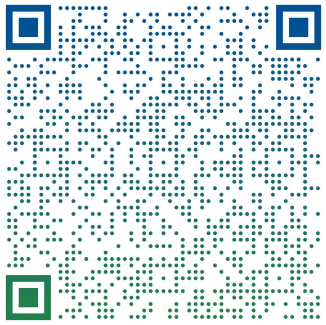
02 Solving Differential Equations with Machine Learning

02.1 Background and Interactive Session: Physics-Informed Neural Networks

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

www.casus.science



IMPRS for Quantum Dynamics and Control

Summer School 2024: Machine Learning and Many Body Systems in or out of Equilibrium

Wroclaw, Poland

July 30, 2024

HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
TECHNISCHE UNIVERSITÄT DRESDEN
CBG Max Planck Institute of Molecular Cell Biology and Genetics
UFZ HELMHOLTZ Centre for Environmental Research
Uniwersytet Wrocławski

SPONSORED BY THE
Federal Ministry of Education and Research
STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS
Freistaat SACHSEN

Generalisability across system parameters

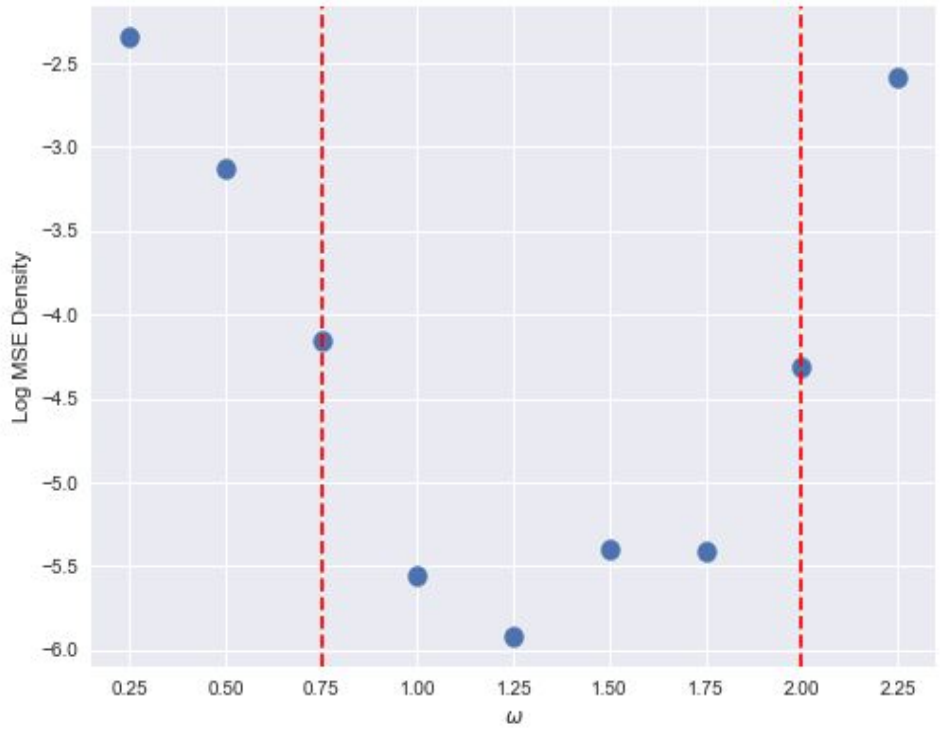


Fig 8. Generalisability across ω (Log MSE Density)

$$\psi_{0,1}(x,t) = \frac{1}{\sqrt{2}} \sqrt[4]{\frac{\omega}{\pi}} \exp\left(-\frac{\omega x^2}{2}\right)\left(exp\left(-i\frac{\omega}{2}t\right) + exp\left(-i\frac{3\omega}{2}t\right)\sqrt{2\omega}x\right)$$

Effects of Collocation point density (1D QHO)

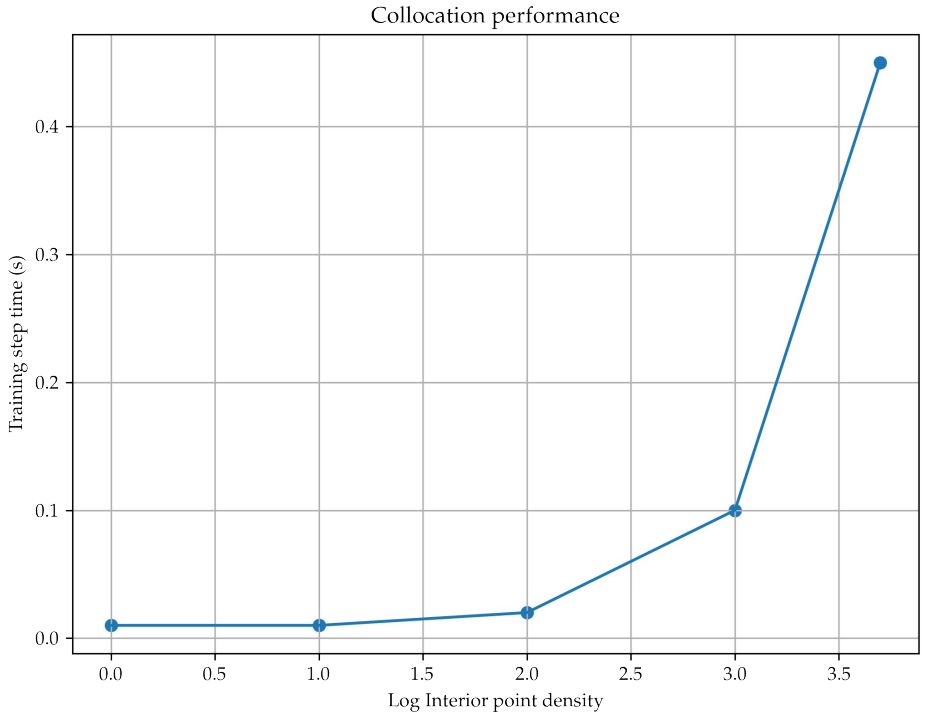


Fig 11a. Effect of increasing training point density on training time.

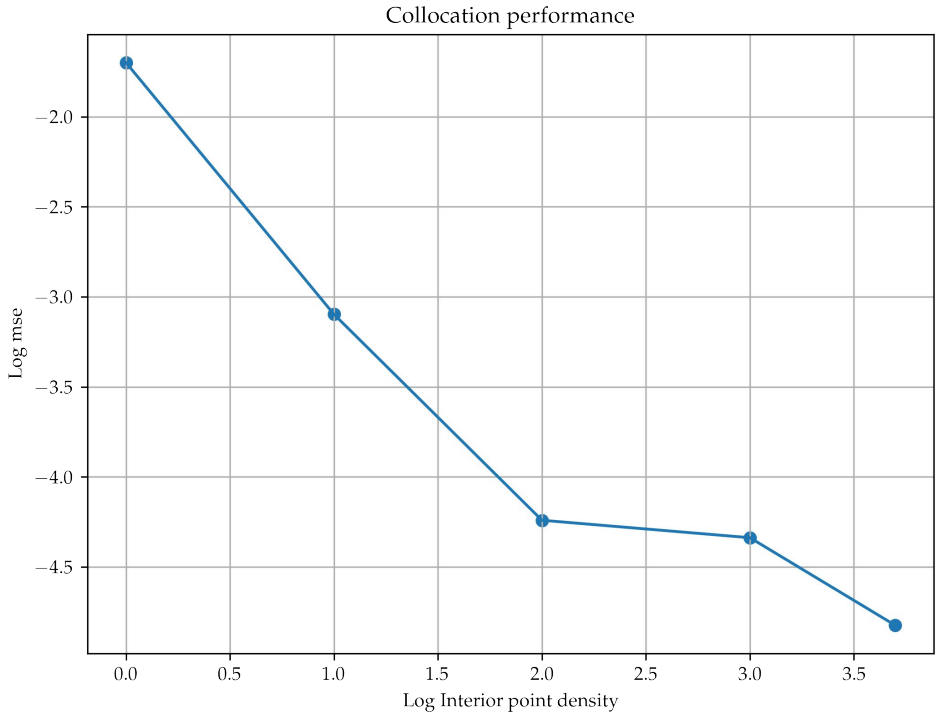
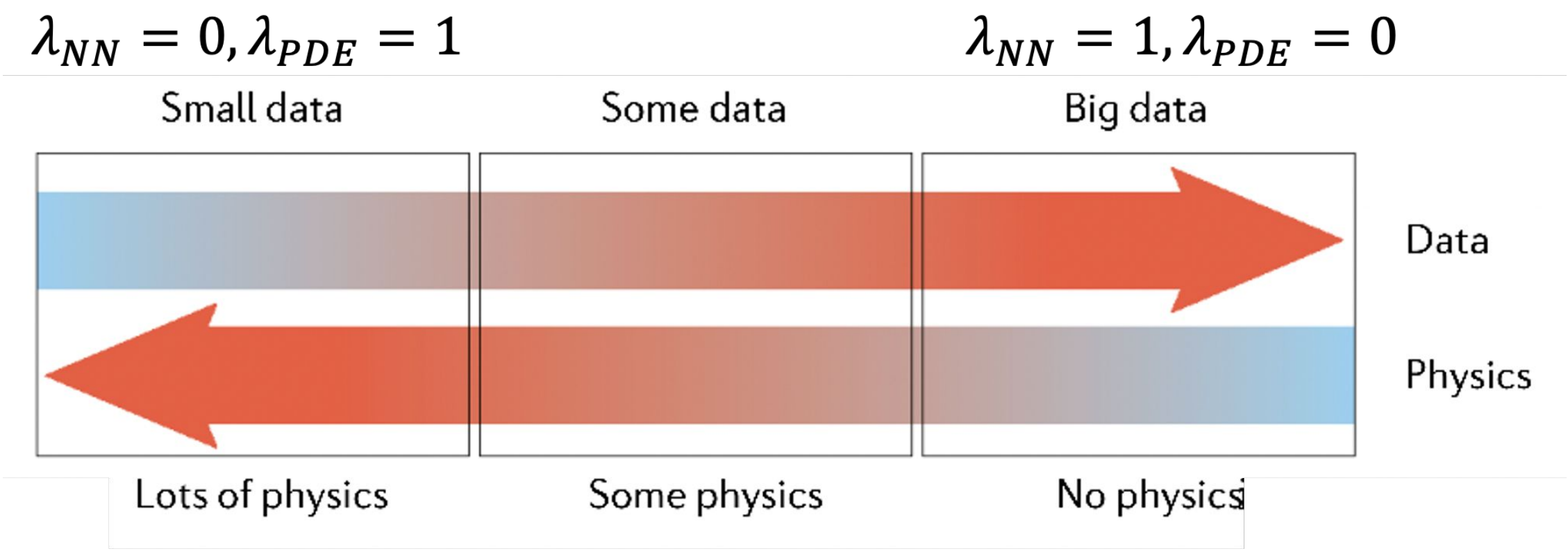


Fig 11b. Effect of increasing training point density on Log mse.

