PAPER • **OPEN ACCESS**

Imaging conductivity from current density magnitude using neural networks*

To cite this article: Bangti Jin *et al* 2022 *Inverse Problems* **38** 075003

OPEN ACCESS
IOP Publishing
Inverse Problems
Inverse Problems **38** (2022) 075003 (36pp)
https://doi.org/10.1088/1361-6420/ac6d03

Imaging conductivity from current density magnitude using neural networks*

Bangti Jin[1,**], **Xiyao Li**[1] **and Xiliang Lu**[2]

[1] Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, United Kingdom
[2] School of Mathematics and Statistics, and Hubei Key Laboratory of Computational Science, Wuhan University, Wuhan 430072, People's Republic of China

E-mail: b.jin@ucl.ac.uk, bangti.jin@gmail.com, xiyao.li.20@ucl.ac.uk and xllv.math@whu.edu.cn

Received 10 January 2022, revised 5 April 2022
Accepted for publication 5 May 2022
Published 8 June 2022

Abstract

Conductivity imaging represents one of the most important tasks in medical imaging. In this work we develop a neural network based reconstruction technique for imaging the conductivity from the magnitude of the internal current density. It is achieved by formulating the problem as a relaxed weighted least-gradient problem, and then approximating its minimizer by standard fully connected feedforward neural networks. We derive bounds on two components of the generalization error, i.e., approximation error and statistical error, explicitly in terms of properties of the neural networks (e.g., depth, total number of parameters, and the bound of the network parameters). We illustrate the performance and distinct features of the approach on several numerical experiments. Numerically, it is observed that the approach enjoys remarkable robustness with respect to the presence of data noise.

Keywords: conductivity imaging, current density imaging, neural network, generalization error

(Some figures may appear in colour only in the online journal)

*The work of B Jin is supported by UK EPSRC Grant EP/T000864/1, and that of X Lu by the National Science Foundation of China (No. 11871385).
**Author to whom any correspondence should be addressed.

1. Introduction

The conductivity value varies widely with soft tissue types [24, 52] and its accurate imaging can provide valuable information about the physiological and pathological conditions of tissue. This underpins several important medical imaging modalities [2, 13, 70]. For example, electrical impedance tomography (EIT) [13] aims at recovering the interior conductivity distribution from given pairs of flux/voltage on the object's boundary. However, it is severely ill-posed, which makes it very challenging to develop a stable numerical algorithm to accurately reconstruct the conductivity [13]. Especially, the attainable resolution of the reconstruction is fairly limited. To lessen the inherent degree of ill-posedness, researchers have proposed several novel conductivity imaging modalities, e.g., magnetic resonance electrical impedance tomography/current density impedance imaging (CDII), impedance-acoustic tomography, acousto-electric tomography and magneto-acoustic tomography with magnetic induction. All these imaging modalities employ internal data that are derived from other modalities (hence the term coupled-physics imaging). See the reviews [5, 70] for extensive discussions on the mathematical model and the mathematical theory, respectively. The availability of internal data promises reconstructions with much improved resolution.

In this work we focus on CDII [53]. Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be an open bounded Lipschitz domain modeling the conducting body with a boundary $\partial\Omega$. The relation between the voltage u and the conductivity σ is described by

$$\begin{cases} -\nabla \cdot (\sigma \nabla u) = 0, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega, \end{cases} \tag{1.1}$$

where g is the applied boundary voltage. In CDII, the current density J is given by $J(x) = -\sigma\nabla u(x)$, for $x \in \Omega$. In practice, one employs an MRI scanner to capture the internal magnetic flux density data $\vec{b}$ induced by an externally injected current [25, 31, 36, 61] and then obtains the current density J according to Ampere's law $J = \mu_0^{-1}\nabla \times \vec{b}$, where μ_0 is the magnetic permeability. This requires measuring all components of the magnetic flux $\vec{b}$, which may be challenging in practice, as it requires a rotation of the object being imaged or of the MRI scanner. CDII aims at recovering the conductivity σ from $a(x) \equiv |J(x)|$ in Ω, the magnitude of one current density field.

CDII has been intensively studied in the past decade, and a number of important theoretical results have been obtained. Nachman *et al* [53] established the uniqueness of the recovery from one internal measurement a together with Cauchy data on a part of the object's boundary. Later, the uniqueness was shown also for anisotropic conductivity with a known conformal class [28]. The Hölder conditional stability for the nonlinear inverse problem of recovering the conductivity distribution σ from one internal measurement was proved in [48]. The case of partial data (i.e., a partial knowledge of one current density field generated inside a body) has also been proved [49]. The conditional stability of the inverse problem under fairly general assumptions was shown in [42].

The development of novel reconstruction algorithms has also received much attention. One popular algorithm is an iterative method to solve the weighted least-gradient formulation [54], which iteratively solves a well-posed direct problem, and the authors proved that the sequence of iterates converges; see section 2.1 for more details about the derivation. It has been extended to other scenarios, e.g., complete electrode model [55]. An alternative approach is based on the level set [53, 67]. A linearized reconstruction technique was developed recently in [73]. The

more conventional output least-squares formulation has not been employed for CDII reconstruction, but it applies more or less directly (see [1, 41] for conductivity imaging from related internal data, and [29] for iterative reconstruction).

In this work, we develop a new numerical method for the recovery of the conductivity σ from the current density magnitude $a(x)$. It is based on the weighted least-gradient reformulation of the inverse problem, which has inspired the iterative algorithm in [54]. Instead of solving the variational problem iteratively, we solve a relaxed version of the problem directly using neural networks. The approach is flexible with domain geometry and problem data, and capitalizes directly on recent algorithmic innovations in machine learning, e.g., stochastic optimization [14] and automatic differentiation [11]. The numerical results in section 4 clearly demonstrate the significant potential of the approach: it enjoys remarkable robustness with respect to the presence of a large amount of data noise. Further, we provide a preliminary analysis of the neural network approximation to the relaxed least-gradient problem, in terms of the approximation and statistical errors. The main tools in the analysis include approximation theory of neural networks [26] and Rademacher complexity from statistical learning theory [63]. The analysis sheds light into the choice of several important algorithmic parameters, e.g., network width and depth, and the number of sampling points in the domain and on the boundary.

In recent years, the use of deep neural networks (DNNs) for solving PDEs has received much attention, and several different methods have been developed; see the review [21] for a recent overview on various ways of using neural networks for different classes of PDEs and a fairly extensive list of relevant references. One notable idea is to utilize neural networks to approximate solutions of PDEs directly, which can be traced back to the 1990s [18, 40]. Notable recent developments include physics-informed neural networks [58], deep Galerkin method [64] and deep Ritz method [22] etc. The first two methods are based on least-squares type residual minimization for solving PDEs. The deep Ritz method is based on the Ritz variational formulation of the elliptic problem. This work adopts a deep Ritz method to the weighted least-gradient problem arising in CDII. Despite the great empirical successes of these methods, rigorous numerical analysis of neural network based PDE solvers remains very challenging and is still in its infancy [19, 30, 34, 35, 43, 44, 71]. The important works [30, 43, 44, 71] derived *a priori* error bounds on the approximations obtained by two-layer neural networks under suitable regularity conditions on the solutions, whereas the work [34] studied DNNs for standard second-order elliptic PDEs with Robin boundary conditions. The present work extends the analysis in [34] to the weighted least-gradient problem arising in CDII.

Very recently, the use of DNNs for solving PDE inverse problems also started to receive attention, and existing methods can roughly be divided into two groups: supervised [27, 37, 62] and unsupervised [6, 7, 57, 72]. The methods in the former group rely on the availability of paired training data, and are essentially concerned with learning the forward operators or its (regularized) inverses, and the methods in the latter group exploit essentially the extraordinary expressivity as universal function approximators. Khoo and Ying [37] proposed a novel neural network architecture, SwitchNet, for solving the wave equation based inverse scattering problems via constructing maps between the scatterers and the scattered field using training data. Seo *et al* [62] developed a supervised approach for the solution of nonlinear inverse problems using a low dimensional manifold for the solution approximation, converting it into a well-posed one using variational autoencoder, and demonstrated the idea on time difference EIT. Guo and Jiang [27] developed a neural network analogue for the direct sampling method for EIT. The works [6, 7] investigated image reconstruction in the classical EIT problem, using the weak formulation and the least-squares formulation (but also with the L^∞ norm consistency), respectively. Pakravan *et al* [57] developed a hybrid approach, aiming at blending high expressivity of DNNs with the accuracy and reliability of traditional

numerical methods for PDEs, and showed the approach for recovering the variable diffusion coefficient in one- and two-dimensional elliptic PDEs. All these works have presented very encouraging empirical results for a range of PDE inverse problems, and clearly demonstrated the significant potentials of DNNs in solving PDE inverse problems. The approach proposed in this work belongs to the second group, but unlike the existing approaches, it does not directly approximate the unknown conductivity σ and thus differs substantially from existing approaches.

The rest of the paper is organized as follows. In section 2 we develop a neural network based approach for imaging the conductivity. Then in section 3 we provide an analysis of the neural network based approach, and derive a convergence rate for the neural network approximation in terms of properties of the neural network, e.g., the activation function, depth, number of parameters, and parameter bound. In section 4, we present extensive numerical experiments to show its performance and the impact of various algorithmic parameters on the reconstruction error (number of training points, network parameters and noise levels), and also present a comparative study of the approach with an existing iterative reconstruction approach [54].

2. Reconstruction algorithm

In this section, we describe the proposed imaging algorithm. It is essentially a neural network discretization of a relaxation of the variational formulation proposed by Nachman *et al* [54]. A preliminary analysis of the neural network approximation is given in section 3.

2.1. Variational formulation

First we briefly recall a variational formulation from [54]. By representing $\sigma = \frac{a}{|\nabla u|}$ in accordance with Ohm's law, problem (1.1) can be recast into the following Dirichlet problem for the weighted one-Laplacian

$$\begin{cases} \nabla \cdot \left(a \dfrac{\nabla u}{|\nabla u|} \right) = 0, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \tag{2.1}$$

This was originally proposed by Kim *et al* [38], who also showed nonuniqueness of the solution when the problem is equipped with a Neumann boundary condition. Formulation (2.1) was utilized by work [53] for recovering the conductivity σ from Cauchy data on a part of the boundary (along with the interior data) on a two-dimensional domain. Due to the singularity and elliptic degeneracy of the differential operator, the concept of a solution requires some care. Therefore, as a mathematical model of CDII, Nachman *et al* [54] employed the following weighted least gradient (Dirichlet) problem

$$\min_{u \in W^{1,1}(\Omega) \cap C(\overline{\Omega}), Tu = g} \left\{ \mathcal{J}(u) = \int_\Omega a |\nabla u| \mathrm{d}x \right\}, \tag{2.2}$$

where T is the trace operator, i.e., $Tu = u|_{\partial\Omega}$. The equivalence can be seen by computing the Euler–Lagrange equation of the functional $\mathcal{J}$ and observing that it formally satisfies problem (2.1). It was proved in [54, theorem 1.3] that if $g \in C^{1,\nu}(\partial\Omega)$, $a \in C^\nu(\overline{\Omega})$, $\nu \in (0,1)$, and $a > 0$ a.e. in Ω, and the data (g, a) are *admissible* (i.e., there exists a conductivity σ that is essentially bounded and bounded away from zero such that if $u \in H^1(\Omega)$ is a weak solution to problem (1.1) then $a = \sigma|\nabla u|$), then problem (2.2) is uniquely solvable in $W^{1,1}(\Omega) \cap C(\overline{\Omega})$ and $\sigma = \frac{a}{|\nabla u|}$ is Hölder continuous. It was also shown that problem (2.1) is, formally, the Euler–Lagrange

equation of the functional $\mathcal{J}(u)$ in (2.2), and that the solution of (2.2) is a weak solution to (2.1).

From the point of view of calculus of variation, the space $W^{1,1}(\Omega)$ is not the most convenient choice for studying problem (2.2) [56]. Indeed, the minimizing sequences stay bounded in $W^{1,1}(\Omega)$. However, due to its non-reflexivity, $\mathcal{J}$ is no longer weakly lower semicontinuous in $W^{1,1}(\Omega)$ (since $L^1_{\rm loc}(\Omega)$ limits of functions in $W^{1,1}(\Omega)$ may no longer belong to $W^{1,1}(\Omega)$). Thus, it is natural to extend $\mathcal{J}(u)$ in (2.2) to the space $BV(\Omega)$ of functions of bounded variation, which preserves the lower-semicontinuity. These considerations naturally lead to the study of the following weighted least-gradient problems in the space $BV(\Omega)$ [56]

$$\min_{u\in BV(\Omega),Tu=g}\left\{\mathcal{J}(u)=\int_\Omega a(x)|Du|\right\}, \tag{2.3}$$

where the distributional derivative Du is a signed Radon measure that can be decomposed into its absolutely continuous and singular parts as $Du = D^a u + D^s u$, with $D^a u = \nabla u \mathcal{L}^d$, where ∇u is the Radon–Nikodym derivative of the measure Du with respect to the Lebesgure measure $\mathcal{L}^d$, and $D^s u$ denotes the singular part. The existence and uniqueness results of problem (2.3) were established for either the case $a \in C^{1,1}(\overline{\Omega})$, $g \in C(\partial\Omega)$ [33] or the case $a \in C(\overline{\Omega})$, $a \geqslant 0$, and that the pair (g, a) is admissible [50].

Once a minimizer u to problem (2.3) is found, the conductivity σ can be recovered by $\sigma = \frac{a}{|\nabla u|}$, following the definition of the current density magnitude a. These observations and the convexity of the energy functional $\mathcal{J}$ motivated several algorithms for recovering the conductivity σ [51, 54]. Nachman *et al* [54] developed an iterative procedure for minimizing problem (2.2) and then recovering the conductivity σ. Specifically, given an initial guess σ^0, they proposed to repeat the following two steps alternatingly

(a) Solve for u^n from the second-order elliptic PDE

$$\begin{cases} -\nabla\cdot(\sigma^n\nabla u^n) = 0, & \text{in } \Omega,\\ u^n = g, & \text{on } \partial\Omega. \end{cases}$$

(b) Update the conductivity σ by $\sigma^{n+1} = \frac{a}{|\nabla u^n|}$.

The authors proved the convergence of the sequence $\{u^n\}_{n=1}^\infty$ to the minimizer of functional $\mathcal{J}$ in $H^1(\Omega)$ for admissible pairs (g, a) [54, proposition 4.4]. This algorithm is appealing since it is easy to implement, and converges within tens of iteration. The main cost is to solve one elliptic PDE at each iteration. It will be employed as the baseline algorithm in the numerical experiments. Note that the algorithm does not incorporate regularization explicitly [32]. Due to the ill-posedness, in the presence of data noise, early stopping is needed in order to obtain satisfactory reconstructions. However, the issue of early stopping has not been studied so far for the algorithm.

2.2. Proposed algorithm

In this work, we take a slightly different route. Instead of iterative update, we propose to solve the minimization problem (2.3) directly by using neural networks to approximate the minimizer u (with parameter θ), and then to recover the conductivity σ using the defining relation $\sigma = \frac{a}{|\nabla u|}$ from Ohm's law. More specifically, we proceed in the following two steps:

(a) Find a neural network approximation u_θ to problem (2.3) by minimizing a suitable loss.
(b) Recover the conductivity σ by $\sigma = \frac{a}{|\nabla u_\theta|}$.

The crucial step to realize the algorithm numerically is to solve (2.3) stably. This is nontrivial due to nonsmoothness of the functional $\mathcal{J}$. Further, the imposition of the essential boundary condition $Tu = g$ is nontrivial, due to the nonlocality of neural networks. For special geometries, one may construct neural networks that satisfy the boundary condition exactly, but generally this is challenging. Thus, we employ an alternative formulation of problem (2.3) from [45] (see also [16, 47]), using the concept of the space of total variation with respect to an anisotropy defined below. Throughout we make the following assumption on the data a, which is also known as the continuity and coercivity of the metric integrand.

Assumption 2.1. $a \in C(\overline{\Omega})$, and there exist constants $\alpha_0, \alpha_1 > 0$ with $\alpha_1 > \alpha_0$ such that $\alpha_0 \leqslant a \leqslant \alpha_1$ in Ω.

Now we recall the space $BV_a(\Omega)$ [16, 47]. Clearly when $a(x) \equiv 1$ in Ω, it recovers the standard space $BV(\Omega)$ of functions of bounded variation.

Definition 2.1. Let $u \in L^1(\Omega)$. Then the a-total variation of u in Ω is defined as

$$\int_\Omega |Du|_a := \sup_{\varphi \in K_a(\Omega)} \int_\Omega u \nabla \cdot \varphi, \quad \text{with} \quad K_a(\Omega) = \{\varphi \in C_0^1(\Omega; \mathbb{R}^d) : |\varphi(x)| \leqslant a(x) \text{ in } \Omega\},$$

and let

$$BV_a(\Omega) = \left\{ u \in L^1(\Omega) : \int_\Omega |Du|_a < \infty \right\},$$

which is a Banach space when endowed with the norm

$$\|u\|_{BV_a(\Omega)} = \|u\|_{L^1(\Omega)} + \int_\Omega |Du|_a.$$

Note that under assumption 2.1, there hold $BV_a(\Omega) = BV(\Omega)$ in the sense of set (but endowed with different norms), and further

$$\alpha_0 \int_\Omega |Du| \leqslant \int_\Omega |Du|_a \leqslant \alpha_1 \int_\Omega |Du|.$$

Given a function $g \in L^1(\partial\Omega)$, problem (2.2) can be equivalently written as

$$\min \mathcal{J}_g(u) = \begin{cases} \int_\Omega a|\nabla u| \mathrm{d}x, & \text{if } u \in W^{1,1}(\Omega), Tu = g, \\ +\infty, & \text{otherwise.} \end{cases}$$

In [47, theorem 4] (see also [16, theorem 3.6] and [45, proposition 3.1]), it was proved that the functional $\mathcal{J}_g$ admits the following relaxation to $L^{\frac{d}{d-1}}(\Omega)$

$$\mathcal{L}(u) = \begin{cases} \int_\Omega |Du|_a + \int_{\partial\Omega} a|Tu - g| \mathrm{d}s, & u \in BV_a(\Omega), \\ +\infty, & u \in L^{\frac{d}{d-1}}(\Omega) \setminus BV_a(\Omega), \end{cases} \tag{2.4}$$

in the following sense

$$\mathcal{L}(u) = \inf \left\{ \liminf_{n\to\infty} \mathcal{J}_g(u_n) : u_n \to u \text{ in } L^1(\Omega),\ u_n \in W^{1,1}(\Omega), Tu_n = g \right\}.$$

Therefore, for every $u \in BV_a(\Omega)$, there exists a sequence $\{u_n\}_{n=1}^{\infty} \subset W^{1,1}(\Omega)$ with $Tu_n = g$ such that $u_n \to u$ in $L^1(\Omega)$ and

$$\int_\Omega a(x)|\nabla u_n(x)|\mathrm{d}x \to \mathcal{L}(u).$$

In particular, this implies the functional $\mathcal{L}$ is weakly lower semicontinuous, which automatically guarantees the existence of a minimizer.

