

Accelerating Electron Dynamics Simulations through Machine Learned Time Propagators

Psi-K 2025 Talk B1.24

August 26, 2025 // Karan Shah^{1,2} Attila Cangi^{1,2}, ¹CASUS ²HZDR



```
    mirror_object to mirror
    mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y"
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z"
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
```



Motivation

Methods

Results

References

Motivation

Methods

Results

References

Real-time propagation:

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

where

$$n(\mathbf{r}, t) = \sum_i |\phi_i(\mathbf{r}, t)|^2.$$

- $v_s[n](\mathbf{r}, t)$ includes time-dependent external, Hartree, and exchange-correlation potentials.
- Numerically integrate this system in time to get evolving densities and orbitals.
[1]

- Real-time (RT) TDDFT: propagate $\phi_j(\mathbf{r}, t)$ in small time steps.
- Common methods: Crank-Nicolson, exponential midpoint, high-order splitting, etc.
- These require many small Δt steps for numerical stability.

- Real-time (RT) TDDFT: propagate $\phi_j(\mathbf{r}, t)$ in small time steps.
- Common methods: Crank-Nicolson, exponential midpoint, high-order splitting, etc.
- These require many small Δt steps for numerical stability.
- **Goal:** Learn a direct mapping from $n(\mathbf{r}, t)$ at past times to $n(\mathbf{r}, t + \Delta t)$, bypassing orbital-level propagation.
- Fewer, larger steps while preserving key observables and stable rollouts.
- Enable on-the-fly predictions under varying laser fields, especially useful for Quantum Optimal Control [2].

Contents

Motivation

Methods

Results

References

Autoregressive FNO

- At each time t , take T_{in} past density *snapshots* and laser potentials ν_l as input.
- FNO outputs $n(\mathbf{r}, t + 1)$.
- Append predicted density to form new input for next time step.

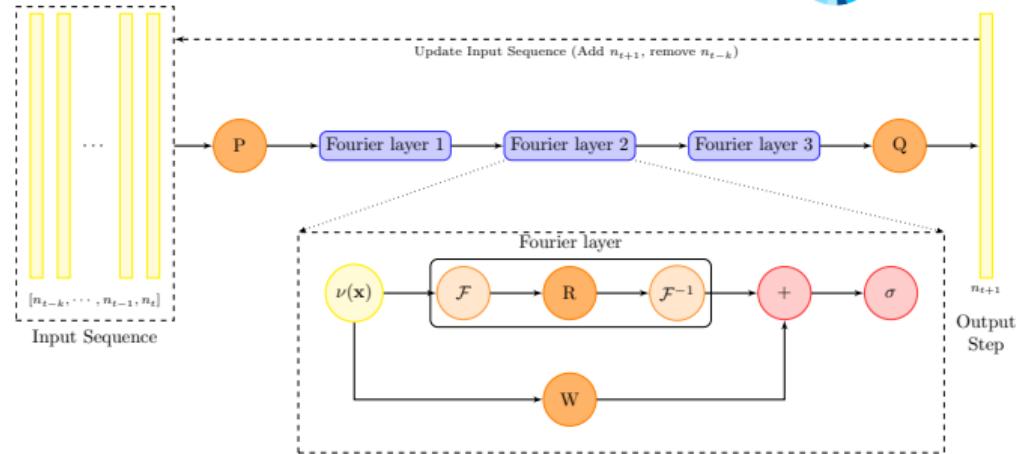


Fig. 1: Autoregressive FNO for density propagation.

Autoregressive FNO

- At each time t , take T_{in} past density *snapshots* and laser potentials ν_t as input.
- FNO outputs $n(\mathbf{r}, t + 1)$.
- Append predicted density to form new input for next time step.

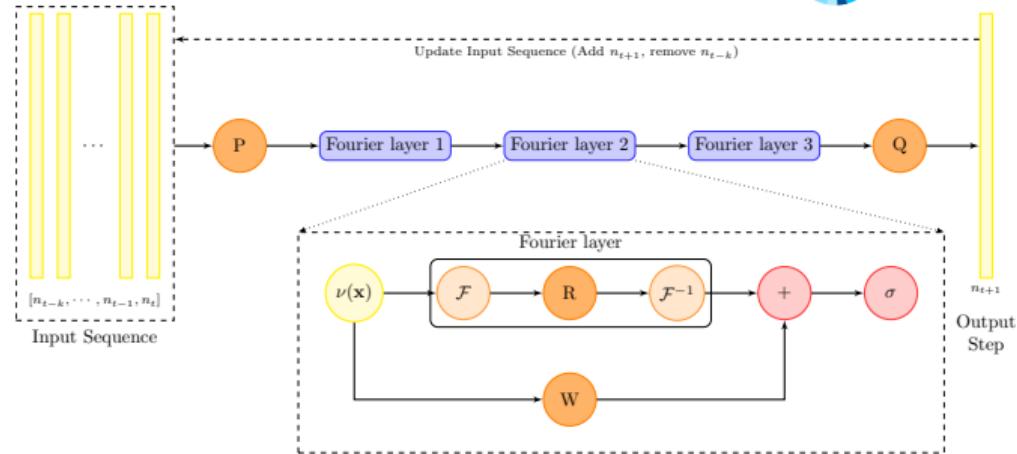


Fig. 1: Autoregressive FNO for density propagation.

Advantages[3]:

- Learns solution operators, naturally aligned with TDDFT propagation.
- Captures oscillatory quantum dynamics via global Fourier modes.
- Generalizes across spatial discretizations.

