scientific reports

Check for updates

OPEN

Physics-constrained machine learning for electrodynamics without gauge ambiguity based on Fourier transformed Maxwell's equations

Christopher Leon✉ & Alexander Scheinker

We utilize a Fourier transformation-based representation of Maxwell's equations to develop physics-constrained neural networks for electrodynamics without gauge ambiguity, which we label the Fourier–Helmholtz–Maxwell neural operator method. In this approach, both of Gauss's laws and Faraday's law are built in as hard constraints, as well as the longitudinal component of Ampère–Maxwell in Fourier space, assuming the continuity equation. An encoder–decoder network acts as a solution operator for the transverse components of the Fourier transformed vector potential, $\hat{\mathbf{A}}_{\perp}(\mathbf{k}, t)$, whose two degrees of freedom are used to predict the electromagnetic fields. This method was tested on two electron beam simulations. Among the models investigated, it was found that a U-Net architecture exhibited the best performance as it trained quicker, was more accurate and generalized better than the other architectures examined. We demonstrate that our approach is useful for solving Maxwell's equations for the electromagnetic fields generated by intense relativistic charged particle beams and that it generalizes well to unseen test data, while being orders of magnitude quicker than conventional simulations. We show that the model can be re-trained to make highly accurate predictions in as few as 20 epochs on a previously unseen data set.

Electrodynamics govern a wide range of physical processes involving charged particle dynamics including universe expansion models, galactic disk formation, accelerator-based high energy X-ray light sources, achromatic metasurfaces, metasurfaces for dynamic holography, on-chip diffractive neural networks, and radiative damping of individual accelerated electrons[1–21].

Despite widely available high performance computing, numerically calculating relativistic charged particle dynamics is still a challenge and an open area of research for large collections of particles undergoing collective effects in dynamics involving plasma turbulence[22], space charge (SC) forces[23,24], and coherent synchrotron radiation (CSR)[25,26]. For example, the photo-injectors of modern X-ray free electron lasers such as the LCLS, SwissFEL, and EuXFEL and plasma wakefield accelerators such as FACET-II can produce high quality intense bunches with few picosecond rms lengths of up to 2 nC charge per bunch that are accelerated and squeezed down to lengths of tens to hundreds of femtoseconds[27–32]. At low energy (~ 5 MeV) the 6D phase space (x, y, z, p_x, p_y, p_z) dynamics of such bunches are strongly coupled through collective SC forces. At high energy, when particle trajectories are curved through magnetic chicanes the dynamics are coupled by CSR.

For example, a 2 nC bunch contains $N \approx 1.25 \times 10^{10}$ electrons, for which calculating individual particle to particle SC or CSR interactions is a computationally expensive $\mathcal{O}\ (N^2)$ process. Conventional approaches can reduce the number of SC calculations by utilizing particle-in-cell methods, but state-of-the-art 3D CSR calculations still rely on point-to-point calculations[33]. Machine learning (ML) methods have demonstrated to be useful in non-invasive measurements and diagnostics of charged beams[34] and in the simulations of magnetic fields through physics-informed neural networks[35]. They have also been found to speed up lattice quantum Monte Carlo simulations[36], the design and simulation of fin field-effect transistors[37], the simulation of spin dynamical systems[38] and finding the optimal ramp up for the production of Bose–Einstein condensates[39].

Recently a new method was developed for enforcing hard physics constraints in ML, respecting physics and improving generalization[40]. In that work, 3D convolutional neural operators were used to generate the vector and

Los Alamos National Laboratory, Los Alamos, NM 87545, USA. ✉email: cleon@lanl.gov

scalar potential fields $\mathbf{A}(\mathbf{r},t)$ and $\varphi(\mathbf{r},t)$, respectively, from which the electromagnetic fields were generated as $\mathbf{B}(\mathbf{r},t)=\nabla\times\mathbf{A}(\mathbf{r},t)$, $\mathbf{E}(\mathbf{r},t)=-\nabla\varphi(\mathbf{r},t)-\partial\mathbf{A}(\mathbf{r},t)/\partial t$. Generating fields based on potentials as in[40] enforces hard physics constraints, such as $\nabla\cdot\mathbf{B}(\mathbf{r},t)=0$ and $\nabla\times\mathbf{E}+\partial\mathbf{B}/\partial t=0$, but it also adds gauge ambiguity. Building on this previous work, in this work we present a novel method for predicting electromagnetic fields, which utilizes a Fourier transformation-based representation of Maxwell's equations and physics-constrained neural networks (PCNN) for electrodynamics without gauge ambiguity. In this approach, many of Maxwell's equations are hard constraints: Gauss's law for the electric and magnetic field, Faraday's law and the longitudinal component (in Fourier space) of the Ampère–Maxwell's law. We demonstrate this method for calculating the electromagnetic fields of intense relativistic charged particle beams.

Note: a hard constraint is one that is automatically satisfied. For example, using potentials, the condition $\nabla\cdot\mathbf{B}(\mathbf{r},t)=0$ automatically follows (up to numerical errors). This contrasts with a soft constraint, where one tries to guide the network to satisfy a condition by having a term in the loss function, such as $L_\nabla=\sum_{i,j}|\nabla\cdot\mathbf{B}(\mathbf{r}_i,t_j)|^2$ .This term encourages the network to decrease the divergence at the specified locations and times during training but offers no guarantee the network's output will be divergenceless.

Formalism

Maxwell's equations in Fourier space

The formalism that will be employed here is also used in the canonical quantization of the electromagnetic field[41], where distinguishing between the physical degrees of freedom and the gauge redundancies is important. In the context of our study, we use it to minimize the number of fields needed to be modeled, while also ensuring that a substantial number of Maxwell's equations are automatically satisfied as hard constraints. We start with Maxwell's equations in position space:

$$\nabla\cdot\mathbf{E}=\frac{\rho}{\varepsilon_0},\quad \nabla\cdot\mathbf{B}=0,\quad \nabla\times\mathbf{E}=-\frac{\partial\mathbf{B}}{\partial t},\quad \nabla\times\mathbf{B}=\mu_0\left(\mathbf{J}+\varepsilon_0\frac{\partial\mathbf{E}}{\partial t}\right),\tag{1}$$

where $\mathbf{E}(\mathbf{r},t)$, $\mathbf{B}(\mathbf{r},t)$, $\mathbf{J}(\mathbf{r},t)$, and $\rho(\mathbf{r},t)$ are the time-varying electric field, magnetic field, current density, and charge density, respectively[42]. We perform a 3D spatial Fourier transform on Maxwell's equations, which is defined for a vector field $\mathbf{F}(\mathbf{r},t)$ as

$$\hat{\mathbf{F}}(\mathbf{k},t)=\frac{1}{(2\pi)^{3/2}}\iiint\mathbf{F}(\mathbf{r},t)e^{-i\mathbf{r}\cdot\mathbf{k}}d^3\mathbf{r},\tag{2}$$

and rewrite Eq. (1) (assuming $\mathbf{k}\neq\mathbf{0}$; see "Appendix A" for a discussion of the zero case):

$$i\mathbf{k}\cdot\hat{\mathbf{E}}=\frac{\hat{\rho}}{\varepsilon_0},\quad i\mathbf{k}\cdot\hat{\mathbf{B}}=0,\quad i\mathbf{k}\times\hat{\mathbf{E}}=-\frac{\partial\hat{\mathbf{B}}}{\partial t},\quad i\mathbf{k}\times\hat{\mathbf{B}}=\mu_0\left(\hat{\mathbf{J}}+\varepsilon_0\frac{\partial\hat{\mathbf{E}}}{\partial t}\right).\tag{3}$$

For each wave vector $\mathbf{k}$, a generic vector can be separated into components longitudinal and transverse to $\mathbf{k}$:

$$\hat{\mathbf{F}}(\mathbf{k},t)=\hat{\mathbf{F}}_\parallel(\mathbf{k},t)+\hat{\mathbf{F}}_\perp(\mathbf{k},t),\quad \text{such that}\quad \mathbf{k}\cdot\hat{\mathbf{F}}_\perp(\mathbf{k},t)=0,\quad \mathbf{k}\times\hat{\mathbf{F}}_\parallel(\mathbf{k},t)=\mathbf{0}.\tag{4}$$

An inverse Fourier transformation of Eq. (4) will lead to a Helmholtz decomposition of $\mathbf{F}(\mathbf{r},t)$. Gauss's law for the electric and magnetic fields (Eq. (3)) can then be expressed as:

$$\hat{\mathbf{E}}_\parallel(\mathbf{k},t)=-i\mathbf{k}\frac{\hat{\rho}(\mathbf{k},t)}{\varepsilon_0|\mathbf{k}|^2},\quad \hat{\mathbf{B}}_\parallel(\mathbf{k},t)=\mathbf{0}.\tag{5}$$

