Long-time accuracy of ensemble Kalman filters for chaotic and machine-learned dynamical systems

Statistical and Computational Challenges in Probabilistic Scientific Machine Learning (SciML)

IMSI

6/10/2025

Nathan Waniorek

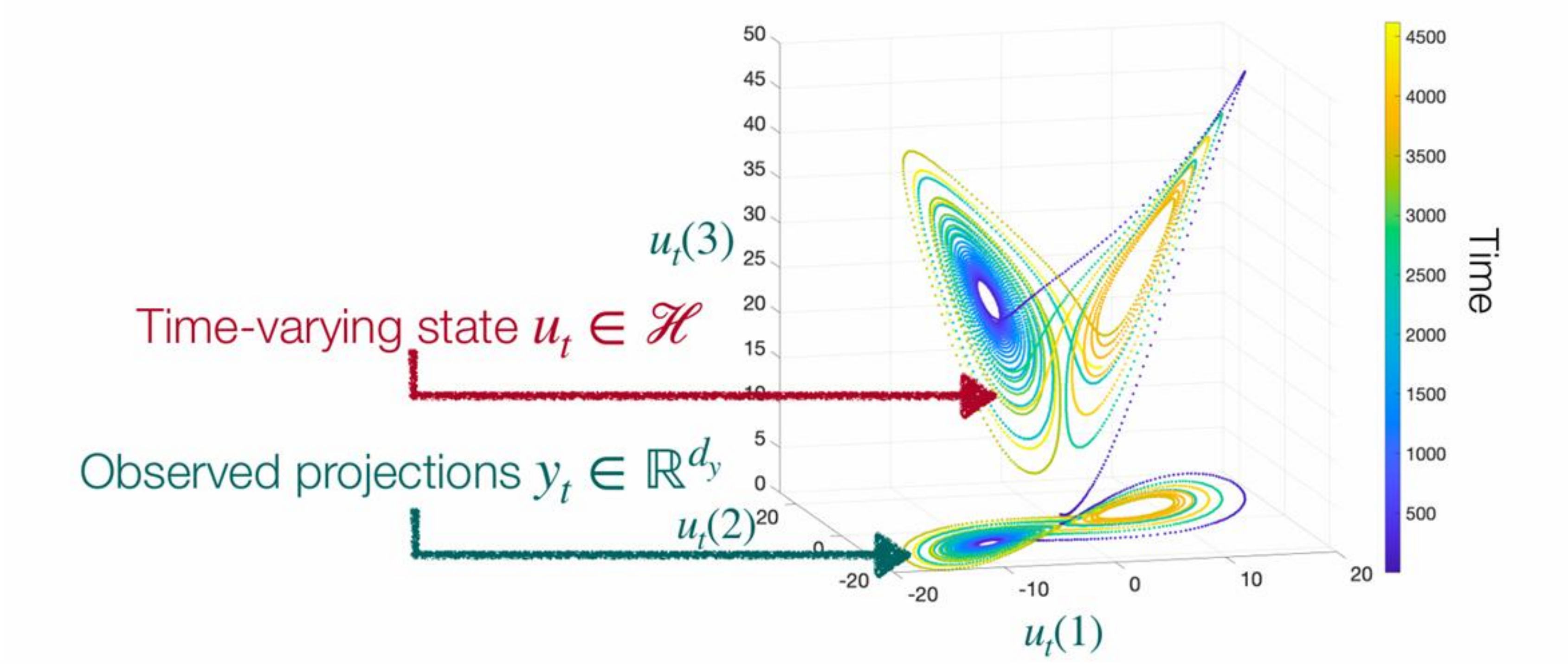
University of Chicago

Joint work with Daniel Sanz-Alonso

Unknown Initialization: $u_0 \in \mathcal{H}$

Dynamical System: $u_t = \Psi(u_{t-1})$

Observation: $y_t = Hu_t + \varepsilon\eta_t, \quad \eta_t \sim \mathcal{N}(0, R)$



Goal: find an estimate $\widehat{m}_t$ of u_t given $\{y_i\}_{i=1}^t$.

Challenges:

- High dimensionality.
- Partial observations.
- Nonlinear dynamics.
- Expensive or unknown dynamics.

The Ensemble Kalman Filter:

- Widely successful in practical applications (NWP).
- Justified in linear-Gaussian setting.
- Provably diverges in certain setting.

Link to arXiv preprint

1st Main Result: We provide conditions on the dynamics and observations that guarantee long-time accuracy of the EnKF with appropriate covariance inflation.

- First EnKF accuracy result for partially-observed nonlinear dynamics.
- Holds for finite ensemble size.
- Ensemble size independent of state dimension.
- Proof proceeds by showing accuracy of “mean-field” EnKF.
- Assumptions hold for Lorenz-63, Lorenz-96, and 2D Navier-Stokes with reasonable observation models.

2nd Main Result: We provide conditions on a surrogate model such that using it within the EnKF preserves long-time accuracy.

- Only requires **short-term** accuracy of the surrogate model.
- Agnostic to source of model error.
- Above desiderata also hold.

Link to arXiv preprint

Energetic Variational Neural Network Discretizations of Variational Models

Yiwei Wang (University of California, Riverside, email: yiweiw@ucr.edu)

Variational models are model specify by energy-dissipation law

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\boldsymbol{z}] = -\triangle(\boldsymbol{z}, \boldsymbol{z}_t)$$

Such a type of model plays an important role in modeling many problems in physics, material science, biology and machine learning

Examples of Variational model

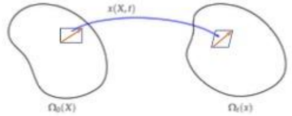
- L^2-gradient flow:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\varphi] = -\int \eta(\varphi)|\varphi_t|^2 \mathrm{d}x \;\Rightarrow\; \eta(\varphi)\varphi_t = -\frac{\delta \mathcal{F}}{\delta \varphi}$$

  Examples: Allen-Cahn; Landau-de Gennes; Chemical reactions.
- Generalized diffusion:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\rho] = -\int \eta(\rho)|\boldsymbol{u}|^2 \mathrm{d}x, \;\; \rho_t + \nabla\cdot(\rho\boldsymbol{u}) = 0 \;\Rightarrow\; \rho_t = \nabla\cdot\left(\frac{\rho^2}{\eta(\rho)}\nabla\frac{\delta\mathcal{F}}{\delta\rho}\right)$$

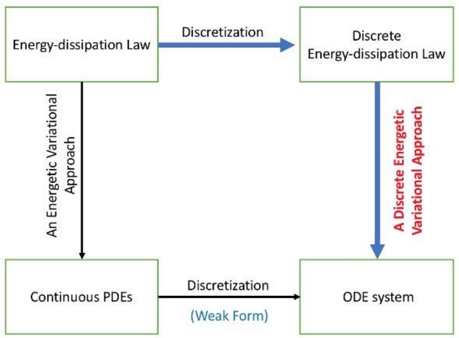
  Examples: Cahn-Hilliard; Fokker-Planck; Poisson-Nernst-Planck
  $\eta(\rho) = \rho$: Wasserstein gradient flow



- Flow map: $x(\mathbf{X}, t) : \Omega^0 \to \Omega^t$
- Velocity $\boldsymbol{u}(x,t) : \boldsymbol{u}(x(\mathbf{X},t),t) = x_t(\mathbf{X},t)$.
- Deformation tensor F : $\mathrm{F}(\mathbf{X},t) = \nabla_{\mathbf{X}} x(\mathbf{X},t)$

- ρ is determined by the flow map: $\rho(x(\mathbf{X},t),t) = \rho_0(\mathbf{X})/\det \mathrm{F}$
- The free energy can be viewed as a functional of the flow map.
- **Generalized diffusion as L^2-gradient flow of flow map**

Structure preserving discretization



- Continuous energy-dissipation law:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}[\varphi] = -\int \eta(\varphi)|\varphi_t|^2 \mathrm{d}\boldsymbol{x}$$

- Discrete energy-dissipation law:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{F}_h(\Xi) = -\Delta_h(\Xi(t), \Xi'(t)), \quad \Xi \in \mathbb{R}^K$$

- ODE system of Ξ_i (a gradient flow):

$$D_{ij}(\Xi)\Xi_j'(t) = -\frac{\delta \mathcal{F}_h}{\delta \Xi_i}, \; D_{ij} = \int \eta(\varphi_h)\frac{\partial \varphi_h}{\partial \Xi_i}\frac{\partial \varphi_h}{\partial \Xi_j}\mathrm{d}\boldsymbol{x}$$

- Variational temporal discretization:

$$\Xi^{n+1} = \underset{\Xi\in\mathcal{A}}{\arg\min}\, \frac{1}{2\tau}\mathrm{D}(\Xi^n)(\Xi - \Xi^n)\cdot(\Xi - \Xi^n) + \mathcal{F}_h(\Xi).$$

- Why neural network: neural network-based spatial discretization provides a mesh-free alternative to spectral methods, and can potentially handle high-dimensional problems.
- Difficulty: how to compute and store $\mathrm{D}(\Xi)$?

Discretization and variation are not commutable in general;
the variation-then-discretize approach may *destroy* the variational structure in the semi-discrete setting.

Neural Network Discretization of Minimizing Movement Scheme

- Construct a minimizing movement scheme based on continuous energy-dissipation law:

$$\varphi^{n+1} = \underset{\varphi \in \mathcal{S}}{\arg\min}\, J^n(\varphi), \quad J^n(\varphi) = \frac{1}{2\tau}\int \eta(\varphi^n)|\varphi - \varphi^n|^2 \mathrm{d}\boldsymbol{x} + \mathcal{F}[\varphi]\ .$$

- Finite dimensional approximation $\varphi_h(\boldsymbol{x};\Xi)$, $\quad \Xi \in \mathbb{R}^K$:

$$\Xi^{n+1} = \underset{\Xi \in \mathcal{S}_h}{\arg\min}\, J_h^n(\Xi), \quad J_h^n(\Xi) = \frac{1}{2\tau}\int \eta^n |\varphi_h(\boldsymbol{x};\Xi) - \varphi_h(\boldsymbol{x};\Xi^n)|^2 \mathrm{d}\boldsymbol{x} + \mathcal{F}_h[\Xi]\ .$$

- Compared with the spatial-discretize-first approach:

$$0 = \nabla_\Xi J_h^n(\Xi)|_{\Xi^{n+1}} = \frac{1}{\tau}\int \eta^n \underbrace{(\varphi_h(\boldsymbol{x};\Xi^{n+1}) - \varphi_h(\boldsymbol{x};\Xi^n))}_{\nabla_\Xi \varphi_h|_{\Xi^*}\cdot(\Xi^{n+1}-\Xi^n)} \nabla_\Xi \varphi_h|_{\Xi^{n+1}} \mathrm{d}\boldsymbol{x} + \nabla_\Xi \mathcal{F}_h|_{\Xi^{n+1}}$$

  Natural "semi-implicit" treatment of $D_{ij}(\Xi)$

- PDE-free discretization / No need to compute and store D(Ξ)
- We expect that if $\varphi_{NN}(x;\Theta^0)$ is a good approximation to $\varphi_0(x)$, then $\varphi_{NN}(x;\Theta^n)$ remains a good approximation to the solution $\varphi(x,t)$ at t^n.

Spatial and temporal discretization are commutable for the linear Galerkin approximation, but not commutable in general.

