Computing coherent light scattering on the millimetre scale using a recurrent neural network without training

Laurynas Valantinas & Tom Vettenburg
School of Science and Engineering, University of Dundee, DD1 4HN, UK
e-mail: t.vettenburg@dundee.ac.uk

Heterogeneous materials such as biological tissue scatter light in random, yet deterministic, ways. Wavefront shaping can reverse the effects of scattering to enable deep-tissue microscopy. Such methods require either invasive access to the internal field or the computational solving of an inverse problem. However, calculating the coherent field on a scale relevant to microscopy remains excessively demanding for consumer hardware. Here we show how a recurrent neural network can mirror Maxwell’s equations without training. By harnessing public machine learning infrastructure, the light-field throughout a 6mm² area or 110³μm³ volume can be calculated in 16 minutes. The elimination of the training phase cuts the calculation time and, importantly, it ensures a fully deterministic solution, free of training bias. We integrated our method with an open-source electromagnetic solver. This enables any researcher with an internet connection to calculate complex light-scattering in volumes that are larger than by two orders of magnitude.

Machine learning has pushed automation into areas that were formerly considered the exclusive remit of humans. Recently it was demonstrated that artificial neural networks can also help us circumvent the curse of dimensionality in traditionally hard scientific computations. Problems that scale poorly and were once thought to be intractable, are now within reach of industrial, yet publicly available, machine learning infrastructure.

Adances in optics and photonics increasingly rely on our ability to accurately compute how light propagates and scatters as described by Maxwell’s Equations. Coherent calculations provide essential information that can be impractical or impossible to obtain experimentally. Notwithstanding, finding a numerical solution to Maxwell’s equations is challenging in its own right. A volume of relevance to microscopy can span hundreds of wavelength per dimension, while tens of samples per wavelength are demanded for most algorithms to keep error accumulation and numerical dispersion in check. The problem’s solution alone may thus take up a significant proportion of the computer memory with its billions of free parameters, not unlike deep learning models such as OpenAI’s GTP-3.

In what follows, we show how Maxwell’s equations can be rephrased as a recurrent neural network that does not require training. This eliminates a large computational cost while ensuring that the solutions are fully deterministic and generalisable. The pre-defined neural network allows us to effectively harness the economy of scale of cloud-based machine learning infrastructure. We demonstrate its potential by calculating the light-scattering of a guide-star embedded deep within a millimetre-scale heterogeneous structure and use the exit field for refocussing. Next, we show how the highly efficient cloud-based calculations make it possible to calculate the complete scattering and deposition matrices. Finally, we analyse and compare the performance on a desktop, a GPU workstation, and on Google Colaboratory; respectively improving the efficiency 20 and 100-fold.

Results
Maxwell’s equations as a neural network We start this section by describing a two-layer forward neural network that requires slow training (Fig. 1a). Next, we incorporate an extra layer for preconditioning that accelerates training (Fig. 1b). This network is transformed into a recurrent neural network that does not require training (Fig. 1c).

A large neural network model is typically trained to produce target output values, $t$, for a large number of test cases. Training is the optimisation of parameters, $p$, to minimise the difference between the neural network’s output and the training target. This can be written as

$$p_{opt} = \arg\min_p \| N(p) - t \|,$$  

where the function $N(p)$ computes the network’s outputs for the corresponding training inputs. Such parameters typically represent the weights and biases of thousands of neurons and millions of connections. Maxwell’s equations can be rewritten in a similar form.

For coherent light with an angular frequency, $\omega$, Maxwell’s equations for an inhomogeneous material can be written as the time-independent complex vector functions,

$$\nabla \times \mathbf{E}(r) = -\frac{\partial \mathbf{B}(r)}{\partial t} = i\omega \mathbf{B}(r), \quad (2)$$

$$\nabla \times \mathbf{H}(r) = \mathbf{j}(r) + \frac{\partial \mathbf{D}(r)}{\partial t} = \mathbf{j}(r) - i\omega \mathbf{D}(r). \quad (3)$$

