Solving optical tomography with deep learning

Yuwei Fan and Lexing Ying

This paper presents a neural network approach for solving two-dimensional optical tomography (OT) problems based on the radiative transfer equation. The mathematical problem of OT is to recover the optical properties of an object based on the albedo operator that is accessible from boundary measurements. Both the forward map from the optical properties to the albedo operator and the inverse map are high-dimensional and nonlinear. For the circular tomography geometry, a perturbative analysis shows that the forward map can be approximated by a vectorized convolution operator in the angular direction. Motivated by this, we propose effective neural network architectures for the forward and inverse maps based on convolution layers, with weights learned from training datasets. Numerical results demonstrate the efficiency of the proposed neural networks.

1. Introduction

Optical tomography (OT) is a non-invasive method for reconstructing the optical properties of the medium from boundary measurements with harmless near-infrared light. A typical experiment is to illuminate a highly-scattering medium by a narrow collimated beam and measure the light on the surface by an array of detectors [4]. Since it is non-destructive to biological tissues, OT is of great interest in early tumor diagnosis in medicine, such as in brain imaging [10] and breast imaging [27]. Other industrial applications include atmospheric remote sensing [63] and semiconductor etching [21], etc. We refer readers to the review paper [4], the book [5] and references therein for more details of OT.

1.1. Background

The governing equation of the near-infrared light depends on the spatial scale, ranging from Maxwell equations at the microscale, radiative transfer equation (RTE) at the mesoscale, and to diffusion theory at the macroscale [4]. Among them, RTE is the most widely accepted model for light propagation in tissues. Let $\Omega \subset \mathbb{R}^n$ for $n = 2$ or 3 be a bounded Lipschitz domain and $\mathbb{S}^{n-1}$ is the unit sphere in $\mathbb{R}^n$. Define $\Gamma_\pm = \{(x, v) \in \partial\Omega \times \mathbb{S}^{n-1} \mid \pm v \cdot \nu(x) > 0\}$ with $\nu(x)$ to be the outward unit normal to $\partial\Omega$ at x. The specific intensity $\Phi(x, v)$, defined as the intensity of the light at the position x in the direction v, satisfies the following RTE, for $(x, v) \in \Omega \times \mathbb{S}^{n-1}$

$$
\begin{aligned}
v \cdot \nabla\Phi(x, v) + \mu_t(x)\Phi(x, v) &= \mu(x) \int_{\mathbb{S}^{n-1}} \sigma(v \cdot v')\Phi(x, v')\, \mathrm{d}v' + Q(x, v), \\
\Phi(x, v) &= F(x, v), \quad \text{on } \Gamma_-.
\end{aligned}
$$

(1)

The scattering phase function σ satisfies $\int_{\mathbb{S}^{n-1}} \sigma(v \cdot v')\, \mathrm{d}v = 1$. $Q(x, v)$ is the source inside Ω and $F(x, v)$ is the boundary condition specified at Γ_-. In this paper, the internal light source is assumed to be absent, i.e., $Q(x, v) = 0$. The transport coefficient $\mu_t(x) = \mu_a(x) + \mu(x)$ measures the total absorption, including the physical absorption quantified by the term $\mu_a(x)$ and the scattering phenomenon quantified by the term $\mu(x)$. Here we focus on the reconstruction of the scattering coefficient $\mu(x)$ under the assumption that μ_a is a known constant.

The scattering phase function $\sigma(v \cdot v')$ describes the probability for a photon entering a scattering process at the direction of propagation v to leave this process at the direction v'. The most common phase function in OT is the Henyey-Greenstein scattering function [38]

$$
\sigma(v \cdot v') = \frac{1}{|\mathbb{S}^{n-1}|} \frac{1 - g^2}{(1 + g^2 - 2gv \cdot v')^{n/2}}.
$$

(2)

The parameter $g \in (-1, 1)$ defines the shape of the probability density. The case $g = 0$ indicates that the scattering is almost isotropic, whereas the value of g close to 1 indicates the scattering is primarily a forward directed. A typical value in biological tissue is $g = 0.9$.

The boundary condition in (1) guarantees the uniqueness of solutions of the RTE [12]. In most applications, $F(x, v)$ is either a delta function (in v) at direction $v = -\nu(x)$ or an angular-uniform illumination source. In both cases, $F(x, v)$ can be written as an angular independent function $f(x)h(\nu(x) \cdot v)$ for some fixed distribution $h(\cdot)$.

The measurement on the boundary can be angular dependent or independent. Here we focus on the angular independent case, where the measurable quantity is given by

$$(3) \qquad b(x) \equiv \mathcal{B}\Phi(x) \equiv \int_{v \cdot \nu(x) > 0} v \cdot \nu(x)\Phi(x, v)\, \mathrm{d}v.$$

The albedo operator is defined as

$$(4) \qquad \Lambda : H^k(\partial\Omega) \to H^{-k}(\partial\Omega), \quad f(x)\,|_{\partial\Omega} \to b(x)\,|_{\partial\Omega},$$

where $k > 2 + n/2$. We refer the readers to [64] for more details of the albedo operator and the spaces $H^{\pm k}(\partial\Omega)$.

For a given $\mu(x)$, the albedo operator is a linear map, hence there exists a μ-dependent distribution kernel $\lambda(r, s)$ for $r, s \in \partial\Omega$ such that

$$(5) \qquad (\Lambda f)(r) = b(r) = \int_{\partial\Omega} \lambda(r, s)f(s)\, \mathrm{d}S(s).$$

The forward problem for the albedo operator is that, given the scattering coefficient μ, to compute the kernel $\lambda(r, s)$, i.e., $\mu \to \lambda$. The inverse problem, which is central to OT, is to recover the optical scattering coefficient μ in Ω based on the observation data. Typically, the observation data is a collection of pairs $(f, \Lambda f)$ of the boundary illumination source f and the measurable quantity Λf. When the observation data is sufficient, it is reasonable to assume that the kernel λ is known and hence the inverse problem is to recover μ from λ, i.e., $\lambda \to \mu$. The solvability of the inverse problem has been well studied [15, 59, 6, 5]. Since the measurements are angularly integrated, the inverse problem is often sensitive to noise [9, 64]. For example, in the diffusion limit where RTE can be approximated by a diffusion equation, the inverse problem is considered ill-conditioned due to the elliptic nature [4] of the equation. In other cases, the inverse problem can suffer Hölder instability due to its transport nature (see [64] for example).

From a computational perspective, both the forward and inverse problems associated with the albedo operator (4) are numerically challenging. For the forward problem, since the unknown field $\Phi(x, v)$ is a $(2n - 1)$-dimensional function in both the space x and the direction v, direct solution of RTE is quite expensive even for the two-dimensional case. For OT problems, the situation is worse since in each round of measurements the number of RTE solves is equal to the number of light sources. For the inverse problem, the map $\lambda \to \mu$ is often numerically unstable [9, 49] due to

the ill-posedness and the measurement noise. In order to avoid instability, an application-dependent regularization term is often required in order to stabilize the inverse problem; see, for instance, [37, 13, 42, 31, 9]. Algorithmically, the inverse problem is usually solved with iterative methods [37, 34, 31, 62], which often require a significant number of iterations.

1.2. Contributions

In the recent years, deep neural networks (DNNs) have been very effective tools in a variety of contexts and have achieved great successes in computer vision, image processing, speech recognition, and many other artificial intelligence applications [39, 46, 33, 54, 50, 60, 48, 58]. More recently, DNNs have been increasingly used in the context of scientific computing, particularly in solving PDE-related problems [43, 8, 35, 25, 3, 55, 47, 28]. First, since neural networks offer a powerful tool for approximating high-dimensional functions [17], it is natural to use them as an ansatz for high-dimensional PDEs [57, 11, 35, 44, 20]. A second main direction focuses on the low-dimensional parameterized PDE problems, by using the DNNs to represent the nonlinear map from the high-dimensional parameters of the PDE solution [52, 36, 43, 25, 24, 23, 51, 7]. Applying DNNs to inverse problems [45, 40, 41, 2, 53, 61, 26, 56] can be viewed as a particularly important case of this direction.