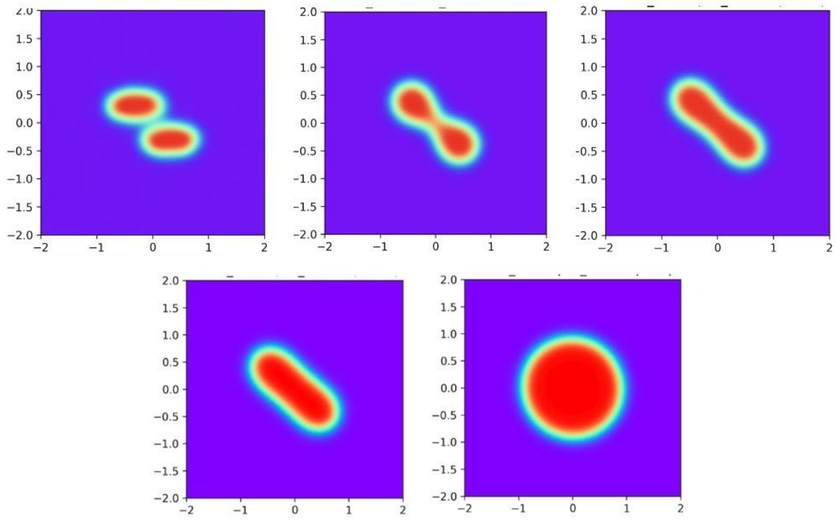
Phase field model for Willmore flow

- Energy-dissipation law:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int \frac{\epsilon}{2}\left(\Delta\varphi - \frac{1}{\epsilon^2}W'(\varphi)\right)^2 \mathrm{d}x = -\int_\Omega |\varphi_t|^2 \mathrm{d}x, \quad W(\varphi) = (1-\varphi^2)^2$$

- A challenging 4-th order equation

$$\partial_t \varphi = \Delta\mu - \frac{1}{\epsilon^2}W''(\varphi)\varphi, \quad \mu = \frac{1}{\epsilon^2}W'(\varphi) - \Delta\varphi$$



t = 0.001, 0.02, 0.05, 0.08, and 1 (long time stability)

Neural-network-based Lagrangian scheme for diffusions

- A generalized diffusion as an L^2-gradient flow of the flow map $x(\mathbf{X}, t)$:

$$\Phi^{n+1} = \underset{\Phi \in \mathrm{Diff}}{\arg\min} \frac{1}{2\tau} \int |\Phi(\mathbf{X}) - \Phi^n(\mathbf{X})|^2 \rho_0(\mathbf{X}) \mathrm{d}\,\mathbf{X} + \mathcal{F}\left[\Phi_{\#}\rho_0\right] \ ,$$

  - $\Phi_{\#}\rho_0(x) = \rho_0(\Phi^{-1}(x)) / \det F(\Phi^{-1}(x))$: push-forward of the density ρ_0 by Φ.
  - You might need a very large neural network to approximate Φ^{n+1} for large n.

- Given ρ^n, one can compute Ψ^{n+1} by solving the following optimization problem

$$\Psi^{n+1} = \underset{\Psi \in \mathrm{Diff}}{\arg\min} \frac{1}{2\tau} \int |\Psi(\boldsymbol{x}) - \boldsymbol{x}|^2 \rho^n(\boldsymbol{x}) \, \mathrm{d}\, \boldsymbol{x} + \mathcal{F}[\Psi_{\#}\rho^n] \ .$$

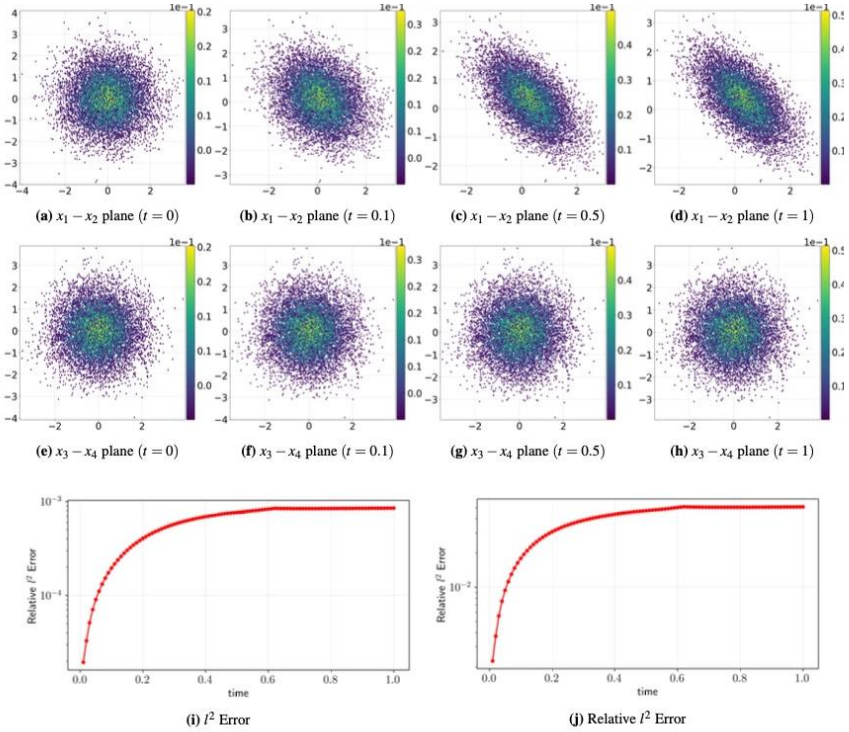
- Update ρ^{n+1} by $\rho^{n+1}(\boldsymbol{x}) = (\Psi_{\#}^{n+1}\rho^n)(\boldsymbol{x})$ / resample can be applied if needed
- Only a small size of neural network is needed to approximate Ψ^{n+1} at each time step when τ is small.
- Related to the JKO scheme for Wasserstein gradient flows:
  - JKO scheme for Wasserstein gradient flows (Eulerian approach):

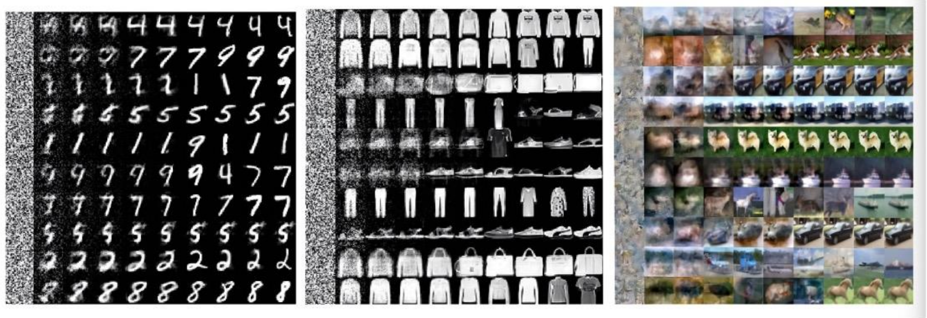
$$\rho^{n+1} = \underset{\rho \in \mathcal{P}_2(\Omega)}{\arg\min} \frac{1}{2\tau} W_2(\rho, \rho^n)^2 + \mathcal{F}[\rho] \ , \quad n = 0, 1, 2 \ldots \ ,$$

  - JKO scheme with the Benamou–Brenier formula in Lagrangian coordinate:

$$\boldsymbol{x}^*(\mathbf{X}, t) = \underset{\boldsymbol{x}(\mathbf{X},t)}{\arg\min} \frac{1}{2} \int_0^{\tau} \int_{\Omega} \rho_0(\mathbf{X}) |\boldsymbol{x}_t(\mathbf{X}, t)|^2 \, \mathrm{d}\, \mathbf{X} \, \mathrm{d}\, t + \mathcal{F}(\hat{\rho}(\boldsymbol{x}, \tau)),$$

  - An optimal condition: $\boldsymbol{x}_{tt}(\mathbf{X}, t) = 0$ on $(0, \tau)$, i.e., $\boldsymbol{x}(\mathbf{X}, t) = t(\Psi(\mathbf{X}) - X)/\tau + X$.





Hu, Z., Liu, C., Wang, Y., & Xu, Z. (2024). Energetic variational neural network discretizations of gradient flows. *SIAM Journal on Scientific Computing*, *46*(4), A2528-A2556.

Hardware Acceleration for HPS Methods in Two and Three Dimensions

Owen Melia

THE UNIVERSITY OF CHICAGO

Daniel Fortunato

FLATIRON INSTITUTE

Vasileios Charisopoulos

THE UNIVERSITY OF CHICAGO

Jeremy Hoskins

THE UNIVERSITY OF CHICAGO

Rebecca Willett

THE UNIVERSITY OF CHICAGO

What does it take to efficiently scale a direct PDE solver on a GPU?

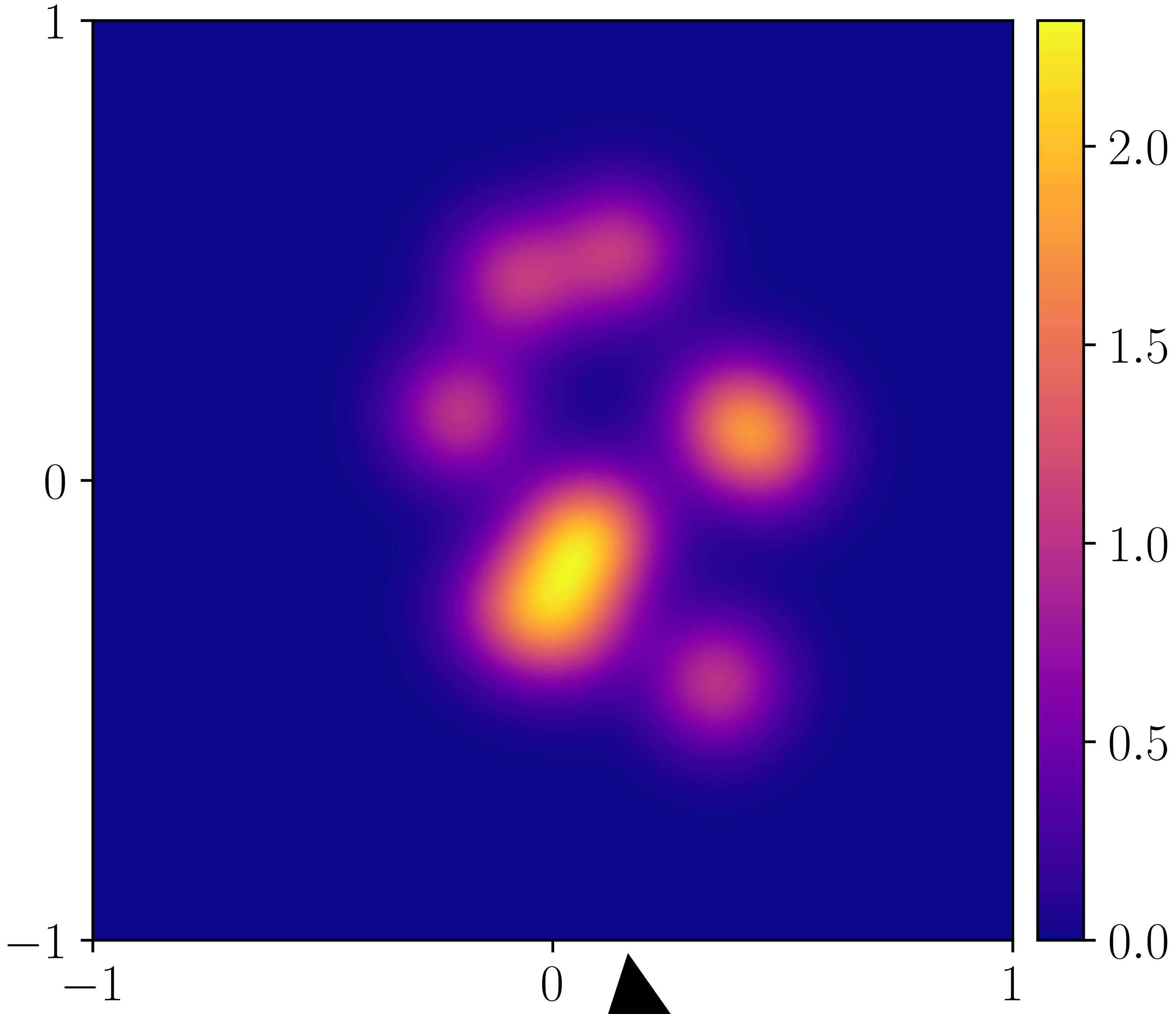
Flatten the computational graph as much as possible.