- L2 Loss:

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

- Density norm conservation:

$$N = \int_{-\infty}^{\infty} n(x, t) dx \approx \sum_i \hat{n}_{t+1, i}^{(d)} \Delta x$$

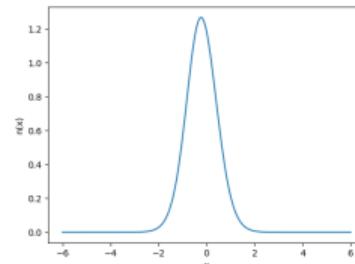
$$\mathcal{L}_{norm}(\theta) = \lambda \left(\sum_i \hat{n}_{t+1, i}^{(d)} \Delta x - 2 \right)^2$$

One-Dimensional Model Systems

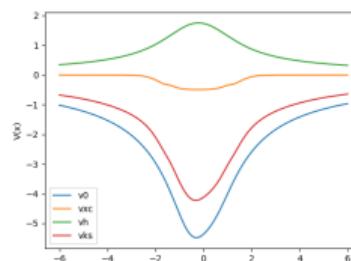
- Diatomic molecule in 1D with 2 electrons:

$$v_{\text{ion}}(x) = -\frac{Z_1}{\sqrt{(x - \frac{d}{2})^2 + a^2}} - \frac{Z_2}{\sqrt{(x + \frac{d}{2})^2 + a^2}}.$$

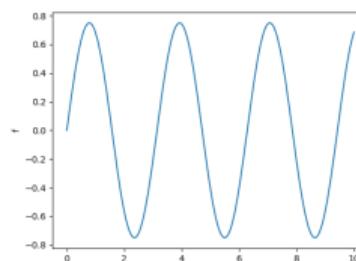
- Laser perturbation:
 $v_{\text{las}}(t) = A \sin(\omega t)$.
- Soft Coulomb potential.
- Parameters varied:
 d, Z_1, Z_2, A, ω .
- Domain: $[-9, 9]$ au, $\Delta x = 0.05$ au, $T = 5$ fs, $\Delta t = 0.1$ fs.
- Generated using Octopus [4].



(a) Ground State Density



(b) Ground State Potentials



(c) Laser Field

Contents

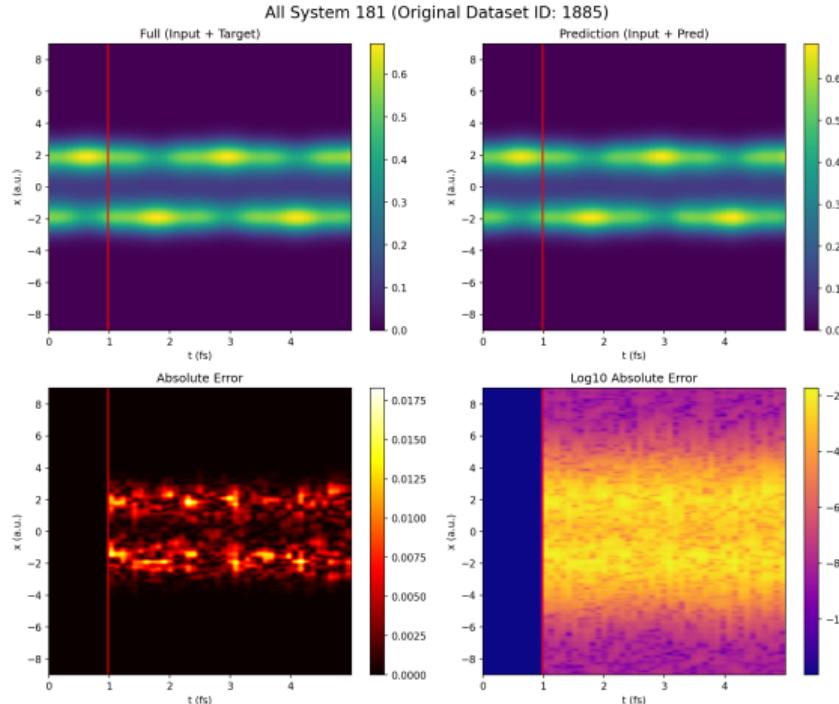
Motivation

Methods

Results

References

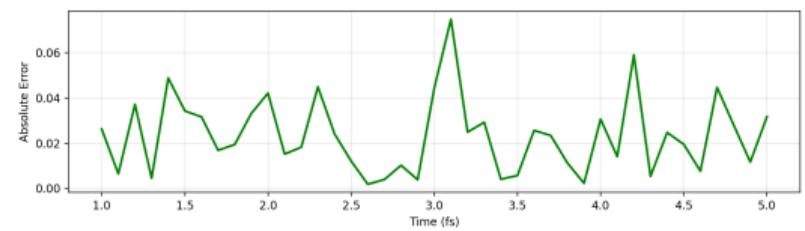
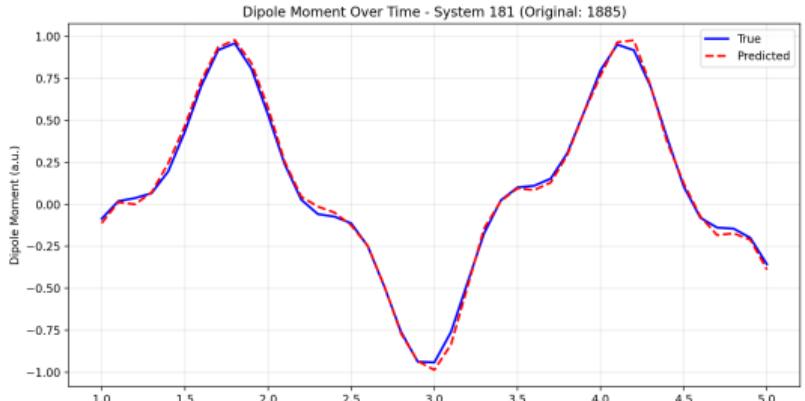
Density Evolution