This paper applies the deep learning approach to the two-dimensional OT problems by representing both the forward and inverse maps using neural network architectures. The starting point of the new architectures is reformulating RTE into an integral form, which allows for writing out explicitly the forward map $\mu \to \lambda$. By applying a perturbative analysis on the forward map followed by reparameterization, we find the forward map contains one-dimensional convolution in the angular direction for the circular tomography geometry. This observation motivates to represent the forward map from 2D coefficient μ to 2D data λ by a *one-dimensional* convolution neural network (with multiple channels). Following the idea of the back-projection method [29], the inverse map $\lambda \to \mu$ can be approximated by reversing the architecture of the forward map followed with a simple two-dimensional neural network. For the test problems being considered, the resulting neural networks have a relatively small number of parameters, thanks to the convolutional structure. This rather small number of parameters allows for rapid and accurate training, even on rather limited data sets, which is friendly for OT problems as solving RTE is computationally quite expensive.

This rest of the paper is organized as follows. The mathematical background on the albedo operator is studied in Section 2. The design and architecture of the DNNs of the forward and inverse maps are discussed in Section 3. Numerical tests are presented in Section 4.

2. Mathematical analysis of the albedo operator

The goal of this section is to make the relationship between the scattering field $\mu(x)$ and the kernel $\lambda(r, s)$ of the albedo operator more explicit. The first step is to reformulate RTE as an equivalent integral equation [12, 22]. Denote by

$$(6) \qquad \mathcal{J}F(x, v) = \exp\left(-\int_0^t \mu_t(x - \tau v)\,\mathrm{d}\tau\right) F(x - tv, v)$$

the extension of boundary values, where $t(x, v)$ is the distance of a photon traveling from x to the domain boundary along the direction $-v$, i.e.,

$$(7) \qquad t(x, v) = \sup\{\tau : x - sv \in \Omega \text{ for } 0 \leq s < \tau\}$$

and $(x - t(x, v)v, v) \in \Gamma_-$. Introduce also the lifting operator

$$(8) \qquad \mathcal{L}Q(x, v) = \int_0^t \exp\left(-\int_0^\tau \mu_t(x - sv)\,\mathrm{d}s\right) Q(x - \tau v, v)\,\mathrm{d}\tau,$$

and the scattering operator

$$(9) \qquad \mathcal{S}\Phi(x, v) = \mu(x) \int_{\mathbb{S}^{n-1}} \sigma(v' \cdot v)\Phi(x, v')\,\mathrm{d}v'.$$

Direct calculations verify that

$$(10) \qquad \begin{aligned} &(v \cdot \nabla + \mu_t)\mathcal{J}F = 0, \\ &(v \cdot \nabla + \mu_t)\mathcal{L}Q = Q, \quad \mathcal{L}Q\mid_{\Gamma_-} = 0. \end{aligned}$$

This indicates that the extension of the boundary value $\mathcal{J}F$ lies in the kernel of the transport operator $v \cdot \nabla + \mu_t$ and the lifting operator is the right inverse of the transport operator. Noticing that the internal source vanishes ($Q = 0$), one can write RTE equivalently in an integral form [12]

$$(11) \qquad \Phi = \mathcal{L}\mathcal{S}\Phi + \mathcal{J}F,$$

which is a Fredholm integral equation of the second kind. The existence and uniqueness of the integral equation is well understood [12, 18] and inverting (11) results in

$$\Phi = (\mathcal{I} - \mathcal{L}\mathcal{S})^{-1}\mathcal{J}F, \tag{12}$$

where $\mathcal{I}$ is the identity operator.

In order to better understand the relationship between the scattering coefficient and the solution, we perform a perturbative analysis for (12). Notice that all the operators $\mathcal{L}$, $\mathcal{S}$ and $\mathcal{J}$ depend the scattering coefficient μ either directly or implicitly through μ_t. Denote the background of the scattering coefficients by μ_0 and introduce the perturbation

$$\tilde{\mu} \equiv \mu - \mu_0.$$

Here we assume that both μ_0 and μ_a are constant. The background of the total absorption coefficient is then $\mu_{t,0} \equiv \mu_a + \mu_0$. In order to carry out the perturbative analysis, we expand the operators $\mathcal{L}, \mathcal{J}, \mathcal{S}$ into terms of different orders of $\tilde{\mu}$:

$$\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_1 + \ldots, \quad \mathcal{J} = \mathcal{J}_0 + \mathcal{J}_1 + \ldots, \quad \mathcal{S} = \mathcal{S}_0 + \mathcal{S}_1 + \ldots \tag{13}$$

where the *background operators* $\mathcal{L}_0$, $\mathcal{S}_0$ and $\mathcal{J}_0$ are independent of $\tilde{\mu}$ while $\mathcal{L}_1$, $\mathcal{S}_1$ and $\mathcal{J}_1$ are all linear in $\tilde{\mu}$. With these new notations, (12) can be reformulated as

$$\Phi = (\mathcal{I} - \mathcal{L}_0\mathcal{S}_0 - \mathcal{L}_1\mathcal{S}_0 - \mathcal{L}_0\mathcal{S}_1 - \ldots)^{-1}(\mathcal{J}_0 + \mathcal{J}_1 + \ldots)F \tag{14}$$

where $\ldots$ stands for higher order terms in $\tilde{\mu}$. Let us introduce $\mathcal{E}_1 = \mathcal{L}_1\mathcal{S}_0 + \mathcal{L}_0\mathcal{S}_1$, which is also first order in $\tilde{\mu}$. When $\tilde{\mu}$ is sufficiently small, one can expand $(\mathcal{I} - \mathcal{L}_0\mathcal{S}_0 - \mathcal{L}_1\mathcal{S}_0 - \mathcal{L}_0\mathcal{S}_1 - \ldots)^{-1} = (\mathcal{I} - \mathcal{L}_0\mathcal{S}_0 - \mathcal{E}_1 - \ldots)^{-1}$ via a Neumann series

$$(\mathcal{I} - \mathcal{L}_0\mathcal{S}_0 - \mathcal{E}_1 - \ldots)^{-1} = \mathcal{G}_0 + \mathcal{G}_0\mathcal{E}_1\mathcal{G}_0 + \ldots. \tag{15}$$

where $\mathcal{G}_0 = (\mathcal{I} - \mathcal{L}_0\mathcal{S}_0)^{-1}$. Putting this back in (14) and keeping only the terms linear in $\tilde{\mu}$, we conclude that the solution of RTE is approximated by

$$\Phi \approx (\mathcal{G}_0\mathcal{J}_0 + \mathcal{G}_0\mathcal{J}_1 + \mathcal{G}_0\mathcal{E}_1\mathcal{G}_0\mathcal{J}_0)F. \tag{16}$$

Combining this with the measurement quantity (3) results in

$$b = \mathcal{B}\Phi = \mathcal{B}(\mathcal{I} - \mathcal{L}\mathcal{S})^{-1}\mathcal{J}F \approx \mathcal{B}(\mathcal{G}_0\mathcal{J}_0 + \mathcal{G}_0\mathcal{J}_1 + \mathcal{G}_0\mathcal{E}_1\mathcal{G}_0\mathcal{J}_0)F. \tag{17}$$

By introducing $b_0 = b \mid_{\mu=\mu_0}$, the boundary measurement obtained with the background scattering coefficient μ_0, it is equivalent to focus on the difference $b - b_0$. This is known as *difference imaging* in medical applications [4] and the formula for the difference is

(18)
$$b - b_0 = \mathcal{B}(\mathcal{I} - \mathcal{LS})^{-1}\mathcal{J}F - \mathcal{B}(\mathcal{I} - \mathcal{L}_0\mathcal{S}_0)^{-1}\mathcal{J}_0F \approx \mathcal{BG}_0\mathcal{J}_1F + \mathcal{BG}_0\mathcal{E}_1\mathcal{G}_0\mathcal{J}_0F.$$