In Fourier space, the electric and magnetic fields' dependence on the Fourier transformed scalar and vector potential, $\hat{\phi}(\mathbf{k},t)$ and $\hat{\mathbf{A}}(\mathbf{k},t)$, is given by $\hat{\mathbf{E}}(\mathbf{k},t)=-i\mathbf{k}\hat{\phi}(\mathbf{k},t)-\frac{\partial\hat{\mathbf{A}}(\mathbf{k},t)}{\partial t}$, $\hat{\mathbf{B}}(\mathbf{k},t)=i\mathbf{k}\times\hat{\mathbf{A}}(\mathbf{k},t)$. The transverse field components are:

$$\hat{\mathbf{E}}_\perp(\mathbf{k},t)=-\frac{\partial\hat{\mathbf{A}}_\perp(\mathbf{k},t)}{\partial t},\quad \hat{\mathbf{B}}_\perp(\mathbf{k},t)=i\mathbf{k}\times\hat{\mathbf{A}}_\perp(\mathbf{k},t).\tag{6}$$

Using Eq. (3), Faraday's law trivially follows:

$$-i\mathbf{k}\times\frac{\partial\hat{\mathbf{A}}_\perp(\mathbf{k},t)}{\partial t}=-\frac{\partial}{\partial t}i\mathbf{k}\times\hat{\mathbf{A}}_\perp(\mathbf{k},t).\tag{7}$$

The transverse components of the Ampère–Maxwell equation become:

$$\frac{1}{c^2}\frac{\partial^2\hat{\mathbf{A}}_\perp(\mathbf{k},t)}{\partial t^2}+|\mathbf{k}|^2\hat{\mathbf{A}}_\perp(\mathbf{k},t)=\mu_0\hat{\mathbf{J}}_\perp(\mathbf{k},t),\tag{8}$$

while using Eq. (5) it can be shown that the longitudinal components of the Ampère–Maxwell equation is equivalent to the continuity equation. The continuity equation also relates $\hat{\mathbf{J}}_\parallel$ to $\hat{\rho}$, thus only $\hat{\rho}$ and $\hat{\mathbf{J}}_\perp$ need be used.

From Eqs. (5) and (6), the EM fields can then be determined by:

$$\hat{\mathbf{E}}(\mathbf{k},t) = -i\mathbf{k}\frac{\hat{\rho}(\mathbf{k},t)}{\varepsilon_0|\mathbf{k}|^2} - \frac{\partial \hat{\mathbf{A}}_\perp(\mathbf{k},t)}{\partial t}, \quad \hat{\mathbf{B}}(\mathbf{k},t) = i\mathbf{k}\times\hat{\mathbf{A}}_\perp(\mathbf{k},t), \tag{9}$$

followed by inverse Fourier transforms.

Fourier–Helmholtz–Maxwell neural operator

Given the charge and current density distributions, the EM Fields can be by determined by $\hat{\mathbf{A}}_\perp(\mathbf{k},t)$, as per Eq. (9), and $\hat{\rho}(\mathbf{k},t)$. $\hat{\mathbf{A}}_\perp(\mathbf{k},t)$ can be found from $\hat{\mathbf{J}}_\perp(\mathbf{k},t)$ via Eq. (8). From this, we propose finding the generated EM fields using the approach depicted in the diagram of Fig. 1, where the hard constraints (ones automatically obeyed) are listed. Since this approach incorporates Fourier transformations, Helmholtz decompositions, and Maxwell's equations within a neural operator framework, we refer to it as the Fourier–Helmholtz–Maxwell neural operator (FoHM-NO) method.

In this work, we attempt to estimate the EM fields from ρ and $\mathbf{J}$ using the FoHM-NO method, specifically with convolutional neural networks. When applying neural networks to physics problems, prior theoretical knowledge can be leveraged to model useful quantities already identified by physicists (i.e., $\hat{\mathbf{A}}_\perp(\mathbf{k},t)$), rather than the direct observables (i.e., $\mathbf{E}(\mathbf{r},t)$ and $\mathbf{B}(\mathbf{r},t)$). This has been demonstrated previously through the use of neural networks to model the Hamiltonian of classical systems[43], the scalar and vector potentials to estimate EM fields[40] and the stream function of an incompressible fluid's velocity flow field[44].

Ultimately, the FoHM-NO method exploits that the EM fields have only 2 independent degrees of freedom, as can be seen from Eq. (9) in the source-less case. On a fundamental level though, this stems from the photon being a spin-1 massless particle. One benefit of the method is that it require fewer fields than the 6 that would be needed for directly estimating both the electric and magnetic field, or the 4 necessary by using the scalar potential and the entire vector potential. With more fields to fit, there comes increasing tension between fitting the data while also satisfying physical conditions. In contrast, with this procedure by construction three of the four Maxwell's equations (Eqs. (5) and (7)) are built-in as hard constraints. This is also the case for the longitudinal component of Ampère–Maxwell's Law, which follows from the continuity equation and Gauss's law for the electric field (see "Appendix B"). Working in Fourier space, meanwhile, has the advantage that spatial derivatives are traded for multiplication by wave vectors. Thus, constraints like Eq. (4) can be easier to satisfy than their spatial derivative counterparts. For a fast Fourier transform (FFT), if N is the total number of spatial data points, then the FFT has a time complexity of $\mathcal{O}\left(N\log N\right)$. This, along with modern GPU computing, means that moving to and fro from Fourier space can be done extremely quickly, even for very large data sets. We did not perform a temporal Fourier transform as the time length of the simulations varied. This variation would pose challenges to the networks to learn in the frequency domain. Additionally, the adoption of a 4D CNN would present computational challenges and significantly increases the number of parameters in models.

Computational EM methods done in configuration space have a notable advantage: an easy, straight-forward incorporation of spatial boundary conditions, which can be useful in solving many real-world EM problems. Here, we don't consider any special spatial boundary conditions. For one, the simulation was just started with the initial conditions of the particles. Secondly, neural networks have been found to be useful in solving inverse PDE problems, where the problem may be ill-posed. For example, insufficient boundary conditions, a case where classical methods struggle or cannot be used[45–47]. We wish to apply our approach to inverse problems in future works.

An additional benefit of FoHM-NO is that by focusing on just $\hat{\mathbf{A}}_\perp(\mathbf{k},t)$ rather than all the potential fields, the issue of gauge redundancy is bypassed. To see this, note that under a gauge transformation,

$$\hat{\phi}(\mathbf{k},t) \to \hat{\phi}(\mathbf{k},t) - \frac{\partial \hat{f}(\mathbf{k},t)}{\partial t}, \quad \hat{\mathbf{A}}(\mathbf{k},t) \to \hat{\mathbf{A}}(\mathbf{k},t) + i\mathbf{k}\hat{f}(\mathbf{k},t), \tag{10}$$

for some scalar function $\hat{f}(\mathbf{k},t)$. Since the vector potential change occurs in the $\mathbf{k}$ direction then,

$$\hat{\mathbf{A}}_\parallel(\mathbf{k},t) \to \hat{\mathbf{A}}_\parallel(\mathbf{k},t) + i\mathbf{k}\hat{f}(\mathbf{k},t), \quad \hat{\mathbf{A}}_\perp(\mathbf{k},t) \to \hat{\mathbf{A}}_\perp(\mathbf{k},t). \tag{11}$$

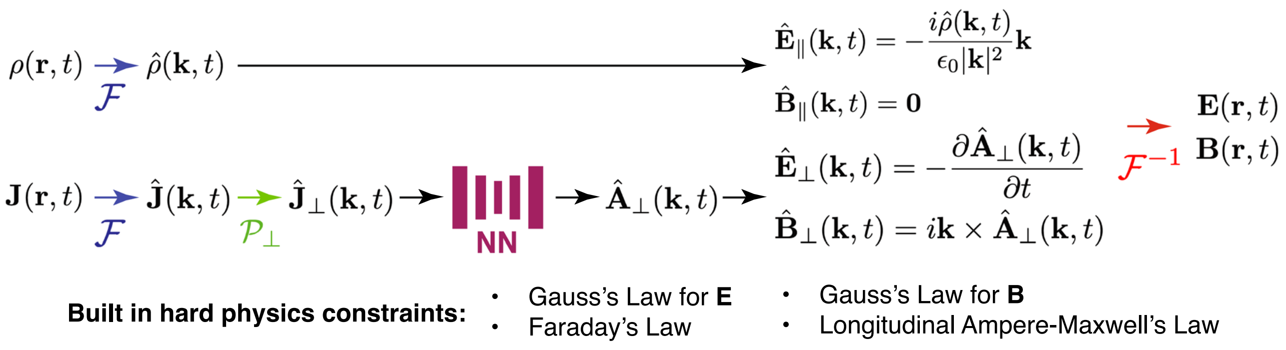
$\rho(\mathbf{r},t)$
$\mathcal{F}$
$\hat{\rho}(\mathbf{k},t)$
$\hat{\mathbf{E}}_\parallel(\mathbf{k},t) = -\frac{i\hat{\rho}(\mathbf{k},t)}{\epsilon_0|\mathbf{k}|^2}\mathbf{k}$
$\hat{\mathbf{B}}_\parallel(\mathbf{k},t) = \mathbf{0}$
$\mathbf{J}(\mathbf{r},t)$
$\mathcal{F}$
$\hat{\mathbf{J}}(\mathbf{k},t)$
$\mathcal{P}_\perp$
$\hat{\mathbf{J}}_\perp(\mathbf{k},t)$
NN
$\hat{\mathbf{A}}_\perp(\mathbf{k},t)$
$\hat{\mathbf{E}}_\perp(\mathbf{k},t) = -\frac{\partial\hat{\mathbf{A}}_\perp(\mathbf{k},t)}{\partial t}$
$\hat{\mathbf{B}}_\perp(\mathbf{k},t) = i\mathbf{k}\times\hat{\mathbf{A}}_\perp(\mathbf{k},t)$
$\mathcal{F}^{-1}$
$\mathbf{E}(\mathbf{r},t)$
$\mathbf{B}(\mathbf{r},t)$
Built in hard physics constraints:
Gauss's Law for **E**
Faraday's Law
Gauss's Law for **B**
Longitudinal Ampere-Maxwell's Law

Figure 1. The FoHM-NO method and its built-in hard physics constraints. $\mathscr{F}$ denotes Fourier transformation, $\mathscr{P}_\perp$ is the projection of the vector field to components transverse to $\mathbf{k}$, NN a neural network, and $\mathscr{F}^{-1}$ is inverse Fourier transformation.