Data-driven vs. physics-informed neural networks



Data-driven vs. physics-informed neural networks

$\lambda_{NN} = 1, \lambda_{PDE} = 0$

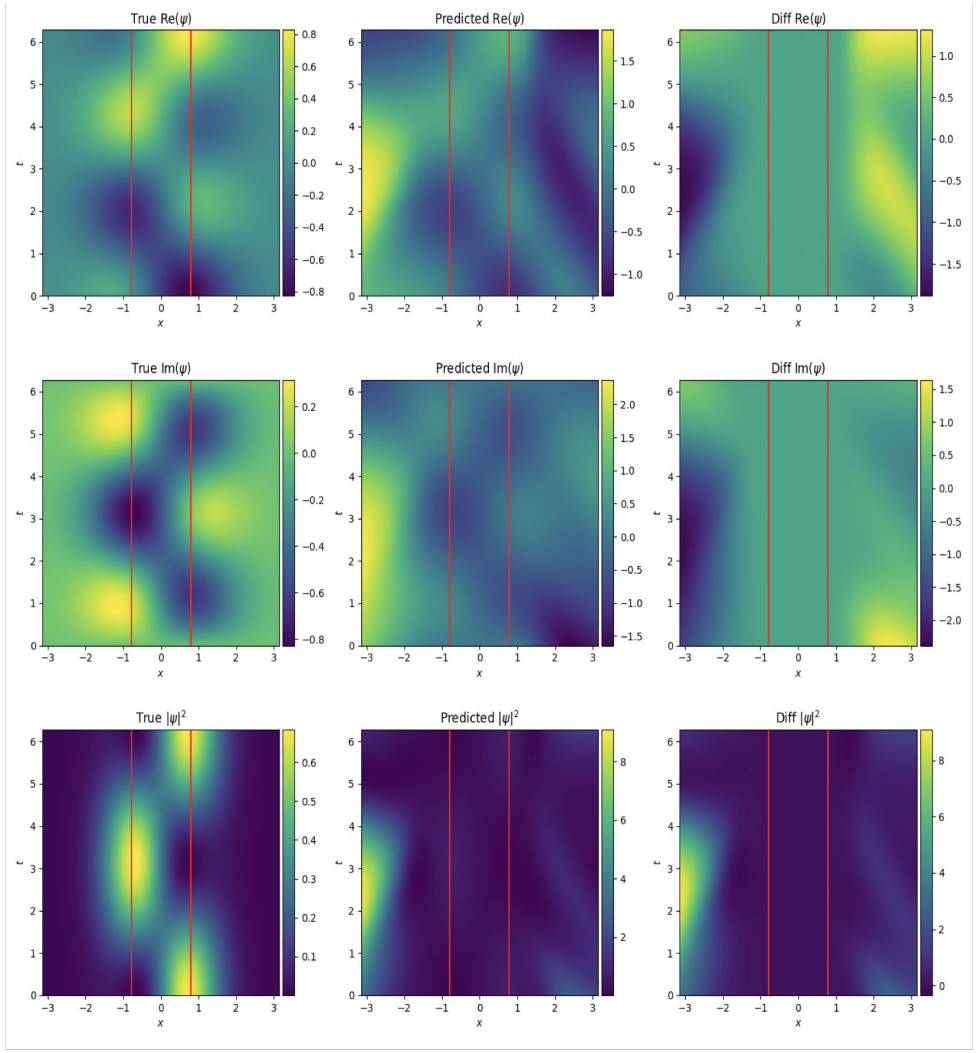


Fig 17a. FCN: MSE (density): 2.6099

$\lambda_{NN} = 1, \lambda_{PDE} = 10^{-4}$

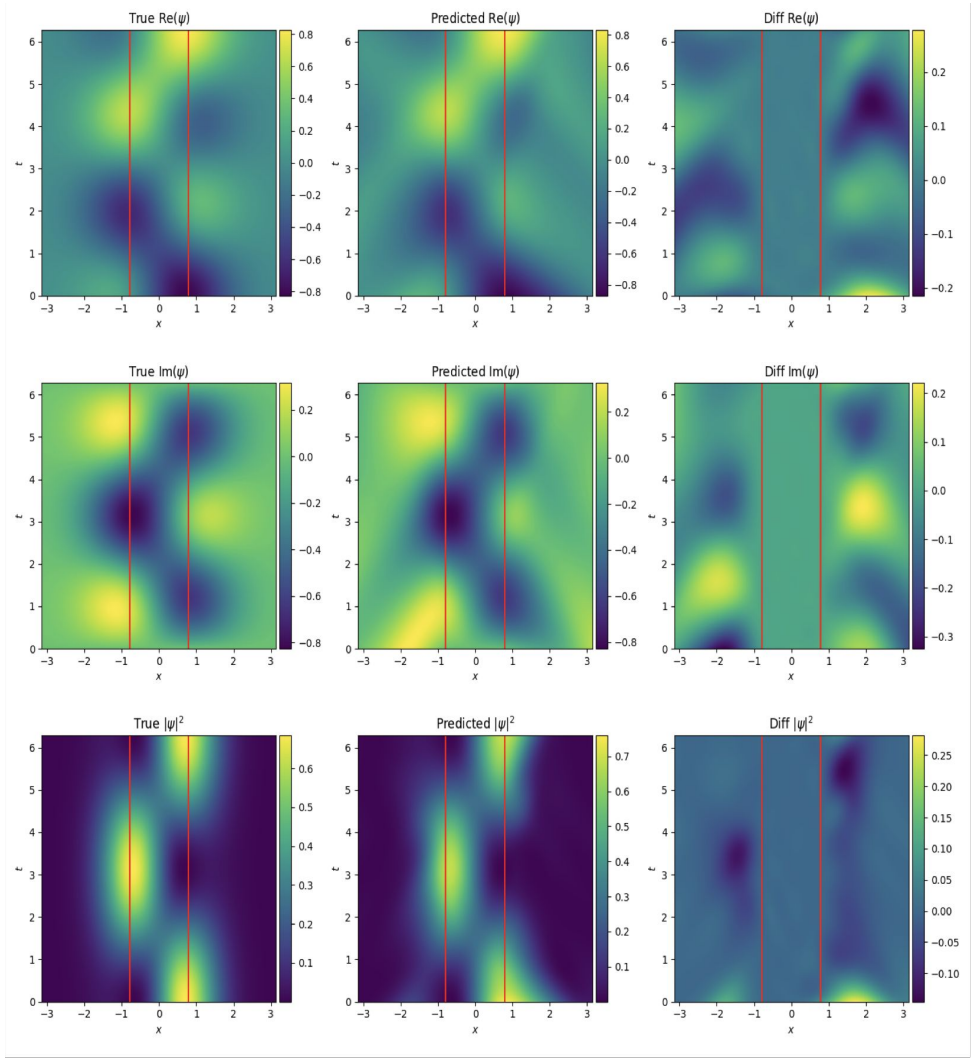


Fig 17b. PINN: MSE (density): 0.0011

Quantum harmonic oscillator in 3D

3D render of density for $\psi_{0,1}$ with $\omega = 1.0$ at $t = \frac{\pi}{2}$

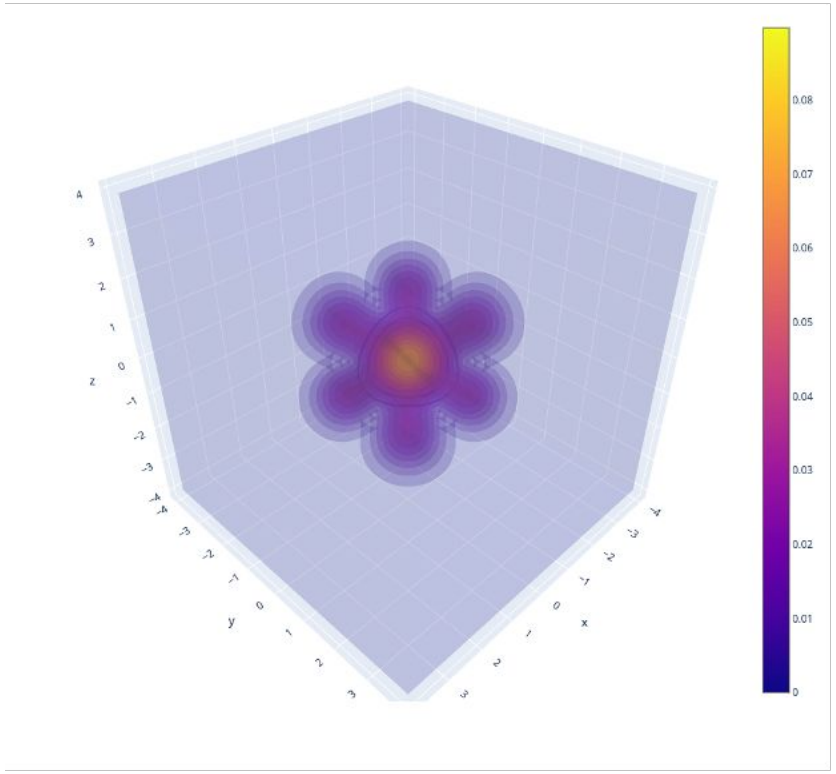


Fig 12a. True density

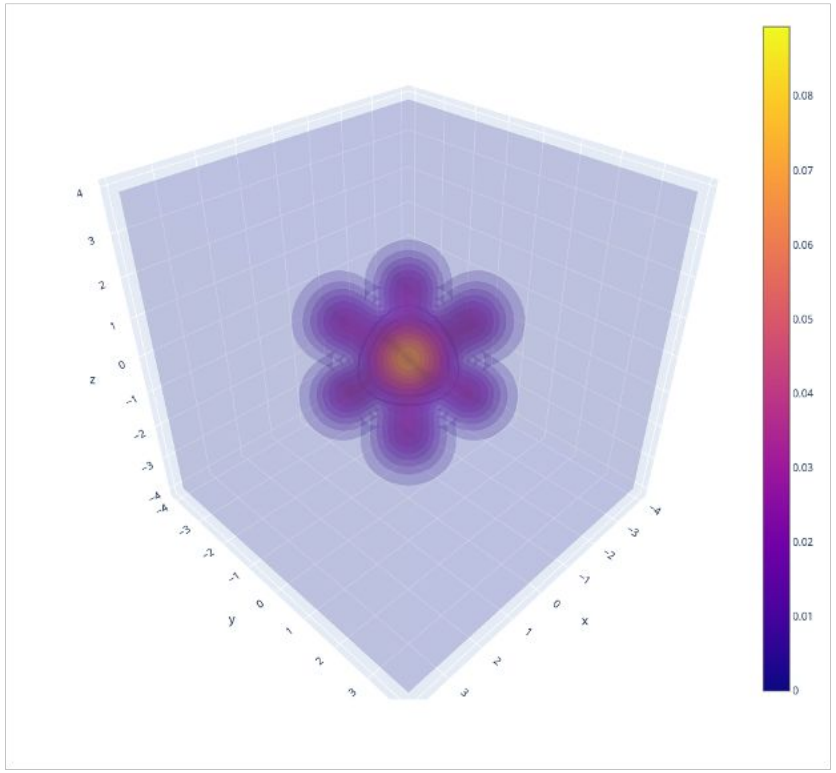


Fig 12b. Predicted density

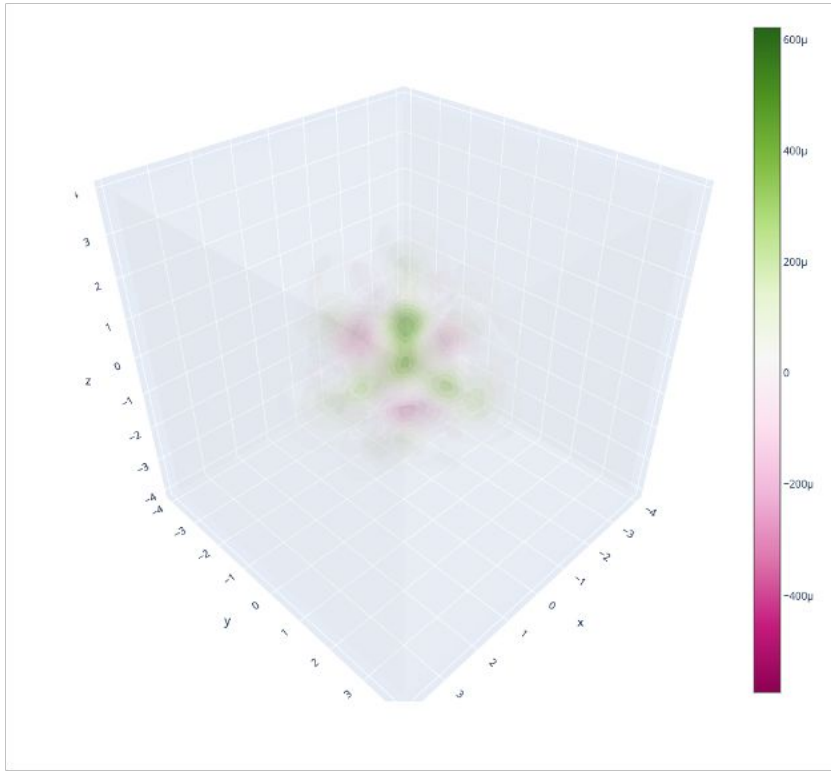


Fig 12c. Difference

Errors

Max Abs: 6.21e-4, Min Abs: 0.0, Mean Abs: 4.05e-6

MSE: 1.81e-9

Quantum harmonic oscillator in 3D

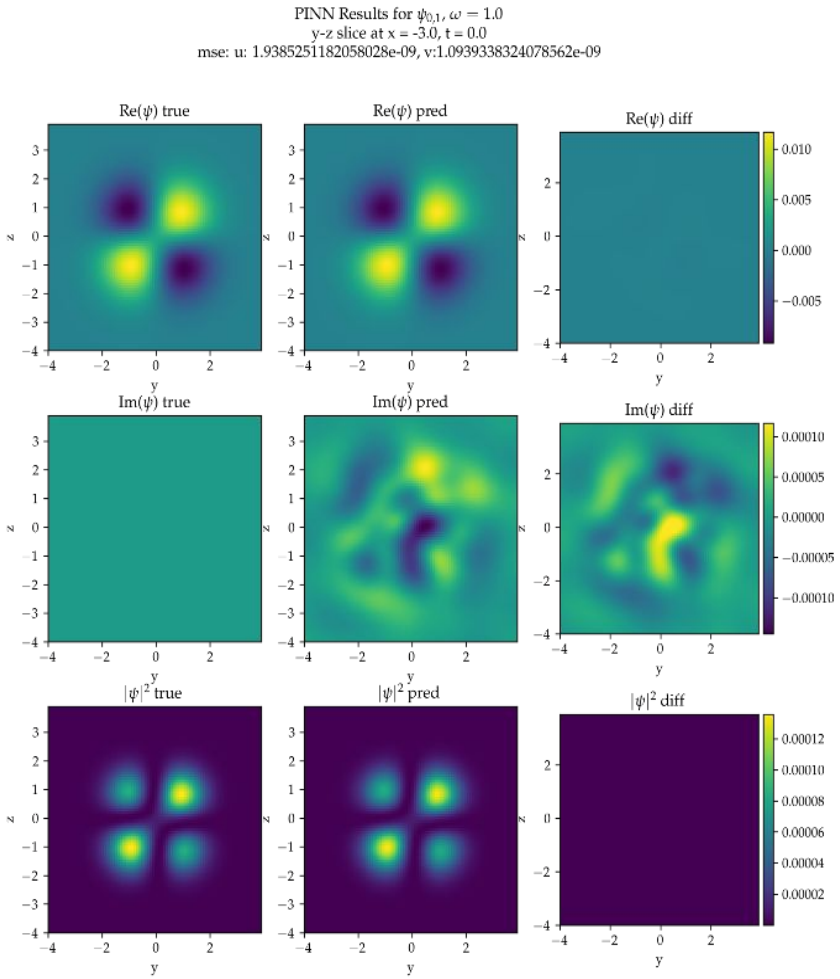


Fig 13a. y-z snapshot for true and predicted values for 3D $\psi_{0,1}$ with ω = 1.0 at x = -3.0, t = 0.0.
MSE_u = 1.94e-9, MSE_v = 1.09e-9