In practical applications, the boundary source can be represented as

(19)
$$F(x, v) = f(x)h(\nu(x) \cdot v).$$

For example, if the boundary source is a laser, $h(\nu(x) \cdot v) = \delta(\nu(x) \cdot v - 1)$; if the source is angular independent, then $h(\nu(x) \cdot v) = 1/|\mathbb{S}^{n-1}|$. Hence, the difference of the albedo operator (4) applied to f is

(20)
$$b - b_0 = (\Lambda - \Lambda_0)\, f = \mathcal{B}(\mathcal{I} - \mathcal{LS})^{-1}\mathcal{J}hf - \mathcal{B}(\mathcal{I} - \mathcal{L}_0\mathcal{S}_0)^{-1}\mathcal{J}_0hf$$
$$\approx \mathcal{BG}_0\mathcal{J}_1hf + \mathcal{BG}_0\mathcal{E}_1\mathcal{G}_0\mathcal{J}_0hf.$$

By setting $f(x)$ to be delta sources, one can extract from $b - b_0$ the kernel

$$\tilde{\lambda} \equiv \lambda - \lambda_0$$

of the difference albedo operator $\Lambda - \Lambda_0$. In order to see $\tilde{\lambda}$ more explicitly, denote the distribution kernel of the operator $\mathcal{G}_0$ by $G_0(x, v, x', v')$, i.e., $\mathcal{G}_0F(x, v) = \int_{\Omega \times \mathbb{S}^{n-1}} G_0(x, v, x', v')F(x', v')\, dx'\, dv'$, and the distribution kernel of the operator $\mathcal{E}_1$ by $E_1(x, v, x', v')$. By defining the operator $\beta[\eta](x, y) = |x - y| \int_0^1 \eta(x + \tau(y - x))\, d\tau$ for any function $\eta(x)$, the operator $\mathcal{BG}_0\mathcal{J}$ can be represented as

(21)
$$\mathcal{BG}_0\mathcal{J}F(x_r) = \int_{\nu(x_r)\cdot v>0} \nu(x_r) \cdot v \int_\Omega \int_{\partial\Omega} G_0\left(x_r, v, x, \widehat{x - x_s}\right) \exp(-\beta[\mu_t](x_s, x))$$
$$F(x_s, \widehat{x - x_s})\, dv\, dx\, dx_s,$$

where $\hat{x} = \frac{x}{|x|}$. Using the approximation

(22)
$$\exp(-\beta[\mu_t](x, y)) = \exp\left(-\beta[\mu_{t,0}](x, y)\right)\exp\left(-\beta[\tilde{\mu}](x, y)\right)$$
$$\approx \exp\left(-\beta[\mu_{t,0}](x, y)\right)\left(1 - \beta[\tilde{\mu}](x, y)\right),$$

from $\tilde{\mu} = \mu - \mu_0 = \mu_t - \mu_{t,0}$, the kernel of the first term $\mathcal{B}\mathcal{G}_0\mathcal{J}_1$ is

(23)
$$d_1(x_r, x_s) = -\int_{\nu(x_r)\cdot v>0} \nu(x_r)\cdot v \int_\Omega G_0\left(x_r, v, x, \widehat{x - x_s}\right)\exp\left(-\beta[\mu_{t,0}](x_s, x)\right)$$
$$\beta[\tilde{\mu}](x_s, x)h(\nu(x_s)\cdot\widehat{x - x_s})\,\mathrm{d}v\,\mathrm{d}x,$$

which is linear in $\tilde{\mu}$ through $\beta[\tilde{\mu}]$. Similarly, the kernel of the second term $\mathcal{B}\mathcal{G}_0\mathcal{E}_1\mathcal{G}_0\mathcal{J}_0$ can be approximated by

(24)
$$d_2(x_r, x_s) = \int_{\nu(x_r)\cdot v>0} \nu(x_r)\cdot v \int_{\Omega^3}\int_{(\mathbb{S}^{n-1})^2} G_0(x_r, v, x_1, v_1)E_1(x_1, v_1, x_2, v_2)$$
$$G_0(x_2, v_2, x_3, \widehat{x_3 - x_s})\exp\left(-\beta[\mu_{t,0}](x_3, x_s)\right)$$
$$h(\nu(x_s)\cdot\widehat{x_3 - x_s})\,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3\,\mathrm{d}v\,\mathrm{d}v_1\,\mathrm{d}v_2,$$

which is also linear in $\tilde{\mu}$ through $E_1(x_1, v_1, x_2, v_2)$. Putting them together, the kernel of the difference of the albedo operator $\Lambda - \Lambda_0$ can then approximated by

(25) $$\tilde{\lambda}(x_r, x_s) \equiv (\lambda - \lambda_0)(x_r, x_s) \approx d(x_r, x_s) \equiv d_1(x_r, x_s) + d_2(x_r, x_s).$$

3. Neural networks for OT

The discussion below focuses on the two-dimensional case, i.e., $n = 2$. For circular tomography geometry, the domain Ω is a unit disk [4, 9, 62, 5]. As illustrated in Figure 1, the light sources are placed on the boundary equidistantly, while the receivers are shifted by a half spacing. The forward problem of OT is to determinate all the outgoing intensity on the receivers when the light source is activated one by one. The measured data is the kernel $\lambda(x_r, x_s)$, where $x_s = (\cos(s), \sin(s))$ with $s = \frac{2\pi k}{N_s}$, $k = 0, \ldots, N_s - 1$ and $x_r = (\cos(r), \sin(r))$ with $r = \frac{(2j+1)\pi}{N_r}$, $j = 0, \ldots, N_r - 1$, where $N_s = N_r$ in the current setup. Both the absorption coefficient μ_a and the background scattering coefficient μ_0 are assumed to be known constants. The inverse problem of OT is to recover the scattering coefficient μ in the domain given the observation data $\lambda(x_r, x_s) - \lambda_0(x_r, x_s)$, where $\lambda_0(x_r, x_s)$ is the measurement data of the medium with scattering coefficient to be μ_0.

3.1. Forward problem of OT

Since the domain Ω is a disk, it is convenient to write the problem in the polar coordinates. Let $x_r = (\cos(r), \sin(r))$, $x_s = (\cos(s), \sin(s))$ and $x =$
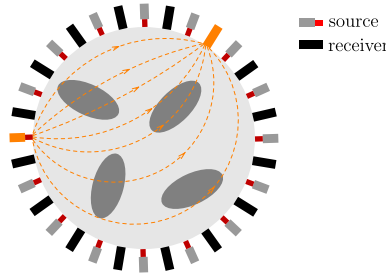
Figure 1: Illustration of the problem setup. The domain is a unit disk and the light sources and the receivers are equidistantly placed on the boundary, with a half spacing shift in between.

$(\rho \cos(\theta), \rho \sin(\theta))$, where $\rho \in [0, 1]$ denotes the radial direction and $r, s, \theta \in [0, 2\pi)$ denotes the angular direction.

3.1.1. Convolution in the angular direction. Figure 2 presents an example of the measurement data $\lambda(x_r, x_s)$ and $\lambda(x_r, x_s) - \lambda_0(x_r, x_s)$. Notice that the main signal concentrates upon the diagonal part and the left-lower and right-upper corners. Due to the circular tomography geometry, it is convenient to "shear" the measurement data by introducing a new angular variable $h = r - s$, where the difference here is understood modulus 2π. As we shall see, this shearing step significantly simplifies the architecture of the NNs. Under the new parameterization, the measurement data is

$$(26) \quad d(h, s) \equiv d(x_{h+s}, x_s) = d((\cos(s + h), \sin(s + h)), (\cos(s), \sin(s))).$$

By also writing $\mu(\rho, \theta) \equiv \mu((\rho \cos(\theta), \rho \sin(\theta)))$ in the polar coordinates, the linear dependence of $d(h, s)$ on $\tilde{\mu}$ in (23) and (24) states that there exists a kernel distribution $K(h, s, \rho, \theta)$ such that

$$(27) \quad d(h, s) = \int_0^1 \int_0^{2\pi} K(h, s, \rho, \theta) \tilde{\mu}(\rho, \theta) \, d\rho \, d\theta.$$

The following proposition states that this can in fact be written as a convolution in the angular direction.

