Physics Constraint Machine Learning for Relativistic, Charged Particle Beams

Christopher Leon

Feb. 2nd, 2024

Goal

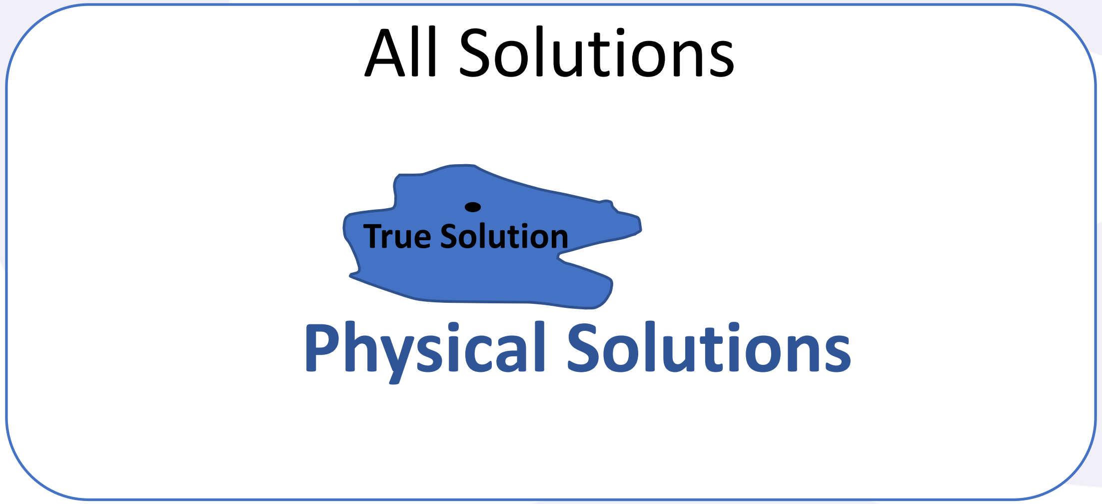
- Use **ML to get speed up over conventional simulations**

- **Incorporate physics into ML tools** for better results and generalization

- Beams : $\rho, \boldsymbol{J} \rightarrow \mathbf{E}, \mathbf{B}$

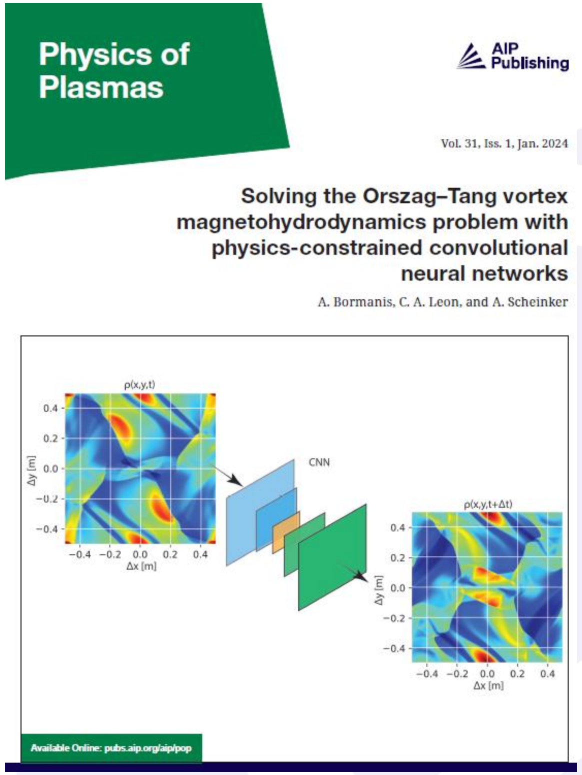
Tool: **Physics Constraint Convolutional Neural Networks**

Created with DALL·E



Previous Work

- Used for magnetohydrodynamics


Physics of Plasmas

AIP Publishing

Vol. 31, Iss. 1, Jan. 2024

Solving the Orszag–Tang vortex magnetohydrodynamics problem with physics-constrained convolutional neural networks

A. Bormanis, C. A. Leon, and A. Scheinker

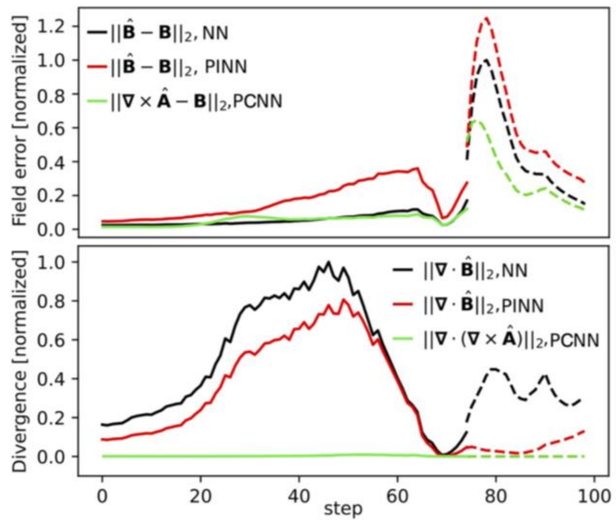
Available Online: pubs.aip.org/aip/pop

Constraint	Implementation	Hard/Soft
Divergence-free	$\mathbf{B}(\mathbf{r},t) = \nabla_{\perp} A(\mathbf{r},t)$	Hard
Translation Equivariance	CNN Architecture	Hard
Non-negativity (e.g., $\rho \geq 0$)	Final Layer ReLU	Hard
Periodic BC's	Padding	Soft
Partial Differential Equation	Term in Cost Function	Soft