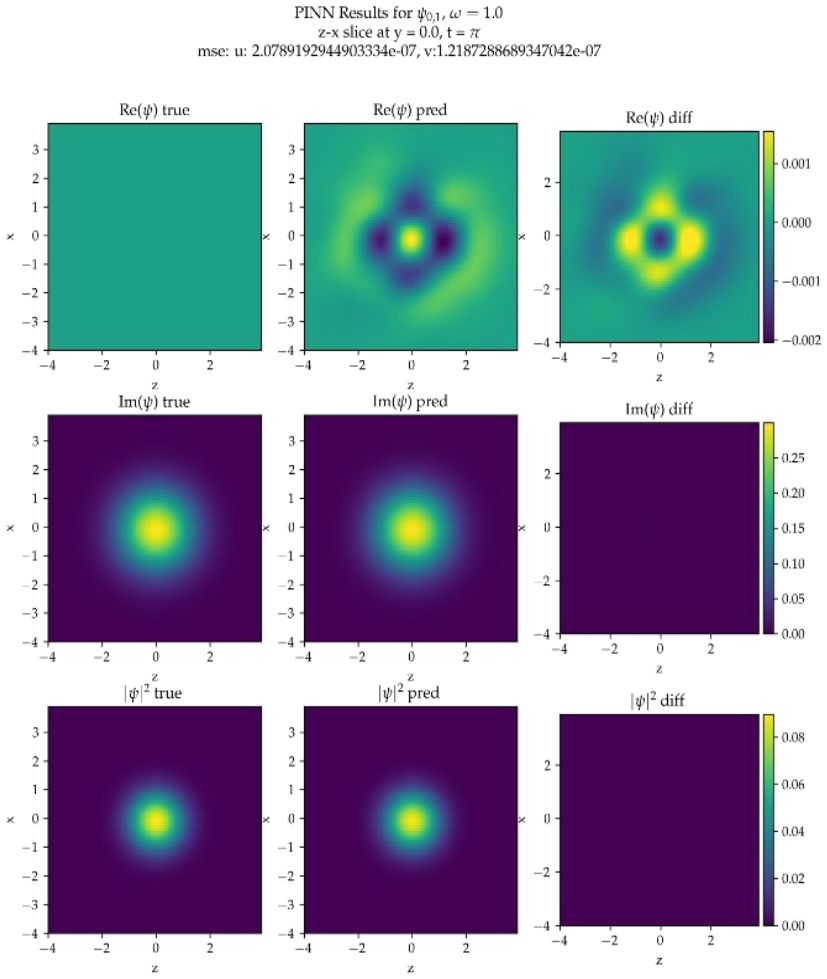


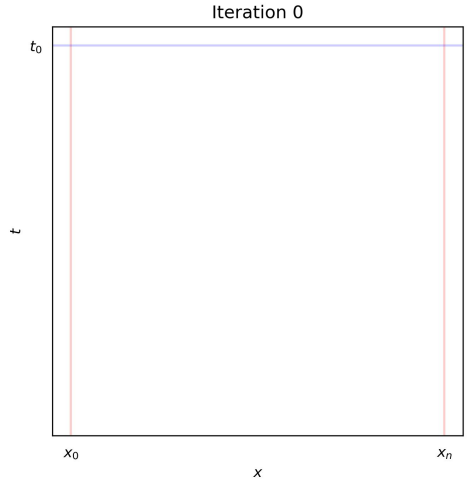
Fig 13b. z-x snapshot for true and predicted values for 3D $\psi_{0,1}$ with ω = 1.0 at y = 0.0, t = π.
MSE_u = 2.07e-7, MSE_v = 1.21e-7

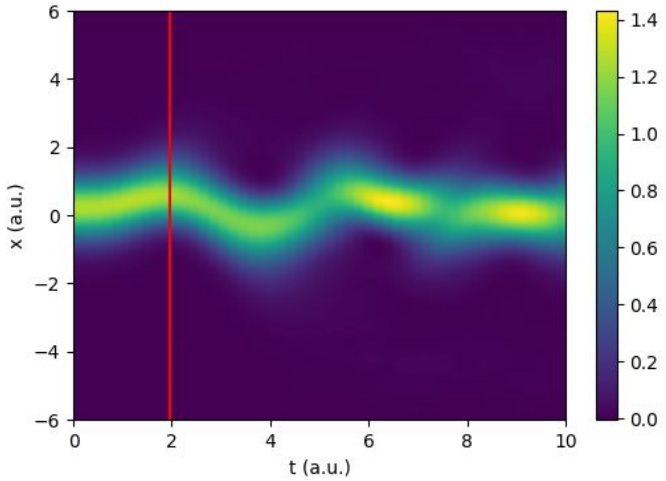
02 Solving Differential Equations with Machine Learning

02.2 Research Application:
Physics-Informed Neural Networks and Neural Operators

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

www.casus.science





IMPRS for Quantum Dynamics and Control

Summer School 2024: Machine Learning and Many Body Systems in or out of Equilibrium

Wroclaw, Poland

July 30, 2024

HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
TECHNISCHE UNIVERSITÄT DRESDEN
CBG Max Planck Institute of Molecular Cell Biology and Genetics
UFZ HELMHOLTZ Centre for Environmental Research
Uniwersytet Wrocławski

SPONSORED BY THE
Federal Ministry of Education and Research
STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS
Freistaat SACHSEN

Outline

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

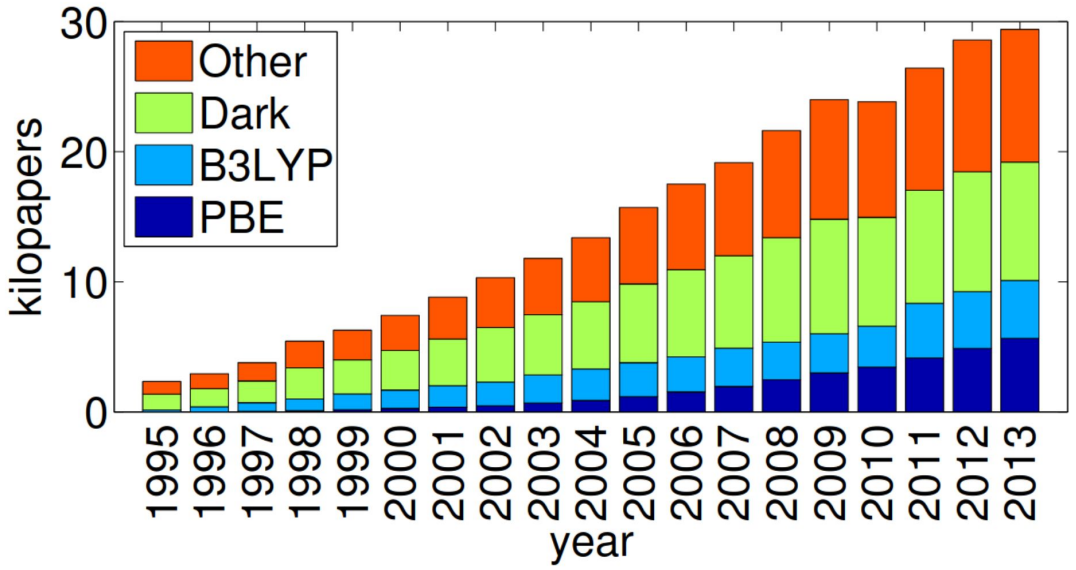
Accelerating Electron Dynamics with Machine Learning

Outlook

Motivation

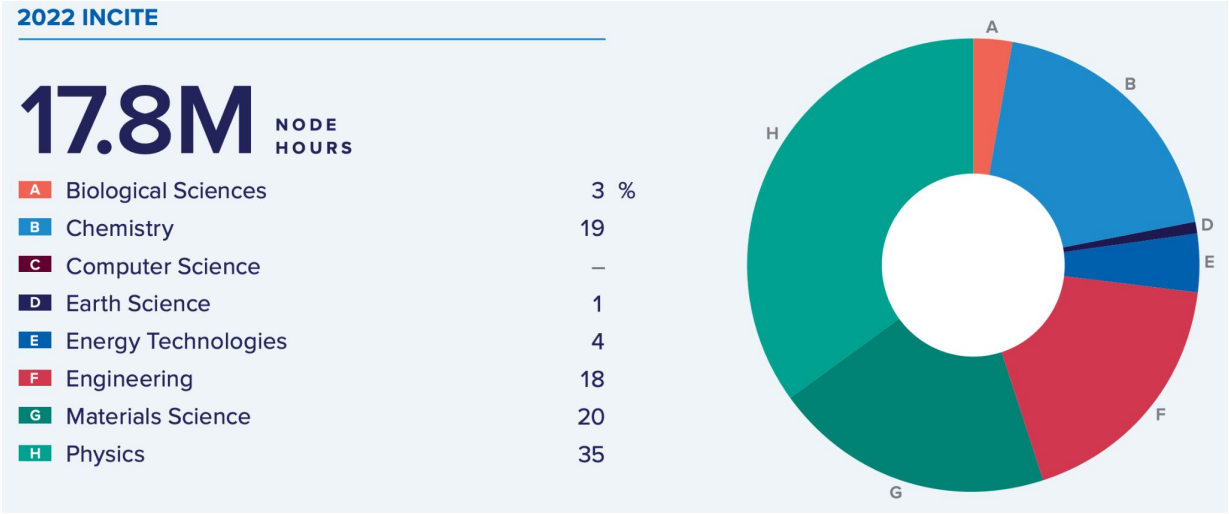
Density functional theory

Most popular method for solving the electronic structure problem



Pribram-Jones et al.,
https://doi.org/10.1146/annurev-physchem-040214-121420

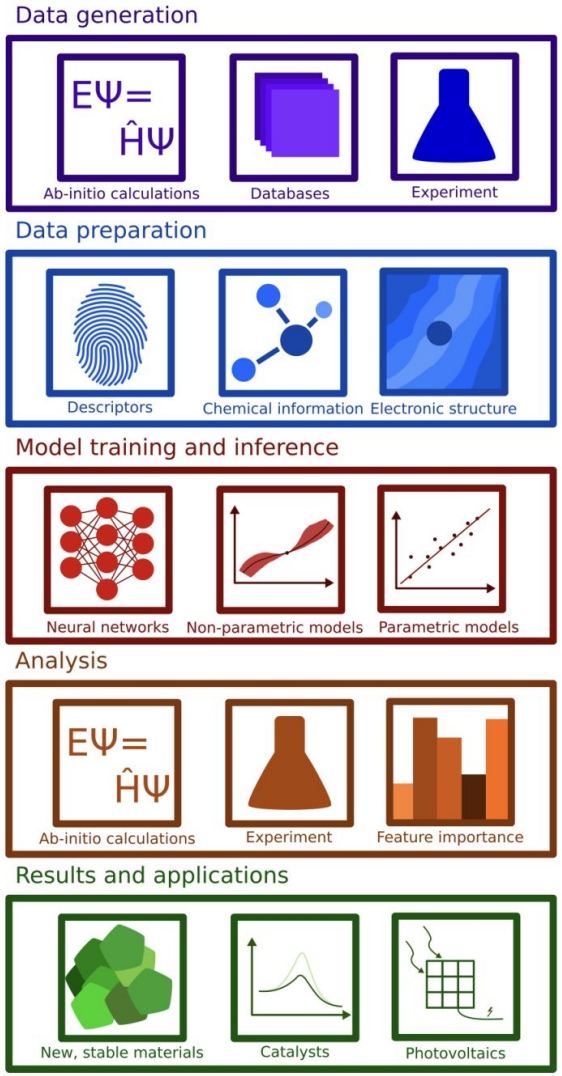
One of the world's largest computational expenses