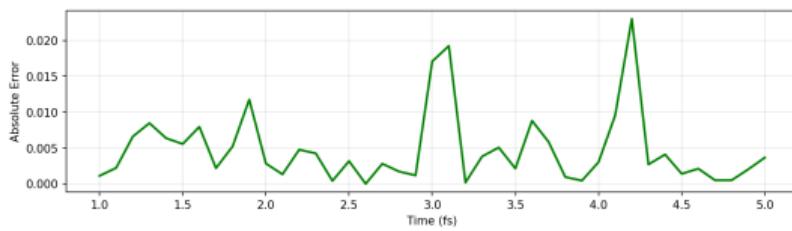
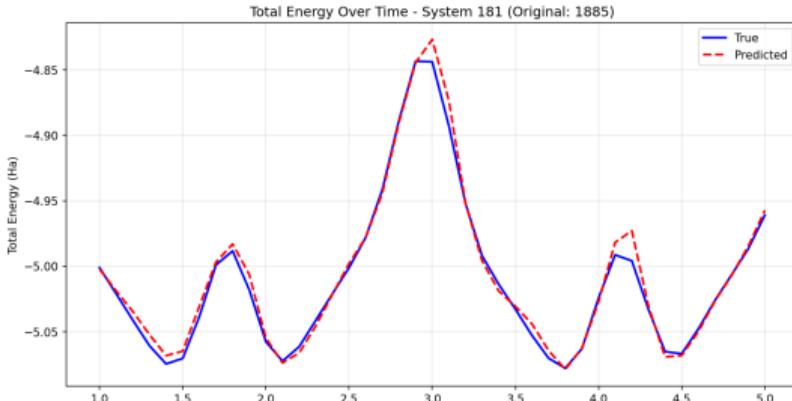
Red line indicates the boundary between input slices (T_{in}) and predicted slices.

Results generated with fully connected layer encoding for densities and lasers.

Observables



Dipole moment calculated from reference and predicted density



Total TF energy calculated from reference and predicted density. Calculated

using Thomas-Fermi approximation, with $T_{TF} = \frac{\pi^2}{6} \int n(x)^3 dx$

Results:

Table 1: Comparison of error metrics across models.

Metric	FNO	Octopus Coarse
AE (10^{-3})	0.872	6.10
MSE (10^{-5})	3.20	31.2
MAPE (%)	1.84	853
SMAPE (%)	1.73	62.3
Dipole MSE (10^{-3})	7.37	79.0
Dipole MAPE (%)	9.35	74.0
Dipole SMAPE (%)	6.08	51.1
Integral (≈ 2)	1.9995 ± 0.0014	$2.0000 \pm 2.0\text{e-}7$
Inference Time (ms)	3.07 ± 0.05	23.94 ± 2.19

Generalization: Time Extension and Super-Resolution



- **Time Extension:** Model trained with domain $T \in [0, 5]$ fs.
Error does not accumulate over larger time scales (10 fs).

Generalization: Time Extension and Super-Resolution

- **Time Extension:** Model trained with domain $T \in [0, 5]$ fs.
Error does not accumulate over larger time scales (10 fs).
- **Super-resolution in space:** FNO trained at 361-point space grid ($\Delta x = 0.05$ au) can be used to infer on a 721-point grid ($\Delta x = 0.025$ au) with only a modest increase in error and inference time.

Generalization: Time Extension and Super-Resolution

- **Time Extension:** Model trained with domain $T \in [0, 5]$ fs.
Error does not accumulate over larger time scales (10 fs).
- **Super-resolution in space:** FNO trained at 361-point space grid ($\Delta x = 0.05$ au) can be used to infer on a 721-point grid ($\Delta x = 0.025$ au) with only a modest increase in error and inference time.
- This invariance is a key advantage of the neural operator approach.

Results: Generalization

Table 2: Comparison of error metrics across FNO, Spatial Superresolution, and Time Extension.

Metric	FNO	Spatial Super-resolution	Time Extension
AE (10^{-3})	0.872	0.942	0.896
MSE (10^{-5})	3.20	3.30	2.53
MAPE (%)	1.84	2.06	2.65
SMAPE (%)	1.73	1.95	2.18
Dipole MSE (10^{-3})	7.37	7.64	5.45
Dipole MAPE (%)	9.35	13.10	10.03
Dipole SMAPE (%)	6.08	7.22	6.29
Integral (≈ 2)	1.9995 ± 0.0014	1.9966 ± 0.0014	1.9995 ± 0.0010
Inference Time (ms)	3.07 ± 0.05	3.37 ± 0.05	3.05 ± 0.04

Physical Properties of Predicted Densities

Density conservation:

$$N = \int n(x, t) dx = 2 \quad (\text{for 2 electrons}).$$

FNO has an additional loss term promoting $\sum_i \hat{n}_i \Delta x \approx 2$.

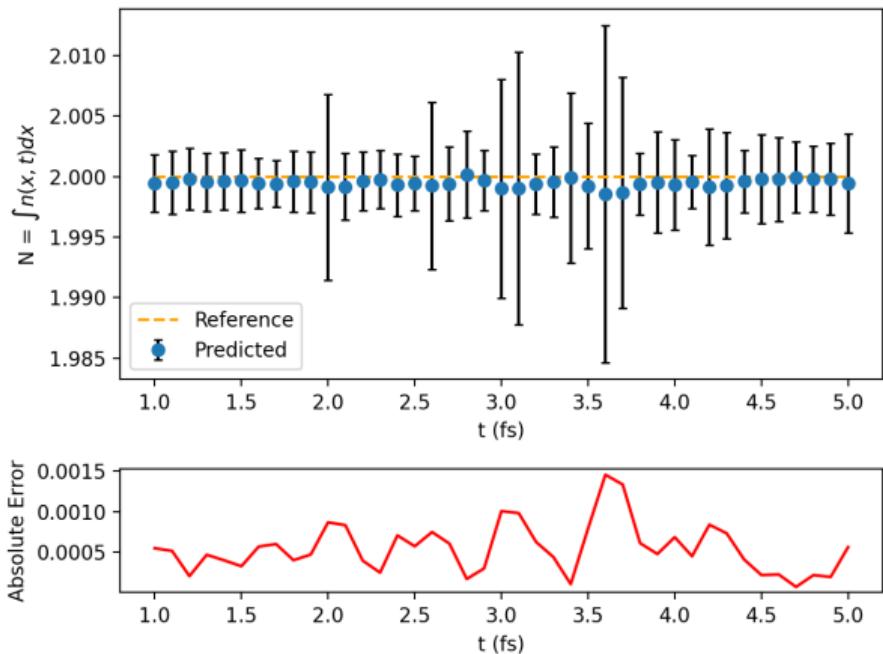


Fig. 3: Evolution of predicted density norm vs. time.