The relaxed functional $\mathcal{L}$ is convex and weakly lower semicontinuous in $L^{\frac{d}{d-1}}(\Omega)$. Furthermore, we have the following results which connect the relaxed functional (2.4) to problem (2.3) (see [45, definition 3.4] for the precise definition of a solution u to problem (2.1)). Thus, under certain conditions, the solution of (2.4) coincides with that of (2.3).

Theorem 2.1. *Under assumption 2.1, for each $g \in L^1(\Omega)$, there exists a solution u to problem (2.1). Further, for $u \in BV_a(\Omega)$ satisfying $Tu = g$, the following three statements are equivalent.*

(*a*) *u is a solution of problem (2.1).*
(*b*) *u is a function of the weighted least gradient in Ω, i.e., solves problem (2.3).*
(*c*) *$\mathcal{L}(u) \leqslant \mathcal{L}(v)$ for all $v \in BV_a(\Omega)$.*

Proof. Note that assumption 2.1 implies that the metric integrand $\phi(x, \xi) = a(x)|\xi|$ is continuous and coercive in Ω. The first statement can be found in [45, theorem 3.6], and the equivalence statements are taken from [45, corollary 3.9]. □

In practice, it is beneficial to introduce a weighing parameter $\gamma > 1$ to the boundary integral

$$\mathcal{L}_\gamma(u) = \int_\Omega a|Du| + \gamma \int_{\partial\Omega} a|Tu - g|\mathrm{d}s. \tag{2.5}$$

Formally, it can be viewed as a nonstandard penalized formulation to impose the boundary condition only weakly, and this idea is widely used in the context of finite element methods [4]. However, the existence of a minimizer u^* in $BV(\Omega)$ is generally unclear, since the trace operator in $BV(\Omega)$ is not continuous with respect to the weak star convergence in $BV(\Omega)$. The existence will be assumed for the analysis below in section 3.

Remark 2.1. There are alternative penalized formulations that ensure the existence of a minimizer:

$$\mathcal{L}_{\gamma,\epsilon}(u) = \int_\Omega a|\nabla u(x)|\mathrm{d}x + \frac{\gamma}{2}\int_{\partial\Omega} |Tu - g|^2\mathrm{d}s + \frac{\epsilon}{2}\int_\Omega |\nabla u|^2\mathrm{d}x,$$

with small $\epsilon > 0$. This formulation was studied in [66]. The neural network approach described below can be extended directly and the analysis also holds upon minor changes.

2.3. Discretization via neural networks

Now we describe the discretization of problem (2.5) via neural networks. We employ the standard fully connected feedforward neural networks, in which each neuron is connected to neurons in the successive layer by an affine-linear map, and then followed by a nonlinear activation function; see figure 1 for a schematic illustration of a three-layer neural network. An L-layer feedforward neural network consists of $(L-1)$ hidden layers, and maps a given input

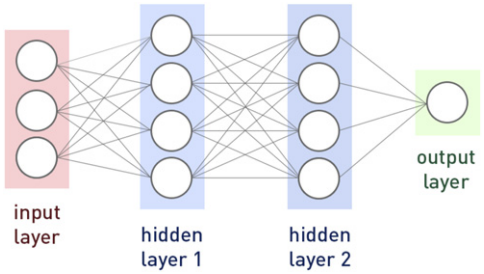


Figure 1. A schematic illustration of a three-layer feedforward neural network.

$x \in \mathbb{R}^{d_0}$ to an output $y \in \mathbb{R}^{d_L}$ through compositions of affine-linear maps and a scalar nonlinear activation function $\rho : \mathbb{R} \to \mathbb{R}$, with the ℓth layer having d_ℓ neurons. The width $\mathcal{W}$ of the network is defined to be $\mathcal{W} := \max_{\ell=0,1,\ldots,L} d_\ell$. We define $\mathcal{P}_N := \prod_{\ell=1}^{L}(\mathbb{R}^{d_\ell \times d_{\ell-1}} \times \mathbb{R}^{d_\ell})$ to be the set of neural network parametrizations. For a parametrization $\theta = \{(W^{(\ell)}, b^{(\ell)})\}_{\ell=1}^{L} \in \mathcal{P}_N$ (which will be identified with a vector below), we define its realization $f^{(L)}(x)$ by

$$\begin{aligned} f^{(0)} &= x, \\ f^{(\ell)} &= \rho(W^{(\ell)} f^{(\ell-1)} + b^{(\ell)}), \quad \text{for } \ell = 1, 2, \ldots, L-1, \\ f^{(L)} &= W^{(L)} f^{(L-1)} + b^{(L)}. \end{aligned}$$

Here the nonlinear activation function $\rho : \mathbb{R} \to \mathbb{R}$ is applied componentwise to a vector, and $f^{(\ell)} \in \mathbb{R}^{d_\ell}$. $W^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$ and $b^{(\ell)} \in \mathbb{R}^{d_\ell}$ for $\ell = 1, 2, \ldots, L$ are commonly known as the weight matrix and bias vector at the ℓth layer, respectively. Note that the total number N_θ of parameters is given by $N_\theta = \sum_{\ell=1}^{L} d_\ell d_{\ell-1} + d_\ell$. Also the activation function ρ should be at least twice differentiable in order to facilitate the training process, due to the presence of one spatial derivative and one derivative with respect to the network parameter θ, which is required by gradient type algorithms. Common choices of ρ include sigmoid, tanh, rectified power unit and softplus etc, but the standard rectified linear unit (ReLU) is not suitable, due to its limited differentiability.

To solve problem (2.5), we approximate the function u with a feedforward neural network $f^{(L)}$. Thus, the input dimension d_0 is taken to be the dimension d of the domain Ω, and the output dimension d_L is taken to be 1. We denote the set of all such L-layer neural networks by $\mathcal{N} \equiv \mathcal{N}(L, N_\theta, R)$, with $R > 0$ being the maximum bound on the network parameters, i.e., all components $|W_{i,j}^{(\ell)}|, |b_i^{(\ell)}| \leqslant R$ for all i, j, ℓ (or equivalently $\|\theta\|_{\ell^\infty} \leqslant R$, with $\|\cdot\|_{\ell^\infty}$ being the Euclidean maximum norm), to explicitly indicate its dependence on the network properties (i.e. depth, total number of parameters and the bound for each parameter).

Now we derive the loss for training neural networks. Let $U(\Omega)$ and $U(\partial\Omega)$ be uniform distributions inside the domain Ω and on the boundary $\partial\Omega$, respectively. The loss (2.5) can be rewritten as

$$\mathcal{L}_\gamma(u) = |\Omega| \mathbb{E}_{U(\Omega)}[a(X)|\nabla u(X)|] + \gamma |\partial\Omega| \mathbb{E}_{U(\partial\Omega)}[a(Y)|Tu(Y) - g|],$$

where $\mathbb{E}_\mu$ denotes taking the expectation with respect to a distribution μ. This formulation is commonly known as the population loss in statistical learning theory. The empirical loss $\widehat{\mathcal{L}}_\gamma$ takes the form:

$$\widehat{\mathcal{L}}_\gamma(\theta) = \frac{|\Omega|}{n_1}\sum_{i=1}^{n_1} a(X_i)|\nabla u_\theta(X_i)| + \gamma\frac{|\partial\Omega|}{n_2}\sum_{j=1}^{n_2} a(Y_j)|Tu_\theta(Y_j) - g(Y_j)|, \tag{2.6}$$

where $u_\theta \in \mathcal{N}(L, N_\theta, R)$ is the neural network realization with parametrization θ, and $\{X_i\}_{i=1}^{n_1}$ and $\{Y_j\}_{j=1}^{n_2}$ are n_1 independent and identically distributed (i.i.d.) training samples drawn from $U(\Omega)$ and n_2 i.i.d. training samples from $U(\partial\Omega)$, respectively. The empirical loss $\widehat{\mathcal{L}}_\gamma(\theta)$ is a Monte Carlo approximation of $\mathcal{L}_\gamma$. Note that in the low-dimensional case, one may employ standard quadrature rules. Then the training process boils down to solving the following optimization problem:

$$\min_\theta \widehat{\mathcal{L}}_\gamma(\theta).$$

Note that the box constraint $\Theta = \{\theta : \|\theta\|_{\ell^\infty} \leqslant R\}$ enforces a compact set on the (finite-dimensional) neural network parameter θ, and the continuity of $\widehat{\mathcal{L}}_\gamma$ in θ (under mild conditions on ρ) ensures the existence of a global minimizer to the empirical loss $\widehat{\mathcal{L}}_\gamma$. We denote any global minimizer of the empirical loss $\widehat{\mathcal{L}}_\gamma(\theta)$ in (2.6) by θ^*, and the corresponding neural network approximation in $\mathcal{N}$ by u_{θ^*}. Note that u_{θ^*} is the neural network approximation to the minimizer u^* of the population loss $\mathcal{L}_\gamma$ in (2.5). However, the empirical loss $\widehat{\mathcal{L}}_\gamma(\theta)$ is nonconvex in the parameter θ and may be fraught with local minimizers, and thus in theory, a global minimizer can be difficult to obtain. Nonetheless, in practice, researchers have found that simple algorithms [14], e.g., (stochastic) gradient descent (SGD) [59] or ADAM [39], can perform fairly well. In practice, the empirical loss $\widehat{\mathcal{L}}_\gamma$ is optimized by one such random solver $\mathcal{A}$ (e.g., SGD and ADAM), which outputs a stochastic approximation $\theta_{\mathcal{A}}$ to the optimal θ^* and also the corresponding network $u_{\theta_{\mathcal{A}}}$.

In practical computation, the term $|\nabla u_\theta(x)|$ in the loss (2.6) requires some care, since its derivative with respect to the network parameters θ may be ill-defined when the gradient $\nabla u_\theta(x)$ vanishes. Thus we replace the term $|\nabla u_\theta(x)|$ with a smooth approximation:

$$\psi(|\nabla u_\theta|) = \begin{cases} |\nabla u_\theta|, & |\nabla u_\theta| \geqslant \zeta, \\ \dfrac{|\nabla u_\theta|^2}{2\zeta} + \dfrac{\zeta}{2}, & \text{otherwise}, \end{cases} \tag{2.7}$$

where $\zeta > 0$ is a small constant controlling the amount of smoothing. The boundary term can be treated similarly. In practice, the gradient ∇u_θ with respect to the spatial variable x is computed using automatic differentiation techniques, which are implemented in many popular platforms, e.g., tensorflow module `tf.gradients`. Thus, the overall computational technique can fully capitalize on modern algorithmic innovations, e.g., automatic differentiation [11].

3. Convergence analysis

Now we present a preliminary analysis of the neural network approximation $u_{\theta_{\mathcal{A}}}$. In the analysis, we take the domain Ω to be the unit hypercube $\Omega \subset (0,1)^d \subset \mathbb{R}^d$, and since the parameter γ is fixed in the analysis, we suppress it from the notation and denote $\mathcal{L}_\gamma$ and $\widehat{\mathcal{L}}_\gamma$ by $\mathcal{L}$ and $\widehat{\mathcal{L}}$, respectively. Let $u_{\widehat{\theta}}$ be the minimizer of the empirical loss $\widehat{\mathcal{L}}(\theta)$, and let $u_{\theta_{\mathcal{A}}}$ be the optimal network approximation to the minimizer u^* of the functional $\mathcal{L}$ obtained by a randomized optimizer $\mathcal{A}$. The main aim is to bound the quantity $\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \mathcal{L}(u^*)$, which is also known as

the generalization error in statistical learning theory [63]. The following lemma gives a crucial decomposition of the generalization error.

Lemma 3.1. *The generalization error $\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \mathcal{L}(u^*)$ can be decomposed into*

$$\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \mathcal{L}(u^*) \leqslant \underbrace{[\mathcal{L}(\bar{u}) - \mathcal{L}(u^*)]}_{\mathcal{E}_{\text{approx}}} + \underbrace{2\sup_{u\in\mathcal{N}}|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)|}_{\mathcal{E}_{\text{stats}}} + \underbrace{[\widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}}) - \widehat{\mathcal{L}}(u_{\widehat{\theta}})]}_{\mathcal{E}_{\text{opt}}},$$

where $\bar{u}$ is any element in the network class $\mathcal{N}$.

Proof. Since $u_{\widehat{\theta}}$ is the minimizer of $\widehat{\mathcal{L}}$, we have

$$\widehat{\mathcal{L}}(u_{\widehat{\theta}}) - \widehat{\mathcal{L}}(\bar{u}) \leqslant 0, \quad \forall\, \bar{u} \in \mathcal{N}.$$

Consequently, by adding and subtracting terms, we deduce

$$\begin{aligned}\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \mathcal{L}(u^*) &= \left[\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}})\right] + \left[\widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}}) - \widehat{\mathcal{L}}(u_{\widehat{\theta}})\right] \\ &\quad + \left[\widehat{\mathcal{L}}(u_{\widehat{\theta}}) - \widehat{\mathcal{L}}(\bar{u})\right] + \left[\widehat{\mathcal{L}}(\bar{u}) - \mathcal{L}(\bar{u})\right] + \left[\mathcal{L}(\bar{u}) - \mathcal{L}(u^*)\right] \\ &\leqslant \left[\mathcal{L}(\bar{u}) - \mathcal{L}(u^*)\right] + 2\sup_{u\in\mathcal{N}}|\mathcal{L}(u) - \widehat{\mathcal{L}}(u)| + \left[\widehat{\mathcal{L}}(u_{\theta_{\mathcal{A}}}) - \widehat{\mathcal{L}}(u_{\widehat{\theta}})\right].\end{aligned}$$

This completes the proof of the lemma. □

By lemma 3.1, the generalization error $\mathcal{L}(u_{\theta_{\mathcal{A}}}) - \mathcal{L}(u^*)$ can be decomposed into three terms, i.e., approximation error $\mathcal{E}_{\text{approx}}$, statistical error $\mathcal{E}_{\text{stat}}$, and optimization error $\mathcal{E}_{\text{opt}}$. The error $\mathcal{E}_{\text{approx}}$ arises because we restrict the sought-for function within the set $\mathcal{N}$, instead of the whole space $BV_a(\Omega)$. The error $\mathcal{E}_{\text{stat}}$ is the quadrature error arising when approximating the population loss $\mathcal{L}$ with the empirical loss $\widehat{\mathcal{L}}$. The error $\mathcal{E}_{\text{opt}}$ arises from the fact that the optimizer we employ may not find a global minimizer. The error $\mathcal{E}_{\text{opt}}$ remains very challenging to analyze, due to the non-convexity nature of the optimization problem. Thus, we shall assume that the network is well trained and ignore the optimization error $\mathcal{E}_{\text{opt}}$. Note that the functional $\mathcal{L}$ is only convex in u but not strictly so. Hence a bound on the state approximation $u^* - u_{\theta_{\mathcal{A}}}$ does not follow directly. Below we analyze the approximation error $\mathcal{E}_{\text{approx}}$ and statistical error $\mathcal{E}_{\text{opt}}$, in the following two parts separately.

3.1. Approximation error

First we analyze the approximation error $\mathcal{E}_{\text{approx}}$, under certain *a priori* regularity assumption $u^* \in W^{2,1}(\Omega)$ on the minimizer u^* to the loss $\mathcal{L}$. Note that since any neural network function $\bar{u} \in \mathcal{N}$ is differentiable (with respect to the input variable x), and also the minimizer u^* is differentiable, the distributional derivative Du actually coincides with ∇u. Now we fix any $\bar{u} \in \mathcal{N}$, and let $v = \bar{u} - u^*$. Then by the triangle inequality and the definition of the loss $\mathcal{L}$, we have

$$\begin{aligned}\mathcal{L}(\bar{u}) &= \int_\Omega a(x)|\nabla v(x) + \nabla u^*(x)|\mathrm{d}x + \gamma\int_{\partial\Omega} a(x)|T(v+u^*) - g|\mathrm{d}s \\ &\leqslant \int_\Omega a(x)(|\nabla v(x)| + |\nabla u^*(x)|)\mathrm{d}x + \gamma\left(\int_{\partial\Omega} a(x)|Tu^* - g|\mathrm{d}s + \int_{\partial\Omega} a(x)|Tv|\mathrm{d}s\right) \\ &= \mathcal{L}(u^*) + \int_\Omega a(x)|\nabla v(x)|\mathrm{d}x + \gamma\int_{\partial\Omega} a(x)|Tv(x)|\mathrm{d}s.\end{aligned}$$

By assumption 2.1, $a(x)$ is bounded by α_1. Moreover, by the trace theorem [23], we have

$$\|Tv\|_{L^1(\partial\Omega)} \leqslant C_{\mathrm{em}}\|v\|_{W^{1,1}(\Omega)},$$

where $C_{\mathrm{em}} > 0$ is the embedding constant from $W^{1,1}(\Omega)$ into $L^1(\partial\Omega)$. Consequently,

$$\begin{aligned}\mathcal{L}(\bar{u}) - \mathcal{L}(u^*) &\leqslant \alpha_1\|v\|_{W^{1,1}(\Omega)} + \alpha_1\gamma C_{\mathrm{em}}\|v\|_{W^{1,1}(\Omega)} \\ &= \alpha_1(1 + C_{\mathrm{em}}\gamma)\|\bar{u} - u^*\|_{W^{1,1}(\Omega)}. \end{aligned} \tag{3.1}$$

To bound $\|\bar{u} - u^*\|_{W^{1,1}(\Omega)}$, we employ the neural network approximation theory from [26]. The main idea is to approximate u^* by localized Taylor polynomials, where the localization is realized by partition of unity (PU), and the polynomials are then approximated by neural networks. Let $\mathbb{1}_{[0,1]^d}$ be the characteristic function of the domain $[0,1]^d$. Note that there is no canonical way to build a PU exactly by neural networks with general activation functions other than ReLU. Gühring and Raslan [26] proposed to approximate $\mathbb{1}_{[0,1]^d}$ with bump functions defined by admissible activation functions with exponential or polynomial decay property. For the analysis below, we assume that the nonlinear activation function ρ is admissible in the following sense [26, definition 4.2].