Proposition 1. *There exists a function $\kappa(h, \rho, \cdot)$ periodic in the last argument such that*

$$(28) \quad K(h, s, \rho, \theta) = \kappa(h, \rho, s - \theta).$$

(a) $\lambda(x_r, x_s)$ (b) $(\lambda - \lambda_0)(x_r, x_s)$

(c) $\lambda(x_h, x_s)$ (d) $(\lambda - \lambda_0)(x_h, x_s)$
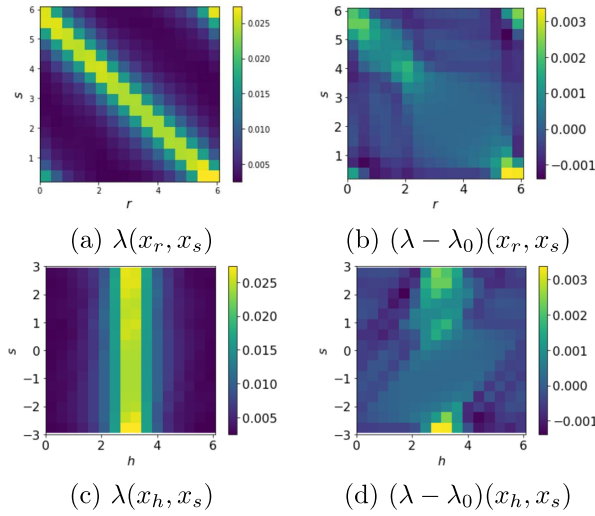
Figure 2: The upper figures are the measurement data $\lambda(x_r, x_s)$ and the difference $\lambda(x_r, x_s) - \lambda_0(x_r, x_s) \approx d(x_r, x_s)$ with respect to the background, respectively. The horizontal and vertical axes are s and r, respectively. The lower figures are shift of their upper figures by $h = r - s$.

The proof of this proposition uses some basic formulas summarized in the following lemma.

Lemma 2. *If $R \in \mathbb{R}^{2 \times 2}$ is a rotation matrix, then*

$$\beta[\mu_0](Rx, Ry) = \beta[\mu_0](x, y), \tag{29}$$

$$G_0(x, v, x', v') = G_0(Rx, Rv, Rx', Rv'). \tag{30}$$

Proof. The definition of β indicates $\beta[\mu_0](x, y) = \mu_0|x - y|$. Thus (29) holds.

Denote the distribution kernel of $\mathcal{L}_0 \mathcal{S}_0$ by $L_0(x, v, x', v')$. Since $\mathcal{G}_0 = (\mathcal{I} - \mathcal{L}_0 \mathcal{S}_0)^{-1}$, we just need to check $L_0(Rx, Rv, Rx', Rv') = L_0(x, v, x', v')$. Notice that the operator $\mathcal{L}\mathcal{S}$ is defined as

$$\begin{aligned}
\mathcal{L}&\mathcal{S}Q(x, v) \\
&= \int_0^t \exp\left(-\beta[\mu_t](x, x - \tau v)\right) \mu(x - \tau v) \int_{\mathbb{S}^1} \sigma(v \cdot v') Q(x - \tau v, v') \, \mathrm{d}v' \, \mathrm{d}\tau \\
&= \int_\Omega \int_{\mathbb{S}^1} \frac{\delta(v - \widehat{x - x'})}{|x - x'|} \exp\left(-\beta[\mu_t](x, x')\right) \mu(x') \sigma(v \cdot v') Q(x', v') \, \mathrm{d}x' \, \mathrm{d}v'.
\end{aligned} \tag{31}$$

Then the kernel L_0 reads

$$L_0(x, v, x', v') = \frac{\delta(v - \widehat{x - x'})}{|x - x'|} \exp\left(-\beta[\mu_{t,0}](x, x')\right) \mu_0 \sigma(v \cdot v').$$

Since $\delta(Rv - R(\widehat{x - x'})) = \delta(v - \widehat{x - x'})$, $Rv \cdot Rv' = v \cdot v'$ and (29), one can directly obtain

$$L_0(Rx, Rv, Rx', Rv') = L_0(x, v, x', v').$$

This completes the proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Proof of Proposition 1. To prove (28), one needs to show that, for any $\rho \in [0, 1)$ and any $h \in [-\pi, \pi)$ and $s, \psi \in [0, 2\pi)$,

$$(32) \qquad d(h, s + \psi) = \int_0^1 \int_0^{2\pi} K(h, s, \rho, \theta)\tilde{\mu}(\rho, \theta + \psi) \, d\rho \, d\theta$$

holds. Notice (25) that d has two parts. We study them one by one. Define the rotation matrix $R = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix}$, then

$$d_1(h, s + \psi) = d_1(Rx_r, Rx_s)$$
$$= -\int_{\nu(Rx_r)\cdot v > 0} \nu(Rx_r) \cdot v \int_\Omega G_0\left(Rx_r, v, x, \widehat{x - Rx_s}\right)$$
$$\times \exp\left(-\beta[\mu_{t,0}](Rx_s, x)\right) \beta[\tilde{\mu}](Rx_s, x) h(\nu(Rx_s) \cdot \widehat{x - Rx_s}) \, dv \, dx.$$

Since Ω is a disk, the integral keeps unchanged if we change of variables as $v \to Rv$ and $x \to Rx$. Using $\nu(Rx_r) \cdot Rv = \nu(x_r) \cdot v$, (30) and (29) to eliminate the rotation and changing the variable again as $Rv \to v$ and $Rx \to x$, we obtain

$$d_1(h, s + \psi) = d_1(Rx_r, Rx_s)$$
$$= -\int_{\nu(x_r)\cdot v > 0} \nu(x_r) \cdot v \int_\Omega G_0\left(x_r, v, x, \widehat{x - x_s}\right)$$
$$\times \exp\left(-\beta[\mu_{t,0}](x_s, x)\right) \beta[\tilde{\mu}](Rx_s, Rx) h(\nu(x_s) \cdot \widehat{x - x_s}) \, dv \, dx.$$

This completes the proof of the d_1 part.

Next we study the second part d_2. Noticing (31), we obtain the kernel distribution $E_1(x, v, x', v')$

$$= \frac{\delta(v - \widehat{x - x'})}{|x - x'|} \exp\left(-\beta[\mu_{t,0}](x, x')\right) \sigma(v \cdot v') \left(-\beta[\tilde{\mu}](x, x') + \tilde{\mu}(x')\right).$$

Using (29), we have

$$E_1(Rx, Rv, Rx', Rv')$$
$$= \frac{\delta(v - \widehat{x - x'})}{|x - x'|} \exp\left(-\beta[\mu_{t,0}](x, x')\right) \sigma(v \cdot v') \left(-\beta[\tilde{\mu}](Rx, Rx') + \tilde{\mu}(Rx')\right).$$

Then using the same technique in the proof of the first part, we can show that (32) also holds for the second part. This completes the proof. $\quad\square$

Proposition 1 shows that K acts on $\tilde{\mu}$ in the angular direction by a convolution, i.e.,

$$(33) \qquad d(h, s) = \int_0^1 (\kappa(h, \rho, \cdot) * \tilde{\mu}(\rho, \cdot))(s) \, d\rho.$$

This effectively reduces the forward map to a family of 1D convolutions, parameterized by ρ and h.