- NN: $\boldsymbol{J} \rightarrow \boldsymbol{B}$

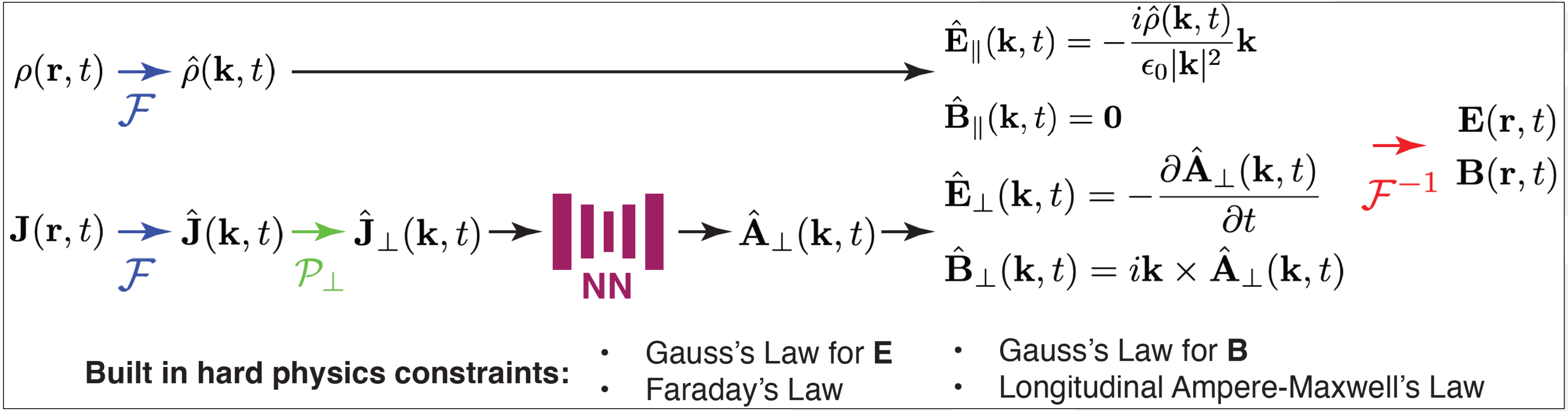
- PCNN: $\boldsymbol{J} \rightarrow \boldsymbol{A} \Rightarrow \boldsymbol{\nabla} \times \boldsymbol{A} = \boldsymbol{B}$

$\Rightarrow \nabla \cdot (\nabla \times \boldsymbol{A}) = 0$ **(Hard Constraint)**



Scheinker, A., and Pokharel, R. *APL Machine Learning* 1.2 (2023).

FoHM-NO Framework

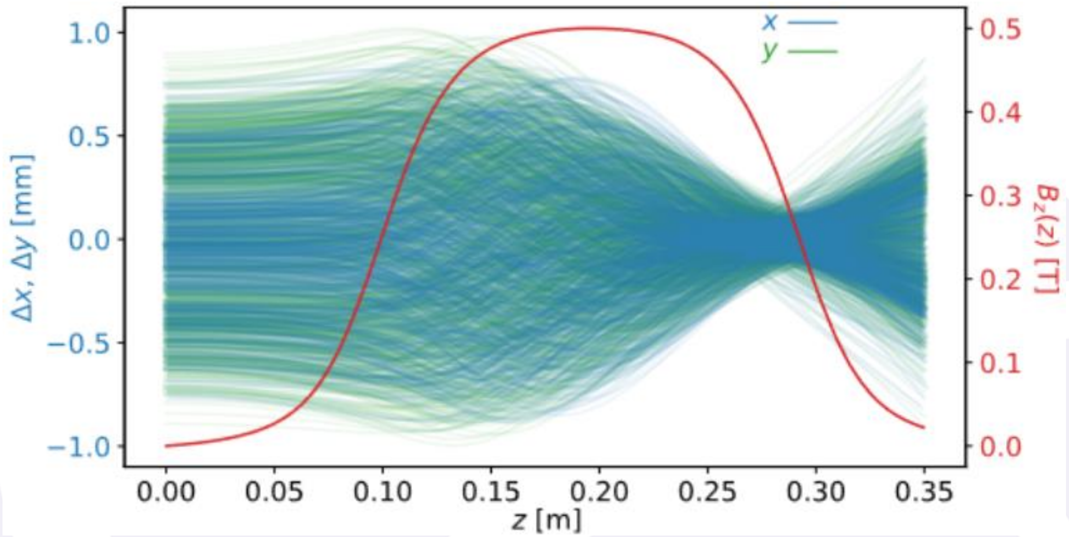


- Neural Network ⇒ **solution operator**: $\hat{A}_{\perp} = O\left[\hat{J}_{\perp}\right]$

$$\frac{1}{c^2}\frac{\partial^2 \hat{A}_{\perp}}{\partial t^2} + |k|^2 \hat{A}_{\perp} = \hat{J}_{\perp}$$