Hence, by using just $\hat{\mathbf{A}}_\perp(\mathbf{k}, t)$ there is no need to choose a particular gauge. The gauge invariance of $\hat{\mathbf{A}}_\perp(\mathbf{k}, t)$ is well known in works involving the quantization of the EM field[41,48,49] and the spin-orbit decomposition of gauge fields[50].

Care must be taken, though, if one were to change inertial reference frames as $\hat{\mathbf{A}}_\perp(\mathbf{k}, t)$, unlike the gauge-dependent $A^\mu = (\phi/c, \mathbf{A})$, is not a Lorentz 4-vector. For our purposes, which is finding efficient and accurate computational methods for accelerator physics, this is not a serious issue.

The rest of this article is a proof-of-concept application of FoHM-NO for predicting the EM fields generated by relativistically charged particle beams. However, it is applicable to any EM setting where ρ and $\mathbf{J}$ are given, although it can also be extended to include the matter density fields in the predictions if the equations of state are used. While FoHM-NO may generally not perform as accurately as mature, state-of-the-art EM codes, the use of neural networks here gives it the advantage of having a very fast runtime. This can be useful in cases where speed is essential, such as accelerator control and diagnostics settings, where beam time can be quite limited. Alternatively, it can be used for exploratory analyses to find approximate solutions before bringing in more accurate, but slower, methods for refinement.

Data

Two simulations of electrons beams entering a solenoid magnetic fields were generated (see Fig. 2) to develop the FoHM-NO method. The first, the complex beam, was generated to train the neural networks. This complex beam was initially composed of several Gaussian bunches with spreads differing both between bunches and along the x, y, and z axes (see Fig. 3). The second, the Gaussian Beam, was generated to test the networks on a completely new unseen distribution which was not used for model training (see Fig. 4). This test beam starts as a Gaussian bunch before entering the magnetic field. This enabled us to test the method under very different scenarios,

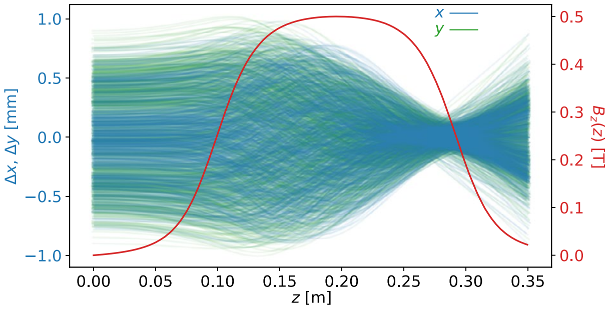

Figure 2. The (x, y) trajectories of 1000 random electrons within the beam are shown along with the z-component of the 0.5 Tesla magnetic field of the solenoid which over-compresses the beam.

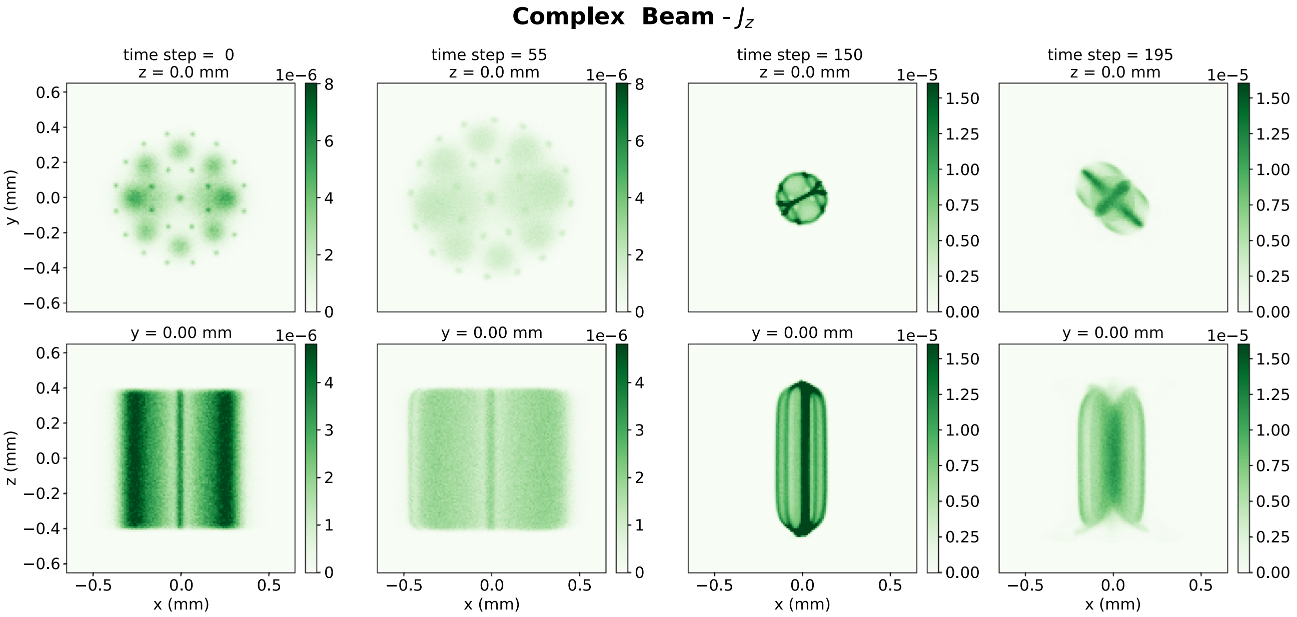

Figure 3. Cross-sections showing the spatial distribution of J_z for the complex beam. The top row of plots shows x–y slices at z = 0 mm for different time steps. The bottom row shows x–z slices at y = 0 at the same time steps.

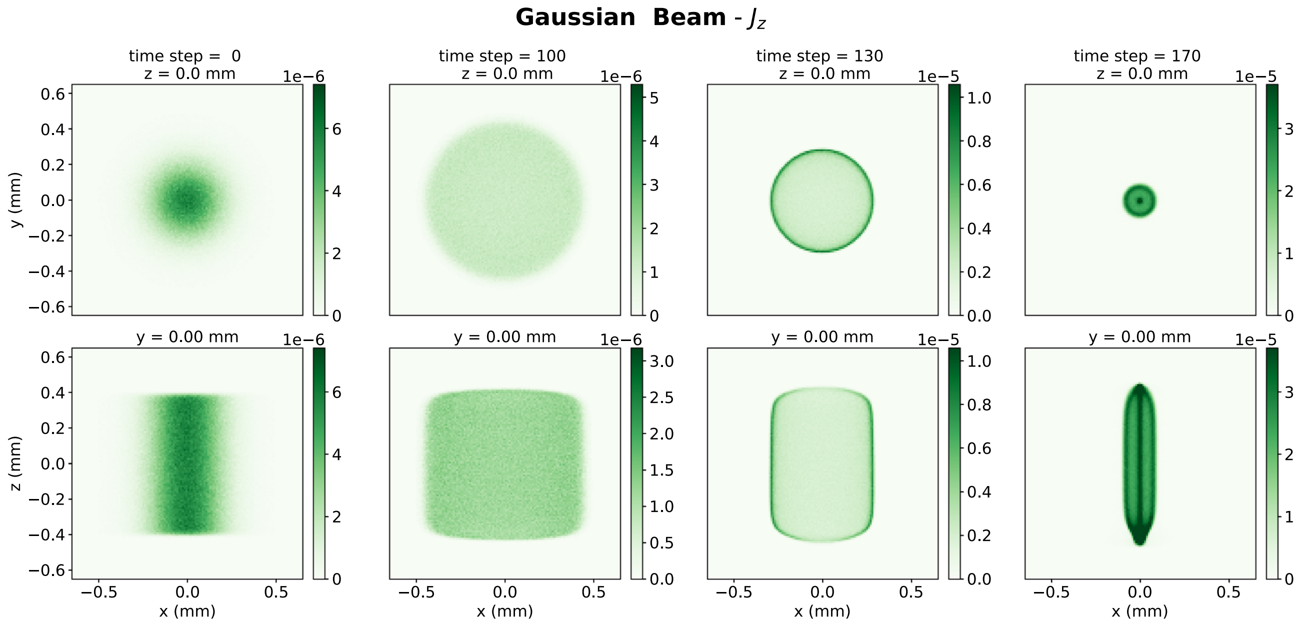

Figure 4. Cross-sections showing the spatial distribution of J_z for the Gaussian beam. The top row of plots shows x–y slices at z = 0 mm for different time steps. The bottom row shows x–z slices at y = 0 at the same time steps.

with the more intricate complex beam having multiple length scales in order to train the networks over a wider range of inputs. The Gaussian beam, which is much more simple, was used as a test as it more closely resembles physical beams in accelerators. All the data was generated using general particle tracer (GPT)[51,52].

