REVIEW

Deep learning in photoacoustic tomography: current approaches and future directions

Andreas Hauptmann[a,b,*] **and Ben Cox**[c]
[a]University of Oulu, Research Unit of Mathematical Sciences, Oulu, Finland
[b]University College London, Department of Computer Science, London, United Kingdom
[c]University College London, Department of Medical Physics and Biomedical Engineering, London, United Kingdom

Abstract. Biomedical photoacoustic tomography, which can provide high-resolution 3D soft tissue images based on optical absorption, has advanced to the stage at which translation from the laboratory to clinical settings is becoming possible. The need for rapid image formation and the practical restrictions on data acquisition that arise from the constraints of a clinical workflow are presenting new image reconstruction challenges. There are many classical approaches to image reconstruction, but ameliorating the effects of incomplete or imperfect data through the incorporation of accurate priors is challenging and leads to slow algorithms. Recently, the application of deep learning (DL), or deep neural networks, to this problem has received a great deal of attention. We review the literature on *learned image reconstruction*, summarizing the current trends and explain how these approaches fit within, and to some extent have arisen from, a framework that encompasses classical reconstruction methods. In particular, it shows how these techniques can be understood from a Bayesian perspective, providing useful insights. We also provide a concise tutorial demonstration of three prototypical approaches to *learned image reconstruction*. The code and data sets for these demonstrations are available to researchers. It is anticipated that it is in *in vivo* applications—where data may be sparse, fast imaging critical, and priors difficult to construct by hand—that DL will have the most impact. With this in mind, we conclude with some indications of possible future research directions. [DOI: 10.1117/1.JBO.25.11.112903]

Keywords: photoacoustic tomography; learned image reconstruction; deep learning; neural networks; data-driven methods; *in vivo* imaging.

Paper 200199VR received Jun. 30, 2020; accepted for publication Sep. 24, 2020; published online Oct. 26, 2020.

1 Introduction

The potential of biomedical photoacoustic tomography (PAT) to obtain high-resolution images based on optical absorption and, moreover, provide images that depend quantitatively on endogenous or exogeneous molecular contrast, has resulted in rapidly growing interest in the modality. For example, the ability to obtain accurate, spatially resolved, estimates of blood oxygenation would have significant impact both clinically and for preclinical applications.

There are two aspects to PAT image reconstruction: an acoustic inversion from the measured acoustic time series to the initial acoustic pressure distribution[1,2] and a spectroscopic optical inversion to recover optical absorption coefficients or quantities derived from them.[3] The acoustic inverse problem can be solved exactly in closed form in the ideal circumstance that complete data are available and the medium has a uniform and known sound speed. In most practical scenarios, however, there are divergences from this ideal case, e.g., heterogeneities in the sound speed or bandlimited detection over an incomplete set of measurement points, making the acoustic inversion challenging. (The use of linear arrays for *in vivo* imaging is a case in point.) When, in addition, a solution is required to the optical inversion, the image reconstruction task becomes more challenging still, as the forward operator is nonlinear. Iterative model-based approaches

*Address all correspondence to Andreas Hauptmann, andreas.hauptmann@oulu.fi

have been devised that manage this greater complexity by providing a flexible way to frame the problem and incorporate prior knowledge of the kind of solution expected.[4–6] However, such approaches, while appealing, are typically computationally intensive and time-consuming, which precludes their use in many applications.

In contrast to purely model-based approaches, data-driven techniques, and in particular deep learning (DL), are increasingly widely used for tomographic image reconstruction.[7–12] These techniques, which primarily originate from computer vision and are known to excel at segmentation and classification tasks, are frequently treated as "black boxes." This is widely considered undesirable in biomedical imaging and inverse problems, and recent work has started to provide insights into why certain network architectures work well for certain tasks,[13–15] and also to provide justifications for the use of DL approaches in the solution of inverse problems including image reconstruction. We will refer to the application of DL within the image reconstruction pipeline as *learned image reconstruction*.

The rising interest in learned image reconstruction has led to a transition from classical analytical methods to such data-driven approaches. Although much of this work has focused on established imaging modalities such as MRI[9,16,17] and CT,[7,8,18] this transition is also clearly discernible in the recent literature on PAT image reconstruction. In this paper, we will review the recent work done in this area and place these approaches into a broader context by drawing connections to classical analytical reconstructions. We also provide a tutorial style introduction to the use of DL in PAT image reconstruction, including describing and demonstrating several different approaches for learned image reconstruction. Code is available for these examples, free to download, allowing researchers to reproduce them, and providing them with a starting point for their own learned reconstructions.

PAT is a particularly suitable area in which to review these methods for several reasons. There is a very active experimental community interested in a wide range of applications, from data-intensive, large-scale, 3D imaging to 2D high-frame rate uses. This results in a wide variety of different approaches for data collection and presents many different challenges in the reconstruction pipeline, including those of limited data, computationally expensive forward operators, uncertainty in model parameters, and the lack of training data; the latter especially a problem for *in vivo* applications. This leads to a final point, which is that the community is not only in a transition from classical to data-driven approaches, but also in a transition from proof-of-concept studies to applying the techniques in challenging clinical and preclinical scenarios. Indeed, these two transitions may prove to be symbiotic: data-driven approaches are rarely needed in proof-of-concept studies, in which complete data are available and time is not of the essence, but many of the problems facing *in vivo* use are not easily tackled within a classical framework. We hope that by describing a framework for learned reconstructions and by presenting an overview of the diverse work done, this review can provide guidance for possible future directions for image reconstruction as PAT transitions from the bench to the clinic.

1.1 *Scope of Tutorial Review*

There are multiple ways in which DL could be used within the context of PAT, so to keep this review to a reasonable length it is necessary to limit its scope. This review will concentrate on DL as applied to tomographic reconstruction in photoacoustics. In other words, the focus will be on using DL networks, sometimes in combination with classical approaches, to reconstruct photoacoustic (PA) images from projections (which here are acoustic time series), this includes pre- and postprocessing approaches with the intent to improve reconstruction quality. With this as the focus, there are several applications of DL to PA imaging that must regrettably wait for a future review. First, this review will be limited to PA *tomography* and will not cover the use of DL in relation to PA *microscopy* (PAM), the principal difference being, for our purposes, that in PAT it is necessary to reconstruct the image from a set of measured projections but in PAM the image can be measured directly. Second, this review will not cover work where DL approaches have been used subsequent to a final reconstruction. This includes, for example, applications where DL has been used to segment or classify PAT images, or regions of images, into, say, diseased or healthy. Third, this review will not cover the use of DL to make diagnostic judgments, e.g., to answer questions such as "Does this image indicate diabetes, rheumatism, cancer, etc.?"

Papers on DL for PAT reconstruction are currently appearing at a steady rate, and we anticipate that trend will continue. This review attempts to cover all relevant papers or preprints appearing up to the end of June 2020.

2 Forward and Inverse Problems in Photoacoustic Tomography

2.1 *PAT Forward Problems*

2.1.1 *Physics of photoacoustic signal generation*

The *PA effect* is the name given to the phenomenon by which the absorption of an optical pulse generates an acoustic pulse. A light pulse incident on soft biological tissue will be scattered around in the tissue, eventually either leaving the tissue or being absorbed by absorbing molecules in the tissue, known as chromophores (hemoglobin being one of the most important). The energy of the excited chromophores is then converted into heat. This all occurs on a timescale ($\sim$ns), which is much shorter than the timescale required for the tissue to move (for the local mass density to change, $\sim\mu$s), so the heating is isochoric and therefore accompanied by an increase in pressure. Tissue is elastic, so the regions of higher pressure will act as sources of acoustic waves. Because of the difference in timescales, the pressure increase is usually treated as occurring instantaneously, and PA wave generation and propagation is modeled as the initial value problem:

$$(\partial_{tt} - c^2\Delta)p(x,t) = 0, \quad p(x,0) = f(x), \quad \partial_t p(x,0) = 0, \tag{1}$$

where $x \in \mathbb{R}^3$ is the spatial variable, $t \in \mathbb{R}^{\geq 0}$ is time, and $p(x,t)$ is the acoustic pressure. The medium properties to which the acoustic wave is sensitive, sound speed and mass density, will in general vary with position. However, for propagation through soft tissue, the variations are often small and are rarely known in advance, so the medium is usually treated as acoustically homogeneous. (Acoustic absorption, not described by Eq. (1), may also become important in some applications.) The initial condition $f(x) \geq 0$ is termed the *initial acoustic pressure distribution* and is related to the optical properties of the tissue by the following equation:

$$f(x,\lambda) = \Gamma\mu_a(x,\lambda)\phi(x,\lambda), \tag{2}$$

where λ is the optical wavelength, μ_a is the optical absorption coefficient (dimensions of reciprocal length), $\phi(x)$ is the optical fluence (dimensions of energy per unit area), and Γ is a dimensionless constant that accounts for the efficiency of the acoustic generation (sometimes called the Grüneisen parameter, which it equals in some circumstances). The dependence of ϕ on wavelength has been made explicit in Eq. (2), but ϕ also depends on the absorption and scattering throughout the tissue, making Eq. (2) nonlinear in the absorption coefficient μ_a. The positivity of the initial pressure distribution $f(x)$ arises from the fact that $\mu_a\phi$ is the energy density due to absorption of the light and Γ is positive for most materials, especially soft tissue.

2.1.2 *Tissue optics*

The nature of the dependence of the fluence ϕ on the absorption and scattering is usually modeled in biological tissue using transport theory,[19,20] i.e., making the assumption that coherent optical effects can be safely ignored. Under this assumption, the light field is described in terms of energy by the radiance $\psi(x,t,\hat{s},\lambda)$, which is the rate of energy flow per unit area per unit solid angle in direction $\hat{s} \in S^2$ at position x at time t (units of power per unit area per unit solid angle). When there are no significant inelastic processes such as fluorescence present, the radiance at each wavelength obeys the following integro-differential equation, known as the *radiative transfer equation*, which can be thought of as a statement of the principle of the conservation of energy:

$$\frac{1}{v}\frac{\partial \psi}{\partial t} = q - (\hat{s}\cdot\nabla + \mu_a + \mu_s)\psi + \mu_s \int_{S^2} \vartheta(\hat{s},\hat{s}')\psi(\hat{s}')\mathrm{d}\hat{s}', \tag{3}$$

where v is the speed of light, q is a source term, μ_s is the scattering coefficient, and $\vartheta(\hat{s},\hat{s}')$ is the scattering "phase" function, a probability density function describing the likelihood of a photon travelling in direction $\hat{s}'$ being scattered into the direction $\hat{s}$. The fluence, for a given wavelength, can be found by integrating the radiance at that wavelength over all angles and time:

$$\phi(x,\lambda) = \int_{S^2}\int_{\mathbb{R}^{\geq 0}} \psi(x,t,\hat{s},\lambda)\mathrm{d}\hat{s}\,\mathrm{d}t. \tag{4}$$

The quantity of interest in quantitative PAT is sometimes the absorption coefficient, but more often it is a related quantity. For example, the absorption coefficient is related to the concentrations of the chromophores present by[21]

$$\mu_a(x,\lambda) = \sum_q \alpha_q(\lambda)C_q(x), \tag{5}$$

where C_q is the concentration of the qth chromophore and $\alpha_q(\lambda)$ is its molar absorption coefficient spectrum. A quantity of considerable clinical interest is blood oxygen saturation,[22] which is related to the concentrations of two particular endogenous chromophores, oxy- and deoxy-hemoglobin, C_{HbO} and C_{Hb}, respectively, by

$$\mathrm{sO}_2(x) = \frac{C_{\mathrm{HbO}}(x)}{C_{\mathrm{HbO}}(x) + C_{\mathrm{Hb}}(x)}. \tag{6}$$

2.1.3 *Photoacoustic measurements*

In PAT, measurements of the PA-generated acoustic waves are made on a surface $\mathfrak{S}$ surrounding a region Ω containing the object to be imaged f with $\mathrm{supp}(f) \subset \Omega$ (see Fig. 1). Note that $\mathfrak{S}$ is not a boundary, i.e., it is assumed not to affect the acoustic field. The measurement operator $\mathcal{M}$ will typically consist of a filtering operator $\mathcal{W}$, which accounts for the angle-dependent frequency response of the detectors, and a spatial sampling operator $\mathcal{S}$, which selects the part of the acoustic field to be detected such that

$$g = \mathcal{M}p + \varepsilon = \mathcal{S}\mathcal{W}p + \varepsilon, \tag{7}$$

where ε is the additive measurement noise. (In some imaging systems, e.g., in those using LED excitation, the signal-to-noise ratio can be very low, and it is necessary to average many times.) A variety of different sampling operators have been considered for PAT, including detection at a set of points, $\{x_s \in \mathfrak{S}\}$, for which $\mathfrak{S}$ is a simple geometric surface such as a plane,[23] cylinder,[24] sphere,[25] ellipsoid,[26] or polyhedron,[27] measurements of spatial integrals of the acoustic field along planes or lines[28] or patterns,[29] 2D measurements using a ring of detectors focused in a plane,[30] and measurements made with a linear array of elements also focused in a plane (as used for conventional ultrasound imaging).[31]

PA signals are by their nature broadband, often more broadband than the ultrasound detectors used to measure them, so the detected frequency range is usually restricted. Furthermore, due to the finite size of real ultrasound detectors, they also filter the spatial wavenumbers. (As the detection area increases, the more directional the detectors become, i.e., the narrower the acceptance angle.) The filtering operator $\mathcal{W}$ accounts for both the frequency and wavenumber filtering effects.

When the detectors are ideal $\mathcal{W} = \mathrm{Id}$, the identity, and neglecting noise, Poisson's solution[32] to the initial value problem in Eq. (1) shows that the relationship between the measured time series $g(x_s,t)$ and the initial acoustic pressure $f(x)$ can be written in the form:

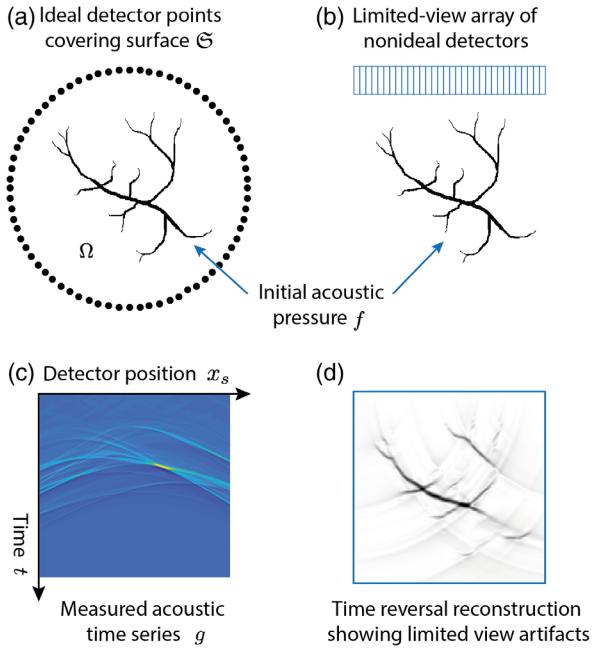

Fig. 1 (a) Ideal PAT measurement setup with point-like omnidirectional detectors covering the surface $\mathfrak{S}$ surrounding the initial pressure distribution f. (Here, shown in 2D but the array would ideally surround f in 3D.) (b) More typical PAT measurement setup using a finite-sized linear array of detectors. This is the setup used for the tutorial in Sec. 4.5. (c) The acoustic time series measured by the linear array. (d) A PAT image reconstructed from these time series using the classical time reversal approach, Sec. 3.1.3, showing arc-like artifacts due to the limited view detection.

$$\frac{1}{t}\int_0^t g(x_s,t')\mathrm{d}t' = \frac{1}{4\pi(ct)^2}\int_{|x_s-x|=ct} f(x)\mathrm{d}A, \tag{8}$$

where dA is an area element of the surface given by the spherical shell $|x_s - x| = ct$. This shows that the time average of g between times 0 and t equals the spatial average of the initial pressure $f(x)$ over a spherical shell of radius ct centered at x_s. More concisely, we can write $g_{\mathrm{sm}} = \Re_{\mathrm{sm}} f(x)$, where $\Re_{\mathrm{sm}}$ is the *spherical mean Radon transform*, and $g_{\mathrm{sm}} = t^{-1}\int_0^t g(x_s,t')\mathrm{d}t'$ is the *spherical mean data*. Some of the literature relevant to PAT image reconstruction considers the data in this form.[1,25]

2.1.4 *Acoustic, optical, and spectroscopic operators*

Before discussing PAT inverse problems, it will be helpful to define three operators describing the forward or direct problems (see Fig. 2). First, the operator $\mathcal{A}$, a linear mapping from the initial acoustic pressure distribution f to the measurements g under additive measurement noise ε, which is based on Eqs. (1) and (7):

$$g = \mathcal{A}f + \varepsilon. \tag{9}$$

$\mathcal{A}$ maps from image space X_f to data space Y. Second, the operator $\mathcal{F}$, a nonlinear mapping from the absorption coefficient μ_a, to the initial pressure distribution f, which is based on Eqs. (2)–(4):

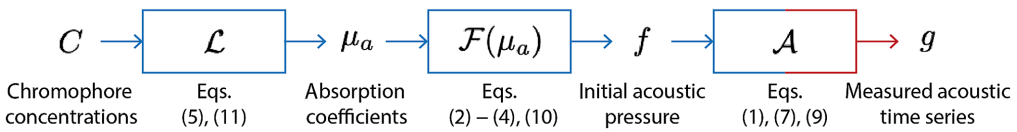

Fig. 2 The three operators describing the PAT forward problem: spectroscopic $\mathcal{L}$, optical $\mathcal{F}$, and acoustic $\mathcal{A}$. (Blue indicates the image space X and red the data space Y.)

$$f = \mathcal{F}[\mu_a](\mu_a). \tag{10}$$

$\mathcal{F}$ maps from image space X_{μ_a} to image space X_f. Finally, a third operator maps chromophore concentrations to absorption coefficients, from X_C to X_{μ_a}, based on Eq. (5):

$$\mu_a = \mathcal{L}C. \tag{11}$$

2.2 *PAT Inverse Problems*

There are two main inverse problems in PAT, corresponding to the acoustic and optical forward operators described already. First, an acoustic inversion from the measured time series to an estimate of the initial acoustic pressure distribution, i.e., an estimate of $\mathcal{A}^{-1}g$, and second, an optical inversion which attempts to recover quantitatively accurate estimates of optical coefficients (or related properties), e.g., an estimate of $\mathcal{F}^{-1}(f)$. It can be shown[1] that the acoustic inverse problem is well-posed when sufficient data have been measured, but what constitutes sufficient data? The data measured by a closely spaced array of omnidirectional, broadband, noise-free point detectors arranged such that all the rays passing through every point in the imaged object reach at least one of the detectors would be sufficient data. For example, if ideal detectors are positioned on the surface of a hemisphere at a spacing of $\lambda_{\min}/2$, where $\lambda_{\min}$ is the shortest wavelength generated (to satisfy the spatial Nyquist criterion), the sound speed is constant everywhere, and the object lies inside the hemisphere's convex hull—the "visible" region[33]—then the acoustic inversion will be well-posed. Given the stringency of these requirements, it is unsurprising that real experimental settings will often diverge from this ideal, leading to inversions that are no longer well-posed. One challenge for the reconstruction, then, relates to dealing with incomplete or imperfect measurement data. There are also challenges relating to the forward operator. These issues are outlined as follows as it is in tackling these issues that DL may be able to make the most useful contributions.

2.2.1 *Incomplete or imperfect data*

For the acoustic inversion, the data could be incomplete or imperfect for several reasons.

- *Noise* is present in any real measurement.
- *Detector responses* are never perfectly broadband or omnidirectional. Compromising on these characteristics is sometimes necessary in order to achieve sufficient detection sensitivity.
- *Limited-view detection*, in other words insufficient coverage of the object, perhaps because of the limitations of the available hardware, e.g., a 2D linear array to image a 3D object, or due to restricted access to the object.
- *Undersampling* in space or time, perhaps in order to achieve faster data acquisition, or due to hardware constraints.

In the optical inversion, when considered separately from the acoustic inversion, the input data are images of the initial pressure distribution f obtained from the acoustic inversion. There are two ways in which this data can be deficient as follows.

- *Artifacts* may be present in the image data due to an imperfect acoustic reconstruction.
- *Wavelengths*. The data must contain images at a set of wavelengths chosen such that the spectroscopic aspect of the optical problem $\mathcal{L}^{-1}$ is well-posed.