2022 ALCF Annual report, https://ar22.alcf.anl.gov/

Motivation

State of the art in combining electronic structure theory with machine learning

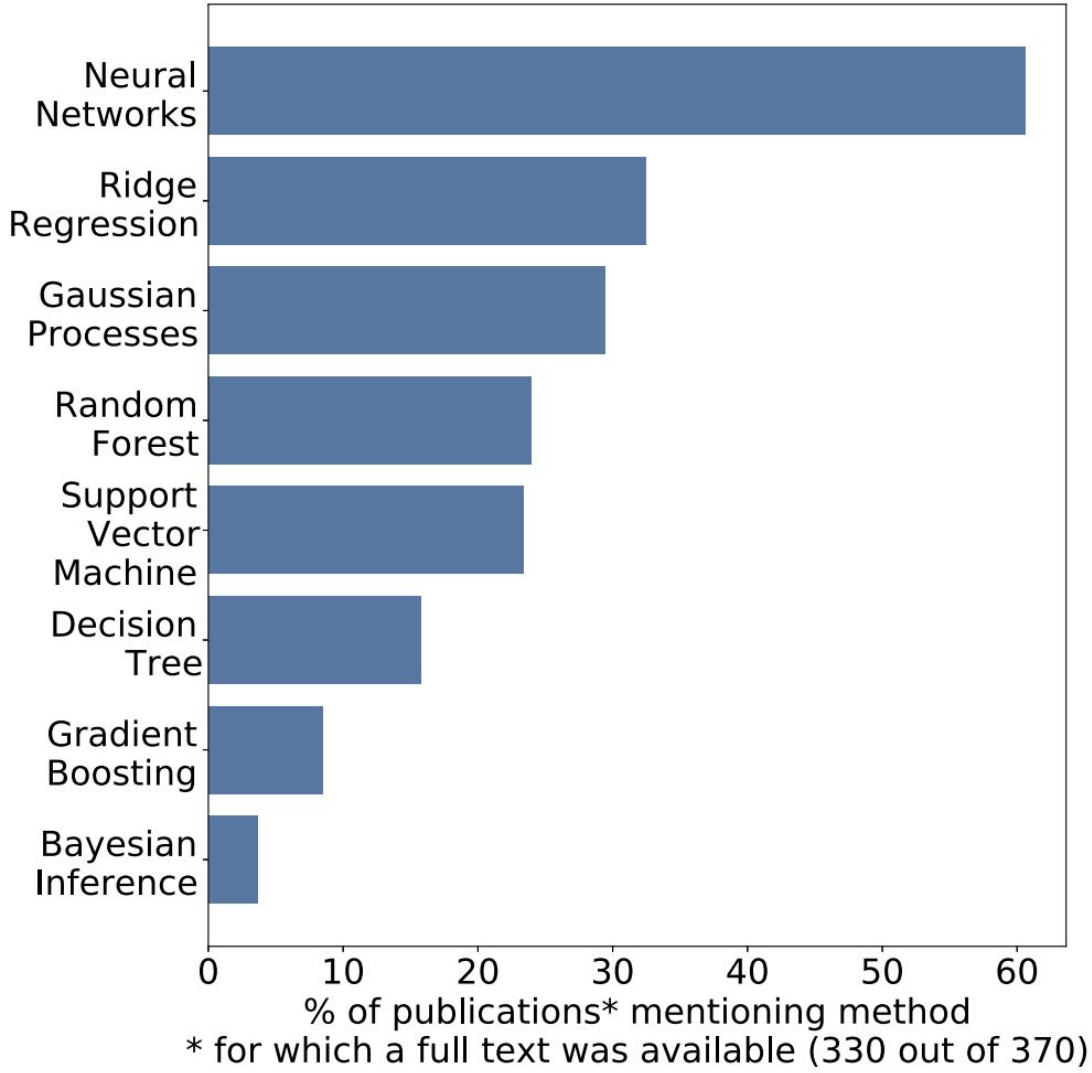


Meta study analyzing 370 research articles

L. Fiedler, K. Shah, M. Bussmann, A. Cangi, Phys. Rev. Mater. 6, 040301 (2022).

Motivation

State of the art in combining electronic structure theory with machine learning

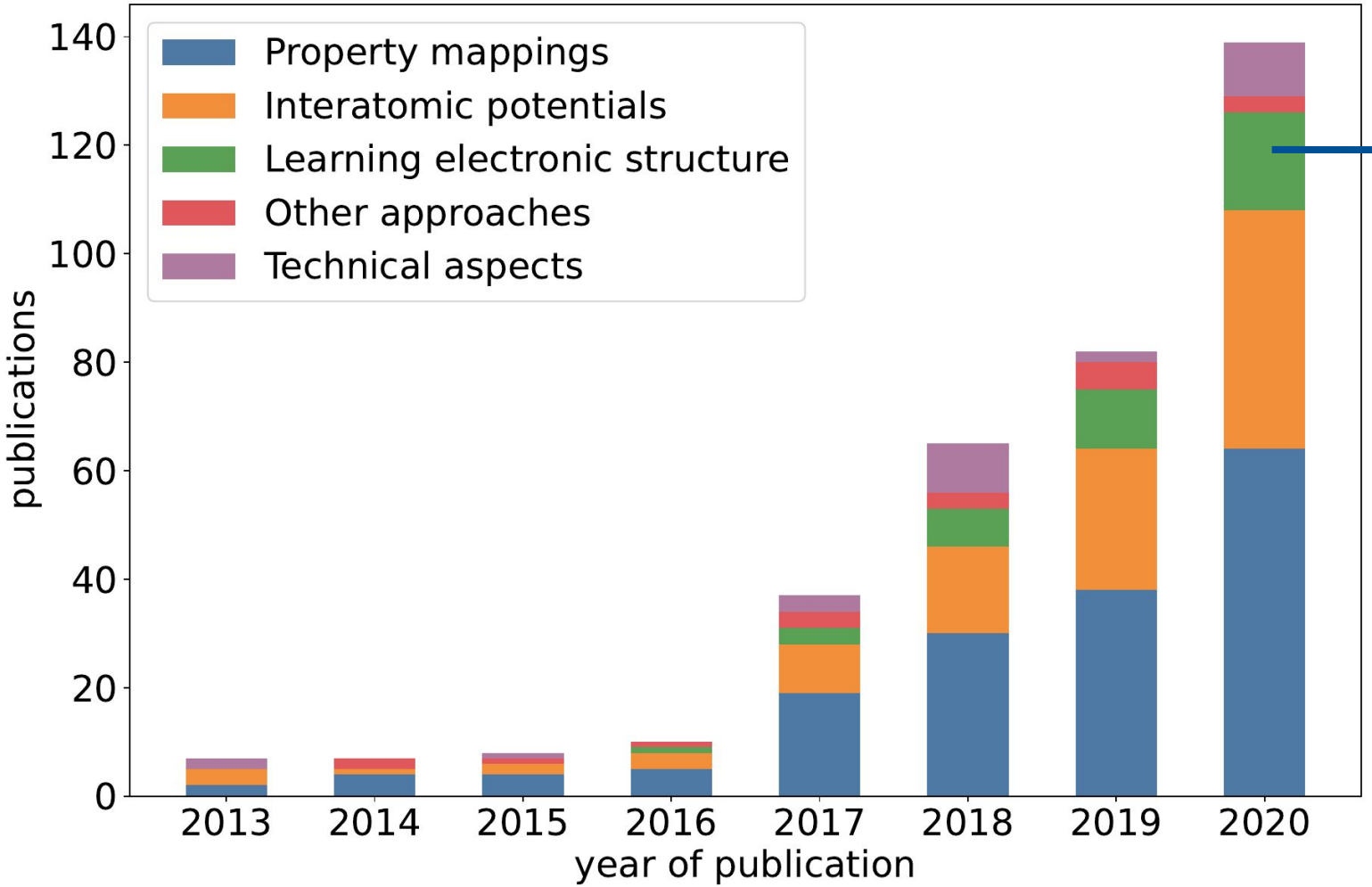


Meta study analyzing 370 research articles

L. Fiedler, K. Shah, M. Bussmann, A. Cangi, Phys. Rev. Mater. 6, 040301 (2022).

Motivation

State of the art in combining electronic structure theory with machine learning



Machine learning the electronic structure (this work)

Meta study analyzing 370 research articles

L. Fiedler, K. Shah, M. Bussmann, A. Cangi, Phys. Rev. Mater. 6, 040301 (2022).

Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

Accelerating Electron Dynamics with Machine Learning

Outlook

Electronic Structure Problem

Non-relativistic Schrödinger equation

$$\hat{H}(\underline{\mathbf{r}}, \underline{\mathbf{R}})\Psi(\underline{\mathbf{r}}, \underline{\mathbf{R}}) = E\Psi(\underline{\mathbf{r}}, \underline{\mathbf{R}})$$

$$\hat{H}(\underline{\mathbf{r}}; \underline{\mathbf{R}}) = \hat{T}_e(\underline{\mathbf{r}}) + \hat{V}_{ee}(\underline{\mathbf{r}}) + \hat{V}_{ei}(\underline{\mathbf{r}}; \underline{\mathbf{R}}) + \hat{V}_{ii}(\underline{\mathbf{R}})$$

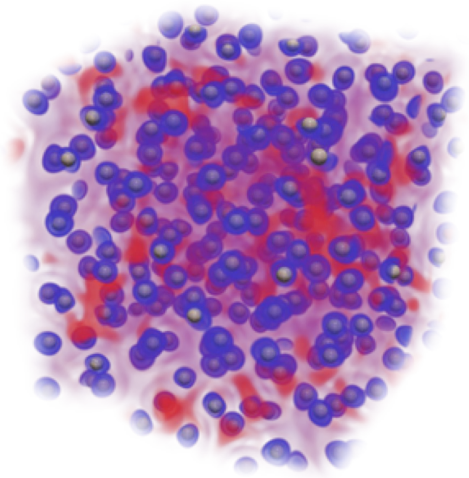
$$\hat{T}_e(\underline{\mathbf{r}}) = \sum_i^{N_e} -\frac{\nabla_i^2}{2}$$

$$\hat{V}_{ee}(\underline{\mathbf{r}}) = \sum_i^{N_e} \sum_{j \neq i}^{N_e} \frac{1}{2} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

$$\hat{V}_{ei}(\underline{\mathbf{r}}; \underline{\mathbf{R}}) = -\sum_i^{N_e} \sum_\alpha^{N_i} \frac{Z_\alpha}{|\mathbf{r}_i - \mathbf{R}_\alpha|}$$

$$\hat{V}_{ii}(\underline{\mathbf{R}}) = -\sum_\alpha^{N_i} \sum_{\beta \neq \alpha}^{N_i} \frac{1}{2} \frac{Z_\alpha Z_\beta}{|\mathbf{R}_\alpha - \mathbf{R}_\beta|}$$

Molecular and materials properties



Molecular structure, Crystal structure, Charge density, Cohesive energy, Elastic properties, Vibrational properties, Magnetic order, Dielectric susceptibility, Magnetic susceptibility, Phase transitions, Bond dissociation, Enthalpies of formation, Ionization potential, Electron affinity, Band gaps, Equation of state

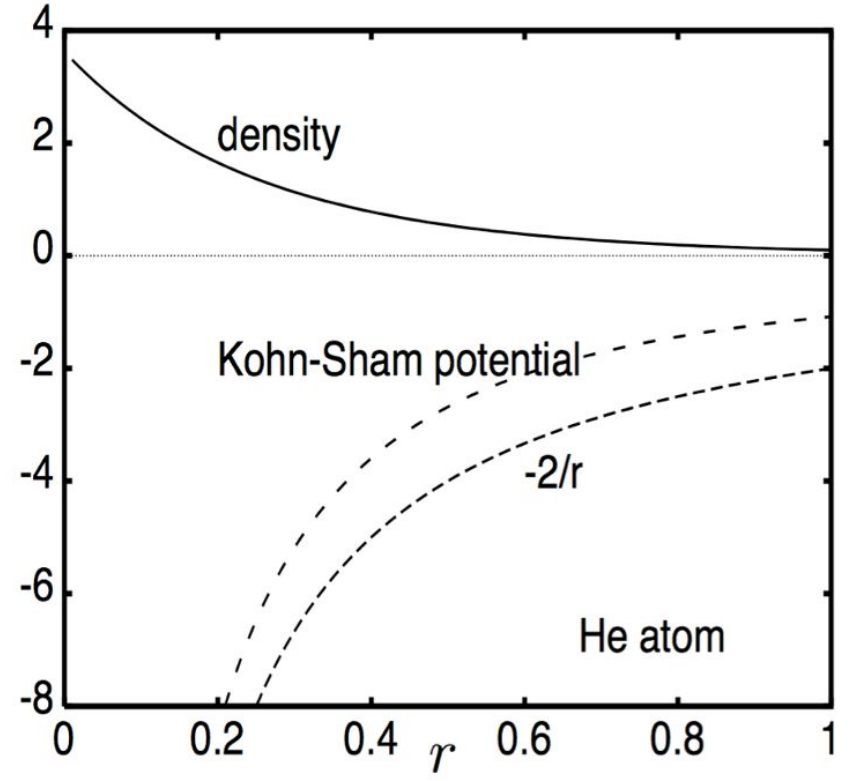
Density Functional Theory

$$\left[-\frac{1}{2}\nabla^2 + v_{\mathrm{S}}(\mathbf{r};\underline{\mathbf{R}})\right]\phi_j(\mathbf{r};\underline{\mathbf{R}}) = \epsilon_j\phi_j(\mathbf{r};\underline{\mathbf{R}})$$

$$v_{\mathrm{S}}(\mathbf{r};\underline{\mathbf{R}}) = \frac{\delta U[n]}{\delta n(\mathbf{r};\underline{\mathbf{R}})} + \frac{\delta E_{\mathrm{XC}}[n]}{\delta n(\mathbf{r};\underline{\mathbf{R}})} + v_{ei}(\mathbf{r};\underline{\mathbf{R}})$$

$$n(\mathbf{r};\underline{\mathbf{R}}) = \sum_j \phi_j^*(\mathbf{r};\underline{\mathbf{R}})\phi_j(\mathbf{r};\underline{\mathbf{R}})$$

$$E[n] = T_{\mathrm{S}}[n] + U[n] + E_{\mathrm{XC}}[n] + \int d^3r\; n(\mathbf{r};\underline{\mathbf{R}})v_{ei}(\mathbf{r};\underline{\mathbf{R}})$$



K. Burke, „The ABC of DFT“.

Time-Dependent Density Functional Theory

Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

Accelerating Electron Dynamics with Machine Learning

Outlook

Theoretical Background

Neural Operators

Neural operators extend neural networks by mapping functions to functions instead of finite-dimensional vectors.

A neural network maps:

$$\mathbb{R}^n \rightarrow \mathbb{R}^m$$

A neural operator maps function spaces.
Our goal is to approximate the non-linear map:

$$\mathcal{G}^\dagger : \mathcal{A} \rightarrow \mathcal{U}$$

With a neural operator:

$$\mathcal{G}_\theta , \quad \theta \in \mathbb{R}^p$$

Theoretical Background

Neural Operators

A neural operator is defined by:

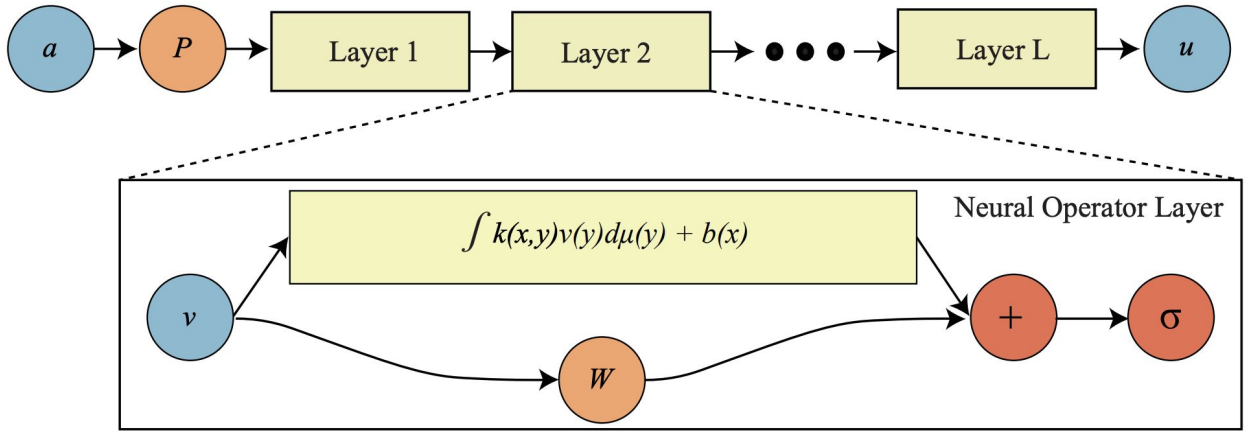
$$\mathcal{G}_\theta(a) = Q(v_L(v_{L-1}(\dots v_1(P(a)))))$$

with layers

$$v_{l+1}(x) = \sigma_{(l+1)}\left(W_l v_l(x) + \left(\mathcal{K}_l(a;\lambda)v_l\right)(x)\right)$$

and a non-local kernel integral operator:

$$\left(\mathcal{K}_l(a;\lambda)v_l\right)(x) = \int_{\Omega_l} \kappa_l(x, y, a(x), a(y); \lambda) v_l(y) dy$$



Training involves minimizing the loss:

$$\theta^* = \min_{\theta \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \|u_i - \mathcal{G}_\theta(a_i)\|_{\mathcal{U}}^2 .$$