Definition 3.1. Let $j \in \mathbb{N}$. The nonlinear activation function $\rho : \mathbb{R} \to \mathbb{R}$ satisfies

(a) ρ and ρ' are uniformly bounded by ρ_0 and $\rho_1 > 0$, respectively.
(b) ρ and ρ' are η_0- and η_1-Lipschitz, respectively.
(c) There exists $I > 0$ with $\rho \in C^j(\mathbb{R} \setminus [-I, I])$ and $\rho' \in W^{j-1,\infty}(\mathbb{R})$, if $j \geqslant 1$.

Let $\tau = \{0, 1\}$ be the order of PU. ρ is said to be exponential (polynomial) (j, τ)-PU-admissible, if additionally there exist $A = A(\rho), B = B(\rho) \in \mathbb{R}$, with $A < B$, some $C = C(\rho, j) > 0$ and $D = D(\rho, j) > 0$, such that

1. $|B - \rho^{(\tau)}(x)| \leqslant C\,\mathrm{e}^{-Dx}$ (Cx^{-D} if polynomial) for all $x > I$;
2. $|A - \rho^{(\tau)}(x)| \leqslant C\,\mathrm{e}^{Dx}$ ($C|x|^{-D}$ if polynomial) for all $x < -I$;
3. $|\rho^{(k)}(x)| \leqslant C\,\mathrm{e}^{-D|x|}$ ($C|x|^{-D}$ if polynomial) for all $x \in \mathbb{R} \setminus [-I, I]$ and all $k = \tau + 1, \ldots, j$.

Remark 3.1. For $\tau = 0$, ρ is approximately piecewise constant outside of a neighborhood of zero (e.g., sigmoid) and for $\tau = 1$, ρ is approximately piecewise affine-linear outside of a neighborhood of zero (e.g., exponential linear unit). In particular arctan and inverse square root unit are polynomial PU-admissible, and tanh and sigmoid are exponential PU-admissible.

Now we state the approximation theorem [26, proposition 4.8].

Theorem 3.1. *Let $d \in \mathbb{N}$, $j, \tau \in \mathbb{N}_0$, $k \in \{0, \ldots, j\}$, $n \geqslant k + 1$, $1 \leqslant p \leqslant \infty$, and $\mu > 0$. Let $\mathcal{F}_{n,d,p} := \{f \in W^{n,p}((0,1)^d) : \|f\|_{W^{n,p}((0,1)^d)} \leqslant 1\}$. Suppose that $\rho(x)$ is an exponential (polynomial) (j, τ)-PU admissible activation function, and there exists $x_0 \in \mathbb{R}$ such that ρ is three times continuously differentiable in a neighborhood of x_0. Then for any $\epsilon > 0$ and for any $f \in \mathcal{F}_{n,d,p}$, there exists a neural network f_{NN} with depth at most $C\log(d + n)$ and at most*

$$N_\theta = \begin{cases} C\epsilon^{-\frac{d}{n-k-\mu(k=2)}}, & \text{if } \rho \text{ is exponential admissible}, \\ C\epsilon^{-\frac{d}{n-k}}, & \text{if } \rho \text{ is polynomial admissible}, \end{cases}$$

non-zero weights, where $\mu \in (0,1)$ *is an arbitrarily small positive number, such that*

$$\|f - f_{\mathrm{NN}}\|_{W^{k,p}([0,1]^d)} \leqslant \epsilon.$$

Moreover, the weights in the neural network are bounded in absolute value by

$$\begin{cases} C(d,n,p,k)\epsilon^{-2-\frac{2(d/p+d+k+\mu(k=2))+d/p+d}{n-k-\mu(k=2)}}, & \text{if } \rho \text{ is exponential admissible}, \\ C(d,n,p,k)\epsilon^{-2-\frac{2(d/p+d+k)+d/p+d}{n-k}}, & \text{if } \rho \text{ is polynomial admissible}. \end{cases}$$

To bound the approximation error (3.1), we apply theorem 3.1 with $k = p = 1$ and $n = 2$. Then for any $\epsilon > 0$ and any $f \in W^{2,1}([0,1]^d)$ such that $\|f\|_{W^{2,1}((0,1)^d)} \leqslant 1$, there exists a neural network $\bar{f}$ with $\bar{c_1}(d)\log(d+2)$ layers and $\bar{c_2}(d)\epsilon^{-\frac{d}{1-\mu}}$ (or $\bar{c_2}(d)\epsilon^{-d}$, if ρ is polynomial admissible) number of network parameters each bounded by $\bar{c_3}(d)\epsilon^{-\frac{4+6d}{1-\mu}}$ (or $\bar{c}_3(d)\epsilon^{-(4+6d)}$ if ρ is polynomial admissible), such that

$$\|f - \bar{f}\|_{W^{1,1}([0,1]^d)} \leqslant \epsilon.$$

The following proposition records the approximation result.

Proposition 3.1. *Let the minimizer* u^* *to the functional* $\mathcal{L}$ *satisfies* $u^* \in W^{2,1}(\Omega)$, *and let* ρ *be the nonlinear activation function. Then for any* $\epsilon > 0$, *there exists a network work class*

$$\mathcal{N} = \begin{cases} \mathcal{N}\left(c_1\ \log(d+2), c_2\epsilon^{-\frac{d}{1-\mu}}, c_3\epsilon^{-\frac{4+6d}{1-\mu}}\right), & \text{if } \rho \text{ is exponential admissible}, \\ \mathcal{N}(c_1\ \log(d+2), c_2\epsilon^{-d},\ c_3\epsilon^{-(4+6d)}), & \text{if } \rho \text{ is polynomial admissable}, \end{cases}$$

with $\mu \in (0,1)$ *being an arbitrarily small positive number, such that there exists a* $\bar{u} \in \mathcal{N}$ *with*

$$\|u^* - \bar{u}\|_{W^{1,1}(\Omega)} \leqslant C\epsilon,$$

with the constant C *depending on* $\|u^*\|_{W^{2,1}(\Omega)}$. *In particular, there exists a neural network* $\bar{u} \in \mathcal{N}$ *such that*

$$\mathcal{L}(\bar{u}) - \mathcal{L}(u^*) \leqslant C\alpha_1(1 + C_{em}\gamma)\epsilon.$$

Proof. By Sobolev extension theorem [23] and the assumption $u^* \in W^{2,1}(\Omega)$, since $\Omega \subset (0,1)^d$, there exists a bounded extension of u^* from Ω to $(0,1)^d$, denoted by Eu^* such that

$$\|Eu^*\|_{W^{2,1}((0,1)^d)} \leqslant C\|u^*\|_{W^{2,1}(\Omega)}.$$

Then by theorem 3.1, we can find a neural network $u^* \in \mathcal{N}$ satisfies the desired approximation for the function $Eu^*/\|Eu^*\|_{W^{2,1}((0,1)^d)}$. Then the desired assertion follows directly. □

Remark 3.2. In proposition 3.1, we have assumed the existence of a minimizer $u^* \in W^{2,1}(\Omega)$ to the loss $\mathcal{L}$. This assumption may be relaxed to an approximate minimizer $u^\epsilon \in W^{2,1}(\Omega)$ such that $\mathcal{L}(u^\epsilon) - \inf_{u \in BV_a(\Omega)} \mathcal{L}(u) \leqslant \epsilon$. However, the $W^{2,1}(\Omega)$ norm of u^ϵ may depend on the tolerance ϵ, which obscures the dependence between the network parameters and the error tolerance ϵ.

3.2. Statistical error

In this part, we bound the statistical error $\sup_{u\in\mathcal{N}}|\mathcal{L}(u)-\widehat{\mathcal{L}}(u)|$. To this end, we define

$$\mathcal{L}_1(u)=|\Omega|\mathbb{E}_{X\sim U(\Omega)}[a(X)|\nabla u(X)|],\quad \widehat{\mathcal{L}}_1(\bar{u})=\frac{|\Omega|}{n_1}\sum_{i=1}^{n_1}a(X_i)|\nabla\bar{u}(X_i)|,$$

$$\mathcal{L}_2(u)=\gamma|\partial\Omega|\mathbb{E}_{Y\sim U(\partial\Omega)}[a(Y)|Tu(Y)-g(Y)|],\quad \widehat{\mathcal{L}}_2(\bar{u})=\gamma\frac{|\partial\Omega|}{n_2}\sum_{j=1}^{n_2}a(Y_j)|T\bar{u}(Y_j)-g(Y_j)|.$$

Then by the triangle inequality, we have

$$\sup_{u\in\mathcal{N}}|\mathcal{L}(u)-\widehat{\mathcal{L}}(u)|\leqslant\sum_{i=1}^{2}\sup_{u\in\mathcal{N}}|\mathcal{L}_i(u)-\widehat{\mathcal{L}}_i(u)|.$$

Below we denote both $U(\Omega)$ and $U(\partial\Omega)$ by μ, and set $n=n_1$ and $n=n_2$ accordingly. Hence, there are n i.i.d. samples drawn from μ, denoted by $Z_n=\{z_i\}_{i=1}^n$ with $z_i\sim\mu$. We analyze $\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|]$ and $\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_2(u)-\widehat{\mathcal{L}}_2(u)|]$ separately. The concept of Rademacher complexity plays a crucial role in the analysis. Rademacher complexity $\mathcal{R}_n(\mathcal{F})$ measures the capacity of a function class $\mathcal{F}$ restricted on n random samples Z_n [9, 10]. For many function classes, the Rademacher complexity is known. For example, see [43, theorem 3] for the class of two-layer neural networks.

Definition 3.2. The Rademacher complexity $\mathfrak{R}_n(\mathcal{F})$ of a function class $\mathcal{F}$ is defined by

$$\mathfrak{R}_n(\mathcal{F})=\mathbb{E}_{Z_n,\Sigma_n}\left[\sup_{u\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i u(z_i)\right|\right],$$

where $\Sigma_n=\{\sigma_i\}_{i=1}^n$ are n i.i.d. Rademacher variables, i.e., with probability $P(\sigma_i=1)=P(\sigma_i=-1)=\frac{1}{2}$.

Given an L-layer neural network class $\mathcal{N}$, we define an associated function class

$$\mathcal{G}=\left\{g:[0,1]^d\to\mathbb{R}\text{ such that }g(x)=|\nabla u(x)|,\quad\forall x\in[0,1]^d,\text{ with }u\in\mathcal{N}\right\}.$$

Recall that $|\nabla u(x)|$ denotes the Euclidean norm of the gradient vector $(\partial_{x_1}u,\ldots,\partial_{x_d}u)^t\in\mathbb{R}^d$. First, we bound $\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|]$ in terms of the Rademacher complexity $\mathfrak{R}_n(\mathcal{G})$. The proof is based on a standard symmetrization argument (see, e.g., [46, theorem 14]), and it is included only for completeness.

Lemma 3.2. *The following bound holds*

$$\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|\right]\leqslant 2\alpha_1|\Omega|\mathfrak{R}_n(\mathcal{G}).$$

Proof. We denote $\mathrm{I}=\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|]$. By the definitions of $\mathcal{L}_1$, $\widehat{\mathcal{L}}_1$, μ and Z_n, we have

$$\begin{aligned}\mathrm{I} &= \mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}\left|\,|\Omega|\mathbb{E}_\mu[a(Z)|\nabla u(Z)|] - \frac{|\Omega|}{n}\sum_{i=1}^n a(z_i)|\nabla u(z_i)|\right|\right]\\ &= \frac{|\Omega|}{n}\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}\left|n\mathbb{E}_\mu[a(Z)|\nabla u(Z)|] - \sum_{i=1}^n a(z_i)|\nabla u(z_i)|\right|\right]\\ &= \frac{|\Omega|}{n}\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \mathbb{E}_{\tilde{Z}_n}[a(\tilde{z}_i)|\nabla u(\tilde{z}_i)|] - \sum_{i=1}^n a(z_i)|\nabla u(z_i)|\right|\right],\end{aligned}$$

where $\tilde{Z}_n = \{\tilde{z}_i\}_{i=1}^n$ denotes n independent samples from the distribution μ, independent from Z_n. Since $\sup(\cdot)$ is a convex function, by Jensen's inequality, we deduce

$$\mathrm{I} \leqslant \frac{|\Omega|}{n}\mathbb{E}_{Z_n,\tilde{Z}_n}\left[\sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \left(a(\tilde{z}_i)|\nabla u(\tilde{z}_i)| - a(z_i)|\nabla u(z_i)|\right)\right|\right].$$

Since z_i and $\tilde{z}_i$ are i.i.d., the distribution of the supremum is unchanged when we swap them. One may insert any $\{\sigma_i\} \in \{\pm 1\}^n$, in particular, the expectation of the supremum is unchanged. Since this is true for any σ_i, we can take the expectation over any random choice of the σ_i. Thus, we deduce

$$\mathrm{I} \leqslant \frac{|\Omega|}{n}\mathbb{E}_{Z_n,\tilde{Z}_n,\Sigma_n}\left[\sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \sigma_i\left(a(\tilde{z}_i)|\nabla u(\tilde{z}_i)| - a(z_i)|\nabla u(z_i)|\right)\right|\right].$$

Then by the triangle inequality, we have

$$\begin{aligned}\mathrm{I} &\leqslant \frac{|\Omega|}{n}\mathbb{E}_{Z_n,\tilde{Z}_n,\Sigma_n}\left[\sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \sigma_i a(\tilde{z}_i)|\nabla u(\tilde{z}_i)|\right| + \sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \sigma_i a(z_i)|\nabla u(z_i)|\right|\right]\\ &= \frac{|\Omega|}{n}\mathbb{E}_{\tilde{Z}_n,\Sigma_n}\left[\sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \sigma_i a(\tilde{z}_i)|\nabla u(\tilde{z}_i)|\right|\right] + \frac{|\Omega|}{n}\mathbb{E}_{Z_n,\Sigma_n}\left[\sup_{u\in\mathcal{N}}\left|\sum_{i=1}^n \sigma_i a(z_i)|\nabla u(z_i)|\right|\right]\\ &= 2|\Omega|\mathbb{E}_{Z_n,\Sigma_n}\left[\sup_{u\in\mathcal{N}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i a(z_i)|\nabla u(z_i)|\right|\right].\end{aligned}$$

Now by assumption 2.1, we have $a \leqslant \alpha_1$ a.e. Ω and by the multiplicative inequality of Rademacher complexity, we obtain

$$\mathrm{I} \leqslant 2\alpha_1|\Omega|\mathbb{E}_{Z_n,\Sigma_n}\left[\sup_{u\in\mathcal{N}}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i|\nabla u(z_i)|\right|\right] = 2\alpha_1|\Omega|\mathfrak{R}(\mathcal{G}).$$

This completes the proof of the lemma. □

By lemma 3.2, it suffices to bound the Rademacher complexity $\mathcal{R}_n(\mathcal{G})$ of the function class $\mathcal{G}$. This can be achieved using Dudley's formula from the theory of empirical process [69]. First we recall the covering number of a function class.

Definition 3.3. Let (X, ρ) be a metric space. An ϵ-cover of a set $A \subset X$ with respect to the metric ρ is a collection of points $\{x_i\}_{i=1}^n \subset A$ such that for every $x \in A$, there exists $i \in \{1, \ldots, n\}$ such that $\rho(x, x_i) \leqslant \epsilon$. The ϵ-covering number $\mathcal{C}(A, \rho, \epsilon)$ is the cardinality of the smallest ϵ-cover of A with respect to the metric ρ.

The Rademacher complexity $\mathcal{R}_n(\mathcal{G})$ is related to the covering number $\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$ by the refined Dudley's formula [20] (see, e.g., [60, 65] for the current form). Note that the statement is slightly different from the standard Dudley's theorem where the covering number is based on the empirical ℓ^2-metric instead of the $L^\infty(\Omega)$-metric. However, the $L^\infty(\Omega)$ metric is stronger than the empirical ℓ^2 metric, and the covering number is monotonically increasing with respect to the metric [60, lemma 2]. The lemma follows directly from the classical Dudley's theorem.

Lemma 3.3. *The Rademacher complexity $\mathfrak{R}_n(\mathcal{G})$ of a function class $\mathcal{G}$ is bounded by*

$$\mathfrak{R}_n(\mathcal{G}) \leqslant \inf_{0<\delta<M} \left(4\delta + \frac{12}{\sqrt{n}} \int_\delta^M \sqrt{\log \mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)}\mathrm{d}\epsilon\right),$$

where $\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$ is the covering number of the set $\mathcal{G}$, and $M := \sup_{g\in\mathcal{G}}\|g\|_{L^\infty(\Omega)}$.