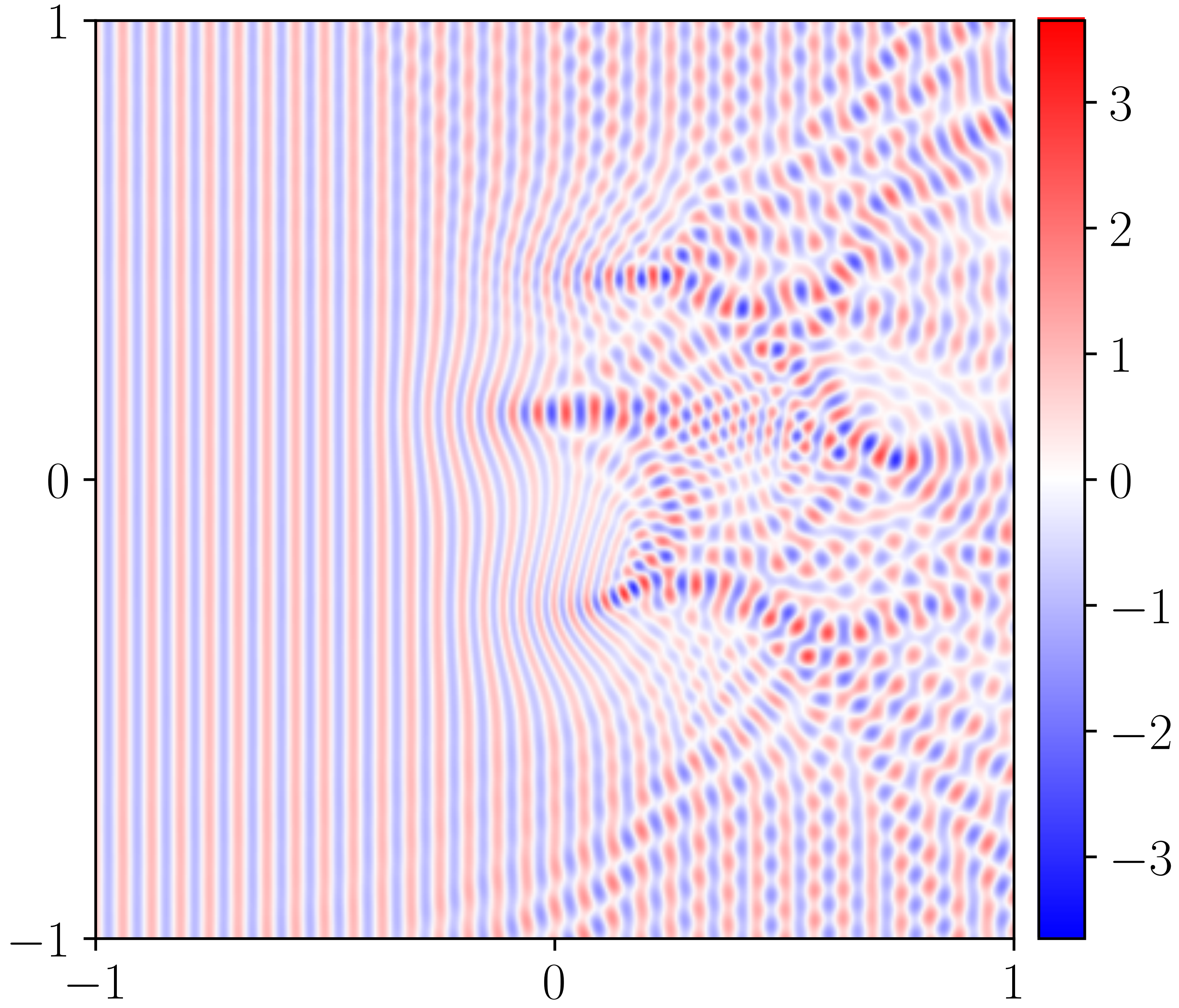
Modify algorithms to alleviate memory bottlenecks.

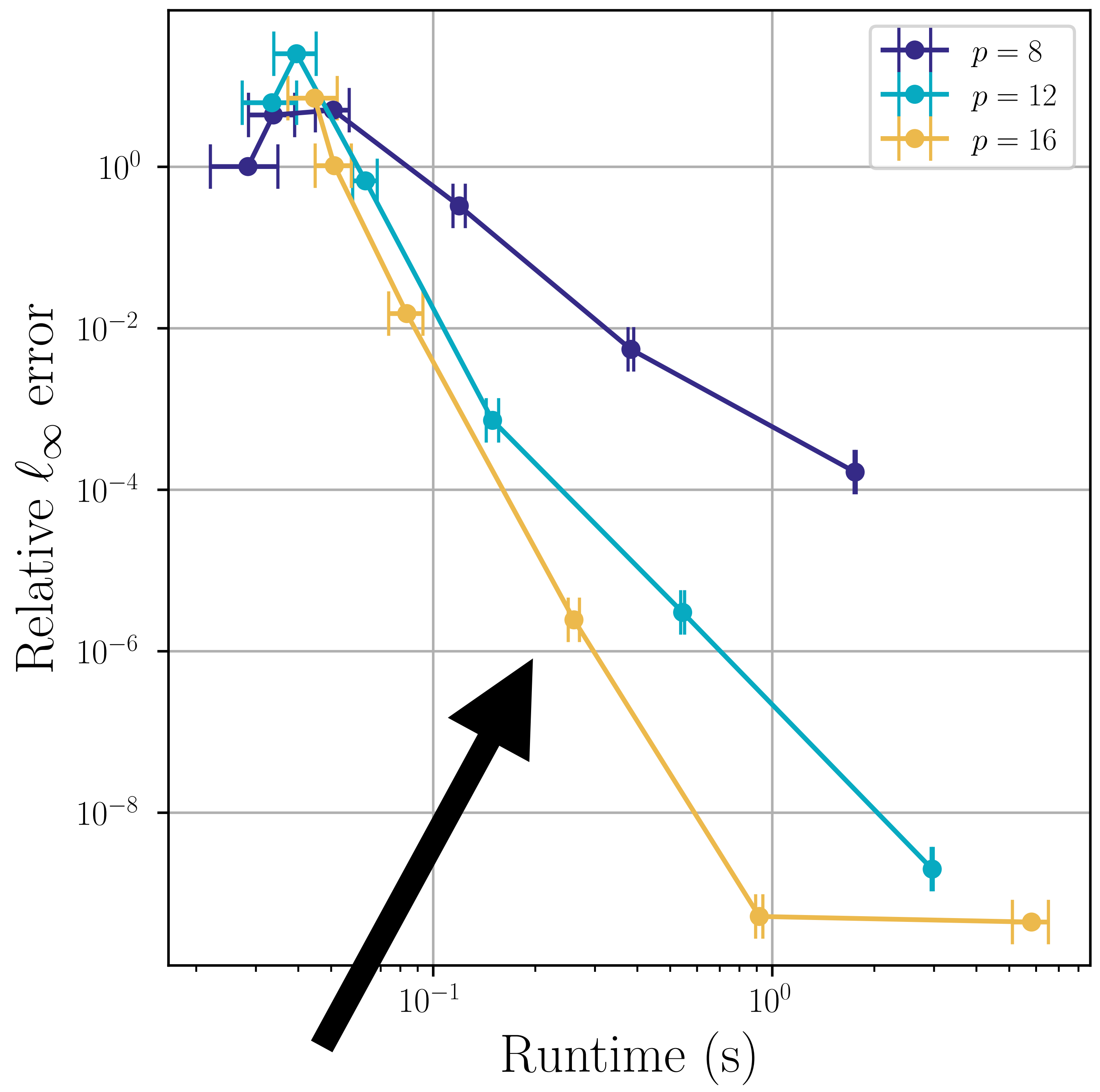
What does it take to efficiently scale a direct PDE solver on a GPU?

What can we do with a GPU-compatible implementation of a direct PDE solver?

Implement a very fast and accurate forward model for time-harmonic wave scattering problems.