Provided with a source of electromagnetic radiation, $\mathbf{j}(r)$, we aim to determine the electric vector field, $\mathbf{E}(r)$. The electric displacement vector field, $\mathbf{D}(r)$, the magnetic flux density, $\mathbf{B}(r)$, and...
Fig. 1 Correspondence between electromagnetism and computational neural networks and the implications of their integration. (a) Training a large neural network with this specific design yields the solution to Maxwell’s equations. This network involves parallel paths and a convolutional layer corresponding to the Laplacian, $\nabla^2$. The green spheres represent neurons that simply sum all inputs with a linear activation function. Symbols above the arrows indicate the connection weights, of which only $E(r)$ requires training. Symbols left and right of the arrows represent the network’s inputs and outputs, respectively. While, for small problems, conventional training algorithm converges to the solution, $E(r)$, we found it to be highly unpredictable and inefficient for this simple network. (b) The preconditioned system, $\Gamma^{-1}ME = \Gamma^{-1}j$, ensures efficient and monotonic convergence. This network involves the modified permittivity, $\varepsilon'$, and modified Green’s function, $G$, which are defined in the text. The out-of-plane unlabelled connection skips a layer of neurons, not unlike a residual block. (c) The appropriate recurrent neural network can convert current density into the electric-field solution to Maxwell’s Equations. Each time-step, data is transferred in the direction of the arrows, multiplied by the arrow’s weight, and summed at the neurons. The out-of-plane connections feed the final layer’s output back to all layers, either identically, or multiplied by $V$ as indicated. The latter model is particularly efficient. (d,e) Timing tests comparing the efficiency of the algorithm shown in panel (c), executed on various platforms. (d) The median iteration time a desktop computer (Intel i7) without relying on a machine learning library, compared to using PyTorch executed on Google Colaboratory (Cloud), (e) The acceleration factor, with respect to execution on a desktop, after switching to machine learning library execution on a GPU workstation and on the cloud. (f-g) Display unprecedented size ($2.5\text{mm} \times 2.5\text{mm}$) coherent light scattering simulations. Computation of these in just 16 min each, has been enabled by the model displayed in (c). (f) show a guide-star shining light from the centre of a random scatterer, while (g) is the phase conjugation of the same system, from the left boundary. The refocusing produced a Strehl ratio of 0.48 in this system.
the magnetising field, \( \mathbf{H}(r) \), are related by the constituent relations. Without loss of generality, we consider \( \mathbf{D}(r) = \varepsilon(r) \mathbf{E}(r) \) and \( \mathbf{B}(r) = \mu_0 \mathbf{H}(r) \), where only the relative permittivity, \( \varepsilon(r) = n^2(r) \), is spatially-variant. More general relations can account for anisotropy, chiral, and magnetic properties. In a dielectric, Eq. (2) and Eq. (3) can be combined in the vector Helmholtz equation

\[
\nabla^2 \mathbf{E}(r) + k_0^2 \varepsilon_r(r) \mathbf{E}(r) = -i \omega \mu_0 \mathbf{j}(r),
\]

where \( k_0 = \omega \sqrt{\varepsilon_0 \mu_0} \). In principle, this partial differential equation can be solved numerically by discretising space and determining the field, \( \mathbf{E} \), for a source, \( \mathbf{j} \), on a sufficiently dense sampling grid. Without the position-dependency, Eq. (4) can be written succinctly as

\[
M \mathbf{E} = \mathbf{j}, \quad \text{with} \ M := \frac{i}{\omega \mu_0} (\nabla^2 + k_0^2 \varepsilon_r).
\]

The matrix, \( M \), is generally too large to be represented directly in computer memory. However, the result of its multiplication with any vector can be computed efficiently. It calculation consists of two terms. The former is a convolution. The latter term can be seen to be a diagonal matrix with the values of \( k_0^2 \varepsilon_r \) at the sample points. We calculate \( \mathbf{E} = M^{-1} \mathbf{j} \) by solving the minimisation problem for the parameters, \( \mathbf{p} \),

\[
\mathbf{E} = \arg\min_{\mathbf{p}} \| M \mathbf{p} - \mathbf{j} \|.
\]

The parallel with Eq. (1) is apparent. The neural network’s response, \( \mathbf{N}(\mathbf{p}) \), is replaced by the electromagnetic response, \( \mathbf{M} \), and the training data, \( \mathbf{t} \), is replaced by the electromagnetic source current-density, \( \mathbf{j} \). However, there are notable differences. The neural network function, \( \mathbf{N}(\mathbf{p}) \), is non-linear, and in general, any solution with sufficiently low loss is adequate for machine learning applications. In contrast, Maxwell’s equations are linear and typically only permit a unique solution. It is thus important that the optimisation of Eq. (6) deterministically converges to the true solution.