Next we bound the covering number $\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$ of the set $\mathcal{G}$. This is based on the Lipschitz continuity of functions in the set $\mathcal{G}$ with respect to the network parameter θ. For $g, \tilde{g} \in \mathcal{G}$, there exist two neural networks $f^{(L)}$ and $\tilde{f}^{(L)}$ (with the corresponding network parameters being $\theta = \{W^{(\ell)}, b^{(\ell)}\}_{\ell=1}^L$ and $\tilde{\theta} = \{\tilde{W}^{(\ell)}, \tilde{b}^{(\ell)}\}_{\ell=1}^L$) such that $f^{(L)}$ and $\tilde{f}^{(L)}$ can be written as

$$\begin{aligned} f^{(L)} &= W^{(L)}\rho(W^{(L-1)}\rho(W^{(L-2)}\ldots + b^{(L-2)}) + b^{(L-1)}) + b^{(L)},\\ \tilde{f}^{(L)} &= \tilde{W}^{(L)}\rho(\tilde{W}^{(L-1)}\rho(\tilde{W}^{(L-2)}\ldots + \tilde{b}^{(L-2)}) + \tilde{b}^{(L-1)}) + \tilde{b}^{(L)}, \end{aligned}$$

and accordingly

$$g(x) = |\nabla f^{(L)}(x)| \quad \text{and} \quad \tilde{g}(x) = |\nabla \tilde{f}^{(L)}(x)|.$$

To indicate the dependence of g on θ, we write g_θ below. To bound the covering number $\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$ of $\mathcal{G}$, we bound $\|g_\theta - \tilde{g}_{\tilde{\theta}}\|_{L^\infty(\Omega)}$ in terms of $\|\theta - \tilde{\theta}\|_{\ell^\infty}$. Meanwhile, we have

$$\|g_\theta - \tilde{g}_{\tilde{\theta}}\|_{L^\infty(\Omega)} = \||\nabla f^{(L)}| - |\nabla \tilde{f}^{(L)}|\|_{L^\infty(\Omega)} \leqslant \|\nabla(f^{(L)} - \tilde{f}^{(L)})|\|_{L^\infty(\Omega)}. \quad (3.2)$$

Thus, it suffices to bound the partial derivatives $\|\partial_{x_i}(f^{(L)} - \tilde{f}^{(L)})\|_{L^\infty(\Omega)}$, for $i = 1, 2, \ldots, d$. The next lemma gives the requisite estimates (as well as auxiliary estimates). Note that under different assumptions (i.e., boundedness assumptions on the activation function, different norms on the parameters, or evaluation of the neural networks on input data), similar approaches can be found in [3, 8, 12].

Lemma 3.4. *Let the activation function ρ satisfy the conditions (a) and (b) in definition 3.1, $\mathcal{W}$ be the width of the network class, and R the ℓ^∞ bound on the network parameters θ. Then with $\rho_0, \rho_1, \eta_0, \eta_1, R$ and $\mathcal{W}$, there holds*

$$\|\partial_{x_i}(f^{(L)} - \tilde{f}^{(L)})\|_{L^\infty(\Omega)} \leqslant L^2\rho_0\rho_1^{L-1}\eta_1\eta_0^{L-2}R^{2L-2}\mathcal{W}^{2L-2}\|\theta - \tilde{\theta}\|_{\ell^\infty}, \quad (3.3)$$

$$\sup_{g\in\mathcal{G}}\|g\|_{L^\infty(\Omega)} \leqslant \sqrt{d}R^L(\rho_1\mathcal{W})^{L-1}. \quad (3.4)$$

Proof. Let $r := \|\theta - \tilde{\theta}\|_{\ell^\infty}$. Recall that $f^{(1)} = \rho(W^{(1)}x + b^{(1)})$, and $f^{(\ell)} = \rho(W^{(\ell)}f^{(\ell-1)} + b^{(\ell)})$, for $\ell = 1, 2, \ldots, L-1$. We denote the jth component of $f^{(\ell)} \in \mathbb{R}^{d_\ell}$ by $f_j^{(\ell)}$. Noting $W^{(L)} \in \mathbb{R}^{1\times d_{L-1}}$ and $b^{(L)} \in \mathbb{R}$, writing out explicitly $f^{(L)}$ and $\tilde{f}^{(L)}$ and applying the triangle inequality lead to

$$
\begin{aligned}
&\left\|\partial_{x_i}(f^{(L)}-\tilde{f}^{(L)})\right\|_{L^\infty(\Omega)}\\
&\quad=\left\|\partial_{x_i}\left(\sum_{j=1}^{d_{L-1}}W_j^{(L)}f_j^{(L-1)}+b^{(L)}\right)-\partial_{x_i}\left(\sum_{j=1}^{d_{L-1}}\tilde{W}_j^{(L)}\tilde{f}_j^{(L-1)}+\tilde{b}^{(L)}\right)\right\|_{L^\infty(\Omega)}\\
&\quad=\left\|\sum_{j=1}^{d_{L-1}}W_j^{(L)}\partial_{x_i}f_j^{(L-1)}-\sum_{j=1}^{d_{L-1}}\tilde{W}_j^{(L)}\partial_{x_i}\tilde{f}_j^{(L-1)}\right\|_{L^\infty(\Omega)}\\
&\quad\leqslant\sum_{j=1}^{d_{L-1}}\left\|W_j^{(L)}\partial_{x_i}f_j^{(L-1)}-\tilde{W}_j^{(L)}\partial_{x_i}\tilde{f}_j^{(L-1)}\right\|_{L^\infty(\Omega)}\\
&\quad\leqslant\sum_{j=1}^{d_{L-1}}\left[|W_j^{(L)}-\tilde{W}_j^{(L)}|\|\partial_{x_i}f_j^{(L-1)}\|_{L^\infty(\Omega)}+|\tilde{W}_j^{(L)}|\|\partial_{x_i}\left(f_j^{(L-1)}-\tilde{f}_j^{(L-1)}\right)\|_{L^\infty(\Omega)}\right]\\
&\quad\leqslant r\sum_{j=1}^{d_{L-1}}\|\partial_{x_i}f_j^{(L-1)}\|_{L^\infty(\Omega)}+R\sum_{j=1}^{d_{L-1}}\|\partial_{x_i}(f_j^{(L-1)}-\tilde{f}_j^{(L-1)})\|_{L^\infty(\Omega)},
\end{aligned}
\tag{3.5}
$$

in view of the definition of r and the condition $|\tilde{W}_j^{(L)}|\leqslant R$. Thus, to bound $\|\partial_{x_i}(f^{(L)}-\tilde{f}^{(L)})\|_{L^\infty(\Omega)}$, it suffices to estimate $\|\partial_{x_i}f_j^{(L-1)}\|_{L^\infty(\Omega)}$ and $\|\partial_{x_i}(f_j^{(L-1)}-\tilde{f}_j^{(L-1)})\|_{L^\infty(\Omega)}$. We derive the requisite bounds below using mathematical induction. The rest of the proof is elementary but fairly lengthy, and hence we divide it into four steps.

Step 1 Bound $\|\partial_{x_i}f_j^{(\ell)}\|_{L^\infty(\Omega)}$ for $\ell=1,2,\ldots,L-1$, $j=1,2,\ldots,d_\ell$. By the chain rule, we have

$$
\partial_{x_i}f_j^{(\ell)}=\rho'\left(\sum_{k=1}^{d_{\ell-1}}W_{jk}^{(\ell)}f_k^{(\ell-1)}+b_j^{(\ell)}\right)\sum_{k=1}^{d_{\ell-1}}W_{jk}^{(\ell)}\partial_{x_i}f_k^{(\ell-1)}.
$$

For the case $\ell=1$, the assumptions $|\rho'|\leqslant\rho_1$ (cf definition 3.1(a)) and $|W_{ji}^{(1)}|\leqslant R$ yields

$$
\|\partial_{x_i}f_j^{(1)}\|_{L^\infty(\Omega)}\leqslant\left\|\rho'\left(\sum_{k=1}^{d}W_{jk}^{(1)}x_k+b_j^{(1)}\right)W_{ji}^{(1)}\right\|_{L^\infty(\Omega)}\leqslant\rho_1R.
$$

For $\ell\geqslant 2$, the triangle inequality and the conditions $|\rho'|\leqslant\rho_1$, $|W_{jk}^{(\ell)}|\leqslant R$ and $d_{\ell-1}\leqslant\mathcal{W}$ imply

$$
\|\partial_{x_i}f_j^{(\ell)}\|_{L^\infty(\Omega)}\leqslant\rho_1\sum_{k=1}^{d_{\ell-1}}|W_{jk}^{(\ell)}|\|\partial_{x_i}f_k^{(\ell-1)}\|_{L^\infty(\Omega)}\leqslant\rho_1R\mathcal{W}\max_k\|\partial_{x_i}f_k^{(\ell-1)}\|_{L^\infty(\Omega)}.
$$

Combining the preceding two estimates directly leads to

$$
\|\partial_{x_i}f_j^{(\ell)}\|_{L^\infty(\Omega)}\leqslant(\rho_1R)^\ell\mathcal{W}^{\ell-1},\quad \ell=1,2,\ldots,L-1,\ \ j=1,2,\ldots,d_\ell.
\tag{3.6}
$$

Step 2 Bound $\|f_j^{(\ell)} - \tilde{f}_j^{(\ell)}\|_{L^\infty(\Omega)}$ for $\ell = 1, 2, \ldots, L-1$, $j = 1, 2, \ldots, d_\ell$, assuming $\rho_0, \rho_1, \eta_0, \eta_1 \geqslant 1$. For the case $\ell = 1$, by the definitions of $f_j^{(1)}$ and $\tilde{f}_j^{(1)}$, the Lipschitz continuity of ρ, and the triangle inequality, we have

$$\begin{aligned}
&\|f_j^{(1)} - \tilde{f}_j^{(1)}\|_{L^\infty(\Omega)} \\
&\quad = \left\| \rho\left(\sum_{k=1}^d W_{jk}^{(1)} x_k + b_j^{(1)} \right) - \rho\left(\sum_{k=1}^d \tilde{W}_{jk}^{(1)} x_k + \tilde{b}_j^{(1)} \right) \right\|_{L^\infty(\Omega)} \\
&\quad \leqslant \eta_0 \left\| \sum_{k=1}^d W_{jk}^{(1)} x_k + b_j^{(1)} - \sum_{k=1}^d \tilde{W}_{jk}^{(1)} x_k - \tilde{b}_j^{(1)} \right\|_{L^\infty(\Omega)} \\
&\quad \leqslant \eta_0 \sum_{k=1}^d |W_{jk}^{(1)} - \tilde{W}_{jk}^{(1)}| \|x_k\|_{L^\infty(\Omega)} + \eta_0 |b_j^{(1)} - \tilde{b}_j^{(1)}| \leqslant \eta_0 r(1 + \rho_0 \mathcal{W}),
\end{aligned}$$

in view of the definition $r = \|\theta - \tilde{\theta}\|_{\ell^\infty}$ and the trivial estimate $\|x_p\|_{L^\infty(\Omega)} \leqslant 1$ for all $p = 1, \ldots, d$, since $x \in [0,1]^d$. Meanwhile, for the case $\ell \geqslant 2$, by the Lipschitz continuity of and the uniform bound ρ_0 on ρ, the triangle inequality and the induction hypothesis, we obtain

$$\begin{aligned}
&\|f_j^{(\ell)} - \tilde{f}_j^{(\ell)}\|_{L^\infty(\Omega)} \\
&\quad \leqslant \eta_0 \left\| \sum_{k=1}^{d_{\ell-1}} W_{jk}^{(\ell)} f_k^{(\ell-1)} + b_j^{(\ell)} - \sum_{k=1}^{d_{\ell-1}} \tilde{W}_{jk}^{(\ell)} \tilde{f}_k^{(\ell-1)} - \tilde{b}_j^{(\ell)} \right\|_{L^\infty(\Omega)} \\
&\quad \leqslant \eta_0 |b_j^{(\ell)} - \tilde{b}_j^{(\ell)}| + \eta_0 \sum_{k=1}^{d_{\ell-1}} \Big[|W_{jk}^{(\ell)} - \tilde{W}_{jk}^{(\ell)}| \|f_k^{(\ell-1)}\|_{L^\infty(\Omega)} \\
&\qquad + |\tilde{W}_{jk}^{(\ell)}| \|f_k^{(\ell-1)} - \tilde{f}_k^{(\ell-1)}\|_{L^\infty(\Omega)} \Big] \\
&\quad \leqslant \eta_0 r + \eta_0 r \rho_0 d_{\ell-1} + \eta_0 R \sum_{k=1}^{d_{\ell-1}} \|f_k^{(\ell-1)} - \tilde{f}_k^{(\ell-1)}\|_{L^\infty(\Omega)} \\
&\quad \leqslant \eta_0 (1 + \rho_0 \mathcal{W}) r + \eta_0 R \mathcal{W} c_{\ell-1}
\end{aligned}$$

with $c_\ell = \max_k \|f_k^{(\ell)} - \tilde{f}_k^{(\ell)}\|_{L^\infty(\Omega)}$. Then the preceding inequality implies

$$c_\ell \leqslant \eta_0 (1 + \rho_0 \mathcal{W}) r + \eta_0 R \mathcal{W} c_{\ell-1}.$$

By repeatedly applying the inequality and using the bound on c_1, we arrive at

$$c_\ell \leqslant \eta_0 r (1 + \rho_0 \mathcal{W}) \left(1 + \cdots + (\eta_0 R \mathcal{W})^{\ell-1}\right).$$

In particular, we directly obtain (for $\eta_0, R, \mathcal{W} \geqslant 1$)

$$\|f_j^{(\ell)} - \tilde{f}_j^{(\ell)}\|_{L^\infty(\Omega)} \leqslant 2\ell \rho_0 \eta_0^\ell \mathcal{W}^\ell R^{\ell-1} r, \quad \ell = 1, 2, \ldots, L-1, \ j = 1, 2, \ldots, d_\ell. \tag{3.7}$$

Step 3 Bound the term $D_j^{(\ell)} := \|\rho'(\sum_{k=1}^{d_{\ell-1}} W_{jk}^{(\ell)} f_k^{(\ell-1)} + b_j^{(\ell)}) - \rho'(\sum_{k=1}^{d_{\ell-1}} \tilde{W}_{jk}^{(\ell)} \tilde{f}_k^{(\ell-1)} + \tilde{b}_j^{(\ell)})\|_{L^\infty(\Omega)}$. By the Lipschitz continuity of ρ', the triangle inequality, and the bounds $|\rho| \leqslant \rho_0$ and $d_{\ell-1} \leqslant \mathcal{W}$, we have

$$\begin{aligned} D_j^{(\ell)} &\leqslant \eta_1 \left\| \sum_{k=1}^{d_{\ell-1}} W_{jk}^{(\ell)} f_k^{(\ell-1)} + b_j^{(\ell)} - \sum_{k=1}^{d_{\ell-1}} \tilde{W}_{jk}^{(\ell)} \tilde{f}_k^{(\ell-1)} - \tilde{b}_j^{(\ell)} \right\|_{L^\infty(\Omega)} \\ &\leqslant \eta_1 |b_j^{(\ell)} - \tilde{b}_j^{(\ell)}| + \eta_1 \sum_{k=1}^{d_{\ell-1}} \Big[|W_{jk}^{(\ell)} - \tilde{W}_{jk}^{(\ell)}| \|f_k^{(\ell-1)}\|_{L^\infty(\Omega)} \\ &\quad + |\tilde{W}_{jk}^{(\ell)}| \|f_k^{(\ell-1)} - \tilde{f}_k^{(\ell-1)}\|_{L^\infty(\Omega)} \Big] \\ &\leqslant \eta_1 r(1 + \rho_0 \mathcal{W}) + \eta_1 R \mathcal{W} \max_k \|f_k^{(\ell-1)} - \tilde{f}_k^{(\ell-1)}\|_{L^\infty(\Omega)}. \end{aligned}$$

This and the bound (3.7) imply

$$D_j^{(\ell)} \leqslant 2\ell \rho_0 \eta_1 \eta_0^{\ell-1} \mathcal{W}^\ell R^{\ell-1} r. \tag{3.8}$$

Step 4 Bound $P_{ij}^{(\ell)} := \|\partial_{x_i}(f_j^{(\ell)} - \tilde{f}_j^{(\ell)})\|_{L^\infty(\Omega)}$ for $\ell = 1, 2, \ldots, L-1$, $j = 1, \ldots, d_\ell$. We claim

$$P_{ij}^{(\ell)} \leqslant \ell(\ell+2) \rho_0 \rho_1^\ell \eta_1 \eta_0^{\ell-1} (R\mathcal{W})^{2\ell-1} r, \quad \ell = 1, 2, \ldots, L-1, \ j = 1, \ldots, d_\ell. \tag{3.9}$$

For the case $\ell = 1$, the chain rule and the triangle inequality give

$$\begin{aligned} P_{ij}^{(1)} &= \left\| \rho'\left(\sum_{k=1}^{d} W_{jk}^{(1)} x_k + b_j^{(1)} \right) W_{ji}^{(1)} - \rho'\left(\sum_{k=1}^{d} \tilde{W}_{jk}^{(1)} x_k + \tilde{b}_j^{(1)} \right) \tilde{W}_{ji}^{(1)} \right\|_{L^\infty(\Omega)} \\ &\leqslant \left\| \rho'\left(\sum_{k=1}^{d} W_{jk}^{(1)} x_k + b_j^{(1)} \right) - \rho'\left(\sum_{k=1}^{d} \tilde{W}_{jk}^{(1)} x_k + \tilde{b}_j^{(1)} \right) \right\|_{L^\infty(\Omega)} |W_{ji}^{(1)}| \\ &\quad + \left\| \rho'\left(\sum_{k=1}^{d} \tilde{W}_{jk}^{(1)} x_k + \tilde{b}_j^{(1)} \right) \right\|_{L^\infty(\Omega)} |W_{ji}^{(1)} - \tilde{W}_{ji}^{(1)}|. \end{aligned}$$