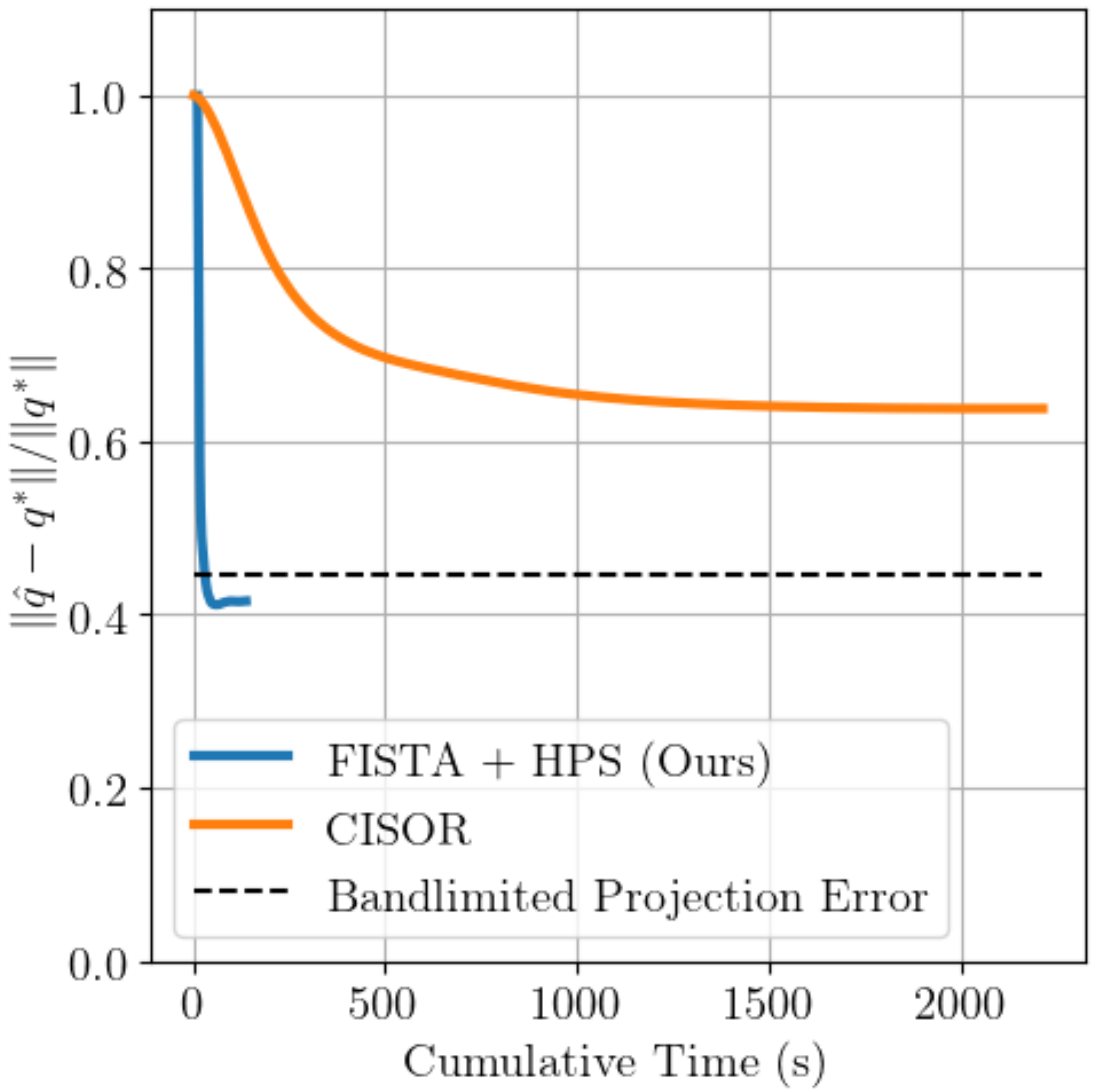


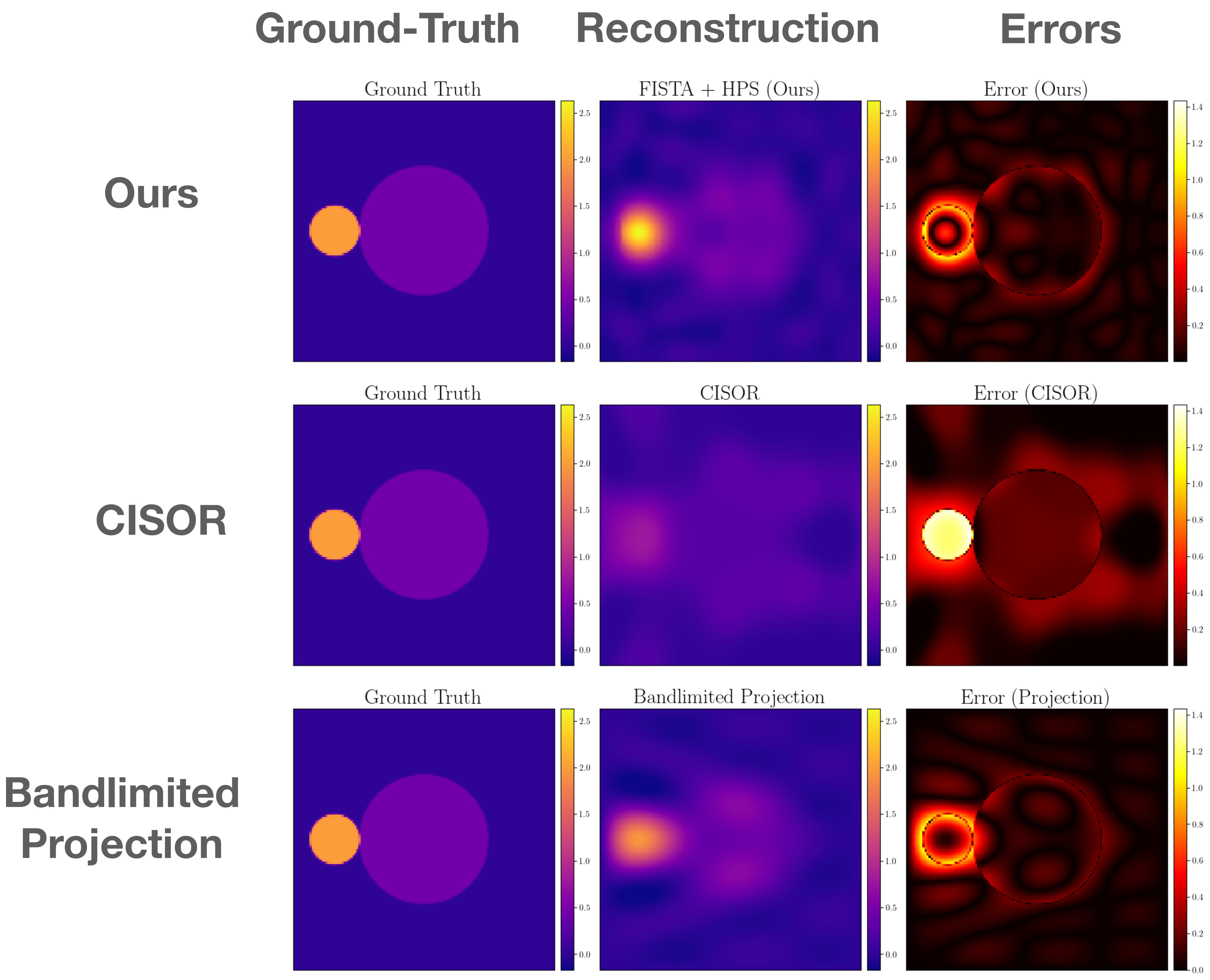


$$
\begin{cases}
\Delta u(x) + k^2(1+q(x))u(x) = -k^2 q(x) e^{ik\langle \hat{s}, x\rangle}, & x \in [-1,1]^2, \\
\sqrt{r}\left(\frac{\partial u}{\partial r} - iku\right) \to 0, & r = \|x\|_2 \to \infty.
\end{cases}
$$

5 digits of accuracy in 0.3 seconds!

Solve inverse scattering problems with experimental data





Solving the Inverse Scattering Problem: Leveraging Symmetries for Diffusion Models

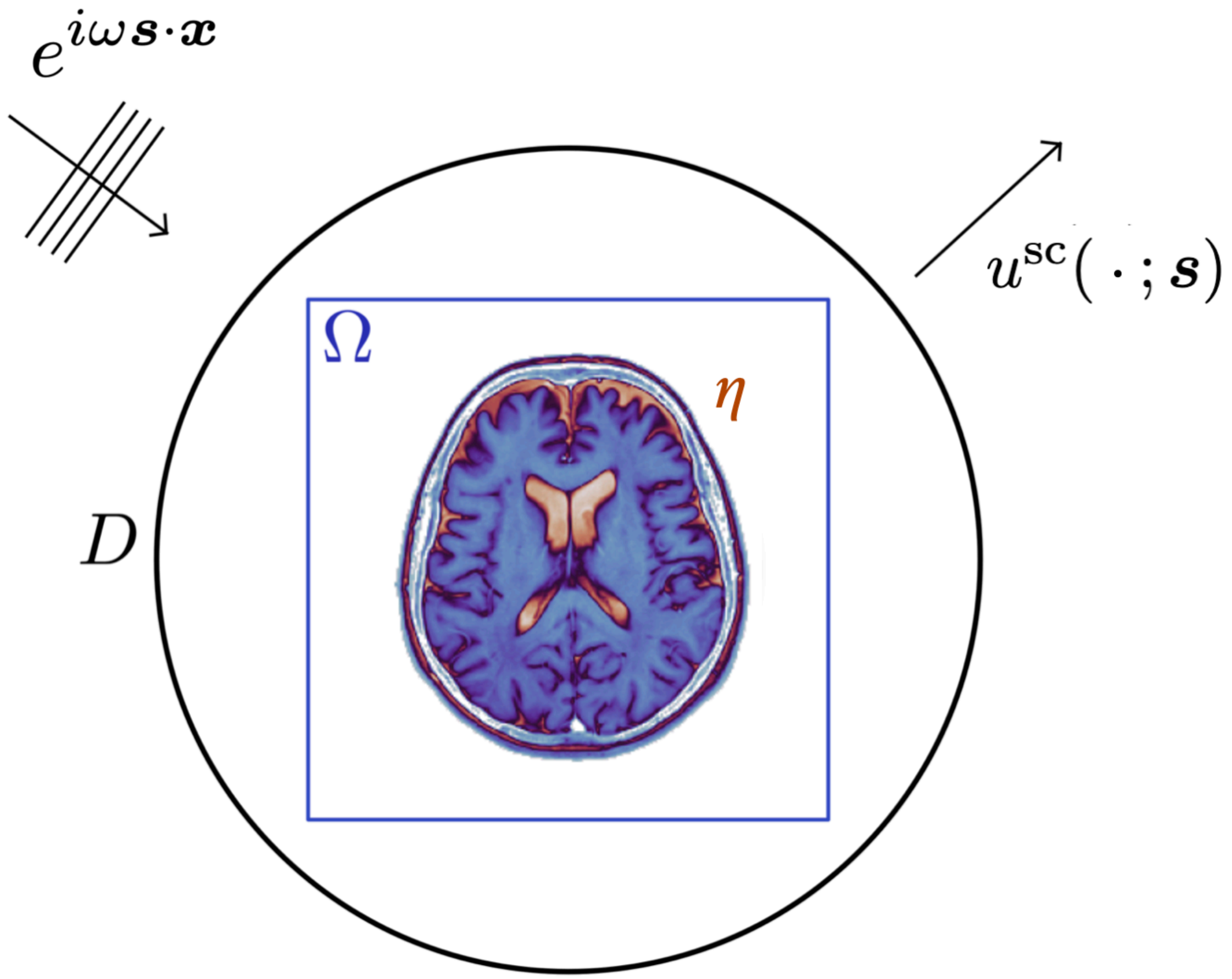
Helmholtz equation $\Delta u(\mathbf{x}) + \omega^2(1 + \eta(\mathbf{x}))u(\mathbf{x}) = 0$

Incident field : $u^{\mathrm{in}}(\boldsymbol{x}; \boldsymbol{s}) = e^{i\omega \boldsymbol{s}\cdot\boldsymbol{x}}$

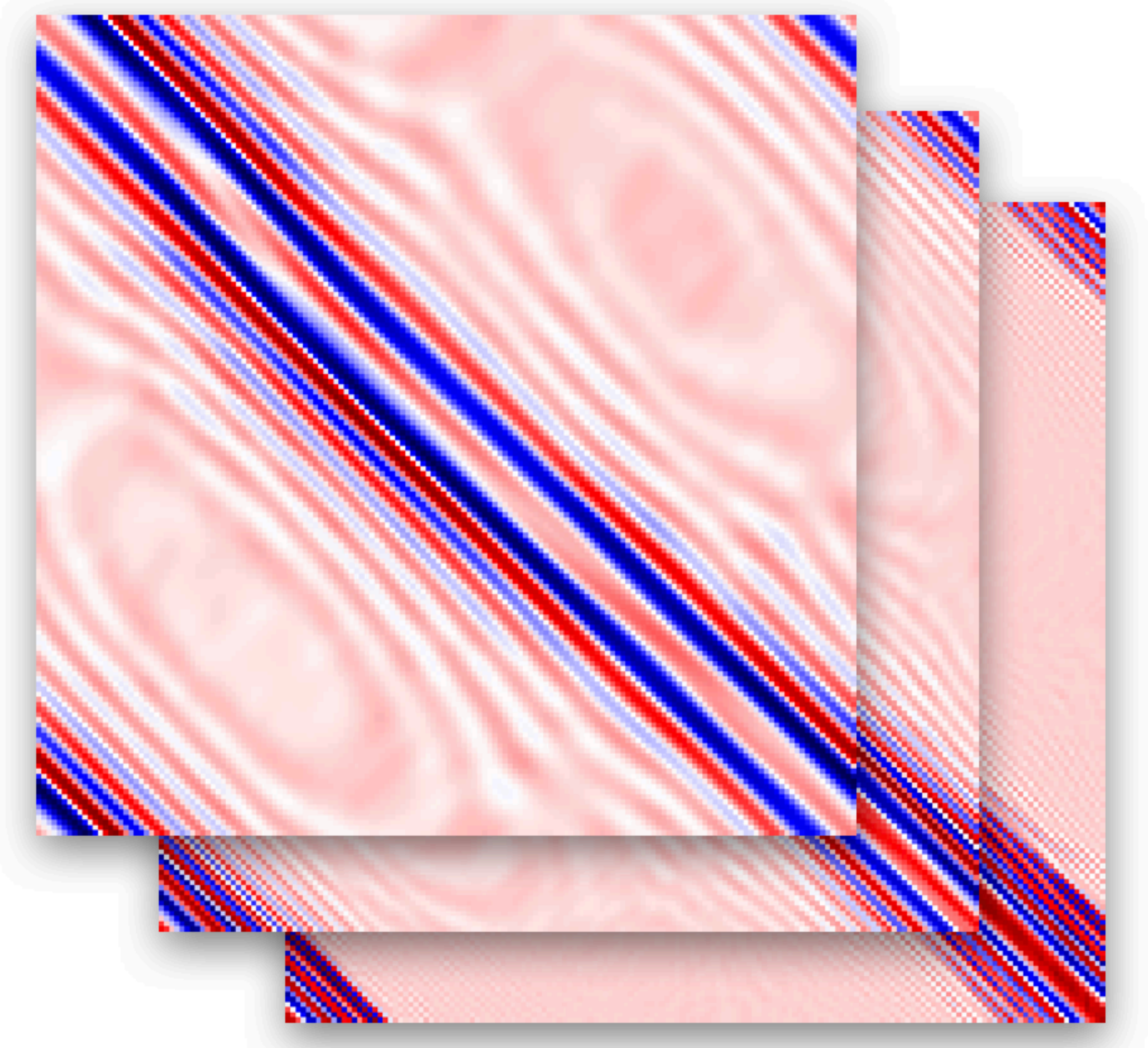
Scattered field : $u^{\mathrm{sc}}(\boldsymbol{x}, \boldsymbol{s}) = u(\boldsymbol{x}) - u^{\mathrm{in}}(\boldsymbol{x}, \boldsymbol{s})$

Far-field pattern: $\Lambda^{\omega}(\mathbf{s}, \mathbf{r}) = u^{\mathrm{sc}}(R\mathbf{r}; \mathbf{s})$

Forward map: $\mathcal{F}^{\omega}$ by $\mathcal{F}^{\omega}[\eta] = \Lambda^{\omega}$



$\Lambda^{\omega}(r, s) = u^{sc}(R\boldsymbol{r}; \boldsymbol{s})$



Inverse problem

$$\boldsymbol{\eta}^* = \mathcal{F}^{-1}(\Lambda^{\omega})$$

Bayesian framework

$$p(\boldsymbol{\eta}|\Lambda^{\omega}) \propto p(\boldsymbol{\eta})p(\Lambda^{\omega}|\boldsymbol{\eta})$$

$$\boldsymbol{\eta}^* = \arg\max_{\boldsymbol{\eta}} p(\boldsymbol{\eta}|\Lambda^{\omega})$$

Borong Zhang, Martin Guerra, Qin Li, and Leonardo Zepeda Núñez

Solving the Inverse Scattering Problem: Leveraging Symmetries for Diffusion Models