2.2.2 *Inaccurate forward operators*

Equations (1)–(4) are broadly considered to capture the physical phenomena relevant to PAT, but to solve practical problems they must be implemented as numerical models. There are two ways in which these models can differ from the ideal.

- *Simplifying approximations* are often made to the forward operators in order to reduce the complexity of the computations necessary to implement them as numerical models. For example, the radiative transfer equation [Eq. (3)] is often approximated using a diffusion approximation, and the wave equation [Eq. (1)] is sometimes substituted with a simpler model, e.g., based on rays.
- *Inaccurate model inputs*. Although the focus in experimental settings is usually on the accuracy of the measurement data, the accurate determination of the auxiliary parameters on which the forward operators depend is often just as crucial. For example, the acoustic operator $\mathcal{A}$ depends on the sound speed and how it varies within the tissue, which is rarely known to a high degree of precision. And the optical operator $\mathcal{F}$ not only requires knowledge of how the tissue was illuminated but also of the tissue's scattering properties, both of which can be hard to determine accurately.

This latter problem, the difficulty of accurately measuring the necessary model inputs, in particular the sound speed and the optical scattering distributions, has led some researchers to consider them as additional unknowns in the inverse problem.[34,35] These inversions have been shown to be less well-posed[36,37] than the inversions of $\mathcal{A}$ and $\mathcal{F}$, and additional data or constraints are usually required to find meaningful solutions.

2.2.3 *Statistical framework: noise models and priors*

The question naturally arises as to what can be done to improve the image reconstruction when the data are imperfect or the forward model is only known approximately. An approach that sounds like common sense, but in practice can be challenging, is to try to find the reconstruction f that is most probable given the data g. This requires a statistical framework,[38] which also provides a way to incorporate in the reconstruction any other information that is already known about the final image, the data, or the operator, in order to constrain the solution to one that has a higher probability of being correct. Specifically, we want to find the posterior probability distribution $\pi(f|g)$, or some related quantity that characterizes the most probable reconstructions. Here the notation $\pi(f)$ is used to denote the probability density function of f, and $\pi(f|g)$ is the conditional probability density of f given g. In the Bayesian framework, we can incorporate our prior knowledge about the problem via Bayes' formula:

$$\pi(f|g) \propto \pi(g|f)\pi(f), \tag{12}$$

where $\pi(f)$ incorporates prior knowledge of the solution f, and $\pi(g|f)$, called the *likelihood*, incorporates the known noise statistics using the forward operator $\mathcal{A}$. For example, if the noise in Eq. (9) is normally distributed with zero mean and variance σ^2, we can express the likelihood as[38]

$$\pi(g|f) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathcal{A}f - g\|_2^2\right). \tag{13}$$

Even though in many applications, we might not be able to explore the full posterior distribution $\pi(f|g)$, the Bayesian framework can provide guidance for the interpretation of specific image reconstruction approaches. For instance, computing the *maximum a posteriori* (MAP) estimate relates to finding the minimizer in variational approaches, as we will discuss later in Sec. 3.1.4.

Classically, both the likelihood and the prior are explicitly modeled and might therefore be limited in their expressibility. Hence a natural question arises in the context of this study: *Can we use learning-based models instead of analytical models to generalize this approach?* In particular, two ways in which learning-based methods could be incorporated are as follows.

(i) Learning a prior $\pi(f)$ that describes the unknown initial acoustic pressure distribution better.
(ii) Compensating, in the likelihood $\pi(g|f)$, for model uncertainties or complex noise statistics.

In DL, it is conceptually easier to address the estimation of a prior, as it relates to the training set, as we will discuss in the later sections; it is not so straight-forward to incorporate model uncertainties into the likelihood estimation. A useful direction on how to tackle this is given by the well-established approach of Bayesian approximation error modeling.[38–40] In this approach, modeling errors in the forward operator $\mathcal{A}$ are estimated as normally distributed and explicitly corrected in the likelihood term [Eq. (13)]. This approach has been applied in PAT to compensate for uncertainty in the measurement parameters (model uncertainty).[41,42]

2.2.4 *Image reconstructions*

Although the two inverse problems described already, the acoustic and optical inversions, are the fundamental image reconstruction problems in PAT, variations on them are often used in practice. Common PAT reconstruction problems that appear in the literature are as follows.

- Reconstructing an image of the initial acoustic pressure distribution from the measured time series data $\mathcal{A}^{-1}g$.
- Reconstructing an image of the optical absorption coefficient from initial acoustic pressure distribution images $\mathcal{F}^{-1}(f)$.
- Reconstructing an image of optical absorption coefficient directly from the measured time series data $(\mathcal{A}\mathcal{F})^{-1}(g)$.
- Reconstructing images of quantities related to optical absorption, e.g., chromophore concentrations or blood oxygenation, from a multiwavelength set of initial acoustic pressure images $(\mathcal{F}\mathcal{L})^{-1}(f)$.
- Reconstructing images of quantities related to optical absorption directly from a multiwavelength set of time series data $(\mathcal{A}\mathcal{F}\mathcal{L})^{-1}(g)$.

Research has already begun on applying DL to several of these tasks; this literature will be reviewed in Sec. 5. The next two sections will give an overview of the classical approaches to PAT image reconstruction and a short tutorial on the kinds of DL that are being used for image reconstruction.

3 Classical Approaches to PAT Image Reconstruction

DL can be used to complement or augment current approaches to PAT reconstruction or replace parts of them. For this reason, as well as to provide context, this section describes several widely used "classical"—i.e., not learning-based—approaches that have been used for solving the PAT inverse problems. This section is not intended to be a comprehensive review of classical methods for PAT image reconstruction, for which the literature is large, but for later reference.

3.1 *Acoustic Reconstruction*

Here we consider the *acoustic* inversion of PAT, i.e., the linear problem of solving Eq. (9) for f, the initial pressure distribution, given g, the measurement data. We will denote a generic reconstruction operator, or data-to-image mapping, by $\mathcal{A}^{\dagger}: Y \to X_f$ throughout this review. Let us now discuss specific choices for such a mapping.

3.1.1 *Backprojection and beamforming*

Algorithms based on the idea of *backprojection* are widely used in PAT. This terminology comes from X-ray tomography, in which the forward operator (the linear ray transform) maps from image to data space by integrating the target along a set of straight lines for each detector, and the backprojection operator maps from data to image space by putting the data back along those straight lines and summing over all detectors. In the X-ray case, these dual operations are also adjoint; the backprojection operator is the adjoint of the forward operator.[43] In PAT, the situation is slightly different. The forward operator $\mathcal{A}$ maps from image space X_f to data space Y by

integrating through f along a set of spherical shells of radius $t = |x - x_s|/c$ centered on the detector points x_s (see Sec. 2.1.3). Correspondingly, the backprojection operator $\mathcal{A}^{\#}$ maps a function of x_s and t, from data space Y to image space X_f by putting the data back onto the same spherical shells with the mapping $t \to |x - x_s|/c$, and summing over all detector points x_s. For some function $h(x_s, t)$, which might be the measurement data or some function of it, the backprojection operator is

$$(\mathcal{A}^{\#}h)(x) = \int_{\mathfrak{S}} [h(x_s, t)]_{t=|x-x_s|/c} \mathrm{d}\mathfrak{S}(x_s), \tag{14}$$

where $d\mathfrak{S}$ is an area element on the measurement surface $\mathfrak{S}$. On the other hand, the adjoint operator $\mathcal{A}^*$ is given by[44]

$$(\mathcal{A}^*g)(x) = \int_{\mathfrak{S}} \left[\frac{1}{4\pi|x - x_s|}\frac{\partial g}{\partial t}(x_s, t)\right]_{t=|x-x_s|/c} \mathrm{d}\mathfrak{S}(x_s), \tag{15}$$

which is clearly a backprojection, but not of the data g (see also Sec. 3.1.4). When the data are processed before backprojection (or sometimes the image is processed after backprojection) the resulting algorithm is often referred to as a *filtered backprojection*. Filtered backprojection formulas for PAT have been found for a variety of measurement surface geometries.[1,25,27,45,46] Perhaps the most well-known, called the "universal backprojection" algorithm,[45] gives exact reconstructions for detector points covering a spherical, cylindrical, or planar measurement surface, and can be written as

$$f(x) = \frac{-2}{\Omega_s c^2}\int_{\mathfrak{S}} \left[\frac{\partial}{\partial t}\left(\frac{g(x_s, t)}{t}\right)\right]_{t=|x_s-x|/c} \cos(\alpha)\mathrm{d}\mathfrak{S}(x_s), \tag{16}$$

where α is the angle between the inward normal to $\mathfrak{S}$ and the vector $(x - x_s)$, and Ω_s is the solid angle of $\mathfrak{S}$ as seen from a point $x \in \Omega$, e.g., $\Omega_s = 4\pi$ when $\mathfrak{S}$ is a sphere. A 2D version of this has also been derived:[47]

$$f_{2\mathrm{D}}(x) = \frac{-4}{\Omega_s c^2}\int_{\mathfrak{S}} \left(\int_{|x-x_s|/c}^{\infty} \frac{1}{\sqrt{t^2 - |x - x_s|^2/c^2}}\frac{\partial}{\partial t}\left(\frac{g(x_s, t)}{t}\right)\mathrm{d}t\right)\kappa(x, x_s)\cos(\alpha)\mathrm{d}\mathfrak{S}(x_s), \tag{17}$$

where the weighting factor $\kappa(x, x_s) = |x - x_s|$ for the universal backprojection algorithm, but has also been treated as a learnable parameter (Sec. 5.1.2).

Linear array transducers of the kind used in conventional ultrasound imaging are increasingly being used for PAT, with backprojection-type formulas commonly used for image reconstruction. In this context, image reconstruction is sometimes referred to as "beamforming" and the backprojection operation $\mathcal{A}^{\#}$ is descriptively dubbed "delay-and-sum." Linear arrays are typically short, consisting of just 128 bandlimited detection elements focused in a plane, so the image reconstruction is very ill-posed. Many variations of backprojection-type algorithms with different pre- and postprocessing steps have been explored to try to maximize the image quality given these severe constraints.[48,49] In DL approaches to PAT image reconstruction, backprojection/beamforming-type algorithms have been used widely to map from data space Y to image space X_f before and after post- and preprocesssing networks, respectively (see Secs. 4.3.2, 5.1, and 5.2).

3.1.2 *Series solutions*

The first analytical solution for $f(x)$ was found in the form of an infinite series, and more have since been derived.[23,24,50–52] A formula for the case of detection points lying on a plane is of particular interest because it is in the form of a Fourier transform, which can be computed efficiently using the Fast Fourier Transform.[23] The solution relies on the fact that any acoustic wavefield $p(x, t)$ can be written as a sum of travelling plane waves whose temporal frequency ω and wavevector $k = (k_1, k_2, k_3)$ are linked by the dispersion relation $\omega = c|k| = c\sqrt{k_1^2 + k_2^2 + k_3^2}$.

The solution takes the form:

$$f(x) = \mathfrak{F}^{-1}_{1,2,3}\{\tilde{f}(k)\}, \quad \tilde{f}(k_1, k_2, \omega) = B(k_1, k_2, \omega)\mathfrak{F}_{1,2}\{\{\mathfrak{C}_t\{g(x_1, x_2, t)\}\}\}, \tag{18}$$

where $B(k_1, k_2, \omega) = \sqrt{(\omega/c)^2 - k_1^2 - k_2^2}/\omega$, $\mathfrak{F}$ and $\mathfrak{C}$ are Fourier and Cosine transforms, respectively, and $\tilde{f}(k)$ is obtained by algebraic transform from $\tilde{f}(k_1, k_2, \omega)$ using the dispersion relation. In DL, this method has been used as a component in learned iterative reconstructions (see Sec. 5.4.2). When used with linear array transducers, this method and its variants are sometimes referred to as "Fourier beamforming."

3.1.3 *Time reversal*

Perhaps the most physically intuitive algorithm is based on the concept of time reversal.[53–55] Consider a measurement surface $\mathfrak{S}$ surrounding a region $\text{supp}(f) \subset \Omega$. Imagine the photoacoustically generated waves propagating outward and being measured as they pass through the surface $\mathfrak{S}$. After a suitably long time T, the acoustic field in Ω will be zero (guaranteed in a 3D homogeneous medium by Huygens' principle[56]). If the measured pressure $g(x_s, t)$ were now reproduced on $\mathfrak{S}$ in time-reversed order, starting with $g(x_s, T)$, then the acoustic field in Ω created by the *in-going* waves would reproduce the out-going wavefield exactly but backward in time. In particular, the field at $t = 0$ would be the initial acoustic pressure distribution $f(x)$. Based on this idea, time reversal image reconstruction uses a numerical acoustic model to solve the following time-varying boundary value problem for the time-reversed field $p_r(x, t_r)$, from time $t_r = 0$ to T:

$$(\partial_{tt} - c^2\Delta)p_r(x, t_r) = 0, \quad p_r(x_s, t_r) = g(x_s, T - t_r), \quad p_r = \partial_t p_r(x, 0) = 0. \tag{19}$$

The solution $p_r(x, T) = f(x)$ for $x \in \Omega$.

In DL studies, the time reversal approach is sometimes used for comparison with network approaches, but care must be exercised here to ensure a fair comparison. To help elucidate two problems with time reversal, note that the time-varying Dirichlet condition $p_r(x_s, t_r) = g(x_s, T - t_r)$ is equivalent to reintroducing the measurement data as a source term within a reflective cavity defined by the measurement surface $\mathfrak{S}$.[53] First, then, time reversal is not a good choice when using data detected on a sparse array of points because during the time reversal procedure they act like point scatterers. Second, when the true sound speed is spatially varying but the reconstruction uses a homogeneous sound speed, the reflective effect of the boundary condition can trap artifacts in the image region.[57] In these scenarios, time reversal may not be the best method for comparison. Furthermore, when the sound speed is spatially varying, resulting in multiple scattering, the requirement that the acoustic field in Ω will fall to zero in a finite time T is no longer satisfied. One solution[58] is to use the following iterative scheme:

$$f^{(n+1)} = f^{(n)} - \mathcal{A}^{\text{TR}}(\mathcal{A}f^{(n)} - g), \tag{20}$$

where $\mathcal{A}^{\text{TR}}$ signifies the time-reversal operator.

3.1.4 *Variational approaches*

The iterative time reversal algorithm points to a more general approach to reconstruction as it looks very similar to this gradient descent scheme:

$$f^{(n+1)} = f^{(n)} - \eta\nabla_f\mathcal{E} = f^{(n)} - \eta\mathcal{A}^*(\mathcal{A}f^{(n)} - g), \tag{21}$$

which solves the least-squares minimization problem:

$$f^* = \arg\min_f \mathcal{E}(f), \quad \mathcal{E}(f) = \frac{1}{2}\|\mathcal{A}f - g\|_2^2, \tag{22}$$

where f^* denotes the optimal solution. [The similarities between Eqs. (21) and (20) become even clearer if we observe that the adjoint operator $\mathcal{A}^*$ can be implemented numerically in a similar

way to the time reversal operator $\mathcal{A}^{\mathrm{TR}}$ except that the pressure time series are reintroduced to the domain in time-reversed order by *adding* them to the existing field rather than enforcing the pressure at the detector points.[44]] The idea of posing the image reconstruction as a numerical optimization is appealing,[4–6,59–61] because it provides a very flexible framework both for how the forward operator is defined (e.g., the sound speed could be spatially varying) and for tackling ill-posedness in the inverse problem. Equation (22) will have a unique solution when g is a complete set of ideal data. However, if g is incomplete or imperfect then Eq. (22) may not have a unique solution or overfitting (in which the model starts to fit to the noise in the data) may become a concern. Early stopping of the iteration in Eq. (21) is one way to avoid overfitting, but a more general approach to restricting the solution space is to add another term to the functional in Eq. (22) that expresses prior information about the kind of solution that is expected, e.g., non-negativity of solutions and smoothness or sparsity conditions. The problem then becomes

$$f^* = \arg\min_f \mathcal{E}(f), \quad \mathcal{E}(f) = \frac{1}{2}\|\mathcal{A}f - g\|_2^2 + \alpha\mathcal{R}(f), \tag{23}$$

where the regularization parameter α balances the importance placed on the first term—the data consistency term—and the second term $\mathcal{R}$, which encodes the prior information about f. There is an extensive literature on methods to solve minimizations such as this.[62,63] If the regularization term $\mathcal{R}(f)$ is differentiable, one could simply employ a gradient descent [Eq. (21)] with $\alpha\partial\mathcal{R}/\partial f$. If not, another approach to computing solutions iteratively is the proximal gradient method, which means computing the iteration:

$$f^{(n+1)} = \mathrm{prox}_{\mathcal{R},\eta\alpha}\left(f^{(n)} - \eta\mathcal{A}^*(\mathcal{A}f^{(n)} - g)\right), \tag{24}$$

where $\mathcal{A}^*(\mathcal{A}f^{(n)} - g)$ is the gradient of the data consistency term and $\mathrm{prox}_{\mathcal{R},\eta\alpha}$, the proximal operator, takes the updated image estimate and projects it into the constrained set defined by the regularization, or in other words the space in which the solution is thought to exist. It is formally defined as the minimization problem:

$$\mathrm{prox}_{\mathcal{R},\eta\alpha}(h) = \arg\min_y \left\{\eta\alpha\mathcal{R}(y) + \frac{1}{2}\|h - y\|_2^2\right\}. \tag{25}$$

The formulation as a minimization problem in Eq. (23) is directly connected to the Bayesian formulation in Eq. (12) and corresponds to maximizing the posterior distribution $\pi(f|g)$ to find the most likely reconstruction f. This represents a point estimator known as the MAP estimate.[38] In this context, the negative logarithm of the prior distribution directly relates to the regularizing term, $-\log\pi(f) \propto \mathcal{R}(f)$. This general framework provided by the variational approach has inspired several learned iterative approaches to PAT reconstruction (see Secs. 4.3.3 and 5.4).

3.1.5 *Matrix formulation*

The acoustic forward operator $\mathcal{A}$ is linear and so can, in principle, be discretized and written as a (large) matrix. When this matrix can actually be explicitly computed, the image reconstruction problem has been reduced to a matrix inversion and all the machinery of linear algebra, and the associated methods of regularization, can be brought to bear to solve it. This includes the variational approaches above in Sec. 3.1.4. For instance, if one considers a quadratic regularization in Eq. (23), such as $\mathcal{R}(f) = \|f\|_2^2$, then the solution can be computed in closed form and is given by

$$f^* = (\mathcal{A}^*\mathcal{A} + \alpha\mathrm{Id})^{-1}\mathcal{A}^*g, \tag{26}$$

where Id denotes the identity, which is sometimes called a Tikhonov-regularized solution.

There are many methods that can be used to discretize the forward operator, from pseudo-spectral methods[4] to semianalytical approaches.[64] However, whether it is convenient—or even possible—to compute and store $\mathcal{A}$ explicitly as a matrix will depend on the number of detectors and the size of the image, and whether sparsity or other structures in the matrix can be exploited.[65] In fact, we will make use of a matrix representation for $\mathcal{A}$ in the tutorial part of this review (Sec. 4.5), as the problem under consideration is sufficiently small.

3.2 *Optical Reconstructions*

This section briefly summarizes classical approaches to solving the nonlinear Eq. (10) for the absorption coefficient or related quantities. From Eq. (2), we can see formally that $\mu_a = \mathcal{F}^{-1}(f) = f/(\Gamma\phi(\mu_a))$. An empirically determined value for Γ is sometimes used, or, when the final quantity of interest is a ratio of concentrations, see Eq. (6), Γ is assumed to be constant with wavelength and cancels out. The dependence of the fluence ϕ on μ_a, however, needs to be considered carefully.

3.2.1 *Noniterative approaches*

A simple approach to deal with the dependence of ϕ on μ_a, but one with questionable accuracy, is to ignore the dependence and apply the spectroscopic inversion directly to the PA data, $\mathcal{L}^{-1}f$. This is sometimes known as *linear unmixing*. Despite its obvious flaws, this stance has been taken (usually implicitly) in many experimental papers, in which the PA spectrum at a point $f(\lambda)$ has been assumed to be proportional to the absorption spectrum at that point $\mu_a(\lambda)$. The difference between $f(\lambda)$ and $\mu_a(\lambda)$, which linear unmixing ignores, is known as *spectral coloring*. A better approach, but still one whose accuracy needs to be demonstrated on a case-by-case basis, is to approximate the fluence using estimated average background absorption and scattering values, and suppose that this fluence remains unchanged by small changes in the optical absorption coefficient. In some cases, the fluence distribution can be measured directly using a second imaging modality in addition to PAT.[66,67] However, this requires complementary hardware to make the additional measurements, and it is difficult to achieve the same spatial resolution for ϕ as for f (or one may as well measure just the fluence distribution and not do PAT at all).

