参数化偏微分方程的多层卷积神经网络

Cosmas Heiß cosmas.heiss@gmail.com
数学系 柏林工业大学 柏林，
德国

Ingo Gühring guehring@math.tu-berlin.de
柏林工业大学机器学习组，德
国柏林

Martin Eigel martin.eigel@wias-柏林.de
魏尔施特拉斯
研究所，德国柏林

arXiv:2304.00388v2 [cs.LG] 4 Apr 2023

摘要

我们将偏微分方程（PDE）的多层求解器理念与基于神经网络的深度学习相结合，提出了一种高效数值求解高维参数偏微分方程的新方法。深入的理论分析表明，所提出的架构能够以任意精度逼近多重网格 V 循环，且权重数量仅随最细网格分辨率呈对数增长。因此，可以推导出神经网络求解参数化偏微分方程的逼近界，且这些界与（随机）参数维度无关。

我们通过高维参数化线性椭圆偏微分方程展示了所提出方法的性能；这类方程是不确定性量化中的常见基准问题。与最先进水平的基于深度学习的求解器相比，我们的方法取得了显著改进。我们还考察了特别具有挑战性的实例：具有高维非仿射高斯场的随机电导率问题（参数维度为 100），以及随机饼干问题。得益于该方法的多层结构，在更细的层级上可以减少训练样本数量，从而显著缩短训练数据的生成时间和方法的训练时间。

关键词： 深度学习、偏微分方程、参数化偏微分方程、多层、表达能力、卷积神经网络、不确定性量化

1 引言

深度学习（DL）在自然科学诸多问题中的应用，已成为最有前景的新兴研究领域之一，有时也称为科学机器学习 （Baldi，2018；Sadowski 和 Baldi，2018；Keith 等人，2021；Noé 等人，2020；Lu 等人，2021）。在这项工作中，我们聚焦于参数化偏微分方程（PDEs）这一具有挑战性的问题，这类方程用于描述复杂物理系统，例如工程应用和自然科学中的系统。参数决定数据或定义域，并用于将不确定性引入模型，例如依据假设或测得的统计性质。由此产生的高维问题，会在不确定性量化（UQ）这一研究领域中进行分析和数值求解（Lord 等人，

2014年）。这类参数化模型在自动化设计、天气预报、风险模拟和材料科学等领域都十分重要，以上仅列举数例。

为使问题设定更加具体，对于可能是可数无限维的参数空间$\Gamma \subseteq \mathbb{R}^{\mathbb{N}}$ 和空间域 $D \subseteq \mathbb{R}^d$，$d \in \{1,2,3\}$，我们关注学习偏微分方程的解映射 $u : \Gamma \times D \to \mathbb{R}$ 参数化定常扩散偏微分方程，它对所有参数 $\mathbf{y} \in \Gamma$均满足以下方程

$$\begin{cases} -\operatorname{div}_x \kappa(\mathbf{y}, x) \nabla_x u(\mathbf{y}, x) &= f(x) \quad \text{on } D, \\ u(\mathbf{y}, x) &= 0 \qquad \text{on } \partial D, \end{cases} \tag{1}$$

其中 $\kappa : \Gamma \times D \to \mathbb{R}$ 是参数化扩散系数，用于描述扩散率，即介质性质。请注意，微分算子是针对物理变量 x作用的。

作为一种直接但计算效率低下的解决方案，可以针对每个$\mathbf{y} \in \Gamma$ 分别对方程 (1) 进行数值求解，例如采用有限元 (FE) 方法，再用蒙特卡罗方法估计统计感兴趣量。更精细的方法，例如随机Galerkin方法（Eigel 等，2014, 2020）、随机配置法（Babuska 等，2007；Nobile 等，2008；Ernst 等，2018）或最小二乘方法（Chkifa 等，2015；Eigel 等，2019a；Cohen 和 Migliorati，2023），则利用解流形$\{u(\mathbf{y}) : \mathbf{y} \in \Gamma\}$ 的光滑性和低维性，通过投影或插值有限个离散解$\{u_h(\mathbf{y}_n)\}_{n=1}^N$来实现。对方程 (1) 在样本点 $\mathbf{y}_n$ 处进行数值求解，并执行投影或插值方案后，只需评估构造出的函数“代理模型”，即可快速近似求得任意$\mathbf{y} \in \Gamma$ 对应的解。然而，尽管低秩近似或约化基方法等压缩技术有助于利用低维结构，但当参数维度较大时，这些方法很快仍会变得不切实际。多层方法是另一种适用于此类问题的方法，例如多层随机配置法（MLSC）（Teckentrup 等，2015）和多层求积（Harbrecht 等，2016；Ballani 等，2016）均已展示了其优势。其核心思想是平衡物理近似与统计近似的误差分量。这可以通过在物理域中对解进行频率分解来实现，该分解基于由适当空间离散化确定的嵌套有限元空间层次。最粗层级上的初始近似逐步通过更细层级上的加性修正加以细化。更细层级上的修正项通常包含迅速减少的信息量，可据此采用随层级变化的随机离散化网格尺度。

对于（通常是非参数化）偏微分方程的数值求解，研究者已设计出大量神经网络架构（例如，参见 Beck 等人，2023；Berner 等人，2020a；Raissi 等人，2019；Sirignano 和 Spiliopoulos，2018；E 等人，2017；Ramabathiran 和 Ramachandran，2021；Kharazmi 等人，2019；Yu 等人，2018；Han 等人，2017；Li 等人，2021b）。与经典方法类似，针对参数化偏微分方程的基于深度学习的求解器通常以监督方式，从计算得到（或观测到）的样本中学习映射 $\mathbf{y} \mapsto u_h(\mathbf{y})$：$\{u^h(\mathbf{y}_n)\}_{n=1}^N$ （例如，参见 Geist 等人，2021；Hesthaven 和 Ubbiali，2018；Anandkumar 等人，2020）。这会导致计算开销很大的数据生成过程：先近似计算$\mathbf{y}_n$ 处的解，随后在这些数据上训练神经网络。训练完成后，只需进行一次开销很小的前向传播，即可近似求解式（1）在任意$\mathbf{y}$处的解。我们注意到

与许多经典机器学习问题不同，数据集可以即时生成，而且原则上可以达到任意精度。

除了数值结果之外，人们在理解神经网络表示偏微分方程解的逼近能力（也称为表达能力）方面已取得重大进展（Kutyniok 等人，2022；Schwab 和 Zech，2019；Berner 等人，2020b；Mishra 和 Molinaro，2022；Grohs 等人，2018；Gühring 等人，2020；Gühring and Raslan，2021）。这些研究表明，神经网络的表达能力至少不逊于前文提及的“经典”顶尖方法。然而，数值结果往往并未体现出这一理论优势（Adcock 和 Dexter，2021）。相反，训练过程中误差通常很早便停止收敛，无法达到经典方法的精度（Geist 等人，2021；Kharazmi 等人，2019；Lu 等人，2021；Grossmann 等人，2023）。

我们提出了一种基于深度学习的神经网络方法，通过借鉴 MLSC 和多重网格算法中的概念，克服其他神经网络架构在求解参数化偏微分方程时的性能不足。子网络以监督方式学习粗网格逼近以及更细层级上的逐次校正，并将它们组合起来得到最终预测。与多层蒙特卡洛（MLMC）和 MLSC 方法类似，我们证明，随机参数空间的采样精细度（以每个层级的训练数据量衡量）可以与空间离散化的精细度相互平衡。所开发的架构由多个 U-Net（Ronneberger 等，2015）组合而成，它们本质上与多分辨率方法密切相关。

下面我们概述本文的主要贡献。

- 理论：我们对用于求解参数化偏微分方程的类 U-Net 卷积神经网络（卷积神经网络）的表达能力进行了深入分析。结果表明，在一致椭圆性假设下，随着精度提高，所需参数的数量与参数维度无关。具体而言，我们证明，常见的多层卷积神经网络架构能够以少于 $\log(1/h)$ 个权重，以任意精度逼近采用 Richardson 平滑器的 V 循环多重网格求解器迭代（Richardson 和 Glazebrook，1911年），其中 h 是底层有限元空间的网格尺寸（定理 6）。利用这一结果，我们用一个具有 $\log(1/h)^2$ 个权重的神经网络，将式 (1) 的解映射逼近至期望 H^1-范数误差 h。

- 数值结果：我们针对不确定性量化中参数化偏微分方程的常见基准问题，对我们的方法以及更广泛的基于 U-Net 的方法进行了深入的数值研究。值得注意的是，测试的方法在求解参数化偏微分方程时，均显著优于最先进的神经网络架构。为了降低训练数据生成和神经网络训练过程中的计算复杂度，我们将大部分训练数据转移到较粗层级，从而使我们的方法能够扩展到更细网格，最终提高精度。

1.1 相关工作

Schwab 和 Zech（2019）；Grohs 和 Herrmann（2022）基于不确定性量化（UQ）中已知的解的稀疏广义多项式混沌表示（Schwab 和 Gittelson，2011；Cohen 和 DeVore，2015），以及用神经网络逼近多项式的一项核心结果（Yarotsky，推导出了关于逼近参数化偏微分方程解的表达能力结果，

2017）。Kutyniok 等人（2022）将约化基方法转化为神经网络架构。这些结果均未给出与（随机）参数维数无关的界。相比之下，将 Kutyniok 等人（2022）的结果转化到我们的设定中（并加以简化），可得渐近上界： $h^{-2}\log(h^{-1})^p + p\log(h^{-1})^{3p}$ 个权重。数值实验要么未予提供，要么不支持积极的理论结果（Geist 等人，2021）。关于神经网络表达能力结果的总体概述，感兴趣的读者可参阅（Gühring 等人，2022）。

与我们的方法相关且对其推导具有启发意义的是 (Teckentrup 等，2015) 以及 (Harbrecht 等，2016；Ballani 等，2016) 中介绍的多层随机配置方法。另一项相关工作是 (Lye 等，2021)，其中介绍了一种具有多层结构的训练过程。在对优化所达到的精度作出某些假设的前提下，研究进行了误差分析，其中部分内容也可以迁移到我们的架构中。

微分算子与卷积层之间的联系众所周知（见 Chen 等，2022），卷积神经网络已被用于近似参数化偏微分方程，以解决诸如预测给定几何形状周围气流等问题（Bhatnagar 等，2019；Guo 等，2016）。在 (Zhu and Zabaras, 2018) 中，研究人员使用贝叶斯卷积神经网络评估由随机偏微分方程控制的过程中的不确定性。(Ummenhofer et al., 2020) 将这一概念扩展到无网格拉格朗日方法中的连续卷积，以解决流体动力学问题。多层分解的概念也已应用于卷积神经网络，以近似偏微分方程的解；此外，(Gao et al., 2018) 将其用于通用生成式建模，(Fan et al., 2019a,b) 则基于层次矩阵表示应用了这一概念。

这种层次矩阵表示甚至已扩展到图卷积神经网络，并作为多极核，用于无网格近似参数化偏微分方程的解（Li 等人，2020）。这种方法与我们的方法相似。然而，尚未探讨其与经典多重网格算法之间的数学联系。这些卷积核基于快速多极建模，与有限元方法没有直接联系。

He 和 Xu（2019）提出了一种称为 MgNet 的多层方法，其中经典多重网格求解器的各个部分被神经网络取代，从而形成类似 U-Net 的神经网络，并将其用于图像分类。Meta-MgNet（Chen 等，2022）是一种元学习方法，它使用 MgNet 预测参数化偏微分方程的解，与我们的方法最为相关。预测解会逐步细化，并通过在每一层使用额外的小型神经网络计算参数化离散算子的编码，将随机参数纳入其中。这些元网络的输出随后会影响 MgNet 架构中的平滑操作。然而，与我们的方法不同，Meta-MgNet 仍依赖于组装后的算子矩阵。此外，尽管采用迭代方法可以获得更准确的近似解，但单次应用卷积神经网络所带来的速度提升因此受到削弱，作者报告的运行时间与经典求解器处于同一量级。最后，作者使用两个或三个自由度而非连续扩散率场，在具有均匀扩散率的问题上测试了该方法。我们的方法与 Meta-MgNet 的区别如下：（i）我们表明，通用 U-Net 已能够近似参数化微分算子，无需采用元网络方法；（ii）我们通过纳入多层优化方案改进了训练流程；（iii）我们提供了理论复杂度分析；（iv）我们通过实验表明，对于多种随机参数场，单次应用我们的方法即可达到高精度。

1.2 概述

第一节介绍参数化达西模型问题及其有限元离散化。第 3 节简要介绍多重网格方法。第 4 节描述我们基于深度学习的求解方法。第5节探讨所述架构的表达能力的理论分析。第 6 节介绍我们针对若干参数化偏微分方程问题开展的数值研究。第 7 节讨论并总结我们的发现。实现细节和证明见附录。

2 问题设定

不确定性量化中的一个标准模型问题是定常线性扩散方程 (1)，其中扩散系数 $\kappa : \Gamma \to L^\infty(D)$ 是一个方差有限的随机场，由一个可能具有高维度的随机参数 $\mathbf{y} \in \Gamma \subseteq \mathbb{R}^{\mathbb{N}}$ 决定，因此解也由该参数确定。参数是服从某种乘积测度 $\pi = \otimes_{k\geq 1}\pi_1$ 的随机向量的像，其中 π_1 通常是 $\Gamma_1 \subseteq \mathbb{R}$ 上的均匀分布（记为 U）或标准正态测度（记为 N）。对于每个参数实现$\mathbf{y} \in \Gamma$，该问题的变分形式如下：给定有界利普希茨区域 $D \subseteq \mathbb{R}^d$ 和源项 $f \in V^* = H^{-1}(D)$，求 $u \in V := H_0^1(D)$，使得对所有试验函数 $w \in V$，

$$\int_D \kappa(\mathbf{y}, x)\nabla u(x) \cdot \nabla w(x)\,\mathrm{d}x = \int_D f(x)w(x)\,\mathrm{d}x. \tag{2}$$

我们假设上述方程始终存在唯一解。事实上，即使对于系数无界（服从对数正态分布）的情形，这一点也以很高的概率成立；此时双线性形式并不满足一致椭圆性。由此得到解算子 v，它将参数的一次实现$\mathbf{y} \in \Gamma$ 映射为对应的解 $v(\mathbf{y})$：

$$v\colon \Gamma \to V, \qquad \mathbf{y} \mapsto v(\mathbf{y}).$$

在这项工作中，我们致力于开发一种高效且准确的神经网络近似 $\tilde{v}\colon \Gamma \to V$，使期望 H^1-距离最小

$$\int_\Gamma \|\nabla(v(\mathbf{y}) - \tilde{v}(\mathbf{y}))\|^2_{L^2(D)}\,\mathrm{d}\pi(\mathbf{y}).$$

在假设系数 κ以很高概率有界的情况下，这等价于针对标准的参数依赖范数进行优化。

我们回顾随机参数维数为 p的截断卡亨南–洛埃夫展开 （KL 展开）（Knio 和 Le Maitre，2006年；Lord 等，2014年），这是一种常见的随机场参数化表示。其形式为

$$\kappa(\mathbf{y}, x) = a_0(x) + \sum_{k=1}^{p} \mathbf{y}_k a_k(x). \tag{3}$$

此处，$p \in \mathbb{N}$ 基函数 $a_k \in L^2(D)$ $1 \leq k \leq p$ 由假定的场的统计性质 κ决定。当考虑高斯随机场时，它们

可由其前两个矩完全表征，而 a_k 是协方差积分算子的特征函数；参数向量 $\mathbf{y}$ 则是标准正态随机向量的像。为了使计算可行，这些函数的范数必须充分衰减，从而保证截断是合理的。请注意，在数值实验中，我们使用了一种广为人知的“人工”KL 展开。

2.1 离散化

为了对无穷维变分形式 (2) 进行数值求解，必须定义一个有限维子空间，并在其中计算解的近似值。与偏微分方程中的常见做法一样，我们采用有限元方法，该方法基于将计算域 D 不相交地分解为单形（也称为单元或网格单元）。令区域为 $D = [0,1]^2$，我们假设将其细分为全等三角形，使得 D 恰好由均匀正方形网格表示，并且每个节点处的连接关系均相同（另见注 11）。离散空间 V_h 随后由协调 P1 有限元帽函数 φ_j 张成：$\mathbb{R}^2 \to \mathbb{R}$，即 $V_h =$ 张成$\{\varphi_j\}_{j=1}^{\dim V_h} \subseteq V$（例如参见 Braess，2007年；Ciarlet，2002年）。对于方程 (2) 的离散化版本 $u, w \in V$，我们将其记为$u_h, w_h \in V_h$，其有限元系数向量为 $\mathbf{u}, \mathbf{w} \in \mathbb{R}^{\dim V_h}$，并将 κ的节点插值记为 κ_h，使得对于每个 $\mathbf{y} \in \Gamma$，$\kappa_h(\mathbf{y}, \cdot) \in V_h$的系数向量为 $\boldsymbol{\kappa}_{\mathbf{y}} \in \mathbb{R}^{\dim V_h}$。因此，

$$u_h = \sum_{i=1}^{\dim V_h} \mathbf{u}_i \varphi_i, \quad w_h = \sum_{i=1}^{\dim V_h} \mathbf{w}_i \varphi_i, \quad \kappa_h(\mathbf{y}, \cdot) = \sum_{i=1}^{\dim V_h} (\boldsymbol{\kappa}_{\mathbf{y}})_i \varphi_i.$$

参数化 Darcy 问题 (2) 的离散形式如下：

问题 2.1（参数化 Darcy 问题，离散形式）对于 $\mathbf{y} \in \Gamma$，求 $u_h \in V_h$使得对所有 $w_h \in V_h$

$$a_{\mathbf{y},h}(u_h, v_h) = f(w_h),$$

其中 $a_{\mathbf{y},\,h}(u_h, v_h) := \int_D \kappa_h(\mathbf{y}\ x) \nabla u_h(x) \cdot \nabla w_h(x) \mathrm{d}x$。使用有限元系数向量可得线性方程组

$$A_{\mathbf{y}} \mathbf{u} = \mathbf{f}, \tag{4}$$

其中 $\mathbf{f} := (f(\varphi_i))_{i=1}^{\dim V_h} = (\int_D f(x)\varphi_i(x)\,\mathrm{d}x)_{i=1}^{\dim V_h}$且 $A_{\mathbf{y}} := (a_{\mathbf{y},\,h}(\varphi_j, \varphi_i))_{i,\,j=1}^{\dim V_h}$。

我们假设该问题始终存在唯一解，并将离散参数到解的算子定义为

$$v_h \colon \Gamma \to V_h, \qquad \mathbf{y} \mapsto v_h(\mathbf{y}), \tag{5}$$

其中 $v_h(\mathbf{y})$ 对任意 $\mathbf{y} \in \Gamma$ 都是问题 2.1 的解。此外，$\mathbf{v}_h \colon \Gamma \to \mathbb{R}^{\dim V_h}$ 将 $\mathbf{y}$ 映射为 $v_h(\mathbf{y})$ 的有限元系数向量。