Observations: $\{(a_i, u_i)\}_{i=1}^N$

$$u_i = \mathcal{G}^\dagger(a_i)$$

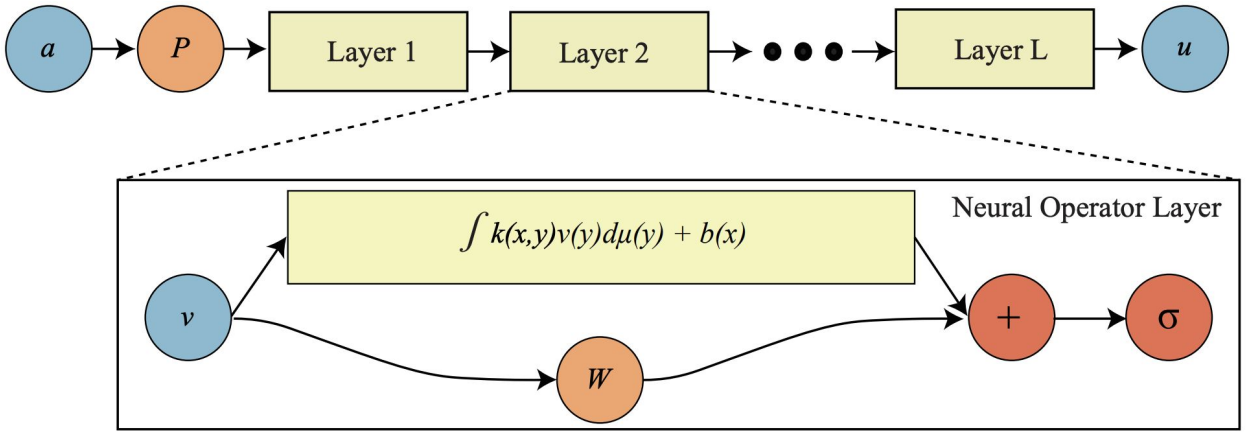
Theoretical Background

Fourier Neural Operators

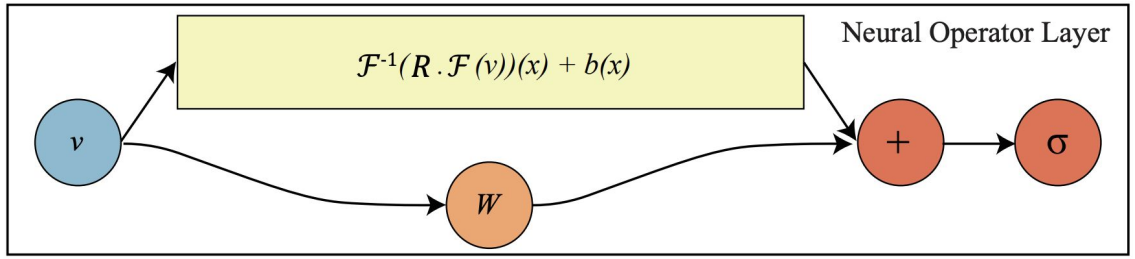
Fourier neural operators (FNOs) use the Fourier transform

$$\left(\mathcal{K}_l(a;\lambda)v_l\right)(x) = \mathcal{F}^{-1}\left(\mathcal{F}\left(\kappa_l\right)\cdot\mathcal{F}\left(v_l\right)\right)(x)$$

This enables efficient kernel computation and effective capture of global patterns using the fast Fourier transform.



FNO layer:



Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

Accelerating Electron Dynamics with Machine Learning

Outlook

Theoretical Background

Physics-Informed Neural Networks

Consider the PDE:

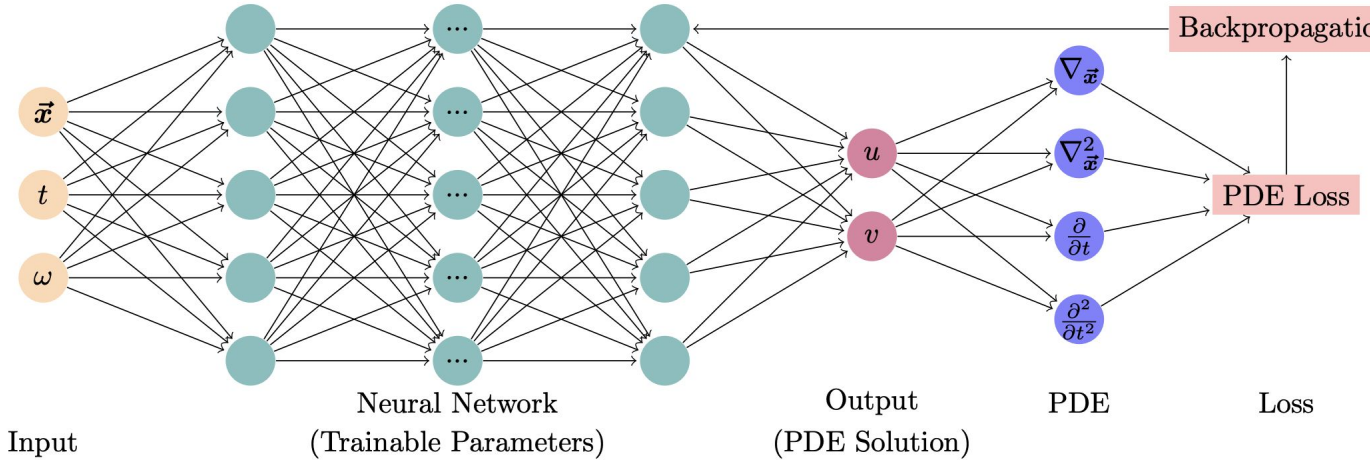
$$u(x,t) + \mathcal{N}[u;\lambda] = 0$$

Construct the solution in terms of a neural network:

$$\tilde{u}(x,t) + \mathcal{N}[\tilde{u};\lambda] := f(x,t)$$

Minimize loss term:

$$L = \gamma_f L_f + \gamma_{BC} L_{BC} + \gamma_{IC} L_{IC}$$



Theoretical Background

Physics-Informed Neural Networks

Consider the PDE:

$$u(x,t) + \mathcal{N}[u;\lambda] = 0$$

Construct the solution in terms of a neural network:

$$\tilde{u}(x,t) + \mathcal{N}[\tilde{u};\lambda] := f(x,t)$$

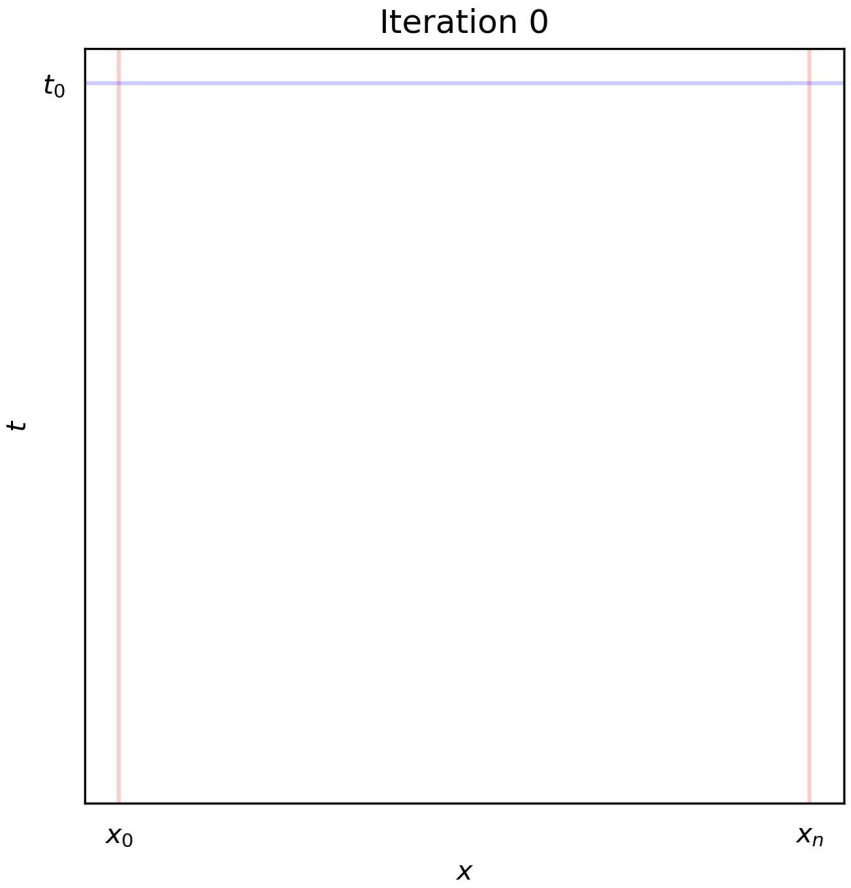
Minimize loss term:

$$L = \gamma_f L_f + \gamma_{BC} L_{BC} + \gamma_{IC} L_{IC}$$

$$L_f = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| f(x_f^i, t_f^i) \right|^2$$

$$L_{BC} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left| \tilde{u}(x_{BC}^i, t_{BC}^i) - u(x_{BC}^i, t_{BC}^i) \right|^2$$

$$L_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} \left| \tilde{u}(x_{IC}^i, t_{IC}^i) - u(x_{IC}^i, t_{IC}^i) \right|^2$$

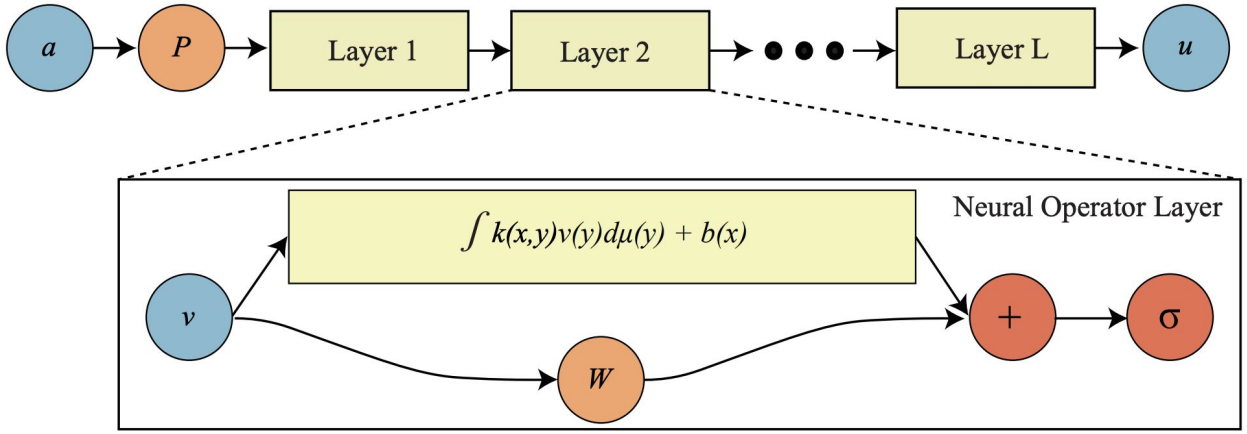


Theoretical Background

Neural Operators vs. Physics-Informed Neural Networks

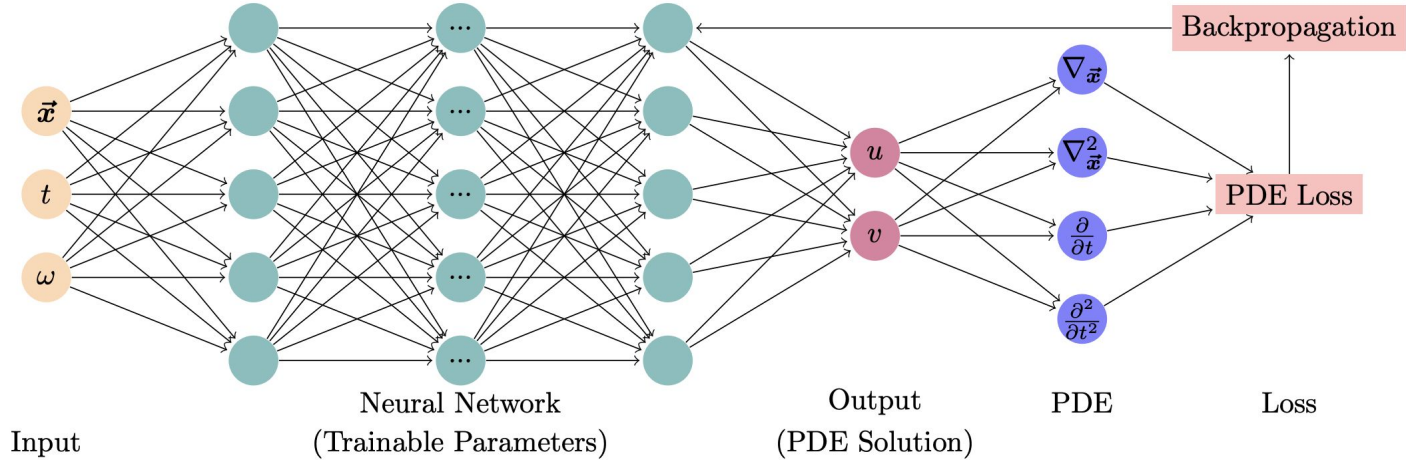
Neural Operators

- learns the underlying operator itself
- needs training data
- enables the prediction of solutions for various conditions without re-solving the PDE
- is mesh-invariant



Physics-Informed Neural Networks

- is also mesh-invariant
- useful for obtaining numerical solutions for specific initial and boundary conditions and PDE parametrizations
- can be used to solve PDEs if the PDEs are defined, even in the absence of training data (unsupervised learning)
- models a specific instance of a PDE



Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

Accelerating Electron Dynamics with Machine Learning

Outlook

Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

Accelerating Electron Dynamics with Machine Learning

Outlook

Inverting the Kohn-Sham Equations with Machine Learning

Summary

Karan Shah

Inverting the Kohn-Sham equations is useful for

- developing exchange-correlation (XC) approximations.
- constructing optical potentials for electron scattering (R-matrix scattering).

Traditional inversion methods are often plagued by the numerical errors and instabilities.

Investigate the potential of both PINNs and FNOs for the inversion problem in Kohn-Sham DFT.

Predict the exchange-correlation (XC) potential based on input electronic densities.

Reference: V. Martinetto, K. Shah, A. Cangi, A. Pribram-Jones, Mach. Learn.: Sci. Technol. 5 015050 (2024).