3.2.2 *Fixed-point iterations*

If the scattering is known, then the absorption coefficient can be found using a model of light transport, such as Eq. (3) or a suitable approximation, to calculate both the fluence and the absorption coefficient iteratively using the fixed point iteration:[68]

$$\mu_a^{(n+1)}(x,\lambda) = f(x,\lambda)/(\Gamma\phi^{(n)}(x,\lambda;\mu_a^{(n)})). \tag{27}$$

3.2.3 *Variational approaches*

As with the acoustic inversion described in Sec. 3.1.4, casting the optical inversion as a minimization problem allows the various constraints and prior information to be included systematically. Here, the inverse problem for the absorption coefficient is stated as

$$\mu_a^*(x) = \underset{\mu_a(x)}{\arg\min}\,\mathcal{E}(\mu_a), \quad \mathcal{E}(\mu_a) = \frac{1}{2}\|\mathcal{F}[\mu_a](\mu_a) - f\|_2^2 + \alpha\mathcal{R}(\mu_a), \tag{28}$$

and a similar expression can be written for the oxygenation saturation or other quantities of interest. As, from Eq. (2), $\mathcal{F}(\mu_a) \coloneqq \Gamma\mu_a\phi(\mu_a)$, the functional gradient is given by $\nabla_{\mu_a}\mathcal{E} = \Gamma(\mu_a D\phi + \phi)$, where $D\phi$ is the Fréchet derivative of ϕ, the form of which will depend on the particular model of light transport used.[34,42,69]

4 Tutorial Introduction to Deep Learning for PAT Image Reconstruction

4.1 *What Role Could Deep Learning Play?*

How can DL help to solve the challenges posed by the twin problems of incomplete data and inaccurate forward models outlined in Sec. 2.2? Or are there other ways in which DL can be used to enhance PAT image reconstruction? There are many areas in which DL could make an impact. For example, a DL network could be used to

- correct for missing or corrupted data in the measured time series data (preprocessing);
- reconstruct images from incomplete or imperfect data given the forward operator (effectively learning prior information to regularise the solution);
- approximate a forward operator (e.g., when it is difficult to write an accurate and computationally efficient forward model explicitly);
- approximate an inverse operator (even when the data is perfect and the forward operator known this may speed up the image reconstruction);
- remove artifacts and noise from reconstructed images (postprocessing);
- segment images;
- classify or label images or regions of images.

(As mentioned in Sec. 1, the last two points are out of the scope of this review.) An important attribute of a DL network is the speed with which it can process an input. For small networks, this can be very fast, which may be useful in settings where reconstructions are required on short time scales such as real-time or dynamic imaging. However, the speed of evaluation will depend on the size of the network and the size of the input data. It is also important to keep in mind that the final reconstruction speed will still depend on how the forward operator is utilized in the processing pipeline.

The motivation to use DL in image reconstruction, which these conceptual advantages provide, can readily be followed by action thanks to the availability of easy-to-use DL tools, such as TensorFlow[70] and PyTorch,[71] which make employing these methods straightforward. Furthermore, the tendency of the machine/DL community to provide open-source algorithms and data accelerates the development of methods and makes it simpler for researchers to try approaches. Consequently, we provide the codes accompanying this review along with a basic example of training and test data.

4.2 *Brief Introduction to Deep Learning*

This review concentrates on the application of DL, by which we mean in particular deep neural networks, to image reconstruction tasks in PAT. Specifically, we will concentrate throughout this section on the acoustic inverse problem, so we are interested in finding a mapping from the measurement data $g \in Y$ to the initial acoustic pressure $f \in X_f$. The driving incentive is the hope that a reconstruction operator $\mathcal{A}_\theta^\dagger : Y \to X_f$ that is parameterized by a set of *learnable* parameters θ, such that

$$f \approx \mathcal{A}_\theta^\dagger(g) \tag{29}$$

can give better (faster, more accurate) reconstructions than classical approaches. The mapping in Eq. (29) may be a composition of model-based parts, involving a known operator describing the acquisition geometry and physics, and pure learning-based components. Before we can review common approaches and network architectures, we will give a short introduction to DL and the main network components. We will concentrate here on a high-level overview to help develop an intuition for the operations involved. For a more extensive review, see for instance.[72–74]

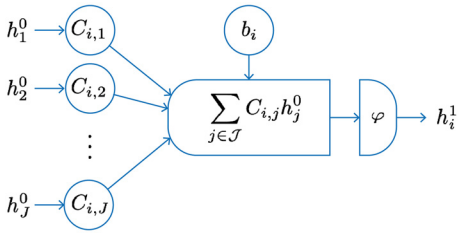

Fig. 3 The ith neuron in one layer of an artificial neural network takes an input vector h^0 and computes an output vector h^1 according to Eq. (31).

4.2.1 *Deep neural networks*

A deep neural network, denoted here by the nonlinear operator Λ_θ, maps an input vector to an output vector. The network consists of several "layers," each of which is a composition of an affine linear function with learnable parameters and a nonlinear function (often referred to as the "activation function" but referred to as a nonlinearity here). The term DL refers, roughly, to networks that consist of multiple layers, in contrast to shallow networks consisting of only a few layers.

Let us now formalize the notion of a layer. Given an input vector $h^0 = \{h_j^0\}_{j=1}^J \in \mathbb{R}^J$, where $j \in \mathcal{J} = \{1, \ldots, J\}$ and an output vector $h^1 = \{h_i^1\}_{i=1}^I \in \mathbb{R}^I$, where $i \in \mathcal{I} = \{1, \ldots, I\}$, a linear map given as a matrix $C \in \mathbb{R}^{I\times J}$, a vector $b \in \mathbb{R}^I$, and a point-wise nonlinear function $\varphi : \mathbb{R} \to \mathbb{R}$, then one layer $\mathfrak{L}$ in a network is given by

$$\mathfrak{L}(h^0) = \varphi(Ch^0 + b) = h^1. \tag{30}$$

In the literature, the term layer is used somewhat ambiguously. Here it will be used to refer to both an operation and its output, not just the output. One exception is the input layer, which refers to just the input data with no prior operation. The individual neurons in such a neural network are now the mapping to one element of the output vector, see Fig. 3 for an illustration. If we write this out for the above case [Eq. (30)], then the result of the ith neuron is the ith element of the output vector h_i^1 and each neuron sums over all input elements of h_0 with a common bias b_i:

$$h_i^1 = \varphi\left(\sum_{j\in\mathcal{J}} C_{i,j}h_j^0 + b_i\right) \quad \text{for each } i \in \mathcal{I}. \tag{31}$$

The network type is essentially defined by the linear mappings in each layer, defined by the structure of the matrix C.

4.2.2 *Fully connected layers*

The basic choice for C is a dense matrix, which gives a *fully connected layer* as all the inputs are related to all the outputs. We then obtain a simple L-layered network by the composition of L fully connected layers. This network can be expressed as the composition of several layers $\mathfrak{L}_l$ for $l = 1, \ldots, L$ to obtain

$$h^L = \Lambda_\theta(h^0) = (\mathfrak{L}_L \circ \mathfrak{L}_{L-1} \circ \ldots \circ \mathfrak{L}_1)(h^0). \tag{32}$$

For example, if we write this out for a two-layer network, we get the relation:

$$h^2 = \varphi(C^2h^1 + b^2) = \varphi(C^2\varphi(C^1h^0 + b^1) + b^2). \tag{33}$$

In the general case, the trainable parameter set θ of the network Λ_θ is given by the matrices and the bias vectors, that is $\theta = \{C^l, C^{l-1}, \ldots, C^1, b^l, b^{l-1}, \ldots, b^1\}$. This basic network architecture, consisting of multiple fully connected layers, is the basis for many deep neural networks.

When using fully connected layers in imaging applications, the input, either an image or other measured signals, must be reshaped into a vector for the input layer. If one then aims to extract some relevant low-dimensional information from the input, the dimensions of successive layers will be gradually decreased until the desired output dimensions are reached. An often-used network architecture worth mentioning in this class is termed an *autoencoder*. Here the input is first *encoded* using a contracting path to extract a low-dimensional representation of relevant features and then subsequently *decoded* using an expanding path to represent a clean version of the input signal. Input h^0 and output h^l typically have the same or similar dimensions.

4.2.3 *Convolutional neural networks*

Often the values in image pixels or voxels are related in some way with those in neighboring pixels or voxels, e.g., both may be part of the same image feature. For applications of DL to imaging, therefore, it seems wise to take spatial relations, and especially local relations, into account. A fully connected layer does not explicitly maintain these spatial relations, as all inputs are connected to all outputs without reference to their respective spatial positions. In other words, the linear mapping C in Eq. (30) does not have any predetermined structure. It is possible, however, to think of structures for C that do retain spatial information and can use local features in the input, such as edges, to encode such features more efficiently in the output. Convolutions, especially with small filters—3×3 say—are a popular and very successful choice for such operations, as these are also translation equivariant and hence encode the same local features under translation of the image and are agnostic to the image size. (We say that a function is translation equivariant if translating the input and then applying the function is equivalent to applying the function followed by translation of the output.) Additionally, localized filters have the advantage of leading to linear mappings with sparse structure that can be efficiently implemented without an explicit matrix representation. In this case, instead of learning the whole matrix C, one needs to learn only the filter coefficients. Usually multiple such filters are used, each one referred to as a "channel" here. Networks using this idea are called *convolutional neural networks* or CNNs.

Consider an application to imaging in $\mathbb{R}^2$. The input is either a single or multichannel image $h^0 = \{h_j^0 \in \mathbb{R}^{m\times m}\}_{j=1}^{J} \in \mathbb{R}^{m\times m\times J}$, where $j \in \mathcal{J} = \{1,\ldots,J\}$ denotes the input channels, and similarly an output $h^1 = \{h_i^1 \in \mathbb{R}^{m\times m}\}_{i=1}^{I} \in \mathbb{R}^{m\times m\times I}$, where $i \in \mathcal{I} = \{1,\ldots,I\}$ denotes the output channels. (These images are square, but this can straightforwardly be extended to nonsquare images.) The affine linear mapping is then defined by a set of I filters $\omega_i \in \mathbb{R}^{m_\omega\times m_\omega\times J}$ where $\omega_i = \{\omega_{i,j} \in \mathbb{R}^{m_\omega\times m_\omega}\}_{j=1}^{J}$, and biases $b \in \mathbb{R}^I$, where each output channel has one bias. The convolutional layer that maps between the two multichannel images h^0 and h^1 is then defined for each channel as

$$h_i^1 = \varphi\left(\sum_{j\in\mathcal{J}} \omega_{i,j} * h_j^0 + b_i\right) \quad \text{for each } i \in \mathcal{I}, \tag{34}$$

where $*$ denotes a discrete convolution (see Fig. 4). The set of parameters in this case is given by the coefficients of the filters ω_i and biases b_i. Each output channel has one scalar bias and the input and output of each channel are connected by one specific filter. Thus we could consider an analogy here: in a CNN, each channel in the convolutional layer [Eq. (34)] acts similar to a neuron in a fully connected layer [Eq. (31)] with the filter components $\omega_{i,j}$ analogous to the point-wise weights $C_{i,j}$; compare Figs. 3 and 4. We also note that the convolutional layer [Eq. (34)] could be written in the general form Eq. (30) by vectorizing the input and representing the convolution as a (sparse) matrix.

In this paper, in common with much of the image processing literature, we use "image resolution" to refer to the number of pixels or voxels in an image, so an image with 128×128 pixels has twice the image resolution of one with 64×64 pixels. The same term is used in imaging physics with a different meaning, there referring to the smallest resolvable features in an image. For example, a blurry image consisting of a large number of pixels would have a high resolution in the terminology we use here, but a low resolution in the sense that the fine features in the image are not distinguishable. An important feature of CNNs is the fact that

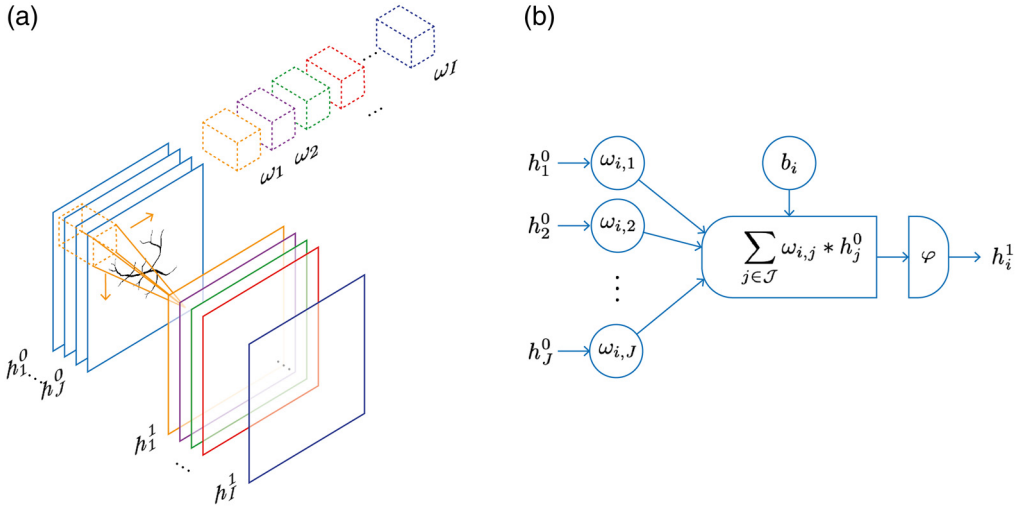

Fig. 4 One layer of a CNN. (a) The input, consisting of J channels $\{h_1^0, \ldots, h_J^0\}$, is convolved with the I filters ω_i, then the nonlinearity φ is applied and biases b_i added (not shown) to give the output in I channels, $\{h_1^1, \ldots, h_I^1\}$. (b) In analogy with Fig. 3 for the fully connected network, the ith output channel of a CNN takes an input $h^0 \in \mathbb{R}^{m\times m\times J}$ and computes an output $h^1 \in \mathbb{R}^{m\times m\times I}$ according to Eq. (34).

each layer maps between multichannel images of the same (or similar) resolution. Therefore, a CNN is a natural choice to represent data-to-data or image-to-image mappings, rather than mappings between spaces with different dimensions such as data-to-image. Nevertheless, in many applications there are reasons why it might be desirable to downsample input images during the processing (e.g., memory constraints, sparser representation, and wider receptive field) and hence many architectures include downsampling operations, called pooling layers, which reduce the image resolution using mean or maximum filters, for instance. This will become clearer in the following section on network architectures, specifically in Sec. 4.3.2. By combining convolutional and fully connected layers, we can define the majority of network architectures that are used in the literature for image reconstruction. The specific networks depend on the task for which they are employed and hence we will discuss the particular architectures later in this section. Let us now focus on how the network parameters are learned.

4.2.4 *Learning task*

After defining the network architecture, the parameters of the network need to be determined. This is done by learning them from a set of training data. Before this can be done, we need to define the actual learning task that will determine the network's mapping properties. That is, we want train the network to perform a specific task, such as either reconstructing or denoising an image, or in other applications to perform segmentation or classification. More precisely, given a network Λ_θ we need to find an optimal set of parameters θ^*, such that our network fulfils the desired mapping property, i.e., it does what we want. The training of the network is nothing more than an optimization problem to find the optimal set of parameters θ^*, which can be formulated in various ways as we will summarize shortly. Specifically, we will consider the reconstruction task of recovering the initial pressure f from the measurement data g given the parameterized reconstruction operator $\mathcal{A}_\theta^\dagger$, such that Eq. (29) is fulfilled.

Supervised training. The first idea that comes to mind is to minimize a distance function between the desired output—the known ground truth—and the actual output of the network. This leads to *supervised training*, in which the optimization problem is formulated with knowledge of a desired ground truth in order to find the parameters of $\mathcal{A}_\theta^\dagger$. For the optimization, we need pairs of measurement data g_i and corresponding ground truth f_i for $\mathrm{i} = 1, \ldots, \mathfrak{I}$. The set of pairs $\{(g_\mathrm{i}, f_\mathrm{i})\}_{\mathrm{i}=1}^{\mathfrak{I}}$ is called the training set. Next we need to define how to measure

the closeness of the resulting reconstruction. For that purpose, one typically formulates a loss function in the L^p-norm, such as

$$L_\theta(f_\mathrm{i}, g_\mathrm{i}) = \|\mathcal{A}_\theta^\dagger(g_\mathrm{i}) - f_\mathrm{i}\|_p^p. \tag{35}$$

The learning task is to find an optimal set of parameters θ^* in the space of possible parameters Θ that minimizes Eq. (35) with respect to the given training set:

$$\theta^* = \arg\min_{\theta\in\Theta} \frac{1}{\mathfrak{I}} \sum_{\mathrm{i}=1}^{\mathfrak{I}} L_\theta(f_\mathrm{i}, g_\mathrm{i}). \tag{36}$$

In fact, one is not limited to loss functions of the form Eq. (35) and depending on the learning task other more suitable choices can be made. Additionally, one can add regularization terms to the loss function, either on the output of the network or even on the parameters, for instance requiring sparsity by minimizing the L^1-norm $\|\theta\|_1$, where the 1-norm here acts element-wise as usual.

Finding a set of optimal parameters θ^* as formulated in Eq. (36) leads to an optimization problem and hence can be solved with suitable optimization techniques. Here gradient-based methods are typically used in DL, where the gradients for the update are computed via backpropagation.[75,76] The most common optimization strategies are stochastic gradient methods, where the stochasticity refers to randomization in the subset of training samples (batches), such as the popular adaptive moments estimation algorithm *Adam*.[77]

Alternative training regimes. Although the majority of learned image reconstruction approaches applied to PAT to date have been fully supervised, one current direction within the DL community is the investigation of possible alternative training regimes. In particular, these are concerned with cases in which only a small number of input and ground-truth pairs are available. Such approaches are typically referred to as *semisupervised* or *self-supervised* training. These developments will not be covered extensively in this review, but we will discuss some possible directions on how to move away from fully supervised training in the conclusions. Roughly speaking, what these approaches have in common is that instead of requiring closeness to a known ground truth for all data pairs, we define an auxiliary measure on the goodness of reconstructions. For instance, one could think of a data consistency term, $\|\mathcal{A}\mathcal{A}_\theta^\dagger(g) - g\|_2^2$, that is used in a similar way to the concept of cycle consistency in the computer vision community.[78] Related directions use the concept of adversarial networks, in which a discriminator is used to evaluate how well reconstructions resemble "realistic" ones during the training procedure.

In summary, regardless of the chosen training regime, defining the learning task leads to an optimization problem, where we aim to find an optimal set of parameters for the network architecture with respect to a chosen measure and training set.

4.3 *Architectures for Learned Reconstruction*

The reconstruction task in PAT can be addressed in various ways, as outlined in Sec. 3, and since learning-based reconstruction algorithms are often inspired by these classical methods there is a wide range of possible approaches. In an attempt to classify learned reconstructions, we could divide the possible approaches into three classes by the number of times the physical model, the forward operator $\mathcal{A}$ or a related operator, is involved in the reconstruction process: never, once, and multiple times. Four common strategies that are directly related to classical schemes are illustrated schematically in Fig. 5. (The middle two strategies in Fig. 5 fall into the same class in this classification.) In the following, we discuss these three classes of approach on a conceptual level, giving one example of a standard architecture for each. As mentioned already, we will concentrate here on the acoustic reconstruction problem [Eq. (29)]; extensions and applications to the optical reconstruction problem will be discussed in the literature review in Sec. 5.6.