Then it follows from the bound (3.8) (with $\ell = 1$) that

$$P_{ij}^{(1)} \leqslant 2\rho_0 \eta_1 \mathcal{W} r \cdot R + \rho_1 r \leqslant 3\rho_0 \eta_1 R \mathcal{W} \rho_1 r.$$

Now suppose that the claim holds for some $\ell \geqslant 1$. Then for $\ell + 1$, by the chain rule again, we have

$$\begin{aligned} P_{ij}^{(\ell+1)} &= \left\| \rho'\left(\sum_{k=1}^{d_\ell} W_{jk}^{(\ell+1)} f_k^{(\ell)} + b_j^{(\ell+1)} \right) \cdot \sum_{k=1}^{d_\ell} W_{jk}^{(\ell+1)} \partial_{x_i} f_k^{(\ell)} \right. \\ &\qquad \left. - \rho'\left(\sum_{k=1}^{d_\ell} \tilde{W}_{jk}^{(\ell+1)} \tilde{f}_k^{(\ell)} + \tilde{b}_j^{(\ell+1)} \right) \cdot \sum_{k=1}^{d_\ell} \tilde{W}_{jk}^{(\ell+1)} \partial_{x_i} \tilde{f}_k^{(\ell)} \right\|_{L^\infty(\Omega)} \\ &\leqslant \left\| \left(\rho'\left(\sum_{k=1}^{d_\ell} W_{jk}^{(\ell+1)} f_k^{(\ell)} + b_j^{(\ell+1)} \right) \right. \right. \end{aligned}$$

$$
\begin{aligned}
&- \rho'\left(\sum_{k=1}^{d_\ell} \tilde{W}_{jk}^{(\ell+1)} \tilde{f}_k^{(\ell)} + \tilde{b}_j^{(\ell+1)}\right)\Bigg) \sum_{k=1}^{d_\ell} W_{jk}^{(\ell+1)} \partial_{x_i} f_k^{(\ell)} \Bigg\|_{L^\infty(\Omega)} \\
&+ \left\| \rho'\left(\sum_{k=1}^{d_\ell} \tilde{W}_{jk}^{(\ell+1)} \tilde{f}_k^{(\ell)} + \tilde{b}_j^{(\ell+1)}\right) \right. \\
&\left. \times \left(\sum_{k=1}^{d_\ell} W_{jk}^{(\ell+1)} \partial_{x_i} f_k^{(\ell)} - \sum_{k=1}^{d_\ell} \tilde{W}_{jk}^{(\ell+1)} \partial_{x_i} \tilde{f}_k^{(\ell)}\right) \right\|_{L^\infty(\Omega)} := \mathrm{I}_1 + \mathrm{I}_2.
\end{aligned}
$$

It follows directly from the bounds (3.8) and (3.6) and the triangle inequality that

$$
\mathrm{I}_1 \leqslant 2(\ell+1)\rho_0 \rho_1^\ell \eta_1 \eta_0^\ell R^{2\ell+1} \mathcal{W}^{2\ell+1} r.
$$

Similarly, the bound (3.6), the induction hypothesis for $P_{ij}^{(\ell)}$, and the condition $d_\ell \leqslant \mathcal{W}$ imply

$$
\begin{aligned}
\mathrm{I}_2 &\leqslant \rho_1 \sum_{k=1}^{d_\ell} \Big[|W_{jk}^{(\ell+1)} - \tilde{W}_{jk}^{(\ell+1)}| \|\partial_{x_i} f_k^{(\ell)}\|_{L^\infty(\Omega)} \\
&\quad + |\tilde{W}_{jk}^{(\ell+1)}| \|\partial_{x_i} f_k^{(\ell)} - \partial_{x_i} \tilde{f}_k^{(\ell)}\|_{L^\infty(\Omega)} \Big] \\
&\leqslant \rho_1 \mathcal{W} r \cdot (\rho_1 R)^\ell \mathcal{W}^{\ell-1} + \rho_1 \mathcal{W} R \cdot \ell(\ell+2)\rho_0 \rho_1^\ell \eta_1 \eta_0^{\ell-1} (R\mathcal{W})^{2\ell-1} r \\
&= \rho_1^{\ell+1} (R\mathcal{W})^\ell r + \ell(\ell+2)\rho_0 \rho_1^{\ell+1} \eta_1 \eta_0^{\ell-1} (R\mathcal{W})^{2\ell} r.
\end{aligned}
$$

Consequently,

$$
P_{ij}^{(\ell+1)} \leqslant \mathrm{I}_1 + \mathrm{I}_2 \leqslant (\ell+3)(\ell+1)\rho_0 \rho_1^{\ell+1} \eta_1 \eta_0^\ell (R\mathcal{W})^{2\ell+1} r,
$$

which completes the induction step and proves the claim (3.9).

Finally, the inequalities (3.5), (3.6) and (3.9) together lead to

$$
\begin{aligned}
\|\partial_{x_i}(f^{(L)} - \tilde{f}^{(L)})\|_{L^\infty(\Omega)} &\leqslant r\mathcal{W}(\rho_1 R)^{L-1} \mathcal{W}^{L-2} \\
&\quad + (L+1)(L-1) R\mathcal{W} \cdot \rho_0 \rho_1^{L-1} \eta_1 \eta_0^{L-2} (R\mathcal{W})^{2L-3} r \\
&\leqslant L^2 \rho_0 \rho_1^{L-1} \eta_1 \eta_0^{L-2} (R\mathcal{W})^{2L-2} r.
\end{aligned}
$$

This shows the bound (3.3). Meanwhile, we have

$$
\sup_{f^{(L)} \in \mathcal{N}} \| |\nabla f^{(L)}(x)| \|_{L^\infty(\Omega)} \leqslant \sup_{f^{(L)} \in \mathcal{N}} \left(\sum_{i=1}^{d} \|\partial_{x_i} f^{(L)}\|_{L^\infty(\Omega)}^2 \right)^{\frac{1}{2}}.
$$

Direct computation gives

$$
\partial_{x_i} f^{(L)} = \partial_{x_i} \left(\sum_{j=1}^{d_{L-1}} W_j^{(L)} f_j^{(L-1)} + B^{(L)} \right) = \sum_{j=1}^{d_{L-1}} W_j^{(L)} \partial_{x_i} f_j^{(L-1)}.
$$

This, the condition $|W_j^{(L)}| \leqslant R$ and the bound (3.6) imply

$$\|\partial_{x_i} f^{(L)}\|_{L^\infty(\Omega)} \leqslant R\mathcal{W}(\rho_1 R)^{L-1}\mathcal{W}^{L-2} = R^L(\rho_1\mathcal{W})^{L-1}.$$

Combining these estimates yields the bound (3.4). This completes the proof of lemma 3.4. □

Remark 3.3. Throughout the proof, without loss of generality, we have assumed $\rho_0, \rho_1, \eta_0, \eta_1 \geqslant 1$. Otherwise, when $\rho_0, \rho_1, \eta_0, \eta_1 \leqslant 1$, we have

$$\|f_k^{(\ell)} - \tilde{f}_k^{(\ell)}\|_{L^\infty(\Omega)} \leqslant 2\ell\mathcal{W}^\ell R^{\ell-1} r, \quad D_{ij}^{(\ell)} \leqslant 2\ell\mathcal{W}^\ell R^{\ell-1} r, \quad \text{and}$$
$$\|\partial_{x_i}(f_j^{(\ell)} - \tilde{f}_j^{(\ell)})\|_{L^\infty(\Omega)} \leqslant \ell(\ell+2)(R\mathcal{W})^{2\ell-1} r.$$

In particular, we have

$$\|\partial_{x_i}(f^{(L)} - \tilde{f}^{(L)})\|_{L^\infty(\Omega)} \leqslant L^2(R\mathcal{W})^{2L-2} r.$$

The next result shows that the covering number $\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$ can be reduced to that of the parameter space Θ.

Corollary 3.1. *Let the activation function ρ satisfy (a) and (b) in definition 3.1. Then there holds*

$$\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon) \leqslant \mathcal{C}\left(\Theta, \|\cdot\|_{\ell^\infty}, \frac{\epsilon}{\Lambda}\right), \quad \textit{with} \quad \Lambda := \sqrt{d}L^2\rho_0\rho_1^{L-1}\eta_1\eta_0^{L-2}(R\mathcal{W})^{2L-2}. \tag{3.10}$$

Proof. It follows from lemma 3.4 that

$$\begin{aligned}\|g_\theta - \tilde{g}_{\tilde{\theta}}\|_{L^\infty(\Omega)} &\leqslant \||\nabla f^{(L)} - \nabla\tilde{f}^{(L)}|\|_{L^\infty(\Omega)} \\ &\leqslant \left(\sum_{i=1}^d \|\partial_{x_i} f^{(L)} - \partial_{x_i}\tilde{f}^{(L)}\|_{L^\infty(\Omega)}^2\right)^{\frac{1}{2}} \\ &\leqslant \sqrt{d}L^2\rho_0\rho_1^{L-1}\eta_1\eta_0^{L-2}(R\mathcal{W})^{2L-2}\|\theta - \tilde{\theta}\|_\infty. \end{aligned} \tag{3.11}$$

Thus, the mapping $\theta \mapsto g_\theta$ is Lipschitz continuous, which enables reducing the covering number of the function class $\mathcal{G}$ to that of the parametric space Θ. With the given choice of Λ, the estimate (3.11) and the definition of $\mathcal{C}(\mathcal{G}, \|\cdot\|_{L^\infty(\Omega)}, \epsilon)$ imply the assertion. □

Moreover, the parametrization Θ is an N_θ-dimensional ball with a radius R (with respect to the Euclidean ℓ^∞ norm $\|\cdot\|_{\ell^\infty}$). Recall that the total number N_θ of parameters in the network $f^{(L)}$ is $N_\theta = \sum_{\ell=1}^L d_\ell d_{\ell-1} + d_\ell$. Next, we recall a basic result on the covering number of a hypercube with respect to the maximum norm $\|\cdot\|_{\ell^\infty}$, which follows directly from a counting argument. Note that a similar statement holds for any ball in a finite-dimensional Banach space [17, proposition 5].

Lemma 3.5. *Let $n \in \mathbb{N}$, $R \in [1, \infty)$, $\epsilon \in (0, 1)$, and $B_R := \{x \in \mathbb{R}^n : \|x\|_{\ell^\infty} \leqslant R\}$. Then there holds*

$$\log \mathcal{C}(B_R, \|\cdot\|_{\ell^\infty}, \epsilon) \leqslant n \log\left(\frac{2R}{\epsilon}\right).$$

Now we can bound the statistical error $\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|]$.

Proposition 3.2. *The following estimate holds*

$$\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|\right]\leqslant C_1\frac{R^L N_\theta^L(\sqrt{\log n}+\sqrt{\log R}+\sqrt{\log N_\theta})}{\sqrt{n}},$$

where the constant $C_1>0$ depends on $|\Omega|$, d, L, ρ_1^L, ρ_0, η_0, and η_1 at most polynomially.

Proof. Combining the estimate (3.10) with lemma 3.5 gives, with $\Lambda:=\sqrt{d}L^2\rho_0\rho_1^{L-1}\eta_1\eta_0^{L-2}(R\mathcal{W})^{2L-2}$

$$\log\mathcal{C}\left(\mathcal{G},\|\cdot\|_{L^\infty(\Omega)},\epsilon\right)\leqslant\log\mathcal{C}\left(\Theta,\|\cdot\|_{\ell^\infty},\frac{\epsilon}{\Lambda}\right)\leqslant N_\theta\log\left(\frac{2R\Lambda}{\epsilon}\right). \tag{3.12}$$

By the estimate (3.4), one may take $M=\sqrt{d}R^L(\rho_1\mathcal{W})^{L-1}$. This choice, the estimate (3.12) and the refined Dudley's formula in lemma 3.3 with the choice $\delta=\frac{1}{\sqrt{n}}$ yield

$$\begin{aligned}\mathfrak{R}_n(\mathcal{G})&\leqslant\inf_{0<\delta<M}\left(4\delta+\frac{12}{\sqrt{n}}\int_\delta^M\sqrt{\log\mathcal{C}(\mathcal{G},\|\cdot\|_{L^\infty},\epsilon)}\,\mathrm{d}\epsilon\right)\\&\leqslant\frac{4}{\sqrt{n}}+\frac{12}{\sqrt{n}}\int_{\frac{1}{\sqrt{n}}}^M\sqrt{N_\theta\log\left(\frac{2R\Lambda}{\epsilon}\right)}\,\mathrm{d}\epsilon\\&\leqslant\frac{4}{\sqrt{n}}+\frac{12}{\sqrt{n}}M\sqrt{N_\theta\log(2R\Lambda\sqrt{n})}\\&\leqslant\frac{4}{\sqrt{n}}+\frac{12}{\sqrt{n}}\sqrt{d}R^L(\rho_1\mathcal{W})^{L-1}\sqrt{N_\theta}\sqrt{\log(2R\cdot\sqrt{d}L^2\rho_0\rho_1^{L-1}\eta_1\eta_0^{L-2}(R\mathcal{W})^{2L-2}\sqrt{n})}.\end{aligned}$$

Since $\mathcal{W}\leqslant N_\theta$ and noting L is of constant layer ($c\log(d+2)$, cf proposition 3.1), we may bound the log term by

$$\begin{aligned}&\log(2R\cdot\sqrt{d}L^2\rho_0\rho_1^{L-1}\eta_1\eta_0^{L-2}(R\mathcal{W})^{2L-2}\sqrt{n})\\&\quad\leqslant 2L\log(R)+2L\log N_\theta+\log n+\log(dL^2\rho_0\rho_1^L\eta_1\eta_0^L)\\&\quad\leqslant 2L(\log(nRN_\theta)+C_0),\end{aligned}$$

with the constant C_0 depending on d, L, ρ_0, ρ_1, η_0 and η_1. Substituting this bound directly yields

$$\begin{aligned}\mathcal{R}_n(\mathcal{G})&\leqslant\frac{4}{\sqrt{n}}+\frac{12}{\sqrt{n}}\sqrt{d}R^L(\rho_1N_\theta)^{L-1}\sqrt{N_\theta}\sqrt{2L}\left(\sqrt{\log n}+\sqrt{\log R}+\sqrt{\log N_\theta}+\sqrt{C_0}\right)\\&\leqslant C_1\frac{R^LN_\theta^L(\sqrt{\log n}+\sqrt{\log R}+\sqrt{\log N_\theta})}{\sqrt{n}},\end{aligned}$$

where the constant $C_1>0$ depends on d, L, ρ_1, ρ_0, η_0, and η_1 at most polynomially. Combining the preceding results gives the desired bound for $\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}|\mathcal{L}_1(u)-\widehat{\mathcal{L}}_1(u)|\right]$. □

Remark 3.4. Now we specialize the result to two popular choices of the activation function, i.e., $\rho=1/(1+\mathrm{e}^{-x})$ and $\rho=(\mathrm{e}^x-\mathrm{e}^{-x})/(\mathrm{e}^x+\mathrm{e}^{-x})$. It can be verified that for both activation

functions, there holds $\rho_0 = \rho_1 = \eta_0 = \eta_1 = 1$, and both are exponential PU admissible of type $(j, 0)$ for any $j \in \mathbb{N}$.