Considering the energy scales involved, GPT simulations only considered the space charge forces. For computational efficiency, GPT only registers EM fields where particles are present, creating artifacts where the EM field suddenly drop to **0** where particles are absent. To address this issue, a large neutral group of particles co-moving with the beam was added to enable the registration of EM fields on the whole domain of interest without altering the motion of the electrons. Although the simulation took place on a 128 × 128 × 128 spatial lattice, there were artifacts where the EM fields abruptly dropped to zero on the edges (not the matter fields, however). Therefore, the EM fields were cropped to 126 × 126 × 126 for both beams.

Each simulations ran a 2 nC bunch of electrons with average energy 5.6 MeV moving through a 0.5 T solenoid magnetic field, first being compressed and then expanding (see Fig. 3). The space is a 128 × 128 × 128 spatial grid. There were 201 time steps for the complex beam and 171 for the simple beam. The beam's dynamics were simulated over a total time of 1.2 ns during which it traveled over a 0.35 m distance. The beam's charge density $\rho(\mathbf{r}, t)$ and current density $\mathbf{J}(\mathbf{r}, t)$ were binned as 3D histograms over cubes with 1.3 mm side lengths, as shown in Figs. 3 and 4. The physical spacing of grid were:

$$\Delta x = \Delta y = \Delta z = 10.2\,\mu\text{m}, \quad \Delta t = 6\,\text{ps}. \tag{12}$$

To improve convergence, we first scaled the data according to standard deviation. The data had a heavy tail, so in addition a log-like transform was performed $\Lambda(x) = \text{sgn}(x) \ln(1 + |x|)$, which has the property that larger values get greatly reduced, but smaller values stay about the same since $\Lambda(x) \approx x$ for $|x| \ll 1$.

In this work, there were no special boundary condition considerations, an external magnetic field as generated by a cylindrically symmetric solenoid was provided as input. As charged particles moved through this field their dynamics were affected by the external solenoid field and by their own self-generated electric and magnetic fields which were calculated according to Maxwell's equations based on the charge density and current density of the charged particle beam. In this setup the accelerator beam-pipe was assumed to be of large enough radius so that image charges and currents induced in the accelerator walls did not need to be considered. In future work, boundary conditions including image charges and Maxwell's equations on the surfaces of conductors will be incorporated into the ML model as additional hard physics constraints.

Physics-constrained neural networks

Neural networks (NN's) are powerful machine learning tools that, by the universal approximation theorem, can be used to approximate a function to arbitrary precision[53,54]. NN's ability to approximate go even further by being universal approximators for operators[55]. This has been harnessed to produce approximate solutions to partial differential equations[56,57], 3D electrodynamics[40], and also two spatial dimensional magnetohydrodynamics simulations[58,59].

The inclusion of physics information, constraints, or symmetries into NN be used to help guide training[60]. In some cases, these physics-incorporated NN's can produce accurate predictions that are less computationally demanding than numerical simulations, while also being easier to develop and implement[61]. Here, we make use of a convolutional encoder–decoder architectures to make an NN that has $\hat{\mathbf{J}}_\perp$ as input and $\hat{\mathbf{A}}_\perp$ as output.

Models and fit

For a relativistic beam propagating along the z direction, we generally have $|J_z| \gg |J_x|, |J_y|$ and consequentially it is expected that the magnetic field is predominately in the x and y direction: $|B_x|, |B_y| \gg |B_z|$. It was verified that this was indeed the case for both beam simulations used here. In fact, B_z was both small and noisy, likely heavily corrupted by numerical errors. Thus, to make things more computationally efficient and increase training speed, the model was trained only using the z component of $\hat{\mathbf{J}}_\perp$ to predict the z component of $\hat{\mathbf{A}}_\perp$:

$$\Re(\hat{J}_{\perp,z}), \Im(\hat{J}_{\perp,z}) \rightarrow NN \rightarrow \Re(\hat{A}_{\perp,z}), \Im(\hat{A}_{\perp,z}). \tag{13}$$

From this the transverse components of the magnetic field, $\hat{B}_x$ and $\hat{B}_y$, are obtained via $\hat{\mathbf{B}} = i\mathbf{k} \times (\hat{A}_{\perp,z}\hat{\mathbf{z}})$. The models were trained on the first 85% of the complex beam run, which is the training set. The remainder, validation set, was used for hyperparameter tuning. The entire Gaussian Beam data was used as test data.

Architectures

Different architectures for the neural network were explored to optimize performance. The first models trained had a CNN encoder–decoder architecture design, which is depicted in Fig. 5 (once the skip connections are removed). This was chosen as the reduction and reconstruction from the small latent space acts as an effective regularizer. In addition, it was successful in a previous beam simulation task[40]. The encoder compresses the input $\hat{J}_{\perp,z}$ to the latent space via various functional compositions, represented as layers. From there, two decoder arms are applied to the latent space, one producing $\Re(\hat{A}_{\perp,z})$ and the other $\Im(\hat{A}_{\perp,z})$. Empirically, this bifurcated structure produced better results than having a single decoder that gave the both real and imaginary components of $\hat{A}_{\perp,z}$ as output.

We made two models following Fig. 5 differing by the number of filters in the latent space, 4 and 6 filters, which we label CNN-small-4 and CNN-small-6, respectively. Also, we used an architecture similar to CNN-small-4, but with more filters in the hidden layers and thus more parameters, dubbed here CNN-large. We also investigated the adaption of an U-Net encoder–decoder architecture[62], visually depicted in Fig. 5, which we call U-Net. U-Nets have found applications in scientific computation[63,64] and computer vision. One limitation of the CNN encoder–decoder architecture in Fig. 5 is that information of the input gets 'forgotten' as it's compressed to the latent space. The U-Net addresses this by adding 'skipped connections' between hidden layers of the encoder and decode that share the same dimensions. These connections concatenate the feature maps, allowing the decoder to pull information from the encoder and generally reproduce finer details more accurately than CNN architectures without skipped connections. The skipped connections are also useful during the training process by allowing for more gradient flow, thus can act as an effective tool against vanishing gradients and can speed up training.

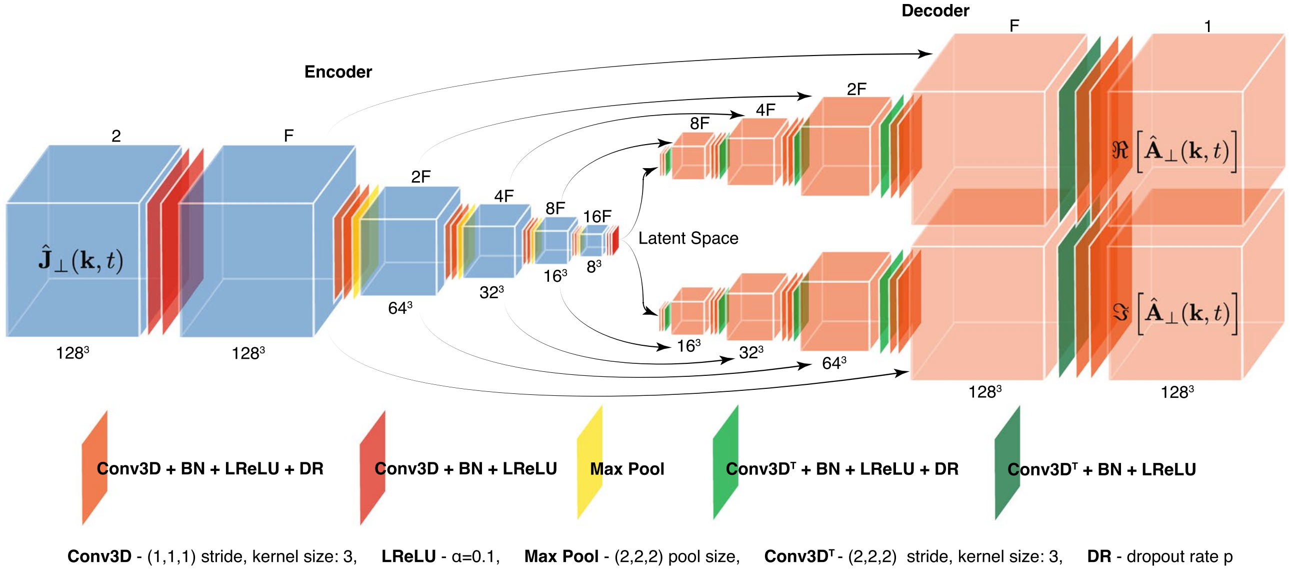
Decoder
Encoder
F
1
2
F
2F
4F
8F
4F
2F
8F
16F
Latent Space
$\hat{\mathbf{J}}_\perp(\mathbf{k}, t)$
$\Re\left[\hat{\mathbf{A}}_\perp(\mathbf{k}, t)\right]$
$\Im\left[\hat{\mathbf{A}}_\perp(\mathbf{k}, t)\right]$
128^3
128^3
64^3
32^3
16^3
8^3
16^3
32^3
64^3
128^3
128^3
Conv3D + BN + LReLU + DR
Conv3D + BN + LReLU
Max Pool
Conv3D^T + BN + LReLU + DR
Conv3D^T + BN + LReLU
Conv3D - (1,1,1) stride, kernel size: 3, LReLU - α=0.1, Max Pool - (2,2,2) pool size, Conv3D^T - (2,2,2) stride, kernel size: 3, DR - dropout rate p