- **Time-reversal symmetry:** Unitary time-propagation is ideally reversible, but FNO predictions degrade if we reverse time order.

- **Time-reversal symmetry:** Unitary time-propagation is ideally reversible, but FNO predictions degrade if we reverse time order.
- **1D Nature:** Current work is limited to 1D systems.

- **Time-reversal symmetry:** Unitary time-propagation is ideally reversible, but FNO predictions degrade if we reverse time order.
- **1D Nature:** Current work is limited to 1D systems.
- **Physical constraints:** Could incorporate more PDE constraints or use physics-informed neural operator frameworks.

Conclusion

- ML time propagators (FNOs) accelerate TDDFT calculations
 - Predict time evolution given initial density and laser pulse
 - Maintain physical consistency
 - Resolution-invariant

- ML time propagators (FNOs) accelerate TDDFT calculations
 - Predict time evolution given initial density and laser pulse
 - Maintain physical consistency
 - Resolution-invariant
- Future work:
 - Incorporate more physical constraints such as continuity and explicit time-reversal symmetry
 - Generalize to 3D

Acknowledgements

- Attila Cangi
- MLMD: Lenz Fiedler, Tim Callow, Bartosz Brzoza, Kushal Ramakrishna
- HZDR: Patrick Stiller, Nico Hoffmann, Lokamani
- CASUS: Juan Esteban Suarez Cardona, Michael Hecht
- UC Merced: Vincent Martinetto, Aurora Pribram-Jones
- Helmholtz AI Association

Machine Learning Time Propagators for Time-Dependent Density Functional Theory Simulations

Karan Shah, Attila Cangi

arxiv.org/abs/2508.16554



Contents

Motivation

Methods

Results

References



CASUS
CENTER FOR ADVANCED
SYSTEMS UNDERSTANDING

www.casus.science



Contents

Motivation

Methods

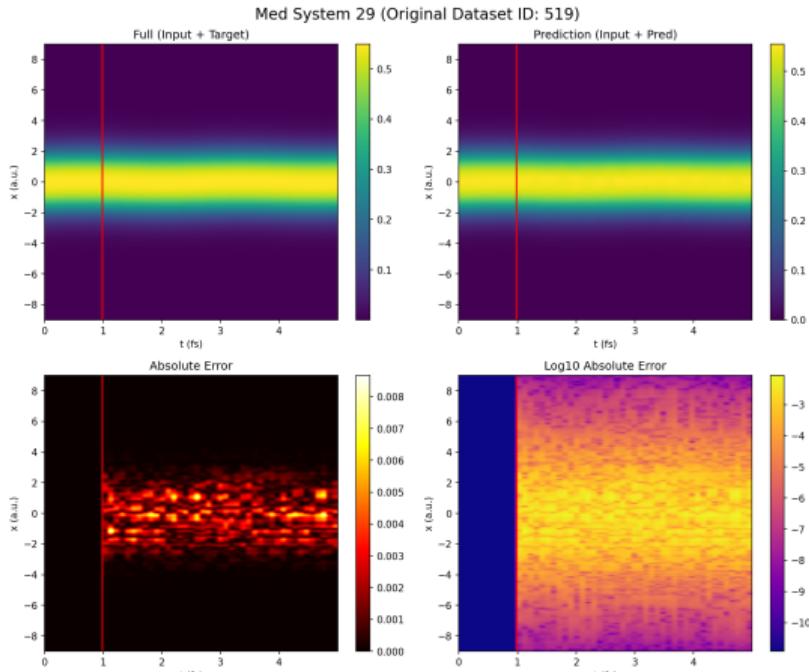
Results

References

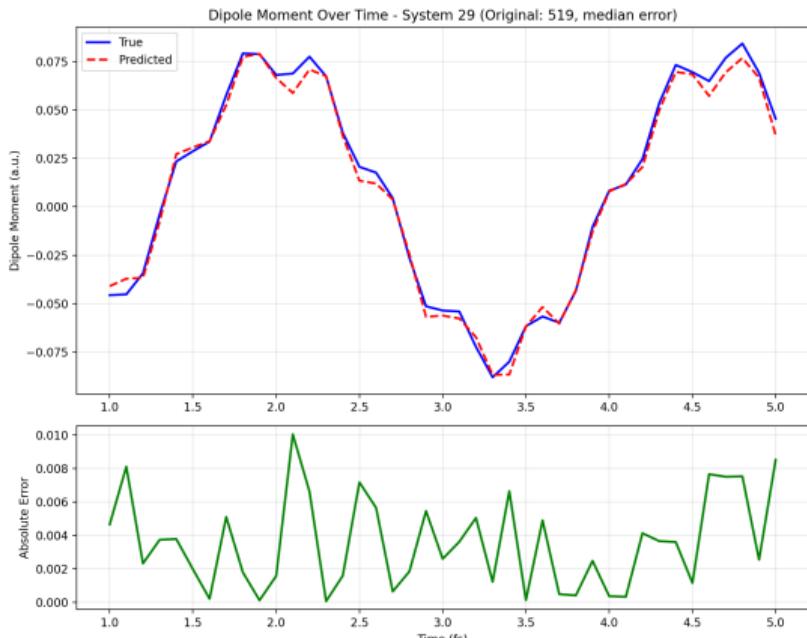
- [1] Alberto Castro, Miguel A. L. Marques, and Angel Rubio. "Propagators for the time-dependent Kohn-Sham equations". In: *The Journal of Chemical Physics* 121.8 (Aug. 2004), pp. 3425–3433. ISSN: 0021-9606. DOI: 10.1063/1.1774980. URL: <https://doi.org/10.1063/1.1774980> (visited on 05/21/2024).
- [2] J. Werschnik and E. K. U. Gross. "Quantum optimal control theory". en. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 40.18 (Sept. 2007), R175. ISSN: 0953-4075. DOI: 10.1088/0953-4075/40/18/R01. URL: <https://dx.doi.org/10.1088/0953-4075/40/18/R01> (visited on 05/24/2024).
- [3] Kamyar Azizzadenesheli et al. "Neural operators for accelerating scientific simulations and design". en. In: *Nature Reviews Physics* 6.5 (May 2024). Publisher: Nature Publishing Group, pp. 320–328. ISSN: 2522-5820. DOI: 10.1038/s42254-024-00712-5. URL: <https://www.nature.com/articles/s42254-024-00712-5> (visited on 05/21/2024).
- [4] Nicolas Tancogne-Dejean et al. "Octopus, a computational framework for exploring light-driven phenomena and quantum dynamics in extended and