Next we bound the statistical error $\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_2(u) - \widehat{\mathcal{L}}_2(u)|]$. Given an L-layer neural network class $\mathcal{N}$, we define an associated function class

$$\mathcal{H} = \left\{h : \partial\Omega \subset [0,1]^d \to \mathbb{R} \text{ such that } h(x) = |Tu(x) - g(x)|, \quad \forall\, x \in \partial\Omega, \text{ with } u \in \mathcal{N}\right\}.$$

Lemma 3.6. *Let the activation function ρ satisfy conditions (a) and (b) in definition 3.1. Then for $h_\theta, \tilde{h}_{\tilde{\theta}} \in \mathcal{H}$, there hold*

$$\begin{aligned}\|h_\theta - \tilde{h}_{\tilde{\theta}}\|_{L^\infty(\partial\Omega)} &\leqslant 2L\rho_0\mathcal{W}^L(R\eta_0)^{L-1}\|\theta - \tilde{\theta}\|_{\ell^\infty},\\ \|h_\theta\|_{L^\infty(\partial\Omega)} &\leqslant \|g\|_{L^\infty(\partial\Omega)} + 2\rho_0 R\mathcal{W}.\end{aligned}$$

Proof. Let $r = \|\theta - \tilde{\theta}\|_{\ell^\infty}$. By the definition of $\mathcal{H}$, there exist two neural networks $f^{(L)}$ and $\tilde{f}^{(L)}$ (with parameters θ and $\tilde{\theta}$, respectively) such that $h = |Tf^{(L)} - g|$ and $\tilde{h} = |T\tilde{f}^{(L)} - g|$. Next we show that the map from $\theta \to h_\theta$ is Lipschitz. Indeed, by the triangle inequality, we have

$$\begin{aligned}\|h_\theta - \tilde{h}_{\tilde{\theta}}\|_{L^\infty(\partial\Omega)} &= \||Tf^{(L)} - g| - |T\tilde{f}^{(L)} - g|\|_{L^\infty(\partial\Omega)}\\ &\leqslant \|Tf^{(L)} - T\tilde{f}^{(L)}\|_{L^\infty(\partial\Omega)}.\end{aligned}$$

By the definitions of $f^{(L)}$ and $\tilde{f}^{(L)}$, the triangle inequality, and the bound (3.7), we have

$$\begin{aligned}\|f^{(L)} - \tilde{f}^{(L)}\|_{L^\infty(\partial\Omega)} &= \left\|\left(\sum_{j=1}^{d_{L-1}} W_j^{(L)} f_j^{(L-1)} + b^{(L)}\right) - \left(\sum_{j=1}^{d_{L-1}} \tilde{W}_j^{(L)} \tilde{f}_j^{(L-1)} + \tilde{b}^{(L)}\right)\right\|_{L^\infty(\partial\Omega)}\\ &\leqslant |b^{(L)} - \tilde{b}^{(L)}| + \sum_{j=1}^{d_{L-1}}\left[|W_j^{(L)} - \tilde{W}_j^{(L)}|\|f_j^{(L-1)}\|_{L^\infty(\partial\Omega)}\right.\\ &\quad \left. + \left|\tilde{W}_j^{(L)}|\|f_j^{(L-1)} - \tilde{f}_j^{(L-1)}\|_{L^\infty(\partial\Omega)}\right]\right.\\ &\leqslant r + r\rho_0\mathcal{W} + R\mathcal{W}\cdot 2(L-1)\rho_0\eta_0^{L-1}\mathcal{W}^{L-1}R^{L-2}r\\ &\leqslant 2L\rho_0 r\mathcal{W}^L(\eta_0 R)^{L-1}.\end{aligned}$$

This shows the first estimate. Similarly, we deduce

$$\begin{aligned}\|f^{(L)}\|_{L^\infty(\partial\Omega)} &= \left\|\sum_{j=1}^{d_{L-1}} W_j^{(L)} f_j^{(L-1)} + b^{(L)}\right\|_{L^\infty(\partial\Omega)}\\ &\leqslant \sum_{j=1}^{d_{L-1}} |W_j^{(L)}|\|f_j^{(L-1)}\|_{L^\infty(\partial\Omega)} + |b^{(L)}|\\ &\leqslant R + \rho_0 R\mathcal{W} \leqslant 2\rho_0 R\mathcal{W}.\end{aligned}$$

This and the triangle inequality imply

$$\sup_{h\in\mathcal{H}}\|h\|_{L^\infty(\partial\Omega)} \leqslant \sup_{f^{(s)}\in\mathcal{N}}\|f^{(L)}\|_{L^\infty(\partial\Omega)} + \|g\|_{L^\infty(\partial\Omega)}.$$

This completes the proof of the lemma. □

Next we bound the statistical error $\mathbb{E}_{Z_n}[\sup_{u\in\mathcal{N}}|\mathcal{L}_2(u) - \widehat{\mathcal{L}}_2(u)|]$.

Proposition 3.3. *The following estimate holds*

$$\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}|\mathcal{L}_2(u) - \widehat{\mathcal{L}}_2(u)|\right] \leqslant C_2\gamma\frac{RN_\theta^{\frac{3}{2}}(\sqrt{\log n} + \sqrt{\log R} + \sqrt{\log N_\theta})}{\sqrt{n}},$$

where the constant C_2 depends on $\sqrt{L}$, ρ_0, η_0^L and $\mathcal{B}$.

Proof. The proof technique is similar to proposition 3.2. First, similar to lemma 3.2, we can derive

$$\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}|\mathcal{L}_2(u) - \widehat{\mathcal{L}}_2(u)|\right] \leqslant 2\gamma|\partial\Omega|\mathfrak{R}_n(\mathcal{H}). \tag{3.13}$$

By lemma 3.6, with $\Lambda' := 2L\mathcal{W}^L(\eta_0 R)^{L-1}\rho_0$, for any $h_\theta, \tilde{h}_{\tilde{\theta}} \in \mathcal{H}$, we have

$$\|h_\theta - \tilde{h}_{\tilde{\theta}}\|_{L^\infty(\partial\Omega)} \leqslant \Lambda'\|\theta - \tilde{\theta}\|_\infty.$$

This and lemma 3.5 directly lead to

$$\log\mathcal{C}\left(\mathcal{H}, \|\cdot\|_{L^\infty(\partial\Omega)}, \epsilon\right) \leqslant \log\mathcal{C}\left(\Theta, \|\cdot\|_{\ell^\infty}, \frac{\epsilon}{\Lambda'}\right) \leqslant N_\theta\log\left(\frac{2R\Lambda'}{\epsilon}\right). \tag{3.14}$$

Similarly, with $\mathcal{B} := \|g\|_{L^\infty(\partial\Omega)}$, by lemma 3.6, we may take $M' = \mathcal{B} + 2\rho_0 R\mathcal{W}$. Using the estimate (3.14) in the refined Dudley's formula from lemma 3.3 with $\delta = \frac{1}{\sqrt{n}}$ yields

$$\begin{aligned}
\mathfrak{R}_n(\mathcal{H}) &\leqslant \frac{4}{\sqrt{n}} + \frac{12}{\sqrt{n}}\int_{\frac{1}{\sqrt{n}}}^{M'}\sqrt{N_\theta\log\left(\frac{2R\Lambda'}{\epsilon}\right)}\mathrm{d}\epsilon \\
&\leqslant \frac{4}{\sqrt{n}} + \frac{12}{\sqrt{n}}M'\sqrt{N_\theta}\sqrt{\log(2R\Lambda'\sqrt{n})} \\
&\leqslant \frac{4}{\sqrt{n}} + \frac{12}{\sqrt{n}}(\mathcal{B} + 2\rho_0 R\mathcal{W})\sqrt{N_\theta}\sqrt{\log(2R\cdot 2L\mathcal{W}^L(\eta_0 R)^{L-1}\rho_0\sqrt{n})}.
\end{aligned}$$

Since $\mathcal{W} \leqslant N_\theta$, $L \geqslant 1$, we have

$$\log(RL\mathcal{W}^L(\eta_0 R)^{L-1}\rho_0\sqrt{n}) \leqslant L\log R + L\log N_\theta + \log n + C_0,$$

with the constant C_0 depending on L, ρ_0 and η_0. Substituting this bound yields

$$\begin{aligned}\mathfrak{R}_n(\mathcal{H}) &\leqslant \frac{4}{\sqrt{n}} + \frac{12}{\sqrt{n}}(\mathcal{B}+2\rho_0 R\mathcal{W})\sqrt{N_\theta}\sqrt{L}\left(\sqrt{\log R}+\sqrt{\log N_\theta}+\sqrt{\log n}+\sqrt{C_0}\right)\\ &\leqslant C_2\frac{RN_\theta^{\frac{3}{2}}(\sqrt{\log n}+\sqrt{\log R}+\sqrt{\log N_\theta})}{\sqrt{n}},\end{aligned}$$

where the constant C_2 depends on L, ρ_0, η_0 and $\mathcal{B}$ at most polynomially. Combining the preceding results gives the desired error bound for $\mathbb{E}_{Z_n}\left[\sup_{u\in\mathcal{N}}|\mathcal{L}_2(u)-\widehat{\mathcal{L}}_2(u)|\right]$. □

Finally we state the main result of the section, i.e., the generalization error bound.

Theorem 3.2. *Let the minimizer u^* to the functional $\mathcal{L}$ satisfy $u^*\in W^{2,1}(\Omega)$, and ρ be exponential/polynomial PU-admissible. Then for any $\epsilon>0$, there exists a neural network class given by*

$$\begin{cases}\mathcal{N}\left(c_1\log(d+2), c_2\epsilon^{-\frac{d}{1-\mu}}, c_3\epsilon^{-\frac{4+6d}{1-\mu}}\right), & \text{if } \rho \text{ is exponential PU-admissible},\\ \mathcal{N}(c_1\log(d+2), c_2\epsilon^{-d}, c_3\epsilon^{-(4+6d)}), & \text{if } \rho \text{ is polynomial PU-admissible},\end{cases}$$

with ρ being the activation function, such that when trained with

$$\begin{cases}n_1 = O\left(\epsilon^{-2-\frac{2c_1(4+7d)\log(d+2)}{1-\mu}-\epsilon'}\right), \quad n_2 = O\left(\epsilon^{-2-\frac{8+15d}{1-\mu}-\epsilon'}\right), & \text{if } \rho \text{ is exponential PU admissible},\\ n_1 = O(\epsilon^{-2-2c_1(4+7d)\log(d+2)-\epsilon'}), \quad n_2 = O(\epsilon^{-2-(8+15d)-\epsilon'}), & \text{if } \rho \text{ is polynomial PU admissible},\end{cases}$$

training points ($\epsilon'>0$ arbitrarily small), and an optimization algorithm $\mathcal{A}$ that well trains the neural network with parameters $\theta_{\mathcal{A}}$, the generalization error between the optimal network approximation $u_{\theta_{\mathcal{A}}}$ and u^* is bounded by

$$\mathcal{L}(u_{\theta_{\mathcal{A}}})-\mathcal{L}(u^*)\leqslant C\gamma\epsilon,$$

where the constant $C>0$ depends on ρ_0, ρ_1, η_0, η_1, α_1 and d.

Proof. Fix an arbitrary $\epsilon>0$. Then the choice of the neural network and proposition 3.1 imply

$$\mathcal{E}_{\text{approx}}\leqslant C(\alpha_1, C_{\text{em}})\gamma\epsilon.$$

Meanwhile, it follows from propositions 3.2 and 3.3, and the inequality $L>1$ that with n_1 sampling points in the domain Ω and n_2 sampling points on the boundary $\partial\Omega$, there holds

$$\begin{aligned}\mathcal{E}_{\text{stats}} &\leqslant C_1\frac{R^L N_\theta^L(\sqrt{\log n_1}+\sqrt{\log R}+\sqrt{\log N_\theta})}{\sqrt{n_1}}\\ &\quad + C_2\gamma\frac{RN_\theta^{\frac{3}{2}}(\sqrt{\log n_2}+\sqrt{\log R}+\sqrt{\log N_\theta})}{\sqrt{n_2}} := \mathrm{I}_1+\mathrm{I}_2.\end{aligned}$$

Now we discuss the case of ρ being exponential PU admissible, and the other case follows analogously. Substituting the network parameters $L=c_1\log(d+2)$, $N_\theta=c_2\epsilon^{-\frac{d}{1-\mu}}$ and $R=$

$\epsilon^{-\frac{4+6d}{1-\mu}}$ into the above estimate for $\mathcal{E}_{\text{stats}}$, we have

$$\mathrm{I}_1 \leqslant C_1' \frac{\epsilon^{-\frac{c_1\ \log(d+2)(4+7d)}{1-\mu}} \left(\sqrt{\log\, n_1} + \sqrt{\log\left(\epsilon^{-\frac{4+6d}{1-\mu}}\right)} + \sqrt{\log\left(\epsilon^{-\frac{d}{1-\mu}}\right)} \right)}{\sqrt{n_1}},$$

where the constant C_1' depends on C_1, c_2, c_3 and d. Then by choosing n_1 to be $O(\epsilon^{-2-\frac{2c_1\ \log(d+2)(4+7d)}{1-\mu}-\epsilon'})$, with a small $\epsilon' > 0$, and using the fact that the function $x^{-\nu}\log x$ is uniformly bounded over $[1,\infty)$ for any $\nu > 0$, we deduce $\mathrm{I}_1 \leqslant C_1''\epsilon$. Similarly, we derive

$$\mathrm{I}_2 \leqslant C_2'\gamma \frac{\epsilon^{-\frac{8+15d}{2(1-\mu)}} \left(\sqrt{\log\, n_2} + \sqrt{\log\left(\epsilon^{-\frac{4+6d}{1-\mu}}\right)} + \sqrt{\log\left(\epsilon^{-\frac{d}{1-\mu}}\right)} \right)}{\sqrt{n_2}},$$

where the constant C_2' depends on C_1, c_2, c_3, and d. Thus the choice $n_2 = O(\epsilon^{-2-\frac{8+15d}{1-\mu}-\epsilon'})$ yields $\mathrm{I}_2 \leqslant C_2''\epsilon$. Consequently, we arrive at

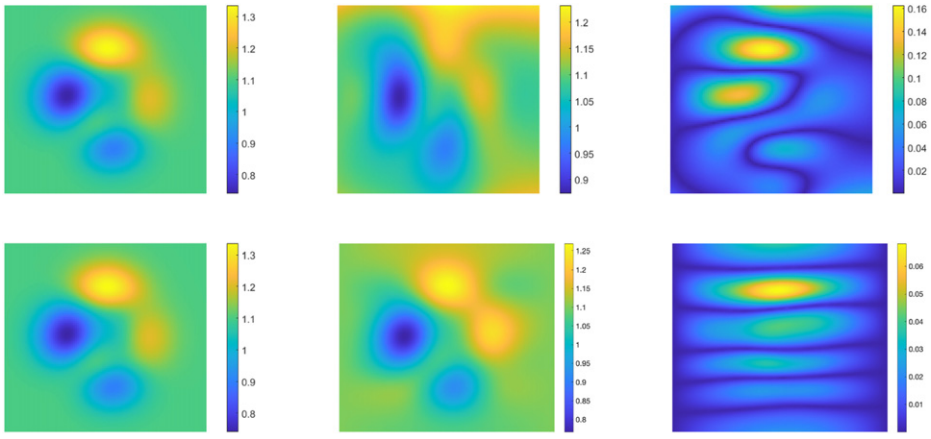
$$\mathcal{E}_{\text{stats}} \leqslant C_1''\epsilon + C_2''\gamma\epsilon = (C_1'' + C_2''\gamma)\epsilon.$$

Then the assertion follows from lemma 3.1, since the optimization error $\mathcal{E}_{\text{opt}}$ is assumed to be small. □

Remark 3.5. Theorem 3.2 indicates that the generalization error can be made arbitrarily small, by choosing the neural network sufficiently wide and trained with sufficiently many training points. The convergence rate is dependent on the numbers of training points (n_1 and n_2), and domain dimension d. It is also observed that the number n_2 of boundary training points can be taken to be much smaller than the number of training points in the domain. Note that in the analysis, γ is taken to be a fixed constant, which can be large. The analysis indicates that the corresponding statistical error can be much reduced by taking a large n_2, but the approximation error on the boundary term behaves in a different way.

4. Numerical experiments and discussions

Now we demonstrate the performance of the proposed algorithm. The activation ρ is taken to be tanh. Unless otherwise specified, the neural network is chosen to have nine layers and 811 parameters in total. The training is conducted with $n_1 = 10\,000$ interior training points and $n_2 = 4000$ boundary training points ($n_2 = 1000$ for example 3), and Huber constant $\zeta = 0.01$. The weighing parameter γ is taken to be $\gamma = 100$ and $\gamma = 10$ for examples 1, 2 and 3, respectively. The resulting empirical loss is minimized by ADAM [39], with a learning rate 8×10^{-4} (for 5000 epochs) and 1×10^{-4} (for 10 000 epochs and 5000 epochs) for examples 1, 2 and 3, respectively. Similar results can be obtained by other optimizers, e.g., L-BFGS [15]. Throughout, the domain Ω is taken to be the unit square $\Omega = (0,1)^2$, and we maintain an almost two-to-one voltage potential g on the boundary given by $g(x,y) = y$, which ensures that the current density magnitude a does not vanish on a set of positive Lebesgue measure in 2D [54]. All computations are performed on TensorFlow 1.15.0 using Intel Core i7-11700K Processor with 16 CPUs.



(a) $\sigma^\dagger$ (b) $\hat{\sigma}$ (c) $|\hat{\sigma} - \sigma^\dagger|$

Figure 2. The reconstructions for example 4.1 with exact data, obtained by the neural network approach (top) and the iterative algorithm of Nachman *et al* (bottom).

We first solve problem (1.1) using MATLAB PDE toolbox, and then compute the exact data $a^\dagger$. The noisy data a^δ is generated by adding Gaussian random noise pointwise as

$$a^\delta(x) = a^\dagger(x) + \delta \cdot a^\dagger(x)\xi(x),$$

where $\delta \geqslant 0$ denotes the (relative) noise level, and the random variable $\xi(x)$ follows the standard Gaussian distribution. In the presence of data noise, computing σ directly via the formula $\sigma = \frac{a^\delta}{|\nabla u|}$ is ill-advised, since the perturbation in a^δ is inherited by σ. To partly overcome the issue, we denoise the data a^δ at the beginning of step (ii) of the algorithm (cf section 2.2) using a feedforward network with nine layers and each hidden layer with 10 neurons, following the idea of deep image prior [68]. Denoising is also employed in the iterative algorithm (cf section 2), without which it is observed to be fairly unstable, since it does not include any regularization directly in the formulation to overcome the inherent ill-posedness of the inverse problem.

We measure the accuracy of the reconstruction $\hat{\sigma}$ (with respect to the exact conductivity $\sigma^\dagger$) by the relative L^2 error $e(\hat{\sigma})$ over the domain Ω (or the subdomain $\Omega' \subset \Omega$ for partial data), defined by

$$e(\hat{\sigma}) = \|\sigma^\dagger - \hat{\sigma}\|_{L^2}/\|\sigma^\dagger\|_{L^2}.$$

The first example is concerned with recovering a smooth conductivity $\sigma^\dagger$ with four modes [54].

Example 4.1. In this example, taken from [54], the conductivity $\sigma^\dagger$ is a four-mode function: $\sigma^\dagger(x, y) = 1.1 + 0.3(\alpha(x, y) - \beta(x, y) - \gamma(x, y))$, with $\alpha(x, y) = 0.3(1 - 3(2x - 1))^2 \mathrm{e}^{-9(2x-1)^2-(6y-2)^2}$, $\beta(x, y) = (\frac{3(2x-1)}{5} - 27(2x - 1)^3 - (3(2y - 1))^5)\mathrm{e}^{-9(2x-1)^2-9(2y-1)^2}$, and $\gamma(x, y) = \mathrm{e}^{-(3(2x-1)+1)^2-9(2y-1)^2}$.

