring generalization in deep learning[C]. Advances in Neural Information Processing Systems: Vol. 30. 2017.

[49] CUOMO S, SCHIANO DI COLA V, GIAMPAOLO F, ROZZA G, RAISSI M, PICCIALLI F. Scientific machine learning through physics-informed neural networks: where we are and what's next[J]. J. Sci. Comput., 2022, 92(3): PaperNo.88,62.

[50] GAO Y, SONG Y, TAN Z, YUE H, ZENG S. Prox-PINNs: a deep learning algorithmic framework for elliptic variational inequalities[A/OL]. arXiv (2025). https://arxiv.org/abs/2505.14430.

[51] LAI M C, SONG Y, YUAN X, YUE H, ZENG T. The hard-constraint PINNs for interface optimal control problems[J]. SIAM J. Sci. Comput., 2025, 47(3): C601-C629.

[52] SONG Y, YUAN X, YUE H. The ADMM-PINNs algorithmic framework for nonsmooth PDE-constrained optimization: a deep learning approach[J]. SIAM J. Sci. Comput., 2024, 46(6): C659-C687.

[53] LU L, PESTOURIE R, YAO W, WANG Z, VERDUGO F, JOHNSON S G. Physics-informed neural networks with hard constraints for inverse design[J]. SIAM J. Sci. Comput., 2021, 43(6): B1105-B1132.

[54] BHATTACHARYA K, HOSSEINI B, KOVACHKI N B, STUART A M. Model reduction and neural networks for parametric PDEs[J]. SMAI J. Comput. Math., 2021, 7: 121-157.

[55] KOVACHKI N, LI Z, LIU B, AZIZZADENESHELI K, BHATTACHARYA K, STUART A, ANANDKUMAR A. Neural operator: learning maps between function spaces with applications to PDEs[J]. J. Mach. Learn. Res., 2023, 24: PaperNo.[89],97.

[56] JIN P, MENG S, LU L. MIONet: learning multiple-input operators via tensor product[J]. SIAM J. Sci. Comput., 2022, 44(6): A3490-A3514.

[57] WANG S, WANG H, PERDIKARIS P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets[J]. Science Advances, 2021, 7(40): eabi8605.

[58] LI Z, KOVACHKI N, AZIZZADENESHELI K, LIU B, BHATTACHARYA K, STUART A, ANANDKUMAR A. Neural operator: graph kernel network for partial differential equations [A/OL]. arXiv (2020). https://arxiv.org/abs/2003.03485.

[59] NELSEN N H, STUART A M. The random feature model for input-output maps between Banach spaces[J]. SIAM J. Sci. Comput., 2021, 43(5): A3212-A3243.

[60] CAO Q, GOSWAMI S, KARNIADAKIS G E. Laplace neural operator for solving differential equations[J]. Nature Machine Intelligence, 2024, 6: 631-640.

[61] YANG L, LIU S, MENG T, OSHER S J. In-context operator learning with data prompts for differential equation problems[J]. Proc. Natl. Acad. Sci. USA, 2023, 120(39): PaperNo.e2310142120,10.

[62] AZIZZADENESHELI K, KOVACHKI N, LI Z, LIU-SCHIAFFINI M, KOSSAIFI J, ANANDKUMAR A. Neural operators for accelerating scientific simulations and design[J]. Nature Reviews Physics, 2024, 6: 320-328.

[63] HWANG R, LEE J Y, SHIN J Y, HWANG H J. Solving PDE-constrained control problems using operator learning[C]. Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 36. 2022: 4504-4512.

[64] KOBAYASHI K, DANIELL J, ALAM S B. Improved generalization with deep neural operators for engineering systems: path towards digital twin[J]. Engineering Applications of Artificial Intelligence, 2024, 131: 107844.

[65] KOBAYASHI K, ALAM S B. Deep neural operator-driven real-time inference to enable digital twin solutions for nuclear energy systems[J]. Scientific Reports, 2024, 14: 2101.