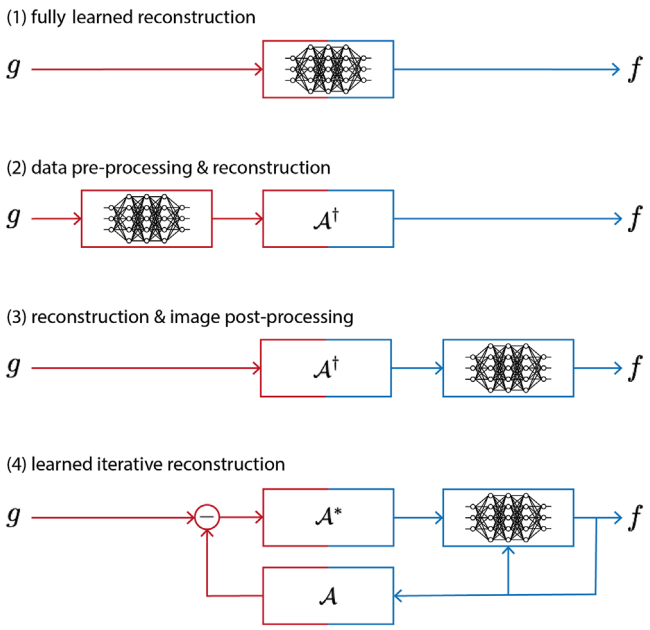

Fig. 5 Four different approaches to using a DL step (a network) within a PAT image reconstruction framework, i.e., four types of learned reconstruction operator $\mathcal{A}_\theta^\dagger$. (1) Fully learned $\mathcal{A}_\theta^\dagger = \Lambda_\theta$ [Eq. (37)]. (2) Data preprocessing and reconstruction $\mathcal{A}_\theta^\dagger = \mathcal{A}^\dagger \circ \Lambda_\theta$. (3) Reconstruction and image postprocessing $\mathcal{A}_\theta^\dagger = \Lambda_\theta \circ \mathcal{A}^\dagger$ [Eq. (38)]. (4) A learned iterative reconstruction based on gradient descent [Eq. (42)]; see Fig. 15 for another example of a learned iterative reconstruction scheme. Red indicates the data space Y and blue the image space X_f.

4.3.1 *Fully learned approach*

In the fully learned approach, the whole learned reconstruction operator $\mathcal{A}_\theta^\dagger$ is given by one network architecture, i.e.,

$$\mathcal{A}_\theta^\dagger := \Lambda_\theta, \tag{37}$$

where $\Lambda_\theta : Y \to X_f$. At first sight, such fully learned approaches seem promising as they eliminate the need for a potentially expensive reconstruction operator. However, the "no free lunch" concept applies here, as this improved reconstruction speed comes with a major limitation, which will be discussed below. First, though, we discuss the potential advantages. The forward operator $\mathcal{A} : X_f \to Y$ is nonlocal in nature. For instance, a point source $f \in X_f$ has a spatially global effect on the measurement data $g \in Y$ (although it is localized in time). Similar nonlocality of data-image relations is observed in most tomographic inverse problems. A fully connected layer has filter coefficients connecting each input to each output, and they can all be different, so it can cope with nonlocality in the data-image relation and can represent any linear mapping. In particular, the linear forward operator $\mathcal{A}$ could be learned by a fully connected network. (It could even be learned by one fully connected layer with no nonlinearity, although that would just be $\mathcal{A}$ represented as a dense matrix, which could be computed directly rather than learned.) Also an inverse mapping such as the backprojection $\mathcal{A}^\#$ in Eq. (14) can be learned by a dense layer and in particular by a composition of dense layers with nonlinearities. In a CNN, on the other hand, a layer acts only locally, meaning an output pixel is only related to nearby input pixels. A fully connected network might therefore seem, at first glance, a better choice than a CNN for this task. (Some ability to learn nonlocalities can be regained using multiscale CNNs such as the U-Net, as described below in Sec. 4.3.2) Another potential advantage of a fully learned approach, depending on the particular architecture, is that it can provide reconstructions quickly, with low latency, as no explicit model evaluation is required.

The use of a fully connected network, however, has a major limitation similar to the problem faced by matrix representations of operators for high-dimensional problems, in that we need to

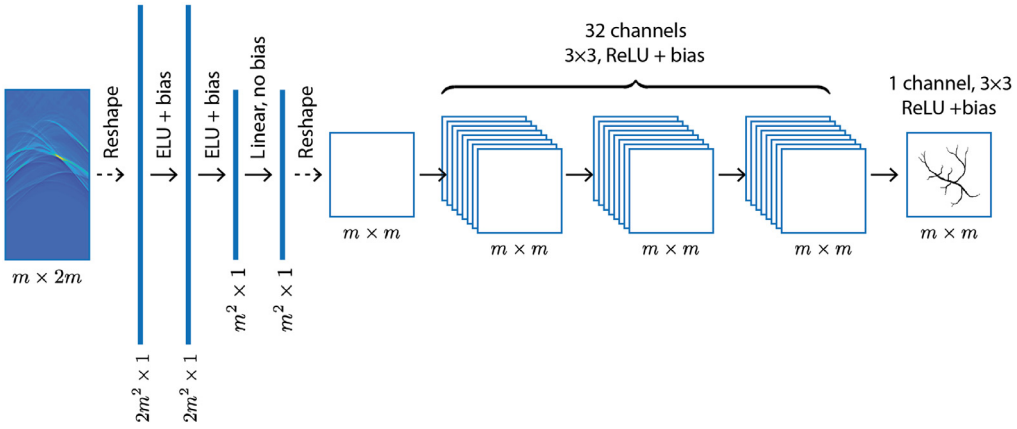

Fig. 6 A fully connected network similar to the AUTOMAP architecture.[11] Three dense layers with ELU nonlinearity and bias are followed by a small CNN of 3 layers with 32 channels followed by a final CNN layer with 1 channel for the output. The ReLU nonlinearity on the final layer imposes a non-negativity constraint.

learn a dense matrix of size $M \times T$, where M is the total number of pixels, or voxels, and T is the product of the number of detectors and the number of sampling points in time. Let us for example consider a 3D setting with $m \times m \times m = M$ voxels and $m \times m \times \mathbf{t} = T$ measurement points, where $m = 64$ and $\mathbf{t} = 128$. Then a single-dense layer, mapping between data and image space, represented in single precision (32 bit) would occupy ~500 GB. Thus reasonable applications of this approach are limited in practice to two-dimensional problems. Also the large number of learnable parameters necessitates a large training set for the training procedure to avoid overfitting to the training samples. Additionally, as the fully connected layer associates each point in the input with the output nodes, the trained network depends specifically on consistent dimensions in the data space as well as image space, and hence the acquisition geometry. For PAT, this means that one needs to train a separate network if the measurement setup changes, such as the number or location of the sensors, or the time-sampling points, or if there is a change in the sound speed distribution, for example.

One could append a small CNN to the fully connected layers to exploit spatial features in the output from the fully connected layers to produce the final reconstructions. This thought leads to the architecture known as AUTOMAP,[11] originally devised for magnetic resonance imaging. A version of this kind of network is shown in Fig. 6. In our case, the input to the network is given by the time-series of measured acoustic pressure. For the application of the fully connected layers, the input must be flattened or vectorized, i.e., reshaped into a vector, before being passed to the network. The vector output of the fully connected layers is reshaped into an image and postprocessed by a small convolutional network to produce the final output (Fig. 6). This is just one architecture that uses fully connected layers and there are many variations on this theme, some of which are discussed in Sec. 5.3.1. This network can be thought of as a learned, regularized backprojection operator (the fully connected layers) followed by a postprocessing network to improve the image (the convolutional layers). This way of seeing the network leads us directly to the next approach, in which a classical backprojection operation is first performed with knowledge of the physical model and the network acts on the output in image space.

4.3.2 *Reconstruction and postprocessing*

A major limitation of the fully learned approach is the inflexibility with regard to the acquisition geometry and acoustic properties, i.e., each network is specific to a fixed arrangement of the detectors and the sound speed. This can be overcome using an explicitly model-based (classical) reconstruction from measured time-series to image space as an initial reconstruction step. This allows for potentially higher image resolutions to be used, as the memory burden of the fully connected layers has been removed, and potentially facilitates efficient initial reconstructions using approximate and computationally cheaper models. In other words, if we substitute the

fully connected part in Fig. 6 with an explicitly known reconstruction operator $\mathcal{A}^\dagger$, we arrive at the approach of an initial analytical reconstruction followed by a learned postprocessing step. More precisely, let $\mathcal{A}^\dagger: Y \to X_f$ be an analytically known reconstruction operator that is ideally known to be robust (small changes in the input give small changes in the output). For example, $\mathcal{A}^\dagger$ could be $\mathcal{A}^\#$ or $\mathcal{A}^*$ or another approximation to $\mathcal{A}^{-1}$. Then one can train a CNN to remove the reconstruction artifacts that arise from using $\mathcal{A}^\dagger$.[78,79] In the PAT case, these artifacts can range from blurred out edges and noise to more severe undersampling and limited-view artifacts. The learned *inverse mapping* is now given as

$$\mathcal{A}_\theta^\dagger = \Lambda_\theta \circ \mathcal{A}^\dagger, \tag{38}$$

where the network $\Lambda_\theta: X_f \to X_f$ maps between the same space. The main advantage in this approach lies in the analytical knowledge of the reconstruction operator, and so the network can be designed to focus instead on exploiting the structure in reconstruction artifacts in order to remove them. Computationally, the evaluation time of the neural network is usually negligible and reconstruction times are typically limited by the complexity of the reconstruction operator. It is important to notice that the evaluation of the reconstruction operator can be decoupled from the training process and just used when creating the training data, and hence this approach is also advantageous in the training phase if the reconstruction operator is expensive to evaluate.

For learned postprocessing, typically one employs a high-capacity and particularly expressive network, i.e., one with many layers and learnable parameters, that are capable of learning complicated image priors. The most prominent architectures for this application are based on the U-Net,[80] which can be roughly described as a multiscale convolutional autoencoder. More precisely, instead of applying convolutions only on the full resolution image, the network includes down-sampling layers that reduce the image size in order to extract larger spatial features. The extracted coarse features are then subsequently upsampled to construct the final image. Intuitively, this process can be related to the principle of multiresolution analysis, such as the wavelet decomposition,[81,82] where the input image is decomposed into a fine-to-coarse basis. For image reconstruction tasks, instead of passing the reconstructed image $f_0 = \mathcal{A}^\dagger g$ directly through a network to produce the output image:

$$f = \Lambda_\theta(f_0) = \Lambda_\theta(\mathcal{A}^\dagger g), \tag{39}$$

the learning task is typically reformulated as a residual problem:

$$f = f_0 + \Lambda_\theta(f_0), \tag{40}$$

in which a correction to the initial image is learned. This is motivated by the notion that the network can be used to identify noise and artifacts to remove from the image. Such networks are often referred to as residual networks, such as a residual U-Net,[8] for which a basic architecture (with three scales) is illustrated in Fig. 7. The classic U-Net architecture[80] has five scales; we use fewer here due to the small image size in the experiments. In the encoder part of the network, the left side, in each scale we employ two convolutional layers, followed by a downsampling of factor 2. This downsampling is done by a max-pooling operation, which takes the maximum value in a window of 2×2 and reduces the image size. The numbers on top of each bar indicate the number of channels and, as can be seen, the number of channels is increased as the resolution is decreased. For the decoder part, we follow a similar approach of using two convolutions in each scale followed by an upsampling by factor two with a transposed convolutional layer. The final result is then added to the input via the residual connection. A particular design choice in the U-Net is the use of skip connections that connect the encoder and decoder parts at each scale by a concatenation. The reason for using these skip connections is two-fold. Computationally, they stabilize the training procedure by avoiding the problem of vanishing gradients in very deep networks. Additionally, the skip connections help to preserve the finer structures in the higher resolution scales. It is interesting that even though CNNs are translation equivariant, the U-Net is able to learn local dependencies due to the decomposition into the coarser scales and the resulting large receptive field. Thus the postprocessing approach proves powerful even in applications with strong local dependencies, such as limited-view problems.

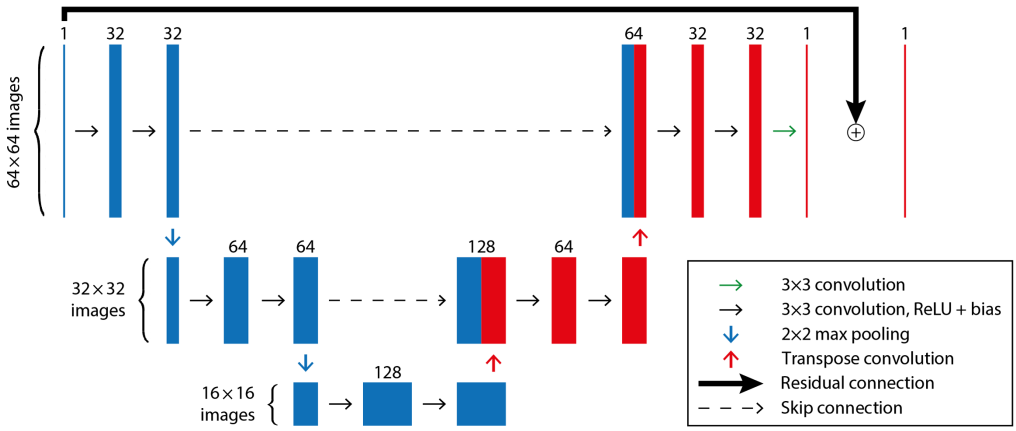

Fig. 7 A residual U-Net with three scales with two convolutional layers at each scale in the encoding and decoding paths and concatenating skip connections. The number of channels in each layer is shown above it.

On the downside, such large capacity networks tend to overfit to the training data if training data are scarce, but still need considerably fewer training samples than the fully learned approach. We will discuss this further in the experimental part in Sec. 4.5. Additionally, the output depends solely on the quality of the initial reconstruction and the capability of the network to correct for these shortcomings. Therefore, without further modifications, we cannot guarantee that the reconstructed image is optimally consistent with the data—one possibility to overcome this will be discussed next.

4.3.3 *Model-based learned iterative reconstruction*

In order to improve data consistency of the reconstructions, one could use the forward operator multiple times in the reconstruction procedure and not only for an initial reconstruction. We call such approaches learned iterative schemes, as neural networks are interlaced with evaluations of the forward operator $\mathcal{A}$, its adjoint $\mathcal{A}^*$, and possibly other hand-crafted explicitly known operators. Typically, such learned iterative schemes outperform other learned reconstruction approaches in reconstruction quality,[9,10,18,83] but come with a higher computational complexity. We also observe that this allows for the use of smaller networks, as the reduced network capacity is compensated for by providing more informative inputs to the network. We will introduce the concept with a simple learned gradient-like scheme.[10,84] For instance, minimizing the data consistency term $\mathcal{D}(f;g) = \frac{1}{2}\|\mathcal{A}f - g\|_2^2$ in a gradient descent scheme, as in Eq. (22), could be formulated as a network with the updates:

$$\Lambda_\theta(f, \nabla_f \mathcal{D}(f;g)) := f - \theta \nabla_f \mathcal{D}(f;g) = f - \theta \mathcal{A}^*(\mathcal{A}f - g). \tag{41}$$

Comparing with Eq. (21), we see that the only learnable parameter of the network is the step length $\theta \in \mathbb{R}$. Extending this idea, we can devise a learned gradient scheme using a CNN $\Lambda_\theta : X_f \times X_f \to X_f$ to compute the update in Eq. (41) and iterate the process such that

$$f^{(n+1)} = \Lambda_{\theta_n}(f^{(n)}, \mathcal{A}^*(\mathcal{A}f^{(n)} - g)), \quad n = 0, \ldots, N-1. \tag{42}$$

Here each network Λ_{θ_n} has its own set of parameters. The iterative process in Eq. (42) then defines a reconstruction operator when stopped after N iterates:

$$\mathcal{A}_\theta^\dagger(g) := f^{(N)}, \tag{43}$$

where

$$\theta = (\theta_0, \ldots, \theta_{N-1}),$$

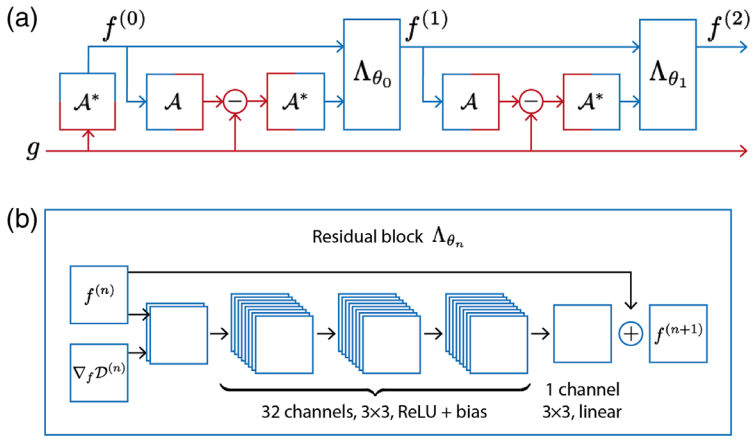

Fig. 8 (a) Unrolled network for two iterations of a learned iterative reconstruction as given by Eq. (44). (Red indicates the data space Y and blue the image space X_f.) (b) The architecture for the residual blocks Λ_{θ_n}, consisting of 3 convolutional layers of 32 channels with ReLU nonlinearity, followed by a linear convolutional layer with 1 channel to give the update for the next iterate. The network parameters θ_n are different for each block (each iteration n). $\nabla_f \mathcal{D}^{(n)}$ denotes the gradient of the data consistency $\nabla_f \mathcal{D}(f^{(n)}; g) = \mathcal{A}^*(\mathcal{A}f^{(n)} - g)$.

with an initialization, such as the adjoint of the measurement data $f^{(0)} = \mathcal{A}^* g$. The initialization is essential in this approach as it maps from data space Y to image space X_f, whereas the networks only map from X_f to X_f. Each network $\Lambda_{\theta_{n-1}}$ is a *learned updating operator* for the nth iterate and we can see a conceptual similarity of Eq. (42) to the proximal gradient descent scheme in Eq. (24), which provides a way to interpret the learned updating operator similar to a proximal mapping. Such learned iterative approaches are also known as *model-based* learned reconstructions, as the learned reconstruction operator $\mathcal{A}_\theta^\dagger$ repeatedly uses the explicit forward and adjoint operators. Clearly, this leads to an increased complexity depending on the number of operator evaluations required, but the additional knowledge supplied to the networks allows the use of smaller architectures to achieve similar, or even superior, reconstruction quality compared to the previously discussed approaches.

A basic network architecture for this task is illustrated in Fig. 8. The whole unrolled reconstruction is shown, for two iterations, in Fig. 8(a) and the architecture of the *residual blocks*, based on the residual network ResNet,[85] is shown in Fig. 8(b). At each iteration, the current reconstruction $f^{(n)}$ and the corresponding gradient of the data consistency term $\nabla_f \mathcal{D}(f^{(n)}; g) = \mathcal{A}^*(\mathcal{A}f^{(n)} - g)$ are concatenated into a two-channel input to the learned updating operator Λ_{θ_n}. This input layer is followed by 3 convolutional layers with 32 channels, 3×3 filters, ReLU nonlinearity, and bias. The final layer (3×3, linear, no bias) reduces the 32 channels to a single residual update to be added to $f^{(n)}$ such that we can rewrite the learned update equation in Eq. (42) as

$$f^{(n+1)} = f^{(n)} + \Lambda_{\theta_n}(f^{(n)}, \mathcal{A}^*(\mathcal{A}f^{(n)} - g)). \tag{44}$$

The last convolutional layer does not use a nonlinearity as the residual updates need to be able to be both positive and negative. After each residual block the intermediate result $f^{(n+1)}$ is used to compute the new gradient $\nabla_f \mathcal{D}(f^{(n+1)}; g)$, which is then passed on to the next residual block.

By using smaller networks than in the postprocessing approach, with both the current iterate and the gradient information as inputs, the networks rely less on prior knowledge from the training data and rather learn a desirable combination of both inputs. In fact, the gradient of the data consistency contains information on where the image needs improvement to fit the observed data. Just as importantly, smaller networks are less prone to overfitting and so require less training data. This aspect is further emphasized by a recent study that showed using explicitly known operators in the network architecture does indeed reduce the training error.[86] As the operator is

used repeatedly in the reconstruction process, this also allows for some flexibility in acquisition geometry that the network can be applied to. Many extensions of the basic learned gradient scheme have been proposed in the literature and applications will be discussed in Sec. 5.4.1.

4.4 *Generating Training Data*

The training set will define the features learned by the network. This essentially defines the probability distribution describing our images of interest, in other words, the prior of possible images $\pi(f)$ in the Bayesian framework [Eq. (12)]. This directly addresses the first point raised in Sec. 2.2.3 of how to learn a better prior. For many biomedical applications, it is difficult to handcraft informative priors that represent structures of interest, e.g., blood vessels, and so the alternative approach of learning a prior from a set of sample images is appealing. The choice of the training data set then becomes highly important as it defines the prior distribution. A suitable training set will have two primary features: good representation of the relevant structures, and enough variety to represent the image distribution. In established medical imaging modalities, such as magnetic resonance imaging, one can use large databases of highly sampled gold standard reconstructions as ground-truth images. In PAT, such a database is not currently available, and, furthermore, many scanner geometries are not able even in principle to collect complete data. There are therefore essentially two options for creating a training set: simulate synthetic data as realistically as possible, or define a high-quality (albeit imperfect) reconstruction using a classical inversion method as the ground truth. It is important to emphasize that the training set, together with the training regime, determines the reconstruction quality one can expect. For instance, in a fully supervised setting with only reconstructions from classical inversion methods as the ground truth, the network would not be expected to provide better reconstruction quality than the classical approach, although it may be able to compute the images more quickly. On the other hand, if augmented training sets or semisupervised approaches are employed, more complicated priors might be learned and classical methods may be outperformed in reconstruction quality.