备注 1 在我们的分析中，我们忽略扩散率场 κ 作为 κ_h的近似所带来的影响，因为这只会引入不必要的技术细节。有限元分析领域早已充分阐明了理论上应如何处理这一问题（例如，参见 *Ciarlet，2002*年，第 *4* 章）。值得注意的是，我们始终假设系数是在所分析并用于实验评估的多重网格算法的最细层上离散的。

3 多重网格方法简介

我们提出的方法从根本上依赖于代数方程的多重网格求解器这一概念。这些迭代求解器具有最优复杂度，即其收敛速度相对于自由度呈线性。它们的数学性质已得到充分理解，并且常用于求解离散化的偏微分方程，尤其是采用有限元和网格层次结构进行离散化的情形（参见 Braess，2007年；Yserentant，1993年；Hackbusch，2013年）。本节将简要介绍多重网格设置。

起点是按层级 $\ell \in \{1, \dots, L\}$ 编号的嵌套空间层次结构，最细层为 L，它决定了近似的精度。在我们的设定中，最细层预先选定，并假定其可提供足够的有限元精度。与上一节类似，我们将 V_ℓ 定义为关联的二分拟均匀三角剖分 $\mathcal{T}_\ell$ 上相容 P1 元所构成的有限元空间，其网格尺寸为 $h_\ell \sim 2^{-\ell}$，自由度为 $\dim V_\ell \sim 2^{\ell d}$。由此得到一列嵌套的有限元空间

$$V_1 \subseteq \ldots \subseteq V_L \subseteq H_0^1(D).$$

这些空间由相应层级上的有限元基函数张成，即 $V_\ell = \text{span}\{\varphi_j^{(\ell)}\}_{j=1}^{\dim V_\ell}$。通过在最粗网格 $\mathcal{T}_1$ 上近似目标函数 $u \in H_0^1(D)$ 的低频部分，并随后在更细网格上计算较高频率的修正项，可得到该目标函数的频率分解。

对于每一层 $\ell = 1, \ldots\ldots, L$，我们按照式 (5) 定义离散解算子 $v_\ell \colon \Gamma \to V_\ell$。相对于下一较粗层 $\ell - 1$，$\ell \in \{2, \ldots\ldots, L\}$ 的修正项 $\hat{v}_\ell$ 为

$$\hat{v}_\ell \colon \Gamma \to V_\ell, \qquad \mathbf{y} \mapsto v_\ell(\mathbf{y}) - v_{\ell-1}(\mathbf{y}). \tag{6}$$

此外，我们定义 $\mathbf{v}_\ell$ 和 $\hat{\mathbf{v}}_\ell$ 为函数，它们将参数 $\mathbf{y}$ 映射到 $v_\ell(\mathbf{y})$ 和 $\hat{v}_\ell(\mathbf{y})$ 在 V_ℓ 中对应的系数向量。如果所考虑偏微分方程的解足够光滑，则修正项会随层级增加而指数衰减，即 $\|\hat{v}_\ell(\mathbf{y})\|_{H^1} \lesssim 2^{-\ell} \|f\|_*$（见 Braess，2007年；Yserentant，1993；Hackbusch，2013）。

多重网格方法的核心是将离散函数在空间层次结构的相邻层之间传递。这通过延拓算子和限制算子实现，其离散形式将在下文定义。

定义 2（延拓矩阵与加权限制矩阵） 令 $L \in \mathbb{N}$，并对某个 $\ell \in \{1, \ldots, L-1\}$，空间 V_ℓ 和 $V_{\ell+1}$ 如上所定义。延拓矩阵 $P_\ell \in \mathbb{R}^{\dim V_{\ell+1} \times \dim V_\ell}$ 是在相应有限元基函数下，将 V_ℓ 嵌入 $V_{\ell+1}$ 的典范嵌入的矩阵表示。$P_\ell^T \in \mathbb{R}^{\dim V_\ell \times \dim V_{\ell+1}}$ 定义了加权限制。

我们多层神经网络架构的理论分析依赖于算法 3.1 所示的多重网格 V 循环。它利用了这样一个观察：简单的迭代求解器能够降低残差中的高频分量。除这些所谓的平滑迭代外，还会在更粗网格上递归计算低频校正。事实上，如果递归调用粗网格校正所产生的近似误差足够小，就能实现与网格无关的压缩率（参见 Braess 和 Hackbusch，1983年；Braess，2007年）。

在分析中，我们使用阻尼 Richardson 平滑器（Richardson 和 Glazebrook，1911 年）。由于收敛率不佳，它在实践中很少使用，但因其简单而便于理论分析。对于合适的阻尼参数 $\omega > 0$，用于求解方程（4）的 Richardson 迭代为

$$\begin{aligned} \mathbf{u}^0 &:= \mathbf{0}, \\ \mathbf{u}^{n+1} &:= \mathbf{u}^n + \omega(\mathbf{f} - A_{\mathbf{y}}\mathbf{u}^n). \end{aligned} \tag{7}$$

我们将 $\mathrm{MG}_{k_0,k}^m\colon \mathbb{R}^{3\times\dim V_h} \to \mathbb{R}^{\dim V_h}$ 记为一个函数，它将初始猜测、扩散率场和右端项映射为执行 m 次多重网格 V 循环（算法 3.1）后的结果，其中最粗层级进行 k_0 次平滑迭代，较细层级进行 k 次平滑迭代。

基于这一迭代，我们结合（Braess 和 Hackbusch，1983年，定理 4.2）中关于多重网格算法的基本收敛结果以及一致椭圆性假设，得到 V 循环多重网格算法的一致压缩性质。对于$\mathbf{x} \in \mathbb{R}^{\dim V_h}$和离散算子 $A \in \mathbb{R}^{\dim V_h \times_{\dim} V_h}$，我们采用能量范数的标准记号 $\|\mathbf{x}\|_A = \|A^{1/2}\mathbf{x}\|_{\ell^2}$。

定理 3 设 $k \in \mathbb{N}$。存在 $\omega > 0$ 和与网格无关的 $k_0 \in \mathbb{N}$、 $C < 0$，使得对问题 *2.1* 应用带有 k 次前平滑和后平滑迭代、最粗网格上 k_0 次迭代以及 *Richardson* 阻尼参数 ω 的 *V* 循环迭代，可使残差$\mathbf{e}_{\mathbf{y}}^i := \mathrm{MG}_{k_0,k}^i(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_h(\mathbf{y})$ 在第 i 次迭代时实现与网格无关的压缩，即

$$\|\mathbf{e}_{\mathbf{y}}^{i+1}\|_{A_{\mathbf{y}}} \leq \mu(k)\|\mathbf{e}_{\mathbf{y}}^i\|_{A_{\mathbf{y}}}, \quad \mu(k) \leq \frac{C}{C+2k} < 1,$$

对所有 $\mathbf{y} \in \Gamma$。

在下一个注 记中，我们详细说明最粗网格上的平滑步骤数 k_0。

注记 4 在 *V*循环算法中，通常假设在最粗层级上求解的方程组是精确求解的（使用直接求解器）。然而，*(Braess* 和 *Hackbusch, 1983*年*,* 定理 *4.2)* 的证明表明，这并非必要。事实上，只要最粗层级上的残差压缩小于 $C/(C+2k)$ 即可。这表明， k_0 与最精细网格的网格尺度无关。严谨的推导请参见 *(Xu* 和 *Zhang, 2022,* 定理 *4.4)*。在我们的设定中，也可以使用阻尼 *Richardson* 迭代在最粗网格上进行校正，从而有效地执行额外的平滑步骤。

4 多层神经网络方法论

本节概述我们提出的多层神经网络架构，该架构旨在高效地数值求解参数化偏微分方程。有些设计选择专门针对定常扩散方程 (1)，但也很容易调整以适用于其他情形，尤其是不同的线性偏微分方程，想必也包括非线性偏微分方程。此外，在分析和实验中，我们将空间维数固定为二。我们再次指出，这些

算法 3.1 `V-cycle`函数

输入： $\mathbf{u}$，$\mathbf{f}$，A，ℓ
对于 k 预平滑步骤**执行**
    $\mathbf{u} \leftarrow \mathbf{u} + \omega(\mathbf{f} - A\mathbf{u})$ // 执行平滑步骤
结束循环
如果 $\ell = 1$ **则**
    求解 $A\mathbf{u} = \mathbf{f}$ 以求 $\mathbf{u}$ // 最粗网格步骤
否则
    $P \leftarrow P_{\ell-1}$
    $\mathbf{r} \leftarrow P^{\intercal}(\mathbf{f} - A\mathbf{u})$ // 计算受限残差
    $\overline{A} \leftarrow P^{\intercal}AP$ // 计算受限算子
    $\overline{\mathbf{e}} \leftarrow \mathbf{0}$
    $\overline{\mathbf{e}} \leftarrow$ `V-cycle`$(\overline{\mathbf{e}}, \overline{\mathbf{r}}, \overline{A}, \ell - 1)$ // 求解粗网格校正 $\overline{\mathbf{e}}$
    $\mathbf{u} \leftarrow \mathbf{u} + P\overline{\mathbf{e}}$ // 添加粗网格校正
结束 if
对于 m 后平滑步骤**执行**
    $\mathbf{u} \leftarrow \mathbf{u} + \omega(\mathbf{f} - A\mathbf{u})$ // 执行平滑步骤
结束 for
返回 u

结果可以通过简单的方式推广到更多维度，例如使用更高维的卷积。

所提出方法的一般流程如下。首先，针对随机抽取的参数（数据点），即产生相应有限元解的系数场实现，使用经典有限元求解器生成解数据集。然后，按照式（6）的形式化定义，为每个数据点生成解的多层分解。训练一个神经网络，使其根据输入参数输出粗网格解以及后续各层的修正项；这些输入参数是在相应有限元空间上表示系数场实现的系数向量。将这些贡献相加，即可得到网络预测。最后，为了检验网络的准确性，生成一组独立的有限元解，并计算神经网络预测值的误差指标。

我们方法的核心称为 ML-Net（即多层网络的简称）。对于给定的层数 $L \in \mathbb{N}$，它由 L 个联合训练的子网络 ML-Net$_\ell$组成。每个子网络都经过优化，将其输入——离散化的参数依赖扩散系数$\kappa_{\mathbf{y}}$ 以及前一层的输出$\mathbf{z}^{(\ell-1)}$ ——映射为第 ℓ 层的校正项 $\hat{}\mathbf{v}_\ell(\mathbf{y})$，如式（6）所示。ML-Net$_\ell$ 子网络架构的灵感来自多重网格 V 循环的级联，其通过一系列 U-Net 实现（Ronneberger 等，2015）。以序列方式使用 U-Net，逐步校正前序模块的输出，是计算机视觉中一种成熟的策略（例如参见 Genzel 等，2022）。下一节将为这一设计选择提供理论依据。由于我们假设右端项 $f \in H^{-1}(D)$ 与$\mathbf{y}$无关，因此不将其作为额外输入。

更确切地说，

$$\text{ML-Net}[\boldsymbol{\theta}; L] : \mathbb{R}^{\dim V_L} \to \bigotimes_{\ell=1}^{L} \mathbb{R}^{\dim V_\ell}, \quad \boldsymbol{\kappa} \mapsto (\mathbf{z}^{(\ell)})_{\ell=1}^{L},$$

其中 $\mathbf{z}^{(\ell)} \in \mathbb{R}^{\dim V_\ell}$ 是第 ℓ 层子网络的输出，$\boldsymbol{\theta}$ 是可学习参数。对于这些子网络，我们有

$$\begin{aligned}
&\text{ML-Net}_1[\boldsymbol{\theta}_1; L] \colon \mathbb{R}^{\dim V_L} \to \mathbb{R}^{\dim V_1}, \\
&\text{ML-Net}_\ell[\boldsymbol{\theta}_\ell; L] \colon \mathbb{R}^{\dim V_L} \times \mathbb{R}^{\dim V_{\ell-1}} \to \mathbb{R}^{\dim V_\ell}, \quad \text{for } \ell = 2, \dots, L.
\end{aligned}$$

输出为

$$\begin{aligned}
\mathbf{z}^{(1)} &= \text{ML-Net}_1[\boldsymbol{\theta}_1; L](\boldsymbol{\kappa}), \\
\mathbf{z}^{(\ell)} &= \text{ML-Net}_\ell[\boldsymbol{\theta}_\ell; L](\boldsymbol{\kappa}, \mathbf{z}^{(\ell-1)}), \quad \text{for } \ell = 2, \dots, L.
\end{aligned}$$

同样，$\boldsymbol{\theta}_\ell$ 是可学习参数，且 $\boldsymbol{\theta} = (\boldsymbol{\theta}_\ell)_{\ell=1}^{L}$。

每一层由一系列深度为 ℓ、输入维度等于 $\dim V_\ell$ 的 U-Net 组成（排列在二维网格上）。通过将 U-Net 的深度与层级耦合，我们确保每一层的 U-Net 在内部将数据下采样至粗网格分辨率 $\dim V_1$（同样排列在二维网格上）。因此，每个 U-Net 的参数数量随层级同步增加。为使各层的计算量大致相等，我们减少每层的 U-Net 数量，记为 $\boldsymbol{R}_\ell$，其中 $\boldsymbol{R} \in \mathbb{N}^L$。第 ℓ 层的神经网络架构为

$$\text{ML-Net}_\ell[\boldsymbol{\theta}_\ell; L](\boldsymbol{\kappa}, \boldsymbol{z}) := \left[\bigcirc_{r=1}^{\boldsymbol{R}_\ell} \text{UNet}[\boldsymbol{\theta}_{\ell,r}; \ell]([\downarrow \boldsymbol{\kappa}, \cdot])\right](\uparrow \boldsymbol{z}),$$

其中，$\downarrow$ ($\uparrow$) 表示至第 ℓ 层分辨率 $\dim V_\ell$ 的上采样（下采样）算子。这些算子的设计借鉴了定义 2 中的延拓算子和限制算子。不过，它们由可训练的步幅卷积（或转置步幅卷积）构成，而非显式构造。第 ℓ 层子网络 $\boldsymbol{\theta}_\ell$ 的参数包括可学习的上采样和下采样参数，以及 U-Net 序列 $(\boldsymbol{\theta}_{\ell,\,r})_{r=1}^{\boldsymbol{R}_\ell}$。

我们使用 ML-Net 处理参数化偏微分方程的完整方法由以下多步流程组成：

步骤 1：采样并计算数据。我们用 $N \in \mathbb{N}$ 表示训练数据集的大小，并根据 π 对参数实现进行采样，得到 $\mathbf{y}_1, \dots, \mathbf{y}_N$。对于每个参数实现 $\mathbf{y}_i$，$1 \le i \le N$，计算式 (6) 中各层 $1 \le \ell \le L$ 的网格修正项 $\hat{\mathbf{v}}_\ell(\mathbf{y}_i)$，以及最细层 L 上 $\kappa_L(\cdot, \mathbf{y}_i)$ 的有限元系数向量 $\boldsymbol{\kappa}_{\mathbf{y}_i} \in \mathbb{R}^{\dim V_L}$。请注意，计算网格修正项需要针对每个参数在最精细网格上求解偏微分方程。

步骤 2：训练 ML-Net。对于每个层级 $\ell \in \{1, \dots, L\}$，计算归一化常数 $\delta_\ell^2 := \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{v}}_\ell(\mathbf{y}_i)\|_2^2$ 以及 H^1 质量矩阵 $\boldsymbol{M}_\ell \in \mathbb{R}^{\dim V_\ell \times \text{维度} V_\ell}$，方法是

$$(\boldsymbol{M}_\ell)_{kj} := \left\langle \varphi_k^{(\ell)}, \varphi_j^{(\ell)} \right\rangle_{H^1(D)}, \quad \text{for} \quad k, j \in \{1, \dots, \dim V_\ell\},$$

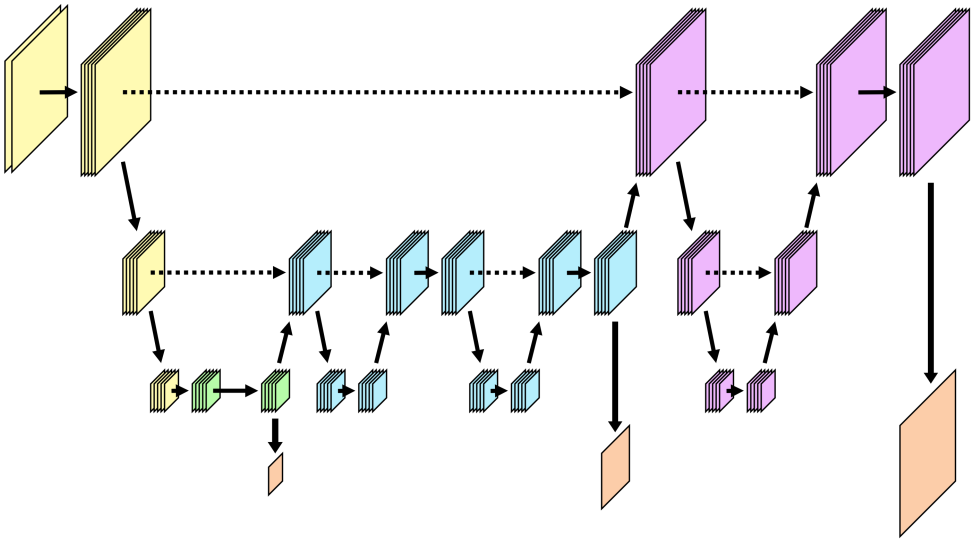


图 1： 所提出的 ML-Net 结构示意图。下采样级联以黄色表示。在此示例中，该网络针对 $L = 3$构建，绿色、青色和洋红色部分分别表示每一层上的各个阶段。输出以橙色表示。实线箭头表示卷积，虚线箭头表示跳跃连接。

其中 $\{\varphi_1^{(\ell)}, \ldots, \varphi^{(\ell)}{}_{\text{维度 } V_\ell}\}$ 构成 V_ℓ 的协调的 P1 有限元基。我们通过近似求解来优化参数 $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} \sum_{\ell=1}^{L} \sum_{i=1}^{N} \mathbf{d}_{\ell i}^T \boldsymbol{M}_\ell \mathbf{d}_{\ell i}, \quad \text{where} \quad \mathbf{d}_{\ell i} \coloneqq \left[\text{ML-Net}[\boldsymbol{\theta}; L](\boldsymbol{\kappa}_{\mathbf{y}_i})\right]_\ell - \frac{1}{\delta_\ell} \hat{\mathbf{v}}_\ell(\mathbf{y}_i)$$

采用小批量随机梯度下降。我们的损失函数计算 H^1距离，其中一项是子网络输出（作为 V_ℓ 的有限元基系数），另一项是直接作用于每一层系数上的归一化网格修正项 ℓ，即，

$$\mathbf{d}_{\ell i}^T \boldsymbol{M}_\ell \mathbf{d}_{\ell i} = \left\| \sum_{j=1}^{\dim V_\ell} \left[\text{ML-Net}[\boldsymbol{\theta}^*; L](\boldsymbol{\kappa}_{\mathbf{y}_i})\right]_{\ell j} \varphi_j^{(\ell)} - \frac{1}{\delta_\ell} \hat{v}_\ell(\mathbf{y}_i) \right\|_{H^1(D)}.$$