[66] LIU N, LI X, RAJANNA M R, REUTZEL E W, SAWYER B, RAO P, LUA J, PHAN N, YU Y. Deep neural operator enabled digital twin modeling for additive manufacturing[J]. Advances in Computational Science and Engineering, 2024, 2: 174-201.

[67] LV K, WANG J, ZHANG Y, YU H. Neural operators for adaptive control of freeway traffic [J]. Automatica J. IFAC, 2025, 182: PaperNo.112553,15.

[68] PATHAK J, SUBRAMANIAN S, HARRINGTON P, RAJA S, CHATTOPADHYAY A, MARDANI M, KURTH T, HALL D, LI Z, AZIZZADENESHELI K, HASSANZADEH P, KASHINATH K, ANANDKUMAR A. FourCastNet: a global data-driven high-resolution weather model using adaptive fourier neural operators[C]. Proceedings of the Platform for Advanced Scientific Computing Conference. 2023.

[69] CHEN T, CHEN H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems[J]. IEEE Transactions on Neural Networks, 1995, 6: 911-917.

[70] CAI S, WANG Z, LU L, ZAKI T A, KARNIADAKIS G E. DeepM&Mnet: inferring the electroconvection multiphysics fields based on operator approximation by neural networks [J]. J. Comput. Phys., 2021, 436: PaperNo.110296,17.

[71] LIN C, LI Z, LU L, CAI S, MAXEY M, KARNIADAKIS G E. Operator learning for predicting multiscale bubble growth dynamics[J]. The Journal of Chemical Physics, 2021, 154.

[72] YIN M, BAN E, REGO B V, ZHANG E, CAVINATO C, HUMPHREY J D, KARNIADAKIS G E. Simulating progressive intramural damage leading to aortic dissection using DeepONet: an operator–regression neural network[J]. Journal of the Royal Society Interface, 2022, 19: 20210670.

[73] FAROUGHI S A, PAWAR N M, FERNANDES C, RAISSI M, DAS S, KALANTARI N K, MAHJOUR S K. Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: fluid and solid mechanics[J]. Journal of Computing and Information Science in Engineering, 2024, 24: 040802.

[74] HAO Z, LIU S, ZHANG Y, YING C, FENG Y, SU H, ZHU J. Physics-informed machine learning: a survey on problems, methods and applications[A]. arXiv (2022).

[75] LU L, MENG X, CAI S, MAO Z, GOSWAMI S, ZHANG Z, KARNIADAKIS G E. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data[J]. Comput. Methods Appl. Mech. Engrg., 2022, 393: PaperNo.114778,35.

[76] GARG S, CHAKRABORTY S. Variational bayes deep operator network: a data-driven bayesian solver for parametric differential equations[A/OL]. arXiv (2022). https://arxiv.org/abs/2206.05655.

[77] KRISHNAPRIYAN A, GHOLAMI A, ZHE S, KIRBY R, MAHONEY M W. Characterizing possible failure modes in physics-informed neural networks[C]. Advances in Neural Information Processing Systems: Vol. 34. 2021: 26548-26560.

[78] WANG S, SANKARAN S, PERDIKARIS P. Respecting causality for training physics-informed neural networks[J]. Comput. Methods Appl. Mech. Engrg., 2024, 421: PaperNo.116813,17.

[79] RUIZ-BALET D, AFFILI E, ZUAZUA E. Interpolation and approximation via momentum ResNets and neural ODEs[J]. Systems Control Lett., 2022, 162: PaperNo.105182,13.

[80] RUIZ-BALET D, ZUAZUA E. Control of neural transport for normalising flows[J]. J. Math. Pures Appl. (9), 2024, 181: 58-90.

[81] KLUSOWSKI J M, BARRON A R. Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls[J]. IEEE Trans. Inform. Theory, 2018, 64(12): 7649-7656.

[82] ADAMS R A. Pure and applied mathematics: sobolev spaces[M]. Academic Press [Harcourt Brace Jovanovich, Publishers], New York-London, 1975: xviii+268.

[83] CIARLET P G. Classics in applied mathematics: the finite element method for elliptic problems[M]. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002: xxviii+530.