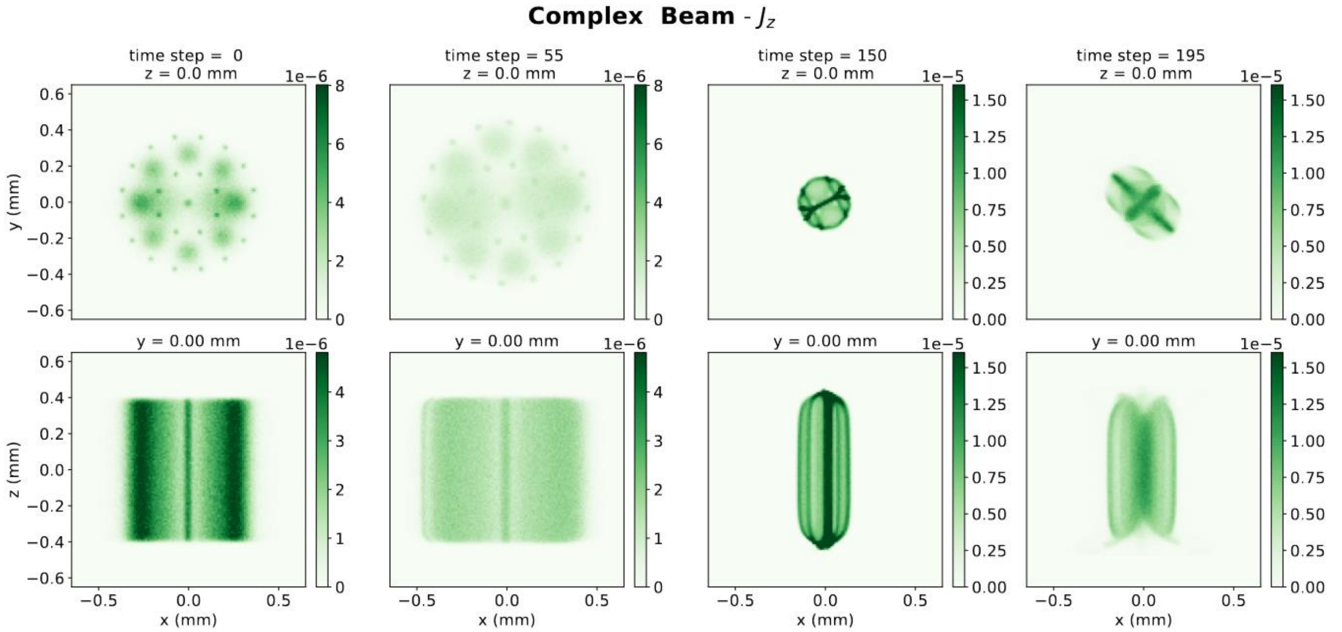
Data – Conventional Simulation

- Used General Particle Tracker (GPT)
- 2nC of e^{-}'s of 5.6 MeV entering solenoid **B** field



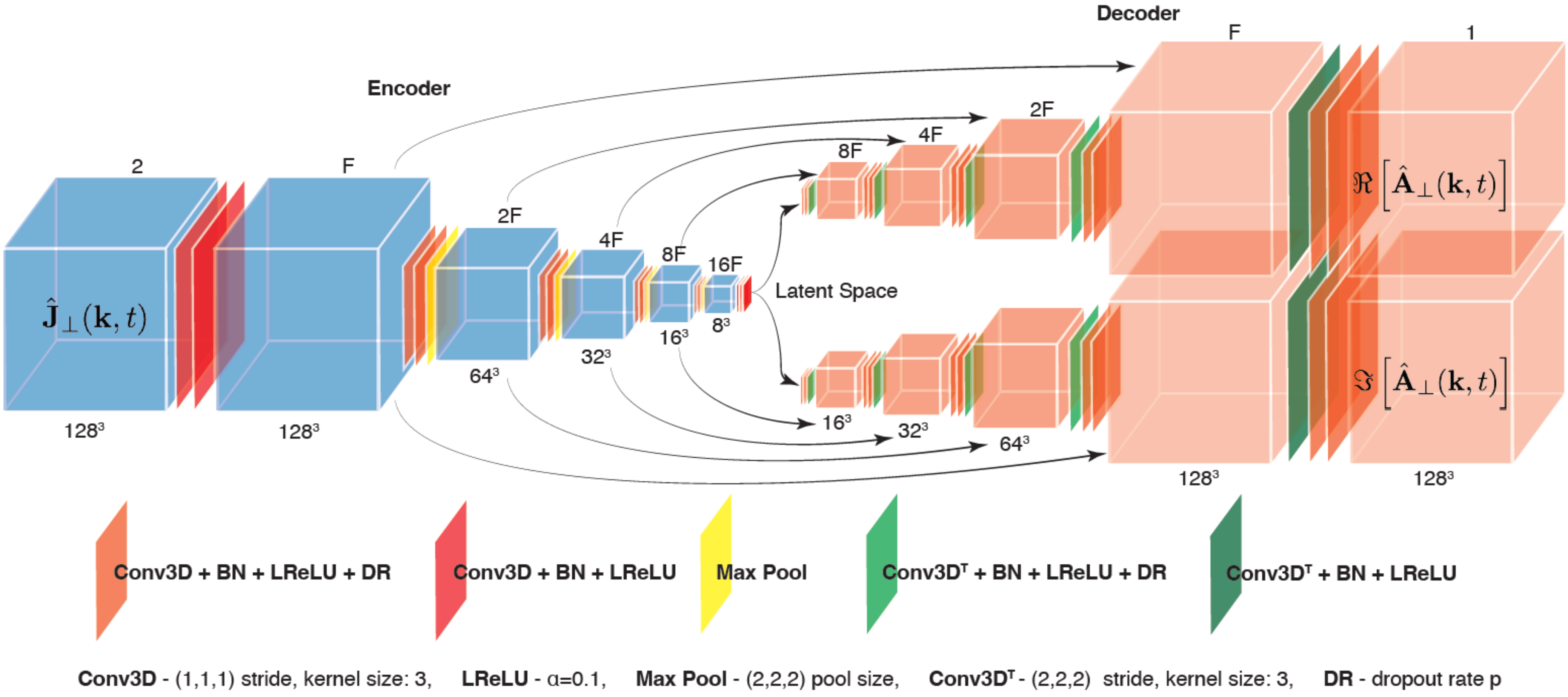
Two Beams:

- **Complex Beam** (Training 85% + Validation 15%)
- **Gaussian Beam** (Test)