通过 δ_ℓ 进行归一化，可确保损失函数中每一层的权重相同。

步骤 3：推理。 将优化后的参数记为 $\boldsymbol{\theta}^*$，则 $\mathbf{y} \in \Gamma$ 的近似解（即神经网络预测）可通过下式计算：

$$v(\mathbf{y}) \approx \sum_{\ell=1}^{L} \delta_\ell \sum_{j=1}^{\dim V_\ell} \left[\text{ML-Net}[\boldsymbol{\theta}^*; L](\boldsymbol{\kappa}_{\mathbf{y}})\right]_{\ell j} \varphi_j^{(\ell)}.$$

在下面的注记中，我们介绍一种更简单的基线方法，它可以视为 ML-Net 的一个特例。

注记 5 为了与所述多层方法进行比较并评估其优势，我们提出一种更简单的神经网络，它将离散化的参数依赖扩散系数 $\kappa_{\mathbf{y}}$ 直接映射为最细层上的系数。为此，我们将由多个 *U-Net* 组成的序列称为 *UNetSeq*。它可以视为 *ML-Net* 的一种变体，其中 $\boldsymbol{R}_\ell = 0$ 用于层级 $\ell = 1, \ldots, L-1$，也就是说，仅使用最高层的子模块。由于 *UNet* 模块内部具有多分辨率结构，这种架构是单层与多层之间的一种混合方法。对于这种架构，我们选择 *U-Net* 的数量，使参数总量大致与 *ML-Net* 相当。

5 理论分析

本节将证明，类 U-Net 神经网络架构能够以任意精度 $\varepsilon > 0$ 逼近 V_h 上的多重网格 V 循环，所需参数数量与 ε 无关，且仅随 $1/h$ 对数增长（定理 6）。我们在推论 8 中利用这一结果，以任意精度 ε 逼近参数化偏微分方程问题 2.1 的离散解 $\mathbf{v}_h\colon \Gamma \to \mathbb{R}^{\dim V_h}$；所用神经网络的参数数量上界为 $\log(1/h)\log(1/\varepsilon)$，且与随机参数维数 p无关。为此，构造一个神经网络，模拟多重网格求解器分别应用于每个$\mathbf{y}$ 的过程，从而近似计算解$\mathbf{v}_h(\mathbf{y})$。最后，我们为 ML-Net 架构推导出类似的逼近界，并指出 ML-Net 相较于推论 8 中纯多重网格近似的计算优势。请注意，本节仅考虑二维情形，但这些结果可以推广到任意空间维数。

对神经网络进行严谨的数学分析需要大量形式化工作，例如可参见（Petersen and Voigtländer，2018；Gühring 等，2020；Gühring and Raslan，2021）。我们假设，有意了解证明的读者熟悉一些常见技巧，例如神经网络的拼接/并行化，以及多项式、恒等函数和代数运算的逼近。为使我们的论述更贴近“实践者的视角”，我们采用了更精简的讲解方式，同时保留了足够的细节，以便读者理解证明的论证过程。所有证明均见附录 C。

我们将卷积神经网络视为由（各种类型的）卷积与常见深度学习库中的标准构建模块（例如跳跃连接和池化层）组合而成的任意计算图。请注意，这一概念也包括著名的 UNet 架构（Ronneberger 等，2015）以及其他具有相同递归结构的类 UNet 架构。

在理论分析中，我们考虑满足以下条件的激活函数 $\varrho \in L^\infty_{\mathrm{loc}}(\mathbb{R})$：存在 $x_0 \in \mathbb{R}$，使得 ϱ 在某个 $x_0 \in \mathbb{R}$ 的邻域内三次连续可微，且 $\varrho''(x_0) \neq 0$。这包括许多标准激活函数，例如指数线性单元、Softsign、Swish 和 Sigmoid 函数。对于（带泄漏的）ReLU，也可以进行类似但更复杂的分析；为简洁起见，我们不作讨论。

对于卷积神经网络 Ψ，我们用 $M(\Psi)$ 表示权重数量。由于我们考虑在均匀加密的正方形网格上、二维情形下的有限元空间 V_h，有限元系数向量 $\mathbf{x} \in \mathbb{R}^{\dim V_h}$ 可以视为二维数组。每当卷积神经网络处理$\mathbf{x}$时，我们都默认采用二维矩阵表示。我们定义$\|\mathbf{x}\|_{H^1} := \big\|\sum_{i=1}^{\dim V_h} \mathbf{x}_i \varphi_i\big\|_{H^1}$。在本节中，我们令 $D = [0,1]^2$，并假设问题 2.1 满足一致椭圆性。

我们首先介绍如何用类 UNet 卷积神经网络近似多重网格 V 循环 $\mathrm{MG}_{k_0,\,k}^{m}$ 的解，其中初始化为$\mathbf{u}_0$, ，扩散系数为 $\boldsymbol{\kappa}$，右端项为$\mathbf{f}$ 。

定理 6 设 $V_h \subseteq H_0^1(D)$ 为均匀正方形网格上的 *P1* 有限元空间。则存在常数 $C>0$，使得对于任意 M、$\varepsilon>0$ 和 m、k、$k_0 \in \mathbb{N}$，都存在一个卷积神经网络Ψ: $\mathbb{R}^{3\times\dim V_h} \to \mathbb{R}^{\dim V_h}$，满足

(i) $\left\|\Psi(\mathbf{u}_0,\boldsymbol{\kappa},\mathbf{f}) - \mathrm{MG}_{k_0,k}^{m}(\mathbf{u}_0,\boldsymbol{\kappa},\mathbf{f})\right\|_{H^1(D)} \le \varepsilon$ *for all* $\boldsymbol{\kappa},\mathbf{u}_0,\mathbf{f} \in [-M,M]^{\dim V_h}$,
(ii) 权重数量不超过 $M(\Psi) \le C\left(k_0 + k\log\left(\frac{1}{h}\right)\right)m$。

接下来的注 释中，我们将进一步介绍 Ψ 的架构及其含义。

注 7 定理 *6* 的证明表明， Ψ 类似于多个类 *UNet* 子网络的拼接，每个子网络都近似一个 *V*循环。由此我们得出两个结论:

- 总体而言，这为 *U-Net* 与多重网格方法之间的启发式关联提供了支持，并通过严格的数学分析指出了它们在多尺度相关任务中取得成功的一种可能原因，例如医学成像（ *Liu* 等，*2020*）或偏微分方程相关问题（见 *Cho* 等，*2021*；*Jiang* 等，*2021*）。

- 结合下一节中的数值结果，这进一步凸显了 *UNetSeq*（见注记 *5*）适用于求解问题 *2.1*。下一推论将对此作更具体的说明。

对于参数化偏微分方程问题 2.1，逐个对$\mathbf{y} \in \Gamma$ 应用多重网格求解器求解方程 (5)，即可轻松得到以下推论，该推论是定理 6 的直接结果。我们利用定理 3 中所需的平滑迭代次数与网格无关这一事实，并将 V 循环的次数选为 $m \approx \log(1/\varepsilon)$。应用三角不等式可得以下结果。

推论 8 考虑具有协调 *P1* 有限元空间 $V_h \subseteq H_0^1(D)$的离散化问题 *2.1*。假设 $\boldsymbol{\kappa}_{\mathbf{y}}$ 在所有$\mathbf{y} \in \Gamma$上均有界。则存在一个常数$C>0$，使得对于任意 $\varepsilon>0$，都存在一个卷积神经网络 Ψ: $\mathbb{R}^{2\times\dim V_h} \to \mathbb{R}^{\dim V_h}$，满足

(i) $\left\|\Psi(\boldsymbol{\kappa}_{\mathbf{y}},\mathbf{f}) - \mathbf{v}_h(\mathbf{y})\right\|_{H^1(D)} \le \varepsilon\|f\|_*$ *for all* $\mathbf{y} \in \Gamma$,
(ii) 权重数量有界于 $M(\Psi) \le C\log\left(\frac{1}{h}\right)\log\left(\frac{1}{\varepsilon}\right)$。

令神经网络近似误差等于有限元离散误差（$\varepsilon = h$），得到界 $M(\Psi) \le C\log(1/h)^2$。这表明参数数量至多按 $1/h$ 的多对数级增长，且与随机参数维数 p 无关。

最后，我们将在下一个推论中给出 ML-Net 架构的复杂度估计。在这里，ML-Net 的每个层级上，都使用定理 6 对多重网格 V 循环（在相应层级上）进行近似。

推论 9 考虑离散化问题 *2.1*。设空间 V_1、……、V_L 按照第 *3* 节针对 $L \in \mathbb{N}$、$f \in H^{-1}(D)$ 所定义，其离散化记为 $\mathbf{f} \in \mathbb{R}^{\dim V_L}$。假设离散化扩散系数 $\boldsymbol{\kappa}_{\mathbf{y}} \in \mathbb{R}^{\dim V_L}$ 在所有 $\mathbf{y} \in \Gamma$ 上均有界。那么，存在一个常数 $C > 0$，使得对于每个 $\varepsilon > 0$，都存在一个 *ML-Net* Ψ（如第 *4* 节所述），满足

(i) $\|\Psi(0, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_L(\mathbf{y})\|_{H^1(D)} \leq \varepsilon \|f\|_* \; for \; all \; \mathbf{y} \in \Gamma$,
(ii) $M(\Psi) \leq CL$ 对数 $\binom{1}{\varepsilon} + CL^2$。

In 在以下注释中，我们讨论前两个推论之间的关系。

注 10 平衡逼近误差与离散化误差，即令 $\varepsilon = h = 2^{-L}$，我们得出两个观察结果：

- 推论 *8* 和推论 *9* 均给出了 $M(\Psi) \leq CL^2$ 的权重数量上界。由此可知，根据我们的上界，使用 *ML-Net* 相较于 *U-NetSeq* 在表达能力方面并无明显优势。我们将在第 *6.2* 节中展示，数值实验支持这一发现。

- 尽管参数数量与逼近精度相近，*ML-Net* 的前向传播比 *U-NetSeq* 更高效。这是因为 *ML-Net* 的多层结构将计算负载更均匀地分配到不同分辨率上。请注意，卷积滤波器的参数数量与输入分辨率无关，而将输入与滤波器进行卷积的计算复杂度则与输入分辨率有关。具体而言，*ML-Net* 一次前向传播所需的运算次数为 $O(2^{2L}) = O(h^{-2})$，而 *U-NetSeq* 则需要 $O(L2^{2L}) = O(\log(h^{-1})h^{-2})$ 次运算。这是 *ML-Net* 的一项优势，因此其训练时间更短（见第 *6* 节）。

6 数值结果

本节旨在通过预测解的平均相对 H^1距离，评估 ML-Net 以及较简单的 UNetSeq 在若干常见基准问题上的表现。此外，我们还测试了在连续网格层级上逐步减少训练样本数量时 ML-Net 的性能。这一做法受多层理论启发，有助于降低训练复杂度（更详细的评估见注 10）。

我们评估了在系数函数 a_k具有不同计算复杂度、扩散率场 κ 由式 (3) 给出且参数分布 π各异的情况下，我们的方法的性能。对于所有测试案例，我们均选择单位正方形区域 $D = [0,1]^2$、$a_0(x) := a_0 \in \mathbb{R}$ 为常数，且右端项为 $f \equiv 1$。测试问题定义如下。

1. **均匀光滑场**。令 $\Gamma = [-1,1]^p$ 和 $\pi = U[-1,1]^p$。此外，令 $a_0 := 1$，并选择 a_k 为平面傅里叶模态，具体如 (Eigel 等人，2019b) 和 (Geist 等人，2021) 所述，其衰减率为 $\|a_k\|_\infty = 0.1k^{-2}$，其中 $1 \leq k \leq p$。
2. **对数正态光滑场**。令 $\Gamma = \mathbb{R}^p$ 和 $\pi = N(\mathbf{0}, \mathrm{Id}_p)$。定义 $\kappa(\mathbf{y}, x) := \exp(\tilde{\kappa}(\mathbf{y}, x))$，其中 $\tilde{\kappa}$ 等于均匀案例中的 κ，且 $a_0 := 0$。

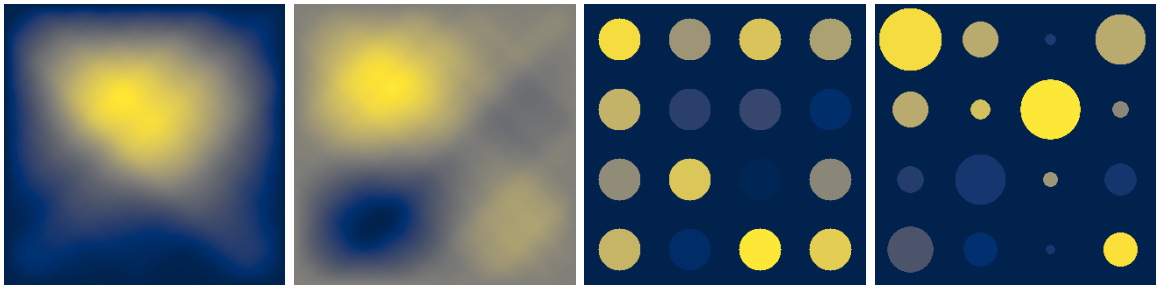


图 2： 四个测试案例中 $\kappa(\cdot,\ \mathbf{y})$ 的示例实现。从左到右：均匀案例（$p = 100$）、对数正态案例（$p = 100$）、饼干问题（$p = 16$）、具有可变半径的饼干问题（$p = 32$）。颜色映射已进行归一化处理，以便观察。

3. **固定半径夹杂物的饼干问题**。令 $\Gamma = [-1,1]^p$，其中 p 有一个自然平方根，且 $\pi = U[-1,1]^p$。此外，定义 $a_0 := 0.1$ 和 $a_k(x) = \mathcal{X}_{D_k}(x)$，其中 D_k 是以立方晶格为中心、半径固定为 $r = 0.3p^{-1/2}$ 的圆盘，适用于 $1 \le k \le p$。

4. **具有可变半径夹杂物的饼干问题**。作为对饼干问题的扩展，我们还假设圆盘半径会发生变化。为此，令 p' 表示（平方数量的）饼干，$p = 2p'$、$\Gamma = [-1,1]^p$ 和 $\pi = U[-1,1]^p$。扩散系数定义为

$$\kappa(\mathbf{y}, x) := a_0 + \sum_{k=1}^{p'} \mathbf{y}_{2k-1} \mathcal{X}_{\mathscr{D}_{k,r(\mathbf{y}_{2k})}}(x),$$

其中，$\mathscr{D}_{k,\ r}$ 是在立方晶格中以参数相关半径 r 为中心的圆盘。我们选择 $r(y)$，使得对于 $y \in [-1,1]$，半径在$[0.5(p')^{-1/2},\ 0。9(p')^{-1/2}]$中服从均匀分布。

图 2 展示了 4 种问题情形各自的一次随机实现。请注意，上述问题并非都满足第5节结果的理论假设。值得注意的是，对数正态案例并非一致椭圆，因此，如上所述，我们的理论保证仅以（任意）高概率成立。此外，对于可变半径的饼干问题，由于基函数 a_k 也依赖于参数，扩散系数非线性地依赖于参数向量。我们纳入这些情形，以便在多样且尤其具有挑战性的设置中研究我们的方法。

我们使用开源软件包 `FEniCS` (Logg and Wells, 2010) 和 GMRES(Saad and Schultz, 1986) 求解器执行有限元模拟以生成训练数据，并使用`PyTorch` (Paszke et al., 2019)构建神经网络模型。

除非另有说明，我们使用包含 10^4 个样本的训练数据集，并使用由独立同分布参数向量计算得到的 1024 个样本进行独立验证。测试时，在 1024 个独立生成的测试样本上评估各方法的性能。

对于多层分解，我们考虑 $L = 7$ 个分辨率层级，起始网格包含 5×5 个网格单元。以因子 $\eta = 2$ 进行迭代均匀网格细化后，最细层包含 320×320 个网格单元。为降低计算复杂度，我们只计算

最细层上的解，并使用节点插值推导较粗层上的解以及式（6）中的修正项。此外，我们在包含 1280×1280 个网格单元的高保真网格上计算参考测试解，其自由度约为 1.6×10^6。

对于 UNetSeq，我们选择 U-Net 的数量，使其参数总数大致与 ML-Net 的 5.6×10^6 参数数量相当。训练时，我们使用 Adam（Kingma 和 Ba，2015）优化器，并在学习率衰减的设置下训练 200 个训练轮次。每次运行后，我们评估验证性能最佳的模型。ML-Net 在 NVIDIA Tesla P100 上的训练耗时约为 33 GPU 小时，UNetSeq 则耗时约为 46 GPU 小时（另见注 10）。

6.1 误差指标

我们考察相对于最精细分辨率层级 L 上的解以及高保真参考解的平均相对 H^1 误差（MRH^1）和平均相对 L^2 误差（MRL^2）。对于参数 $\mathbf{y}_1, \dots, \mathbf{y}_N \in \Gamma$、预测值 $u_1, \dots, u_N \in V_L$，以及分别映射到最精细网格和参考网格上的离散空间的解算子 $v_L \colon \Gamma \to V_L$ 和 $v_{\mathrm{ref}} \colon \Gamma \to H^1(D)$，我们定义

$$\mathcal{E}_{\mathrm{MR}*} := \sqrt{\frac{\sum_{i=1}^N \|u_i - v_L(\mathbf{y}_i)\|_*^2}{\sum_{i=1}^N \|v_L(\mathbf{y}_i)\|_*^2}} \quad \text{and} \quad \mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}*} := \sqrt{\frac{\sum_{i=1}^N \|u_i - v_{\mathrm{ref}}(\mathbf{y}_i)\|_*^2}{\sum_{i=1}^N \|v_{\mathrm{ref}}(\mathbf{y}_i)\|_*^2}},$$

其中 $* \in \{H^1, \ L^2\}$。

6.2 测试案例结果

表 1 和表 2 展示了上述数值测试案例在随机参数维数变化时，ML-Net 和 UNetSeq 的$\mathcal{E}_{\mathrm{MR}H^1}$ 和 $\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}H^1}$。附录 B 的表 4 和表 5 展示了$\mathcal{E}_{\mathrm{MR}L^2}$ 和 $\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}L^2}$。图 3 还展示了随机选取的解和神经网络预测值。

与此前报道的这些基准问题的基于深度学习的方法的表现（Geist 等人，2021；Li 等人，2021a；Lu 等人，2021；Grossmann 等人，2023）相比，我们的方法取得了一个至两个数量级的提升。

Geist 等人（2021）发现，其基于深度学习的方法的表现强烈依赖于随机参数维数。相比之下，尽管随着参数维数增加，问题变得更加复杂，ML-Net 和 UNetSeq 的表现始终非常稳定。这与第5节推导出的卷积神经网络参数无关界相符。

$\mathcal{E}_{\mathrm{MR}H^1}$ 通常显著低于$\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}H^1}$。这表明，对于这些情况，最精细网格 $L = 7$ 上的神经网络近似误差可能比有限元近似误差低几个数量级。相反，表 4 和表 5 中$\mathcal{E}_{\mathrm{MR}L^2}$ 与$\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}L^2}$ 之间的差异则远不那么明显。