[84] VELDMAN D W M, BORKOWSKI A, ZUAZUA E. Stability and convergence of a randomized model predictive control strategy[J]. IEEE Trans. Automat. Control, 2024, 69(9): 6253-6260.

[85] PAPAMAKARIOS G, NALISNICK E, REZENDE D J, MOHAMED S, LAKSHMINARAYANAN B. Normalizing flows for probabilistic modeling and inference[J]. J. Mach. Learn. Res., 2021, 22: PaperNo.57,64.

[86] REZENDE D, MOHAMED S. Variational inference with normalizing flows[C]. Proceedings of the 32nd International Conference on Machine Learning. PMLR, 2015: 1530-1538.

[87] VILLANI C. Grundlehren der mathematischen wissenschaften [Fundamental principles of mathematical Sciences]: optimal transport[M]. Springer-Verlag, Berlin, 2009: xxii+973.

[88] MAO T, SIEGEL J W, XU J. Approximation rates for shallow ReLU^k neural networks on sobolev spaces via the radon transform[A/OL]. arXiv (2024). https://arxiv.org/abs/2408.10996.

[89] TAO T. CBMS regional conference series in mathematics: nonlinear dispersive equations [M]. Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 2006: xvi+373.

[90] BACH F. Breaking the curse of dimensionality with convex neutral networks[J]. J. Mach. Learn. Res., 2017, 18: PaperNo.19,53.

[91] MATOUŠEK J. Improved upper bounds for approximation by zonotopes[J]. Acta Math., 1996, 177(1): 55-73.

[92] PISIER G. Remarques sur un résultat non publié de B. Maurey[M]. Seminar on Functional Analysis, 1980–1981. École Polytech., Palaiseau, 1981: Exp.No.V,13.

[93] LI Z, MA C, WU L. Complexity measures for neural networks with general activation functions using path-based norms[A/OL]. arXiv (2020). https://arxiv.org/abs/2009.06132.

[94] AMBROSIO L, CRIPPA G. Continuity equations and ODE flows with non-smooth velocity [J]. Proc. Roy. Soc. Edinburgh Sect. A, 2014, 144(6): 1191-1244.

[95] LIU K, ZUAZUA E. Representation and regression problems in neural networks: relaxation, generalization, and numerics[J]. Math. Models Methods Appl. Sci., 2025, 35(6): 1471-1521.

[96] LUENBERGER D G. Optimization by vector space methods[M]. John Wiley & Sons, Inc., New York-London-Sydney, 1969: xvii+326.

[97] DOSWELL C A. A kinematic analysis of frontogenesis associated with a nondivergent vortex [J]. Journal of Atmospheric Sciences, 1984, 41(7): 1242-1248.

[98] MORALES-HERNÁNDEZ M, ZUAZUA E. Adjoint computational methods for 2D inverse design of linear transport equations on unstructured grids[J]. Comput. Appl. Math., 2019, 38 (4): PaperNo.168,25.

[99] KOVACHKI N B, LANTHALER S, STUART A M. Operator learning: algorithms and analysis[M]. MISHRA S, TOWNSEND A. Handbook of Numerical Analysis: Vol. 25 Numerical Analysis Meets Machine Learning. Elsevier, 2024: 419-467.

[100] ONG Y Z, SHEN Z, YANG H. Integral autoencoder network for discretization-invariant learning[J]. J. Mach. Learn. Res., 2022, 23: PaperNo.286,45.

[101] CHOI J, YUN T, KIM N, HONG Y. Spectral operator learning for parametric PDEs without data reliance[J]. Comput. Methods Appl. Mech. Engrg., 2024, 420: PaperNo.116678,24.

[102] HUA N, LU W. Basis operator network: a neural network-based model for learning nonlinear operators via neural basis[J]. Neural Networks, 2023, 164: 21-37.

[103] PRASTHOFER M, RYCK T D, MISHRA S. Variable-input deep operator networks[A/OL]. arXiv (2022). https://arxiv.org/abs/2205.11404.