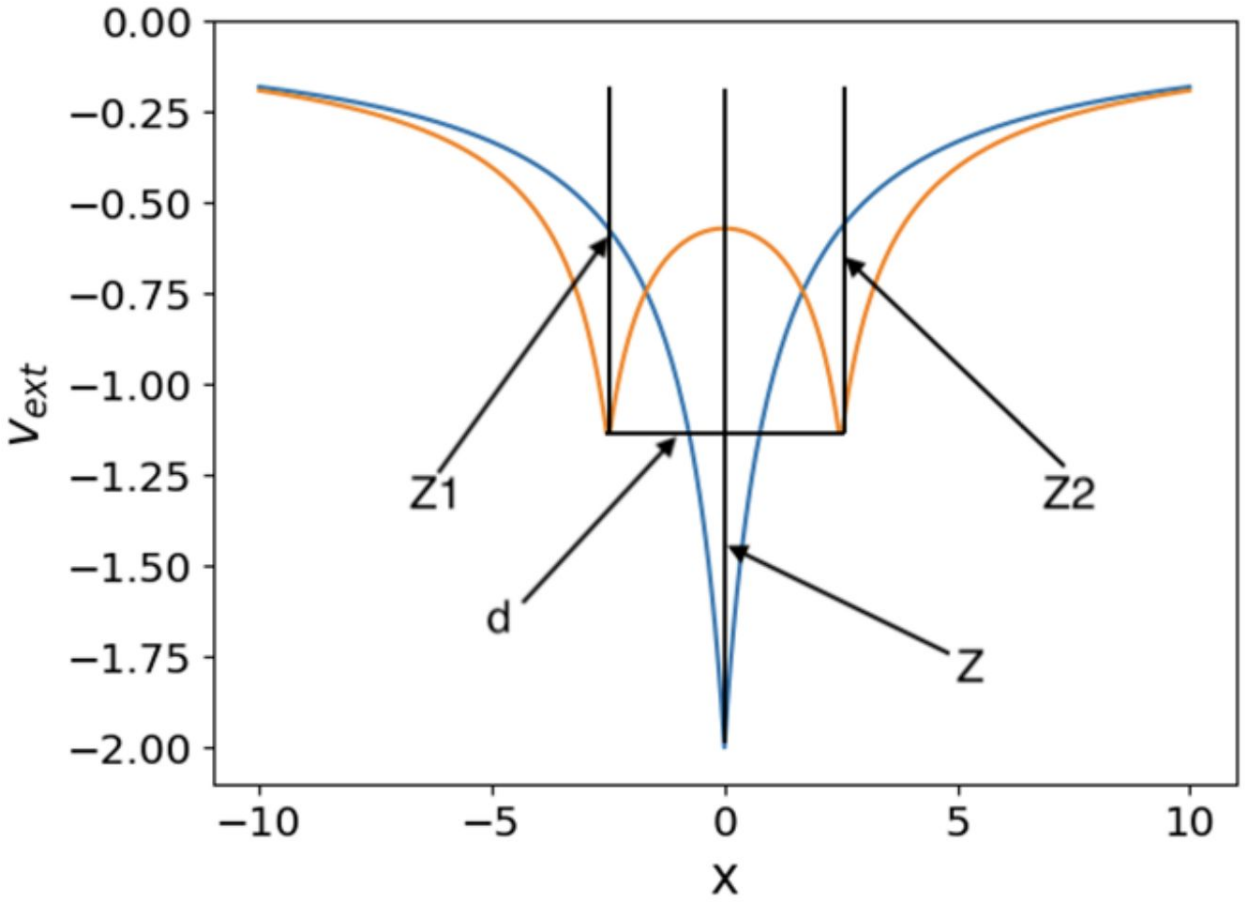
Inverting the Kohn-Sham Equations with Machine Learning

Model system

One-dimensional diatomic molecules

$$v_{\text{ext}}(\mathbf{r}) = -\frac{Z_1}{|\mathbf{r} - \frac{d}{2}| + a} - \frac{Z_2}{|\mathbf{r} + \frac{d}{2}| + a}$$

Data set with 729 molecules (590 in training, 66 in validation, 73 in test set)



Inverting the Kohn-Sham Equations with Machine Learning

Model system

One-dimensional diatomic molecules

$$v_{\text{ext}}(\mathbf{r}) = -\frac{Z_1}{|\mathbf{r} - \frac{d}{2}| + a} - \frac{Z_2}{|\mathbf{r} + \frac{d}{2}| + a}$$

Data set with 729 molecules (590 in training, 66 in validation, 73 in test set)

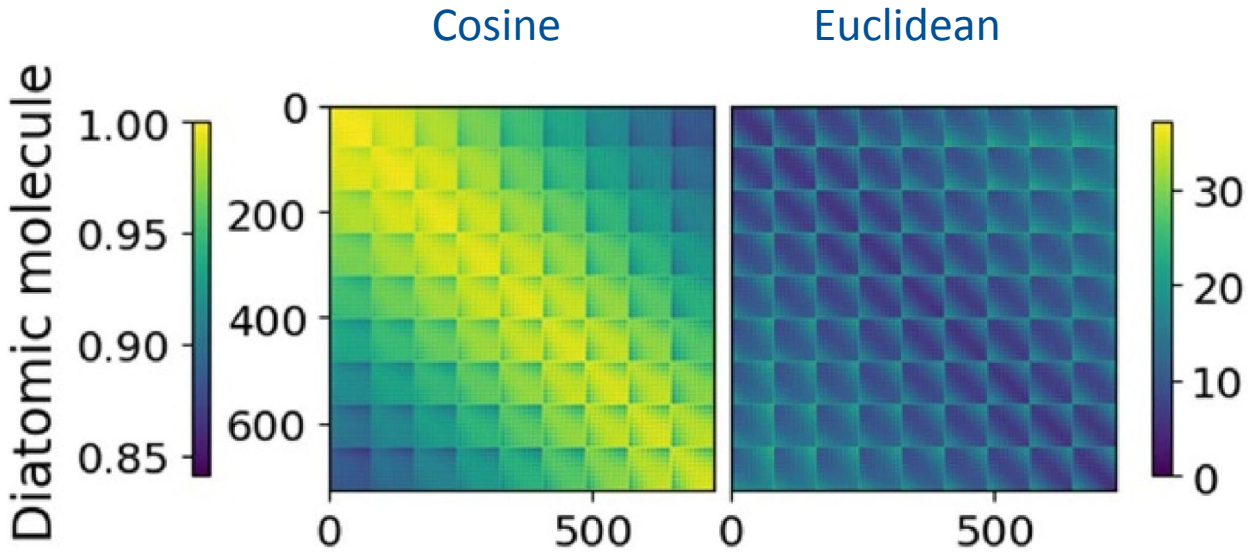
Complexity of the data set analyzed in terms of similarity measures:

- cosine similarity distance (values between -1,1)

$$\cos(\theta) = \frac{\mathbf{X} \cdot \mathbf{Y}}{||\mathbf{X}||||\mathbf{Y}||}$$

- Euclidean distance

$$D(\mathbf{X}, \mathbf{Y}) = ||\mathbf{X} - \mathbf{Y}||$$



Inverting the Kohn-Sham Equations with Machine Learning

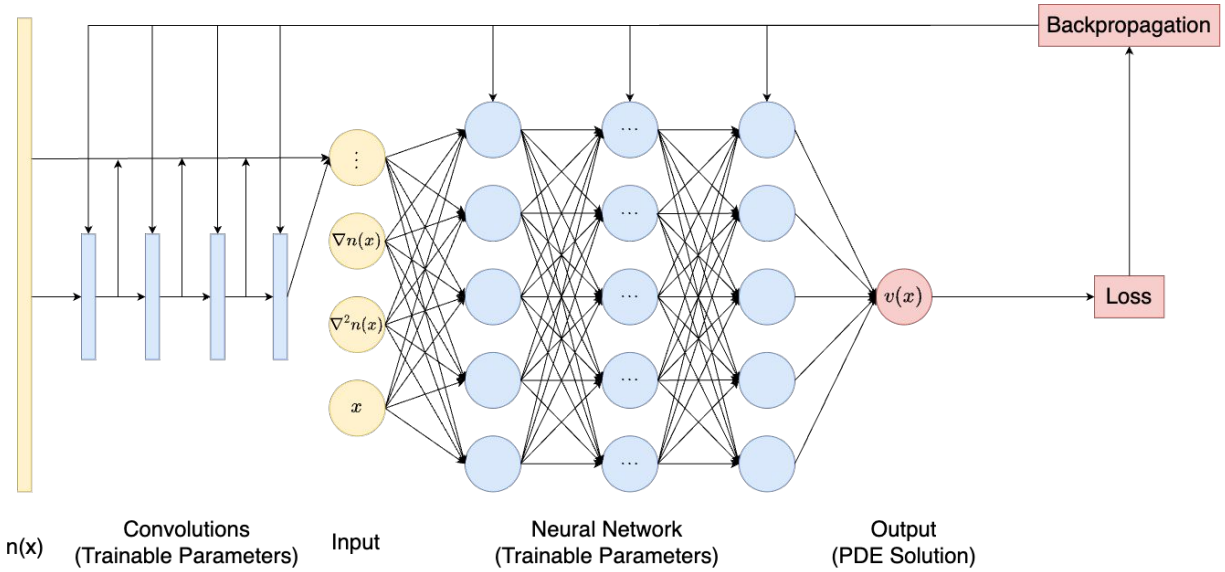
Implementation in terms of PINNs

$$u(x,t) + \mathcal{N}[u;\lambda] = 0$$

$$\tilde{u}(x,t) + \mathcal{N}[\tilde{u};\lambda] := f(x,t)$$

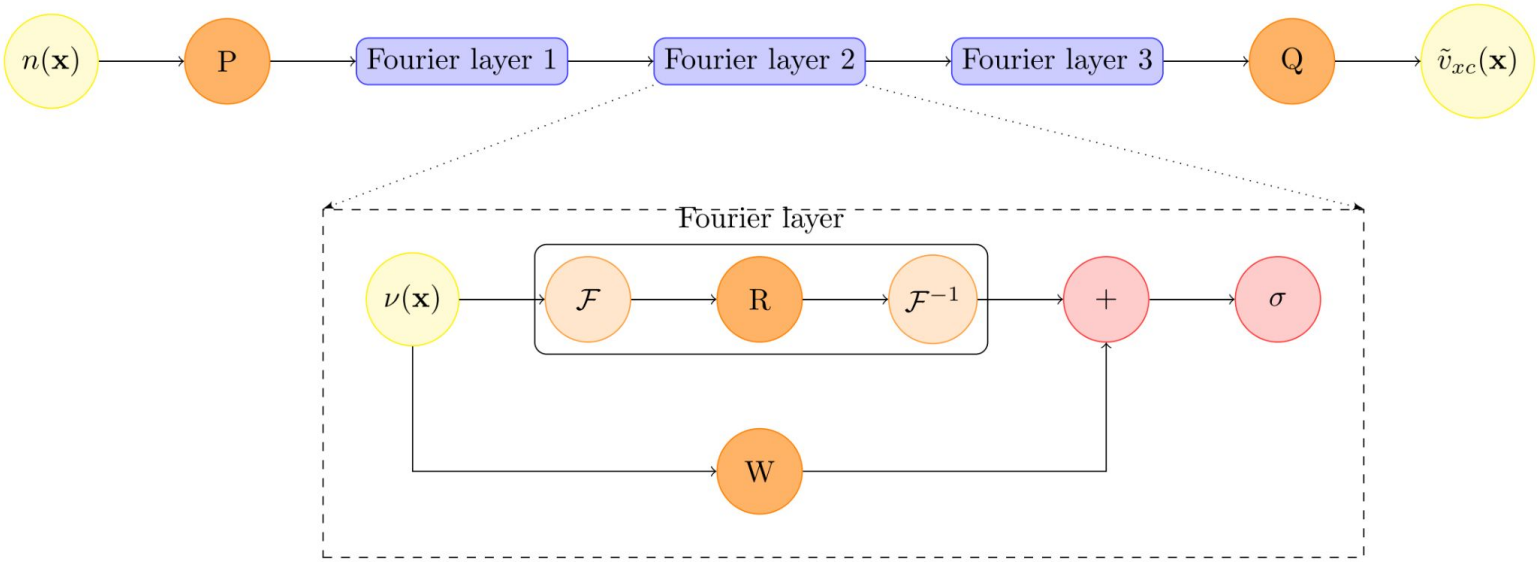
$$f(\mathbf{r}) = \epsilon_i \ \phi_i(\mathbf{r}) + \frac{\nabla^2 \ \phi_i(\mathbf{r})}{2} - v_s(\mathbf{r}) \ \phi_i(\mathbf{r})$$

$$L_{pde} = MSE(f,0) = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| f(x_f^i) \right|^2$$



Inverting the Kohn-Sham Equations with Machine Learning

Implementation in terms of FNOs

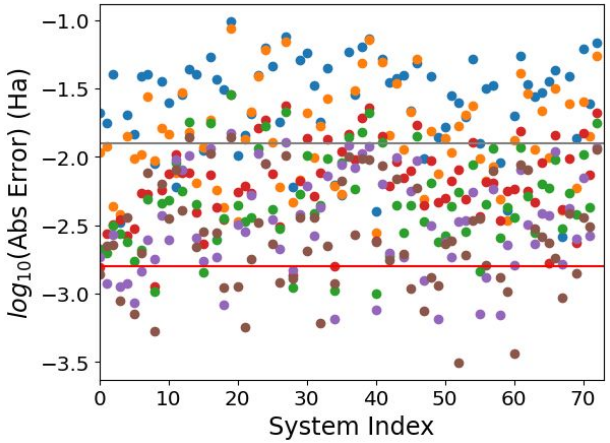


$$L_{FNO} = MSE(v_{\mathrm{XC}}, \tilde{v}_{\mathrm{XC}}) = \frac{1}{N}\sum_{i=1}^{N}\left|v_{\mathrm{XC}}^{(i)} - \tilde{v}_{\mathrm{XC}}^{(i)}\right|^2$$