First we show the reconstruction performance. Figure 2 shows the recovered conductivity $\hat{\sigma}$ for exact data and the error $|\hat{\sigma} - \sigma^\dagger|$, along with the results by the iterative approach (cf

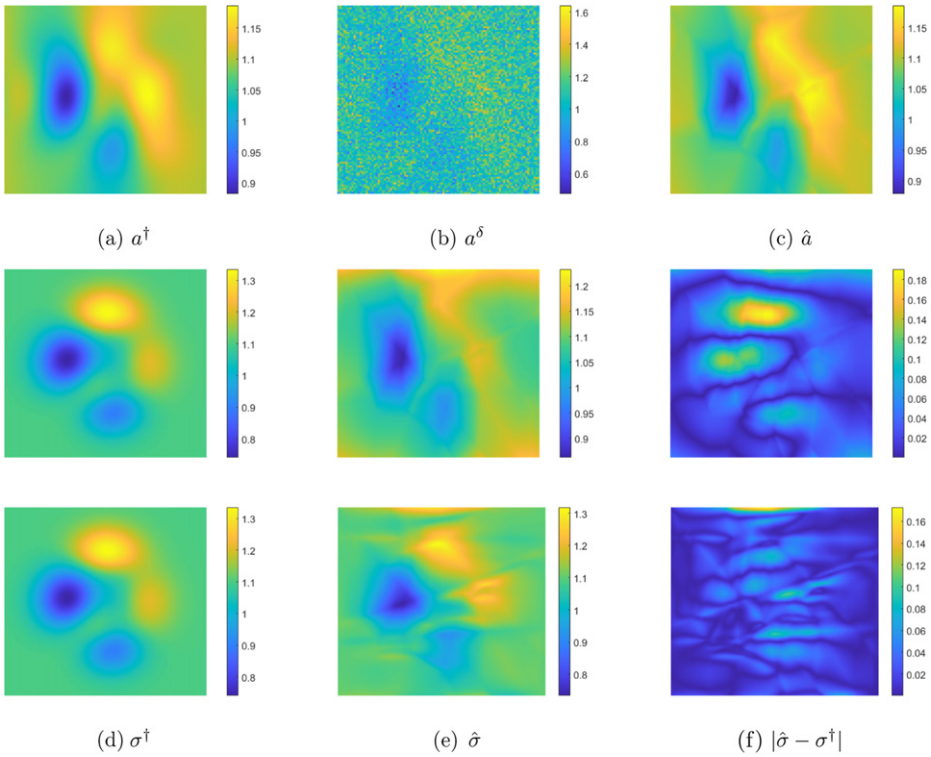
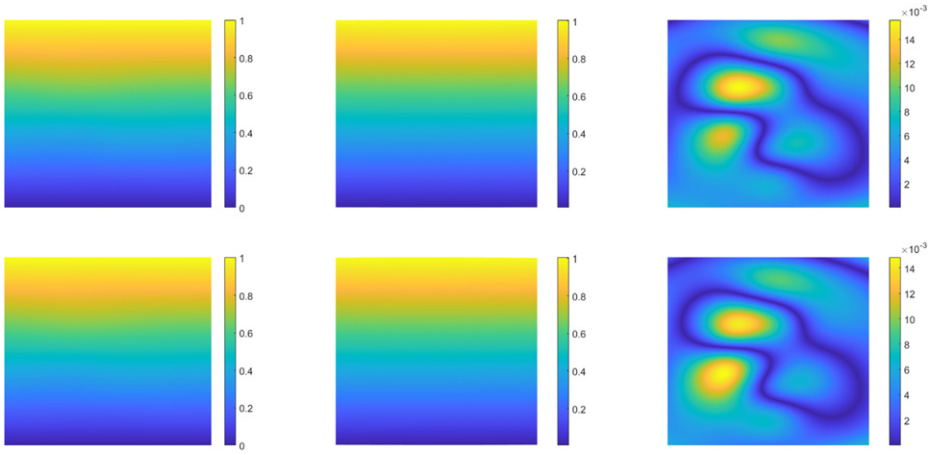


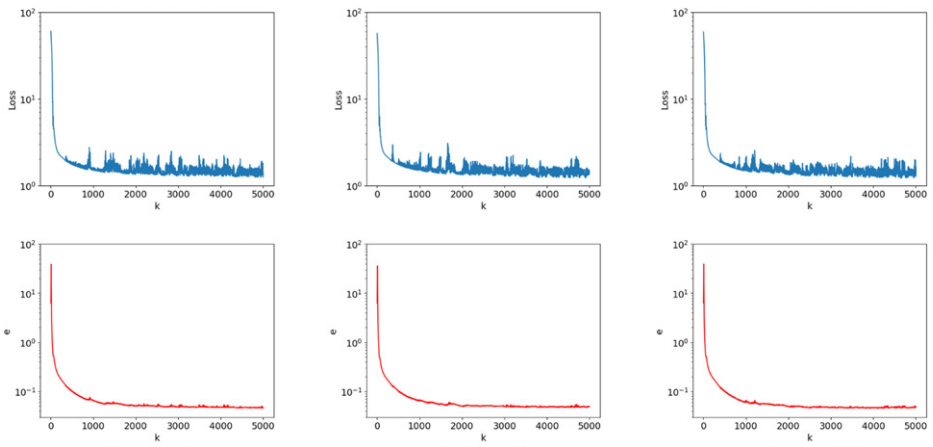
Figure 3. Noisy data a^δ versus denoised data for example 4.1 with $\delta = 10\%$ noise (top), and the reconstructions by the neural network approach (middle) and the iterative algorithm at the 14th iterations (bottom).

section 2). The error plots show that the neural network approximation has largest error in regions near the top-bottom edges, and that the attainable accuracy is inferior to that by the iterative algorithm (which can be made arbitrarily accurate for exact data, since the algorithm converges to the exact conductivity $\sigma^\dagger$ [54]). This accuracy limitation is attributed to the optimization error; see the discussions below. For the data with 10% noise, denoising using neural networks is quite effective in recovering the current density magnitude a, cf figure 3, concurring with the empirical success for deep image prior [68]. It is worth noting that for noisy data a^δ, denoising alone is insufficient to ensure the convergence of the iterative algorithm, which is only guaranteed for admissible data pairs. Thus the iterative algorithm requires careful early stopping, in order to obtain the best possible reconstruction, and a few extra iterations can greatly deteriorate the reconstruction quality. To the best of our knowledge, a provably convergent stopping rule for the algorithm is still unavailable. Hence, in the numerical experiments, we have chosen the optimal iteration index so that the error is smallest. In the proposed approach, the neural network learns the direct solution u from noisy a^δ, and it is observed to be very robust with respect to the presence of noise, cf figure 4. More surprisingly, the approach seems to be fairly stable in the iteration index, cf figure 5 below, and additional iterations do not lead to much deteriorated reconstructions, despite the fact that the employed neural network has high expressivity for approximating rather irregular functions and thus in principle might be susceptible to severe over-fitting.



(a) $u^\dagger$ (b) $\hat{u}$ (c) $|\hat{u} - u^\dagger|$

Figure 4. The reconstructions of u for example 4.1 with exact data (top) and with data with $\delta = 10\%$ noise (bottom), obtained by the neural network approach.



(a) $\delta = 0\%$ (b) $\delta = 1\%$ (c) $\delta = 10\%$

Figure 5. The convergence of the empirical loss and the L^2-relative error $e(\hat{\sigma})$ versus training epoch k for example 4.1 at various noise levels.

In the neural network approach, there are various problem/algorithmic parameters influencing the overall performance, e.g., number of training points (n_1 and n_2), network parameters (width, depth, and activation function) and noise level δ. However, a comprehensive guidance for properly choosing these parameters is still completely missing. Instead, we explore the issue empirically. Tables 1 and 2 show the relative L^2-error of the recovered voltage $\hat{u}$ and conductivity $\hat{\sigma}$, respectively, at different noise levels and different n_1. The algorithm is observed to be very robust with respect to the presence of data noise, and the reconstruction remains fairly accurate even for up to 10% data noise. This contrasts sharply with more traditional optimization based approaches. However, there is also an accuracy limitation of the approach, i.e., the

Table 1. The L^2-relative error of the recovered u vs δ and n_1.

n_1	δ		
	0%	1%	10%
4000	1.73×10^{-2}	9.98×10^{-3}	1.06×10^{-2}
6000	9.57×10^{-3}	1.50×10^{-2}	1.02×10^{-2}
8000	9.95×10^{-3}	9.95×10^{-3}	9.83×10^{-3}
10 000	1.23×10^{-2}	1.51×10^{-2}	9.50×10^{-3}

Table 2. The variation of the relative L^2 error $e(\hat{\sigma})$ with respect to various parameters.

(a) e vs n_1 and δ

n_1	δ		
	0%	1%	10%
4000	4.83×10^{-2}	4.79×10^{-2}	4.80×10^{-2}
6000	4.82×10^{-2}	5.06×10^{-2}	4.70×10^{-2}
8000	4.89×10^{-2}	4.82×10^{-2}	4.75×10^{-2}
10 000	4.68×10^{-2}	4.91×10^{-2}	4.70×10^{-2}

(b) e vs γ and ζ

γ	ζ		
	0.01	0.1	1
10	4.99×10^{-2}	5.06×10^{-2}	4.71×10^{-2}
100	4.70×10^{-2}	4.81×10^{-2}	4.79×10^{-2}
1000	4.79×10^{-2}	4.79×10^{-2}	4.87×10^{-2}
10 000	4.79×10^{-2}	4.79×10^{-2}	4.92×10^{-2}

(c) e vs L and $\mathcal{W}$

L	$\mathcal{W}$		
	10	20	40
2	4.17×10^{-2}	4.17×10^{-2}	4.20×10^{-2}
4	4.31×10^{-2}	4.08×10^{-2}	4.26×10^{-2}
6	4.67×10^{-2}	4.14×10^{-2}	4.30×10^{-2}
9	4.70×10^{-2}	4.50×10^{-2}	4.73×10^{-2}

(d) e vs n_1 and n_2

n_2	n_1			
	4000	6000	8000	10 000
40	8.11×10^{-2}	9.04×10^{-2}	6.96×10^{-2}	7.21×10^{-2}
400	4.79×10^{-2}	4.75×10^{-2}	5.06×10^{-2}	4.81×10^{-2}
1000	4.66×10^{-2}	4.57×10^{-2}	4.88×10^{-2}	4.70×10^{-2}
4000	4.69×10^{-2}	4.64×10^{-2}	4.63×10^{-2}	4.78×10^{-2}

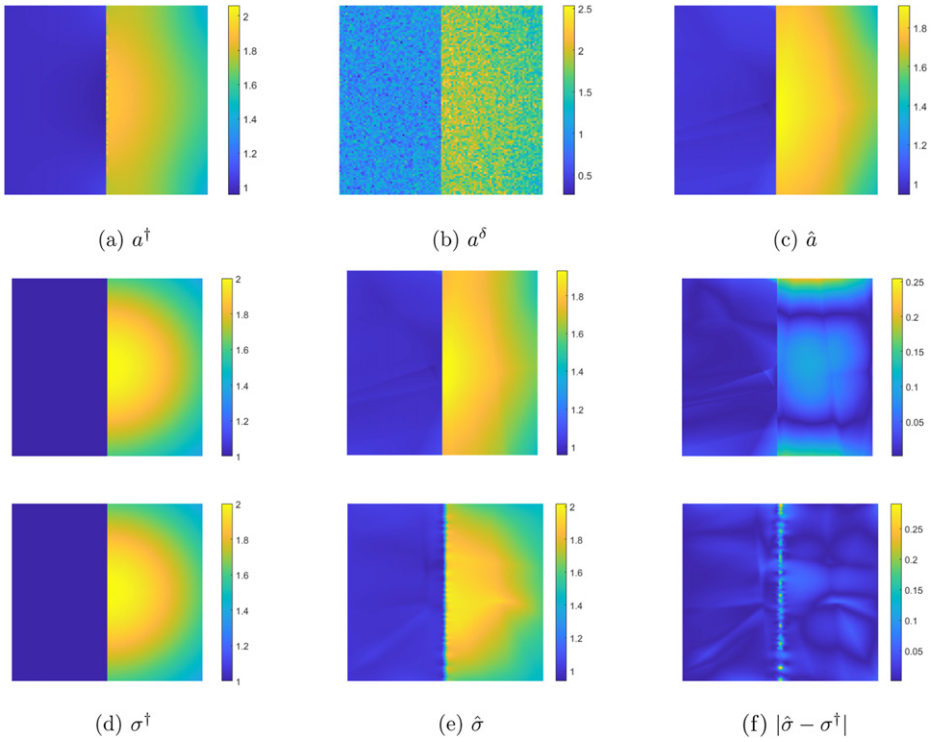
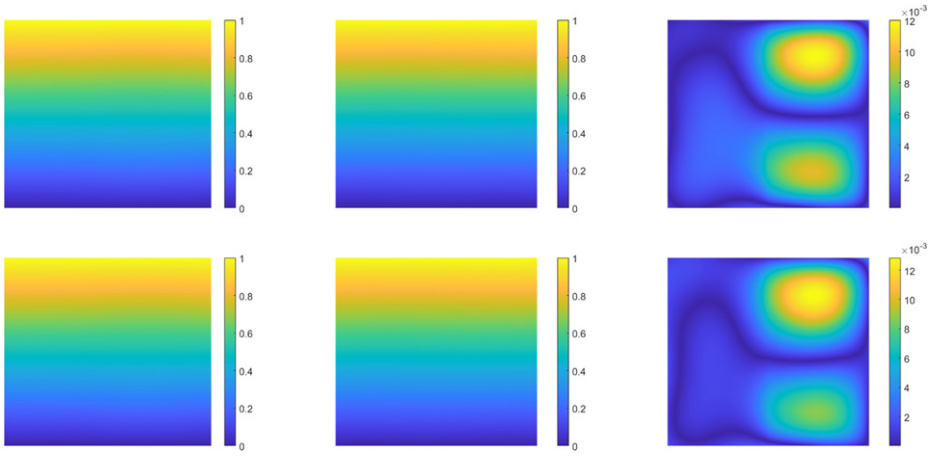


Figure 6. Noisy data a^δ versus denoised data for example 4.2 with $\delta = 10\%$ noise (top), and the reconstructions obtained by the neural network (middle) and the iterative algorithm (bottom) at six iterations.

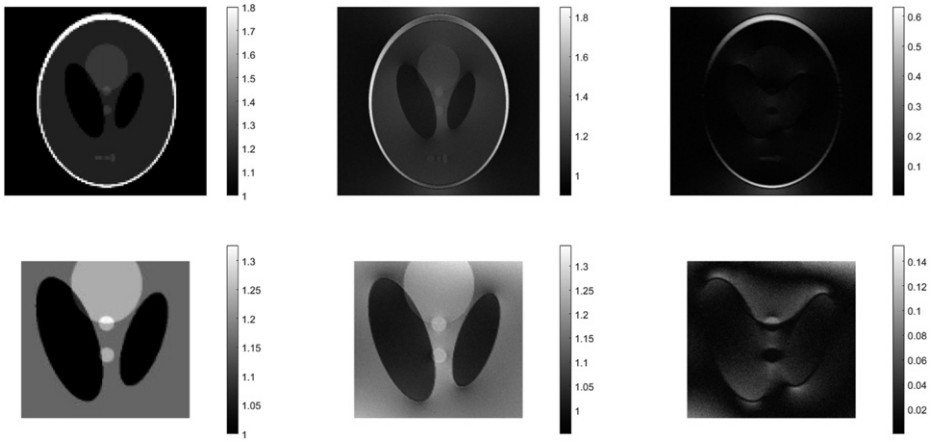
reconstruction cannot be made arbitrarily accurate for exact data $a^\dagger$. This is attributed to the optimization error, which has also been observed across a broad range of solvers based on neural networks [22, 58]. Table 2 shows that the error $e(\hat{\sigma})$ of the recovered conductivity $\hat{\sigma}$ does not vary much with various parameters, e.g., different network architectures. This agrees with the convergence behavior of the optimization algorithm in figure 5: it is largely independent of the noise level δ, and the value of the loss eventually stagnates at a certain level, so is for the reconstruction error $e(\hat{\sigma})$. Thus, the optimization error seems dominating when the noise level δ is low. In particular, further iterations do not affect much the accuracy of the reconstructions. Although not presented, a similar convergence behavior is also observed for much larger neural networks. Of course, if the neural network is vastly expressive and the optimization algorithm continues running for many iterations, it is expected and also numerically observed that overfitting eventually will kick in, due to the lack of explicit regularization, necessitating the use of early stopping or explicit regularization then. These studies show the typical behavior of neural network based approaches, i.e., high-robustness to the data noise and the low sensitivity to the stopping iteration index.

Last we briefly comment on the computational expense. Due to the high non-convexity of the empirical loss $\widehat{\mathcal{L}}_\gamma(\theta)$ (in θ), a global optimizer is often challenging to obtain. The standalone optimizers, e.g., ADAM/L-BFGS, often take hundreds of iterations to reach convergence, cf figure 5. Thus, overall the neural network approach appears less efficient than the iterative algorithm when the direct problem is solved using the standard Galerkin finite element method,



(a) $u^\dagger$ (b) $\hat{u}$ (c) $|\hat{u} - u^\dagger|$

Figure 7. The reconstructions of u for example 4.2 without (top) and with (bottom) $\delta = 10\%$ noise in the data, obtained by the neural network approach.