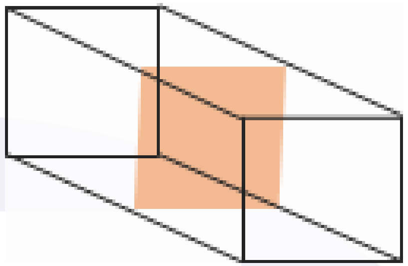


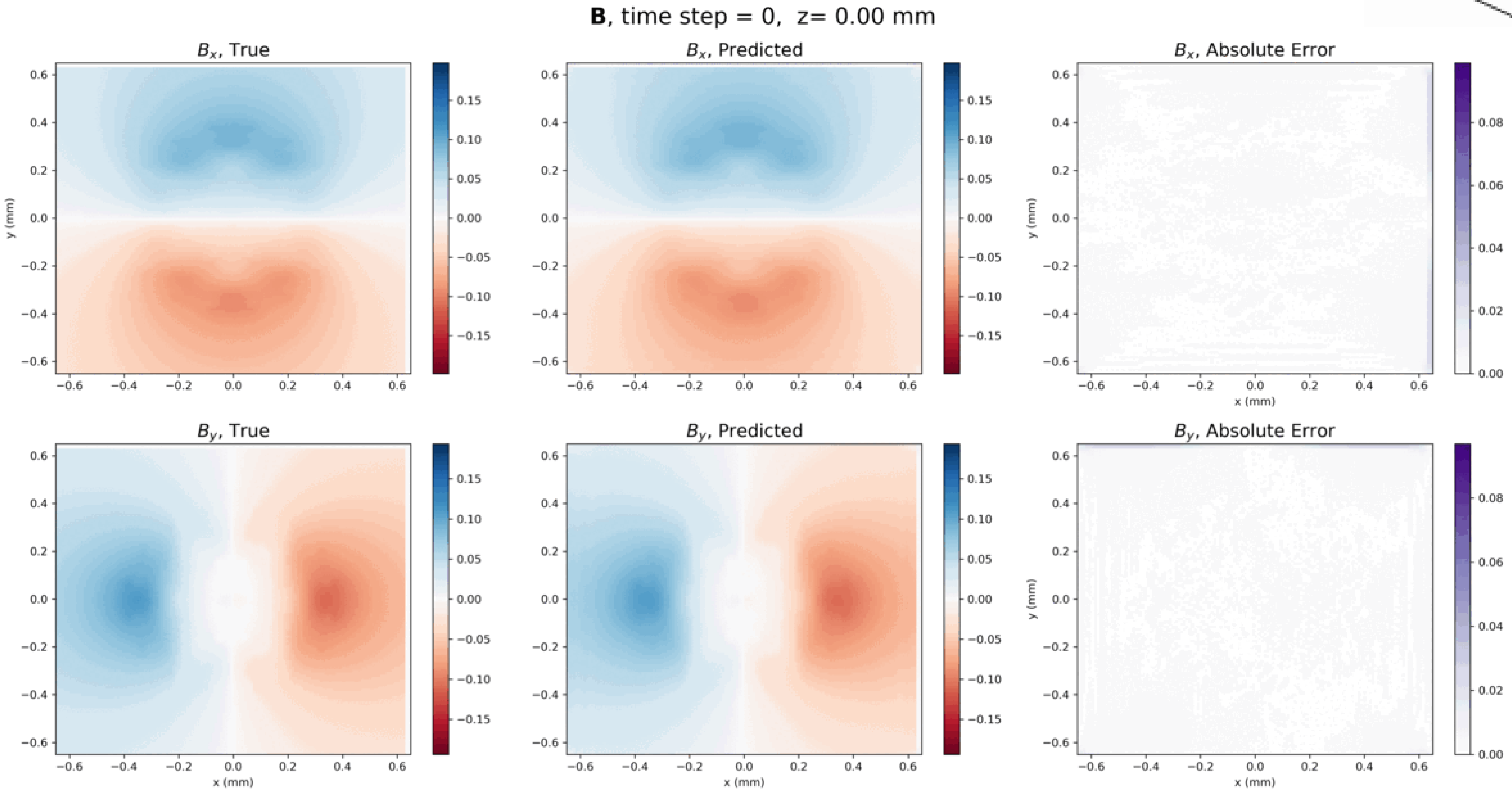
Best Model: U-Net

- Add skipped connections ⇒ Concatenation

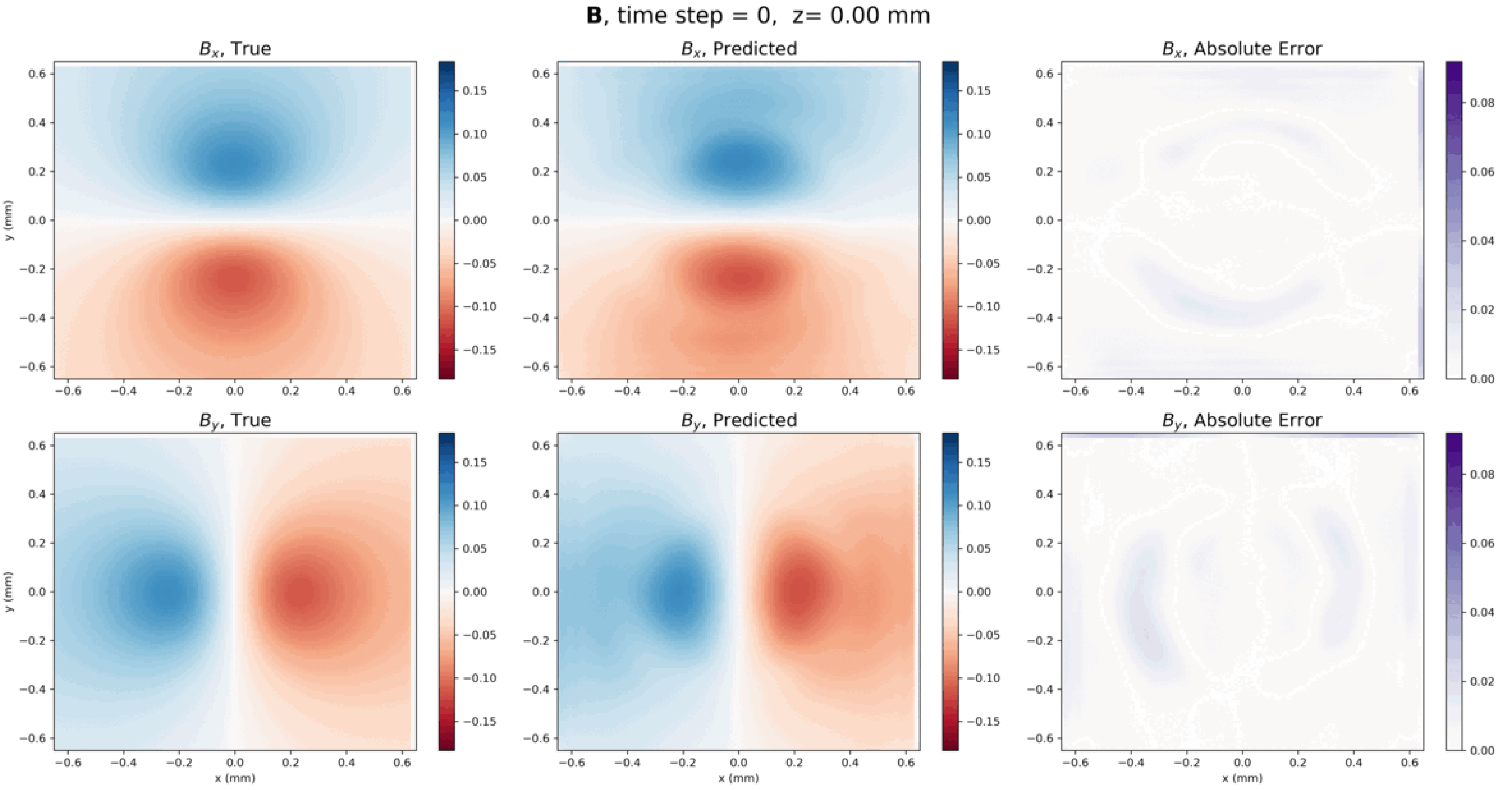


U-Net: Complex Beam





U-Net: Gaussian Beam



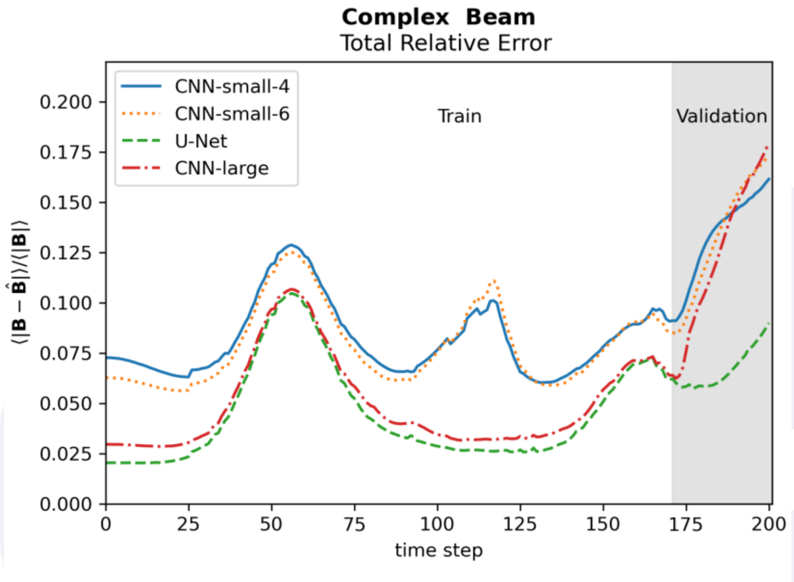
Results – Gaussian Beam

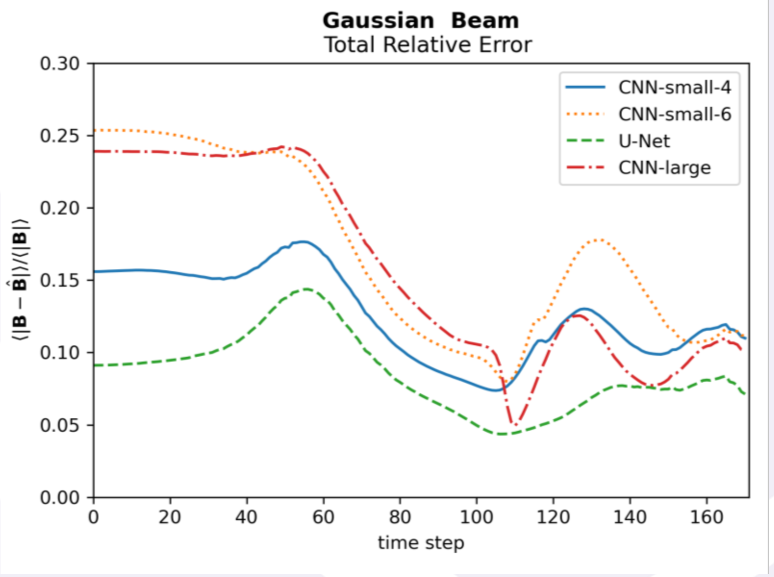
- U-Net: perform bests

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

Time for Gaussian Beam

- GPT simulation: ~ **1 day**
- U-Net ~ **few seconds**

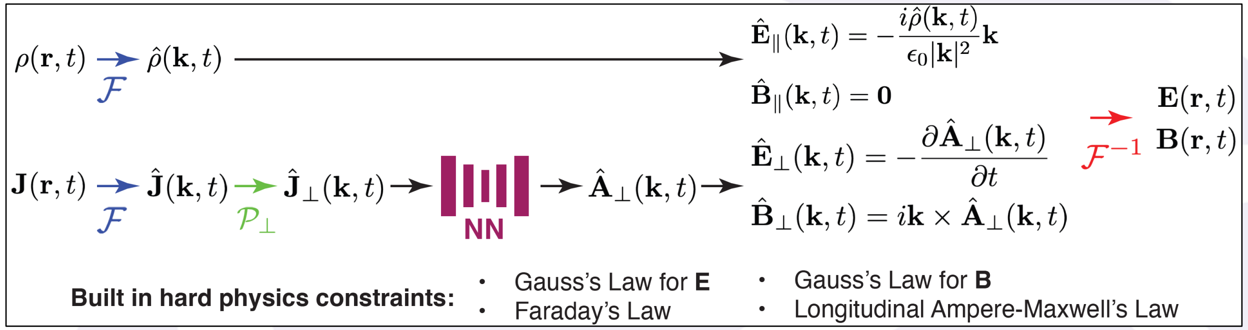




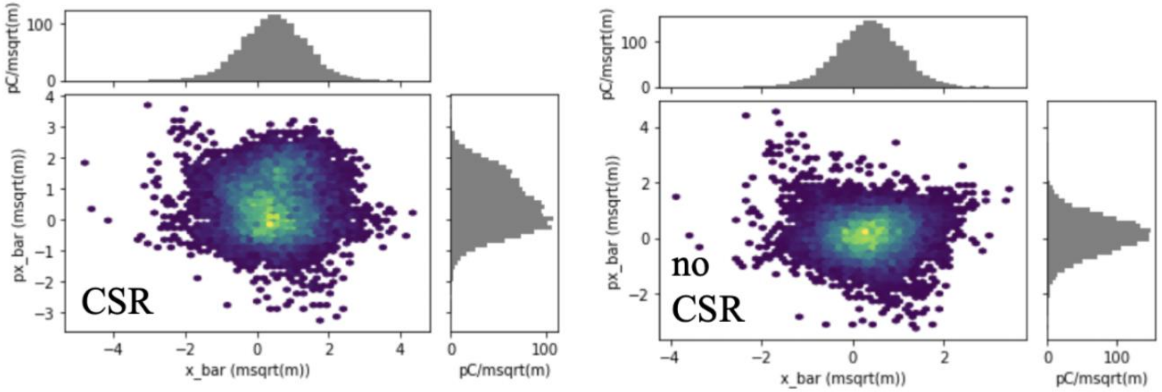
CL, Scheinker, A
Preprint: Research Square
https://doi.org/10.21203/rs.3.rs-3879834/v1

In the Works

- Including **Uncertainty Quantification** to FoHM-NO



- Applying to **Coherent Synchrotron Radiation** (IPAC' 24 poster)



Edelen, A. and Mayes, C.
arXiv: 2203.07542

Funded by LANL-LDRD

Backup: Universal Operator Approximation Theorem