Till now all the analysis is in the continuous space. One can apply a discretization on the RTE (1) by the finite volume method on the space and discrete velocity method on the direction domain [30]. The kernel distribution G_0 and E are replaced by its discrete version. The actual discretization is often problem-dependent and we leave it to Section 4. Here with a slight abuse of notation, we use the same letters to denote the continuous kernels, variables and their discretization. Then the discretization version of (33) is

$$(34) \qquad d(h, s) \approx \sum_\rho (\kappa(h, \rho, \cdot) * \tilde{\mu}(\rho, \cdot))(s).$$

3.1.2. Neural network architecture. The perturbative analysis shows that if $\tilde{\mu}$ is sufficiently small, the forward map $\tilde{\mu}(\rho, \theta) \to \tilde{\lambda}(h, s)$ can be approximated by (34). This indicates that the forward map (34) can be approximated by a convolution layer for small $\tilde{\mu}$. For larger $\tilde{\mu}$, this linear approximation is no longer accurate. In order to extend the neural network for (34) to the nonlinear case, we propose to increase the number of convolution layers and include nonlinear activation functions, as shown in Algorithm

1. Here $\mathsf{Conv1d}[\alpha, w, \mathsf{ReLU}]$ stands for a one-dimensional layer with channel number α, window size w, and activation function as ReLU. Note that because the value of the measurement data ranges in $\mathbb{R}$, no activation function is applied after the last layer. Since the convolution in (34) is global, the architectural parameters are chosen with

$$(35) \qquad w N_{\mathrm{cnn}} \geq N_s$$

so that the resulting network is capable of capturing global interactions. When N_s is large, it is possible that the recently proposed multiscale neural networks, for example MNN-$\mathcal{H}$-net [25], MNN-$\mathcal{H}^2$-net [24], and BCR-net [23], are more efficient for such global interactions. However in order to simplify the presentation, the discussion here sticks to the convolutional layers.

Algorithm 1 Neural network architecture for the forward problem $\tilde{\mu} \to \tilde{\lambda}$.

Require: α, w, $N_{\mathrm{cnn}} \in \mathbb{N}^+$, $\tilde{\mu} \in \mathbb{R}^{N_\rho \times N_\theta}$
Ensure: $\tilde{\lambda} \in \mathbb{R}^{N_h \times N_s}$
 1: $\xi^{(0)} = \tilde{\mu}$ with ρ as the channel direction
 2: **for** k from 1 to $N_{\mathrm{cnn}} - 1$ by 1 **do**
 3: $\quad \xi^{(k)} \leftarrow \mathsf{Conv1d}[\alpha, w, \mathsf{ReLU}](\xi^{(k-1)})$
 4: **end for**
 5: $\tilde{\lambda} \leftarrow \mathsf{Conv1d}[N_h, w, \mathsf{id}](\xi^{(N_{\mathrm{cnn}}-1)})$
 6: return $\tilde{\lambda}$

3.2. Inverse problem of OT

The perturbative analysis shows that if $\tilde{\mu}$ is sufficiently small, the forward map can be approximated by

$$(36) \qquad \tilde{\lambda} \approx K\tilde{\mu},$$

which is the operator notation of the discretization (34). Here $\tilde{\mu}$ is a vector indexed by (ρ, θ), $\tilde{\lambda}$ is a vector indexed by (h, s), and K is a matrix with row indexed by (h, s) and column indexed by (ρ, θ). The filtered back-projection method [29] suggests the following formula to recover $\tilde{\mu}$:

$$(37) \qquad \tilde{\mu} \approx (K^\mathsf{T} K + \epsilon I)^{-1} K^\mathsf{T} \tilde{\lambda}.$$

Since $K^\mathsf{T}\tilde{\lambda}$ can also be written as a family of convolutions

$$(38) \qquad (K^\mathsf{T}\tilde{\lambda})(\rho, \theta) = \sum_h (\kappa(h, \rho, \cdot) * \tilde{\lambda}(h, \cdot))(\theta),$$

the application of K^{T} to $\tilde{\lambda}$ can be approximated with a one-dimensional convolutional neural network, similar to K. For the part $K^{\mathsf{T}}K + \epsilon I$, which can be viewed as a post-processing in the (ρ, θ) space, we implement this with several two-dimensional convolutional layers for simplicity. However, for problems with larger sizes, multiscale neural networks such as [25, 24, 23] can be also used. The resulting architecture for the inverse map is summarized in Algorithm 2 and illustrated in Figure 3

Algorithm 2 Neural network architecture for the inverse problem $\tilde{\lambda} \to \tilde{\mu}$.

Require: $\alpha_1, \alpha_2, w_1, w_2, N_{\mathrm{cnn}_1}, N_{\mathrm{cnn}_2} \in \mathbb{N}^+$, $\tilde{\lambda} \in \mathbb{R}^{N_h \times N_s}$
Ensure: $\tilde{\mu} \in \mathbb{R}^{N_\rho \times N_\theta}$
1: $\zeta^{(0)} = \tilde{\lambda}$ with h as the channel direction
2: **for** k from 1 to N_{cnn_1} by 1 **do**
3:     $\zeta^{(k)} \leftarrow \mathsf{Conv1d}[\alpha_1, w_1, \mathsf{ReLU}](\zeta^{(k-1)})$
4: **end for**
5: $\xi^{(0)} \leftarrow \zeta^{(N_{\mathrm{cnn}_1})}$
6: **for** k from 1 to $N_{\mathrm{cnn}_2} - 1$ by 1 **do**
7:     $\xi^{(k)} \leftarrow \mathsf{Conv2d}[\alpha_2, w_2, \mathsf{ReLU}](\xi^{(k-1)})$
8: **end for**
9: $\tilde{\mu} \leftarrow \mathsf{Conv2d}[1, w_2, \mathsf{id}](\xi^{(N_{\mathrm{cnn}_2}-1)})$
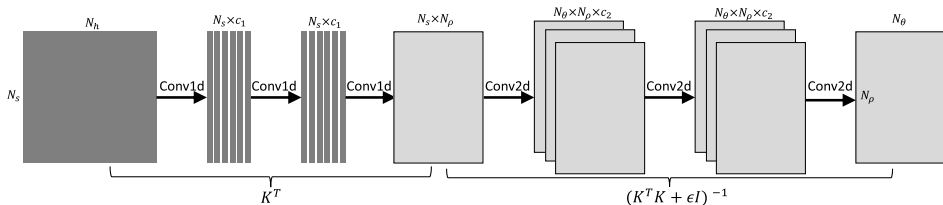10: **return** $\tilde{\mu}$

Figure 3: Neural network architecture for the inverse map of OT.

4. Numerical tests

This section reports the numerical performance of the proposed neural network architectures for the forward and inverse maps.

4.1. Experimental setup

The RTE in (1) is discretized with a finite volume method in x and a discrete velocity method in v. The upwind scheme is used for the convection term and

the composite trapezoidal rule is applied for the integral of the scattering term. The value of $\sigma(v' \cdot v)$ is replaced by its value on the discretization points with a scaling such that its numerical quadrature is 1. The multi-level method proposed in [30] is adopted to solve the discrete system. The domain Ω is partitioned by triangle mesh with 6976 elements and 3553 points. The direction v is uniformly discretized using 32 points. In the polar coordinates, the domain $(\rho, \theta) \in [0, 1] \times [0, 2\pi]$ is partitioned by a uniformly Cartesian mesh with 96×192 points. As a technical note, since Algorithms 1 and 2 are designed for the scattering coefficient in the polar coordinates, the scattering coefficient on the triangle mesh is treated as a piece-constant function and it is further interpolated on to the polar grid.

To mimic the setup of realistic medical applications, Ω is a disc with the radius equal to 20mm and the background scattering and absorption coefficient are 1mm^{-1} and 0.01mm^{-1}, respectively [31, 9, 62]. The parameter g in (2) is set as $g = 0.9$, a typical value of biological tissues. In the experiment, $N_s = 16$ light sources and $N_r = 16$ receivers are equidistantly placed on the boundary of the domain with a half spacing shift (see Figure 1). The source light is an angular independent pointolite, i.e., the s-th light source is $F(x, v) = \delta(x - x_s)$.