Figure 5. The adapted U-Net architecture is shown. For the number of starting filters we took $F = 16$. The skip connection concatenates two feature maps of the same dimensions. The CNN architectures were very similar, with the skip connections removed and the number of filters in each layer changed with CNN-small-4 having filter numbers 16, 32, 64, 128, 64, 4, 64, 64, 32, 16 between the inputs and outputs. Having a double convolution layer before the in the input and output improved results. The output here is also cropped to $126 \times 126 \times 126$ (see text).

Fits
An L_2 penalty for the weights and dropout with $p = 0.05$ were used for regularization. All the NN's were created, trained and tested using Tensorflow[65], with training being done using the Adam optimizer[66] with early stopping based on validation loss. All the machine learning training was done on a Dell Precision 7920 with two Nvidia RTX A6000 GPU's with 48 GB VRAM each. A summary of the four architectures with a description can be found in Table 1.

Cost function and metric
NNs were trained to minimize mean squared error (MSE) in Fourier space summed over each time step in the training data:

$$C = \sum_i \int d^3\mathbf{k} \left| \hat{\mathbf{B}}(\mathbf{k}, t_i) - \hat{\mathbf{B}}_{NN}(\mathbf{k}, t_i) \right|^2, \tag{14}$$

where $\hat{\mathbf{B}}_{NN}$ is the NN prediction for the Fourier transformed magnetic field. By the Plancherel theorem, this quantity is equivalent to the MSE in position space over the training data:

$$C = \sum_i \int d^3\mathbf{r} |\mathbf{B}(\mathbf{r}, t_i) - \mathbf{B}_{NN}(\mathbf{r}, t_i)|^2. \tag{15}$$

To evaluate the model performance at time step t we used the relative error of the magnetic field in position space:

$$\varepsilon(t) = \frac{\int d^3\mathbf{r} |\mathbf{B}(\mathbf{r}, t) - \mathbf{B}_{NN}(\mathbf{r}, t)|}{\int d^3\mathbf{r} |\mathbf{B}(\mathbf{r}, t)|} = \frac{\langle |\mathbf{B} - \mathbf{B}_{NN}| \rangle}{\langle |\mathbf{B}| \rangle}. \tag{16}$$

Here, $\langle \cdot \rangle$ denotes a spatial integral over the domain. The time averaged $\varepsilon(t)$ was used to evaluate a model over a data set.

Results

The U-Net performed best across all data sets—the training, validation and test data (see Table 2). In addition, as seen in Fig. 6, at each time step for each data set it either had the lowest error or was virtually tied. Remarkably, the U-Net was able to accomplish this with approximately half the training epochs of the smaller CNN's and a quarter that of CNN-large. Figure 7 shows predictions for B_x at $z = 0$ for the two models which performed best on the test data sets and how they compare to the true values. The results for B_y were similar, and thus aren't shown.

The small CNN networks struggled to capture the complexity of the training data. As seen in Fig. 7, while they had a qualitatively correct picture, they struggled on getting many of the finer details correct. The large CNN model was able to capture the training data better, but its predictions quickly deteriorated on the validation set as it got further away from the training data, eventually surpassing the error of CNN-small-4, likely due to overfitting of the larger network. The U-Net captured the training data well and although its error also increases on the validation set, as seen in Fig. 6, it does so at a much smaller rate than any of the CNN models, indicating it learned to generalize better.

All the models struggled the most at the beginning of the Gaussian Beam, where the initial beam was most different from what they were trained on. Of the CNN models, CNN-small-4 did the best job generalizing, though it still lagged behind the U-Net. These results reinforce that the U-Net is doing the best at generalizing.

Model	Description
CNN-small-4	CNN encoder–decoder, with 4 filters in the 8 × 8 × 8 latent space
CNN-small-6	CNN encoder–decoder, with 6 filters in the 8 × 8 × 8 latent space
U-Net	Encoder–decoder with skipped connection, with $F = 16$
CNN-large	The architecture is similar to CNN-small-4, but with more filters in the hidden layers

Table 1. Investigated architectures are arranged in descending order based on total number of parameters.

Model	Training error (%)	Validation error (%)	Test error (%)
CNN-small-4	8.2	13.3	12.6
CNN-small-6	7.9	13.5	17.0
U-Net	**4.5**	**6.7**	**8.6**
CNN-large	5.0	12.4	15.9

Table 2. Summary of results. Error over a data set is $\varepsilon(t)$, defined by Eq. (16), averaged over time. Bold indicates best performing model according to column metric.

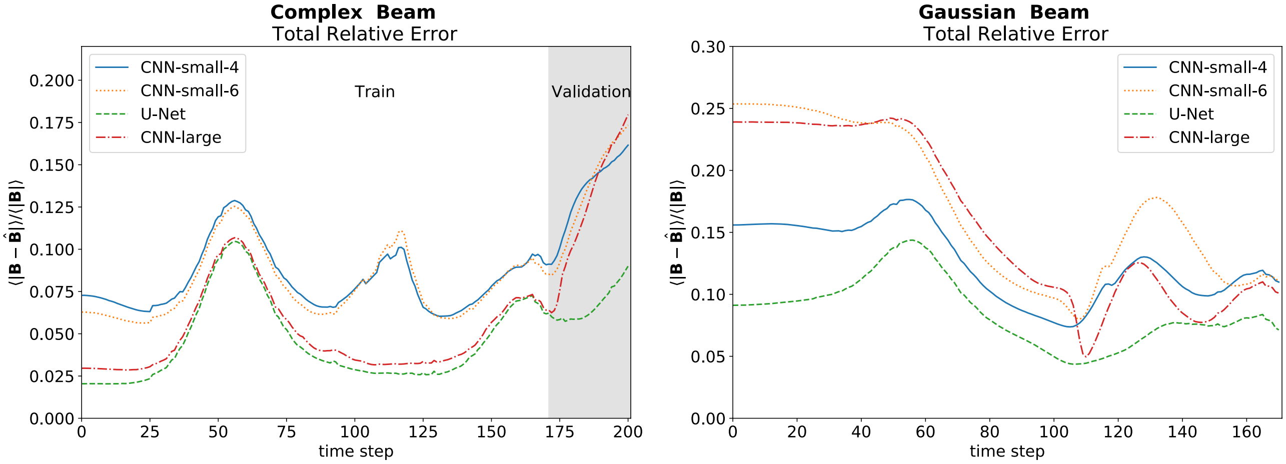

Figure 6. Relative Error of the models on both the complex beam and the Gaussian beam.

Transfer learning

Transfer learning is the process where a model trained on one set of data is retrained on another set. Since much of what was learned earlier can be helpful for the new data set, this cuts down on the training time and is quicker than if one started from scratch. It has been used in cases like Physics-Informed Neural Networks to learn beyond a single instance of a PDE[67].

The U-Net model took 380 training epochs. We took that trained model and retrained it on the first 85% of Gaussian Beam. In only 20 epochs, the model's relative error on the Gaussian beam dropped dramatically, and its performance was comparable to what it was on the complex beam (see Fig. 8). This demonstrates that our general approach can be useful for a wide range of new beam configurations which the model has not previously seen, so that it can be applied for the study of new different setups.

Improving generalization

To demonstrate the robustness and generalization capability of our approach when trained with more data, two new datasets were created, complex beam 2 and simple beam 2. Like the previous simulations, it consisted of an electron bunch entering a solenoid field, however now the initial distribution gradually dropped off in the z direction, as opposed to suddenly (see Figs. 3 and 4).

Using the pre-trained U-Net from earlier, the network was trained on both complex beam and complex beam 2. The network was tested on both the Gaussian Beam and Gaussian Beam 2, then its performance compared to the network trained on just the complex beam. The results can be seen in Fig. 9. For the Gaussian Beam, the average relative error of the new network on the Gaussian beam was 7.3%, while the network trained exclusively on the complex beam was 8.6%. Thus, the new network improved on predictions for the Gaussian beam with the inclusion of more data in the training set. For Gaussian Beam 2, the network trained on both complex beams consistently performed better, with an average relative error of 9.1% compared to 24.6%. Thus, we see the inclusion of more data led to an improvement in the network"s ability to generalize.