(Informal) A neural network can approximate any operator (Chen and Chen, 1995)

- Note: architecture is not necessarily a CNN encoder-decoder
- Operator can offer family of solutions to PDE's

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 6, NO. 4, JULY 1995 911

Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems

Tianping Chen and Hong Chen

***Abstract*— The purpose of this paper is to investigate neural network capability systematically. The main results are: 1) every Tauber–Wiener function is qualified as an activation function in the hidden layer of a three-layered neural network, 2) for a continuous function in $S'(R^1)$ to be a Tauber–Wiener function, the necessary and sufficient condition is that it is not a polynomial, 3) the capability of approximating nonlinear functionals defined on some compact set of a Banach space and nonlinear operators has been shown, which implies that 4) we show the possibility by neural computation to approximate the output as a whole (not at a fixed point) of a dynamical system, thus identifying the system.**

Mhaskar and Micchelli [11] showed that under some restriction on the amplitude of a continuous function near infinity, any nonpolynomial function is qualified to be an activation function.

It is clear that all the aforementioned works are concerned with approximation to a continuous function defined on a compact set in $\boldsymbol{R}^n$ (a space of finite dimensions). In engineering problems such as computing the output of dynamic systems or designing neural system identifiers, however, we often encounter the problem of approximating nonlinear functionals

FOURCASTNET: A GLOBAL DATA-DRIVEN HIGH-RESOLUTION WEATHER MODEL USING ADAPTIVE FOURIER NEURAL OPERATORS

Backup Slide: Solution Operator

Example: Poisson's equation

$$\nabla^2 \phi(\boldsymbol{r}) = \frac{\rho(\boldsymbol{r})}{\epsilon_0}$$

Solution:

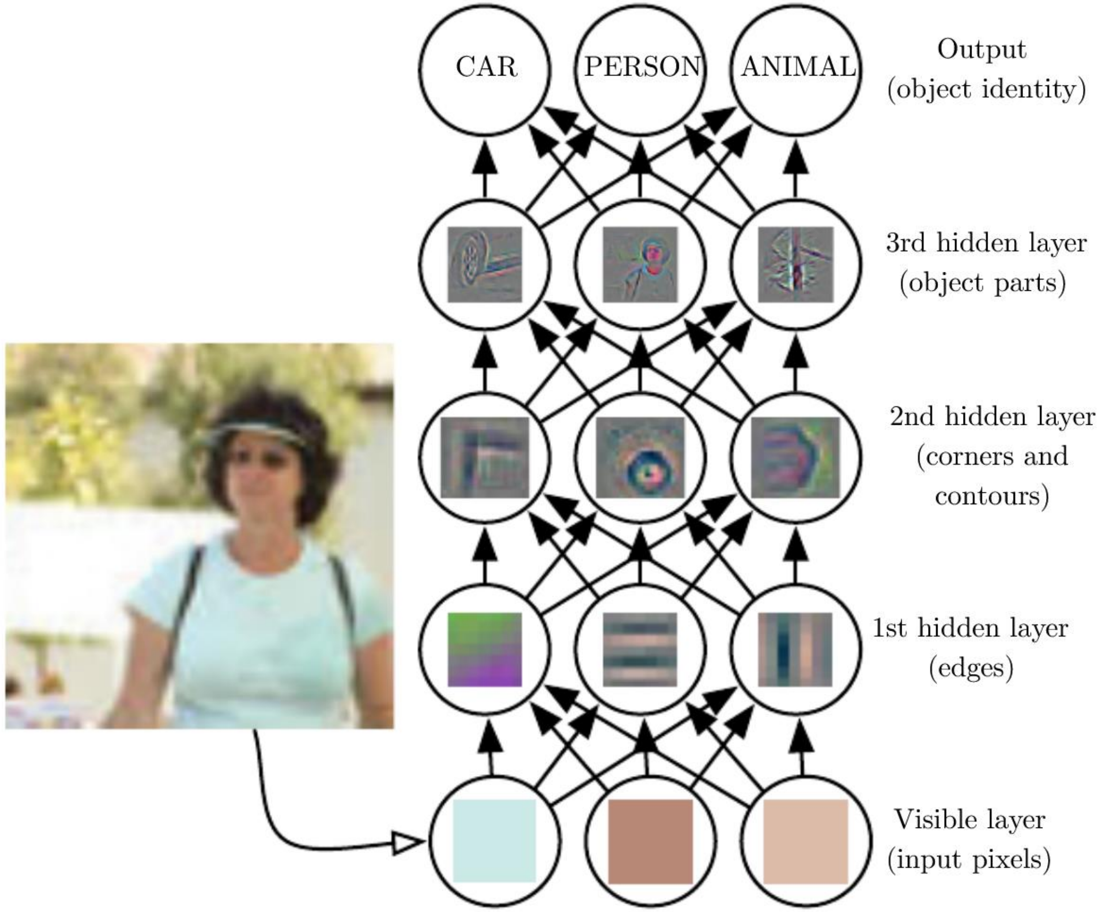
$$\phi(\boldsymbol{r}) = \frac{-1}{4\pi\epsilon_0} \int d^3\boldsymbol{r}' \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|}$$