[104] ZHANG Z, LEUNG W T, SCHAEFFER H. BelNet: basis enhanced learning, a mesh-free neural operator[J]. Proc. A., 2023, 479(2276): PaperNo.20230043,20.

[105] BOICHUK A A, SAMOILENKO A M. Generalized inverse operators and Fredholm boundary-value problems[M]. VSP, Utrecht, 2004: xiv+317.

[106] KRYLOV N V. Graduate studies in mathematics: lectures on elliptic and parabolic equations in Hölder spaces[M]. American Mathematical Society, Providence, RI, 1996: xii+164.

[107] KRYLOV N V. Graduate studies in mathematics: lectures on elliptic and parabolic equations in Sobolev spaces[M]. American Mathematical Society, Providence, RI, 2008: xviii+357.

[108] BARNARD E, WESSELS L F A. Extrapolation and interpolation in neural network classifiers[J]. IEEE Control Syst. Mag., 1992, 12: 50-53.

[109] XU K, ZHANG M, LI J, DU S, KAWARABAYASHI K, JEGELKA S. How neural networks extrapolate: from feedforward to graph neural networks[C]. International Conference on Learning Representations. 2021.

[110] ZHU M, ZHANG H, JIAO A, KARNIADAKIS G E, LU L. Reliable extrapolation of deep neural operators informed by physics or sparse observations[J]. Comput. Methods Appl. Mech. Engrg., 2023, 412: PaperNo.116064,36.

[111] LANTHALER S, MISHRA S, KARNIADAKIS G E. Error estimates for DeepONets: a deep learning framework in infinite dimensions[J]. Trans. Math. Appl., 2022, 6(1): tnac001,141.

[112] GONON L, GRIGORYEVA L, ORTEGA J P. Approximation bounds for random neural networks and reservoir systems[J]. Ann. Appl. Probab., 2023, 33(1): 28-69.

[113] GRIGORYEVA L, ORTEGA J P. Echo state networks are universal[J]. Neural Networks, 2018, 108: 495-508.

[114] BERRY T, DAS S. Learning theory for dynamical systems[J]. SIAM J. Appl. Dyn. Syst., 2023, 22(3): 2082-2122.

[115] GRIGORYEVA L, LOUW J, ORTEGA J P. Forecasting causal dynamics with universal reservoirs[J]. Nonlinearity, 2025, 38(5): PaperNo.055005.

[116] RUTHOTTO L, HABER E. Deep neural networks motivated by partial differential equations [J]. J. Math. Imaging Vision, 2020, 62(3): 352-364.

攻读博士期间已完成的论文目录

[1] **Ziqian Li**; Enrique Zuazua. *Hamiltonian Interface Dynamics for Reduced-Order Optimization of Incompressible Mixing*. arXiv:2605.04688 (2026).

[2] **Ziqian Li**; Kang Liu; Yongcun Song; Hangrui Yue; Enrique Zuazua. *Deep Neural ODE Operator Networks for PDEs*. Mathematical Models and Methods in Applied Sciences (SCI 一区影响因子 3.0) 36 (2026), pp. 1739-1782.

[3] **Ziqian Li**; Kang Liu; Lorenzo Liverani; Enrique Zuazua. *Universal Approximation of Dynamical Systems by Semiautonomous Neural ODEs and Applications*. SIAM Journal on Numerical Analysis (SCI 一区影响因子 2.9), 64 (2026), pp. 193-223.

[4] Weiwei Hu; **Ziqian Li**; Yubiao Zhang; Enrique Zuazua. *A Structure-Preserving Numerical Scheme for Optimal Control and Design of Mixing in Incompressible Flows*. arXiv:2601.06294 (2026).

[5] Qingguo Hong; Jiwei Jia; Young Ju Lee; **Ziqian Li**. *Greedy Algorithm for Neural Networks for Indefinite Elliptic Problems*. Journal of Scientific Computing (SCI 一区影响因子 3.3), 104 (2025), no. 106.