Filtered back-projection

$$\Lambda^\omega = \mathcal{F}^\omega[\eta] \approx F^\omega \eta.$$

$$= \int_{\mathbb{R}^2} e^{i\omega(\boldsymbol{s}-\boldsymbol{r})\cdot \boldsymbol{y}}\, \eta(\boldsymbol{y})\, d\boldsymbol{y}$$

$$\eta^* = \arg\min_{\eta} \|\Lambda^\omega - F^\omega \eta\|^2 + \epsilon\|\eta\|^2$$

$$= \underbrace{((F^\omega)^* F^\omega + \epsilon I)^{-1}}_{\text{Filtering}} \overbrace{\underbrace{(F^\omega)^* \Lambda^\omega}_{\text{Back-scattering}}}^{\text{Intermediate field } \alpha^\omega}$$

Filtering: Translational Equivariance

Back-scattering: Rotational Equivariance

Score-based diffusion models

$$d\boldsymbol{\eta}_t = f(t)\, \boldsymbol{\eta}_t\, dt + g(t)\, dW_t$$

$$d\boldsymbol{\eta}_t = \Big[f(t)\, \boldsymbol{\eta}_t - g^2(t) \underbrace{\boxed{\nabla_{\boldsymbol{\eta}} \log p_t(\boldsymbol{\eta}_t)}}_{\text{Score function}}\Big]\, dt + g(t)\, dW'_t$$

$$\boldsymbol{\eta}_0 \sim p_{\text{data}} \qquad \boldsymbol{\eta}_T \sim \mathcal{N}(\mathbf{0}, C^2 I)$$

$$d\boldsymbol{\eta}_t = \Big[f(t)\, \boldsymbol{\eta}_t - g^2(t) \underbrace{\boxed{\nabla_{\boldsymbol{\eta}} \log p_t(\boldsymbol{\eta}_t \mid \Lambda^\omega)}}_{\text{Conditional score}}\Big]\, dt + g(t)\, dW'_t.$$

$$\boldsymbol{\eta}^* \sim p_{\text{data}}(\boldsymbol{\eta} \mid \Lambda^\omega)$$

Borong Zhang, Martin Guerra, Qin Li, and Leonardo Zepeda Núñez

Back-Projection Diffusion

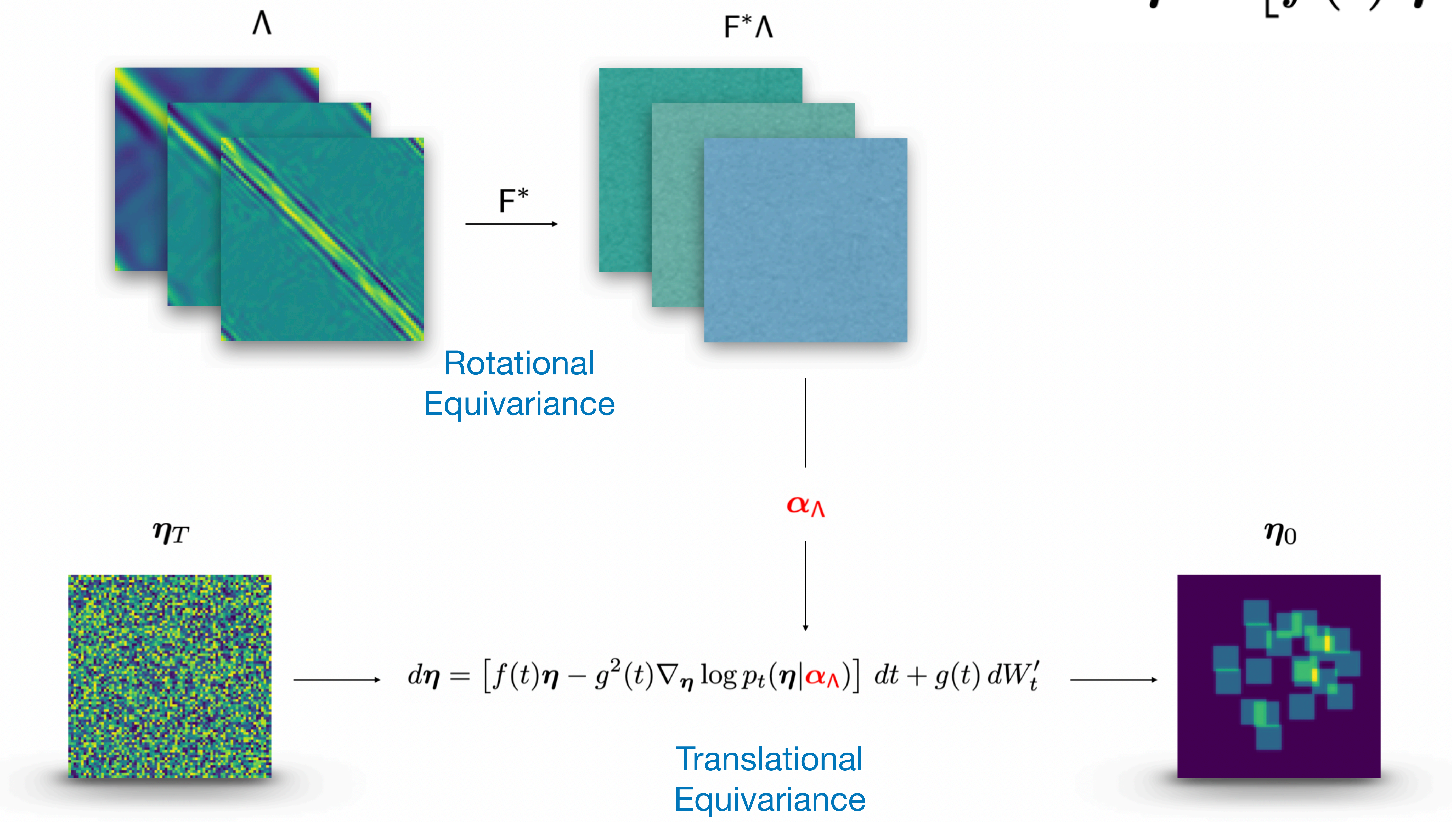
Stage 1: rotational equivariance

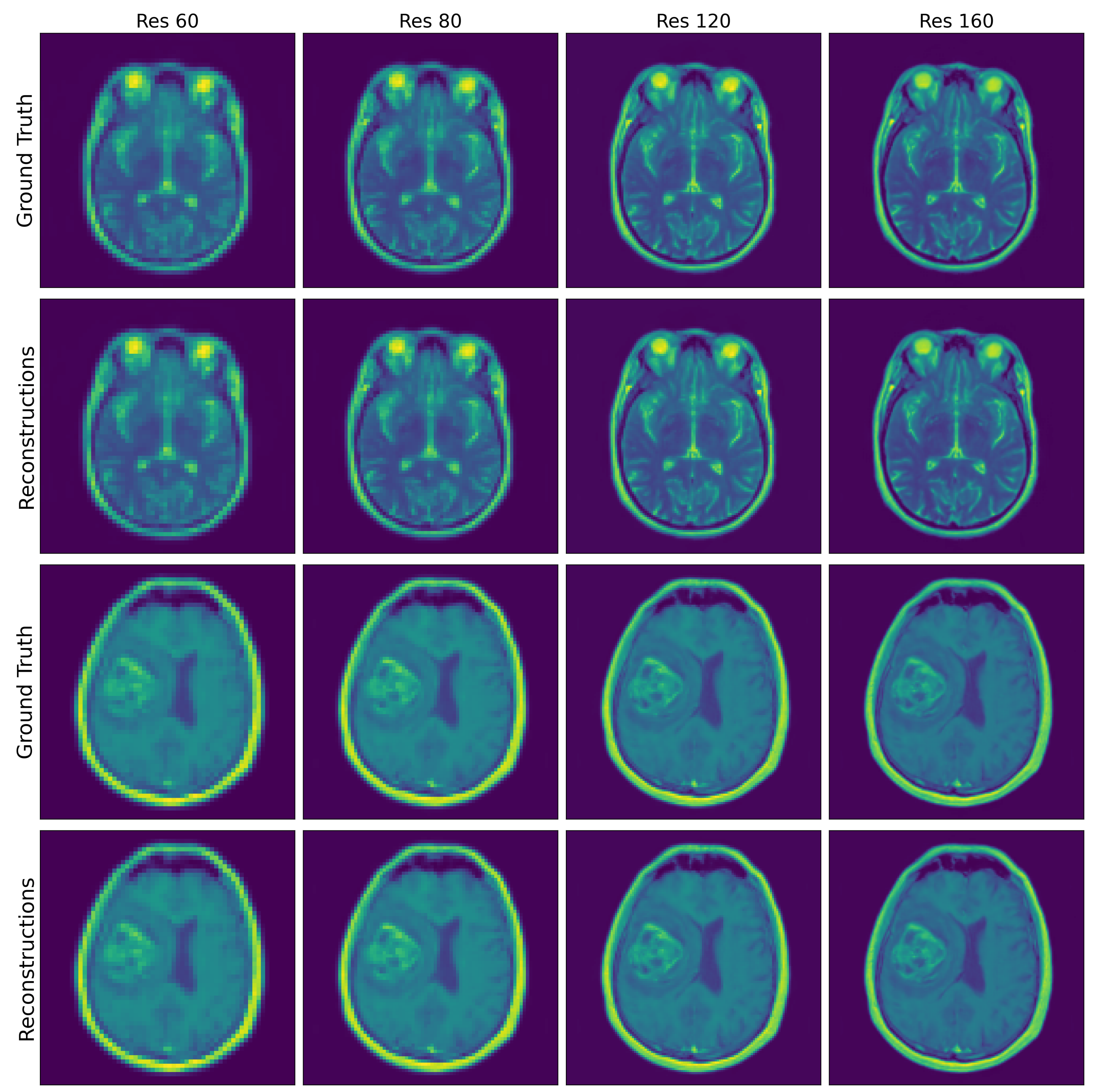
$$\boldsymbol{\alpha}_\Lambda(\boldsymbol{y}) = \boxed{F^*}\Lambda(\boldsymbol{y})$$

Stage 2: translational equivariance

$$\boldsymbol{\eta}^* \sim p_{\text{data}}(\boldsymbol{\eta}|\boldsymbol{\alpha}_\Lambda)$$

$$d\boldsymbol{\eta} = \left[f(t)\boldsymbol{\eta} - g^2(t)\boxed{\nabla_{\boldsymbol{\eta}} \log p_t(\boldsymbol{\eta}|\boldsymbol{\alpha}_\Lambda)}\right] dt + g(t)\, dW'_t$$





Borong Zhang, Martin Guerra, Qin Li, and Leonardo Zepeda Núñez

Beyond Closure Models: Learning Chaotic Systems via Physics-Informed Neural Operators

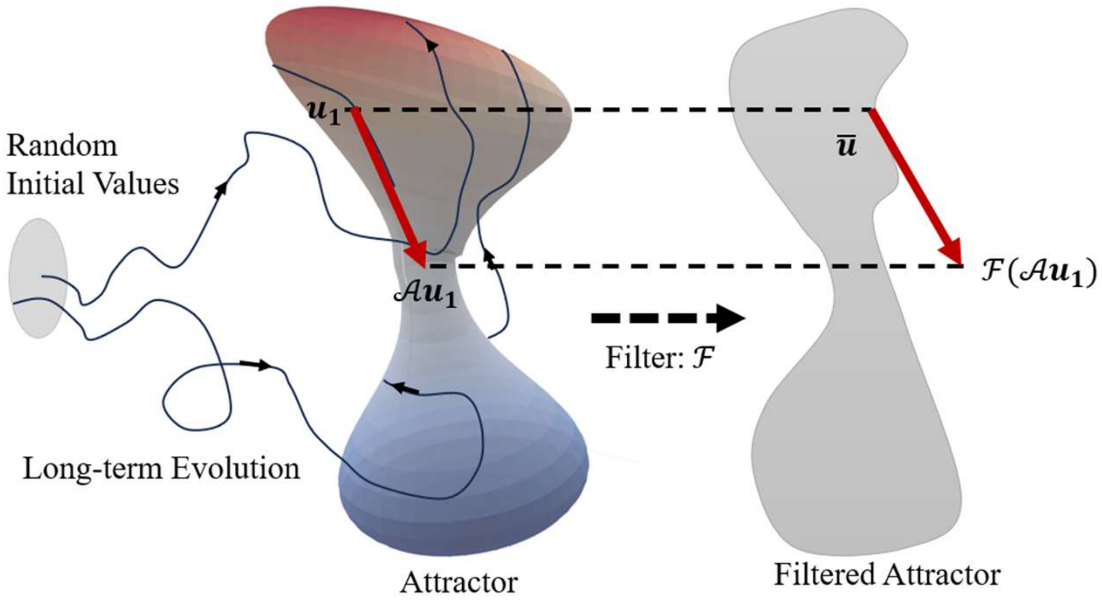
Chuwei Wang
Caltech

Joint work with Julius Berner, Zongyi Li, Di Zhou, Jiayun (Peter) Wang, Jane Bae, and Anima Anandkumar