Solution operator

$$\rho \to \phi$$

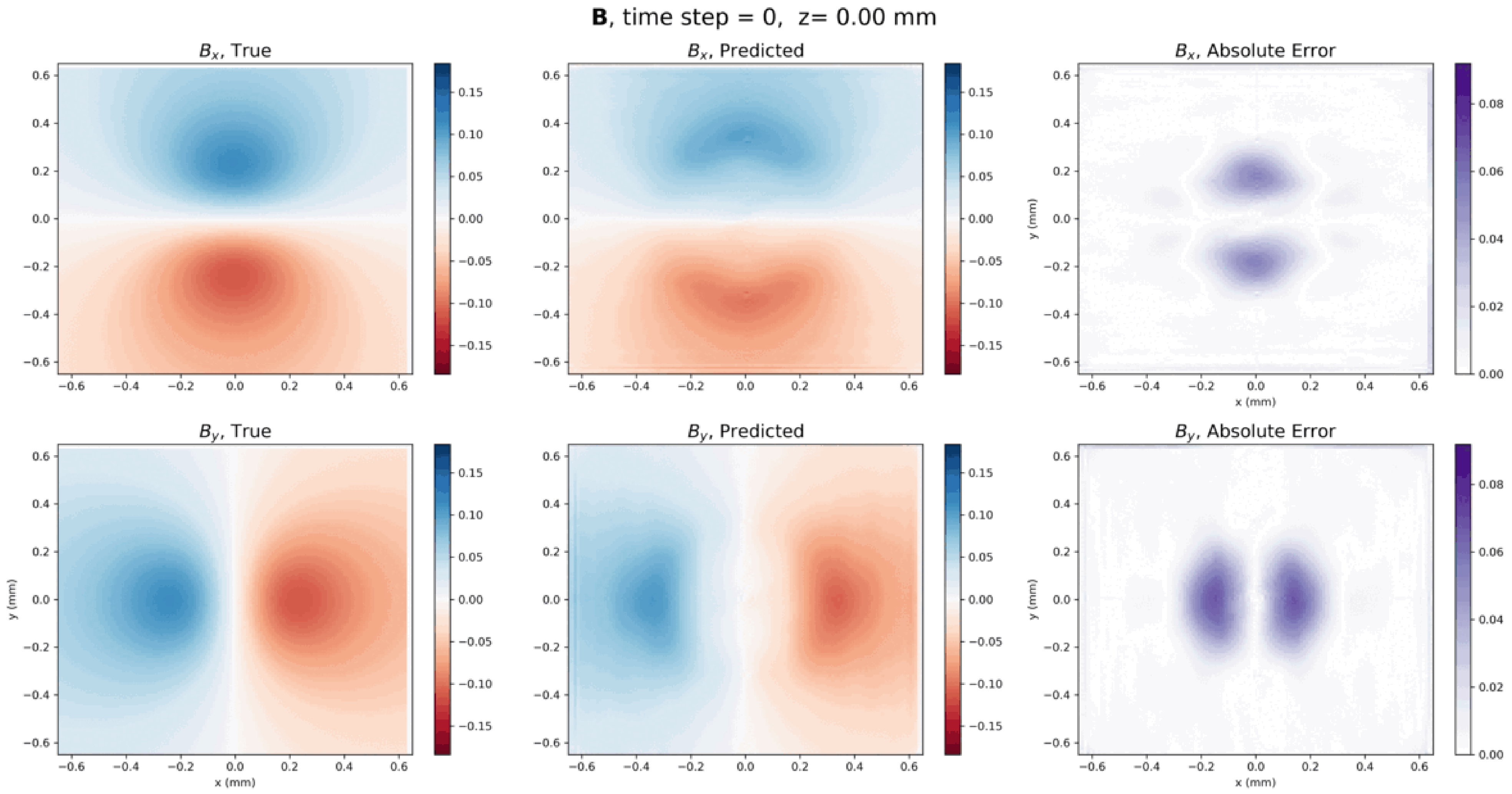
Backup: Abstraction Through Depth

- Build complexity from ground up
- First layers: lower-level features
- Later layers: high level features

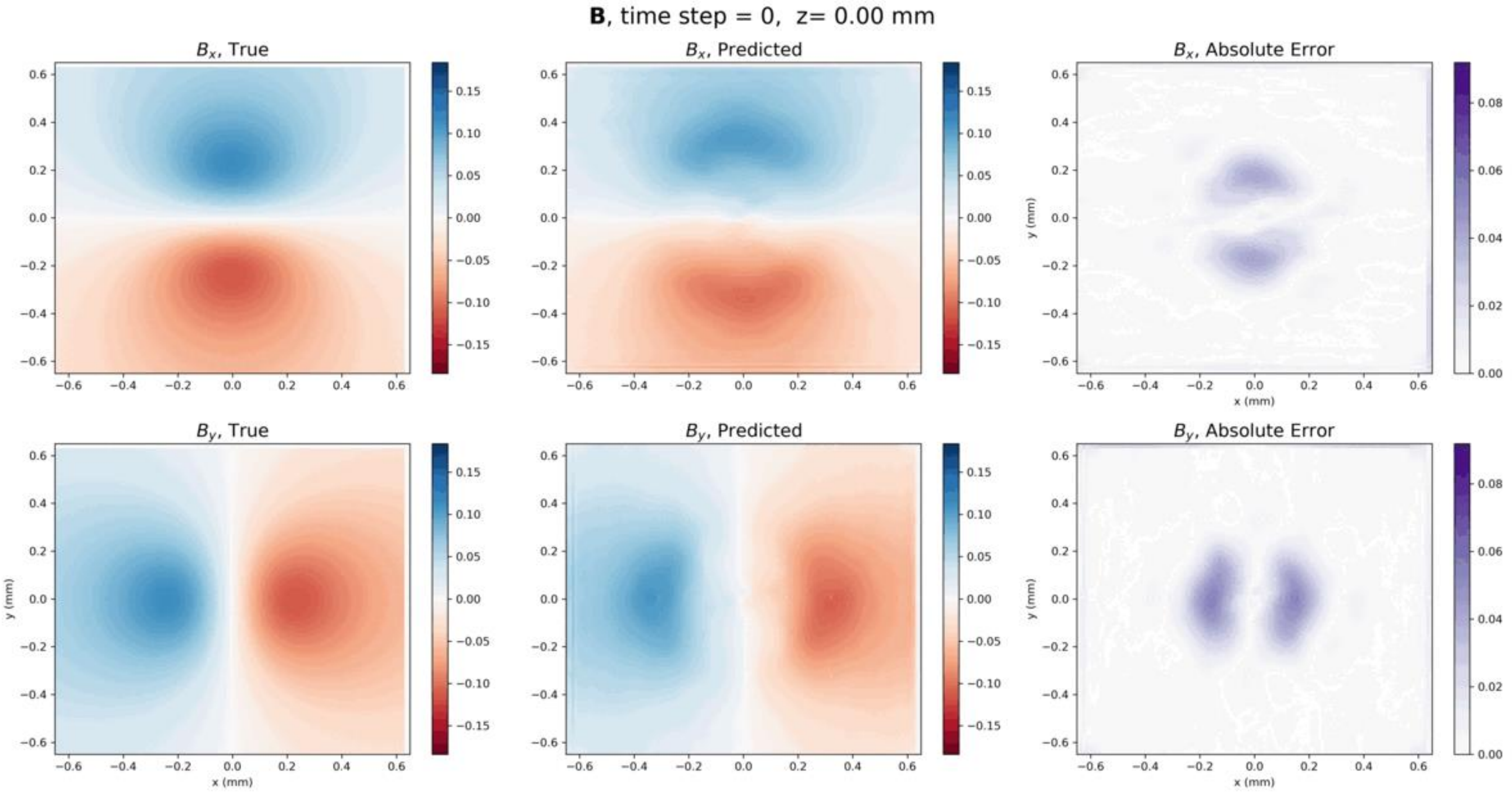


[*Deep Learning*]
Goodfellow, Bengio and Courville (2016)

Back up: CNN-small-4: Four Bunch Beam



Back up: CNN-small-4: Gaussian Beam



Backup: CSR - Computationally Expensive

- $N_e \sim 10^{10}$
- $\sim N_e^2$ interactions + saving distribution at earlier times
- 1D approx. has been used, can slow down by factor ~ 10
- Can CNN's help with speed up?