[6] **Ziqian Li**; Jiwei Jia; Guidong Liao; Young Ju Lee; Siyu Liu. *Neural network method and multiscale modeling of the COVID-19 epidemic in Korea*. The European Physical Journal Plus (SCI 二区影响因子 2.9), 138 (2023), no. 752.

致谢

值此论文完成之际, 谨向十年来在吉林大学学习, 工作与成长过程中给予我教诲, 支持与帮助的师长, 同学, 亲友, 致以最诚挚的感谢. 自 2016 年进入吉林大学数学学院攻读本科起, 到 2020–2021 年在学院担任科研助理, 再到 2021 年至今完成博士阶段的学习与研究, 吉林大学见证了我的求学道路, 也承载了我最重要的成长记忆. 在这里, 我有幸遇见诸位良师益友, 并在他们的陪伴与帮助下不断成长.

首先, 我要衷心感谢我的导师汤涛教授. 本科期间, 我便系统研读了汤老师撰写的谱方法专著, 受益匪浅. 博士阶段有幸师从汤老师, 在接受科研训练的同时, 全程参与了汤老师组织的数值分析线上编程平台的开发与实践工作, 并在汤老师的鼓励下, 先后五次担任吉林大学数学学院数值分析课程助教. 期间, 无论在理论理解, 教学表达, 还是程序实现方面, 我都获得了切实而深刻的提升. 汤老师严谨的学术态度, 开阔的学术视野和身体力行的育人精神, 使我受益至深.

其次, 我要感谢我的另一位导师张然教授. 2020 年我担任数学学院科研助理期间, 正是张老师鼓励我留在学院继续深造, 才使我得以走上后续的研究道路, 并拥有此后诸多宝贵的学习与实践机会. 六年来, 张老师始终以言传身教引导我前行. 无论是科研训练中的细致要求, 还是教学工作中的执着投入, 以及对学院学科发展所体现出的责任意识与长远视野, 都在潜移默化中深深影响着我. 对我而言, 每一声 “老师”, 都包含着最深的敬重与由衷的钦佩.

我还要感谢我在德国的导师 Enrique Zuazua 教授. 自 2024 年 3 月起, 我在埃尔朗根-纽伦堡大学随 Zuazua 教授学习. 正是老师将我引入神经常微分方程这一研究方向, 才有了本文的主要工作. Zuazua 教授在科研学习中给予我极大帮助, 并督促我系统学习控制论及新型计算方法相关知识, 极大地拓宽了我的学术视野, 也让我对交叉研究有了更深的理解.

我要感谢课题组各位同门一直以来给予我的帮助与陪伴. 感谢翟起龙师兄, 王瑞姝师姐, 王秀丽师姐, 彭辉师兄, 封玥师姐, 以及课题组所有师兄师

姐, 师弟师妹. 与你们相处, 讨论与合作的过程, 使我的求学之路始终充满温暖, 也让我在困惑与压力之中不断获得鼓励和力量.

同时, 我也要感谢一路以来在科研与生活中给予我帮助的各位老师. 感谢吉林大学金鑫老师, 张与彪老师, 张凯老师, 佐治亚大学胡薇薇老师, 勃艮第大学刘康老师, 复旦大学王玥老师, 李运章老师, 南洋理工大学宋永存老师. 诸位老师在不同阶段给予我的指导, 支持与关照, 我都铭记于心.

最重要的是, 我要感谢我的父母. 多年来, 你们始终以最朴素而坚定的方式支持我, 包容我, 成全我. 无论我身处何地, 面对何种压力, 你们始终是我最坚实的依靠. 你们未必总能进入我所研究的世界, 却始终以全部的理解, 耐心与爱, 托举我一步一步走到今天. 对你们的感激, 远非 “感谢” 二字所能尽述.

最后, 谨以本文献给一路陪伴我成长的恩师, 父母与朋友. 正因为曾收获如此深厚而真挚的爱与支持, 我才更懂得以郑重之心表达敬意与感恩. 愿这篇论文也作为我成长道路上的一份答卷, 献给所有成就我, 鼓励我, 陪伴我的人.