Discussion

One reason for the great success of the U-Net could be simply, as mentioned, the skipped connections help with the optimization. Another possibility is that, given that local J_z will correlate strongly with local $\mathbf{B}$, the network can start with a good first approximation by not altering the input that much. Indeed, one can compare, for example, J_z at the $z = 0$ mm slice for time step = 150 in the complex beam shown in Fig. 3 with the corresponding magnetic field in Fig. 7. The time of inputting $\hat{J}_{\perp,z}$ and outputting $\hat{A}_{\perp,z}$ for the entire test set was of the order of a few seconds. This is in contrast to the GPT simulation, which ran for $\sim$ a day. Of course, the GPT simulator is creating both the matter and electromagnetic fields, so this is not exactly a direct comparison. Nonetheless, it does illustrate some of the potential of the FoHMM-NO method to speed EM simulations greatly.

One way to improve our results would be exposing the model to many distinct simulations during the training phase. This would increase the duration of training, but our results in 5.2 suggests that this will make the model more accurate, robust and would generalize better. There are still some of Maxwell's equations that are not built in as hard constraints, namely Eq. (8). One possible way to add this as an inductive bias would be to include in the cost function a penalty term that enforces Eq. (8), as is done with physics informed neural networks[61]. Such terms have also been used to enforce the Lorenz gauge for PCNN's predicting electromagnetic potential fields[40] and enforcing the mangetohydrodynamics equations with physics informed neural operators[58]. Although this is not a hard constraint, during training it would guide the model to predict more physical solutions. As mentioned, the approach here can be expanded to include ρ and $\mathbf{J}$ in the predictions. Alternatively, it can be incorporated into a conventional simulator to speed up it up.

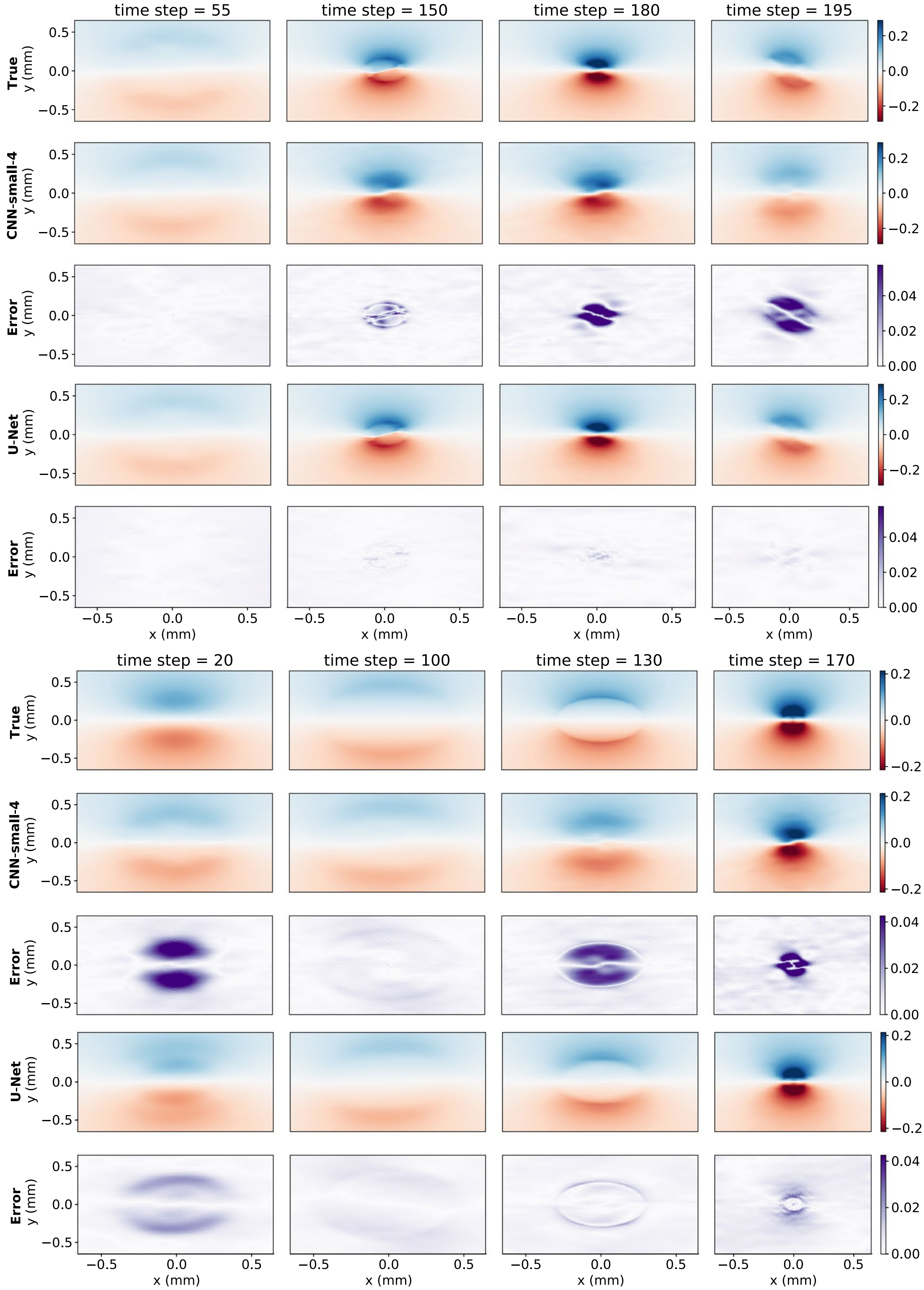

Figure 7. The top 5 rows show model predictions and absolute errors for the complex beam and the bottom 5 rows for the test Gaussian beam for B_x at the $z = 0$ mm slice at various time points. Only the 2 best performing models are shown. The scale of the absolute errors is zoomed in to 1/5 that of the true and model predictions.

Conclusions

Using Maxwell's equation in Fourier space and modeling EM fields via $\hat{A}_{\perp,z}$, we have developed a novel approach to applying machine learning to EM problems. This method has the advantage of incorporating several of

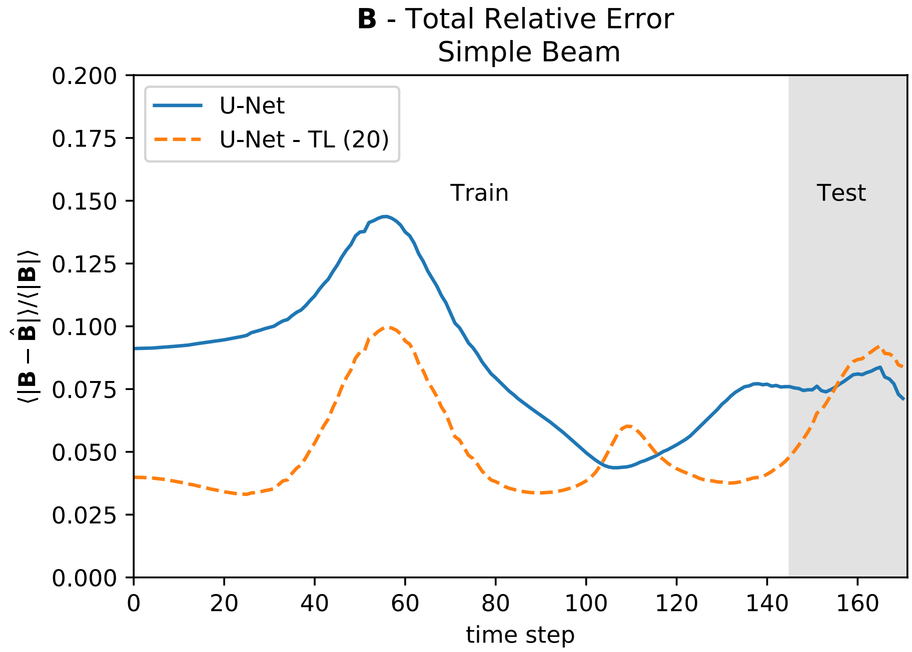

Figure 8. The original U-Net that was trained on part of the complex beam (U-Net) and the model after 20 training epochs on 85% of the Gaussian beam (U-Net-TL(20)).

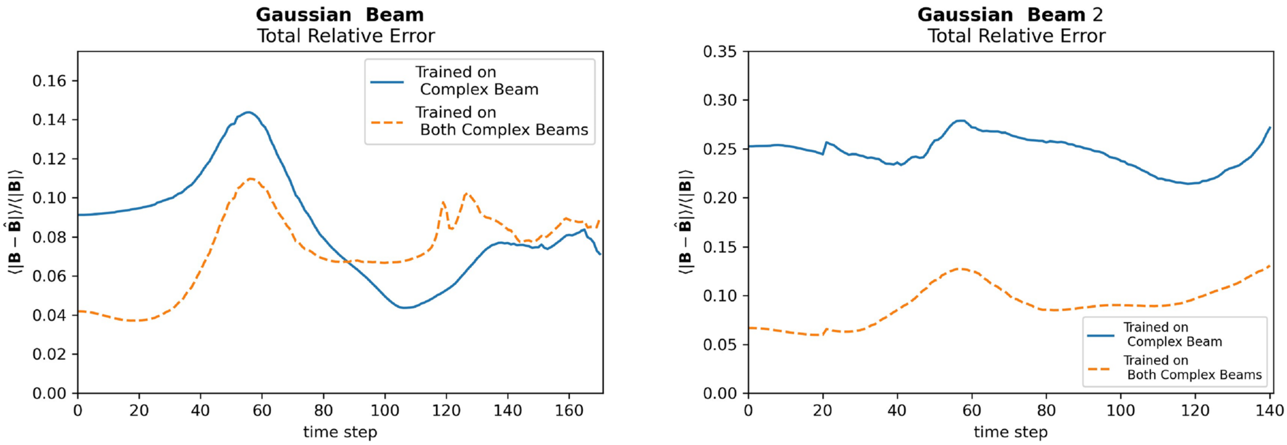

Figure 9. Relative Error of models on the on Gaussian beam and Gaussian beam 2.