2025.6.10

arxiv.org/abs/2408.05177

Caltech



(Chaotic) PDE Dynamics
$$\begin{cases} \partial_t u(x,t) = \mathcal{A}u(x,t) \\ u(x,0) = u_0(x),\ u_0 \in \mathcal{H} \end{cases}$$

Goal: long-term behavior / property of the attractor

Computational constraints: coarse-grid simulations

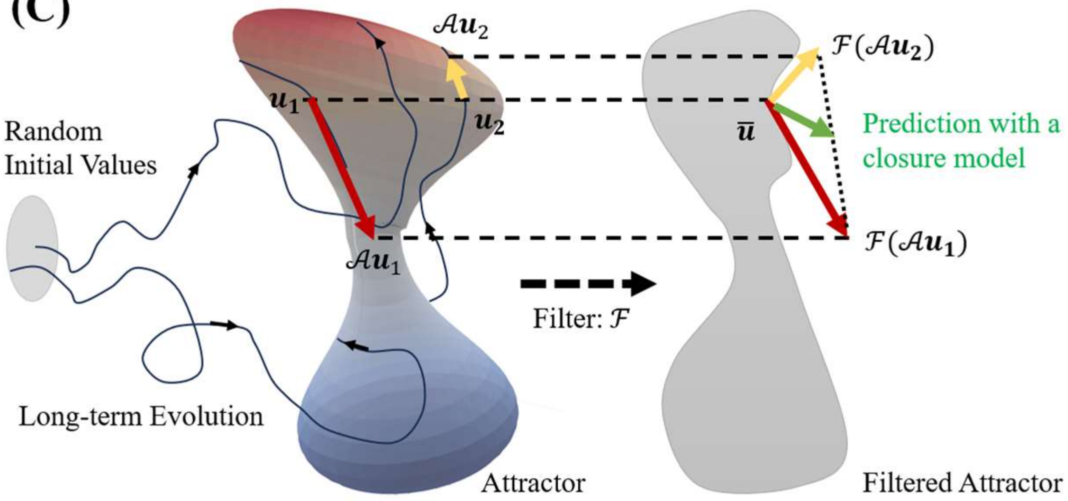
Coarse-graining (CG) : design a dynamics in the filtered space (i.e. coarse-grid system).

Closure Modeling:

$$\begin{cases} \partial_t v(x,t) = \mathcal{A}v(x,t) + clos(v;\theta),\ x \in D' \\ v(x,0) = \overline{u}_0(x),\ \overline{u}_0 \in \mathcal{F}(\mathcal{H}), \end{cases}$$

Fundamental Limitation of (data-driven) Closure Modeling

(C)



$$\begin{cases} \partial_t v(x,t) = \mathcal{A}v(x,t) + clos(v;\theta), \ x \in D' \\ v(x,0) = \overline{u}_0(x), \ \overline{u}_0 \in \mathcal{F}(\mathcal{H}), \end{cases}$$

The target mapping is not well-defined (a multi-map).

Model learns to predict the mean, not necessarily make sense.

[Theorem-1]

[Single-State & History-aware Closures]

Approximation error has a large lower bound regardless of model capacity!

[Stochastic Closures]

Cannot derive $\mathcal{F}_{\#}\mu^*$ (filtered invariant measure) when there is non-zero randomness in the dynamics.

Learning-based Closure Models

- Supervised Learning (Single-State Model)

$$J_{ap}(\theta;\mathfrak{D}) = \frac{1}{|\mathfrak{D}|}\sum_{i\in\mathfrak{D}} \|clos(\overline{u}_i;\theta) - (\mathcal{F}\mathcal{A} - \mathcal{A}\mathcal{F})u_i\|^2$$

[Advanced Variants]

- [Posterior Training Loss]
- [History-aware Models] Model's input:

$$\{\overline{u}(x_i, t-s)\}_{x_i\in D',\ 0<s\leq t_0}$$

- [Stochastic Closure Models]

Optimal Closure Model (Improving the results in Langford et al)

$$clos^*(v) = \mathbb{E}_{u\sim\mu^*}[\mathcal{F}\mathcal{A}u | \mathcal{F}u = v] - \mathcal{A}v, \quad v \in \mathcal{F}(\mathcal{H})$$

Technical Tool: functional Liouville flow- check the evolution of measures (of functions $u(x,t)$).

[Theorem-2]

- Suppose the NN closure model ansatz is expressive enough.
- N snapshots (datapoints) sampled from μ^* , fully-resolved simulation.
- $clos^*_\theta$: the closure model after training (minimizer of training loss).

$$\|clos_{\theta^*} - clos^*\| = \Omega(N^{-\frac{1}{d_0+2}})$$

- d_0: intrinsic dimension of the attractor. d_0 scales with Reynolds number.

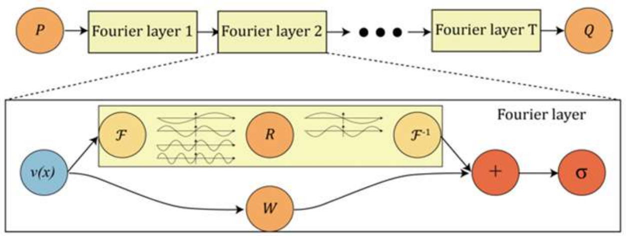
We need nonlinear interaction between different scales (e.g. with Neural Operators)

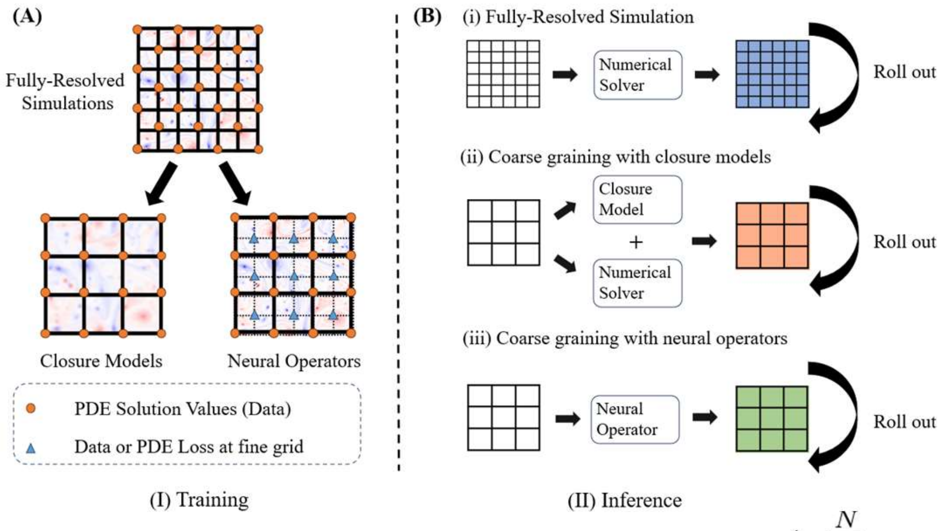
[Contradiction] If hundreds of thousands of fully-resolved data are available, there is no need to train a closure model! Directly estimate the long-term statistics with $O(N^{-\frac{1}{2}})$ error!

Key Takeaway

> ➢We need **nonlinear** interaction between information from different scales (i.e. resolved part in coarse-grid system and unresolved parts)!

➢**Previous Ansatz:** $\mathcal{A} + clos(\cdot, \boldsymbol{\theta})$ **New Ansatz:** $\tilde{A}_\theta$





Key Property of FNO:

- ➢ Naturally support input of different resolution.
- ➢ Ensure consistency among different resolutions.
- ➢ Model weights save the interaction of large- and small-scale information after trained with fine-grid input functions.

Theorem 3.1. *For any $h > 0$, denote $\hat{\mu}_{h,\theta} := \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \delta_{\mathcal{G}_\theta^n v_0(x)}$, any $v_0(x)$ with $x \in D'$. For any $\epsilon > 0$, there exists $\delta > 0$ s.t. as long as $\|(\mathcal{G}_\theta u)(\cdot, h) - S(h)u\|_{\mathcal{H}} < \delta, \forall u \in \mathcal{H}$, we have $\mathcal{W}_{\mathcal{H}}(\hat{\mu}_{h,\theta}, \mathcal{F}_{\#}\mu^*) < \epsilon$, where $\mathcal{W}_{\mathcal{H}}$ is a generalization of Wasserstein distance in function space.*

The newest version: arxiv.org/abs/2408.05177

Graph Neural Networks and Non-commuting Operators

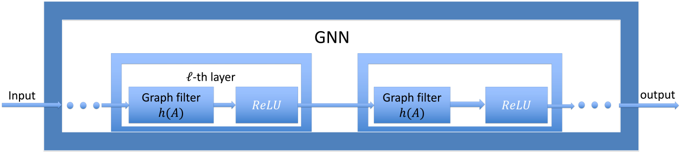
Kaiying O'Hare

Johns Hopkins University

SciML Lightning Talk
June 10, 2025

GNNs and GtNNs

- Graph Neural Networks (GNNs):



- Graph filter $h(A)$:
  linear layer; polynomial of the adjacent matrix A of a graph

$$h(A) = c_0 I + c_1 A + c_2 A^2 + c_3 A^3$$

- Goal of Graph-tuple Neural Networks (GtNNs):
  Extend to several similarity relations A_j on the same vertex set!
- Graph filter for GtNNs: non-commutative polynomial

$$h(A_1, A_2) = c_0 I + c_1 A_1 + c_2 A_2 + c_3 A_1^2 + c_4 A_1 A_2 + c_5 A_2 A_1 + c_6 A_2^2$$

Main Result: Stability and Transferability

- Graph Tuples: $\vec{A} = (A_1, A_2)$ and $\vec{B} = (B_1, B_2)$
- Stability:
  if $\vec{A} \approx \vec{B}$, the outputs of the GtNNs are similar

- Sequence of Graph Tuples $\vec{A}^{(n)}$ with different size n, i.e., A_1 and A_2 have n vertex
- Transferability:
  if $\vec{A}^{(n)} \to \vec{W}$ as $n \to \infty$, the outputs of the GtNNs converge
- $\Rightarrow$ if $\vec{A}^{(n)} \approx \vec{B}^{(m)}$, the outputs are similar via interpolation and sampling

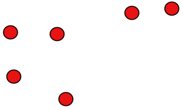
Reference

Mauricio Velasco, Kaiying O'Hare, Bernardo Rychtenberg, and Soledad Villar. Graph neural networks and non-commuting operators. Advances in neural information processing systems, 37:95662–95691, 2024.

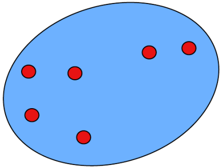
Learning Where to Learn:
Training Distribution Selection for Provable OOD Performance

Nicolas Guerra (Cornell), Nicholas H. Nelsen (MIT), Yunan Yang (Cornell)

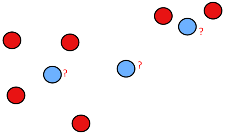
Problem: Models trained on a single distribution perform poorly on unseen test domains.
Goal: Find a training distribution that minimizes average test error over a family of distributions.



Family of Test Distributions.



Ideal: A broad training distribution (blue) covering all test scenarios.



In practice: constrained.