将 ML-Net 与 UNetSeq 进行比较，我们发现这两种模型的表现通常相当。ML-Net 的模型准确度方差较低，表明采用合适的多层分解时，训练稳定性有所提升。

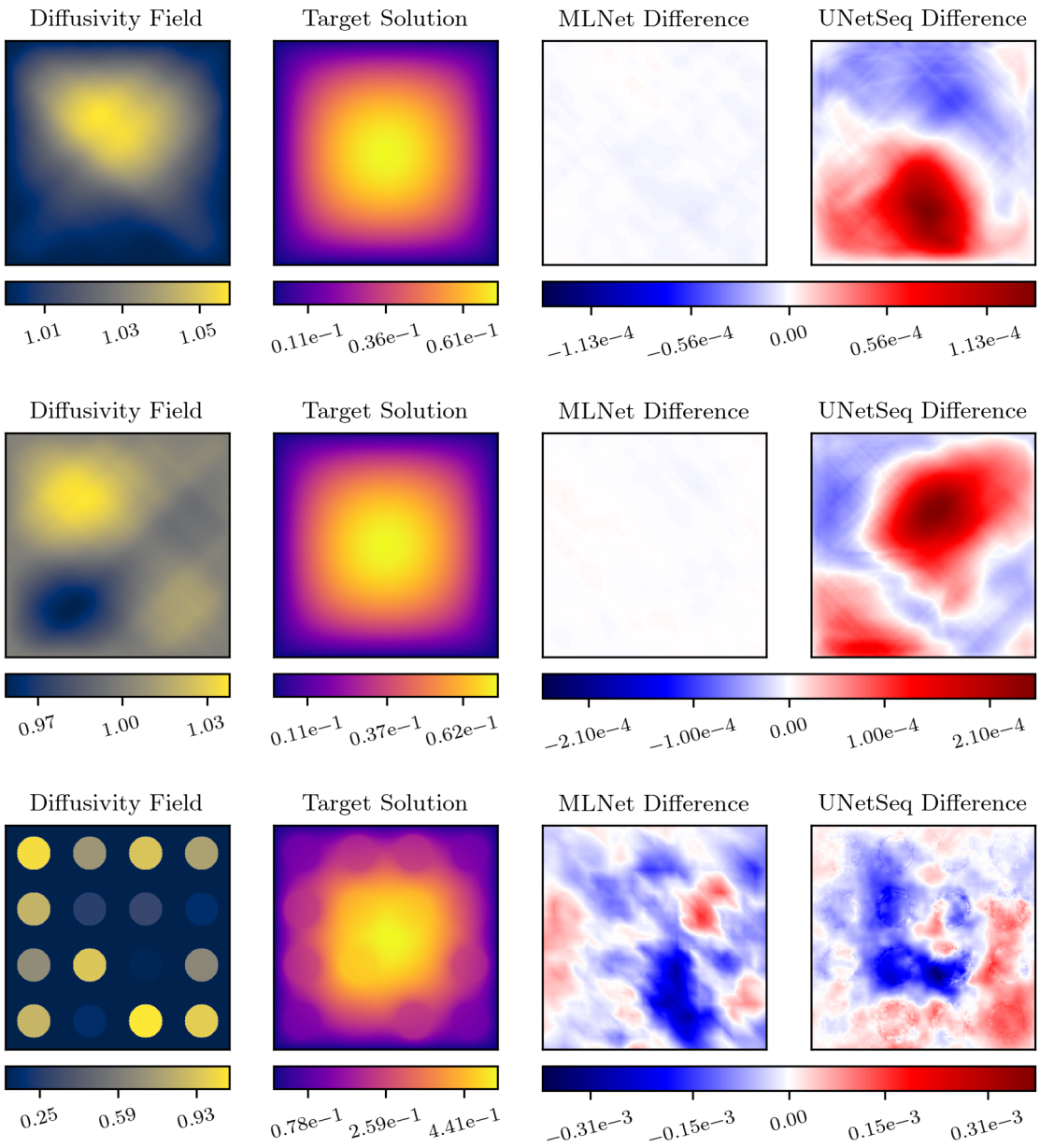


图 3： 不同基准问题的实现（从上到下：均匀分布$p = 100$、对数正态分布 $p = 100$、固定 Cookie）。左侧各列展示场实现及其对应的解。右侧展示 ML-Net 和 UNetSeq 的预测值相对于精确目标解的误差。

问题	参数维度 p	$\mathcal{E}_{\mathrm{MR}H^1}$	
		ML-Net	UNetSeq
均匀分布	10	$2.24\mathrm{e}{-4} \pm 9.62\mathrm{e}{-5}$	$9.75\mathrm{e}{-5} \pm 2.10\mathrm{e}{-5}$
	50	$2.21\mathrm{e}{-4} \pm 2.96\mathrm{e}{-5}$	$2.08\mathrm{e}{-3} \pm 2.79\mathrm{e}{-3}$
	100	$2.27\mathrm{e}{-4} \pm 2.72\mathrm{e}{-5}$	$8.89\mathrm{e}{-5} \pm 1.00\mathrm{e}{-5}$
	200	$2.31\mathrm{e}{-4} \pm 3.50\mathrm{e}{-5}$	$2.05\mathrm{e}{-3} \pm 2.76\mathrm{e}{-3}$
对数正态分布	10	$2.15\mathrm{e}{-4} \pm 6.18\mathrm{e}{-5}$	$2.83\mathrm{e}{-3} \pm 3.70\mathrm{e}{-3}$
	50	$9.73\mathrm{e}{-4} \pm 1.04\mathrm{e}{-3}$	$3。71\mathrm{e}{-3} \pm 4.91\mathrm{e}{-3}$
	100	$2.64\mathrm{e}{-4} \pm 1.73\mathrm{e}{-5}$	$2.82\mathrm{e}{-3} \pm 4.31\mathrm{e}{-3}$
	200	$3。00\mathrm{e}{-4} \pm 1.43\mathrm{e}{-5}$	$1.97\mathrm{e}{-4} \pm 4.22\mathrm{e}{-5}$
固定 Cookie	16	$9。41\mathrm{e}{-4} \pm 1.12\mathrm{e}{-4}$	$1.10\mathrm{e}{-3} \pm 3.76\mathrm{e}{-4}$
	64	$1。85\mathrm{e}{-3} \pm 1.38\mathrm{e}{-4}$	$7.20\mathrm{e}{-4} \pm 1.43\mathrm{e}{-4}$
cookie变量	32	$4.69\mathrm{e}{-3} \pm 1.41\mathrm{e}{-3}$	$3。69\mathrm{e}{-3} \pm 4.53\mathrm{e}{-4}$
	128	$5.98\mathrm{e}{-3} \pm 1.13\mathrm{e}{-4}$	$3.08\mathrm{e}{-3} \pm 5.07\mathrm{e}{-4}$

表 1： $\mathcal{E}_{\mathrm{MR}H^1}$ ML-Net 和 UNetSeq 在所有测试案例上的评估结果。

问题	参数维度 p	$\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}H^1}$	
		ML-Net	UNetSeq
均匀分布	10	$5.33\mathrm{e}{-3} \pm 4.50\mathrm{e}{-6}$	$5.33\mathrm{e}{-3} \pm 7.01\mathrm{e}{-7}$
	50	$5.33\mathrm{e}{-3} \pm 1.15\mathrm{e}{-6}$	$6.24\mathrm{e}{-3} \pm 1.28\mathrm{e}{-3}$
	100	$5.33\mathrm{e}{-3} \pm 1.63\mathrm{e}{-6}$	$5.33\mathrm{e}{-3} \pm 8.55\mathrm{e}{-7}$
	200	$5.33\mathrm{e}{-3} \pm 1.46\mathrm{e}{-6}$	$6。22\mathrm{e}{-3} \pm 1.25\mathrm{e}{-3}$
对数正态分布	10	$5.33\mathrm{e}{-3} \pm 2.93\mathrm{e}{-6}$	$6.78\mathrm{e}{-3} \pm 2.04\mathrm{e}{-3}$
	50	$5.51\mathrm{e}{-3} \pm 2.49\mathrm{e}{-4}$	$7.53\mathrm{e}{-3} \pm 3.10\mathrm{e}{-3}$
	100	$5.33\mathrm{e}{-3} \pm 5.54\mathrm{e}{-7}$	$6.90\mathrm{e}{-3} \pm 2.71\mathrm{e}{-3}$
	200	$5.34\mathrm{e}{-3} \pm 3.03\mathrm{e}{-6}$	$5.33\mathrm{e}{-3} \pm 1.69\mathrm{e}{-6}$
固定 Cookie	16	$7.09\mathrm{e}{-2} \pm 1.87\mathrm{e}{-5}$	$7.09\mathrm{e}{-2} \pm 7.39\mathrm{e}{-6}$
	64	$9.73\mathrm{e}{-2} \pm 1.16\mathrm{e}{-5}$	$9.73\mathrm{e}{-2} \pm 5.62\mathrm{e}{-6}$
cookie变量	32	$7.83\mathrm{e}{-2} \pm 3.22\mathrm{e}{-4}$	$7.81\mathrm{e}{-2} \pm 2.39\mathrm{e}{-4}$
	128	$1。12\mathrm{e}{-1} \pm 2.58\mathrm{e}{-4}$	$1。12\mathrm{e}{-1} \pm 2.76\mathrm{e}{-5}$

表 2： $\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}H^1}$ ML-Net 和 UNetSeq 在所有测试案例上的评估结果。

6.3 在更精细层级上逐步减少样本进行训练

随机多层方法的一个显著优势是，大多数样本点都可以在较粗层级上以较低的成本计算，而在最精细网格上只需少量样本（Teckentrup 等，2015；Lye 等，2021；Harbrecht 等，2016；Ballani 等，2016）。我们将这一理念应用于 ML-Net，使训练样本数量随层级递增而呈指数级减少。为此，我们在训练 ML-Net 的每个层级时仅使用数据集的一部分。具体而言，我们将后续每个层级的样本数量减半，即对于层级 $\ell = 1$、.....、 L，我们使用 $N_\ell := 2^{1-\ell} \times 10^4$ 训练样本。我们发现，训练时交替使用低层级样本和已知所有修正项的样本，是实现稳定优化所必需的。在最粗层级上生成一个包含 1000 个样本、且后续层级的样本数量呈指数级衰减的多尺度数据集，其计算预算相当于 232 个全分辨率样本。

表 3 展示了两个测试案例（均匀分布和对数正态分布，带有 $p = 100$）的误差：一个是使用衰减的多尺度数据集训练的 ML-Net，另一个是使用在可比计算预算下生成的全分辨率数据集（232 个样本）训练的 UNetSeq。我们有两点观察：第一，与表 4 中包含 1000 个样本的完整数据集相比，使用每个层级样本数递减的数据训练 ML-Net，几乎不会降低其性能。第二，与使用 1000 个样本训练 UNetSeq（表 4）以及使用衰减数据集训练 ML-Net（表 3）相比，使用计算预算相当的全分辨率数据集训练 UNetSeq 会显著降低其性能。

表 3 所示的 MRL^2 误差表明，可以达到较低的误差，至少比其他论文报告的结果低一个数量级；另见附录 B 中的表 4 和表 5。请注意，在我们的实验中，观察到的 MRH^1 误差受有限元近似误差的下限约束，即受最细网格的分辨率限制。要充分发挥多层方法的优势，需要在更细网格上进行训练，以降低有限元近似误差，使其与较小的神经网络近似误差相当。因此，减少高保真训练样本是训练模型的关键一步，能够使模型达到与传统有限元求解器相当的总体 MRH^1 精度。

我们得出结论：ML-Net 可以在更细的层级上用更少的样本进行有效训练，从而显著降低总体训练成本和数据生成成本，进而使我们的模型能够应用于更细网格。

7 结论

在这项工作中，我们将成熟的多层算法中的概念与神经网络相结合，以高效求解具有挑战性的高维参数偏微分方程。我们给出了理论复杂度分析，揭示了类 UNet 架构能够以任意高的精度逼近经典多重网格算法。我们还表明，ML-Net 架构为学习参数化 Darcy 问题的参数到解算子提供了一种有优势的方法。我们通过多项数值基准实验展示了该方法的性能，结果表明，相较于以往针对参数化偏微分方程的基于深度学习的方法，其性能显著提升。事实上，所展示的逼近质量与低秩最小二乘方法、随机 Galerkin 方法、压缩感知和随机配置法等同类最佳技术相当。此外，我们还表明，多层架构能够在训练过程中减少用于更精细修正的数据点数量，而对实际性能的影响微乎其微

方法	误差	数据集	均匀分布	对数正态分布
ML-Net	$E_{\mathrm{MR}H^1}$	衰减	$2.86\mathrm{e}{-4} \pm 2.24\mathrm{e}{-5}$	$4.05\mathrm{e}{-4} \pm 2.48\mathrm{e}{-5}$
UNetSeq	$E_{\mathrm{MR}H^1}$	固定	1。$14\mathrm{e}{-3} \pm 6.47\mathrm{e}{-4}$	$5.92\mathrm{e}{-3} \pm 4.66\mathrm{e}{-3}$
ML-Net	$E^{\mathrm{ref}}_{\mathrm{MR}H^1}$	衰减	$5.34\mathrm{e}{-3} \pm 1.22\mathrm{e}{-6}$	$5.34\mathrm{e}{-3} \pm 1.61\mathrm{e}{-6}$
UNetSeq	$E^{\mathrm{ref}}_{\mathrm{MR}H^1}$	固定	$5.49\mathrm{e}{-3} \pm 1.59\mathrm{e}{-4}$	$8.53\mathrm{e}{-3} \pm 3.53\mathrm{e}{-3}$
ML-Net	$E_{\mathrm{MR}L^2}$	衰减	$5.75\mathrm{e}{-5} \pm 4.43\mathrm{e}{-6}$	$9.39\mathrm{e}{-5} \pm 5.42\mathrm{e}{-6}$
UNetSeq	$E_{\mathrm{MR}L^2}$	固定	$5.38\mathrm{e}{-4} \pm 3.93\mathrm{e}{-4}$	$3.76\mathrm{e}{-3} \pm 3.23\mathrm{e}{-3}$
ML-Net	$E^{\mathrm{ref}}_{\mathrm{MR}L^2}$	衰减	$6.65\mathrm{e}{-5} \pm 2.16\mathrm{e}{-6}$	$9.91\mathrm{e}{-5} \pm 4.59\mathrm{e}{-6}$
UNetSeq	$E^{\mathrm{ref}}_{\mathrm{MR}L^2}$	固定	$5.40\mathrm{e}{-4} \pm 3.91\mathrm{e}{-4}$	$3.76\mathrm{e}{-3} \pm 3.23\mathrm{e}{-3}$

表 3: ML-Net 和 UNetSeq 在两种参数维度为 $p = 100$ 的问题情形下，使用缩减的数据集规模进行训练时的所有误差。ML-Net 在每个级别使用数量减半的训练样本进行训练，而 UNetSeq 总共使用 232 个样本进行训练。

。这使得我们能够扩展到精细得多的分辨率，而不会导致计算复杂度大幅增加。最后需要指出的是，尽管本文研究的是标量线性椭圆方程，但我们的方法也可用于各种可能具有非线性或向量值特征的问题。未来的研究方向包括对此类问题进行分析，以及将我们的方法扩展到自适应网格。

致谢与资金披露

作者感谢 Reinhold Schneider 就该主题进行富有成效的讨论。I. Gühring 感谢德国研究联合会（DFG）资助的研究培训组“生命科学与流体动力学中的微分方程驱动与数据驱动模型：跨学科研究培训组（DAEDALUS）”（GRK 2433），以及柏林工业大学为“理论与实践视角下的（参数化）偏微分方程深度学习”提供的博士后奖学金。M. Eigel 感谢德国研究联合会（DFG）SPP 1886“用于结构数值设计的多态不确定性建模”和 DFG SPP 2298 “深度学习的理论基础”提供的部分资助。我们感谢魏尔施特拉斯研究所提供计算设施，并感谢其 IT 支持团队。

参考文献

Ben Adcock 和 Nick Dexter。深度神经网络函数逼近中的理论与实践鸿沟。*SIAM* 数据科学数学杂志，3(2): 624–655，2021。

Anima Anandkumar、Kamyar Azizzadenesheli、Kaushik Bhattacharya、Nikola Kovachki、Zongyi Li、Burigede Liu 和 Andrew Stuart。神经算子：用于偏微分方程的图核网络。载于*ICLR 2020* 深度神经模型与微分方程融合研讨会，2020。Ivo Babuska、Fabio Nobile 和 Raul Tempone。针对具有随机输入数据的椭圆偏微分方程的随机配置法。*SIAM*数值分析杂志，45:1005–1034，2007。doi: 10.1137/050645142。Pierre Baldi。生物医学数据科学中的深度学习。《生物医学数据科学年度评论》，1(1):181–205，2018。Jonas Ballani、Daniel Kressner 和 Michael Peters。随机数据下的 PDE 多层张量近似。随机过程与偏微分方程：分析与计算，5:400–427，2016。Christian Beck、Martin Hutzenthaler、Arnulf Jentzen 和 Benno Kuckuck。偏微分方程基于深度学习的近似方法概述。《离散与连续动力系统-*B*》，28(6):3697–3746，2023。ISSN 1531-3492。doi: 10.3934/dcdsb.2022238。Julius Berner、Markus Dablander 和 Philipp Grohs。通过深度学习数值求解高维柯尔莫哥洛夫偏微分方程参数族。神经信息处理系统进展，33:16615–16627，2020a。Julius Berner、Philipp Grohs 和 Arnulf Jentzen。泛化误差分析：深度人工神经网络上的经验风险最小化克服了数值逼近 Black–Scholes 偏微分方程中的维数灾难。*SIAM* 数据科学数学杂志，2(3):631–657，2020b。Saakaar Bhatnagar、Yaser Afshar、Shaowu Pan、Karthik Duraisamy 和 Shailendra Kaushik。使用卷积神经网络预测气动流场。《计算力学》，64(2):525–545，2019。ISSN 1432-0924。doi: 10.1007/s00466-019-01740-0。Dietrich Braess。有限元：理论、快速求解器及其在固体力学中的应用。剑桥大学出版社，2007。Dietrich Braess 和 Wolfgang Hackbusch。包含 V 循环的多重网格方法新收敛性证明。*SIAM*数值分析杂志，20:967–975，1983。doi: 10.1137/0720066。Yuyan Chen、Bin Dong 和 Jinchao Xu。Meta-MgNet：用于求解参数化偏微分方程的元多重网格网络。《计算物理学杂志》，页码 110996，2022。Abdellah Chkifa、Albert Cohen、Giovanni Migliorati、Fabio Nobile 和 Raul Tempone。采用随机评估的离散最小二乘多项式逼近：应用于参数化与随机椭圆偏微分方程。*ESAIM*：数学建模与数值分析，49(3):815–837，2015。