Maxwell's equations as hard constraints, making the output more physically plausible. The FoHM-NO method can be beneficial in computational EM cases where speed is paramount. Here, we exchanged an error level of a few percent for a gain in speed of several orders of magnitude. We found that a U-Net architecture did best both at recreating the fields and generalizing beyond the training data. Given the exact and quantitative nature of physics constraints, domain knowledge can be incorporated into machine learning tools in unique ways. Integrating physics and machine learning in a synergistic way has great potential, but the approaches are still in their infancy. With the advent of powerful new machine learning algorithms, physicists have the opportunity to explore how these new tools can be used to complement or augment more traditional methods to solve previously intractable problems.

Data availibility

Datasets used and analyzed during the current study are available from the corresponding author upon reasonable request.

Appendix A: k = 0 case

Taking the Fourier transform of Gauss's Law for the electric field gives:

$$\frac{1}{(2\pi)^{3/2}}\int d^3\mathbf{r}e^{-i\mathbf{k}\cdot\mathbf{r}}\nabla\cdot\mathbf{E}(\mathbf{r},t)=\frac{\hat{\rho}(\mathbf{k},t)}{\varepsilon_0}. \tag{17}$$

Integrating the left-hand side by parts:

$$\lim_{R\to\infty}\frac{1}{(2\pi)^{3/2}}\int_{S^2(R)}d\mathbf{S}\cdot\mathbf{E}(\mathbf{r},t)e^{-i\mathbf{k}\cdot\mathbf{r}}+i\mathbf{k}\cdot\hat{\mathbf{E}}(\mathbf{k},t)=\frac{\hat{\rho}(\mathbf{k},t)}{\varepsilon_0}, \tag{18}$$

where $S^2(R)$ means a sphere of radius R.

For $\mathbf{k}=\mathbf{0}$, $\hat{\rho}(\mathbf{0},t)=\frac{Q_{tot}}{(2\pi)^{3/2}}$. The boundary term on the left-hand side of Eq. (18) is a flux, the second term disappears and we get the integral form of Gauss's law for a large sphere:

$$\lim_{R\to\infty}\int_{S^2(R)}d\mathbf{S}\cdot\mathbf{E}(\mathbf{r},t)=\frac{Q_{tot}}{\varepsilon_0}. \tag{19}$$

For $\mathbf{k}\neq\mathbf{0}$, the phase inside the boundary term on the left-hand side of Eq. (18) oscillates rapidly because of the large R. The boundary term thus disappears and we get the expected expression:

$$i\mathbf{k}\cdot\hat{\mathbf{E}}(\mathbf{k},t)=\frac{\hat{\rho}(\mathbf{k},t)}{\varepsilon_0}. \tag{20}$$

Appendix B: Longitudinal Ampère–Maxwell

The longitudinal component of Ampère–Maxwell's equation in Fourier space follows from the continuity equation and Gauss"s law for **E**.

$$\begin{aligned}
0&=\frac{\partial\hat{\rho}(\mathbf{k},t)}{\partial t}+i\mathbf{k}\cdot\hat{\mathbf{J}}(\mathbf{k},t),\\
0&=\frac{\partial\left(\varepsilon_0 i\mathbf{k}\cdot\hat{\mathbf{E}}(\mathbf{k},t)\right)}{\partial t}+ik\hat{J}_{\parallel}(\mathbf{k},t),\\
0&=\varepsilon_0 ik\frac{\partial\hat{E}_{\parallel}(\mathbf{k},t)}{\partial t}+ik\hat{J}_{\parallel}(\mathbf{k},t),\\
0&=\varepsilon_0\mu_0\frac{\partial\hat{E}_{\parallel}(\mathbf{k},t)}{\partial t}+\mu_0\hat{J}_{\parallel}(\mathbf{k},t).
\end{aligned} \tag{21}$$

Since $\left(i\mathbf{k}\times\hat{\mathbf{B}}(\mathbf{k},t)\right)_{\parallel}=0$ then:

$$\left(i\mathbf{k}\times\hat{\mathbf{B}}(\mathbf{k},t)\right)_{\parallel}=\varepsilon_0\mu_0\frac{\partial\hat{E}_{\parallel}(\mathbf{k},t)}{\partial t}+\mu_0\hat{J}_{\parallel}(\mathbf{k},t). \tag{22}$$

Received: 19 January 2024; Accepted: 21 June 2024
Published online: 26 June 2024

References

1. De Lorenci, V., Klippert, R., Novello, M. & Salim, J. Nonlinear electrodynamics and FRW cosmology. *Phys. Rev. D* **65**, 063501 (2002).
2. Novello, M., Bergliaffa, S. P. & Salim, J. Nonlinear electrodynamics and the acceleration of the universe. *Phys. Rev. D* **69**, 127301 (2004).
3. Kruglov, S. Universe acceleration and nonlinear electrodynamics. *Phys. Rev. D* **92**, 123523 (2015).
4. Bandos, I., Lechner, K., Sorokin, D. & Townsend, P. K. Nonlinear duality-invariant conformal extension of Maxwell's equations. *Phys. Rev. D* **102**, 121703 (2020).
5. Li, J. *et al.* Addressable metasurfaces for dynamic holography and optical information encryption. *Sci. Adv.* **4**, eaar6768 (2018).
6. Evoli, C., Blasi, P., Morlino, G. & Aloisio, R. Origin of the cosmic ray galactic halo driven by advected turbulence and self-generated waves. *Phys. Rev. Lett.* **121**, 021102. https://doi.org/10.1103/PhysRevLett.121.021102 (2018).
7. Nättilä, J. & Beloborodov, A. M. Heating of magnetically dominated plasma by Alfvén-wave turbulence. *Phys. Rev. Lett.* **128**, 075101. https://doi.org/10.1103/PhysRevLett.128.075101 (2022).
8. Luo, X. *et al.* Metasurface-enabled on-chip multiplexed diffractive neural networks in the visible. *Light Sci. Appl.* **11**, 1–11 (2022).
9. Aieta, F., Kats, M. A., Genevet, P. & Capasso, F. Multiwavelength achromatic metasurfaces by dispersive phase compensation. *Science* **347**, 1342–1345 (2015).
10. Miao, J., Ishikawa, T., Robinson, I. K. & Murnane, M. M. Beyond crystallography: Diffractive imaging using coherent X-ray light sources. *Science* **348**, 530–535 (2015).
11. Bacchini, F., Pucci, F., Malara, F. & Lapenta, G. Kinetic heating by Alfvén waves in magnetic shears. *Phys. Rev. Lett.* **128**, 025101. https://doi.org/10.1103/PhysRevLett.128.025101 (2022).
12. Chaston, C., Carlson, C., Peria, W., Ergun, R. & McFadden, J. Fast observations of inertial Alfvén waves in the dayside aurora. *Geophys. Res. Lett.* **26**, 647–650. https://doi.org/10.1029/1998GL900246 (1999).
13. Deng, J. *et al.* Correlative 3D X-ray fluorescence and ptychographic tomography of frozen-hydrated green algae. *Sci. Adv.* **4**, eaau4548 (2018).
14. Gurram, H., Egedal, J. & Daughton, W. Shear Alfvén waves driven by magnetic reconnection as an energy source for the aurora borealis. *Geophys. Res. Lett.* **48**, e2021GL094201. https://doi.org/10.1029/2021GL094201 (2021).
15. Tang, S. *et al.* Stabilization of Alfvén eigenmodes in DIII-D via controlled energetic ion density ramp and validation of theory and simulations. *Phys. Rev. Lett.* **126**, 155001. https://doi.org/10.1103/PhysRevLett.126.155001 (2021).
16. Tang, H. *et al.* Stable and scalable multistage terahertz-driven particle accelerator. *Phys. Rev. Lett.* **127**, 074801. https://doi.org/10.1103/PhysRevLett.127.074801 (2021).