Fig. 1a shows how Maxwell’s equations can be equated with a particular neural network with parallel convolutional and multiplication layers. Although the neurons are represented as a flat layer, the activity of each input neuron corresponds to the electric field, \( \mathbf{E} \), at a point in 3D space and for each polarisation. As in the hidden layer of d’Arco et al., the activity of the corresponding neurons in the output layer is proportional to the 3D current-density source, \( \mathbf{j} \), of the electric field. Determining the scattered light field, \( \mathbf{E} \), thus corresponds to training, the neural network, i.e. optimising its parameters, so that it outputs the source current-density, \( \mathbf{j} \), of the waves. This can be done using standard training algorithms such as stochastic gradient descent (SGD) and adaptive moment estimation (Adam). While it converged to the true solution for small problems, we found that training such neural network was impractically slow. A 16\( \times \)16\( \times \)3D size system could take up to a minute to converge on an average desktop computer.

To make the neural network more responsive to training, we alter its layers. Eq. (5) can be left-multiplied by \( \Gamma^{-1} \), the inverse of a non-singular pre conditioner, leading to the equivalent expression

\[
\Gamma^{-1} M \mathbf{E} = \Gamma^{-1} \mathbf{j} \quad \text{with} \quad \Gamma := (s V - M) V^{-1}.
\]

The modified potential, \( V := -1 - k_0^2 (\varepsilon_r - \varepsilon_0) / (i \omega \mu_0) \), is determined from the material property distribution. The background permittivity \( \varepsilon_0 \in \mathbb{C} \) can be chosen to minimise \( \max_{r} |\varepsilon_r(r) - \varepsilon_0| \) for optimal convergence. The complex scaling constant, \( s \), must be chosen so that \( |1 + V| < 1 \) and \( \mathbf{N}(\mathbf{p}, s^{-1} M \mathbf{p}) \equiv 0 \) for all \( \mathbf{p} \). This is possible for any gain-free system.

The inverse preconditioner can also be written as \( \Gamma^{-1} = -s^{-1} \mathbf{V} \mathbf{G} \), using the spatially-invariant Green’s function, \( \mathbf{G} := (s^{-1} M - V)^{-1} \). This can be implemented as a convolutional neural network layer that performs the inversion \( \mathbf{G} := \mathcal{F}^{-1} \left[ \frac{1}{i \omega \mu_0} (\| \mathbf{k} \|^2 - k_0^2 \varepsilon_0) + 1 \right]^{-1} \mathcal{F} \), where \( \mathcal{F} \) denotes the Fourier transform operator. Eq. (7) can now be rewritten as \( V(1 + \mathbf{G} \mathbf{V}) \mathbf{E} = -\Gamma^{-1} \mathbf{j} \), and solved by training the neural network in Fig. 1b. Although, compared to that of Fig. 1a, this network has an extra layer, the use of preconditioning reduced the required training time by approximately 8-fold. More importantly, the preconditioning enables us to write the solution to Eq. (5) and Eq. (7) as the Neumann power series \( \mathbf{E} = \sum_{i=0}^{\infty} (1 - \Gamma^{-1} M)^i \Gamma^{-1} \mathbf{j} \) and corresponding iteration \( E_{i+1} = E_i + V E_i + V G V E_i + V G^2 \mathbb{J} \). This maps directly to the recurrent neural network depicted in Fig. 1c.

This neural network does not require training. When the current density distribution of a light source is provided as recurrent input, it infers the electric field distribution, \( \mathbf{E} \). Its accuracy can be readily verified by inserting the solution in the forward problem Helmholtz equation, represented by the network of Fig. 1a.