The NN is implemented with Keras [14] running on top of TensorFlow [1]. Nadam is chosen as the optimizer [19] and the mean squared error is used as the loss function. The parameters of the network are initialized by Xavier initialization [32]. In the training process, the batch size and the learning rate is firstly set as 16 and 10^{-3} respectively, and the NN is trained 100 epochs. Then we increase the batch size by a factor 2 till to 256 with the learning rate unchanged, and then decrease the learning rate by a factor $10^{1/2}$ to 10^{-5} with the batch size fixed as 256. In each step, we train the NN 50 epochs. The selection of the channel number α, number of convolution layers N_{cnn} and the window size w will be discussed in the numerical results.

4.2. Numerical results

For a fixed scattering coefficient field μ, $\lambda(h, s) = \lambda((\cos(s + h), \sin(s + h)), (\cos(s), \sin(s)))$ stands for the *exact* measurement data solved by numerical discretization of (1). The prediction of the forward NN from μ is denoted by λ^{NN}, while the one of the inverse NN from λ is denoted by μ^{NN}. The accuracy for the forward problem is measured by the relative error in the ℓ^2 norm:

$$(39) \qquad \frac{\|\lambda - \lambda^{\text{NN}}\|_{\ell^2}}{\|\lambda\|_{\ell^2}}.$$

For each experiment, the test error is then obtained by averaging (39) over a given set of test samples. The numerical results presented below are obtained by repeating the training process three times, using different random seeds for the NN initialization.

The scattering coefficient $\mu(x)$ is assumed to be piecewise constant. For each sample $\mu(x)$, we randomly generate N_e ellipses in Ω and set $\mu(x) = 2\text{mm}^{-1}$ in the ellipses and 1mm^{-1} otherwise. For each ellipse, the width and height are sampled from the uniform distributions $\mathcal{U}(0.0075, 0.015)$ and $\mathcal{U}(0.00375, 0.0075)$, respectively, the direction is uniformly random over the unit circle, and the position is uniformly sampled in the disk. It is also required that each ellipse lies in the disk and there is no intersection between each two ellipses. For each test, 10,204 samples $\{(\mu_i, \lambda_i)\}$ are generated with 8192 used for training and the remaining 2048 for testing.

While Algorithms 1 and 2 assume for simplicity that $N_\theta = N_s$, this is often not the case in the experimental setup. To deal with this issue, for the forward problem we first compress μ from $N_r \times N_\theta$ to $\alpha \times N_s$ by a one-dimensional convolution layer with channel number α, window size N_r/N_s, and strides N_r/N_s. For the inverse problem, an interpolation operator for extending the data of size $\alpha \times N_s$ to $N_r \times N_\theta$ is added after the one-dimensional convolution neural networks. In the implementation, the interpolation is implemented by two layers. The first layer interpolates the data of size $\alpha \times N_s$ along with the angular direction to $\alpha \times N_\theta$ by a one-dimensional convolution layer with channel number $\alpha \times N_\theta/N_s$ and window size 1, and a column major reshape. The second layer interpolates the data of size $\alpha \times N_\theta$ along with the radial direction to $N_r \times N_\theta$ by a convolution layer with channel number N_r and window size 1.

4.2.1. Forward problem. The data set is generated with the number of ellipses $N_e = 4$ and the window size w in Algorithm 1 is set to be 5. Multiple numerical experiments are performed to study how the test error depends on the channel number α and the convolution layer number N_{cnn}, with the results presented in Figure 4. As the number of channels increases, the test error first consistently decreases and then saturates. The same is observed for the number of convolution layers. The choices of the hyper-parameters $\alpha = 32$ and $N_{\text{cnn}} = 8$ offers a reasonable balance between accuracy and efficiency. For this specific case, the number of parameters is 7.5×10^4 and the test error is 1.1×10^{-3}. Figure 5 illustrates the NN prediction λ^{NN} and its corresponding references λ of a sample in the test data.
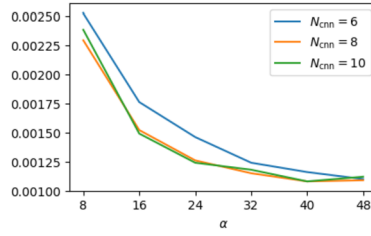
Figure 4: The test error for different channel numbers α and different convolution layer numbers N_{cnn}.

(a) $\lambda(h, s)$

(b) $\lambda^{\mathrm{NN}}(h, s)$

(c) $\lambda(h, s) - \lambda_0(h, s)$
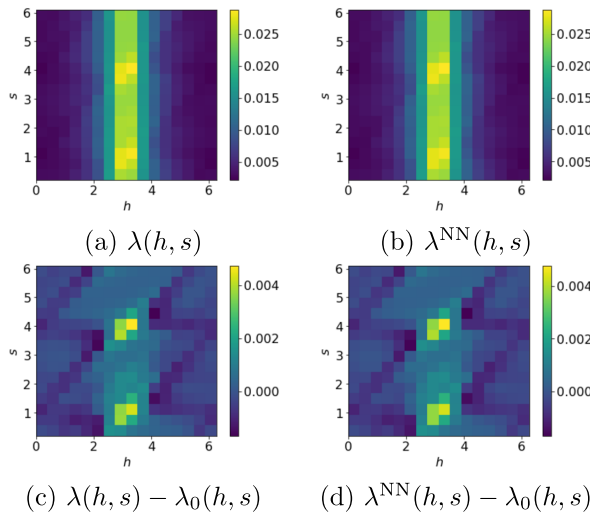
(d) $\lambda^{\mathrm{NN}}(h, s) - \lambda_0(h, s)$

Figure 5: Illustration of a sample in the test data for the forward problem with the number of ellipses $N_e = 4$ in Ω. The channel number $\alpha = 32$ and the convolution layer number $N_{\mathrm{cnn}} = 8$.

4.2.2. Inverse problem. Two date sets corresponding to $N_e = 2, 4$ are generated. The hyper-parameters in Algorithm 2 are set as $(\alpha_1 = 32, w_1 = 5, N_{\mathrm{cnn}_1} = 6)$ and $(\alpha_2 = 4, w_2 = 3 \times 3, N_{\mathrm{cnn}_2} = 5)$ and the number of parameters in NN is 4.8×10^4. To model the uncertainty in the measurement data, we introduce noises to the albedo operator in the data set by defining $\lambda_i^\delta \equiv (1 + Z_i \delta) \lambda_i$, where Z_i is a Gaussian random variable with zero mean and unity variation and δ controls the signal-to-noise ratio. For each noisy level $\delta = 0$, 0.5%, 1%, 2% and 5%, an independent NN is trained and tested with the noisy data set $\{(\lambda_i^\delta, \mu_i)\}$. Note that in our experiments the mean

of $\frac{\|\lambda - \lambda_0\|}{\lambda}$ for all the samples is about 5% and hence the signal-to-noise ratio for the difference $\lambda - \lambda_0$ is almost 100% when the noise level $\delta = 5\%$.
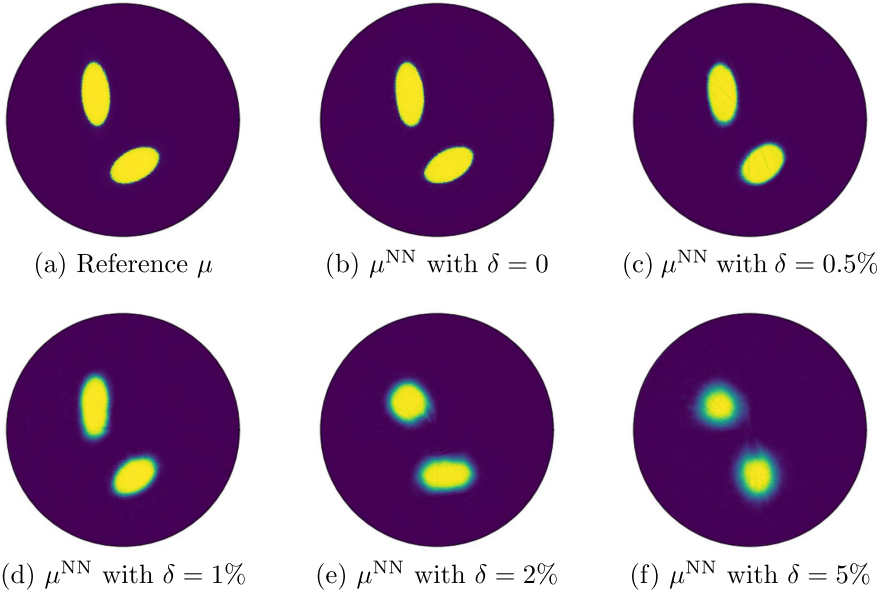

(a) Reference μ (b) μ^{NN} with $\delta = 0$ (c) μ^{NN} with $\delta = 0.5\%$

(d) μ^{NN} with $\delta = 1\%$ (e) μ^{NN} with $\delta = 2\%$ (f) μ^{NN} with $\delta = 5\%$

Figure 6: NN prediction of a sample in the test data for the number of ellipses $N_e = 2$ in Ω and for different noise level $\delta = 0, 0.5\%, 1\%, 2\%, 5\%$.