4.4.1 *Synthetic training data*

The first step in the creation of synthetic training data is to define the ground-truth images $f_i \in X_f$ for $i = 1, \ldots, \Im$. From these ground-truth images, we can then simulate the corresponding synthetic measurement data $g_i \in Y$ according to Eq. (9). Note that this includes the simulation of measurement noise. The pairs of ground-truth image and synthetic measurement data then define the training set $\{(g_i, f_i) \in Y \times X_f\}_{i=1}^{\Im}$. In PAT, we are often interested in imaging vasculature, so we need a way to create a large enough set of images with relevant vessel structures. A standard way to obtain such structures is to use other imaging modalities that provide images or volumes of vessels and then to segment them to extract the relevant vessels as a ground-truth image. For the following experiments, we designed two datasets from different image databases. The first set was created from a set of lung CT scans[87] via vessel segmentation and projection to two dimensions, and the second set was taken from retina scans[88] with a

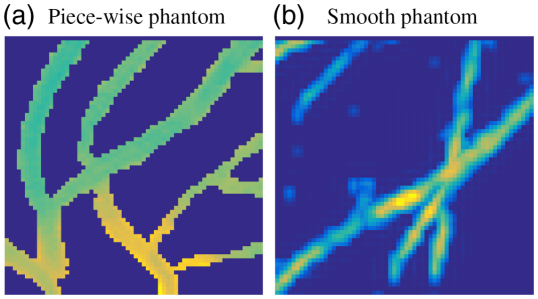

Fig. 9 Example images for the two data sets used in the experimental section: (a) phantoms promote piece-wise constant or linear features and (b) promotes smoother features.

segmentation provided. As Fig. 9 shows, these two sets have very different characteristics, one having piece-wise constant features the other smoother features; in other words, the prior distributions $\pi(f)$ are different. As we will see, this difference will have a major impact on the reconstructions obtained depending on how the two training sets are used in the training and testing. (In this particular case, as an alternative to the segmentation of vessel structures from other modalities, one could imagine creating ground-truth images by, for example, using vessel growing algorithms[89] to create a large set of synthetic training data.)

4.4.2 *Experimental training data*

An alternative to synthetic training data is to use measured data for the creation of the training set, i.e., start with measurement data $g_\mathrm{i} \in Y$ and create a reference reconstruction $f_\mathrm{i} \in X_f$. In this scenario, ideally one would have complete measurement data available that can be used to create a high-quality reference reconstruction, for instance with a variational approach, Sec. 3.1.4. Then one can either train a network on the pairs of $(g_\mathrm{i}, f_\mathrm{i})$ to speed up reconstruction times, or one can, retrospectively, undersample the measurement data to obtain $\tilde{g}_\mathrm{i}$ and train with pairs $(\tilde{g}_\mathrm{i}, f_\mathrm{i})$ to improve reconstructions from undersampled measurements. In our experience, we have found that in the application to real measurement data, it is essential to include some experimental data in the training procedure, as structures and noise can vary significantly from synthetic to experimental data.

4.4.3 *Transfer training*

A third option is to combine synthetic and experimental data. This is usually a good idea if one does not have sufficient measurement data available. Here one can exploit a concept known as *transfer training* or update training. We refer the reader to two discussions on the topic.[90,91] In our case, the underlying idea is to perform pretraining with a large set of synthetic training data that represents a good prior for the targets we are interested in. Then after the first training phase on the synthetic data, we can update the obtained parameters with a shorter training on the limited set of available measurement data. This fine-tunes the network parameters to the characteristics of the experimental data, for instance by adjusting threshold capabilities to the noise level. This update is typically done with a reduced learning rate. Sometimes, rather than updating all parameters of the network, the majority are fixed and only the first and/or last layers are updated with the new data.

Finally, we remark that here self-supervised training regimes, as mentioned in Sec. 4.2.4, might be promising in the transition to experimental measurement data, although this area has not yet been widely explored.

4.5 *Comparison of Learned Image Reconstruction Approaches*

In this section, we use the synthetic data sets introduced above to examine the performance of the three learned image reconstruction approaches described in Sec. 4.3 with respect to accuracy and robustness. We consider a 2D limited-view scenario with a line detector at the top of the domain, as illustrated in Fig. 10. For simplicity, we create a matrix representation of the acoustic forward model as described in Sec. 3.1.5 by sampling the forward operator with k-Wave.[92] Since this section serves in part as a tutorial, we will describe the individual steps required to set up the experiments in detail.

4.5.1 *Experimental design*

Here we describe the steps necessary to train and evaluate the "reconstruction and postprocessing" reconstruction approach as outlined in Sec. 4.3.2. This will cover all the concepts needed to set up the examples for the other learned reconstruction approaches.

1. *Data acquisition geometry and definition of the forward operator.* The essential first step is the definition of the imaging setup under consideration, which also defines the forward

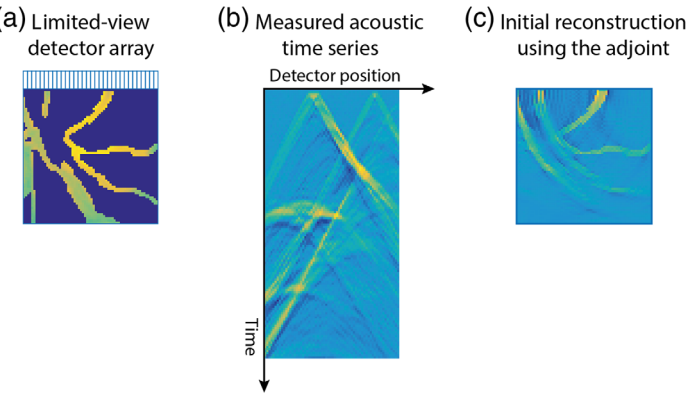

Fig. 10 Illustration of the experimental setup for the examples. (a) We consider here a limited view geometry in two dimensions with a line detector on the top of the domain. (b) The corresponding time series measurements. (c) The initial reconstruction obtained by application of the adjoint to the measurements.

operator $\mathcal{A}$. Here we chose a limited-view planar acquisition geometry in a two-dimensional domain, see Fig. 10, and we use k-wave[92] for the PA time series simulation in MATLAB. For flexibility in the training data creation and reconstruction, we use a matrix representation of the forward operator by sampling each pixel in the image domain; this is done in the script: `createForwMat.m`. The resulting matrix is saved to disk, so it can be loaded within the learning framework in Python for data creation and reconstruction in the following. To enable readability by Python, we added the flag -v7.3 in MATLAB when saving the mat-file. Additionally, when loading the matrix in Python, it will be transposed, so we transpose the matrix before saving.

2. *Training data creation.* First, we need to choose the set of ground-truth images $\{f_i\}_{i=1}^{\mathfrak{I}}$ we want to use for the training, i.e., one or both of the sets described already and shown in Fig. 9. Then we create the corresponding synthetic measurement data using the matrix form of $\mathcal{A}$ to create $g_i = \mathcal{A}f_i + \varepsilon_i$, where ε_i is normally distributed noise added to the measurements with standard deviation of 1% of the maximum measurement amplitude. We then create the initial reconstruction using the adjoint, such that $f_i^{\text{rec}} = \mathcal{A}^* g_i$. Recall that in the matrix representation, the adjoint corresponds simply to the transpose. The training set $\{(f_i^{\text{rec}}, f_i)\}_{i=1}^{\mathfrak{I}}$ for supervised training has now been generated. Test and validation sets can be created in the same way. These preparations are done in the script: `callNetwork.py`. For the other experiments, with the fully learned approach and learned iterative schemes, we can just use the generated data as input and hence the set is given by $\{(g_i, f_i)\}_{i=1}^{\mathfrak{I}}$.
3. *Network selection and training regime.* Given the training set, we can now set up the network and define the training regime. All the relevant functions can be imported from the supplied script: `PATnets.py`. For this case, we choose a classic residual U-Net for the network Λ_θ, and as a loss function the classic squared ℓ^2-norm for supervised training. We note, that the code package provides a set of standard architectures that can be called instead of the U-Net. Then the optimization problem reads as

$$\theta^* = \arg\min_{\theta\in\Theta} \frac{1}{\mathfrak{I}} \sum_{i=1}^{\mathfrak{I}} \|\Lambda_\theta(f_i^{\text{rec}}) - f_i\|_2^2. \tag{45}$$

The optimization is performed with the Adam algorithm, initial learning rate 10^{-4}, batch size 4, and a total of 5×10^4 iterations.

4. *Training supervision and evaluation.* During the training, we need to ensure both that our cost function is minimized and converges and also that the learned parameters generalize to other samples not contained in the training set. To achieve this, we use the visualization support provided by tensorboard,[93] which can then be called locally from a web browser to

provide real-time supervision of the training procedure. After the training, the optimal set of parameters θ^* can be saved for later evaluation, or a direct evaluation can be performed. For evaluation, we load the test set and record the average reconstruction quality.

4.5.2 *Reconstructions: robustness and generalization*

In this section, we will evaluate the three reconstruction approaches described in Sec. 4.3, fully learned, postprocessing, and learned iterative reconstruction, following the four-step process outlined above. In particular, we will examine how these methods compare in generalizability with respect to changes in the data sets they are trained on. For that purpose, we will consider three scenarios for the training and test sets as follows.

(i) *Consistent sets.* Trained on the retina data [Fig. 9(b)] of 1000 samples and tested on a separate but consistent test set from the retina data with 151 samples. This corresponds to a scenario where the priors are the same, $\pi_{\text{test}}(f) = \pi_{\text{train}}(f)$.

(ii) *Different test sets.* Trained on the retina data [Fig. 9(b)] and tested on the lung CT set [Fig. 9(a)] with 151 samples. This corresponds to a scenario where the priors are different, $\pi_{\text{test}}(f) \neq \pi_{\text{train}}(f)$.

(iii) *Combined set:* Trained on both data sets with a total of 3760 samples and tested on a separate combined test set with 308 samples. Here the priors are consistent, but more complicated than in (i). We emphasise that this training set is also larger.

We trained the three reconstruction approaches for each scenario using the same training regime, as outlined in Sec. 4.5, with minor tuning as necessary to ensure the parameters are close to optimal. This ensured the results were representative and allowed useful conclusions to be drawn. Nevertheless, as we will see below, not all of these architectures are conceptually the right choice for the scenarios under consideration and it was not possible to improve the performance significantly through further parameter tuning. Note that the fully learned approach uses a regularizer of the learned parameters $\|\theta\|_1$ to reduce overfitting.

Let us first discuss the obtained reconstructions from a visual perspective. The results obtained for the first case (i) are displayed in Fig. 11. Most striking here is the result obtained by the fully learned approach, which clearly falls short in reconstruction quality compared to the other two approaches. We observe that this is primarily due to the limited size of the training data and hence the network strongly overfitting, even though we use regularization to reduce this. This is clear from the training error plots shown in Fig. 12. In contrast, the other two approaches correctly learned some form of representation of the prior π_{train} from the training data. Consequently, the reconstructions for case (i) are visually close to the ground truth. In particular, we can see a good reconstruction quality close to the detector, but on the boundary where limited-view artifacts are stronger the reconstructions lose quality.

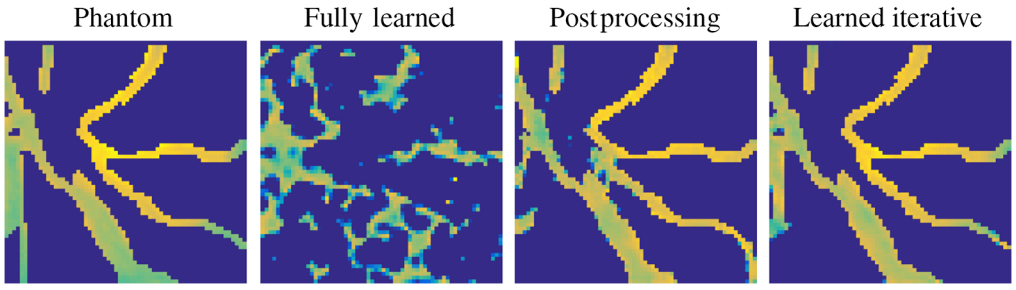

Fig. 11 Reconstructions obtained for test case (i) with consistent priors $\pi_{\text{test}}(f) = \pi_{\text{train}}(f)$. Trained and tested on the piece-wise constant phantoms. The fully learned approach does not perform satisfactorily due to strong overfitting to the training data, whereas the other two approaches are able to produce quantitatively and qualitatively superior results, but still exhibit errors in the reconstruction.

Fig. 12 Training curves for the fully learned approach and learned gradient descent in comparison for case (i), exported from the tracking tool tensorboard. Although both approaches have a tendency to overfit the training data during training, the fully learned approach does suffer more compared to the learned iterative reconstruction.

For the second case (ii), the results are shown in Fig. 13. It is clear, on the first sight, that the networks produce results according to the learned piece-wise prior from the training data, as one would expect. Additionally, all the algorithms show a deterioration in reconstruction quality from the consistent case (i). For the postprocessing and learned iterative scheme features close to the detector are to some extent correctly reconstructed, but they struggle further away. The fully learned approach, due again to strong overfitting, produces a result with very limited resemblance to the ground truth. One interesting feature is that the networks, and especially the learned iterative scheme, tend to smear out features where there is uncertainty in the reconstruction.

In the final case (iii), the training samples are combined and so the size of the training data set is increased. The results are shown in Fig. 14. There is a clear improvement over the second case, as the test data are consistent with the mixed prior, and both approaches that use a model in the reconstruction do fairly well in reconstructing the target. There is a slight influence of the mixed prior visible in the results, as the reconstructions for the piece-wise constant phantom exhibit some smoother features related to the smooth phantoms. Finally, the fully learned approach seems to struggle with the mixed priors, but the reconstruction is still arguably closer to the ground truth than in the other cases, as the increased training data reduced the overfitting. Nevertheless, the result is still not satisfactory.

These observations are supported by the quantitative values shown in Table 1, which shows the mean and standard deviation over the whole test data for each case. We computed the peak-signal-to-noise ratio (PSNR), which is a logarithmically relative root mean squared error and hence related to the quantity we minimized in the training. Additionally, we computed the structural similarity index measure (SSIM) as an indication of the perceived similarity in the

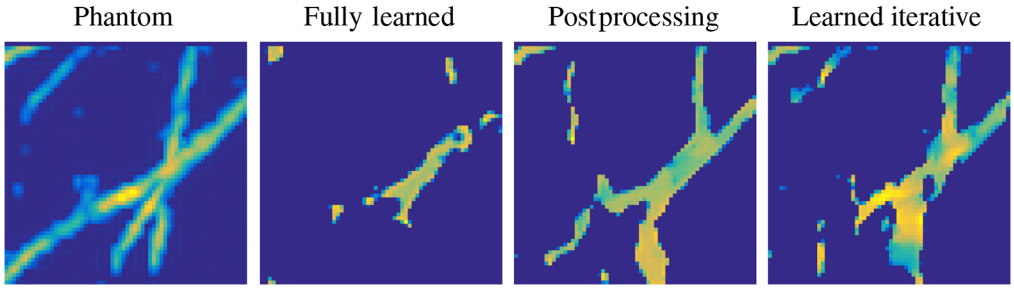

Fig. 13 Reconstructions obtained for test case (ii) with inconsistent priors $\pi_{\text{test}}(f) \neq \pi_{\text{train}}(f)$. Trained on the piece-wise constant phantoms and tested on smooth phantoms. All methods struggle to produce satisfactory results and one can see that the piece-wise constant prior from the training data is reproduced by each method.

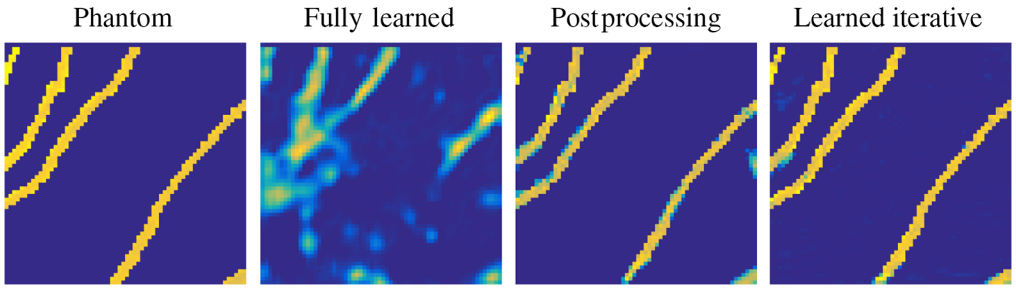

Fig. 14 Reconstructions obtained for test case (iii) with consistent priors $\pi_{\text{test}}(f) = \pi_{\text{train}}(f)$. Trained and tested on combined phantoms with piece-wise constant as well as smooth features. The reconstruction quality of the fully learned approach improved slightly compared to the other test cases due to the larger training set, but it is clearly outperformed by both methods that use the model in the reconstruction pipeline.

Table 1 Quantitative values for the three test cases in SSIM and PSNR

	case (i)		case (ii)		case (iii)	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
Fully learned	0.624 ± 0.181	13.34 ± 3.54	0.491 ± 0.182	16.00 ± 2.19	0.592 ± 0.170	17.34 ± 3.95
Postprocessing	0.946 ± 0.042	21.57 ± 5.85	0.570 ± 0.185	16.56 ± 2.23	0.902 ± 0.133	23.22 ± 5.07
Learned iterative	0.983 ± 0.025	28.76 ± 8.10	0.679 ± 0.165	18.28 ± 2.35	0.949 ± 0.089	28.04 ± 5.82

reconstructions. We can see that the postprocessing and learned iterative schemes perform better in this test, but with a strong deterioration when changing the prior distribution for the test data, as is also seen visually for case (ii). For case (iii) neither method using a model showed a strong improvement over case (i), in fact both methods deteriorate in terms of SSIM as the priors are not perfectly reproduced, although PSNR is either stable or improves for the postprocessing. For the fully learned approach, PSNR improved considerably with the larger training size, although SSIM slightly deteriorated, most likely due to the difficulty in reproducing the prior correctly. This is further an indicator of the large quantity of data needed for the fully learned approach to work well. Similar observations are made by Baguer et al.:[94] in their overview, they show that a fully learned approach only performs well with large amounts of data, which explains in parts the poor performance in this limited data setting.

5 Deep Learning in PAT—Literature Review

The majority of the journal articles in which DL techniques have been applied to PAT image reconstruction are concerned with the acoustic part of the reconstruction and there are fewer papers tackling the optical part. Most of the sections that follow will therefore focus on the acoustic reconstruction. The papers concerned with DL approaches applied to the optical reconstructions will be reviewed in Sec. 5.6. We also draw attention to related reviews on the matter of optical imaging and/or learned image reconstruction.[12,95–97]

5.1 *Postprocessing*

Early approaches to learned image reconstruction concentrated on the reconstruction and postprocessing approach as outlined in Sec. 4.3.2. The work by Antholzer et al.[79,98,99] investigated the approach of using filtered backprojection (Sec. 3.1.1) to reconstruct an initial image and then

train a U-Net, with five scales, to do postprocessing. This was in a sparse and limited-view data setting and followed the residual learning approach[8] given by Eq. (40). Similar to our observations in Sec. 4.5, the authors report that consistent training and test data, i.e., $\pi_{\text{test}}(f) \approx \pi_{\text{train}}(f)$, is crucial for optimal performance of the trained network;[79] this seems to be more so in the case of limited-view detection geometries. This observation was confirmed and clearly demonstrated in the study by Guan et al.,[100] who proposed a dense U-Net to ameliorate this negative effect. Other extensions have been proposed too: using a leaky ReLU nonlinearity[101] or using the first iterate of a model-based iterative approach (Sec. 3.1.4) instead of a backprojection-type reconstruction.[102] Awasthi et al.[103] proposed combining a reconstruction obtained with the adjoint with the first iterate of an iterative algorithm in a learned fusion process.