Formulation:

$$\inf_{\nu\in\mathscr{P}_2(\mathcal{U})}\left\{\mathbb{E}_{\nu'\sim\mathbb{Q}}\mathbb{E}_{u\sim\nu'}\left\|\widehat{\mathcal{G}}^{(\nu)}(u)-\mathcal{G}^\star(u)\right\|^2 \;\middle|\; \widehat{\mathcal{G}}^{(\nu)}\in\arg\min_{\mathcal{G}\in\mathcal{H}}\mathbb{E}_{u\sim\nu}\left\|\mathcal{G}^\star(u)-\mathcal{G}(u)\right\|^2_{\mathcal{Y}}\right\}$$

Training distribution ν; Test distributions $\nu'\sim\mathbb{Q}$; Model $\widehat{\mathcal{G}}^{(\nu)}$ trained on ν; True operator $\mathcal{G}^\star:\mathcal{U}\to\mathcal{Y}$

Theoretical Foundation and Algorithmic Approaches

Problem Recap: We aim to choose a training distribution ν that minimizes either the OOD error or a generalization upper bound.

$$\inf_{\nu \in \mathscr{P}_2(\mathcal{U})} \left\{ \mathbb{E}_{\nu' \sim \mathbb{Q}} \mathbb{E}_{u \sim \nu'} \left\| \widehat{\mathcal{G}}^{(\nu)}(u) - \mathcal{G}^\star(u) \right\|^2 \;\middle|\; \widehat{\mathcal{G}}^{(\nu)} \in \arg\min_{\mathcal{G} \in \mathcal{H}} \mathbb{E}_{u \sim \nu} \left\| \mathcal{G}^\star(u) - \mathcal{G}(u) \right\|_{\mathcal{Y}}^2 \right\}$$

1. Bilevel Optimization

Idea: Directly minimize empirical OOD loss over test distributions $\{\nu_i'\}$:

$$\theta^{(k+1)} = \theta^{(k)} - t_k \nabla \mathsf{J}(\theta^{(k)})$$

Pro: Optimizes the true OOD loss.
Con: Requires access to test distributions ν_i', Requires computing gradient $\nabla \mathsf{J}$.

2. Alternating Minimization Algorithm (AMA)

OOD Upper Bound: OOD error $\leq$ ID training error $+$ distribution mismatch
Idea: Iteratively minimize upper bound:

$$\nu_\theta^{(0)} \xrightarrow{\text{train}} \widehat{\mathcal{G}}^{(0)} \xrightarrow{\text{optimize}} \nu_\theta^{(1)} \xrightarrow{\text{train}} \cdots$$

Pro: Avoids test distribution samples; leverages structure of meta test distribution $\mathbb{Q}$.
Con: Minimizes a surrogate, not the true OOD loss.

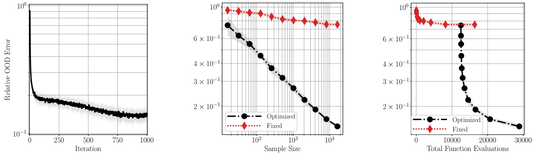
Key Insight: Both approaches optimize the training distribution ν to improve generalization.

Examples

Bilevel Example: Function Approximation

Goal: Optimize C in $x \sim \mathcal{N}(m_0, C) \in \mathscr{P}_2(\mathbb{R}^d)$, with fixed m_0 to learn the function:

$$g(x) = 10\sin(\pi x_1 x_2) + 20(x_3 - 1/2)^2 + 10x_4 + 5x_5$$

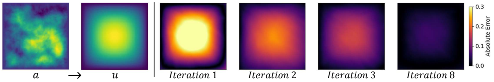


(Left) Error over 1000 iterations. (Center) Error as sample size grows. (Right) Error as number of function evaluations grows.

AMA Example: Darcy Flow

Goal: Optimize the mean function m in $a \sim \mathrm{GP}(m, C)$, where C is fixed, to learn the map $\mathcal{G} : a \mapsto u$.

$$\begin{cases} \nabla \cdot (a \nabla u) = 1 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases}$$



Visual comparison of solution errors over AMA iterations for Darcy Flow.

What metric to optimize for suppressing instability in a Vlasov-Poisson system?

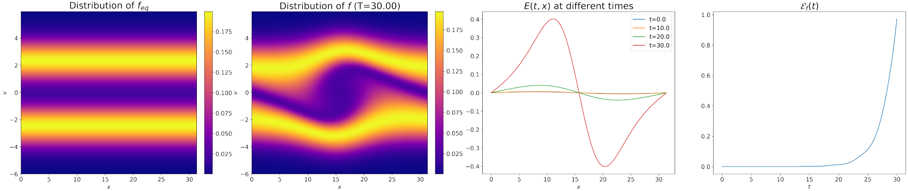
Martin Guerra, Qin Li, Yukun Yue and Leonardo Zepeda-Núñez

Statistical and Computational Challenges in Probabilistic SciML
10 June 2025

UNIVERSITY OF WISCONSIN–MADISON

Motivation:

- Fusion energy promises a limitless, clean, and safe power source for the future.
- Achieving it requires understanding and stabilizing high-temperature plasma to prevent turbulence.



PDE-constrained Optimization:

$$\begin{cases} \partial_t f + v\partial_x f - (E_f + H)\cdot \partial_v f = 0\,, \\ E_f = \partial_x V_f\,, \\ \partial_{xx} V_f = 1 - \rho_f = 1 - \int f \, \mathrm{d}v\,. \end{cases} \quad (1)$$

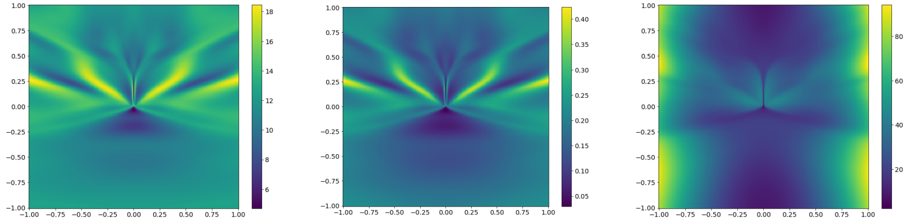
$$\min_H \quad \mathcal{J}(f[H]) \qquad \text{s.t.} \quad (1) \quad (2)$$

Given a desired equilibrium f_{eq},

Which objective $\mathcal{J}$ would be the best to optimize (2)?

- L^2: $\frac{1}{2}\|f[H]-f_{\mathrm{eq}}\|_{L^2(x,v)}$
- KL: $\mathrm{KL}(f[H]||f_{\mathrm{eq}})$
- $\mathcal{E}_f$: $\int_0^T \int_0^{L_x} [E_{f[H]}(t,x)]^2 \,\mathrm{d}x\,\mathrm{d}t\,.$

$$H(x) = \sum_{k=1}^{N} a_k \cos\left(\frac{2\pi k x}{L_x}\right) + b_k \sin\left(\frac{2\pi k x}{L_x}\right).$$



Can you tell which landscape corresponds to which objective?

A score-based particle method for homogeneous Landau equation

Yan Huang (joint with Li Wang)

School of Mathematics, University of Minnesota

IMSI, Chicago
June 10, 2025

Score-based Particle Method

The Landau equation models the density of charged particles undergoing the Coulomb force in plasmas:

$$\partial_t f = \nabla_{\boldsymbol{v}} \cdot \int_{\mathbb{R}^d} A(\boldsymbol{v}-\boldsymbol{v}_*) \left(f(\boldsymbol{v}_*) \nabla_{\boldsymbol{v}} f(\boldsymbol{v}) - f(\boldsymbol{v}) \nabla_{\boldsymbol{v}_*} f(\boldsymbol{v}_*) \right) \mathrm{d}\boldsymbol{v}_* \,,$$

with the collision kernel $A(\boldsymbol{z}) = C_\gamma |\boldsymbol{z}|^{\gamma+2} \left(I_d - \frac{\boldsymbol{z} \otimes \boldsymbol{z}}{|\boldsymbol{z}|^2} \right)$.

- A "Log" form of continuity equation:

$$\partial_t f + \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{U}[f] f) = 0 \,,$$
$$\boldsymbol{U}[f] = - \int_{\mathbb{R}^d} A(\boldsymbol{v} - \boldsymbol{v}_*) (\underbrace{\nabla_{\boldsymbol{v}} \log f(\boldsymbol{v})}_{\text{score}} - \nabla_{\boldsymbol{v}_*} \log f(\boldsymbol{v}_*)) f_* \mathrm{d}\boldsymbol{v}_* \,.$$

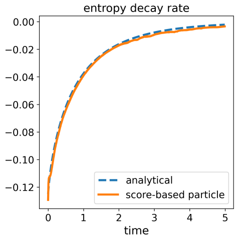
- Learn score via the score-matching loss:

$$\boldsymbol{s}_\theta^n(\boldsymbol{v}) \in \arg\min_\theta \frac{1}{N} \sum_{i=1}^N |\boldsymbol{s}_\theta(\boldsymbol{v}_i^n)|^2 + 2\nabla \cdot \boldsymbol{s}_\theta(\boldsymbol{v}_i^n)$$

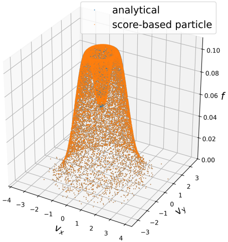
- Update particles: $\boldsymbol{v}_i^{n+1} = \boldsymbol{v}_i^n - \Delta t \frac{1}{N} \sum_{j=1}^N A(\boldsymbol{v}_i^n - \boldsymbol{v}_j^n) [\boldsymbol{s}_\theta^n(\boldsymbol{v}_i^n) - \boldsymbol{s}_\theta^n(\boldsymbol{v}_j^n)]$.
- Update density (no kernel density estimation):

$$l_i^{n+1} = -\Delta t \frac{1}{N} \sum_{j=1}^N \nabla_{\boldsymbol{v}_i} \cdot \{ A(\boldsymbol{v}_i^n - \boldsymbol{v}_j^n) [\boldsymbol{s}_\theta(\boldsymbol{v}_i^n) - \boldsymbol{s}_\theta(\boldsymbol{v}_j^n)] \} \,, \; f^{n+1}(\boldsymbol{v}_i^{n+1}) = f^n(\boldsymbol{v}_i^n) / \exp(l_i^{n+1}) \,.$$

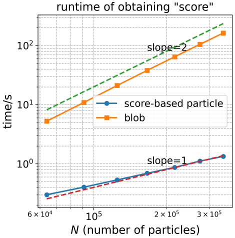
Numerical Experiments



(a) Time evolution of the entropy decay rate.



(b) Density visualization at particle locations.

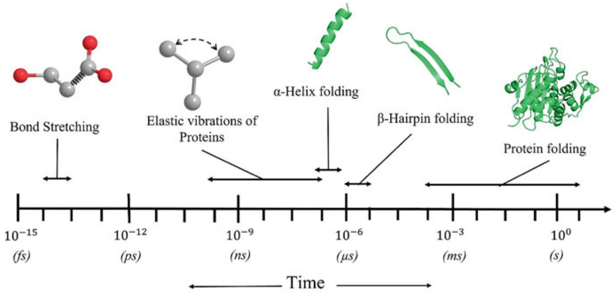


(c) Computational time of obtaining "score" on GPU

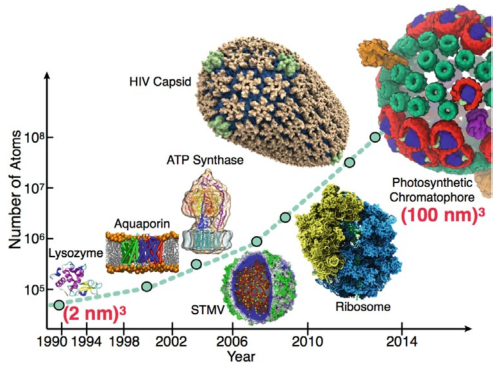
Reference: Y. Huang and L. Wang, A score-based particle method for homogeneous Landau equation, Journal of Computational Physics, (2025), p. 114053.