Sung-Jin Cho、Seo-Won Ji、Jun-Pyo Hong、Seung-Won Jung 和 Sung-Jea Ko。重新思考单图像去模糊中的由粗到精方法。载于*2021*年 *IEEE/CVF*计算机视觉国际会议（*ICCV*），第 4621–4630 页，2021。doi：10.1109/ICCV48922.2021.00460。Philippe G. Ciarlet。椭圆问题的有限元方法。工业与应用数学学会，2002年。doi：10.1137/1.9780898719208。Albert Cohen 和 Ronald DeVore。高维参数化偏微分方程的近似。*Acta Numerica*，24：1–159，2015。Albert Cohen 和 Giovanni Migliorati。具有对数正态系数的椭圆偏微分方程的近乎最优近似方法。*Mathematics of Computation*，2023。Weinan E、Jiequn Han 和 Arnulf Jentzen。基于深度学习的高维抛物型偏微分方程和反向随机微分方程数值方法。*Communications in Mathematics and Statistics*，5(4)：349–380，2017年。ISSN 2194-671X。doi：10.1007/s40304-017-0117-6。Martin Eigel、Claude Jeffrey Gittelson、Christoph Schwab 和 Elmar Zander。自适应随机Galerkin有限元法。*Computer Methods in Applied Mechanics and Engineering*，270：247–269，2014年3月。doi：10.1016/j.cma.2013.11.015。URL `https://doi.org/10.1016/j.cma.2013.11.015`。Martin Eigel、Reinhold Schneider、Philipp Trunschke 和 Sebastian Wolf。变分蒙特卡罗——连接机器学习与高维偏微分方程的概念。*Advances in Computational Mathematics*，45(5-6)：2503–2532，2019年10月。doi：10.1007/s10444-019-09723-8。URL `https://doi.org/10.1007/s10444-019-09723-8`。Martin Eigel、Reinhold Schneider、Philipp Trunschke 和 Sebastian Wolf。变分蒙特卡罗——连接机器学习与高维偏微分方程的概念。*Advances in Computational Mathematics*，45(5)：2503–2532，2019b。Martin Eigel、Manuel Marschall、Max Pfeffer 和 Reinhold Schneider。层次张量表示中对数正态系数的自适应随机Galerkin有限元法。*NumerischeMathematik*，145(3)：655–692，2020年6月。doi：10.1007/s00211-020-01123-1。Oliver G Ernst、Bjorn Sprungk 和 Lorenzo Tamellini。针对可数多个高斯随机变量函数的稀疏配置收敛性（及其在椭圆偏微分方程中的应用）。*SIAM*数值分析杂志，56(2)：877–905，2018。Yuwei Fan、Jordi Feliu-Faba、Lin Lin、Lexing Ying 和 Leonardo Zepeda-Núnez。基于层次嵌套基的多尺度神经网络。数学科学研究，6(2)：1–28，2019a。Yuwei Fan、Lin Lin、Lexing Ying 和 Leonardo Zepeda-Núnez。基于层次矩阵的多尺度神经网络。*Multiscale Modeling & Simulation*，17(4)：1189–1213，2019b。

Ruiqi Gao、Yang Lu、Junpei Zhou、Song-Chun Zhu 和 Ying Nian Wu。通过多重网格建模与采样学习生成式卷积网络。*2018*年 *IEEE/CVF*计算机视觉与模式识别会议，第 9155–9164 页，2018年。Moritz Geist、Philipp Petersen、Mones Raslan、Reinhold Schneider 和 Gitta Kutyniok。利用深度神经网络求解参数扩散方程。科学计算杂志，88(1)：1–37，2021。Martin Genzel、Ingo Gühring、Jan MacDonald 和 Maximilian März。通过深度学习实现断层成像逆问题的近乎精确恢复。载于Ka malika Chaudhuri、Stefanie Jegelka、Le Song、Csaba Szepesvári、Gang Niu 和 Sivan Sabato 编，国际机器学习会议，*ICML 2022*，*2022*年*7*月*17–23*日，美国马里兰州巴尔的摩，第 162 卷，机器学习研究会议录，第 7368–7381 页。PMLR，2022年。Ian Goodfellow、Yoshua Bengio 和 Aaron Courville。深度学习。麻省理工学院出版社，剑桥，2016。Philipp Grohs 和 Lukas Herrmann。具有边界条件的高维椭圆偏微分方程的深度神经网络近似。*IMA*数值分析杂志，42(3)：2055–2082，2022。Philipp Grohs、Fabian Hornung、Arnulf Jentzen 和 Philippe Von Wurstemberger。证明人工神经网络能够克服数值近似布莱克-斯科尔斯偏微分方程时的维数灾难。*arXiv*预印本 *arXiv:1809.02362*，2018年。Tamara G Grossmann、Urszula Julia Komorowska、Jonas Latz 和 Carola-Bibiane Schönlieb。物理信息神经网络能否胜过有限元方法? *arXiv*预印本 *arXiv:2302.04107*，2023。Ingo Gühring 和 Mones Raslan。光滑度空间中具有可编码权重的神经网络的近似率。*Neural Networks*，134：107–130，2021。Xiaoxiao Guo、Wei Li 和 Francesco Iorio。用于稳态流近似的卷积神经网络。载于第*22*届*ACM SIGKDD*知识发现与数据挖掘国际会议论文集，KDD ’16，第 481–490 页，美国纽约，2016年。计算机协会。ISBN 9781450342322。doi：10.1145/2939672. 2939738。Ingo Gühring、Gitta Kutyniok 和 Philipp Petersen。深度 ReLU 神经网络在 $W^{s,\,p}$ 范数下进行近似的误差界。分析与应用（新加坡），18(05)：803–859，2020。Ingo Gühring、Mones Raslan 和 Gitta Kutyniok。深度神经网络的表达能力。载于Philipp Grohs 和 Gitta Kutyniok 编，深度学习的数学方面，第 149–199 页。剑桥大学出版社，2022年。Wolfgang Hackbusch。多重网格方法及其应用，第 4 卷。施普林格科学与商业媒体，2013年。

Jiequn Han、Arnulf Jentzen 等。用于高维抛物型偏微分方程和倒向随机微分方程的基于深度学习的数值方法。数学与统计学通讯，5(4):349–380，2017年。Helmut Harbrecht、Michael Peters 和 Markus Siebenmorgen。针对具有对数正态分布扩散系数的偏微分方程的多层加速求积。*SIAM/ASA*不确定性量化杂志，4(1):520–551，2016年。Juncai He 和 Jinchao Xu。Mgnet：一种融合多重网格与卷积神经网络的统一框架。中国科学：数学，62(7):1331–1354，2019年。ISSN 1869-1862。doi：10.1007/s11425-019-9547-2。J.S. Hesthaven 和 S. Ubbiali。使用神经网络对非线性问题进行非侵入式降阶建模。《计算物理学杂志》，363:55–78，2018年。ISSN 0021-9991。doi：https://doi.org/10.1016/j.jcp.2018.02.037。URL `https://www.sciencedirect.com/science/article/pii/S0021999118301190`。Zhihao Jiang、Pejman Tahmasebi 和 Zhiqiang Mao。采用自回归策略的深度残差U-Net卷积神经网络，用于大规模地质系统中的流体流动预测。水资源进展，150:103878，2021年2月。doi：10.1016/j.advwatres. 2021.103878。John A Keith、Valentin Vassilev-Galindo、Bingqing Cheng、Stefan Chmiela、Michael Gastegger、Klaus-Robert Müller 和 Alexandre Tkatchenko。结合机器学习与计算化学，对化学系统进行预测性洞察。化学评论，121(16):9816–9872，2021年。Ehsan Kharazmi、Zhongqiang Zhang 和 George Em Karniadakis。用于求解偏微分方程的变分物理信息神经网络。*arXiv*预印本 *arXiv:1912.00873*，2019年。Diederik P. Kingma 和 Jimmy Ba。Adam：一种随机优化方法。载于第三届国际学习表征会议（*ICLR 2015*），美国加利福尼亚州圣迭戈，*2015*年*5*月*7*日至*9*日，会议论文集，2015年。Omar M Knio 和 OP Le Maitre。利用多项式混沌分解进行计算流体力学中的不确定性传播。流体动力学研究，38(9):616，2006年。Gitta Kutyniok、Philipp Petersen、Mones Raslan 和 Reinhold Schneider。深度神经网络与参数化偏微分方程的理论分析。构造逼近，55 (1):73–125，2022年。Zongyi Li、Nikola Kovachki、Kamyar Azizzadenesheli、Burigede Liu、Andrew Stuart、Kaushik Bhattacharya 和 Anima Anandkumar。用于参数化偏微分方程的多极图神经算子。神经信息处理系统进展，33:6755–6766，2020年。Zongyi Li、Nikola Borislavov Kovachki、Kamyar Azizzadenesheli、Burigede liu、Kaushik Bhattacharya、Andrew Stuart 和 Anima Anandkumar。用于

参数化偏微分方程。在国际学习表征会议，2021a年。

Zongyi Li、Nikola Borislavov Kovachki、Kamyar Azizzadenesheli、Burigede liu、Kaushik Bhattacharya、Andrew Stuart 和 Anima Anandkumar。用于参数化偏微分方程的傅里叶神经算子。载于国际学习表征会议，2021b。

Liangliang Liu、Jianhong Cheng、Quan Quan、Fang-Xiang Wu、Yu-Ping Wang 和 Jianxin Wang。医学图像分割中U形网络综述。*Neurocomputing*，409：244–258，2020年。ISSN 0925-2312。

Anders Logg 和 Garth N. Wells。DOLFIN。*ACM*数学软件汇刊，37(2)：1–28，2010年。ISSN 1557-7295。doi：10.1145/1731022.1731030。

Gabriel J Lord、Catherine E Powell 和 Tony Shardlow。计算随机偏微分方程导论，第50卷。剑桥大学出版社，2014年。

Lu Lu、Xuhui Meng、Zhiping Mao 和 George Em Karniadakis。《Deepxde：用于求解微分方程的深度学习库》。*SIAM*评论，63(1):208–228，2021。

Kjetil O Lye、SiddharthaMishra 和 Roberto Molinaro。提升机器学习算法精度的多层级过程。欧洲应用数学杂志，32(3):436–469，2021。

Siddhartha Mishra 和 Roberto Molinaro。物理信息神经网络在近似偏微分方程时的泛化误差估计。*IMA*数值分析杂志，43(1):1–43，01 2022。ISSN 0272-4979。doi: 10.1093/imanum/drab093。

Fabio Nobile、Raúl Tempone 和 Clayton G Webster。针对具有随机输入数据的偏微分方程的稀疏网格随机配置法。*SIAM*数值分析杂志，46(5):2309–2345，2008。

Frank Noé、Alexandre Tkatchenko、Klaus-Robert Müller 和 Cecilia Clementi。用于分子模拟的机器学习。物理化学年度评论，71(1):361–390，2020。

Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga、Alban Desmaison、Andreas Kopf、Edward Yang、Zachary DeVito、Martin Raison、Alykhan Tejani、Sasank Chilamkur- thy、Benoit Steiner、Lu Fang、Junjie Bai 和 Soumith Chintala。PyTorch：一种命令式风格的高性能深度学习库。载于 H. Wallach、H. Larochelle、A. Beygelzi- mer、F. d'Alché-Buc、E. Fox 和 R. Garnett 编，神经信息处理系统进展 *32*，第 8024–8035 页。Curran Associates公司，2019。

Philipp Petersen 和 Felix Voigtländer。使用深度 ReLU 神经网络对分段光滑函数进行最优逼近。*Neural Networks*，108:296–330，2018。

Maziar Raissi、Paris Perdikaris 和 George Em Karniadakis。物理信息神经网络：一种用于求解涉及非线性偏微分方程的正问题与逆问题的深度学习框架。《计算物理学杂志》，378:686–707，2019。ISSN 0021-9991。doi: 10.1016/j.jcp.2018.10.045。Amuthan A Ramabathiran 和 Prabhu Ramachandran。Spinn：用于偏微分方程的稀疏、基于物理且部分可解释的神经网络。《计算物理学杂志》，445: 110600，2021。Lewis Fry Richardson 和 Richard Tetley Glazebrook。IX. 用有限差分近似求解涉及微分方程的物理问题的算术方法，并应用于砌体坝的应力分析。伦敦皇家学会哲学汇刊。*A*辑：数学或物理类论文，210(459-470):307–357，1911年。doi: 10.1098/rsta.1911.0009。Olaf Ronneberger、Philipp Fischer 和 Thomas Brox。U-net：用于生物医学图像分割的卷积网络。载于医学图像计算与计算机辅助干预国际会议，第234–241页。施普林格，2015。Youcef Saad 和 Martin H Schultz。GMRES：一种用于求解非对称线性方程组的广义最小残差算法。*SIAM*科学计算杂志，7(3): 856–869，1986。ISSN 0196-5204。Peter Sadowski 和 Pierre Baldi。自然科学中的深度学习：物理学应用。载于 Lev Rozonoer、Boris Mirkin 和 Ilya Muchnik（编），《布拉弗曼机器学习读本：从创始至今的关键思想——纪念 *Emmanuil Braverman* 逝世*40*周年国际会议，波士顿，*MA*，美国，*2017*年*4*月*28*日至*30*日，特邀报告》，第269–297页。施普林格国际出版公司，查姆，2018。Christoph Schwab 和 Claude Jeffrey Gittelson。高维参数化与随机偏微分方程的稀疏张量离散化。*Acta Numerica*，20:291–467，2011。Christoph Schwab 和 Jakob Zech。高维深度学习：广义多项式混沌展开在不确定性量化中的神经网络表达率。分析与应用，17(01):19–55，2019。W. Shi、J. Caballero、F. Huszar、J. Totz、A. P. Aitken、R. Bishop、D. Rueckert 和 Z. Wang。使用高效的子像素卷积神经网络进行实时单图像和视频超分辨率。载于*2016 IEEE*计算机视觉与模式识别会议（*CVPR*），第1874–1883页，美国加利福尼亚州洛斯阿拉米托斯，2016年6月。IEEE计算机学会。doi: 10.1109/CVPR.2016.207。Justin Sirignano 和 Konstantinos Spiliopoulos。深度Galerkin方法：一种用于求解偏微分方程的深度学习算法。《计算物理学杂志》，375:1339–1364，2018。ISSN 0021-9991。doi: 10.1016/j.jcp.2018.08.029。A. L. Teckentrup、P. Jantsch、C. G. Webster 和 M. Gunzburger。一种用于随机输入数据偏微分方程的多层随机配置法。*SIAM/ASA*不确定性量化杂志，3(1):1046–1074，2015。doi: 10.1137/140969002。

Benjamin Ummenhofer、Lukas Prantl、Nils Thuerey 和 Vladlen Koltun。采用连续卷积的拉格朗日流体模拟。发表于国际学习表征会议，2020年。Xuefeng Xu 和 Chen-Song Zhang。非精确二重网格方法的收敛性分析：一个理论框架。*SIAM*数值分析杂志，60(1):133–156，2022年。Dmitry Yarotsky。深度 ReLU 网络近似的误差界。神经网络，94:103–114，2017年。Harry Yserentant。多重网格方法收敛性证明的新旧进展。*Acta Numerica*，2:285–326，1993年。Bing Yu 等。深度 Ritz 方法：一种用于求解变分问题的深度学习数值算法。*Communications in Mathematics and Statistics*，6(1):1–12，2018年。Yinhao Zhu 和 Nicholas Zabaras。用于代理建模和不确定性量化的贝叶斯深度卷积编码器–解码器网络。《计算物理学杂志》，366:415–447，2018年。ISSN 0021-9991。doi: 10.1016/j.jcp.2018.04.018。

附录 A. 精确的 ML-Net 架构和训练细节

在我们的实验中，我们构建了包含 $L = 7$ 个层级的 ML-Net 和 UNetSeq。卷积层使用带零填充的 3×3 卷积核。上采样（↑）和下采样（↓）层分别通过 5×5 转置步幅卷积和步幅卷积实现，步幅均为 2。我们不使用任何归一化层，并在所有架构中始终采用 ReLU 激活函数。除上采样和下采样层外，ML-Net 和 UNetSeq 的 UNet 子单元在每个尺度上都包含两个卷积层。跳跃连接通过拼接前序层的输出实现。

对于 ML-Net，我们选择每个层级的 U-Net 数量$\mathbf{R}_\ell$，$\ell = 1, \dots, L$，使每个 ML-Net 子网络$_\ell$ 约有 8×10^5 个可训练参数。经验证，这一数量能够产生令人满意的结果，并得到 $\mathbf{R} = [11, 9, 5, 4, 3, 2, 2]$。为将扩散率场 $\boldsymbol{\kappa}$ 下采样至子网络 ML-Net$_\ell$所需的输入分辨率，$L - \ell + 1$ 采用步幅卷积，随后进行 ReLU 激活。除最粗层级外，ML-Net 的卷积层均使用 32 个通道；最粗层级则使用 64 个通道。对于除最粗层级以外的所有层级 $\ell = 2, \dots, L$，ML-Net$_\ell$ 子网络会以半分辨率输出四个通道的预测值，并使用像素反洗牌层（Shi 等，2016）组合出完整预测值。这样做的原因是，对于二分细分，细网格修正中每隔一个值之间的相似性高于数组中相邻值之间的相似性。对于 UNetSeq，所有卷积层均使用 64 个通道。

ML-Net 和 UNetSeq 训练 200 个训练轮次，前 60 个训练轮次的初始学习率为 10^{-3}。随后，在接下来的 100 个训练轮次中，学习率线性衰减至 2×10^{-5}，之后在剩余训练期间保持不变。受内存限制，ML-Net 和 UNetSeq 的批量大小分别设为 20 和 16。采用标准配置的 Adam 优化器（Kingma 和 Ba，2015），参数为 β_1 =0.99, $\beta_2 = 0.999$，且不使用权重衰减。

附录 B. 附加表格

以下表格展示了所考察的神经网络架构在所有已呈现测试案例中的相对 L^2 误差。

问题情形	参数维度 p	$\mathcal{E}_{\mathrm{MR}L^2}$	
		ML-Net	UNetSeq
均匀分布	10	$9.28\mathrm{e}{-5} \pm 6.92\mathrm{e}{-5}$	$3.72\mathrm{e}{-5} \pm 9.18\mathrm{e}{-6}$
	50	$4.88\mathrm{e}{-5} \pm 6.82\mathrm{e}{-6}$	$1.21\mathrm{e}{-3} \pm 1.66\mathrm{e}{-3}$
	100	$4。90\mathrm{e}{-5} \pm 9.56\mathrm{e}{-6}$	$3。21\mathrm{e}{-5} \pm 3.06\mathrm{e}{-6}$
	200	$4。81\mathrm{e}{-5} \pm 7.38\mathrm{e}{-6}$	$1。17\mathrm{e}{-3} \pm 1.61\mathrm{e}{-3}$
对数正态分布	10	$7。44\mathrm{e}{-5} \pm 1.75\mathrm{e}{-5}$	$1。66\mathrm{e}{-3} \pm 2.22\mathrm{e}{-3}$
	50	$7.46\mathrm{e}{-4} \pm 9.61\mathrm{e}{-4}$	$2。30\mathrm{e}{-3} \pm 3.13\mathrm{e}{-3}$
	100	$6.54\mathrm{e}{-5} \pm 6.71\mathrm{e}{-6}$	$1。72\mathrm{e}{-3} \pm 2.80\mathrm{e}{-3}$
	200	$7.35\mathrm{e}{-5} \pm 2.82\mathrm{e}{-6}$	$7.94\mathrm{e}{-5} \pm 2.14\mathrm{e}{-5}$
固定 Cookie	16	$3。29\mathrm{e}{-4} \pm 3.26\mathrm{e}{-5}$	$2.35\mathrm{e}{-4} \pm 1.37\mathrm{e}{-5}$
	64	$5.32\mathrm{e}{-4} \pm 1.80\mathrm{e}{-5}$	$1。40\mathrm{e}{-4} \pm 4.32\mathrm{e}{-5}$
cookie变量	32	$1.33\mathrm{e}{-3} \pm 1.05\mathrm{e}{-4}$	$7.68\mathrm{e}{-4} \pm 2.45\mathrm{e}{-5}$
	128	$2.01\mathrm{e}{-3} \pm 8.14\mathrm{e}{-5}$	$7.14\mathrm{e}{-4} \pm 7.74\mathrm{e}{-5}$