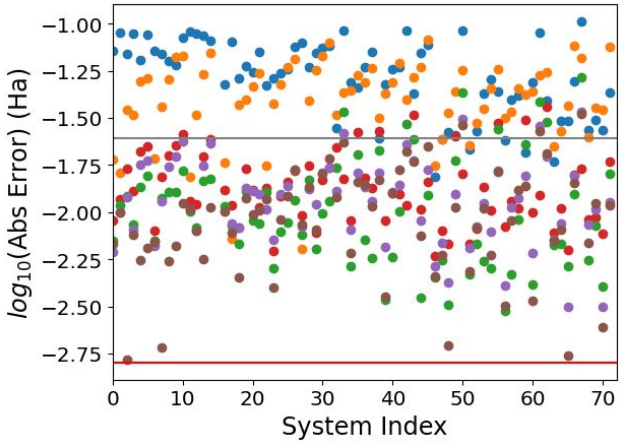
Inverting the Kohn-Sham Equations with Machine Learning

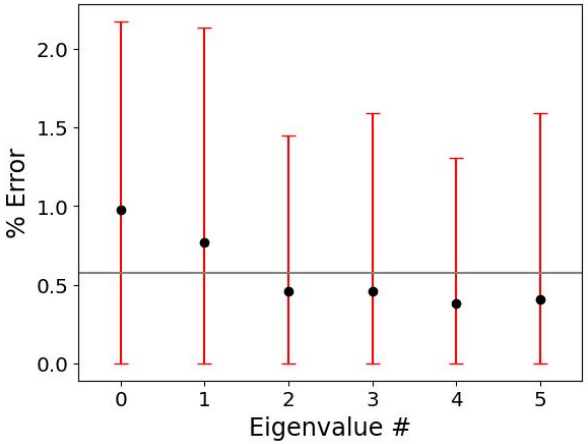
Results

PINNs

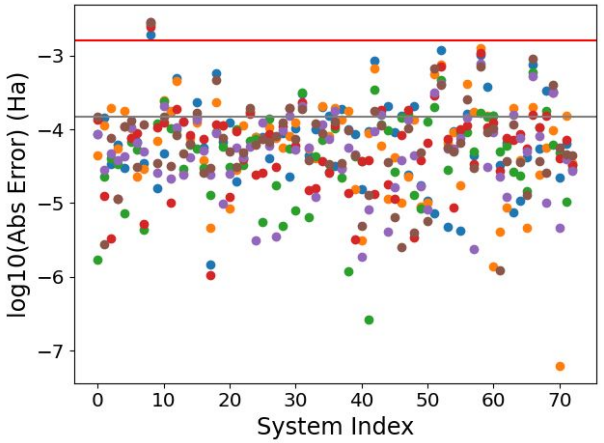


Extrapolation

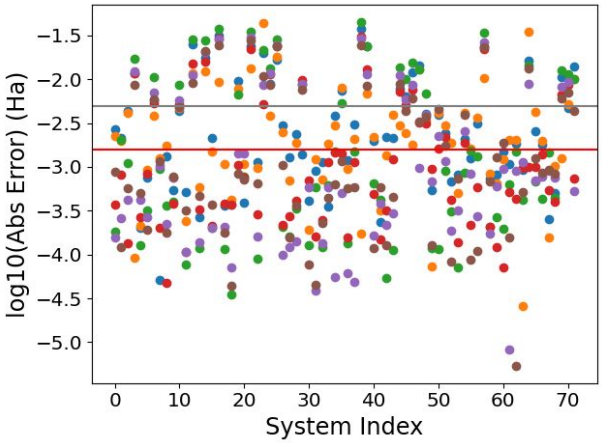


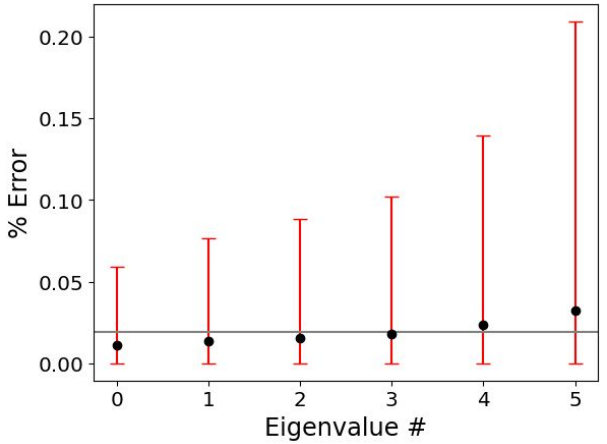


FNOs



Extrapolation





Inverting the Kohn-Sham Equations with Machine Learning

Results: PINNs vs. FNOs

Table 1. Performance comparison of PINN and FNO models on both model systems, showing the MAE (in Ha) and MAPE (%) and denoting the maximum absolute errors and the maximum percentage errors in braces. The labels 301 and 501 indicate the grid resolutions and E denotes extrapolation.

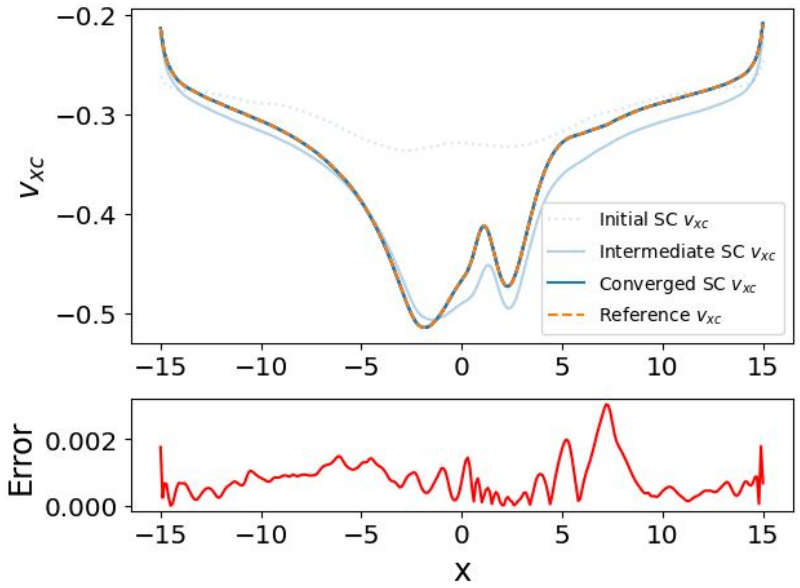
Model	Atom (301)	Atom (501)	Molecule (301)	Molecule (301 E)
PINN	0.0018 Ha *(0.0110 Ha)* 0.13% *(0.30 %)*	0.0007 Ha *(0.0034 Ha)* 0.06% *(0.15%)*	0.0125 Ha *(0.1000 Ha)* 0.58% *(2.37%)*	0.0248 Ha *(0.1200 Ha)* 0.73% *(3.25%)*
FNO	0.0002 Ha *(0.0009 Ha)* 0.03% *(0.15%)*	0.0002 Ha *(0.0010 Ha)* 0.03% *(0.19%)*	0.0002 Ha *(0.0051 Ha)* 0.02% *(0.46%)*	0.0054 Ha *(0.0584 Ha)* 0.25% *(3.51%)*

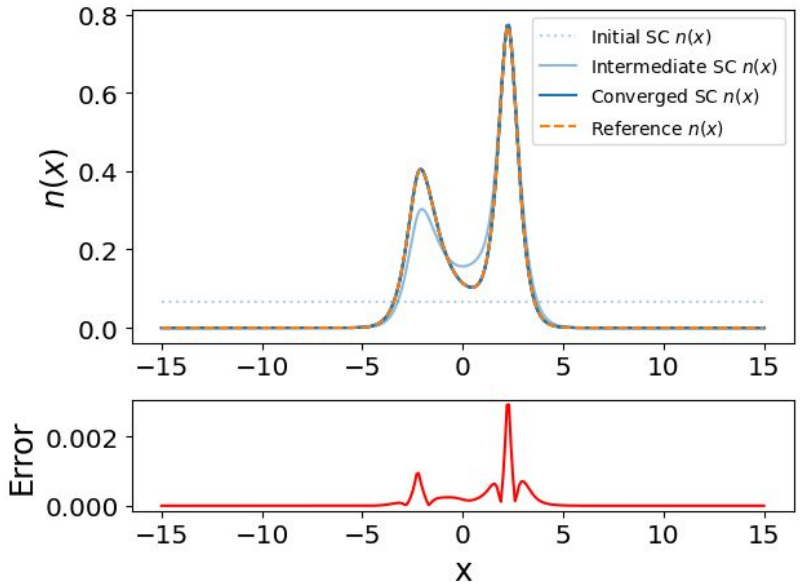
Table 2. Computational timings for inversions (in seconds), comparing a conventional method (iDEA) with the PINN and FNO models.

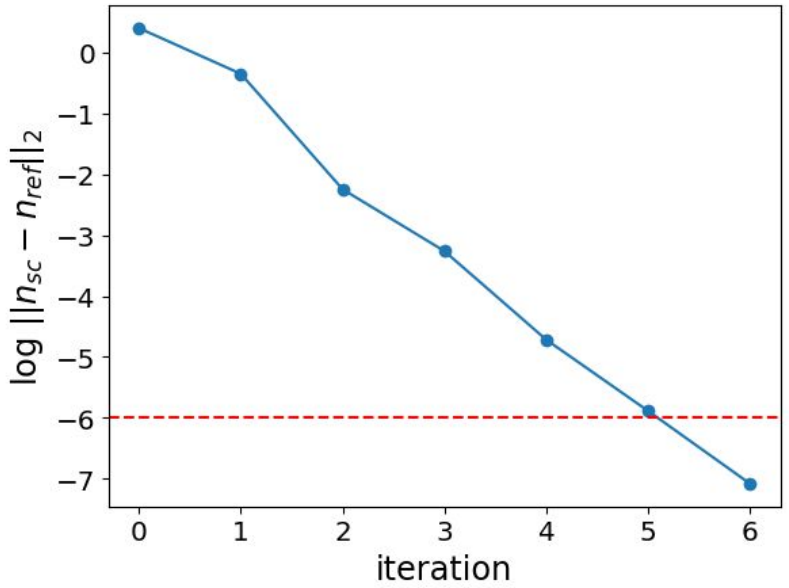
Model	Atom (301)	Molecule (301)	Molecule (501)
iDEA	135 ± 29	305 ± 262	806 ± 447
PINN	0.0016 ± 0.0011	0.0026 ± 0.0013	N/A
FNO	0.0022 ± 0.0009	0.0020 ± 0.0003	0.0037 ± 0.0004

Inverting the Kohn-Sham Equations with Machine Learning

Results: Self-consistent cycle with machine learning model







Outline

Motivation

Theoretical Background

Electronic Structure Problem and Density Functional Theory

Neural Operators

Physics-Informed Neural Networks

Inverting the Kohn-Sham Equations with Machine Learning

Accelerating Electron Dynamics with Machine Learning

Outlook

Accelerating Electron Dynamics with Machine Learning

Summary

Karan Shah

Demonstrate the effectiveness of FNOs in propagating the electron density in time under the TDDFT framework.

Instead of propagating orbitals in time as done conventionally in terms of the time-dependent Kohn-Sham equations, we use the FNO propagator to directly evolve the density.

This has two advantages:

- the computational cost does not scale with the number of orbitals and larger propagation time steps can be used, thus using fewer iterations. Rapid modeling of density evolution of laser-excited molecules and materials in various scattering experiments given just the ground state density and the shape of a laser pulse.
- Design of laser pulses to precisely control quantum dynamics under quantum optimal control theory.

Reference: K. Shah, A. Cangi, ICML Workshop 2024, AI for Science: Scaling in AI for Scientific Discovery, arXiv:2407.09628 (2024).

Accelerating Electron Dynamics with Machine Learning

Model system

$$i\frac{\partial \phi_j(\mathbf{r},t)}{\partial t} = \left[-\frac{1}{2}\nabla^2 + v_s[n](\mathbf{r},t)\right]\phi_j(\mathbf{r},t)$$

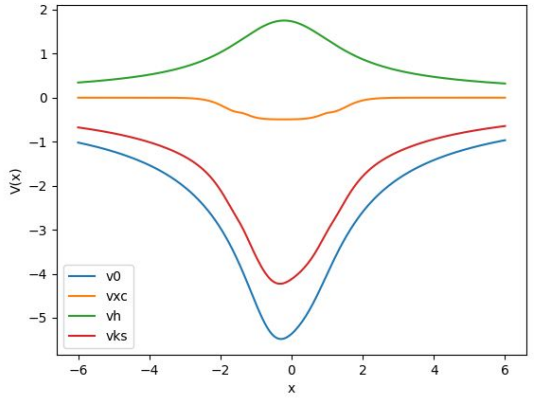
$$n(\mathbf{r},t) = \sum_j |\phi_j(\mathbf{r},t)|^2, \quad j = 1,...,N$$

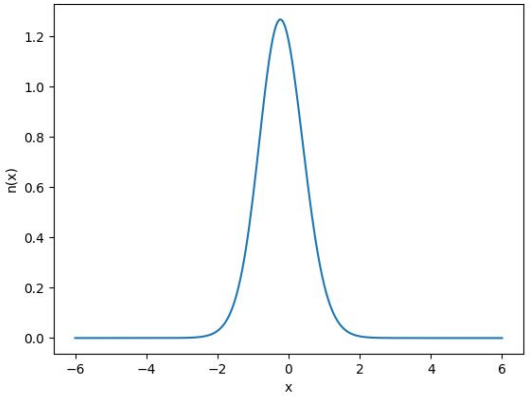
$$v_{\mathrm{S}}[n](\mathbf{r},t) = v_{\mathrm{ext}}(\mathbf{r},t) + v_{\mathrm{H}}[n](\mathbf{r},t) + v_{\mathrm{XC}}[n](\mathbf{r},t)$$