In comparison to other approaches, U-Net-based networks generally performed better than other architectures, e.g., compared to a simple three-layer CNN,[98] VGG,[101] and compared to applying U-Net directly to the measurement data g,[104] especially with respect to robustness. It is interesting that Antholzer et al.[99] compare their results to a classic ℓ^1-regularization approach for compressed sensing and report that when the system matrix is randomly sampled, and hence undersampling artifacts change as well, the classical variational approach clearly outperforms the network-based postprocessing approach. This enforces the observation that consistent training and test data is needed for this approach to be successful.

5.1.1 *Application to* in vivo *imaging*

In an extensive study, the U-Net-based postprocessing approach was successfully applied to *in vivo* measurements[105] and showed clear improvements over backprojection-based algorithms when the data were undersampled or detected over a partial aperture (limited-view problem). Hariri et al.[106] showed that this approach can improve *in vivo* imaging when using low-fluence sources. The observation of improved visual performance for *in vivo* applications was also reported in other studies.[107,108]

5.1.2 *Extensions of the postprocessing approach*

The primary problem with the postprocessing approach is that the result depends on a network that is determined only by the information content of the training data and not the physics of the problem. To tackle this, Antholzer et al.[79,98] proposed a nullspace projection to ensure data consistency after postprocessing. In other words, only components in the nullspace $\mathbf{N}(\mathcal{A})$ of the forward operator $\mathcal{A}$ are added to the reconstruction and as such do not change the data consistency term $\|\mathcal{A}f - g\|_2^2$. The solution therefore takes the form

$$f = f_0 + \mathcal{P}_{\mathbf{N}(\mathcal{A})}\Lambda_\theta(f_0), \tag{46}$$

where $\mathcal{P}_{\mathbf{N}(\mathcal{A})}$ denotes the orthogonal projection to the null space. Schwab et al.[109,110] combined postprocessing by a U-Net with a learning-based filter in the backprojection step [κ in Eq. (17)] to improve initial reconstructions from limited-view measurements.

Recently, LED-based excitation systems have become popular but because of their low-power output many averages (thousands) are required to improve the signal-to-noise ratio. The resulting long-duration measurements are sensitive to motion artifacts. To compensate for this, Anas et al.[111] proposed using a recurrent neural network, a convolutional LSTM network,[112,113] to exploit the temporal dependencies in the noisy measurements. They report a considerable improvement over single-frame postprocessing. In our opinion, this explicit consideration of the temporal aspect with recurrent units is more promising for low-power systems than just postprocessing with a U-Net.[114] With a similar motivation to expand on the information before postprocessing, Kim et al.[115] proposed to use the delay part of delay-and-sum but without taking the sum [Eq. (14) without the integral]. The resulting 3D input is then processed and collapsed by a U-Net to produce the final reconstruction.

5.1.3 *Beyond fully supervised training regimes*

A possibility to provide an uncertainty estimation for reconstructed images by the postprocessing approach was investigated by Godefroy et al.[116] The authors proposed to train a U-Net with Monte Carlo (MC) dropout to provide reconstructions and an uncertainty estimate. Here a set of images is sampled with the MC dropout procedure, which provides a reconstruction (the mean of these images) and a standard deviation indicating instabilities in the reconstruction.

Finally, we observe that the approaches here were all trained in a supervised manner by minimizing an explicit loss function given by the ℓ^1 or ℓ^2 error. In a recent study, Vu et al.[117] explored the possibility of using a generative adversarial network (GAN) to process the image. In this setting, the U-Net is interpreted as the generator producing a clean PA image and the discriminator acts as the loss function evaluating reconstruction quality. GAN-based approaches lead the way to applications where no paired training data are available.

5.2 *Preprocessing*

In a similar manner to the previous approach of using a network for postprocessing reconstructions, one can instead focus the learning task on the data side and then use a classical reconstruction algorithm (Sec. 3) to obtain the PA image; see Fig. 5. In this sense, we reformulate the learned postprocessing reconstruction operator in Eq. (38) to its analogue for learned preprocessing as

$$\mathcal{A}_\theta^\dagger = \mathcal{A}^\dagger \circ \Lambda_\theta. \tag{47}$$

Here the network Λ_θ can act as a denoising and artifact removal step on the data side to make the inversion step easier (essentially it changes the learning task from an inversion step to a denoising step).

5.2.1 *Artifact removal for source localization*

Defining a clear purpose for an application enables the formulation of task specific processing algorithms, for instance in the case of tracking applications as explored in the work by Allman et al.[118–120] Here the aim is to localize a point-like source and to this end it is essential to distinguish clearly the true signal from noise and artifacts. The authors propose to use an object detection and classification approach to separate artifacts from the true signal. Their approach is based on a network architecture known as faster R-CNN[121] that produces a classification between signal and artifact, a confidence score and locations as a bounding box. After a subsequent artifact removal step, the final PA image is reconstructed using beamforming (Sec. 3.1.1). The authors show that their networks for accurate source location trained on simulated data can be transferred successfully to experimental data,[118] as well as *ex vivo* and *in vivo*[122] measurements.

5.2.2 *Sampling and bandwidth enhancement*

The PAT reconstruction problem is well-posed if perfect measurement data are available (see Sec. 2.2). One approach to preprocessing is, therefore, to aim to produce ideal data for the inversion from the nonideal measurement data. This was investigated in the work by Awasthi et al.[123,124] The authors considered a sparse data (but full-view) scenario with limited bandwidth detectors and trained a network to produce high-quality data from the degraded input. In particular, the network attempted to upsample the data from 100 detectors to 200, to denoise it, and to increase the bandwidth. The improved data were then reconstructed by filtered backprojection (Sec. 3.1.1). Two architectures were used for Λ_θ: a simple seven-layer CNN[123] and a U-Net-based architecture.[124] In general, the U-Net architecture performed better, but it is interesting that for low noise, the simple CNN architecture was highly competitive. Translation to *in vivo* measurements without retraining was successful for both methods.[123,124]

Conceptually, such a preprocessing approach can be understood as learning a representation of the likelihood $\pi(g|f)$ conditioned with a training set for the images f. Nevertheless, the reconstruction quality is essentially limited by the goodness of the preprocessed measurement data and hence we believe this approach is only viable in fairly simple measurement scenarios, such as the tracking applications discussed above.[118,120]

5.3 *Fully Learned*

When considering a fully learned reconstruction, it is important to keep in mind that the measurement data $g \in Y$ lies in a different spatiotemporal space than the reconstructed images $f \in X_f$ and as such a mapping between the spaces $Y \to X_f$ needs to be constructed. In Sec. 4.3.1, we discussed the nonlocal nature of the mapping, and that in principle a fully connected layer can account for this. Although the mapping may, therefore, be done by a fully connected layer, we nevertheless clearly saw in Sec. 4.5 that with a limited amount of data the fully connected layers are hard to train to achieve high-quality reconstructions. Additionally, we observed that the CNN following the fully connected layers did most of the visual "heavy-lifting" for the final reconstruction. This observation is in line with what has been reported in the literature, as discussed below in Sec. 5.3.1. Following this idea, Shang et al.[125] proposed a two-step approach, where first a fully connected layer is trained to transform measurements into the image space, and then a U-Net is trained to process the result while the weights of the fully connected layer are fixed.

5.3.1 *Convolutional approaches*

Even though there is no clear theoretical justification to use a CNN directly to transform a spatiotemporal signal from Y into an image in X_f, as they learn spatially equivariant mappings, many studies in fact explore this scenario. The strength of convolutional-based networks lies in their capability to exploit local relations in the data and as such can deal efficiently with noise in the input. The issue of spatial invariance can be overcome using multiple pooling layers to increase the receptive field of the network, and the representation on the coarse scales effectively encodes the locality of the information. This implies that large multiscale networks are needed to transform the signal into the sought-after PAT image effectively. In early studies by Waibel et al.[104] and Gröhl et al.,[126] it was shown that using an asymmetric U-Net to reconstruct the PA image directly from raw sensor data is feasible in a limited-view setting. In comparison to a postprocessing approach using a U-Net, it was competitive in terms of mean reconstruction error, but exhibited a higher variance in reconstruction error. To overcome this, various solutions have been investigated in the literature, including enlarging the network to increase the capacity.[127,128] Others proposed to introduce a preprocessing step to provide more informative input to the network, either by a hand-crafted interpolation[129] or even learned preprocessing with a separate CNN.[130] Note that in the latter case, the transformation after the preprocessing is in fact done by a dense layer and hence is the closest to the AUTOMAP architecture discussed in Sec. 4.3.1. In both cases, the preprocessing step seems to be essential to provide an input, reduced in dimensionality, to the network performing the transform to the image space. Additionally, Tong et al.[130] motivated the preprocessing architecture based on the universal backprojection [Eq. (16)] and provide time-series and as well as the time-derivative to the network. Lan et al.[131] reduce 120 time series to 1 by summing them with delays, then feed this single time series into a LSTM network followed by a fully connected layer and a subsequent CNN to form the reconstructed image.

Following the discussion in Sec. 5.2.1, there are situations in which the full reconstruction problem can be simplified to the case where only a source location must be found. This can be achieved by, for example, using a feature detection network[132] or first forming a reconstructed image using an extended U-Net then converting to a numerical value for the source location.[133,134]

5.3.2 *Discussion of fully learned approaches*

In summary, the more advanced fully learned approaches seem to provide a slight improvement over reconstruction followed by postprocessing with a U-Net. However, the fully learned approach does not explicitly include the acquisition geometry and sound speed in the inversion procedure. Although this generality might conceivably be useful, it means that for the network to be robust to changes in these experimental parameters, the training data must account for the full range over which they might vary. As we see it, the fully learned approach might therefore be useful in cases where a measurement device is available with corresponding data-image pairs (g, f) to be used as training data, but the acquisition geometry and other underlying parameters needed for reconstruction are not known. (i.e., if it is a "black box" with examples of known inputs and outputs but the parameters implicit in $\mathcal{A}$ are not known.) A fully learned approach would then provide a way to improve the imaging pipeline without having to go through the potentially difficult procedure of determining the instrument characteristics. Finally, following our observation in Sec. 4.5, the fully learned approach needs substantially more training data than other approaches that involve $\mathcal{A}$ explicitly. This might constitute a major limitation when transitioning to experimental measurement data, where data availability is inherently scarce. Nevertheless, preprocessing approaches, as in Refs. 129 and 130, are potentially promising in reducing the hunger for training data.

5.4 *Learned Iterative Reconstructions*

Learned iterative schemes, as described in Sec. 4.3.3, are model-based reconstructions that use known forward and adjoint models within a learned update. Given the reconstruction operator $\mathcal{A}_\theta^\dagger$ in Eq. (43), defined by the iterates in Eq. (42), we can formulate the training task in an end-to-end manner. This means, given paired training data $(g_i, f_i) \in Y \times X_f$, then an optimal parameter θ^* is found by solving the optimization problem in Eq. (36), where the loss function is given as

$$L_\theta(f, g) \coloneqq \|\mathcal{A}_\theta^\dagger(g) - f\|_2^2 \quad \text{for}(f, g) \in X_f \times Y. \tag{48}$$

Computing the gradient of the loss function with respect to θ requires performing back-propagation through all of the unrolled iterates $n = 0, \ldots, N-1$. This requires storage as well as evaluation of forward and adjoint in each training step for each iterate and hence can be computationally burdensome and so has mostly been demonstrated in 2D imaging scenarios.

In Ref. 135, the basic learned iterative reconstruction approach has been applied with an extension to simultaneously reconstruct sound speed as well, which constitutes a learned version of Ref. 136. Following the illustration in Fig. 8, the authors suggested to also add a residual connection updating a sound speed estimate together with the reconstruction.

5.4.1 *Learned primal dual in 2D*

For reconstructions in PAT, the work by Boink et al.[137–139] has demonstrated the robustness of these learned iterative schemes to a number of *in silico* phantoms as well as in an experimental study. The authors consider an extension to the learned gradient schemes introduced above called *learned primal dual*[18] (LPD) based on the successful primal-dual hybrid gradient method[140] (also known as the Chambolle–Pock algorithm). The LPD method can be formulated in a similar manner to Eq. (42) by learning updating operators in the primal space X_f and the dual space Y:

$$h^{(n+1)} = \Gamma_\theta(h^{(n)}, \mathcal{A}f^{(n)}, g), \tag{49}$$

$$f^{(n+1)} = \Lambda_\theta(f^{(n)}, \mathcal{A}^* h^{(n+1)}). \tag{50}$$

In this case, the network Γ_θ operates in data space Y, whereas the network Λ_θ operates in image space X_f. See also the illustration in Fig. 15, in which it is clearly seen to be an extension to the learned iterative scheme in Fig. 5(b). In their work,[137–139] the authors examined the robustness of

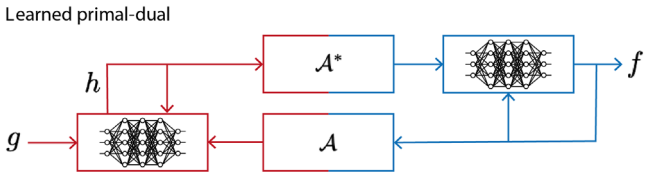

Fig. 15 Schematic of the *learned primal-dual* reconstruction scheme, a learned iterative reconstruction based on the primal-dual hybrid gradient algorithm [Eq. (49)]; see also Fig. 5. Red indicates the data space Y and blue the image space X_f.

LPD with respect to changes in the target, including the contrast, background, structural changes, and noise level. They found that if the network is trained only on the basic training data, it generalizes fairly well with respect to noise (1 dB degradation in PSNR) and structural changes (3 dB), but is most sensitive to changes in background (7 dB) and contrast (11 dB).[138] Additionally, the authors combine their learned reconstruction with a joint segmentation that is learned with the same network as an additional output and is shown to provide increased robustness compared to a reconstruction by filtered backprojection and segmentation with U-Net.

5.4.2 *Learned iterative reconstructions in 3D*

As already indicated, learned iterative reconstruction methods are ideally (and typically in 2D) trained in an end-to-end manner. Although this can provide an optimal set of network parameters, if suitable optimization procedures have been used, it also comes with two computational challenges. First, the memory footprint of storing and manipulating the network tends to be large and exceeds single GPU configurations making it necessary to use costly (and often less readily available) multi-GPU clusters. More significantly, however, during training the loss function must be evaluated several times, and each of these involves evaluating the forward and adjoint operators for each iterate. This quickly leads to unreasonable training times for 3D images, especially when considering large volume sizes, i.e., many voxels, and accurate forward models.

To overcome this limitation, Hauptmann et al.[83] proposed greedy training for learned gradient schemes for 3D PAT. That is, instead of looking for a reconstruction operator that is optimal end-to-end, only iterate-wise optimality is required. For the learned gradient scheme in Eq. (42), this amounts to the following loss function for the nth unrolled iterate:

$$L_{\theta_n}(f^{(n)}, g) = \|\Lambda_{\theta_n}(f^{(n)}, \mathcal{A}^*(\mathcal{A}f^{(n)} - g)) - f\|_2^2 \tag{51}$$

given the output of the previous iterate $f^{(n)} := \Lambda_{\theta_{n-1}}(f^{(n-1)}, \mathcal{A}^*(\mathcal{A}f^{(n-1)} - g))$ and initialization $f^{(0)} = \mathcal{A}^*g$. It is important to note that, as only iterate-wise optimality is required and the parameters θ_n are not jointly minimized over all iterates, such a greedy scheme constitutes an upper bound on the minimized loss function for end-to-end networks. Nevertheless, this renders the training procedure feasible since training can be separated from the evaluation of the model; the gradient of the data consistency term $\mathcal{A}^*(\mathcal{A}f^{(n)} - g)$ used in Eq. (51) can be computed before the parameter optimization is performed. In their study,[83] the authors showed that in this way a learned iterative reconstruction algorithm can be trained for realistic 3D volumes of size $240 \times 240 \times 80$ in a limited-view acquisition geometry. The results suggest that improved reconstructions can be obtained compared to both postprocessing with a U-Net and iterative reconstruction with total variation regularization. Application to *in vivo* measurement data was presented after transfer training was performed, as outlined in Sec. 4.4. As the authors use an accurate, full-wave, solver for the forward and adjoint operators, reconstruction times were still slow in the order of minutes, but with an 4× speed-up compared to iterative reconstruction with total variation.

In a follow-up study,[141] the authors considered the use of a faster but approximate forward model to overcome the slow reconstruction times. Here the fast k-space method discussed in Sec. 3.1.2 was used for the inverse as well as the forward propagation model, but as the forward model includes a singularity[142] this results in an approximate gradient only. Following the greedy

training scheme [Eq. (51)], the networks learned to reduce the resulting artifacts to produce a useful update. Using this fast approximate forward model, the authors achieve a reconstruction time in the order of seconds, more precisely an 8× speed-up compared to their previous learned approach,[83] and 32× compared to iterative reconstruction with total variation. Results are presented for *in vivo* measurements of a human target.

Finally, Yang et al.[143] extended the previous study using an approximate model[141] using recurrent inference machines[84,144] for the network architecture. This way the authors are able to improve reconstruction results for *in silico* experiments in 2D by 2 dB in PSNR. In conclusion, learned iterative approaches seem to provide an improvement in reconstruction quality compared to other learned reconstructions discussed in this review, but come with the major limitation of reconstruction speed due to the repeated application of the forward model and its adjoint.

5.5 *Hybrid Approaches*

From the previous sections, it is apparent that most approaches, while having clear advantages, come with their own shortcomings. To try to mitigate these, a few groups have investigated hybrid approaches. For instance, to overcome the missing model dependence in the fully learned approach, the work by Lan et al.[145–147] proposed augmenting the end-to-end approaches[127,128] by additionally feeding the network a reconstructed image, either directly into the network at a suitable location[147] or with a separate processing branch.[145,146]

5.5.1 *Augmented analytical approaches*

Another route is to incorporate learned methods into classical inversion approaches more explicitly than the learned iterative approaches in Sec. 5.4. For instance, by formulating a variational problem with a learned regularizer in the variational formulation of Eq. (23), such that the functional to be minimized becomes

$$\mathcal{E}(f) = \frac{1}{2}\|\mathcal{A}f - g\|_2^2 + \alpha\Lambda_\theta(f), \tag{52}$$

and explicit minimization of $\mathcal{E}(f)$ can be performed in an iterative algorithm. This approach has been proposed as the NETT framework[148] and applied to PAT.[149] The strength of this approach is in the emphasis on the model in the data consistency term and convergence guarantees under certain conditions,[148] but time consuming iterative minimization with the explicit forward and adjoint models is still needed, similar to the learned gradient schemes. Another possibility for an augmented analytical approach is presented by Schwab et al.,[150] who consider a data-driven extension of the truncated singular value decomposition, where the network is trained to produce the singular vectors corresponding to small singular values to improve reconstruction quality. We emphasize that such augmented analytical approaches are especially important where reconstruction convergence guarantees are needed, such as in critical clinical applications, but they seem to fall short in visual performance compared to the most advanced learned reconstruction approaches.

5.6 *Optical Inversions*

There is not, to date, a large literature using DL to tackle the optical inversions in PAT image reconstruction (see Secs. 2.2.4 and 3.2). What there is all assumes that the acoustic inversion has already been solved, which is to say the initial acoustic pressure distribution f is either given as the basic measured quantity or has already been estimated by solving $\mathcal{A}^{-1}g$. The inverse problems subsequently tackled fall largely into two classes: solving $\mathcal{F}^{-1}(f)$ to estimate optical absorption coefficients or solving $(\mathcal{F}\mathcal{L})^{-1}(f)$ to estimate chromophore concentrations or, more often than not, blood oxygen saturation sO_2. The primary task of the networks in these cases is to account for the effect of the fluence, which is felt in two related ways: voxelwise it makes the PA spectra different from the absorption spectra (spectral coloring) and spatially the PAT image is no longer linearly related to the absorption coefficient distribution. These are related because the

absorption coefficient at one voxel can affect the PAT image at another through the fluence. Although this nonlocality of the operator $\mathcal{F}$ can be strong, e.g., a large absorber close to the light source may "shadow" a large part of the image region, for small absorbers the effect can be quite localized. The first application of machine learning to this problem[151] used "fluence contribution maps" that made this assumption. In the DL approaches discussed below the use of U-Net-type architectures is common, and it is known that their multiscale nature can help mitigate the spatial-invariance implicit in CNNs (see Sec. 5.3.1).

5.6.1 *U-Net-based optical inversions*

Cai et al.,[152] in an early contribution, used a variation on the U-Net, named the ResU-Net, to obtain estimates of sO_2 and a contrast agent from 2D multiwavelength PAT images. In this architecture, all the convolutional stages of a standard U-Net are replaced by residual blocks.[85] In a similar approach, Yang et al.[153] proposed another U-Net variant, DR2U-Net, the principal difference being that the residual blocks contain recurrent loops. Both these networks were shown to outperform linear unmixing $\mathcal{L}^{-1}f$—which ignores the effect of the fluence—in simple *in silico* tests.