Figures 6 and 7 show samples in the test data for different noise level δ and different number of ellipses N_e in Ω. When there is no noise in the measurement data, the NN offers an accurate prediction of the scattering coefficient μ, in the position, shape and direction of the ellipses. For the small noise levels, for example $\delta = 0.5\%$ and 1%, the boundary of the shapes in the prediction is blurred while the position and direction of the ellipses are still correct. As the noise level δ increases, the shapes become fuzzy but the position and number of shapes are still correctly predicted. This demonstrates the NN architecture in Algorithm 2 is capable of learning the inverse problem of OT.

To test the generalization performance of the NN, we train the NN using the data set of $N_e = 2$ at a given noise level and test the NN by the data of $N_e = 4$ with the same noise level (and vice versa). The results, summarized in Figure 8, indicate that the NN trained by the data, with two inclusions is capable of recovering the measurement data of the case with four inclusions, and vice versa. This is an indication that the trained NN is capable of predicting beyond the training scenario.
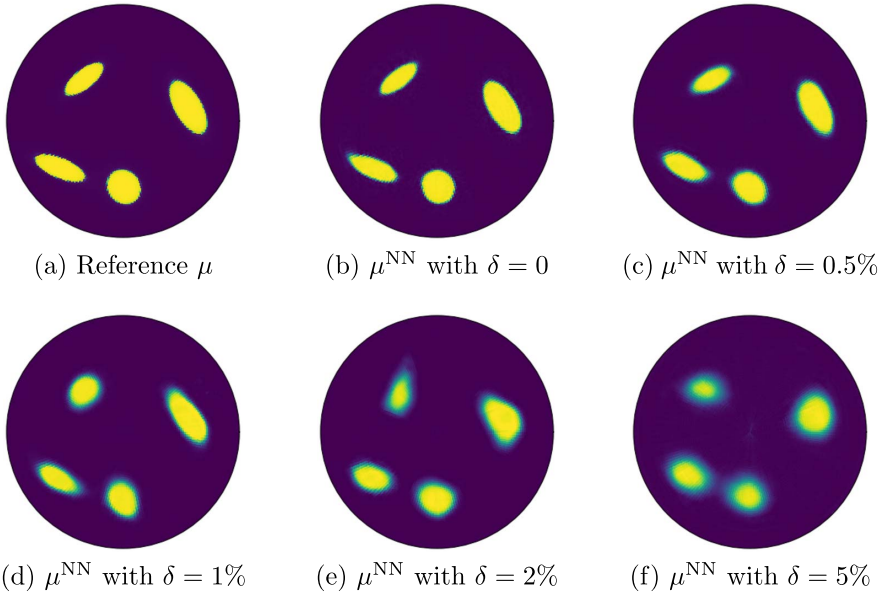
(a) Reference μ (b) μ^{NN} with $\delta = 0$ (c) μ^{NN} with $\delta = 0.5\%$

(d) μ^{NN} with $\delta = 1\%$ (e) μ^{NN} with $\delta = 2\%$ (f) μ^{NN} with $\delta = 5\%$

Figure 7: NN prediction of a sample in the test data for the number of ellipses $N_e = 4$ in Ω and for different noise level $\delta = 0, 0.5\%, 1\%, 2\%, 5\%$.

5. Discussions

This paper presents a neural network approach for OT problems. Mathematically, these NNs approximate the forward and inverse maps between the scattering coefficient and the kernel distribution of the albedo operator. The perturbative analysis, which indicates that the linearized forward map can be represented by a one-dimensional convolution with multiple channels, inspires the design of the NN architectures.

NNs have offered a few clear advantages in approximating the forward and inverse problems. For both the forward and inverse maps, once the NN is trained, applying the map is significantly accelerated as it only involves a single inference with the trained NN. For the inverse problem, two critical issues for more traditional approaches are the choices of the solution algorithm and the regularization term. NNs seem to bypass the algorithm issue by choosing an appropriate architecture and learning the map from the data, and at the same time, identify an appropriate regularization by automatically learning the key features from the training set. Numerical results also demonstrate that the proposed NNs are capable of approximating the forward and inverse maps accurately. However, although empirically
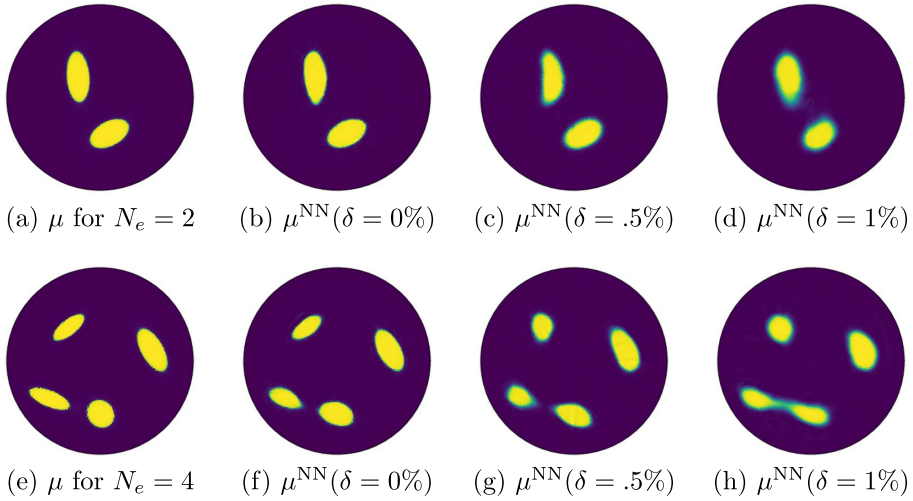
Figure 8: NN generation test. The upper figures: the NN is trained by the data of the number of ellipses $N_e = 4$ in Ω with noise level $\delta = 0$ 0.5% or 1% and test by the data of $N_e = 2$ with the same noise level. The lower figures correspond the case of training data $N_e = 2$ and test data $N_e = 4$.

encouraging, theoretical justification of these advantages require significant work.

The discussion in this paper focuses on the reconstruction of the scattering coefficient. Using a similar analysis, one can extend the work to the reconstruction of the absorption coefficient or both. The analysis in this paper can also extended to the three-dimensional OT problems by leveraging recent work such as [16].

The NN for the inverse map implicitly uses a strong prior: the testing and training data are alike. In the reported numerical examples, the testing and training data are indeed sampled from the same statistical model and therefore the recovery is quite accurate. When the testing and training data are statistically different, this becomes of the problem of distributional shift or transfer learning, which is much harder and left for future work.

Acknowledgments

References

[1] M. Abadi et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017. MR3729789

[3] M. Araya-Polo, J. Jennings, A. Adler, and T. Dahlke. Deep-learning tomography. *The Leading Edge*, 37(1):58–66, 2018.

[4] S. R. Arridge and J. C. Schotland. Optical tomography: forward and inverse problems. *Inverse problems*, 25(12):123010, 2009. MR3460046

[5] G. Bal. Introduction to inverse problems. *Lecture Notes-Department of Statistics and Mathematics, University of Chicago*, 2019.