表 4: $\mathcal{E}_{\mathrm{MR}L^2}$ 对 ML-Net 和 UNetSeq 在所有测试案例上的评估。

问题情形	参数维度 p	$\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}L^2}$	
		ML-Net	UNetSeq
均匀分布	10	$1。05\mathrm{e}{-4} \pm 6.75\mathrm{e}{-5}$	$4.77\mathrm{e}{-5} \pm 6.80\mathrm{e}{-6}$
	50	$5.86\mathrm{e}{-5} \pm 7.02\mathrm{e}{-6}$	$1。22\mathrm{e}{-3} \pm 1.66\mathrm{e}{-3}$
	100	$5.50\mathrm{e}{-5} \pm 5.07\mathrm{e}{-6}$	$4.19\mathrm{e}{-5} \pm 3.41\mathrm{e}{-6}$
	200	$5。67\mathrm{e}{-5} \pm 5.78\mathrm{e}{-6}$	$1.18\mathrm{e}{-3} \pm 1.60\mathrm{e}{-3}$
对数正态分布	10	$7.76\mathrm{e}{-5} \pm 1.29\mathrm{e}{-5}$	$1。66\mathrm{e}{-3} \pm 2.22\mathrm{e}{-3}$
	50	$7.59\mathrm{e}{-4} \pm 9.68\mathrm{e}{-4}$	$2。30\mathrm{e}{-3} \pm 3.13\mathrm{e}{-3}$
	100	$7。06\mathrm{e}{-5} \pm 4.38\mathrm{e}{-6}$	$1。72\mathrm{e}{-3} \pm 2.80\mathrm{e}{-3}$
	200	$8.31\mathrm{e}{-5} \pm 3.91\mathrm{e}{-6}$	$8.69\mathrm{e}{-5} \pm 1.97\mathrm{e}{-5}$
固定 Cookie	16	$6。19\mathrm{e}{-3} \pm 6.60\mathrm{e}{-5}$	$6。05\mathrm{e}{-3} \pm 3.50\mathrm{e}{-5}$
	64	$9.63\mathrm{e}{-3} \pm 4.33\mathrm{e}{-5}$	$9。48\mathrm{e}{-3} \pm 2.79\mathrm{e}{-5}$
cookie变量	32	$8。81\mathrm{e}{-3} \pm 3.41\mathrm{e}{-5}$	$8.49\mathrm{e}{-3} \pm 9.10\mathrm{e}{-5}$
	128	$1.64\mathrm{e}{-2} \pm 5.36\mathrm{e}{-5}$	$1.62\mathrm{e}{-2} \pm 6.70\mathrm{e}{-5}$

表5： $\mathcal{E}^{\mathrm{ref}}_{\mathrm{MR}L^2}$ 在所有测试案例上对 ML-Net 和 UNetSeq 进行评估。

附录 C. 第5节结果的证明

本节旨在证明第5节中的理论分析。为此，我们先介绍一些数学记号。

C.1 数学记号

对于某个函数 $f\colon D \subseteq \mathbb{R}^d \to \mathbb{R}^n$，我们记 $\|f\|_{L^\infty(D)} :=$ 本质上确界$_{x\in D} \|f(x)\|_\infty$，其中$\|\cdot\|_\infty$表示向量的最大范数。对于两个张量$\mathbf{x}$、$\mathbf{y} \in \mathbb{R}^{W\times H}$，其中 W， $H \in \mathbb{N}$,

我们用$\mathbf{x}\odot\mathbf{y}\in\mathbb{R}^{W\times H}$表示它们的逐点乘法。由于我们考虑的是二维情形下、定义在均匀加密的正方形网格上的有限元空间V_h，因此有限元系数向量$\mathbf{x}\in\mathbb{R}^{\dim V_h}$可以视为二维数组。每当用卷积神经网络处理 x 时，我们都默认其采用二维矩阵表示。

式 (2) 中的双线性形式诱导出与问题相关且依赖于参数的能量范数，其定义如下：

$$\|w\|_{A_{\mathbf{y}}}^2 := \int_D \kappa(\mathbf{y},x)|\nabla w(x)|^2\,\mathrm{d}x \quad \text{for} \quad w\in V. \tag{8}$$

考虑问题 2.1。在一致椭圆性假设[1], 下，能量范数与 H^1 范数等价（例如参见 Cohen 和 DeVore (2015)）。我们用 $c_{H^1}, C_{H^1}>0$ 表示与网格无关的常数，使得对于所有 $u\in V$ 和 $\mathbf{y}\in\Gamma$

$$c_{H^1}\|u\|_{A_{\mathbf{y}}} \le \|u\|_{H^1} \le C_{H^1}\|u\|_{A_{\mathbf{y}}}. \tag{9}$$

此外，对于截至第 $L\in\mathbb{N}$ 层的多层分解，以及按式 (6) 定义的 $\hat{}v_1,\dots,\hat{}v_L$，用 $C_{\mathrm{corr}}>0$ 表示常数，使得对于所有 $\mathbf{y}\in\Gamma$ 和 $\ell=1,\dots,L$

$$\|\hat{v}_\ell(\mathbf{y})\|_{H^1} \le C_{\mathrm{corr}}2^{-\ell}\|f\|_*. \tag{10}$$

注意，对于任意 $\mathbf{y}\in\Gamma$，问题 2.1 的解 $v_h(\mathbf{y})$ 满足

$$\|v_h(\mathbf{y})\|_{A_{\mathbf{y}}} \le C_{H^1}\|f\|_*. \tag{11}$$

为避免证明中引入过多数学负担，我们将讨论范围限制在以下注释中指定的某类均匀正方形网格。

注 11 对于 $D=[0,1]^2$、网格宽度 $h>0$ 和 $m:=1/h\in\mathbb{N}$，我们仅考虑由 m^2 个相同正方形组成的均匀正方形网格，每个正方形再细分为两个三角形，如图*4*所示。

C.2 卷积神经网络：术语与基本性质

本节简要介绍基本的卷积运算及其相应记号。我们主要关注二维情形。不过，所有定义和结果都可以轻松推广到任意维度。关于相关概念的深入阐述，请参阅（Goodfellow 等，2016，第9章）。

卷积神经网络中使用的卷积在一些细节上偏离了公认的卷积数学定义。在二维情况下，输入由一个张量 $\mathbf{x}_{\mathrm{in}}\in\mathbb{R}^{C_{\mathrm{in}}\times W_{\mathrm{in}}\times H_{\mathrm{in}}}$ 构成，其空间维度为 $W_{\mathrm{in}}, H_{\mathrm{in}}\in\mathbb{N}$，通道数为 $C_{\mathrm{in}}\in\mathbb{N}$；输出则由张量 $\mathbf{x}_{\mathrm{out}}\in\mathbb{R}^{C_{\mathrm{out}}\times W_{\mathrm{out}}\times H_{\mathrm{out}}}$给出。这里，$C_{\mathrm{out}}\in\mathbb{N}$ 表示输出通道数，$W_{\mathrm{out}}, H_{\mathrm{out}}\in\mathbb{N}$ 表示可能经过变换的空间维数。$\mathbf{x}_{\mathrm{out}}$是将 $\mathbf{x}_{\mathrm{in}}$ 与可学习的卷积核 $K\in\mathbb{R}^{C_{\mathrm{in}}\times C_{\mathrm{out}}\times W_K\times W_K}$进行卷积，并逐通道加上可学习偏置 $B\in\mathbb{R}^{C_{\mathrm{out}}}$后得到的结果；该卷积核的宽度为 $W_K\in\mathbb{N}$。我们使用三种不同类型的卷积：(i) 普通卷积，记作 $\mathbf{x}_{\mathrm{in}}*K$，其中 $W_{\mathrm{out}}=W_{\mathrm{in}}$和 $H_{\mathrm{out}}=H_{\mathrm{in}}$；(ii) 步长为二的卷积，记作 $\mathbf{x}_{\mathrm{in}}*^{2\mathrm{s}}K$，其中 $W_{\mathrm{out}}=\mathtt{b}W_{\mathrm{in}}/2\mathtt{c}-1$ 和$H_{\mathrm{out}}=\mathtt{b}H_{\mathrm{in}}/2\mathtt{c}-1$；(iii) 步长为二的转置卷积，记作 $\mathbf{x}_{\mathrm{in}}*^{2\mathrm{ts}}K$，其中 $W_{\mathrm{out}}:=2W_{\mathrm{in}}+1$

1. 在我们的设定中，这一点始终以很高的概率成立

以及 $H_{\text{out}} := 2H_{\text{in}} + 1$。此外，我们用 $M(\Phi) \in \mathbb{N}$ 表示可训练参数的数量，用 $L(\Phi) \in \mathbb{N}$ 表示（可能是卷积）神经网络 Φ 的层数。

许多先前的工作都集中于使用全连接神经网络逼近函数。通过一个简单的技巧，我们将这些逼近结果迁移到卷积神经网络。

定理 12 设 $D \subseteq \mathbb{R}^d$ 和 $f\colon D \to \mathbb{R}$ 为某个函数。此外，设 $\varepsilon > 0$ 和 Φ 为一个具有 d 维输入和一维输出的全连接神经网络，满足 $\|f - \Phi\|_{L^\infty(D)} \leq \varepsilon$，则存在一个卷积神经网络 Ψ，其

(*i*) 具有 d 个输入通道和一个输出通道；

(*ii*) 所有卷积核的空间维度均为 1×1；

(*iii*) 具有与 Φ 相同的激活函数；

(*iv*) $M(\Psi) = M(\Phi)$ 和 $L(\Psi) = L(\Phi)$；

(*v*) 对所有 $W \in \mathbb{N}$，我们有

$$\left\| \Psi - \hat{f} \right\|_{L^\infty(D^{W \times W})} \leq \varepsilon,$$

其中 $\hat{f}\colon D^{W \times W} \to \mathbb{R}^{W \times W}$ 是对 f 逐分量应用。

证明 证明可直接通过将 Φ 每一层中的仿射线性变换视为通道维度上的 1×1 卷积得出。 ■

在下一个推论中，使用定理 12 通过卷积神经网络近似两个输入张量的逐点乘法函数。

推论 13 设 $\varrho \in L^\infty_{\text{loc}}(\mathbb{R})$ 满足存在 $x_0 \in \mathbb{R}$，使得 ϱ 在某个 $x_0 \in \mathbb{R}$ 的邻域内三次连续可微，且 $\varrho''(x_0) \neq 0.$。设 $W \in \mathbb{N}$、$B > 0$ 和 $\varepsilon \in (0, 1/2)$，则存在一个激活函数为 ϱ、具有双通道输入和单通道输出的卷积神经网络 $\widetilde{\times}$，满足以下性质：

1. $\left\| \widetilde{\times}(\mathbf{x}, \mathbf{y}) - \mathbf{x} \odot \mathbf{y} \right\|_{L^\infty([-B,B]^{2 \times W \times W},\, d\mathbf{x}d\mathbf{y})} \leq \varepsilon$；
2. $L(\widetilde{\times}) = 2$ 和 $M(\widetilde{\times}) \leq 9$；
3. 所有卷积核的空间维度为 1×1。

证明 该证明由定理 12 以及（Gühring and Raslan，2021，推论 C.3）得出。 ■

C.3 逼近孤立的 V循环构件

使用卷积神经网络近似完整多重网格循环的一个主要中间步骤，是近似算子 A_κ。本节的理论基础是这一观察： $A_\kappa \mathbf{u}$ 可以由一个作用于 κ （在定义 14 中定义的积分表示下）和$\mathbf{u}$的卷积神经网络近似。我们首先定义证明中使用的一些基本概念。在本节余下部分中，我们令 $D = [0,1]^2$。

定义 14 设 T 是 D 的均匀三角剖分，并具有相应的协调 $P1$ 有限元空间 V_h，其基为 $\{\varphi_1, \dots, \varphi_{\dim V_h}\}$。此外，设 $\kappa \in H_0^1(D)$。

(i) 对于 $i \in \{1, \dots, \dim V_h\}$ 我们定义 $\mathrm{Neigh2D}(i) := \{j \in \{1, \dots, \dim V_h\} : \operatorname{supp} \varphi_i \cap \operatorname{supp} \varphi_j \neq \emptyset\}$。

(ii) 对于三角剖分中的每个顶点 $i \in \{1, \dots, \dim V_h\}$，我们枚举其六个相邻三角形（任意枚举，但对每个 i 均采用相同顺序），并将它们记为 $T_i^{(k)}$ $with k = 1, \dots, 6$ （示意见图4）。

(iii) 对于 $i = 1, \dots, \dim V_h$ 和 $k = 1, \dots, 6$，我们定义

$$\boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, i) := \int_{T_i^k} \kappa \, \mathrm{d}x$$

并使用记号

$$\boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k) := [\boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, i)]_{i=1}^{\dim V_h} \in \mathbb{R}^{\dim V_h}$$

and

$$\boldsymbol{\Upsilon}(\kappa, \mathcal{T}) := [\boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k)]_{k=1}^{6} \in \mathbb{R}^{6 \times \dim V_h}.$$

在下一个引理中，我们证明积分 $\boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, i)$ 可以通过卷积从（离散化的） κ 计算得出；此外，粗网格上的积分也可以通过卷积从细网格积分计算得出。

引理 15 设 T_h、T_{2h} 是网格尺度分别为 h 和 $2h$ 的嵌套三角剖分，其对应的有限元空间为 $V_{2h} \subseteq V_h \subseteq H_0^1(D)$。则有：

(i) 存在卷积核 $K \in \mathbb{R}^{1 \times 6 \times 3 \times 3}$，使得对于任意 $\kappa \in V_h$，当 $k \in \{1, \dots, 6\}$ 时，均有

$$(\boldsymbol{\kappa} * K)[k] = \boldsymbol{\Upsilon}(\kappa, \mathcal{T}_h, k).$$

(ii) 存在一个卷积核 $K \in \mathbb{R}^{6 \times 6 \times 3 \times 3}$，使得对于每个 $\kappa \in H_0^1(D)$ 和$k = 1, \dots, 6$，我们有

$$([\boldsymbol{\Upsilon}(\kappa, \mathcal{T}_h, 1), \dots, \boldsymbol{\Upsilon}(\kappa, \mathcal{T}_h, 6)] * K)[k] = \boldsymbol{\Upsilon}(\kappa, \mathcal{T}_{2h}, k).$$

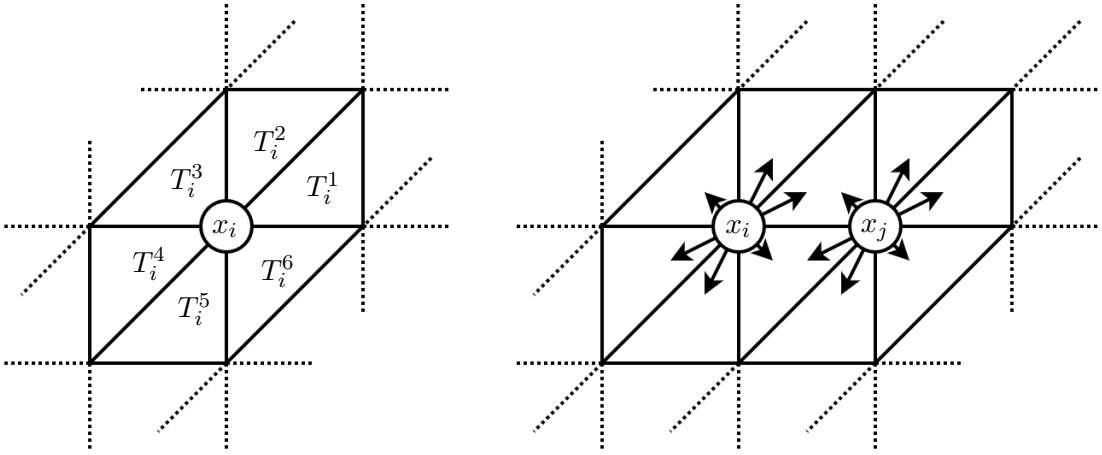


图4： 图示展示了如何对定义 14 (ii) 中均匀正方形网格上的一个网格点的六个相邻三角形进行一种可能的编号。请注意，这种分配存在冗余，因为从右图可以看出，$T_i^{(6)} = T_j^{(4)}$ 和 $T_{(1)\,i} = T_{(3)\,j}$。

证明 我们先证明 (i)。对于 $k = 1, \ldots 6$ 和 $i = 1, \ldots, \dim V_h$,

$$\boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, i) = \int_{T_i^k} \kappa \,\mathrm{d}x = \sum_{j=1}^{\dim V_h} \boldsymbol{\kappa}_j \int_{T_i^k} \varphi_j^{(h)} \,\mathrm{d}x = \sum_{\{j:\operatorname{supp}\varphi_j^{(h)} \cap T_i^k \neq \emptyset\}} \boldsymbol{\kappa}_j \frac{h^2}{3},$$

其中最后一步用到了 $\int_{T_i^k} \kappa \,\mathrm{d}x = h^2/3$ 当 $\operatorname{supp}\, \varphi_j^{(h)} \cap T_i^k \neq \emptyset$。显然，对于所有满足 $j \notin \mathrm{Neigh2D}(i)$ 的 $j \in \{1, \ldots, \dim\, V_h\}$，$\operatorname{supp}\, \varphi_j^{(h)} \cap T_i^k = \emptyset$。此外，相关邻居 $\{j\colon \operatorname{supp}\, \varphi_j^{(h)} \cap T_i^k \neq \emptyset\}$ 相对于索引 i 的位置，在 i 的二维平移下保持不变。结合这两个观察即可完成证明。

(ii) 直观上，步幅卷积能够在局部计算来自更精细的三角剖分的所有系数的均值，这一点显而易见。然而，要严谨地证明这一点将是一项繁琐的工作。因此，我们省略形式化证明，并参见图 5，其中展示了每个网格点处相邻的子三角形。 ■