Chen et al.[154] trained a U-Net to recover a 2D optical absorption coefficient distribution from a single-wavelength 2D PAT image. The loss term included a TV regularizer. The network was initially trained and tested with simple simulated examples and then demonstrated on 2D experimentally measured data. The measured training set was augmented by rotating the images in steps of 1 deg. The one result shown is promising, but the geometric simplicity and similarity of the training and test cases means the general applicability of the network remains unclear.

Exploiting the fact the U-Net was designed for segmentation of biomedical images,[80] Luke et al.[155] combined two U-Nets, one for segmentation and one for estimating blood sO_2, into a single "O-Net" with common input and output layers. The network input consists of two 2D slices from two 3D images obtained at different wavelengths, and the output is two 2D images: a segmentation and a map of sO_2. The network gives promising results on simulated data—it is shown to work better than linear unmixing—but the digital phantoms are simple geometric shapes. To overcome this concern, Bench et al.[156] performed a similar inversion but using 3D multiwavelength training images generated from vessel-like phantoms within a multilayered tissue. These images also contained limited-view artifacts from the acoustic reconstruction, and therefore, incorporated many aspects that would be present in real *in vivo* data. In these simulations, the vessel sO_2 estimates were accurate to within 1% on average, with a standard deviation of 6.3%.

Yang et al.[157] also used more realistic simulated data based on a 3D digital breast phantom, using a 3D light model, and acoustically processing 2D slices as input to the network to mimic the limited-view measurements made by a linear array transducer. Their network architecture, called an EDA-Net, uses the idea of "iterative deep aggregation"[158] to enhance the basic U-Net. In this architecture, every skip-connection is replaced with multiple nodes at the same scale, each of which is fed from below by (nonlinear) upsampling. This network was shown to perform slightly better than ResU-Net and U-Net++[159] and much better than linear unmixing.

In a detailed study, Gröhl et al.[126] used U-Nets to estimate the absorption coefficient in various ways. In two fluence-estimation approaches, asymmetric and symmetric U-Nets were used to estimate the fluence map ϕ from time series data g and from initial pressure distribution f, respectively. (This was subsequently divided out of f to estimate μ_a.) Also a one-step approach was described in which an asymmetric U-Net was used to estimate μ_a directly from limited-view and limited bandwidth time series data, i.e., solving $(\mathcal{A}\mathcal{F})^{-1}g$ directly. This one-step approach fared worse than the fluence estimation approaches in the *in silico* tests, but the comparison is perhaps unfair. Unlike the fluence estimation approaches, which just have to learn a mapping from one image space X_f to another X_{μ_a}, this inversion requires the network to also learn the mapping from Y to X_f from incomplete data.

5.6.2 *Learned uncertainty estimation*

All the U-Net variants in Sec. 5.6.1 already mentioned have been shown to give a degree of accuracy when demonstrated on simulated data (some more realistic than others), that if repeatable with *in vivo* data would be useful in applications. Moving to *in vivo* data, however, is a challenge, as discussed as follows in Sec. 5.6.4. One of the difficulties with translating sO_2 estimation techniques, for example, to a clinical setting is knowing how much confidence one should have in the estimates. This problem is tackled by Gröhl et al.,[126] who trained a U-Net to act as an error-estimating network, using {(PAT image, error image)} pairs, to give an estimate of the uncertainty in the μ_a estimates. The uncertainty correlated well with the actual error in the images in this *in silico* study. The use of a meta-network to observe the performance of a given estimator and output confidence levels for its estimates is very interesting given the difficulties inherent to translating quantitative PAT algorithms to *in vivo* cases.

5.6.3 *Learned spectral unmixing*

In contrast to the U-Net-based approaches discussed above, which exploit the spatial information about the fluence that is present in the PA images, pixelwise approaches attempt to solve the optical inversion using the spectral data alone.

Durairaj et al.[160] proposed a two-stage autoencoder architecture (Sec. 4.2.2) to estimate chromophore concentrations and molar absorption spectra simultaneously. One potentially significant advantage of this approach is that autoencoder networks by their nature do not require ground-truth data for the training. As discussed in Sec. 4.2.2, by having a smaller hidden layer than input and output layers, autoencoders aim to find a compressed representation of the input. Durairaj et al. chose the hidden layer to have as many dimensions as there are chromophores contributing to the data, in the hope that the values at the hidden layer are estimates of the chromophore concentrations (endmember abundances in their terminology) and the network weights are estimates of the molar absorption spectra (endmember spectra). Because this approach aims to solve the ill-posed problem of finding both the concentrations and spectra simultaneously, it requires strong prior information. As well as a positivity condition, which is well-justified, they impose the condition that the chromophore concentrations sum to one. However, this is unrealistic, as there will also be nonabsorbing molecules present in real tissue. Furthermore, it is not clear how this approach can account for the effect of the fluence on the PAT images and therefore unclear the extent to which the approach outlined in this preliminary simulation study will be useful in practice.

A different approach to learned spectral unmixing was taken by Gröhl et al.,[161] who used a fully connected network with 8 hidden layers to convert pixelwise PAT spectra into estimates of sO_2. The training data were taken from 2D simulated PAT images of vessels, and when the network was tested with simulated data it gave promising results. With some bravado, this network was then tested *in vivo* on images of a porcine brain and human forearm, and in the case of the pig brain "seems to provide physiologically more plausible estimations" than linear unmixing.

5.6.4 *Training data*

As a concluding remark on this section, we note that several classical approaches to quantitative PAT have been demonstrated over the past decade (see Refs. 3, 69, 162, and 163 and their references and citations) but it has proved difficult to translate these methods to work convincingly with measurement data obtained *in vivo*, largely due to the challenge of obtaining all the auxiliary input parameters with sufficient accuracy under experimental conditions. DL holds the promise of overcoming this problem by learning the model, thereby not requiring auxiliary inputs, but a new difficulty arises: obtaining a large collection of experimentally measured *in vivo* data with a known ground truth to use for the training. As discussed in Sec. 4.4, there are two approaches: simulating the data or reconstructing ground-truth images using a "gold standard" classical reconstruction technique. The papers discussed in this section have used the former approach of simulating the data, typically using an MC method such as MCX[164] for

modeling the light propagation and collocation method such as *k*-Wave for modeling the acoustic propagation.[92] The degree to which the simulations are realistic will determine how well a network trained with this data will work on data measured *in vivo*, and therefore, will determine the confidence with which any conclusions can be drawn from a study using such an approach. In conclusion then, the use of DL to tackle the quantitative PAT problem appears to hold promise but the translation to practical, *in vivo*, cases remains a significant challenge.

6 Conclusions and Future Directions

The diversity of the work that has been done on learned image reconstruction in PAT in just the last few years, and the increasing rate at which it is being produced, suggests that the field will continue to develop for some time. In particular, we notice that already a move has begun from straightforward proof-of-concept applications of DL to more sophisticated approaches. Nevertheless, there are many issues that remain to be addressed. For instance, on the one hand there are model agnostic reconstruction pipelines using fully learned approaches that get a lot of attention due to low latency. On the other hand, as described already, there are learned reconstructions that use a physical model in combination with a network, which have been shown to be more stable and require less training data but are (considerably) slower in providing a reconstruction. This is in part because accurate numerical models of the physics are often slow compared to networks. Therefore, a major question remains: *Is it possible to obtain network speed without sacrificing the stability and accuracy that comes from explicitly incorporating a model?*

Another challenge, which hangs over learned image reconstructions with all biomedical applications, is how to ensure oddities (like a tumor) appear accurately in the image even though nothing quite like them was in the training data. In other words, how do we ensure the distribution of the training data matches that of the imaged target? And if it does not, will there be problems, as suggested by results from the tutorial, Sec. 4.5? Could this problem be ameliorated by ensuring additional constraints, such as data consistency?

To conclude this review, we describe a few current research directions that address these questions, either by considering new training regimes or by combining physical models with neural networks in different ways.

6.1 *Data Consistency is Important*

Many approaches are still missing a data-consistency term and hence the reconstructions obtained might look realistic but there is no way to assess their correctness. As we have discussed, there are a few approaches that do consider such data consistency during the reconstruction and hence provide a possible direction for further developments, such as the null space approaches discussed in Sec. 5.1.2 or learned iterative reconstructions in Sec. 5.4. Another possible way to tackle this limitation is using networks that consider uncertainty or provide an uncertainty estimate on top of the reconstruction. First steps in this direction have been taken for PAT,[116] see also Sec. 5.6.2, but there is also rising interest in other fields to incorporate such uncertainty estimates into a learned reconstruction framework,[165–167] which could be taken as inspiration.

6.2 *Lack of* in vivo *Training Data*

For experimental scenarios, especially *in vivo*, using simulated training data is risky because it is hard to ensure the training set distribution matches that of the target.

As the majority of the algorithms discussed already used fully supervised training, these approaches are primarily limited by the available ground-truth data. As this is seldom a viable option when developing imaging pipelines for *in vivo* applications, it may be that different training regimes will be needed, such as semisupervised approaches, as discussed in Sec. 4.2.4. For instance, by including a data consistency in the transfer to experimental data (also known as cycle consistency) or discriminator (GAN) based approaches.[117]

Another possibility might be to consider the framework of *physics-informed neural networks*,[168] in which the physical model, given by a partial differential equation, is incorporated directly into the loss function. In this case, rather than the network needing to learn the whole physical operator from the data, as in the fully learned cases presented already, the network learns much of the physics by virtue of the terms in the loss function.

6.3 *3D Nature of PAT*

The high computational complexity caused by the inherently 3D nature of PAT is another challenge for learned approaches, as computational models tend to be time-consuming and simply storing the data requires large amounts of memory. Possible methods to overcome this have been discussed in some recent papers, for instance using invertible networks,[169,170] which do not require the storage of intermediate states in the network to compute the gradients for training. Another idea of how to scale learned iterative schemes to 3D is by computing the forward model on multiple lower resolutions in the reconstruction process.[171]

6.4 *Model Augmentation and Correction*

The learned schemes that use a model in conjunction with a network are typically slow, and also face the additional problem of uncertainty in model parameters, especially the sound speed and, for the optical inversion, the scattering (see Sec. 2.2.2).

There may be advantages, therefore, of considering different ways to incorporate some of the behavior of the model equation directly into the network. For instance, by designing or constraining networks based on the discretization of the forward model—similar work has already been done for diffusion equations.[15,172] This way, it is possible to explicitly embed the properties of the model into the network architecture, with a computationally more efficient (network-based) solver.

Another possibility is to use approximate models that are faster or easier to compute in place of the true (expensive) model, and train a network to learn a correction.[173,174] The error may arise from an efficient, but inaccurate, numerical discretization of the correct model[141,175] or because the accurate model has been replaced with a more-easily solvable approximation.[174] We believe that this direction could be particularly fruitful for PAT as model information is essential to provide stability and robustness in the inversion, but we need to overcome the two major limitations: computational speed and the inherent uncertainty in the model parameters. Nevertheless, these improvements come with a major increase in training times for such networks.

6.5 *Trade-Offs and Choices*

There are so many options that trade-offs and choices will need to be made in practice. This is not a problem *per se*, but rather an opportunity. There are many possible ways in which a network can be incorporated into the reconstruction pipeline, and the approach that will be best suited to a particular application will depend on the nature of the application. It is the responsibility of the designer of the image reconstruction algorithm to consider the trade-offs and constraints, e.g., is reconstruction speed or a data-consistency guarantee more important? Does the algorithm need to be able to work well with more than one hardware system? What hardware is available for the computations?—and construct the algorithm accordingly. This plethora of choice is good, because it gives sufficient flexibility for properly crafted, well-thought-through algorithms to be designed to be optimal for specific tasks. The key to realizing that is developing an understanding of the strengths and weaknesses of particular architectures and approaches. We are only at the beginning of this journey, but we hope this paper has illuminated at least a little of the way along the path.

Disclosures

No conflicts of interest, financial or otherwise, are declared by the authors.

Acknowledgments

The authors would like to express their thanks to Paul Beard and UCL's Photoacoustic Imaging Group for many helpful discussions on all aspects of PAT over many years and also to Simon Arridge, Jonas Adler, Sebastian Lunz, Felix Lucka, Marta Betcke, Bradley Treeby, Antonio Stanziola, Ashkan Javaherian, Ciaran Bench, and UCL's Biomedical Ultrasound Group for very useful and informative discussions on inverse problems, image reconstruction, deep learning, and acoustic modeling. This work was partly funded by the European Union's Horizon 2020 Research and Innovation Program H2020 ICT 2016-2017 under Grant Agreement No. 732411, which is an initiative of the Photonics Public Private Partnership, and partly by the Academy of Finland Project 312123 (Finnish Centre of Excellence in Inverse Modeling and Imaging, 2018–2025) and the CMIC-EPSRC platform grant (No. EP/M020533/1).

Code, Data, and Materials Availability

Codes and training/test data for all experiments discussed will be made available at: https://github.com/asHauptmann/PAT_CODES.

References

1. P. Kuchment and L. Kunyansky, "Mathematics of Photoacoustic and Thermoacoustic Tomography," *Handbook of Mathematical Methods in Imaging*, O. Scherzer, Ed., pp. 817–865, Springer, New York (2015).
2. J. Poudel, Y. Lou, and M. A. Anastasio, "A survey of computational frameworks for solving the acoustic inverse problem in three-dimensional photoacoustic computed tomography," *Phys. Med. Biol.* **64**(14), 14TR01 (2019).
3. B. T. Cox et al., "Quantitative spectroscopic photoacoustic imaging: a review," *J. Biomed. Opt.* **17**(6), 061202 (2012).
4. C. Huang et al., "Full-wave iterative image reconstruction in photoacoustic tomography with acoustically inhomogeneous media," *IEEE Trans. Med. Imaging* **32**(6), 1097–1110 (2013).
5. S. Arridge et al., "Accelerated high-resolution photoacoustic tomography via compressed sensing," *Phys. Med. Biol.* **61**(24), 8908–8940 (2016).
6. Y. E. Boink et al., "A framework for directional and higher-order reconstruction in photoacoustic tomography," *Phys. Med. Biol.* **63**(4), 045018 (2018).
7. E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose x-ray CT reconstruction," *Med. Phys.* **44**(10), e360–e375 (2017).
8. K. H. Jin et al., "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.* **26**(9), 4509–4522 (2017).
9. K. Hammernik et al., "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.* **79**(6), 3055–3071 (2018).
10. J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Prob.* **33**(12), 124007 (2017).
11. B. Zhu et al., "Image reconstruction by domain-transform manifold learning," *Nature* **555**(7697), 487 (2018).
12. S. Arridge et al., "Solving inverse problems using data-driven models," *Acta Numer.* **28**, 1–174 (2019).
13. J. C. Ye, Y. Han, and E. Cha, "Deep convolutional framelets: a general deep learning framework for inverse problems," *SIAM J. Imaging Sci.* **11**(2), 991–1048 (2018).
14. E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Prob.* **34**(1), 014004 (2017).
15. L. Ruthotto and E. Haber, "Deep neural networks motivated by partial differential equations," *J. Math. Imaging Vision* **62**(3), 352–364 (2020).
16. J. Schlemper et al., "A deep cascade of convolutional neural networks for MR image reconstruction," *Lect. Notes Comput. Sci.* **10265**, 647–658 (2017).

17. A. Hauptmann et al., "Real-time cardiovascular MR with spatio-temporal artifact suppression using deep learning–proof of concept in congenital heart disease," *Magn. Reson. Med.* **81**(2), 1143–1156 (2018).
18. J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imaging* **37**(6), 1322–1332 (2018).
19. S. R. Arridge, "Optical tomography in medical imaging," *Inverse Prob.* **15**(2), R41–R93 (1999).
20. A. J. Welch et al., *Optical-Thermal Response of Laser-Irradiated Tissue*, Vol. 2, Springer, New York (2011).
21. I. J. Bigio and S. Fantini, *Quantitative Biomedical Optics: Theory, Methods, and Applications*, Cambridge University Press, Cambridge (2016).
22. M. Li, Y. Tang, and J. Yao, "Photoacoustic tomography of blood oxygenation: a mini review," *Photoacoustics* **10**, 65–73 (2018).
23. K. P. Köstli et al., "Temporal backward projection of optoacoustic pressure transients using Fourier transform methods," *Phys. Med. Biol.* **46**(7), 1863–1872 (2001).
24. Y. Xu, M. Xu, and L. V. Wang, "Exact frequency-domain reconstruction for thermoacoustic tomography. II. Cylindrical geometry," *IEEE Trans. Med. Imaging* **21**(7), 829–833 (2002).
25. D. Finch and S. K. Patch, "Determining a function from its mean values over a family of spheres," *SIAM J. Math. Anal.* **35**(5), 1213–1240 (2004).
26. M. Haltmeier, "Exact reconstruction formula for the spherical mean radon transform on ellipsoids," *Inverse Prob.* **30**(10), 105006 (2014).
27. L. Kunyansky, "Reconstruction of a function from its spherical (circular) means with the centers lying on the surface of certain polygons and polyhedra," *Inverse Prob.* **27**(2), 025012 (2011).
28. P. Burgholzer et al., "Thermoacoustic tomography with integrating area and line detectors," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **52**(9), 1577–1583 (2005).
29. N. Huynh et al., "Single-pixel camera photoacoustic tomography," *J. Biomed. Opt.* **24**(12), 121907 (2019).
30. X. Wang et al., "Noninvasive laser-induced photoacoustic tomography for structural and functional in vivo imaging of the brain," *Nat. Biotechnol.* **21**(7), 803–806 (2003).
31. B. Yin et al., "Fast photoacoustic imaging system based on 320-element linear transducer array," *Phys. Med. Biol.* **49**(7), 1339 (2004).
32. L. Landau and E. Lifshitz, *Fluid Mechanics*, Vol. 6, 2nd ed., Butterworth-Heineman, Oxford (1987).
33. Y. Xu et al., "Reconstructions in limited-view thermoacoustic tomography," *Med. Phys.* **31**(4), 724–733 (2004).
34. B. T. Cox, S. R. Arridge, and P. C. Beard, "Estimating chromophore distributions from multiwavelength photoacoustic images," *J. Opt. Soc. Am. A* **26**(2), 443–455 (2009).
35. C. Huang et al., "Joint reconstruction of absorbed optical energy density and sound speed distributions in photoacoustic computed tomography: a numerical investigation," *IEEE Trans. Comput. Imaging* **2**(2), 136–149 (2016).
36. T. Tarvainen et al., "Reconstructing absorption and scattering distributions in quantitative photoacoustic tomography," *Inverse Prob.* **28**(8), 084009 (2012).
37. P. Stefanov and G. Uhlmann, "Instability of the linearized problem in multiwave tomography of recovery both the source and the speed," *Inverse Prob. Imaging* **7**(4), 1367 (2013).
38. J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Vol. 160, Applied Mathematical Sciences, Springer Verlag (2004).
39. J. Kaipio and E. Somersalo, "Statistical inverse problems: discretization, model reduction and inverse crimes," *J. Comput. Appl. Math.* **198**(2), 493–504 (2007).
40. S. R. Arridge et al., "Approximation errors and model reduction with an application in optical diffusion tomography," *Inverse Prob.* **22**(1), 175–195 (2006).
41. T. Sahlström et al., "Modeling of errors due to uncertainties in ultrasound sensor locations in photoacoustic tomography," *IEEE Trans. Med. Imaging* **39**, 2140–2150 (2020).
42. T. Tarvainen et al., "Bayesian image reconstruction in quantitative photoacoustic tomography," *IEEE Trans. Med. Imaging* **32**(12), 2287–2298 (2013).