[6] G. Bal and A. Jollivet. Stability estimates in stationary inverse transport. *Inverse Problems and Imaging*, 2(4):427–454, 2008. MR2558276

[7] L. Bar and N. Sochen. Unsupervised deep learning algorithm for PDE-based forward and inverse problems. *arXiv preprint arXiv:1904.05417*, 2019.

[8] J. Berg and K. Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41, 2018. MR3796894

[9] B. Bi, B. Han, W. Han, J. Tang, and L. Li. Image reconstruction for diffuse optical tomography based on radiative transfer equation. *Computational and mathematical methods in medicine*, 2015, 2015. MR3303278

[10] D. A. Boas, A. M. Dale, and M. A. Franceschini. Diffuse optical imaging of brain activation: approaches to optimizing image sensitivity, resolution, and accuracy. *Neuroimage*, 23:S275–S288, 2004.

[11] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. MR3642415

[12] K. Case and P. Zweifel. Existence and uniqueness theorems for the neutron transport equation. *Journal of Mathematical Physics*, 4(11):1376–1385, 1963. MR0158716

[13] T. F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM journal on scientific computing*, 20(6):1964–1977, 1999. MR1694649

[14] F. Chollet et al. Keras. https://keras.io, 2015.

[15] M. Choulli and P. Stefanov. Reconstruction of the coefficients of the stationary transport equation from boundary measurements. *Inverse Problems*, 12(5):L19, 1996. MR1413417

[16] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018.

[17] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. MR1015670

[18] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology: Volume 6 Evolution Problems II.* Springer-Verlag Berlin Heidelberg, 2000. MR1156075

[19] T. Dozat. Incorporating Nesterov momentum into adam. *International Conference on Learning Representations*, 2016.

[20] W. E and B. Yu. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018. MR3767958

[21] C. Edwards, A. Arbabi, G. Popescu, and L. L. Goddard. Optically monitoring and controlling nanoscale topography during semiconductor etching. *Light: Science & Applications*, 1(9):e30, 2012.

[22] H. Egger and M. Schlottbom. An L^p theory for stationary radiative transfer. *Applicable Analysis*, 93(6):1283–1296, 2014. MR3195888

[23] Y. Fan, C. O. Bohorquez, and L. Ying. BCR-Net: a neural network based on the nonstandard wavelet form. *Journal of Computational Physics*, 384:1–15, 2019. MR3915332

[24] Y. Fan, J. Feliu-Fabà, L. Lin, L. Ying, and L. Zepeda-Núñez. A multiscale neural network based on hierarchical nested bases. *Research in the Mathematical Sciences*, 6(2):21, 2019. MR3921459

[25] Y. Fan, L. Lin, L. Ying, and L. Zepeda-Núñez. A multiscale neural network based on hierarchical matrices. *arXiv preprint arXiv:1807.01883*, 2018. MR4033684

[26] Y. Fan and L. Ying. Solving electrical impedance tomography with deep learning. *arXiv preprint arXiv:1906.03944*, 2019. MR4044725

[27] S. Fantini and A. Sassaroli. Near-infrared optical mammography for breast cancer detection with intrinsic contrast. *Annals of biomedical engineering*, 40(2):398–407, 2012.

[28] J. Feliu-Faba, Y. Fan, and L. Ying. Meta-learning pseudo-differential operators with deep neural networks. *arXiv preprint arXiv:1906.06782*, 2019. MR4065434

[29] T. Feng, P. Edström, and M. Gulliksson. Levenberg–Marquardt methods for parameter estimation problems in the radiative transfer equation. *Inverse Problems*, 23(3):879, 2007. MR2329921

[30] H. Gao and H. Zhao. A fast-forward solver of radiative transfer equation. *Transport Theory and Statistical Physics*, 38(3):149–192, 2009. MR2572432

[31] H. Gao and H. Zhao. Multilevel bioluminescence tomography based on radiative transfer equation part 1: l1 regularization. *Optics Express*, 18(3):1854–1871, 2010. MR2421884

[32] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[33] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. MR3617773

[34] E. Haber, U. M. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse problems*, 16(5):1263, 2000. MR1798355

[35] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018. MR3847747

[36] J. Han, L. Zhang, R. Car, and W. E. Deep potential: A general representation of a many-body potential energy surface. *Communications in Computational Physics*, 23(3):629–639, 2018. MR3869647

[37] M. Hanke. Regularizing properties of a truncated Newton-CG algorithm for nonlinear inverse problems. *Numerical Functional Analysis and Optimization*, 18(9-10):971–993, 1997. MR1485990

[38] L. G. Henyey and J. L. Greenstein. Diffuse radiation in the galaxy. *The Astrophysical Journal*, 93:70–83, 1941.

[39] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[40] S. R. H. Hoole. Artificial neural networks in the solution of inverse electromagnetic field problems. *IEEE transactions on Magnetics*, 29(2):1931–1934, 1993.

[41] H. Kabir, Y. Wang, M. Yu, and Q.-J. Zhang. Neural network inverse modeling and applications to microwave filter design. *IEEE Transactions on Microwave Theory and Techniques*, 56(4):867–879, 2008.

[42] J. P. Kaipio, V. Kolehmainen, M. Vauhkonen, and E. Somersalo. Inverse problems with structural prior information. *Inverse problems*, 15(3):713, 1999. MR1696902

[43] Y. Khoo, J. Lu, and L. Ying. Solving parametric PDE problems with artificial neural networks. *arXiv preprint arXiv:1707.03351*, 2017. MR4253972

[44] Y. Khoo, J. Lu, and L. Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1):1, 2019. MR3887217

[45] Y. Khoo and L. Ying. SwitchNet: a neural network model for forward and inverse scattering problems. *arXiv preprint arXiv:1810.09675*, 2018. MR4018415

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems – Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[47] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider. A theoretical analysis of deep neural networks and parametric PDEs. *arXiv preprint arXiv:1904.00377*, 2019. MR4376560

[48] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(436), 2015.

[49] C. Leng, D. Yu, S. Zhang, Y. An, and Y. Hu. Reconstruction method for optical tomography based on the linearized Bregman iteration with sparse regularization. *Computational and mathematical methods in medicine*, 2015, 2015. MR3396006

[50] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129, 2014.

[51] Y. Li, J. Lu, and A. Mao. Variational training of neural network approximations of solution maps for physical models. *arXiv preprint arXiv:1905.02789*, 2019. MR4074519

[52] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-net: Learning PDEs from data. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3208–3216, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[53] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos. Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Processing Magazine*, 35(1):20–36, 2018.

[54] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik. Deep neural nets as a method for quantitative structure-activity relationships. *Journal of Chemical Information and Modeling*, 55(2):263–274, 2015.

[55] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018. MR3759415

[56] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. MR3881695

[57] K. Rudd and S. Ferrari. A constrained integration (CINT) approach to solving partial differential equations using artificial neural networks. *Neurocomputing*, 155:277–285, 2015. MR3211506

[58] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[59] P. Stefanov et al. Optical tomography in two dimensions. *Methods and Applications of Analysis*, 10(1):001–010, 2003. MR2014159

[60] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[61] C. Tan, S. Lv, F. Dong, and M. Takei. Image reconstruction based on convolutional neural network for electrical resistance tomography. *IEEE Sensors Journal*, 19(1):196–204, 2018.

[62] S. Tong, B. Han, Y. Chen, J. Tang, B. Bi, and R. Gu. RTE-based parameter reconstruction with TV + L1 regularization. *Journal of Computational and Applied Mathematics*, 337:256–273, 2018. MR3766455

[63] C. Weitkamp. *Lidar: range-resolved optical remote sensing of the atmosphere*, volume 102. Springer Science & Business, 2006.

[64] H. Zhao and Y. Zhong. Instability of an inverse problem for the stationary radiative transport near the diffusion limit. *arXiv preprint arXiv:1809.01790*, 2018. MR4007614

Yuwei Fan
Stanford University
Stanford, CA 94305
USA
E-mail address: ywfan1989@gmail.com

Lexing Ying
Stanford University
Stanford, CA 94305
USA
E-mail address: lexing@stanford.edu