We implemented the recurrent method in PyTorch and integrated it into the electromagnetic solver MacroMax. Its previously CPU-bound calculations can now be seamlessly executed on top-end machine-learning infrastructure. We used the publicly available Google Colaboratory to directly study coherent light scattering on scales relevant to microscopy.

**Coherent optical scattering on the millimetre scale.** The high efficiency and minimal memory requirements of the recurrent algorithm opens a door to calculations in large heterogeneous materials. To demonstrate this, we determine the field emitted from a visible light guide-star embedded deep within a heterogeneous scatterer. Next, we refocus on the source using phase-conjugation and analyse the effectiveness of the phase conjugation. Fig. 1f shows the emission over a 2.5mm \( \times \)2.5mm-area on both sides of the 2.5mm \( \times \)1.25mm scattering material.

To refocus on a guide-star embedded within a multi-millimetre sized scattering medium we first record the wavefront on the left boundary of (Fig. 1f). Next, we conjugate phase of this wavefront and use it as a source. The wavefront can be seen to focus back onto the guide star in (Fig. 1g). It took only 16min to calculate the field in the displayed 2.5mm \( \times \)2.5mm system, running the algorithm on freely available Google Colaboratory. It is equally possible to simulate 3D spaces with an equivalent number of sample points (i.e. in this case it would be a system of approximately 110\( \mu \mathbf{m} \times 110\mu \mathbf{m} \times 110\mu \mathbf{m} \) size). To emphasise the magnitude of the problem, it is worth noting that the data displayed in Fig. 1f-g alone contains 2.2 GB of information.

**Calculating all possible internal fields.** With the significantly improved calculation efficiency, we were able to solve problems that were previously deemed impractically large to address numerically. An example that fulfils this criterion is calculating all
Fig. 2 Wavefront shaping using scattering matrix basis. (a) is the intensity distribution of a mode with the highest left-to-right transmission, meanwhile (b) is the mode with lowest transmission. The zoomed in box displays the complex field values of the ROI. The reference scalebar has length 25 μm. (c) The result of accumulative intensity timelapse of wavefront shaping into the system using a deposition matrix approach. (d) shows the average energy density from left to right for the closed channel and open channel. (e) Histogram of the transmitted fraction of the field amplitude, from left to right, and colour legend for the modes. It displays a bimodal distribution of the transmission modes predicted by the random matrix theory. Vectors corresponding to the extremes of this graph were used as input modes for graphs a-b. (f) Complete scattering matrix with 1022 modes in total. Coherent backscattering can be observed to manifest in reflection matrices (located in bottom-left and top-right quadrants). This corresponds to the increased field amplitude on the diagonals of those regions.

The internal fields for a scattering matrix. The scattering matrix is a mathematical construct that holds complete information on all the ways the light can enter and exit the system. While the mathematical formalism for the scattering matrix was derived in the early days of quantum mechanics, it did not see usage in adaptive optics until recently. Furthermore, instead of determining full scattering matrices most work focused on transmission and reflection matrices, as these are more feasible to measure experimentally. Unfortunately, physical measurements of these matrices are limited for analysis purposes, as it is practically impossible to capture all light modes in an experimental setup. But luckily, numerical field calculations can model complete and lossless data acquisition. While, recently new methods were proposed to calculate the entire scattering matrix rapidly, calculating the internal fields of scattering matrix systems remains impractically slow.

We calculated scattering matrix using visible light for a 128 μm x 128 μm heterogeneous system, with all the internal fields, in just 42 min on Google Colaboratory (Fig. 2). We use this scattering matrix (Fig. 2f) to calculate open and closed channels of the system, which are light modes that are either completely transmitted or blocked by the system, respectively (Fig. 2a-b). These modes correspond to the extremes in the transmission graph displayed in Fig. 2e, and their energy density projection as they propagate through the system is displayed in Fig. 2d.

While the scattering matrix shown in Fig. 2f is sufficient to control transmission. The controlled deposition of energy at the centre of the sample, as shown in Fig. 2c, requires full knowledge of the complete deposition matrix of size 640,000 x 1022. Focussing onto the infinity-shaped path inside the scattering system (Fig. 2c) was done by a pseudo-inversion of the matrix (see Methods).