finite systems". In: *The Journal of Chemical Physics* 152.12 (Mar. 2020), p. 124119.
ISSN: 0021-9606. DOI: 10.1063/1.5142502. URL:
<https://doi.org/10.1063/1.5142502> (visited on 05/22/2024).

Additional Results: Baseline

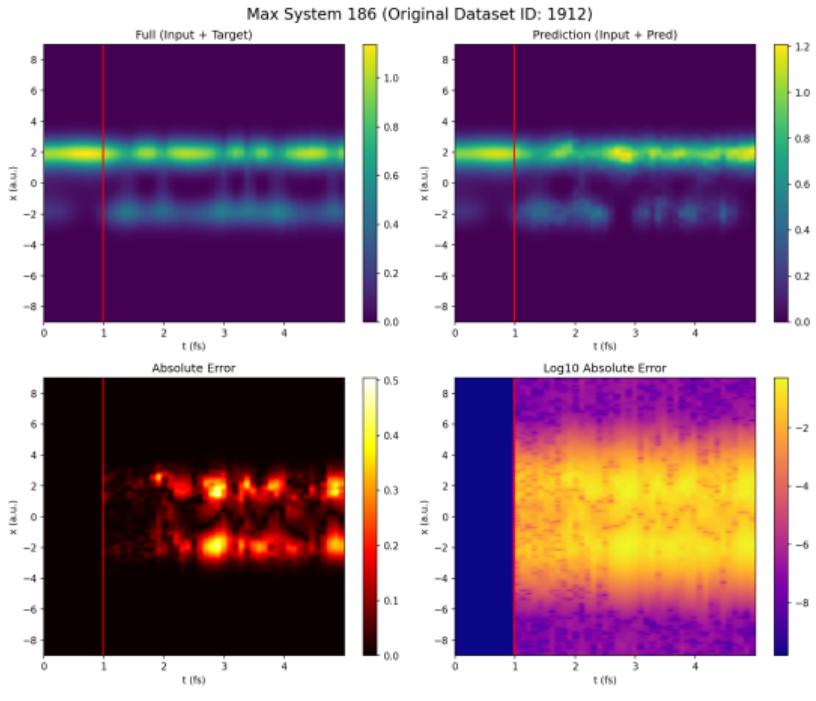


Snapshot

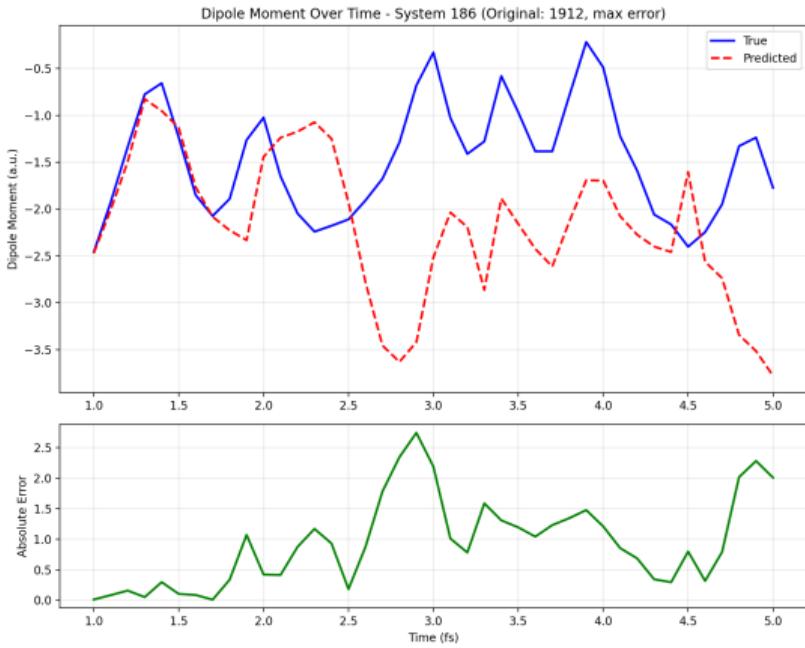


Dipole Moment

Additional Results: Limitations

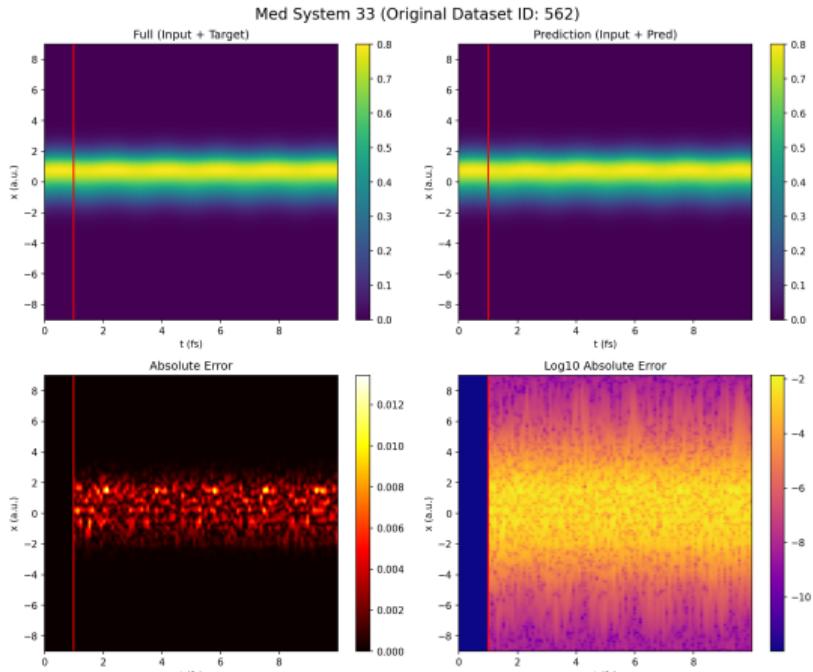


Snapshot

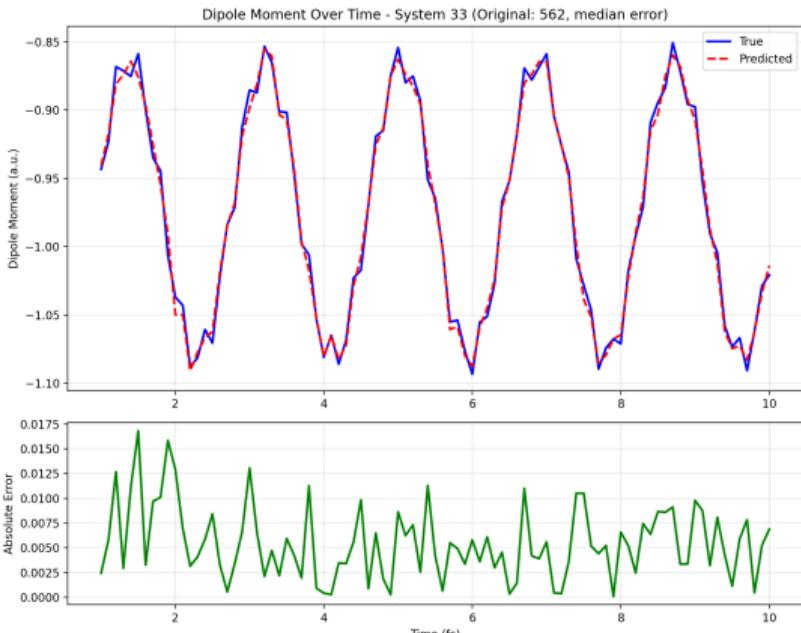


Dipole Moment

Additional Results: Time Extension

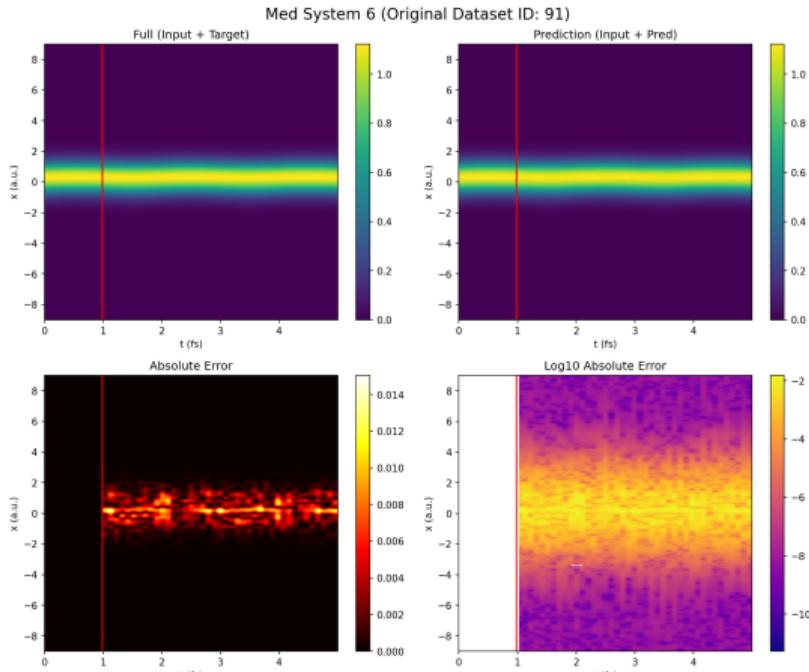


Snapshot

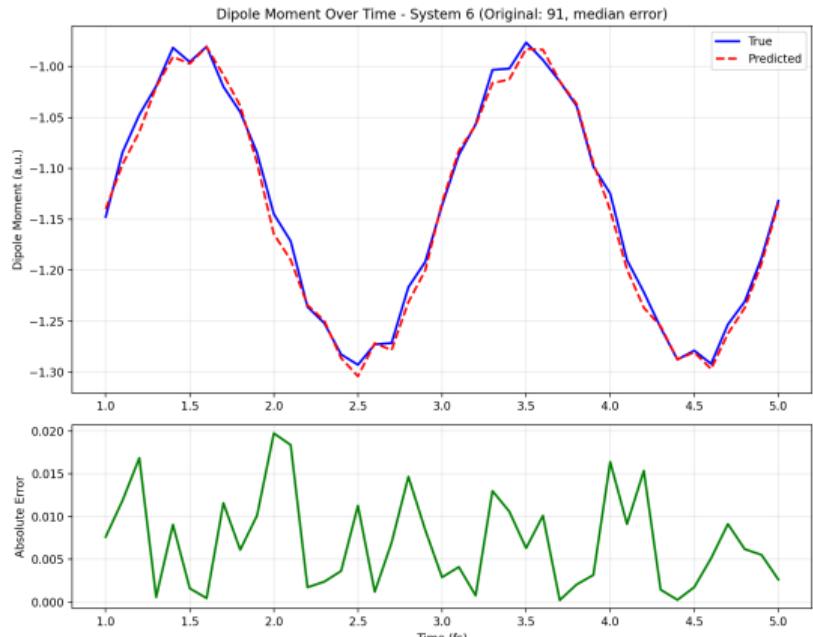


Dipole Moment

Additional Results: Spatial Super-resolution



Snapshot



Dipole Moment

- Diatomic Molecule (double well):

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

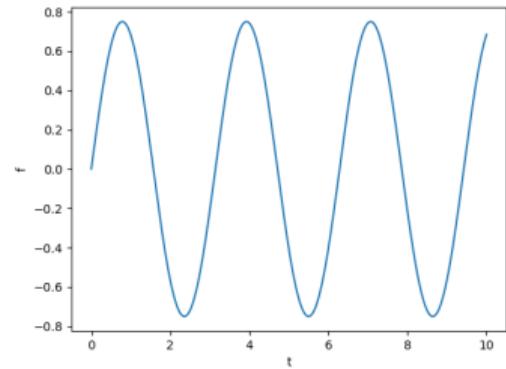
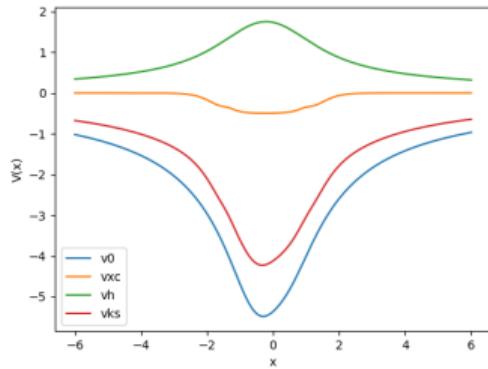
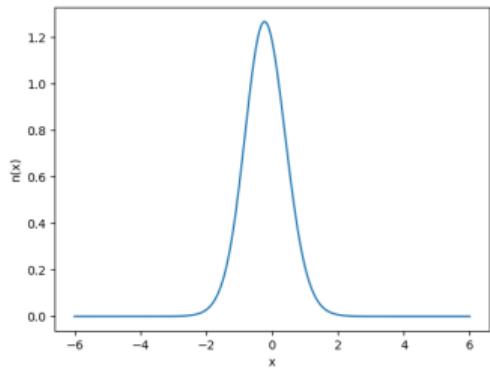
- With time dependent laser (under dipole approximation)
- $v_{\text{las}}(\mathbf{r}, t) = A \sin \omega t$

Training data:

- Radius: [-9, 9] au, $\Delta x = 0.05$ au