43. F. Natterer, *The Mathematics of Computerized Tomography*, Vol. 32, John Wiley & Sons, Chichester; B. G. Teubner, Stuttgart, Germany (1986).
44. S. Arridge et al., "On the adjoint operator in photoacoustic tomography," *Inverse Prob.* **32**(11), 115012 (2016).
45. M. Xu and L. V. Wang, "Universal back-projection algorithm for photoacoustic computed tomography," *Phys. Rev. E* **71**(1), 016706 (2005).
46. M. Haltmeier and S. Pereverzyev Jr., "The universal back-projection formula for spherical means and the wave equation on certain quadric hypersurfaces," *J. Math. Anal. Appl.* **429**(1), 366–382 (2015).
47. P. Burgholzer et al., "Temporal back-projection algorithms for photoacoustic tomography with integrating line detectors," *Inverse Prob.* **23**(6), S65 (2007).
48. S. Park et al., "Adaptive beamforming for photoacoustic imaging," *Opt. Lett.* **33**(12), 1291–1293 (2008).
49. M. Mozaffarzadeh et al., "Double-stage delay multiply and sum beamforming algorithm: application to linear-array photoacoustic imaging," *IEEE Trans. Biomed. Eng.* **65**(1), 31–42 (2018).
50. Y. Xu, D. Feng, and L. V. Wang, "Exact frequency-domain reconstruction for thermoacoustic tomography. I. Planar geometry," *IEEE Trans. Med. Imaging* **21**(7), 823–828 (2002).
51. L. A. Kunyansky, "A series solution and a fast algorithm for the inversion of the spherical mean radon transform," *Inverse Prob.* **23**(6), S11 (2007).
52. L. Kunyansky, "Fast reconstruction algorithms for the thermoacoustic tomography in certain domains with cylindrical or spherical symmetries," *Inverse Prob. Imaging* **6**(1), 111–131 (2012).
53. Y. Xu and L. V. Wang, "Time reversal and its application to tomography with diffracting sources," *Phys. Rev. Lett.* **92**(3), 033902 (2004).
54. P. Burgholzer et al., "Exact and approximative imaging methods for photoacoustic tomography using an arbitrary detection surface," *Phys. Rev. E* **75**(4), 046706 (2007).
55. Y. Hristova, P. Kuchment, and L. Nguyen, "Reconstruction and time reversal in thermoacoustic tomography in acoustically homogeneous and inhomogeneous media," *Inverse Prob.* **24**(5), 055006 (2008).
56. B. Baker and E. Copson, *The Mathematical Theory of Huygens' Principle*, Vol. 329, American Mathematical Society, Rhode Island (2003).
57. B. T. Cox and B. E. Treeby, "Artifact trapping during time reversal photoacoustic imaging for acoustically heterogeneous media," *IEEE Trans. Med. Imaging* **29**(2), 387–396 (2010).
58. P. Stefanov and G. Uhlmann, "Thermoacoustic tomography with variable sound speed," *Inverse Prob.* **25**(7), 075011 (2009).
59. Z. Guo et al., "Compressed sensing in photoacoustic tomography in vivo," *J. Biomed. Opt.* **15**(2), 021311 (2010).
60. K. Wang et al., "Investigation of iterative image reconstruction in three-dimensional optoacoustic tomography," *Phys. Med. Biol.* **57**(17), 5399 (2012).
61. M. Haltmeier and L. V. Nguyen, "Analysis of iterative methods in photoacoustic tomography with variable sound speed," *SIAM J. Imaging Sci.* **10**(2), 751–781 (2017).
62. A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numer.* **25**, 161–319 (2016).
63. M. Benning and M. Burger, "Modern regularization methods for inverse problems," *Acta Numer.* **27**, 1 (2018).
64. A. Rosenthal, D. Razansky, and V. Ntziachristos, "Fast semi-analytical model-based acoustic inversion for quantitative optoacoustic tomography," *IEEE Trans. Med. Imaging* **29**(6), 1275–1285 (2010).
65. G. Paltauf et al., "Iterative reconstruction algorithm for optoacoustic imaging," *J. Acoust. Soc. Am.* **112**(4), 1536–1544 (2002).
66. A. Q. Bauer et al., "Quantitative photoacoustic imaging: correcting for heterogeneous light fluence distributions using diffuse optical tomography," *J. Biomed. Opt.* **16**(9), 096016 (2011).
67. A. Hussain et al., "Photoacoustic and acousto-optic tomography for quantitative and functional imaging," *Optica* **5**(12), 1579–1589 (2018).

68. B. T. Cox et al., "Two-dimensional quantitative photoacoustic image reconstruction of absorption distributions in scattering media by use of a simple iterative method," *Appl. Opt.* **45**(8), 1866–1875 (2006).
69. J. Buchmann et al., "Three-dimensional quantitative photoacoustic tomography using an adjoint radiance Monte Carlo model and gradient descent," *J. Biomed. Opt.* **24**(6), 066001 (2019).
70. M. Abadi et al., "TensorFlow: large-scale machine learning on heterogeneous systems," 2015, tensorflow.org
71. A. Paszke et al., "Pytorch: an imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. 32*, Curran Associates, Inc., pp. 8024–8035 (2019).
72. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015).
73. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge (2016).
74. J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks* **61**, 85–117 (2015).
75. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature* **323**(6088), 533–536 (1986).
76. Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.* **1**(4), 541–551 (1989).
77. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *3rd Int. Conf. Learn. Represent.* (2015).
78. J.-Y. Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 2223–2232 (2017).
79. S. Antholzer, M. Haltmeier, and J. Schwab, "Deep learning for photoacoustic tomography from sparse data," *Inverse Prob. Sci. Eng.* **27**(7), 987–1005 (2019).
80. O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
81. I. Daubechies, *Ten Lectures on Wavelets*, Vol. 61, SIAM, Philadelphia (1992).
82. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(7), 674–693 (1989).
83. A. Hauptmann et al., "Model-based learning for accelerated, limited-view 3D photoacoustic tomography," *IEEE Trans. Med. Imaging* **37**(6), 1382–1393 (2018).
84. P. Putzky and M. Welling, "Recurrent inference machines for solving inverse problems," arXiv:1706.04008 (2017).
85. K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).
86. A. K. Maier et al., "Learning with known operators reduces maximum error bounds," *Nat. Mach. Intell.* **1**(8), 373–380 (2019).
87. ELCAP Public Lung Image Database, http://www.via.cornell.edu/lungdb.html (accessed 5 October 2020).
88. DRIVE: Digital Retinal Images for Vessel Extraction, https://drive.grand-challenge.org/ (accessed 5 October 2020).
89. M. Scianna, C. Bell, and L. Preziosi, "A review of mathematical models for the formation of vascular networks," *J. Theor. Biol.* **333**, 174–209 (2013).
90. D. Erhan et al., "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.* **11**, 625–660 (2010).
91. J. Yosinski et al., "How transferable are features in deep neural networks?" in *Adv. Neural Inf. Process. Syst.*, pp. 3320–3328 (2014).
92. B. E. Treeby and B. T. Cox, "k-wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *J. Biomed. Opt.* **15**(2), 021314 (2010).
93. TensorBoard, https://www.tensorflow.org/tensorboard (accessed 5 October 2020).
94. D. O. Baguer, J. Leuschner, and M. Schmidt, "Computed tomography reconstruction using deep image prior and learned reconstruction methods," *Inverse Prob.* **36**(9), 094004 (2020).
95. Y. An et al., "Application of machine learning method in optical molecular imaging: a review," *Sci. China Inf. Sci.* **63**(1), 111101 (2020).

96. L. Zhang et al., "Brief review on learning-based methods for optical tomography," *J. Innovative Opt. Health Sci.* **12**(6), 1930011 (2019).
97. K. Sivasubramanian and L. Xing, "Deep learning for image processing and reconstruction to enhance led-based photoacoustic imaging," in *LED-Based Photoacoustic Imaging*, M. K. Ajith Singh, Ed., pp. 203–241, Springer, Singapore (2020).
98. S. Antholzer, J. Schwab, and M. Haltmeier, "Deep learning versus ℓ^1-minimization for compressed sensing photoacoustic tomography," in *IEEE Int. Ultrason. Symp.*, IEEE, pp. 206–212 (2018).
99. S. Antholzer et al., "Photoacoustic image reconstruction via deep learning," *Proc. SPIE* **10494**, 104944U (2018).
100. S. Guan et al., "Fully dense UNet for 2D sparse photoacoustic tomography artifact removal," *IEEE J. Biomed. Health. Inf.* **24**(2), 568–576 (2020).
101. H. Deng et al., "Machine-learning enhanced photoacoustic computed tomography in a limited view configuration," *Proc. SPIE* **11186**, 111860J (2019).
102. H. Shan, G. Wang, and Y. Yang, "Accelerated correction of reflection artifacts by deep neural networks in photo-acoustic tomography," *Appl. Sci.* **9**(13), 2615 (2019).
103. N. Awasthi et al., "Pa-fuse: deep supervised approach for the fusion of photoacoustic images with distinct reconstruction characteristics," *Biomed. Opt. Express* **10**(5), 2227–2243 (2019).
104. D. Waibel et al., "Reconstruction of initial pressure from limited view photoacoustic images using deep learning," *Proc. SPIE* **10494**, 104942S (2018).
105. N. Davoudi, X. L. Deán-Ben, and D. Razansky, "Deep learning optoacoustic tomography with sparse data," *Nat. Mach. Intell.* **1**(10), 453–460 (2019).
106. A. Hariri et al., "Deep learning improves contrast in low-fluence photoacoustic imaging," *Biomed. Opt. Express* **11**(6), 3360–3373 (2020).
107. P. Farnia et al., "High-quality photoacoustic image reconstruction based on deep convolutional neural network: towards intra-operative photoacoustic imaging," *Biomed. Phys. Eng. Express* **6**(4), 045019 (2020).
108. H. Zhang et al., "A new deep learning network for mitigating limited-view and undersampling artifacts in ring-shaped photoacoustic tomography," *Comput. Med. Imaging Graphics* **84**, 101720 (2020).
109. J. Schwab et al., "Real-time photoacoustic projection imaging using deep learning," arXiv:1801.06693 (2018).
110. J. Schwab, S. Antholzer, and M. Haltmeier, "Learned backprojection for sparse and limited view photoacoustic tomography," *Proc. SPIE* **10878**, 1087837 (2019).
111. E. M. A. Anas et al., "Enabling fast and high quality led photoacoustic imaging: a recurrent neural networks based approach," *Biomed. Opt. Express* **9**(8), 3852–3866 (2018).
112. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.* **9**(8), 1735–1780 (1997).
113. S. Xingjian et al., "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *Adv. Neural Inf. Process. Syst.*, pp. 802–810 (2015).
114. M. K. A. Singh et al., "Deep learning-enhanced led-based photoacoustic imaging," *Proc. SPIE* **11240**, 1124038 (2020).
115. M. W. Kim et al., "Deep-learning image reconstruction for real-time photoacoustic system," *IEEE Trans. Med. Imaging* (2020).
116. G. Godefroy, B. Arnal, and E. Bossy, "Solving the visibility problem in photoacoustic imaging with a deep learning approach providing prediction uncertainties," arXiv:2006.13096 (2020).
117. T. Vu et al., "A generative adversarial network for artifact removal in photoacoustic computed tomography with a linear-array transducer," *Exp. Biol. Med.* **245**(7), 597–605 (2020).
118. D. Allman, A. Reiter, and M. A. L. Bell, "Photoacoustic source detection and reflection artifact removal enabled by deep learning," *IEEE Trans. Med. Imaging* **37**(6), 1464–1477 (2018).
119. D. Allman, A. Reiter, and M. Bell, "Exploring the effects of transducer models when training convolutional neural networks to eliminate reflection artifacts in experimental photoacoustic images," *Proc. SPIE* **10494**, 104945H (2018).

120. D. Allman, A. Reiter, and M. A. L. Bell, "A machine learning method to identify and remove reflection artifacts in photoacoustic channel data," in *IEEE Int. Ultrason. Symp.*, IEEE, pp. 1–4 (2017).
121. S. Ren et al., "Faster R-CNN: towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst.*, pp. 91–99 (2015).
122. D. Allman et al., "Deep neural networks to remove photoacoustic reflection artifacts in ex vivo and in vivo tissue," in *IEEE Int. Ultrason. Symp.*, IEEE, pp. 1–4 (2018).
123. N. Awasthi et al., "Sinogram super-resolution and denoising convolutional neural network (SRCN) for limited data photoacoustic tomography," arXiv:2001.06434 (2020).
124. N. Awasthi et al., "Deep neural network based sinogram super-resolution and bandwidth enhancement for limited-data photoacoustic tomography," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* (2020).
125. R. Shang, K. Hoffer-Hawlik, and G. P. Luke, "A two-step-training deep learning framework for real-time computational imaging without physics priors," arXiv:2001.03493 (2020).
126. J. Gröhl et al., "Confidence estimation for machine learning-based quantitative photoacoustics," *J. Imaging* **4**(12), 147 (2018).
127. H. Lan et al., "Deep learning approach to reconstruct the photoacoustic image using multi-frequency data," in *IEEE Int. Ultrason. Symp.*, IEEE, pp. 487–489 (2019).
128. H. Lan et al., "Reconstruct the photoacoustic image based on deep learning with multi-frequency ring-shape transducer array," in *41st Annu. Int. Conf. IEEE Eng. Med. and Biol. Soc.*, IEEE, pp. 7115–7118 (2019).
129. S. Guan et al., "Limited-view and sparse photoacoustic tomography for neuroimaging with deep learning," *Sci. Rep.* **10**(1), 8510 (2020).
130. T. Tong et al., "Domain transform network for photoacoustic tomography from limited-view and sparsely sampled data," *Photoacoustics* **19**, 100190 (2020).
131. H. Lan et al., "Real-time photoacoustic tomography system via single data acquisition channel," arXiv:2001.07454 (2020).
132. A. Reiter and M. A. L. Bell, "A machine learning approach to identifying point source locations in photoacoustic data," *Proc. SPIE* **10064**, 100643J (2017).
133. K. Johnstonbaugh et al., "Novel deep learning architecture for optical fluence dependent photoacoustic target localization," *Proc. SPIE* **10878**, 108781L (2019).
134. K. Johnstonbaugh et al., "A deep learning approach to photoacoustic wavefront localization in deep-tissue medium," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* (2020).
135. H. Shan, G. Wang, and Y. Yang, "Simultaneous reconstruction of the initial pressure and sound speed in photoacoustic tomography using a deep-learning approach," *Proc. SPIE* **11105**, 1110504 (2019).
136. T. P. Matthews et al., "Parameterized joint reconstruction of the initial pressure and sound speed distributions for photoacoustic computed tomography," *SIAM J. Imaging Sci.* **11**(2), 1560–1588 (2018).
137. Y. E. Boink et al., "Sensitivity of a partially learned model-based reconstruction algorithm," *PAMM* **18**(1), e201800222 (2018).
138. Y. E. Boink, S. Manohar, and C. Brune, "A partially-learned algorithm for joint photoacoustic reconstruction and segmentation," *IEEE Trans. Med. Imaging* **39**(1), 129–139 (2020).
139. Y. E. Boink, C. Brune, and S. Manohar, "Robustness of a partially learned photoacoustic reconstruction algorithm," *Proc. SPIE* **10878**, 108781D (2019).
140. A. Chambolle, S. E. Levine, and B. J. Lucier, "An upwind finite-difference method for total variation-based image smoothing," *SIAM J. Imaging Sci.* **4**(1), 277–299 (2011).
141. A. Hauptmann et al., "Approximate k-space models and deep learning for fast photoacoustic reconstruction," *Lect. Notes Comput. Sci.* **11074**, 103–111 (2018).
142. B. T. Cox and P. C. Beard, "Fast calculation of pulsed photoacoustic fields in fluids using k-space methods," *J. Acoust. Soc. Am.* **117**(6), 3616–3627 (2005).
143. C. Yang, H. Lan, and F. Gao, "Accelerated photoacoustic tomography reconstruction via recurrent inference machines," in *41st Annu. Int. Conf. IEEE Eng. Med. and Biol. Soc.*, IEEE, pp. 6371–6374 (2019).

144. K. Lønning et al., "Recurrent inference machines for accelerated MRI reconstruction," in *Int. Conf. Med. Imaging Deep Learn.* (2018).
145. H. Lan et al., "Hybrid neural network for photoacoustic imaging reconstruction," in *41st Annu. Int. Conf. IEEE Eng. Med. and Biol. Soc.*, IEEE, pp. 6367–6370 (2019).
146. H. Lan et al., "KI-GAN: knowledge infusion generative adversarial network for photoacoustic image reconstruction in vivo," *Lect. Notes Comput. Sci.* **11764**, 273–281 (2019).
147. H. Lan et al., "Y-net: a hybrid deep learning reconstruction framework for photoacoustic imaging in vivo," arXiv:1908.00975 (2019).
148. H. Li et al., "Nett: solving inverse problems with deep neural networks," *Inverse Prob.* **36**(6), 065005 (2020).
149. S. Antholzer et al., "Nett regularization for compressed sensing photoacoustic tomography," *Proc. SPIE* **10878**, 108783B (2019).
150. J. Schwab et al., "Deep learning of truncated singular values for limited view photoacoustic tomography," *Proc. SPIE* **10878**, 1087836 (2019).
151. T. Kirchner, J. Gröhl, and L. Maier-Hein, "Context encoding enables machine learning-based quantitative photoacoustics," *J. Biomed. Opt.* **23**(5), 056008 (2018).
152. C. Cai et al., "End-to-end deep neural network for optical inversion in quantitative photoacoustic imaging," *Opt. Lett.* **43**(12), 2752–2755 (2018).
153. C. Yang et al., "Quantitative photoacoustic blood oxygenation imaging using deep residual and recurrent neural network," in *IEEE 16th Int. Symp. Biomed. Imaging*, IEEE, pp. 741–744 (2019).
154. T. Chen et al., "A deep learning method based on U-Net for quantitative photoacoustic imaging," *Proc. SPIE* **11240**, 112403V (2020).
155. G. P. Luke et al., "O-net: a convolutional neural network for quantitative photoacoustic image segmentation and oximetry," arXiv:1911.01935 (2019).
156. C. Bench, A. Hauptmann, and B. Cox, "Toward accurate quantitative photoacoustic imaging: learning vascular blood oxygen saturation in three dimensions," *J. Biomed. Opt.* **25**(8), 085003 (2020).
157. C. Yang and F. Gao, "EDA-Net: dense aggregation of deep and shallow information achieves quantitative photoacoustic blood oxygenation imaging deep in human breast," *Lect. Notes Comput. Sci.* **11764**, 246–254 (2019).
158. F. Yu et al., "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2403–2412 (2018).
159. Z. Zhou et al., "Unet++: a nested U-Net architecture for medical image segmentation," *Lect. Notes Comput. Sci.* **11045**, 3–11 (2018).
160. D. Durairaj et al., "Unsupervised deep learning approach for photoacoustic spectral unmixing," *Proc. SPIE* **11240**, 112403H (2020).
161. J. Gröhl et al., "Estimation of blood oxygenation with learned spectral decoloring for quantitative photoacoustic imaging (LSD-qPAI)," arXiv:1902.05839 (2019).
162. B. T. Cox, T. Tarvainen, and S. R. Arridge, "Multiple illumination quantitative photoacoustic tomography using transport and diffusion models," *Contemp. Math.* **559**, 1–12 (2011).
163. W. C. Vogt et al., "Photoacoustic oximetry imaging performance evaluation using dynamic blood flow phantoms with tunable oxygen saturation," *Biomed. Opt. Express* **10**(2), 449–464 (2019).
164. Q. Fang and D. A. Boas, "Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units," *Opt. Express* **17**(22), 20178–20190 (2009).
165. J. Adler and O. Öktem, "Deep posterior sampling: uncertainty quantification for large scale inverse problems," in *Int. Conf. Med. Imaging Deep Learn.* (2019).
166. J. Schlemper et al., "Bayesian deep learning for accelerated MR image reconstruction," *Lect. Notes Comput. Sci.* **11074**, 64–71 (2018).
167. A. Denker et al., "Conditional normalizing flows for low-dose computed tomography image reconstruction," arXiv:2006.06270 (2020).
168. 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," *J. Comput. Phys.* **378**, 686–707 (2019).

169. C. Etmann, R. Ke, and C.-B. Schönlieb, "iUNets: learnable invertible up- and down-sampling for large-scale inverse problems," in *IEEE 30th Int. Workshop Mach. Learn. Signal Process.*, pp. 1–6, Espoo, Finland (2020).
170. P. Putzky and M. Welling, "Invert to learn to invert," in *Adv. Neural Inf. Process. Syst.*, pp. 444–454 (2019).
171. A. Hauptmann et al., "Multi-scale learned iterative reconstruction," *IEEE Trans. Comput. Imaging* (2020).
172. S. Arridge and A. Hauptmann, "Networks for nonlinear diffusion problems in imaging," *J. Math. Imaging Vision* **62**(3), 471–487 (2020).
173. S. Lunz et al., "On learned operator correction in inverse problems," arXiv:2005.07069 (2020).
174. D. Smyl et al., "Learning and correcting non-Gaussian model errors," arXiv:2005.14592 (2020).
175. A. Siahkoohi, M. Louboutin, and F. J. Herrmann, "Neural network augmented wave-equation simulation," arXiv:1910.00925 (2019).

Biographies of the authors are not available.