(a) $\sigma^\dagger$ (b) $\hat{\sigma}$ (c) $|\hat{\sigma} - \sigma^\dagger|$

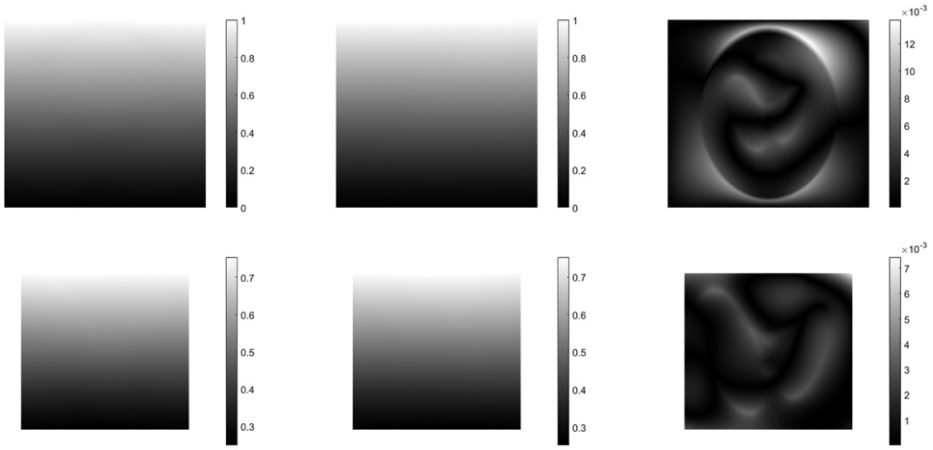
Figure 8. The reconstructions for example 4.3 for full data (top) and partial data with 1% noise.

for which there are highly customized and thus very efficient linear solvers. One important issue is to accelerate the neural network approach.

The second example is concerned with recovering a discontinuous conductivity $\sigma^\dagger$.

Example 4.2. The exact conductivity $\sigma^\dagger$ is $\sigma^\dagger(x, y) = 1 + \chi_{\{x>0.5\}}\, \mathrm{e}^{-2((x-0.5)^2+(y-0.5)^2)}$, where χ_S denotes the characteristic function of the set S.

We present reconstructions for the data with 10% noise. The results by the neural network approach and the iterative one in figure 6 indicate that the reconstructions by the two algorithms



(a) $u^\dagger$ (b) $\hat{u}$ (c) $|\hat{u} - u^\dagger|$

Figure 9. The reconstructions of u for example 4.3 for full data (top) and partial data (bottom), with $\delta = 10\%$ noise.

are of very similar qualities. The error plots indicate that for both approaches, the error is mainly along the discontinuous interface. Quantitatively, the relative L^2 error of the conductivity by the neural network approach is 3.68×10^{-2}, which is of almost no difference when compared to that for the noiseless case (3.99×10^{-2}). This clearly shows the remarkable robustness of the approach for noisy data. These observations fully agree with that for the recovery of the voltage u for exact and noisy data in figure 7: visually there is no difference between the two cases.

The last example is concerned with recovering the Shepp–Logan CT phantom.

Example 4.3. In this example, the exact conductivity $\sigma^\dagger$ is a piecewise constant function corresponding to the standard Shepp–Logan CT image. The intensity of the image is rescaled to a conductivity distribution σ ranging from 1 to 1.8 S m^{-1}.

In this example, for the reconstruction of σ, we consider 1% noise, since the current density magnitude a is highly challenging for denoising, due to the low contrast of conductivity in different regions (within the range from 1 to 1.8). The reconstructions of the conductivity for data with 1% noise in figure 8 is nearly identical with that for exact data (which is not shown). It only tends to be less accurate near the top-bottom edges of the outer circle, where the exact conductivity $\sigma^\dagger$ undergoes big sudden jumps. This observation agrees with the previous examples. Nevertheless, the learning of the neural network at step (a) of the algorithm (cf section 2.2) is not affected much by high noise levels: even for up to 10% noise, the recovered voltage u remains highly accurate, cf figure 9, confirming the remarkable robustness of the neural network approach with respect to data noise.

Last, we examine the case of partial interior data, i.e. with a on a subdomain $\Omega' \subset \Omega$. Then the population loss $\mathcal{L}'_\gamma(u)$ is given by

$$\mathcal{L}'_\gamma(u) = \int_{\Omega'} a|Du| + \gamma \int_{\partial\Omega} a|u - g|\mathrm{d}s.$$

This functional is then discretized by neural networks, but with random sampling points in the subdomain Ω'. In this case, we reconstruct only the conductivity distributions inside Ω'. In the experiment, we take Ω' to be a square region inside the outer circle. The reconstructions for data with 1% noise in figure 8 show that the network can accurately recover the conductivity values from partial data apart from the regions near the outer circle. This shows the feasibility of the approach for partial data, corroborating existing theoretical results [49]. Interestingly, even with 10% noise in the data, the recovery of u remains very accurate, cf figure 9, which again shows the robustness of the approach with respect to data noise.

5. Conclusion

In this work we have developed a direct and novel neural network based reconstruction technique for imaging the conductivity distribution from the magnitude of the internal current density. The reconstruction problem was formulated as a relaxed weighted least-gradient problem, whose minimizer was then approximated by standard fully connected feedforward neural networks. We have also provided a preliminary analysis for the convergence rate of the generalization error, which provides guidelines for properly choosing the depth, width, the total number of parameters of neural networks, and the number of training points in order to achieve the desired convergence rate. The performance and distinct features of the proposed approach were illustrated on a wide range of numerical experiments.

The excellent performance of the neural network based algorithm motivates further research, for which there are several interesting directions. First, the numerical findings suggest that the neural network reconstruction is highly robust with respect to noise. This is commonly attributed to the implicit bias induced by the neural network architecture (e.g., deep image prior [68]) as well as the optimizer. However, the precise characterization of the implicit bias within the context or the mechanism behind the robustness remains mysterious. Second, the relative approximation errors for the neural network reconstructed conductivities are usually only of order 10^{-2}, even for relatively large neural networks. This appears to be suboptimal, in view of the approximation capacity of DNNs. The experiments indicate that the source of error might be attributed to the optimization aspect: the optimizer may have only found a local minimizer due to the complex landscape, and may be unable to reach a global optimizer. Then one natural question is how to achieve better approximation by choosing optimization algorithms different from stand-alone optimizers, e.g., L-BFGS, SGD and Adam. Note that these algorithms often take many iterations to reach convergence, and acceleration strategies are highly desired for better computational efficiency. Third, it is interesting to extend the convergence analysis to related models, e.g., complete electrode model for CDII or other imaging modalities with variational formulations. Fourth and last, one highly acclaimed feature of approaches based on DNNs is that they may hold significant potentials to overcome the notorious curse of dimensionality when the solution satisfies certain favorable properties, e.g., lying in Barron space [43]. It is thus of much interest to extend the analysis and numerics to the high-dimensional setting.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

Bangti Jin https://orcid.org/0000-0002-3775-9155
Xiliang Lu https://orcid.org/0000-0002-7592-5994

References

[1] Adesokan B J, Jensen B, Jin B and Knudsen K 2019 Acousto-electric tomography with total variation regularization *Inverse Problems* **35** 035008
[2] Ammari H 2008 *An Introduction to Mathematics of Emerging Biomedical Imaging* (Berlin: Springer)
[3] Anthony M and Bartlett P L 1999 *Neural Network Learning: Theoretical Foundations* (Cambridge: Cambridge University Press)
[4] Babuška I 1973 The finite element method with penalty *Math. Comput.* **27** 221–8
[5] Bal G 2013 Hybrid inverse problems and internal functionals *Inverse Problems and Applications: Inside Out: II* (Cambridge: Cambridge University Press) pp 325–68
[6] Bao G, Ye X, Zang Y and Zhou H 2020 Numerical solution of inverse problems by weak adversarial networks *Inverse Problems* **36** 115003
[7] Bar L and Sochen N 2021 Strong solutions for PDE-based tomography by unsupervised learning *SIAM J. Imaging Sci.* **14** 128–55
[8] Bartlett P L, Foster D J and Telgarsky M 2017 Spectrally-normalized margin bounds for neural networks *31st Conf. Advances in Neural Information Systems* pp 6240–9
[9] Bartlett P L, Harvey N, Liaw C and Mehrabian A 2019 Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks *J. Mach. Learn. Res.* **20** 63
[10] Bartlett P L and Mendelson S 2002 Rademacher and Gaussian complexities: risk bounds and structural results *J. Mach. Learn. Res.* **3** 463–82
[11] Baydin A G, Pearlmutter B A, Radul A A and Siskind J M 2017 Automatic differentiation in machine learning: a survey *J. Mach. Learn. Res.* **18** 153
[12] Berner J, Grohs P and Jentzen A 2020 Analysis of the generalization error: empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black–Scholes partial differential equations *SIAM J. Math. Data Sci.* **2** 631–57
[13] Borcea L 2002 Electrical impedance tomography *Inverse Problems* **18** R99–R136
[14] Bottou L, Curtis F E and Nocedal J 2018 Optimization methods for large-scale machine learning *SIAM Rev.* **60** 223–311
[15] Byrd R H, Lu P, Nocedal J and Zhu C 1995 A limited memory algorithm for bound constrained optimization *SIAM J. Sci. Comput.* **16** 1190–208
[16] Caselles V, Facciolo G and Meinhardt E 2009 Anisotropic Cheeger sets and applications *SIAM J. Imaging Sci.* **2** 1211–54
[17] Cucker F and Smale S 2002 On the mathematical foundations of learning *Bull. Am. Math. Soc.* **39** 1–49
[18] Dissanayake M W M G and Phan-Thien N 1994 Neural-network-based approximations for solving partial differential equations *Commun. Numer. Methods Eng.* **10** 195–201
[19] Duan C, Jiao Y, Lai Y, Li D, Null X L and Yang J Z 2022 Convergence rate analysis for deep Ritz method *Commun. Comput. Phys.* **31** 1020–48
[20] Dudley R M 1967 The sizes of compact subsets of Hilbert space and continuity of Gaussian processes *J. Funct. Anal.* **1** 290–330
[21] Weinan E, Han J and Jentzen A 2022 Algorithms for solving high dimensional PDEs: from nonlinear Monte Carlo to machine learning *Nonlinearity* **35** 278–310
[22] Weinen E and Yu B 2018 The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems *Commun. Math. Stat.* **6** 1–12
[23] Evans L C and Gariepy R F 2015 *Measure Theory and Fine Properties of Functions* (*Textbooks in Mathematics*) revised edn (Boca Raton, FL: CRC Press)
[24] Foster K R and Schwan H P 1989 Dielectric properties of tissues and biological materials: a critical review *Crit. Rev. Biomed. Eng.* **17** 25–104
[25] Gamba H R, Bayford R and Holder D 1999 Measurement of electrical current density distribution in a simple head phantom with magnetic resonance imaging *Phys. Med. Biol.* **44** 281–91

[26] Gühring I and Raslan M 2021 Approximation rates for neural networks with encodable weights in smoothness spaces *Neural Netw.* **134** 107–30
[27] Guo R and Jiang J 2021 Construct deep neural networks based on direct sampling methods for solving electrical impedance tomography *SIAM J. Sci. Comput.* **43** B678–711
[28] Hoell N, Moradifam A and Nachman A 2014 Current density impedance imaging of an anisotropic conductivity in a known conformal class *SIAM J. Math. Anal.* **46** 1820–42
[29] Hoffmann K and Knudsen K 2014 Iterative reconstruction methods for hybrid inverse problems in impedance tomography *Sens. Imaging* **15** 27
[30] Hong Q, Siegel J W and Xu J 2021 *A priori* analysis of stable neural network solutions to numerical PDEs (arXiv:2104.02903)
[31] Ider Y Z and Muftuler L T 1997 Measurement of AC magnetic field distribution using magnetic resonance imaging *IEEE Trans. Med. Imaging* **16** 617–22
[32] Ito K and Jin B 2015 *Inverse Problems: Tikhonov Theory and Algorithms* (New York: World Scientific)
[33] Jerrard R L, Moradifam A and Nachman A I 2018 Existence and uniqueness of minimizers of general least gradient problems *J. Reine Angew. Math.* **2018** 71–97
[34] Jiao Y, Lai Y, Lou Y, Wang Y and Yang Y 2021 Error analysis of deep Ritz methods for elliptic equations (arXiv:2107.14478)
[35] Johannes M and Zeinhofer M 2021 Error estimates for the variational training of neural networks with boundary penalty (arXiv:2103.01007)
[36] Joy M, Scott G and Henkelman M 1989 *In vivo* detection of applied electric currents by magnetic resonance imaging *Magn. Reson. Imaging* **7** 89–94
[37] Khoo Y and Ying L 2019 SwitchNet: a neural network model for forward and inverse scattering problems *SIAM J. Sci. Comput.* **41** A3182–201
[38] Kim S, Kwon O, Seo J K and Yoon J-R 2002 On a nonlinear partial differential equation arising in magnetic resonance electrical impedance tomography *SIAM J. Math. Anal.* **34** 511–26
[39] Kingma D P and Ba J 2015 Adam: a method for stochastic optimization *3rd Int. Conf. Learning Representations* (San Diego 2015)
[40] Lagaris I E, Likas A and Fotiadis D I 1998 Artificial neural networks for solving ordinary and partial differential equations *IEEE Trans. Neural Netw.* **9** 987–1000
[41] Liu H, Jin B and Lu X 2022 Imaging anisotropic conductivities from current densities *SIAM J. Imag. Sci.* (arXiv:2203.02164) (accepted)
[42] Lopez R and Moradifam A 2020 Stability of current density impedance imaging *SIAM J. Math. Anal.* **52** 4506–23
[43] Lu Y, Lu J and Wang M 2021 *A priori* generalization analysis of the deep Ritz method for solving high dimensional elliptic partial differential equations *Conf. Learning Theory PMLR* pp 3196–241
[44] Luo T and Yang H 2020 Two-layer neural networks for partial differential equations: optimization and generalization theory (arXiv:2006.15733)
[45] Mazón J M 2016 The Euler–Lagrange equation for the anisotropic least gradient problem *Nonlinear Anal. R. World Appl.* **31** 452–72
[46] Mendelson S 2003 A few notes on statistical learning theory *Advanced Lectures on Machine Learning* ed S Mendelson and A J Smola (Berlin: Springer) pp 1–40
[47] Moll J S 2005 The anisotropic total variation flow *Math. Ann.* **332** 177–218
[48] Montalto C and Stefanov P 2013 Stability of coupled-physics inverse problems with one internal measurement *Inverse Problems* **29** 125004
[49] Montalto C and Tamasan A 2017 Stability in conductivity imaging from partial measurements of one interior current *Inverse Probl. Imaging* **11** 339–53
[50] Moradifam A, Nachman A and Tamasan A 2018 Uniqueness of minimizers of weighted least gradient problems arising in hybrid inverse problems *Calc. Var.* **57** 6
[51] Moradifam A, Nachman A and Timonov A 2012 A convergent algorithm for the hybrid problem of reconstructing conductivity from minimal interior data *Inverse Problems* **28** 084003
[52] Morimoto T, Kimura S, Konishi Y, Komaki K, Uyama T, Monden Y, Kinouchi D Y and Iritani D T 1993 A study of the electrical bio-impedance of tumors *J. Invest. Surg.* **6** 25–32
[53] Nachman A, Tamasan A and Timonov A 2007 Conductivity imaging with a single measurement of boundary and interior data *Inverse Problems* **23** 2551–63
[54] Nachman A, Tamasan A and Timonov A 2009 Recovering the conductivity from a single measurement of interior data *Inverse Problems* **25** 035014

[55] Nachman A, Tamasan A and Veras J 2016 A weighted minimum gradient problem with complete electrode model boundary conditions for conductivity imaging *SIAM J. Appl. Math.* **76** 1321–43
[56] Nashed Z M and Tamasan A 2011 Structural stability in a minimization problem and applications to conductivity imaging *Inverse Probl. Imaging* **5** 219–36
[57] Pakravan S, Mistani P A, Aragon-Calvo M A and Gibou F 2021 Solving inverse-PDE problems with physics-aware neural networks *J. Comput. Phys.* **440** 110414
[58] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707
[59] Robbins H and Monro S 1951 A stochastic approximation method *Ann. Math. Stat.* **22** 400–7
[60] Schreuder N 2020 Bounding the expectation of the supremum of empirical processes indexed by Hölder classes *Math. Methods Stat.* **29** 76–86
[61] Scott G C, Joy M L G, Armstrong R L and Henkelman R M 1991 Measurement of nonuniform current density by magnetic resonance *IEEE Trans. Med. Imaging* **10** 362–74
[62] Seo J K, Kim K C, Jargal A, Lee K and Harrach B 2019 A learning-based method for solving ill-posed nonlinear inverse problems: a simulation study of lung EIT *SIAM J. Imaging Sci.* **12** 1275–95
[63] Shalev-Shwartz S and Ben-David S 2014 *Understanding Machine Learning: From Theory to Algorithms* (Cambridge: Cambridge University Press)
[64] Sirignano J and Spiliopoulos K 2018 DGM: a deep learning algorithm for solving partial differential equations *J. Comput. Phys.* **375** 1339–64
[65] Srebro N, Sridharan K and Tewari A 2010 Smoothness, low noise and fast rates *Advances in Neural Information Processing Systems* pp 2199–207
[66] Tamasan A and Timonov A 2019 A regularized weighted least gradient problem for conductivity imaging *Inverse Problems* **35** 045006
[67] Tamasan A, Timonov A and Veras J 2015 Stable reconstruction of regular one-harmonic maps with a given trace at the boundary *Appl. Anal.* **94** 1098–115
[68] Ulyanov D, Vedaldi A and Lempitsky V 2020 Deep image prior *Int. J. Comput. Vis.* **128** 1867–88
[69] van de Geer S A 2000 *Applications of Empirical Process Theory* (Cambridge: Cambridge University Press)
[70] Widlak T and Scherzer O 2012 Hybrid tomography for conductivity imaging *Inverse Problems* **28** 084008
[71] Xu J 2020 Finite neuron method and convergence analysis *Commun. Comput. Phys.* **28** 1707–45
[72] Xu K and Darve E 2022 Physics constrained learning for data-driven inverse modeling from sparse observations *J. Comput. Phys.* **453** 110938
[73] Yazdanian H and Knudsen K 2021 Numerical conductivity reconstruction from partial interior current density information in three dimensions *Inverse Problems* **37** 105010