Discussion

The timing comparisons of Fig. 1d and e show that electromagnetic field calculations are more efficient by two orders of magnitude and, conversely, that problems can be addressed with a
size that is a 100-fold larger. The efficiency advantage is already visible for 2D calculations with $10^4$ sample points, and it grows approximately linearly with the problem size.

Compared to other algorithms, such as FDFD, our neural network method is significantly more accurate for larger step sizes, it can compute the same problems much faster and it requires less memory to do so, enabling the calculation of larger systems, as shown in Fig. 3. To compute an electric field matrix, the method requires storage of 7 additional matrices of the same size, which can be reduced to 4 by a sacrifice of some computation speed.

In conclusion, we demonstrated how Maxwell’s equations can be rephrased as a recurrent neural network that yields their solution without the need for training. This stands in sharp contrast to conventional machine learning approaches, where the connection weights are determined during a computationally expensive learning phase that is prone to introduce training bias. Our recurrent neural network has a single hidden layer which performs a shifted Green’s function, while the other connection weights have a one-to-one correspondence to the problem’s permittivity distribution. By incorporating preconditioning from the outset, convergence is efficient and fully deterministic. This auspicious recurrent neural network structure enabled us to leverage powerful, yet publicly available, machine learning infrastructure on the cloud. We integrated this into the open source electromagnetic calculation library, MacroMax. This enables any researcher with an internet connection to solve 100-fold larger complex scattering problems.

Methods

Implementation of the solver. The electromagnetic solver was implemented using the PyTorch 1.1.0.1+cud11 machine learning library²⁷, and integrated into the MacroMax library⁹. The widely used PyTorch framework was chosen for its built-in ability to handle complex numbers and perform fast Fourier transforms. Its ubiquity allows us to leverage the latest technological advances and cloud computing platforms. The training of the forward neural networks, depicted in Fig. 1a-b, was done using PyTorch’s Adam optimiser. We found that Stochastic Gradient Descent converged slower for this network topology. A fast-Fourier-transform-based convolution layer was used to avoid finite differences and handle the Laplacian operation more efficiently than the built-in version. The recurrent neural network of Fig. 1c also implements the Green’s function as such convolution. Unlike the preceding topologies, this network has fixed weights and the solution as output. As such, it does not require training, and can infer the electric field on mm-scale. We integrated this approach with the electromagnetic solver, MacroMax¹³, to make it more accessible and extend its use to e.g. birefringent and magnetic materials.

Scattering system parameters. Fig. 1f-g was calculating for a wavelength of 633 nm. The scattering layer consists of packed spheres with a radius of $30 \mu m \pm 10\%$ and a refractive index of 1.33. It is 1.25 mm thick. The system is padded with 16-wavelength thick absorbing boundaries with a linearly increasing extinction coefficient from 0 to 0.2.

In Fig. 2a-c we use a 500 nm light source. Similarly, the system contains a scattering system made of spheres with radius of $2.0 \mu m \pm 5\%$ and a refractive index of 1.33. In Fig. 2a-b these spheres compose a scatterer that is 64 \mu m thick. The layer in Fig. 2c is thicker (85 nm) but has a rectangular gap in the centre of the target region. For the purpose of the mode transmission diagram in Fig. 2e, the scattering system was padded with 254 thickness boundaries, which have a linearly increasing absorption coefficient from 0 to 0.025. This was done to reduce reflection effects from the edges of the system, which interfered with the accurate calculation of displayed values. Otherwise, 163 boundaries were used, with a linearly increasing absorption coefficient from 0 to 0.2.

Calculation of deposition and scattering matrices. We compute the internally and externally scattered fields for a complete basis of incident plane waves. To facilitate the study of transmission matrices, we assume a wide slab geometry for the scatterer, orthogonal to the $z$-axis. To minimise edge effects for a finite slab, we adopt periodic boundary conditions in the transverse dimension. Common fast Fourier components are used for inward and outward travelling waves so that free space propagation corresponds to the identity matrix. The code included with this manuscript lists the basis vectors in raster-scan order for the forward and backwards propagating waves, respectively. Although deemed out of scope, pairs of orthogonal propagating polarisations are listed for vector fields. In total, the scattering and deposition matrices used to produce Fig. 2 have 1022 columns.