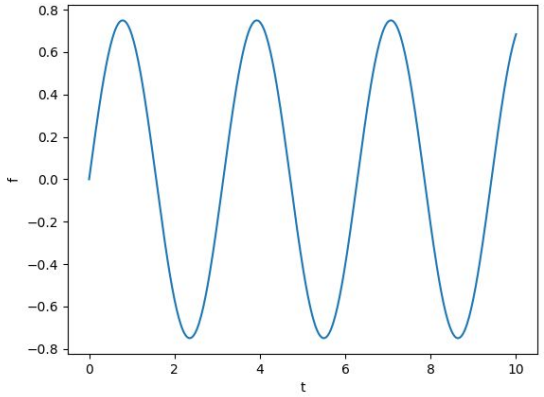
$$v_{\mathrm{ext}}(\mathbf{r},t) = v_{\mathrm{ion}}(\mathbf{r}) + v_{\mathrm{las}}(t)$$

$$v_{\mathrm{ion}}(\mathbf{r}) = -\frac{Z_1}{\sqrt{(\mathbf{r}-\frac{d}{2})^2 + a^2}} - \frac{Z_2}{\sqrt{(\mathbf{r}+\frac{d}{2})^2 + a^2}}$$

$$v_{\mathrm{las}}(t) = A\sin\omega t.$$







Accelerating Electron Dynamics with Machine Learning

Model system

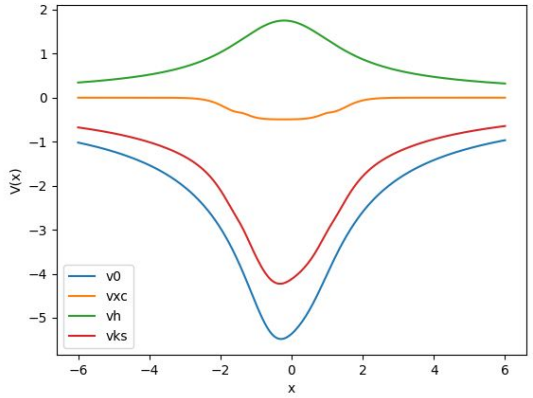
Atoms and diatomic molecules (710 systems)

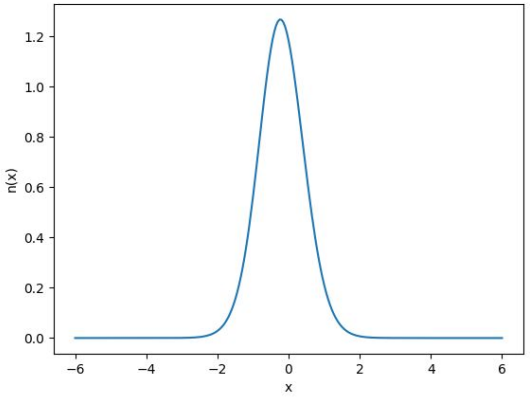
Laser parameters:

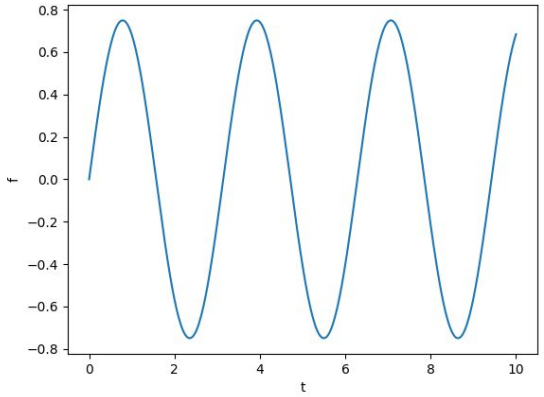
- Intensity: $1.97 \cdot 10^{16}$ W/cm^2
- Wavelength: 22.78 nm
- Time step: 0.2 a.u. (4.83 as)
- 51 time slices each on a grid of 241 spatial points.

Dataset

- 600 systems for training
- 10 systems for validation dataset
- 100 systems for testing







Accelerating Electron Dynamics with Machine Learning

Implementation in terms of FNOs

FNO predicts future values based on past observations.

Input to the model:

$$\mathbf{N}_t = [\mathbf{n}_{t-T_{\text{in}}+1}, \mathbf{n}_{t-T_{\text{in}}+2}, \ldots, \mathbf{n}_t]$$

FNO predicts the next time step:

$$\hat{\mathbf{n}}_{t+1} = \mathcal{G}_\theta(\mathbf{N}_t)$$

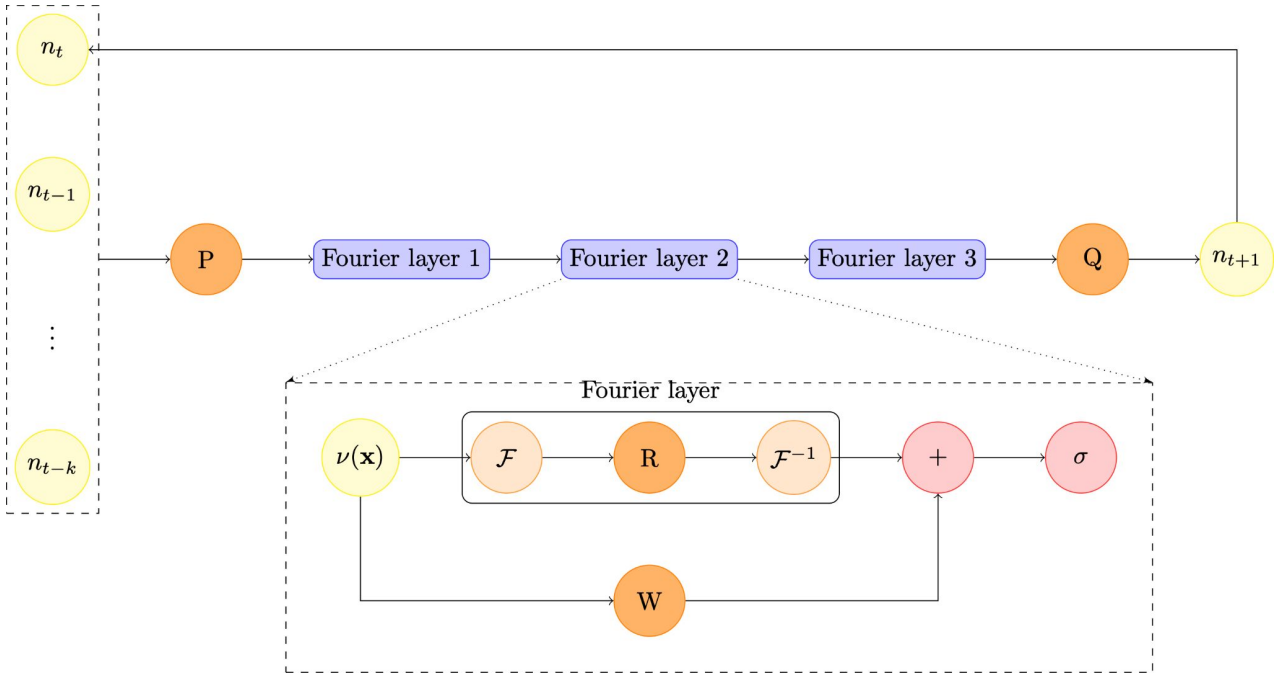
Input sequence is updated for the next prediction:

$$\mathbf{N}_{t+1} = [\mathbf{n}_{t-T_{\text{in}}+2}, \mathbf{n}_{t-T_{\text{in}}+3}, \ldots, \mathbf{n}_t, \hat{\mathbf{n}}_{t+1}]$$

FNO is trained with density slices (entire spatial domain) in time windows with a given width sweeping across the entire time domain, minimizing the loss :

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left\| \mathbf{n}_t^{(d)} - \hat{\mathbf{n}}_t^{(d)} \right\|_2^2$$

$$\mathcal{L}_{norm}(\theta) = \lambda \left(\sum_i \hat{n}_{t,i}^{(d)} \Delta x - 2 \right)^2$$

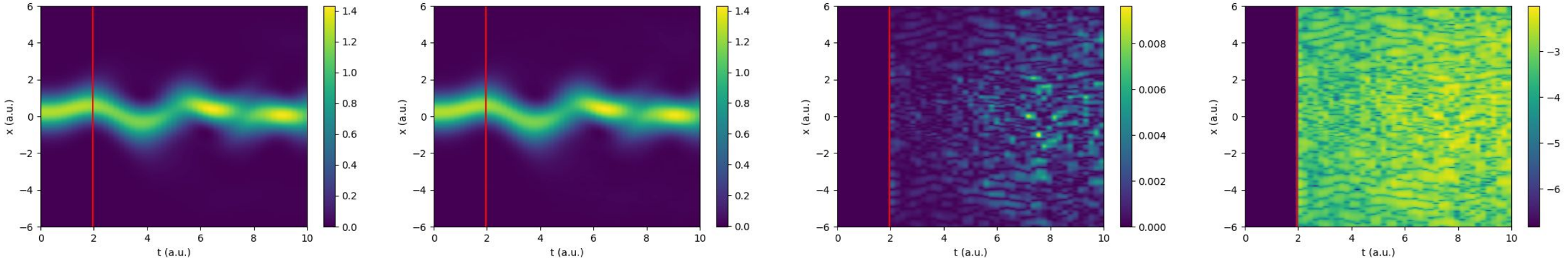


Accelerating Electron Dynamics with Machine Learning

Results

Reference

Prediction

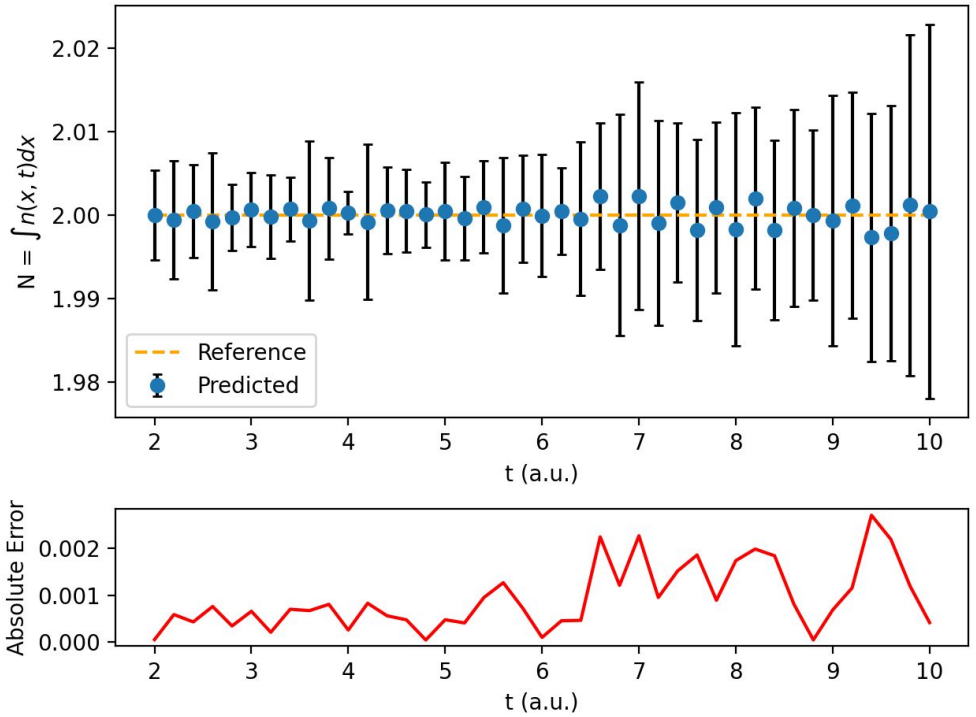


Model	MAE ($\times 10^{-3}$)	MSE ($\times 10^{-4}$)	Time (ms)
FNO $T_{in} = 05$	8.595	4.18	1.58
FNO $T_{in} = 10$	7.087	4.10	1.62
FNO $T_{in} = 15$	6.101	3.16	1.60
FNO $T_{in} = 20$	5.652	3.32	1.60
Baseline	5.118	2.56	1.75
Octopus (Coarse)	19.616	24.69	4.00
Time Offset	5.779	2.78	1.75
Super-resolution	7.335	6.56	1.75

Accelerating Electron Dynamics with Machine Learning

Tests: Density conservation

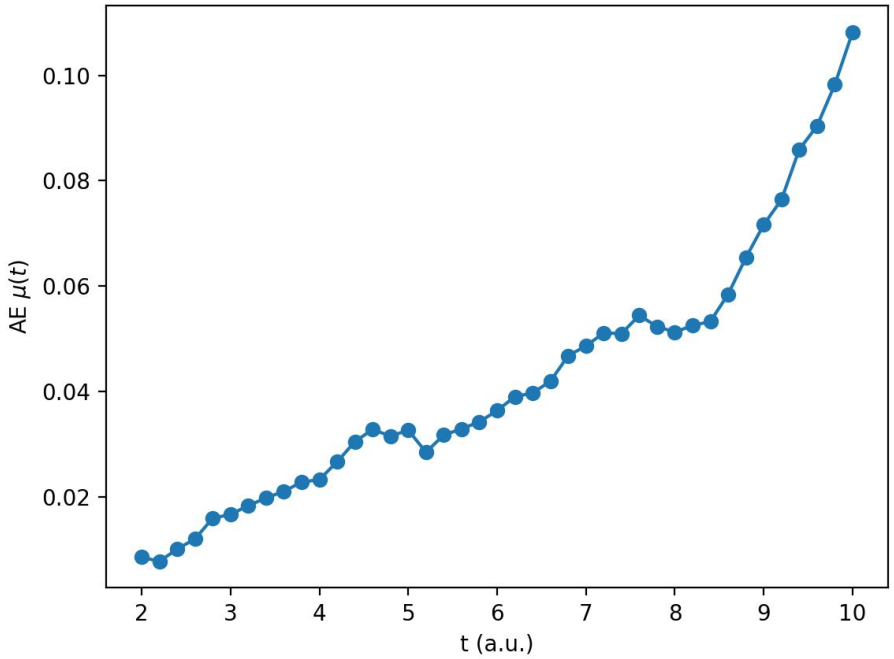
$$N = \int_{-\infty}^{\infty} n(x,t)dx$$



Accelerating Electron Dynamics with Machine Learning

Tests: Dipole moment

$$\mu(t) = \int_{-\infty}^{\infty} x\, n(x,t)\, dx \approx \sum_i x_i\, n_{t,i}\, \Delta x$$



Thank you for your attention

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

Thanks to my team and collaborators!

Attila Cangi, Tom Jungnickel, Bartosz Brzoza
Karan Shah, Lenz Fiedler, Timothy Callow
Kushal Ramakrishna, Uwe Hernandez-Acosta, Hossein Tahmasbi

HELMHOLTZ AI
HiDA HELMHOLTZ Information & Data Science Academy
HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF

Mani Lokamani, Steve Schmerler,
Peter Steinbach, Guido Juckeland

UNIVERSITY OF CALIFORNIA MERCED

Vincent Martinetto,
Aurora Pribram-Jones