下一个定理表明，利用定义 14 中 κ 的积分表示，我们现在可以通过将 κ 积分与卷积作用于$\mathbf{u}$后的输出进行逐点乘法，来表示 $A_\kappa \mathbf{u}$。

定理 16（ $A_\kappa \mathbf{u}$的表示） 对于 $k = 1, \ldots, 6$，存在卷积核 $K^{(k)} \in \mathbb{R}^{1\times1\times3\times3}$，使得对于函数

$$F : \mathbb{R}^{7\times \dim V} \to \mathbb{R}^{\dim V}, \quad (\mathbf{u}, \overline{\boldsymbol{\kappa}}^{(1)}, \ldots, \overline{\boldsymbol{\kappa}}^{(6)}) \mapsto \sum_{k=1}^{6} \overline{\boldsymbol{\kappa}}^{(k)} \odot (\mathbf{u} * K^{(k)}),$$

有 F 连续，并且对于任意 $\kappa \in H_0^1(D)$，我们有

$$F(\mathbf{u}, \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, 1), \ldots, \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, 6)) = A_\kappa \mathbf{u}.$$

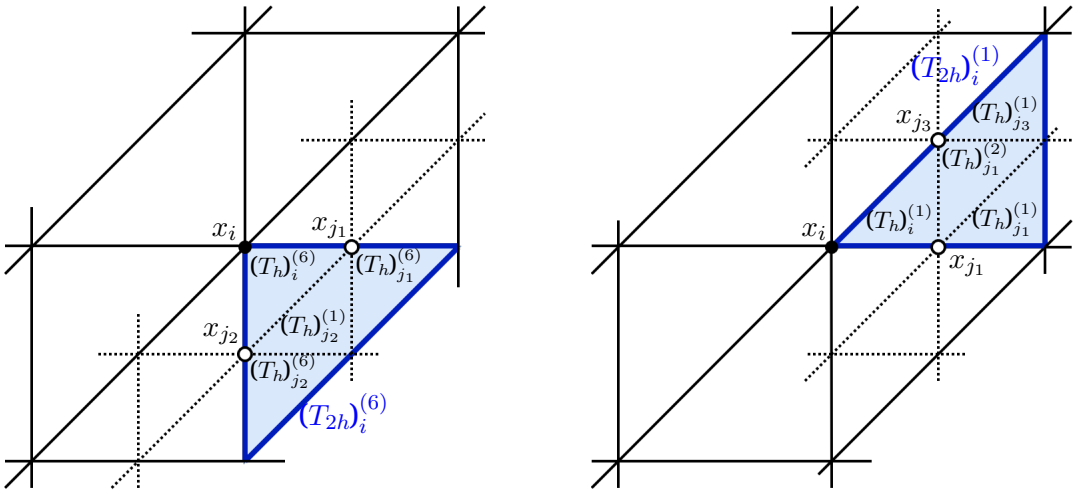


图 5： 在某个网格点处，更细网格的相邻三角形和子三角形示意图。该图展示了粗网格中与网格点 x_i 相邻的三角形（蓝色）如何细分。请注意，所有包含在其中的更细三角形都可以分配给细网格中与 x_i 直接相邻的网格点。因此，对与更细三角形相关联的扩散率值求和，可以通过细网格中的卷积（3×3）核来表示。

证明 我们先引入一些记号：令 $i, j \in \{1, \dots, \dim V_h\}$ 和 $k \in \{1, \dots, 6\}$，并将 $C_{ijk} \in \mathbb{R}$ 设为 $\langle \nabla\varphi_i, \nabla\varphi_j \rangle$ 在 T_i^k 上取到的常数。注意，如果 $i \notin \text{Neigh2D}(j)$，则 $C_{ijk} = 0$。现在我们将 A_κ 的每个元素表示为若干乘积之和：对于 $i, j \in \{1, \dots, \dim V_h\}$，我们有

$$(A_\kappa)_{ij} = \int_D \kappa \langle \nabla\varphi_i, \nabla\varphi_j \rangle \, \mathrm{d}x = \sum_{k=1}^{6} \int_{T_i^k} \kappa \langle \nabla\varphi_i, \nabla\varphi_j \rangle \, \mathrm{d}x = \sum_{k=1}^{6} \Upsilon(\kappa, \mathcal{T}, k, i) C_{ijk},$$

其中最后一步用到了定义 14（ii）。对于 $j \in \{1, \dots, \dim V_h\}$，我们现在可以将 $(A_\kappa \mathbf{u})_j$ 改写为

$$\begin{aligned}
(A_\kappa \mathbf{u})_j &= \sum_{i=1}^{\dim V_h} \mathbf{u}_i \sum_{k=1}^{6} \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, j) C_{ijk} \\
&= \sum_{k=1}^{6} \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, j) \sum_{i=1}^{\dim V_h} \mathbf{u}_i C_{ijk} \\
&= \sum_{k=1}^{6} \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k, j) \sum_{i \in \text{Neigh2D}(j)} \mathbf{u}_i C_{ijk}.
\end{aligned}$$

这里，第一步使用 A_κ 的定义，第三步则仅对相邻指标使用 $\operatorname{supp} \varphi_i \cap \operatorname{supp} \varphi_j \neq \emptyset$。由于 C_{ijk} 仅取决于 i 相对于 j 的相对位置（在二维意义上），且与 j 无关，因此内层求和可表示为

向量化形式的零填充卷积，卷积核为 3×3（邻居数量），其取值取决于 k，即:

$$A_\kappa \mathbf{u} = \sum_{k=1}^{6} \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, k) \odot (\mathbf{u} * K^{(k)}) = F(\mathbf{u}, \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, 1), \ldots, \boldsymbol{\Upsilon}(\kappa, \mathcal{T}, 6)),$$

其中 $K^{(k)}$ 是相应的卷积核。 ■

在下面的注释中，我们将详细说明边界条件的处理方式。

注释 17 在我们的设定中，我们仅考虑内部网格点上的基函数。另一种做法是纳入边界基函数，并将其系数约束为零，以施加齐次狄利克雷边界条件。将有限元算子表示为内部网格顶点上的零填充卷积，可通过填充时隐式地将边界基函数系数置为零，自然实现齐次狄利克雷边界条件。

利用定理 16 中将 $A_\kappa \mathbf{u}$ 表示为卷积后接逐点乘法的形式，现在可以应用推论 13，得到一个卷积神经网络近似。

定理 18（对 $A_\kappa \mathbf{u}$ 的逼近） 设 T 是 D 的均匀三角剖分，其对应的协调 *P1* 有限元空间为 V。此外，设激活函数 $\varrho \in L^\infty_{\mathrm{loc}}(\mathbb{R})$满足：存在 $x_0 \in \mathbb{R}$，使得 ϱ 在某个 $x_0 \in \mathbb{R}$ 的邻域内三次连续可微，且 $\varrho''(x_0) \neq 0$。那么，对于任意 $\varepsilon > 0$ 和 $M > 0$，都存在一个卷积神经网络 $\Psi_{\varepsilon,\, M}$：$\mathbb{R}^{7\times \dim V_h} \to \mathbb{R}^{\dim V_h}$，其规模与 ε和 M无关，使得

$$\|\Psi_{\varepsilon,M} - F\|_{L^\infty([-M,M]^{7\times \dim V_h})} \leq \varepsilon.$$

证明 通过以下三个步骤得到最终结果:

(i) 显然，存在一个单层卷积神经网络可以实现该映射

$$[\mathbf{u}, \overline{\boldsymbol{\kappa}}^{(1)}, \ldots, \overline{\boldsymbol{\kappa}}^{(6)}] \mapsto [\mathbf{u} * K^{(k)}, \overline{\boldsymbol{\kappa}}^{(k)}]_{k=1}^{6}.$$

(ii) 推论 13 表明，存在一个两层卷积神经网络 $\tilde{\Psi}$，其权重和1×1 卷积核的数量至多为 9，使得

$$\left\| \tilde{\Psi}([\mathbf{u} * K^{(k)}, \overline{\boldsymbol{\kappa}}^{(k)}]_{k=1}^{6}) - [\overline{\boldsymbol{\kappa}}^{(k)} \odot (\mathbf{u} * K^{(k)})]_{k=1}^{6} \right\|_\infty \leq \frac{\varepsilon}{6}.$$

(iii) 逐通道加法显然可以通过一个带有1×1 卷积核的单层卷积神经网络来构造。

将上述三个卷积神经网络拼接起来，即可完成证明。 ■

回顾一下，延拓及其对应的加权限制是多重网格循环中的关键运算。下面的备注指出，这些运算可以自然地表示为卷积。

备注19（加权限制与延拓）考虑在均匀正方形网格上，使用嵌套协调 *P1* 有限元空间 $V_1 \subseteq V_2 \subseteq H_0^1(D)$ 进行两层分解。令 P 为相应的延拓矩阵（定义 *2*）。那么，存在卷积核 K_1、$K_2 \in \mathbb{R}^{1\times1\times3\times3}$，使得对于任意有限元系数向量

(i) $\mathbf{u} \in \mathbb{R}^{\dim V_2}$，我们有 $\mathbf{u} *^{2s} K_1 = P^T\mathbf{u}$；

(ii) $\mathbf{u} \in \mathbb{R}^{\dim V_1}$，我们有 $\mathbf{u} *^{2ts} K_2 = P\mathbf{u}$。

C.4 综合起来

本节将卷积神经网络对多重网格算法各个计算步骤的近似结合起来，以便用卷积神经网络近似任意数量的多重网格迭代。为了组合各个近似解，我们首先给出一个辅助引理。

引理 20 设 $n \in \mathbb{N}$ 和 $d_1, \dots, d_{n+1} \in \mathbb{N}$。对于 $i = 1, \dots, n$，设 $f_i\colon \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$ 为连续函数，并将它们拼接定义为 F，即

$$F\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_{n+1}}, \quad F := f_n \circ \dots \circ f_1.$$

那么，对于每个 $M, \varepsilon > 0$，都存在 $M_1, \dots, M_n > 0$ 和 $\varepsilon_1, \dots, \varepsilon_n > 0$，使得以下结论成立：若 $\tilde{f}_i\colon \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$ 是满足 $\|f_i - \tilde{f}_i\|_{L^\infty([-M_i,M_i]^{d_i})} \leq \varepsilon_i$ 的函数，其中 $i = 1, \dots, n$，则

$$\left\|F - \tilde{f}_n \circ \dots \circ \tilde{f}_1\right\|_{L^\infty([-M,M]^{d_1})} \leq \varepsilon.$$

证明 我们对 $n \in \mathbb{N}$ 使用归纳法证明该命题。对于 $n = 1$，该命题显然成立。定义函数 $g\colon \mathbb{R}^{d_1} \to \mathbb{R}^{d_n}$，对于 $n > 1$，定义如下

$$g := f_{n-1} \circ \dots \circ f_1.$$

令 $\varepsilon_n := \varepsilon/2$，并令 $M_n := \|g\|_{L^\infty([-M,M]^{d_1})} + \varepsilon < \infty$（因为 g 是连续的）。由于 f_n 在紧集上均匀连续，因此存在 $0 < \delta \leq \varepsilon$，使得对于所有 $z, \tilde{z} \in [-M_n, M_n]^{d_n}$，若 $\|z - \tilde{z}\|_\infty \leq \delta$，则有 $\|f_n(z) - f_n(\tilde{z})\|_\infty \leq \frac{\varepsilon}{2}$。

现在设 $M_1, \dots, M_{n-1}$ 和 $\varepsilon_1, \dots, \varepsilon_{n-1}$ 由归纳法假设（取 $M = M$ 和 $\varepsilon = \delta$）给出，并设 $\tilde{f}_i\colon \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$ 是 $i = 1, \dots, n-1$ 的相应近似解。那么（同样根据归纳法假设），有

$$\left\|g - \tilde{f}_{n-1} \circ \dots \circ \tilde{f}_1\right\|_{L^\infty([-M,M]^{d_1})} \leq \delta.$$

令定值为 $\tilde{g} := f_{n-1} \circ \dots \circ f_1$，可得 $\|\tilde{g}\|_{L^\infty([-M,M]^{d_1})} \leq \|g\|_{L^\infty([-M,M]^{d_1})} + \delta \leq M_n$。通过推导可得……，证明至此完成。

$$\begin{aligned}
&\left\|F - (\tilde{f}_n \circ \dots \circ \tilde{f}_1)\right\|_{L^\infty([-M,M]^{d_1})} \\
&\leq \|(f_n \circ g) - (f_n \circ \tilde{g})\|_{L^\infty([-M,M]^{d_1})} + \left\|(f_n \circ \tilde{g}) - (\tilde{f}_n \circ \tilde{g})\right\|_{L^\infty([-M,M]^{d_1})} \\
&\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.
\end{aligned}$$

■

经过前述准备，我们现在可以证明主要结果了。我们首先证明卷积神经网络对多重网格算法的逼近。

定理 6 的证明 总体证明思路是将函数 $\mathrm{MG}_{k_0,k}^m\colon \mathbb{R}^{3\times\dim V_L}\to\mathbb{R}^{\dim V_L}$（其中 $V_L=V_h$）分解为有限个连续函数的拼接，每个函数要么可由卷积神经网络实现，要么可由卷积神经网络任意精确地逼近。应用引理 20 可得，这些卷积神经网络的拼接（仍为一个卷积神经网络）能够逼近 $\mathrm{MG}_{k_0,k}^m$。下面给出以下定义：

(i) **积分扩散系数**。令 $K\in\mathbb{R}^{1\times6\times3\times3}$ 为引理 15(i) 中的卷积核。对于函数

$$f_{\mathrm{in}}\colon \mathbb{R}^{3\times\dim V_L}\to\mathbb{R}^{8\times\dim V_L},\quad \begin{pmatrix}\mathbf{u}\\ \boldsymbol{\kappa}\\ \mathbf{f}\end{pmatrix}\mapsto\begin{pmatrix}\mathbf{u}\\ \boldsymbol{\kappa}*K\\ \mathbf{f}\end{pmatrix},$$

由此，根据引理 15(i)，可得 $f_{\mathrm{in}}([\mathbf{u}\ \boldsymbol{\kappa},\ \mathbf{f}])=[\mathbf{u}\ \boldsymbol{\Upsilon}(\kappa_h,\ \mathcal{T}^L),\ \mathbf{f}]$。

(ii) **平滑迭代**。对于 $\ell=1,\dots,L$，令 $F_\ell\colon\mathbb{R}^{7\times\dim V_\ell}\to\mathbb{R}^{\dim V_\ell}$ 如定理 16 中所定义。然后定义

$$f_{\mathrm{sm}}^\ell\colon\mathbb{R}^{8\times\dim V_\ell}\to\mathbb{R}^{8\times\dim V_\ell},\quad \begin{pmatrix}\mathbf{u}\\ \overline{\boldsymbol{\kappa}}^{(1)}\\ \vdots\\ \overline{\boldsymbol{\kappa}}^{(6)}\\ \mathbf{f}\end{pmatrix}\mapsto\begin{pmatrix}\mathbf{u}+\omega(\mathbf{f}-F_\ell(\mathbf{u},\overline{\boldsymbol{\kappa}}^{(1)},\dots,\overline{\boldsymbol{\kappa}}^{(6)}))\\ \overline{\boldsymbol{\kappa}}^{(1)}\\ \vdots\\ \overline{\boldsymbol{\kappa}}^{(6)}\\ \mathbf{f}\end{pmatrix}.$$

由定理 16 可得

$$f_{\mathrm{sm}}^\ell([\mathbf{u},\boldsymbol{\Upsilon}(\kappa_h,\mathcal{T}^\ell),\mathbf{f}])=[\mathbf{u}+\omega(\mathbf{f}-A_{\kappa_h}^\ell\mathbf{u}),\boldsymbol{\Upsilon}(\kappa_h,\mathcal{T}^\ell),\mathbf{f}].$$

(iii) **受限残差**。与前一步类似，我们定义

$$f_{\mathrm{resi}}^\ell\colon\mathbb{R}^{8\times\dim V_\ell}\to\mathbb{R}^{9\times\dim V_\ell},\quad \begin{pmatrix}\mathbf{u}\\ \overline{\boldsymbol{\kappa}}^{(1)}\\ \vdots\\ \overline{\boldsymbol{\kappa}}^{(6)}\\ \mathbf{f}\end{pmatrix}\mapsto\begin{pmatrix}\mathbf{u}\\ \mathbf{f}-F_\ell(\mathbf{u},\overline{\boldsymbol{\kappa}}^{(1)},\dots,\overline{\boldsymbol{\kappa}}^{(6)})\\ \overline{\boldsymbol{\kappa}}^{(1)}\\ \vdots\\ \overline{\boldsymbol{\kappa}}^{(6)}\\ \mathbf{f}\end{pmatrix},$$

并由定理 16 得到

$$f_{\mathrm{resi}}^\ell([\mathbf{u},\boldsymbol{\Upsilon}(\kappa_h,\mathcal{T}^\ell),\mathbf{f}])=[\mathbf{u},\mathbf{f}-A_{\kappa_h}^\ell\mathbf{u},\boldsymbol{\Upsilon}(\kappa_h,\mathcal{T}^\ell),\mathbf{f}].$$

现在令 $[K_1, \dots, K_6] \in \mathbb{R}^{6\times6\times3\times3}$ 为引理 15(ii) 中的卷积核，令$P_{\ell-1} \in \mathbb{R}^{\dim V_\ell \times \text{维数}}$ $V_{\ell-1}$ 为定义 2 中的延拓矩阵（其转置矩阵用作加权限制），则我们定义

$$f^\ell_{\mathrm{rest}} \colon \mathbb{R}^{9\times\dim V_\ell} \to \mathbb{R}^{8\times\dim V_\ell} \times \mathbb{R}^{8\times\dim V_{\ell-1}}, \quad \begin{pmatrix} \mathbf{u} \\ \mathbf{r} \\ \overline{\boldsymbol{\kappa}}^{(1)} \\ \vdots \\ \overline{\boldsymbol{\kappa}}^{(6)} \\ \mathbf{f} \end{pmatrix} \mapsto \begin{bmatrix} \begin{pmatrix} \mathbf{u} \\ \overline{\boldsymbol{\kappa}}^{(1)} \\ \vdots \\ \overline{\boldsymbol{\kappa}}^{(6)} \\ \mathbf{f} \end{pmatrix} \\ \begin{pmatrix} \mathbf{0} \\ \overline{\boldsymbol{\kappa}}^{(1)} * K_1 \\ \vdots \\ \overline{\boldsymbol{\kappa}}^{(6)} * K_6 \\ P_{\ell-1}^T \mathbf{r} \end{pmatrix} \end{bmatrix}.$$

注意，对于 $\boldsymbol{\Upsilon}(\kappa_h, \mathcal{T}^\ell, k) * K_k = \boldsymbol{\Upsilon}(\kappa_h, \mathcal{T}^{\ell-1}, k)$, $k = 1, \dots, 6$。