The deposition matrix used for Fig. 2c has 640,000 rows, one for each internal and external value of the field. While the scattering matrix is a square matrix with the same plane wave basis for the row space. The scattering matrix can be considered a $2 \times 2$-block matrix with four quadrants. The two quadrants on the diagonal correspond to the forward and backward transmission matrices, while the off-diagonal quadrants are the front and back-reflection matrices¹⁴.

To get the open and closed channels (Fig. 2a-b), we calculated the singular value decomposition of the forward transmission matrix, i.e. top-left quadrant of scattering matrix in Fig. 2f) using LAPACK’s algorithm based on a divide-and-conquer method.

A pseudo-inversion of the deposition matrix was done to refocus within the deposition region inside the scattering material. The singular value decomposition of the deposition matrix was calculated using ARPACK’s implicitly restarted Lanczos algorithm. The large computer memory requirements of this matrix restricted us to 750 eigenvectors for use in the inversion, instead of full 1022.

Evaluation of time efficiency. In Fig. 1d-e, the median iteration time is used to avoid giving excessive weight to the initial iteration which includes the initialisation steps. While the initialisation overhead is negligible for typical problems, it hampered a direct comparison for the smallest problem sizes. Typical systems require around 1000 iterations to converge, depending on the range of permittivity and the size of the system. Vectorial calculations with a polarised electric field take approximately 3.9 times longer than scalar calculations with the same number of sample points.

Data availability

All data underlying the results was generated by the algorithm described in this manuscript. The complex-valued scattering matrix shown in Fig. 2f will be made available for download after the manuscript is accepted.

Code availability

The algorithm, as well as the data visualisation, is implemented in Python using PyTorch. The complete source code with examples is openly available as a Git
repository\textsuperscript{13}. The PyTorch implementation is integrated in the MacroMax electromagnetic calculation library, which is freely available on the Python Package Index\textsuperscript{18}.