- Time: [0, 5] fs, $\Delta t = 0.1$ fs for ML, 0.01 fs for reference data generation
- Laser: Wavelength 400-750 nm (optical range)
- Intensity $10^{12} - 10^{14}$ W/cm²

Time Propagation Systems (cont.)

- Errors:
 - MSE, MAE over density
 - Dipole moment, Density norm conservation, TF total energy



Hyperparameters & Training Details

- **Data:** $S = 361$, $T = 41$, $T_{\text{in}} = 10$, $T_{\text{total}} = 51$, $\Delta x = 0.05 \text{ au}$, $\Delta t = 0.1 \text{ fs}$
- **Training:**
 - Epochs: 800, Batch size: 40
 - Train/Val/Test split: 800 / 150 / 200 (out of 2048)
 - Optimizer: AdamW, Learning rate: 1×10^{-3} , Weight decay: 1×10^{-4}
 - Scheduler: CosineAnnealingLR ($T_{\text{max}} = 1000$, $\eta_{\text{min}} = 1 \times 10^{-5}$)
- **Model:** Fourier Neural Operator
 - Width: 128, Layers: 3, Modes: 32
 - Input representation: separate FC add (default)
 - Padding: 40 (for finite boundary conditions)
 - Use grid features: true
- **Losses:**
 - MSE (density), with optional log transform
 - Integral loss enabled (weight = 0.1)

- **Density-Only:** Use only the density history as input. A single fully connected layer maps to model width.
- **Direct Concatenation:** Concatenate density, laser potential, and grid features directly, then apply a fully connected layer.
- **Separate + Concatenation:** Process density and potential through independent fully connected layers, then concatenate the outputs.
- **Separate + Addition (Default): Process density and potential through independent fully connected layers, then add the outputs.**
- **Space-Time Convolution:** Reshape density and potential as a spatio-temporal field, apply 2D convolution, temporal pooling, and a linear grid embedding.

Results: Input Representations

Table 3: Comparison of error metrics across input representations.

Metric	Density Only	Concat	FC Add	FC Concat	Spacetime Conv
AE (10^{-3})	1.97	0.899	0.872	0.879	1.48
MSE (10^{-5})	6.15	3.13	3.20	3.27	5.08
MAPE (%)	3.71	1.90	1.84	1.85	3.80
Dipole MSE (10^{-3})	11.4	6.75	7.37	7.27	8.92
Dipole MAPE (%)	27.6	18.6	9.35	10.7	27.0
Integral (≈ 2)	1.9990	2.0000	1.9995	2.0000	2.0005
Inference Time (ms)	1.33	1.32	1.35	1.38	1.51

- Laser-aware encodings outperform density-only baseline.