(iv) 将粗网格校正添加到细网格。如上所述，设 $P_{\ell-1} \in \mathbb{R}^{\dim V_\ell \times \text{维} V_{\ell-1}}$ 为定义 2 中的延拓矩阵。我们定义函数

$$f^\ell_{\mathrm{prol}} \colon \mathbb{R}^{8\times\dim V_\ell} \times \mathbb{R}^{8\times\dim V_{\ell-1}} \to \mathbb{R}^{8\times\dim V_\ell}, \quad \begin{bmatrix} \begin{pmatrix} \mathbf{u} \\ \overline{\boldsymbol{\kappa}}^{(1)} \\ \vdots \\ \overline{\boldsymbol{\kappa}}^{(6)} \\ \mathbf{f} \end{pmatrix} \\ \begin{pmatrix} \mathbf{e} \\ \vdots \end{pmatrix} \end{bmatrix} \mapsto \begin{pmatrix} \mathbf{u} + P_{\ell-1}\mathbf{e} \\ \overline{\boldsymbol{\kappa}}^{(1)} \\ \vdots \\ \overline{\boldsymbol{\kappa}}^{(6)} \\ \mathbf{f} \end{pmatrix}.$$

(v) **返回解**。我们定义

$$f_{\mathrm{out}} \colon \mathbb{R}^{8\cdot\dim V_\ell} \to \mathbb{R}^{\dim V_\ell}, \quad \begin{pmatrix} \mathbf{u} \\ \overline{\boldsymbol{\kappa}}^{(1)} \\ \vdots \\ \overline{\boldsymbol{\kappa}}^{(6)} \\ \mathbf{f} \end{pmatrix} \mapsto \mathbf{u}.$$

我们按如下方式构造一个位于第 $\ell = 1$ 层,k 上的单个 V循环 $\mathrm{VC}^\ell_{k_0} L$

$$\begin{aligned} \mathrm{VC}^\ell_{k_0,k} &:= \left(\bigcirc_{i=1}^k f^\ell_{sm}\right) \circ f^\ell_{\mathrm{prol}} \circ (\mathrm{Id}, \mathrm{VC}^{\ell-1}_{k_0,k}) \circ f^\ell_{\mathrm{rest}} \circ f^\ell_{\mathrm{resi}} \circ \left(\bigcirc_{i=1}^k f^\ell_{sm}\right), \\ \mathrm{VC}^1_{k_0,k} &:= \bigcirc_{i=1}^{k_0} f^1_{sm}. \end{aligned}$$

最后，$\mathrm{MG}_{k_0,k}^m$ 定义为：

$$\mathrm{MG}_{k_0,k}^m = f_{\mathrm{out}} \circ \left(\bigcirc_{i=1}^m \mathrm{VC}_{k_0,k}^L \right) \circ f_{\mathrm{in}}.$$

显然，上述定义的函数是连续的，因此，引理 20 规定了在每一层 $\ell = 1, \ldots, L$ 上对多重网格算法的各个部分（f_{sm}^ℓ、$f^\ell{}_{\mathrm{prol}}$、$f^\ell{}_{\mathrm{rest}}$、$f^\ell{}_{\mathrm{resi}}$、$f^L{}_{\mathrm{in}}$、f_{out}^L）进行近似时所需达到的精度，从而对于由 $M > 0$界定的输入，最终近似精度可达到 $\delta > 0$。

在每一层 $\ell = 1, \ldots, L$上，我们使用定理 18 推导出满足这些要求的 $f^\ell{}_{\mathrm{sm}}$和 $f^\ell{}_{\mathrm{resi}}$的近似，按照备注19推导出 f_{prol}^ℓ和 $f^\ell{}_{\mathrm{rest}}$的近似，并使用引理 15 推导出 f_{rest}^ℓ的近似。 f_{in}^L的近似使用引理 15 推导，而 $f^L{}_{\mathrm{out}}$则可以简单地用单个卷积表示。将各个卷积神经网络拼接起来，可得到一个架构类似于多个拼接的 U-Net 的卷积神经网络 Ψ，因此

$$\left\| \Psi - \mathrm{MG}_{k_0,k}^m \right\|_{L^\infty([-M,M]^{3\times n})} \leq \delta.$$

有限维情形下$\|\cdot\|_{L^\infty([-M,M]^{3\times n})}$ 与$\|\cdot\|_{H^1}$ 的范数等价性直接推出，只要适当选择 δ，不等式 (i) 即成立。参数界 (ii) 则由 Ψ的构造得出。

■

推论 8 的证明 固定阻尼参数 $\omega > 0$ 和 $k \in \mathbb{N}$，使得 $\mu(k) < 1$，并令 $m \in \mathbb{N}$ 取满足以下条件的最小值：

$$m \geq \log\left(\frac{1}{\mu(k)}\right)^{-1} \log\left(\frac{2C_{H^1}^2}{\varepsilon}\right).$$

给定 $k_0 \in \mathbb{N}$，使得最粗网格上进行 k_0 次阻尼 Richardson 迭代的收缩因子小于 $\mu(k)$。令 Ψ 为定理 6 中取$M =$的卷积神经网络，其中 $\max\left\{\sup_{\mathbf{y}\in\Gamma} \|\boldsymbol{\kappa}_{\mathbf{y}}\|_\infty, \|\mathbf{f}\|_\infty\right\}$，使得对所有 $\boldsymbol{\kappa}, \mathbf{f} \in [-M, M]^{n^2}$

$$\left\| \Psi(\mathbf{0}, \boldsymbol{\kappa}, \mathbf{f}) - \mathrm{MG}_{k_0,k}^m(\mathbf{0}, \boldsymbol{\kappa}, \mathbf{f}) \right\|_{H^1} \leq \frac{1}{2}\varepsilon \|f\|_*.$$

利用定理 3 中多重网格算法的压缩性质，可得

$$\begin{aligned} \left\| \mathrm{MG}_{k_0,k}^m(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_h(\mathbf{y}) \right\|_{A_{\mathbf{y}}} &\leq \mu(k)^m \|\mathbf{v}_h(\mathbf{y})\|_{A_{\mathbf{y}}} \\ &\leq \frac{1}{2}\varepsilon C_{H^1}^{-1} \|f\|_*. \end{aligned}$$

由此得到

$$\begin{aligned} &\|\Psi(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_h(\mathbf{y})\|_{H^1} \\ &\leq \left\| \Psi(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathrm{MG}_{k_0,k}^m(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) \right\|_{H^1} + \left\| \mathrm{MG}_{k_0,k}^m(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_h(\mathbf{y}) \right\|_{H^1} \\ &\leq \frac{1}{2}\varepsilon \|f\|_* + C_{H^1} \left\| \mathrm{MG}_{k_0,k}^m(\mathbf{0}, \boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_h(\mathbf{y}) \right\|_{A_{\mathbf{y}}} \\ &\leq \varepsilon \|f\|_*. \end{aligned}$$

假设 $\varepsilon \leq 1$，存在一个常数 $C > 0$，使得 $m \leq C\log\left(\frac{1}{\varepsilon}\right)$。结合定理 6 中的参数数量，证明至此完成。 ∎

推论 9 的证明 本证明的结构如下。我们考虑一种由 V 循环构成的多层结构，以近似细网格解，并推导该近似的误差估计。与推论 8 类似，我们说明规模足够大的卷积神经网络如何表示这一算法，并将误差估计转化为网络近似的误差估计。

为此，我们引入一些常数，并固定在每个网格层级上执行的 V 循环迭代次数：令 $c := \max\{C_{H^1}, C_{\mathrm{corr}}\}$。固定 $k \in \mathbb{N}$，使得 $\mu(k) < 1$，并设$m_1, \ldots, m_L \in \mathbb{N}$，使得

$$m_L \geq \log\left(\frac{1}{\mu(k)}\right)^{-1}\left(\log\left(\frac{2c^2L}{\varepsilon}\right) - L\log(2)\right),$$
$$m_\ell \geq \log\left(\frac{1}{\mu(k)}\right)^{-1}\log(2), \text{ for } \ell = 1, \ldots, L-1.$$

对于任意 $\mathbf{y} \in \Gamma$，我们用 $A_{\mathbf{y}}^\ell \in \mathbb{R}^{\dim V_\ell \times \dim V_\ell}$ 表示离散算子 (4)，用 $\mathbf{f}_\ell$ 表示问题 2.1 在层级 $\ell = 1, \ldots, L$ 上的右端项，使得 $A_{\mathbf{y}}^\ell \mathbf{v}_\ell(\mathbf{y}) = \mathbf{f}_\ell$。此外，为简化记号，我们对定理 6 的证明中引入的函数作相应调整，并令

(i) 对每个层级 $\ell \in 1, \ldots, L$, $\boldsymbol{\kappa}_{\mathbf{y}}^\ell \in \mathbb{R}^{6\times \dim V_\ell}$ 作为六通道张量

$$\overline{\boldsymbol{\kappa}}_{\mathbf{y}}^\ell := \left[\Upsilon(\kappa_{\mathbf{y}}, \mathcal{T}^\ell, 1), \ldots, \Upsilon(\kappa_{\mathbf{y}}, \mathcal{T}^\ell, 6)\right].$$

(ii) 函数 $\widetilde{\mathrm{MG}}_{k_0}^{m_\ell,\,k}\colon \mathbb{R}^{8\times \dim V} \to \mathbb{R}^{\dim V}$as

$$\widetilde{\mathrm{MG}}_{k_0,k}^{m_\ell} := f_{\mathrm{out}}^\ell \circ \left(\bigcirc_{i=1}^{m_\ell} \mathrm{VC}_{k_0,k}^\ell\right),$$

使得 $\widetilde{\mathrm{MG}}_{k_0}^{m_\ell,\,k} \circ f_{\mathrm{in}}^\ell = \bigcirc_{i=1}^{m_\ell} \mathrm{MG}_{k_0}^{m_\ell,\,k}$。

我们概述以下使用多重网格求解问题 2.1 的多层过程，其中$\mathbf{y} \in \Gamma$ 和 $\boldsymbol{\kappa}_{\mathbf{y}} \in \mathbb{R}^{\dim V_L}$ 作为输入给定：

- $\boldsymbol{\kappa}_{\mathbf{y}}$ **子采样**。使用 $\boldsymbol{\kappa}_{\mathbf{y}}$，按照引理 15 (i) 中的说明计算 $\boldsymbol{\kappa}^L{}_{\mathbf{y}}$。随后，按照引理 15 (ii) 中的说明，对每个层级 $\ell = 2, \ldots, L-1$ 计算 $\boldsymbol{\kappa}^\ell{}_{\mathbf{y}}$。
- $\boldsymbol{\ell = 1}$. 使用 m_1 次多重网格循环迭代求解 $\mathbf{v}_1(\mathbf{y})$。令

$$\tilde{\mathbf{v}}_1(\mathbf{y}) := \widetilde{\mathrm{MG}}_{k_0,k}^{m_1}(0, \overline{\boldsymbol{\kappa}}_{\mathbf{y}}^1, \mathbf{f}).$$

- $\boldsymbol{\ell = 2, \ldots, L}$. 虽然 $\hat{\mathbf{v}}_\ell(\mathbf{y})$ 是相对于伽辽金近似 $\mathbf{v}_{\ell-1}(\mathbf{y})$ 的真实修正，但我们将 $\check{\mathbf{v}}_\ell(\mathbf{y})$ 定义为相对于 $\tilde{\mathbf{v}}_{\ell-1}(\mathbf{y})$ 的修正。为此，令

$$\check{\mathbf{v}}_\ell(\mathbf{y}) := \mathbf{v}_\ell(\mathbf{y}) - P_\ell \tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}),$$
$$\tilde{\mathbf{f}}_\ell := \mathbf{f}_\ell - A_{\mathbf{y}}^\ell P_\ell \tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}) = A_{\mathbf{y}}^\ell \check{\mathbf{v}}_\ell(\mathbf{y}).$$

使用多重网格 V 循环进行 m_ℓ 次迭代，求解 $A_{\mathbf{y}}^\ell \check{\mathbf{v}}_\ell(\mathbf{y}) = \tilde{\mathbf{f}}_\ell$，并设

$$\tilde{\mathbf{v}}_\ell(\mathbf{y}) := \widetilde{\mathrm{MG}}_{k_0,k}^{m_\ell}(0, \overline{\boldsymbol{\kappa}}_{\mathbf{y}}^\ell, \tilde{\mathbf{f}}_\ell) + P_\ell \tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}).$$

This 该过程利用定理中的压缩得到以下估计 3.

- $\boldsymbol{\ell = 1}$. 利用定理 3 中的压缩性质以及 $m_1 \geq -\log(2)/\log(\mu(k))$，可得

$$\|\tilde{\mathbf{v}}_1(\mathbf{y}) - \mathbf{v}_1(\mathbf{y})\|_{A^1_{\mathbf{y}}} \leq \frac{1}{2} c \|f\|_* .$$

- $\boldsymbol{\ell = 2, \ldots, L-1}$. 再次利用定理 3，以及 m_ℓ，$\tilde{}\mathbf{v}_\ell$，$\check{}\mathbf{v}_\ell$ 和 $\hat{}\mathbf{v}_\ell = \mathbf{v}_\ell - P_\ell \mathbf{v}_{\ell-1}$ 的定义，可得

$$\begin{aligned}
\|\tilde{\mathbf{v}}_\ell(\mathbf{y}) - \mathbf{v}_\ell(\mathbf{y})\|_{A^\ell_{\mathbf{y}}} &= \left\| \widetilde{\mathrm{MG}}^{m_\ell}_{k_0,k}(0, \overline{\boldsymbol{\kappa}}^\ell_{\mathbf{y}}, \tilde{\mathbf{f}}_\ell) + P_\ell \tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}) - \check{\mathbf{v}}_\ell(\mathbf{y}) - P_\ell \tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}) \right\|_{A^\ell_{\mathbf{y}}} \\
&= \left\| \widetilde{\mathrm{MG}}^{m_\ell}_{k_0,k}(0, \overline{\boldsymbol{\kappa}}^\ell_{\mathbf{y}}, \tilde{\mathbf{f}}_\ell) - \check{\mathbf{v}}_\ell(\mathbf{y}) \right\|_{A^\ell_{\mathbf{y}}} \\
&\leq \frac{1}{2} \|\check{\mathbf{v}}_\ell(\mathbf{y})\|_{A^\ell_{\mathbf{y}}} \\
&\leq \frac{1}{2} \|\hat{\mathbf{v}}_\ell(\mathbf{y})\|_{A^\ell_{\mathbf{y}}} + \frac{1}{2} \|P_\ell \tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}) - P_\ell \mathbf{v}_{\ell-1}(\mathbf{y})\|_{A^\ell_{\mathbf{y}}} \\
&\leq \frac{1}{2} c 2^{-\ell} \|f\|_* + \frac{1}{2} \|\tilde{\mathbf{v}}_{\ell-1}(\mathbf{y}) - \mathbf{v}_{\ell-1}(\mathbf{y})\|_{A^{\ell-1}_{\mathbf{y}}} .
\end{aligned}$$

  这里，$\hat{}\mathbf{v}_\ell(\mathbf{y})$ 的界由公式 (10) 得出。

- $\boldsymbol{\ell = L}$. 按照处理 $\ell = 2, \ldots, L-1$ 情形的相同论证，结合 (9) 中的范数不等式和 m_L 的定义，可得最细层上 H^1-距离的如下估计。

$$\begin{aligned}
&\|\tilde{\mathbf{v}}_L(\mathbf{y}) - \mathbf{v}_L(\mathbf{y})\|_{H^1} \\
&\leq c \|\tilde{\mathbf{v}}_L(\mathbf{y}) - \mathbf{v}_L(\mathbf{y})\|_{A^L_{\mathbf{y}}} \\
&= c \left\| \widetilde{\mathrm{MG}}^{m_L}_{k_0,k}(0, \overline{\boldsymbol{\kappa}}^L_{\mathbf{y}}, \tilde{\mathbf{f}}_L) - \check{\mathbf{v}}_L(\mathbf{y}) \right\|_{A^L_{\mathbf{y}}} \\
&\leq \frac{1}{2cL} 2^L \varepsilon \|\check{\mathbf{v}}_L(\mathbf{y})\|_{A^L_{\mathbf{y}}} \\
&\leq \frac{1}{cL} 2^L \varepsilon \left(\frac{1}{2} \|\hat{\mathbf{v}}_L(\mathbf{y})\|_{A^L_{\mathbf{y}}} + \frac{1}{2} \|P_L \tilde{\mathbf{v}}_{L-1}(\mathbf{y}) - P_L \mathbf{v}_{L-1}(\mathbf{y})\|_{A^L_{\mathbf{y}}} \right) \\
&\leq \frac{1}{cL} 2^L \varepsilon \left(\frac{1}{2} c 2^{-L} \|f\|_* + \frac{1}{2} \|\tilde{\mathbf{v}}_{L-1}(\mathbf{y}) - \mathbf{v}_{L-1}(\mathbf{y})\|_{A^{L-1}_{\mathbf{y}}} \right)
\end{aligned}$$

  递归地可得

$$\|\tilde{\mathbf{v}}_L(\mathbf{y}) - \mathbf{v}_L(\mathbf{y})\|_{H^1} \leq \frac{1}{cL} 2^L \varepsilon \sum_{\ell=1}^{L} \left(\prod_{n=\ell}^{L} \frac{1}{2} \right) 2^{-\ell} c \|f\|_* = \frac{1}{2} \varepsilon \|f\|_* .$$

总的来说，该过程由下采样操作、多重网格 V 循环以及中间的上采样操作组成，所有这些操作都可以由卷积神经网络任意精确地逼近：扩散率场的下采样按照引理 15 实现，V 循环按照定理 6 所示实现，而用于上采样的延拓算子

则按照备注19实现。因此，该过程由有限个函数拼接而成，每个函数都可以由卷积神经网络以任意精度表示。利用引理 20（其中 $M = M\text{and}\,\varepsilon = \frac{1}{2}\varepsilon\|f\|_*$），可知存在一个卷积神经网络 Ψ，使得对于所有 $\mathbf{y} \in \Gamma$ 以及所有右端项 $\mathbf{f}$，都有

$$\begin{aligned}\|\Psi(\boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \mathbf{v}_L(\mathbf{y})\|_{H^1} &\leq \|\Psi(\boldsymbol{\kappa}_{\mathbf{y}}, \mathbf{f}) - \tilde{\mathbf{v}}_L(\mathbf{y})\|_{H^1} + \|\tilde{\mathbf{v}}_L(\mathbf{y}) - \mathbf{v}_L(\mathbf{y})\|_{H^1} \\ &\leq \varepsilon \|f\|_* .\end{aligned}$$

所提出的多层算法的结构以及定理 6、引理 15 和备注19中的参数界意味着 Ψ 是一个 ML-Net 架构，并且存在一个常数 $C > 0$，满足

$$\begin{aligned}M(\Psi) &\leq C \sum_{\ell=1}^{L-1} \ell + C \left(\log\left(\frac{2L}{\varepsilon}\right) - L\log(2) \right) L \\ &\leq CL \left(L + \log\left(\frac{1}{\varepsilon}\right) + \log(2L) - L\log(2) \right) \\ &\leq CL \log\left(\frac{1}{\varepsilon}\right) + CL^2.\end{aligned}$$

■