References

1. Jumper, J. \textit{et al.} Highly accurate protein structure prediction with AlphaFold. \textit{Nature} \textbf{596}, 583–589 (2021).
2. Thendiyammal, A., Osnabrugge, G., Knop, T. \& Vellekoop, I. M. Model-based wavefront shaping microscopy. \textit{Optics letters} \textbf{45}, 5101–5104 (2020).
3. Molesky, S. \textit{et al.} Inverse design in nanophotonics. \textit{Nature Photonics} \textbf{12}, 659–670 (2018).
4. Tafove, A., Hagness, S. C. \& Piket-May, M. Computational electromagnetics: the finite-difference time-domain method. \textit{The Electrical Engineering Handbook} \textbf{3} (2005).
5. Floridi, L. \& Chiriatti, M. GPT-3: Its nature, scope, limits, and consequences. \textit{Minds and Machines} \textbf{30}, 681–694 (2020).
6. He, K., Zhang, X., Ren, S. \& Sun, J. Deep residual learning for image recognition in Proceedings of the IEEE conference on computer vision and pattern recognition (2016), 770–778.
7. Goodfellow, I., Bengio, Y. \& Courville, A. \textit{Deep learning} (MIT press, 2016).
8. Devlin, J., Chang, M.-W., Lee, K. \& Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. \textit{arXiv preprint arXiv:1810.04805} (2018).
9. Vettenburg, T., Horsley, S. A. R. \& Bertolotti, J. Calculating coherent light-wave propagation in large heterogeneous media. \textit{Opt. Express} \textbf{27}, 11946–11967. doi: 10.1364/OE.27.011946 (Apr. 2019).
10. d’Arco, A., Xia, F., Boniface, A., Dong, J. \& Gigan, S. Physics-based neural network for non-invasive control of coherent light in scattering media https://arxiv.org/abs/2206.00487. 2022.
11. Kingma, D. P. \& Ba, J. Adam: A method for stochastic optimization. \textit{arXiv preprint arXiv:1412.6980} (2014).
12. Osnabrugge, G., Leedumrongwatthanakun, S. \& Vellekoop, I. M. A convergent Born series for solving the inhomogeneous Helmholtz equation in arbitrarily large media. \textit{Journal of Computational Physics} \textbf{322}, 113–124 (2016).
13. T. Vettenburg \& L. Valantinas. MacroMax: Calculating electromagnetic scattering in heterogeneous materials. \textit{https://github.com/corilim/MacroMax}. 2022. Online; last accessed August 3, 2022.
14. Rotter, S. \& Gigan, S. Light fields in complex media: Mesoscopic scattering meets wave control. \textit{Rev. Mod. Phys.} \textbf{89}, 015005. doi: 10.1103/RevModPhys.89.015005 (1 Mar. 2017).
15. Akkermans, E., Wolf, P. \& Maynard, R. Coherent backscattering of light by disordered media: Analysis of the peak line shape. \textit{Physical review letters} \textbf{56}, 1471 (1986).
16. Prashanth, B., Mendu, M. \& Thallapalli, R. Cloud based Machine learning with advanced predictive Analytics using Google Colaboratory. \textit{Materials Today: Proceedings} (2021).
17. Mehra, J. \& Rechenberg, H. \textit{The historical development of quantum theory} (Springer Science \& Business Media, 2000).
18. Popoff, S. M. \textit{et al.} Measuring the Transmission Matrix in Optics: An Approach to the Study and Control of Light Propagation in Disordered Media. \textit{Phys. Rev. Lett.} \textbf{104}, 100601 (Mar. 2010).
19. Yu, H. \textit{et al.} Measuring large optical transmission matrices of disordered media. \textit{Phys. Rev. Lett.} \textbf{111}, 153902–153902. ISSN: 0031-9007 (Oct. 2013).
20. H. Yu, J.-H. Park \& Y. Park. Measuring large optical reflection matrices of turbid media. \textit{Opt. Commun.} \textbf{352}, 33–38. ISSN: 0030-4018. doi:https://doi.org/10.1016/j.optcom.2015.04.073 (2015).
21. Yu, H., Lee, K. \& Park, Y. Energy leakage in partially measured scattering matrices of disordered media. \textit{Physical Review B} \textbf{93}, 104202. doi: 10.1103/PhysRevB.93.104202 (10 Mar. 2016).
22. Lin, H.-C., Wang, Z. \& Hsu, C. W. Full-wave solver for massively multi-channel optics using augmented partial factorization. \textit{arXiv preprint arXiv:2205.07887} (2022).
23. Bender, N. \textit{et al.} Depth-targeted energy delivery deep inside scattering media. \textit{Nature Physics}, 1–7 (2022).
24. Shin, W. \& Fan, S. Choice of the perfectly matched layer boundary condition for frequency-domain Maxwell’s equations solvers. \textit{Journal of Computational Physics} \textbf{231}, 3406–3431. ISSN: 0021-9991. doi:https://doi.org/10.1016/j.jcp.2012.01.013 (2012).
25. Shin, W. \textit{MaxwellFDFD} https://github.com/wsshin/maxwellfdfd. 2015. Online; last accessed August 3, 2022.
26. Vettenburg, T. \& Vellekoop, I. M. A universal preconditioner for linear systems https://arxiv.org/abs/2207.14222. 2022.
27. Paszke, A. \textit{et al.} Pytorch: An imperative style, high-performance deep learning library. \textit{Advances in neural information processing systems} \textbf{32}, 8026–8037 (2019).
28. \textit{Python Package Index - PyPI - MacroMax} https://pypi.org/project/macromax/.

Acknowledgments

L. Valantinas’ research is supported by EPSRC Grant: EP/R513192/1. T. Vettenburg is a UKRI Future Leaders Fellow supported by grant MR/S034900/1.

Author contributions

LV integrated PyTorch into the MacroMax library, performed the numerical calculations, and created all figures. TV developed the theory and a proof-of-principle implementation, he adapted the MacroMax library for multi-platform execution and extended it to automate scattering matrix calculations. Both authors wrote and reviewed the manuscript.

Competing interests

The authors declare no competing interests.
Additional information

Correspondence and requests for materials can be addressed to
T. Vettenburg <t.vettenburg@dundee.ac.uk>. 