17. Zhang, Z. & Satpathy, S. Electromagnetic wave propagation in periodic structures: Bloch wave solution of Maxwell's equations. *Phys. Rev. Lett.* **65**, 2650. https://doi.org/10.1103/PhysRevLett.65.2650 (1990).
18. Zachariasen, W. A general theory of X-ray diffraction in crystals. *Acta Crystallograph.* **23**, 558–564. https://doi.org/10.1107/S0365110X67003202 (1967).
19. Monico, L. *et al.* Probing the chemistry of CdS paints in the scream by in situ noninvasive spectroscopies and synchrotron radiation X-ray techniques. *Sci. Adv.* **6**, eaay3514. https://doi.org/10.1126/sciadv.aay3514 (2020).
20. Mehrabi, P. *et al.* Serial femtosecond and serial synchrotron crystallography can yield data of equivalent quality: A systematic comparison. *Sci. Adv.* **7**, eabf1380. https://doi.org/10.1126/sciadv.abf1380 (2021).
21. Hartemann, F. & Luhmann, N. Jr. Classical electrodynamical derivation of the radiation damping force. *Phys. Rev. Lett.* **74**, 1107. https://doi.org/10.1103/PhysRevLett.74.1107 (1995).
22. Grošelj, D., Mallet, A., Loureiro, N. F. & Jenko, F. Fully kinetic simulation of 3D kinetic Alfvén turbulence. *Phys. Rev. Lett.* **120**, 105101. https://doi.org/10.1103/PhysRevLett.120.105101 (2018).
23. Franchetti, G., Hofmann, I. & Jeon, D. Anisotropic free-energy limit of halos in high-intensity accelerators. *Phys. Rev. Lett.* **88**, 254802. https://doi.org/10.1103/PhysRevLett.88.254802 (2002).
24. Franchetti, G., Hofmann, I. & Aslaninejad, M. Collective emittance exchange with linear space charge forces and linear coupling. *Phys. Rev. Lett.* **94**, 194801. https://doi.org/10.1103/PhysRevLett.94.194801 (2005).
25. Cai, Y. Coherent synchrotron radiation by electrons moving on circular orbits. *Phys. Rev. Accel. Beams* **20**, 064402. https://doi.org/10.1103/PhysRevAccelBeams.20.064402 (2017).
26. Zagorodnov, I., Tomin, S., Chen, Y. & Brinker, F. Experimental validation of collective effects modeling at injector section of X-ray free-electron laser. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip.* **995**, 165111. https://doi.org/10.1016/j.nima.2021.165111 (2021).
27. Emma, P. *et al.* First lasing and operation of an Rangström-wavelength free-electron laser. *Nat. Photon.* **4**, 641 (2010).
28. Milne, C. *et al.* Swissfel: The Swiss X-ray free electron laser. *Appl. Sci.* **7**, 720 (2017).
29. Decking, W. *et al.* A MHz-repetition-rate hard X-ray free-electron laser driven by a superconducting linear accelerator. *Nat. Photon.* **14**, 391–397 (2020).
30. Gessner, S. *et al.* Demonstration of a positron beam-driven hollow channel plasma wakefield accelerator. *Nat. Commun.* **7**, 1–6 (2016).
31. Yakimenko, V. *et al.* Facet-ii facility for advanced accelerator experimental tests. *Phys. Rev. Accel. Beams* **22**, 101301 (2019).
32. Chen, Y., Zagorodnov, I. & Dohlus, M. Beam dynamics of realistic bunches at the injector section of the European X-ray free-electron laser. *Phys. Rev. Accel. Beams* **23**, 044201. https://doi.org/10.1103/PhysRevAccelBeams.23.044201 (2020).
33. Dohlus, M., Limberg, T. *et al.* Csrtrack: Faster calculation of 3D CSR effects. In *Proceedings of the 2004 FEL Conference*, 18–21 (2004).
34. Scheinker, A., Cropp, F., Paiagua, S. & Filippetto, D. An adaptive approach to machine learning for compact particle accelerators. *Sci. Rep.* **11**, 1–11 (2021).
35. Coskun, U. H., Sel, B. & Plaster, B. Magnetic field mapping of inaccessible regions using physics-informed neural networks. *Sci. Rep.* **12**, 12858 (2022).
36. Li, S., Dee, P. M., Khatami, E. & Johnston, S. Accelerating lattice quantum Monte Carlo simulations using artificial neural networks: Application to the Holstein model. *Phys. Rev. B* **100**, 020302 (2019).
37. Kim, I. *et al.* Simulator acceleration and inverse design of fin field-effect transistors using machine learning. *Sci. Rep.* **12**, 1140 (2022).
38. Park, S., Kwak, W. & Lee, H. K. Accelerated spin dynamics using deep learning corrections. *Sci. Rep.* **10**, 13772 (2020).
39. Wigley, P. B. *et al.* Fast machine-learning online optimization of ultra-cold-atom experiments. *Sci. Rep.* **6**, 25890 (2016).
40. Scheinker, A. & Pokharel, R. Physics-constrained 3D convolutional neural networks for electrodynamics. *APL Mach. Learn.* **1**, 026109 (2023).
41. Cohen-Tannoudji, C., Dupont-Roc, J. & Grynberg, G. *Atom-Photon Interactions: Basic Processes and Applications* (Wiley, 1998).
42. Jackson, J. D. *Classical Electrodynamics* (Wiley, 1999).
43. Greydanus, S., Dzamba, M. & Yosinski, J. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, vol. 32 (2019).
44. Mohan, A. T., Lubbers, N., Chertkov, M. & Livescu, D. Embedding hard physical constraints in neural network coarse-graining of three-dimensional turbulence. *Phys. Rev. Fluids* **8**, 014604 (2023).
45. Lin, B., Mao, Z., Wang, Z. & Karniadakis, G. E. Operator learning enhanced physics-informed neural networks for solving partial differential equations characterized by sharp solutions. arXiv preprint arXiv:2310.19590 (2023).
46. Adler, J. & Öktem, O. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.* **33**, 124007 (2017).
47. Zhao, Q., Lindell, D. B. & Wetzstein, G. Learning to solve PDE-constrained inverse problems with graph networks (2022).
48. Vukics, A., Kónya, G. & Domokos, P. The gauge-invariant Lagrangian, the Power–Zienau–Woolley picture, and the choices of field momenta in nonrelativistic quantum electrodynamics. *Sci. Rep.* **11**, 16337 (2021).
49. Gustin, C., Franke, S. & Hughes, S. Gauge-invariant theory of truncated quantum light-matter interactions in arbitrary media. *Phys. Rev. A* **107**, 013722 (2023).
50. Chen, X.-S., Lü, X.-F., Sun, W.-M., Wang, F. & Goldman, T. Spin and orbital angular momentum in gauge theories: Nucleon spin structure and multipole radiation revisited. *Phys. Rev. Lett.* **100**, 232002 (2008).
51. Van Der Geer, S., Luiten, O., De Loos, M., Pöplau, G. & Van Rienen, U. 3D space-charge model for GPT simulations of high brightness electron bunches. *Inst. Phys. Conf. Ser.* **175**, 101 (2005).
52. Brynes, A. *et al.* Beyond the limits of 1D coherent synchrotron radiation. *New J. Phys.* **20**, 073035. https://doi.org/10.1088/1367-2630/aad21d (2018).
53. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
54. Park, S., Yun, C., Lee, J. & Shin, J. Minimum width for universal approximation. arXiv preprint arXiv:2006.08859 (2020).
55. Chen, T. & Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.* **6**, 911–917 (1995).
56. Lu, L., Jin, P., Pang, G., Zhang, Z. & Karniadakis, G. E. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **3**, 218–229. https://doi.org/10.1038/s42256-021-00302-5 (2021).
57. Li, Z. *et al.* Fourier neural operator for parametric partial differential equations. arXiv preprint arXiv:2010.08895 (2020).
58. Rosofsky, S. G. & Huerta, E. Magnetohydrodynamics with physics informed neural operators. arXiv preprint arXiv:2302.08332 (2023).
59. Bormanis, A., Leon, C. A. & Scheinker, A. Solving the Orszag–Tang vortex magnetohydrodynamics problem with physics-constrained convolutional neural networks. *Phys. Plasmas* **31**, 012101 (2024).
60. Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440. https://doi.org/10.1038/s42254-021-00314-5 (2021).
61. Raissi, M., Perdikaris, P. & Karniadakis, G. E. 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).

62. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, 234–241 (Springer, 2015).
63. Rahman, M. A., Ross, Z. E. & Azizzadenesheli, K. U-shaped neural operators (U-NO). *Trans. Mach. Learn. Res.* **2023**, 1–17 (2023).
64. Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A. & Benson, S. M. U-FNO–An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Adv. Water Resour.* **163**, 104180 (2022).
65. Abadi, M. *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org (2015).
66. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
67. Desai, S., Mattheakis, M., Joy, H., Protopapas, P. & Roberts, S. One-shot transfer learning of physics-informed neural networks. arXiv preprint arXiv:2110.11286 (2021).

Acknowledgements

This work was supported by the U.S. Department of Energy (DOE), Office of Science High Energy Physics contract number 89233218CNA000001 and the Los Alamos National Laboratory LDRD Program Directed Research (DR) project 20220074DR.

Author contributions

C.L.: Conceptualization (equal); data curation (equal); formal analysis (equal); investigation (equal); writing—original draft (equal); writing—review and editing (equal). A.S.: Conceptualization (equal); data curation (equal); formal analysis (equal); funding acquisition (lead); investigation (equal); methodology (lead); supervision (equal); writing—original draft (supporting); writing—review and editing (equal)

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to C.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.