Algorithm 1 Real-Time TDDFT Propagation

- 1: Initialize $\phi_i(\mathbf{r}, 0)$ from ground-state DFT or another initial condition.
 - 2: **for** $t = 0, 1, 2, \dots$ until final time T **do**
 - 3: Compute $v_s^t(\mathbf{r}) = v(\mathbf{r}, t) + v_H[n^t] + v_{xc}[n^t]$.
 - 4: Propagate orbitals $\phi_i^t \rightarrow \phi_i^{t+1}$ using a time integrator (e.g., Crank-Nicolson, exponential integrator).
 - 5: Update $n^{t+1}(\mathbf{r}) = \sum_i |\phi_i^{t+1}(\mathbf{r})|^2$.
 - 6: $t_{n+1} = t_n + \Delta t$.
 - 7: **end for**
-

Numerical methods:

- **Time-step selection:** must be small enough to capture electron dynamics.
- **Propagation schemes:** split-operator, Magnus expansions, etc.

Definition

$$\hat{U}(t, t_0) = \mathcal{T} \exp \left(-i \int_{t_0}^t \hat{H}(t') dt' \right)$$

Key Properties

- Unitarity: $\hat{U}^\dagger(t, t_0) \hat{U}(t, t_0) = \mathbb{I}$
- Group Property:
 $\hat{U}(t_2, t_0) = \hat{U}(t_2, t_1) \hat{U}(t_1, t_0)$
- Time Reversal Symmetry:
 $\hat{U}(t_2, t_1) = \hat{U}^\dagger(t_1, t_2)$

Numerical Implementation

- Magnus expansion for accurate evolution
- Preserves norm of KS wavefunctions
- Ensures causality in electron dynamics

Iterative procedure for time-series prediction:

Algorithm 2 Autoregressive FNO Iteration

- 1: **Given:** $\{n(\mathbf{r}, 1), n(\mathbf{r}, 2), \dots, n(\mathbf{r}, T_{\text{in}})\}$
- 2: **Given:** laser potentials $\{v_l(1), v_l(2), \dots, v_l(T)\}$
- 3: **Given:** $\mathcal{G} : \underbrace{\mathcal{X} \times \dots \times \mathcal{X}}_{T_{\text{in}}} \times \mathcal{V} \rightarrow \mathcal{X}$ (FNO operator)

- 4: **for** $t = T_{\text{in}}, T_{\text{in}} + 1, \dots, T - 1$ **do**
- 5: **Form input:**

$$(n(\mathbf{r}, t - T_{\text{in}} + 1), \dots, n(\mathbf{r}, t), v_l(t))$$

- 6: **Predict next snapshot:**

$$\hat{n}(\mathbf{r}, t + 1) = \mathcal{G}(n(\mathbf{r}, t), \dots, n(\mathbf{r}, t - T_{\text{in}} + 1), v_l(t))$$

- 7: **Update state:**

$$n(\mathbf{r}, t + 1) \leftarrow \hat{n}(\mathbf{r}, t + 1)$$

- 8: **end for**

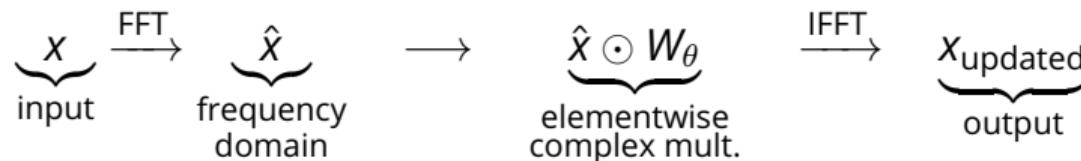
Idea: FNOs learn operators (mappings between function spaces) by representing integral kernels in the frequency domain.

Fourier Transform and its Inverse (1D Discrete Case):

$$\hat{x}_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}}, \quad x_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_k e^{2\pi i \frac{kn}{N}}.$$

- \hat{x}_k are the *Fourier coefficients* (in practice, often computed via FFT).

Fourier Layer in FNO:



- $\hat{x} \odot W_\theta$ involves complex-valued parameters W_θ .
- Often, only low-frequency modes are retained (spectral truncation).

- **Resolution Independence:** Once trained, FNOs can generalize to different input resolutions.
- **Fast Inference:** FFT-based operations reduce complexity compared to naive convolution.
- **Expressivity:** Captures global interactions via frequency domain (long-range dependencies).
- **Applicability:** Commonly used for PDE solution operators (e.g., Navier-Stokes, Darcy flow).

For small time steps, we have:

$$\hat{U}(t_j + \Delta t, t_j) \approx e^{-i\hat{H}(t_j + \Delta t/2)} \equiv e^{-i\hat{H}(t_{j+1/2})}$$

Approximate exponential as:

$$e^{-i\hat{H}\Delta t} \approx \frac{1 - i\hat{H}\Delta t/2}{1 + i\hat{H}\Delta t/2}$$

We get midpoint rule:

$$\left(1 + \frac{i}{2}\hat{H}(t_{j+1/2})\Delta t\right)\phi(t_{j+1}) = \left(1 - \frac{i}{2}\hat{H}(t_{j+1/2})\Delta t\right)\phi(t_j)$$

Predictor:

$$\left(1 + \frac{i}{2} \hat{H}(t_j) \Delta t\right) \varphi^{(1)}(t_{j+1}) = \left(1 - \frac{i}{2} \hat{H}(t_j) \Delta t\right) \varphi(t_j)$$

Corrector:

$$\hat{H}^{(n)}(t_{j+1/2}) = \frac{1}{2} \left[\hat{H}(t_j) + \hat{H}^{(n)}(t_{j+1}) \right]$$

$$\left(1 + \frac{i}{2} \hat{H}^{(n)}(t_{j+1/2}) \Delta t\right) \varphi^{(n+1)}(t_{j+1}) = \left(1 - \frac{i}{2} \hat{H}^{(n)}(t_{j+1/2}) \Delta t\right) \varphi(t_j)$$