Adaptive compression for sampling rare events in high dimensions

Time scale problem in molecular simulations

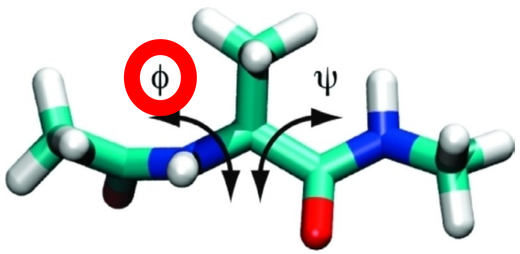


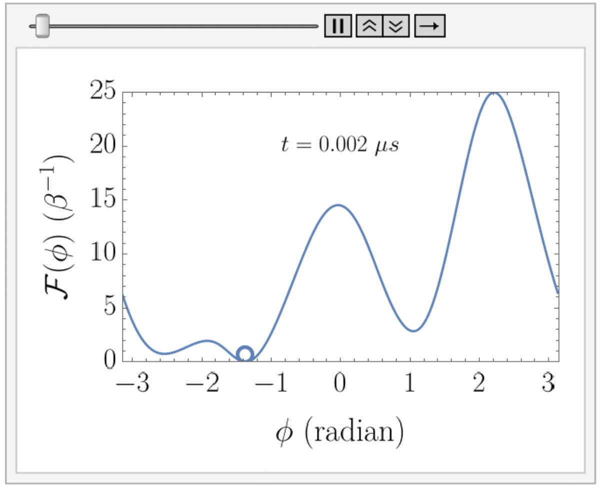
E. Wilson et al., 2021 *Structure and Function of Membrane Proteins* vol. 2302



V. Junghare et al., 2023 *Markov State Models of Molecular Simulations to Study Protein Folding and Dynamics*

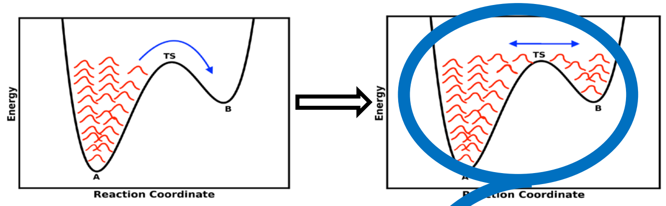
Alanine dipeptide: 2 relevant collective variables (ϕ, ψ Ramachandran angles)



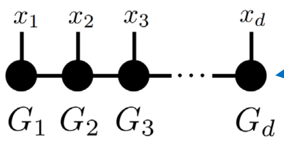


>200 million time steps!

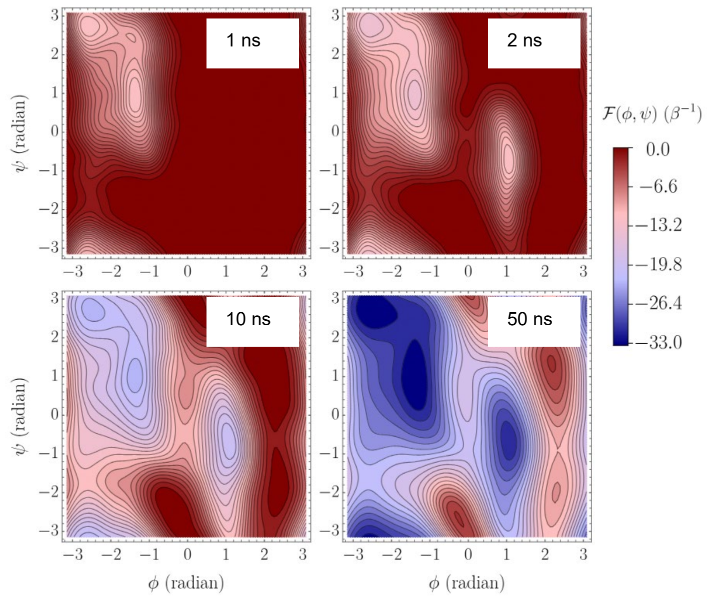
TT-Metadynamics



A. Laio and M. Parrinello 2002 *J. Phys. Chem. B* **99** 12562—12566

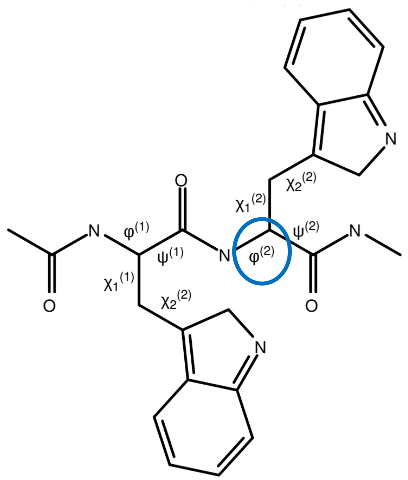


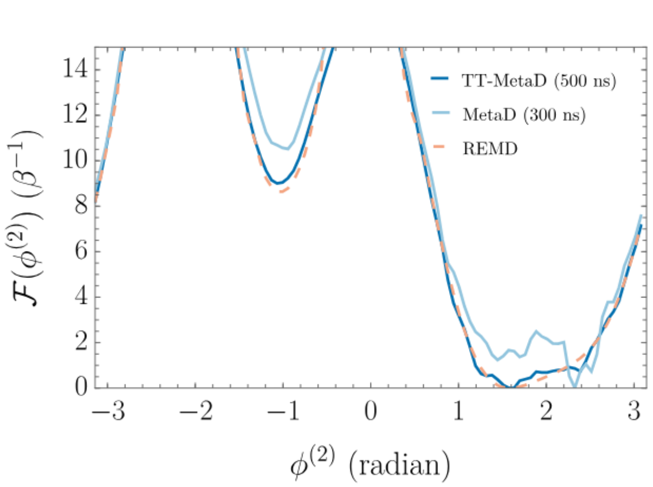
Periodically fit bias potential to a low-rank tensor train (using TT sketch algorithm)

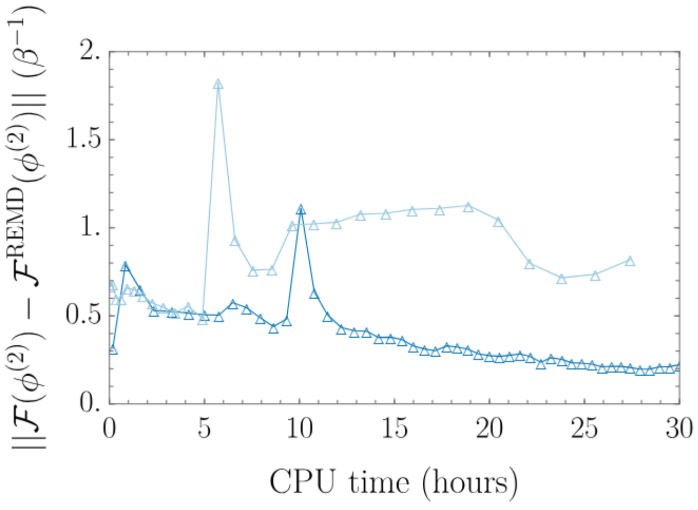


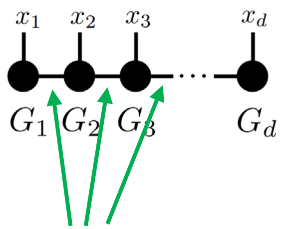
Nils Strand—University of Chicago—Lightning Talk—June 10, 2025
Joint work with Siyao Yang, Yuehaw Khoo, and Aaron Dinner

Why tensor compression matters: ditryptophan (8D) as a test case

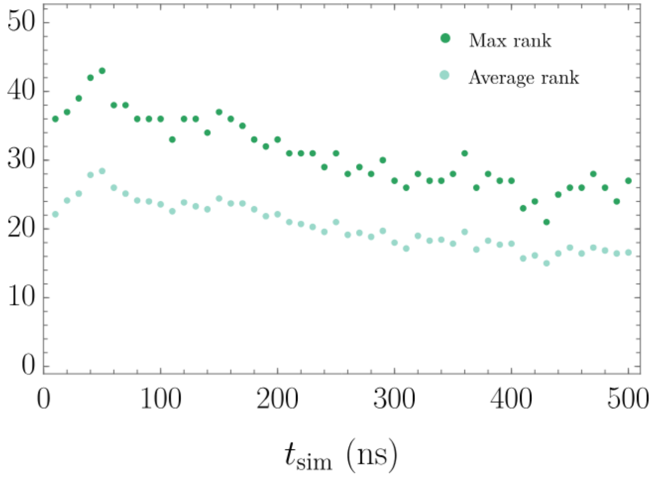








Rank = number of possible states joining dimensions (= correlations)

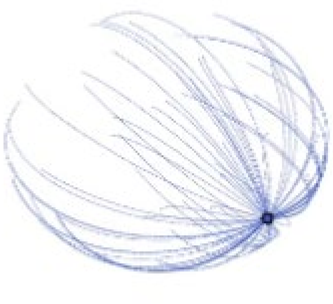


Takeaways:

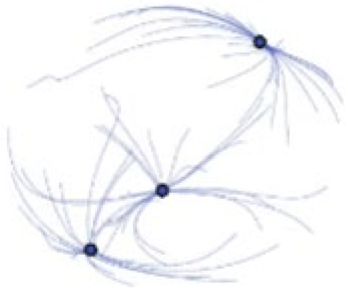
- Adaptive bias compression
- Linear scaling in dimension
- No grid needed
- Improved stability/smoothness over time

Nils Strand—University of Chicago—Lightning Talk—June 10, 2025
Joint work with Siyao Yang, Yuehaw Khoo, and Aaron Dinner

Quantitative Clustering in Mean-Field Transformer Models



Shi Chen
Department of Mathematics
Massachusetts Institute of Technology

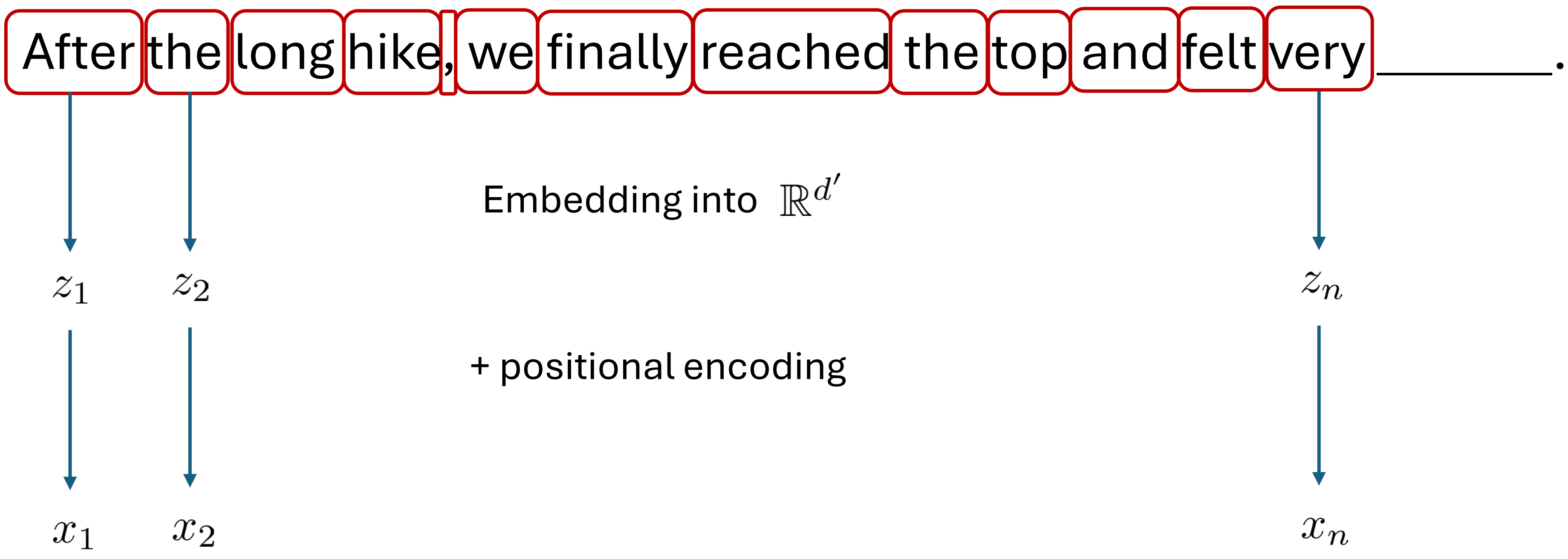


Joint work with Zhengjiang Lin, Yury Polyanskiy and Philippe Rigollet

SciML Workshop, IMSI

June 10, 2025

Massachusetts Institute of Technology



	Prob.
"tired":	0.22
"happy":	0.20
"relieved":	0.18
"exhausted":	0.17
"accomplished":	0.13
"satisfied":	0.10

$\mathbb{R}^d \ni x_1, \dots, x_n$ **: tokens** $\iff$ $\frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$

- **Transformers are trained to predict next token:**

Input = emprical measure on tokens ← **Empirical distribution of tokens in prompt**

Output = probability measure on tokens ← **Likelihood of token being next**

Transformer = Flow map

- **Interacting particle system** $\mu(t) = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i(t)}$ **Continuity equation**

$$\dot{x}_i(t) = \mathcal{X}_t[\mu(t)](x_i(t)) \qquad i = 1, \ldots, n \qquad \Longleftrightarrow \qquad \partial_t \mu(t) + \nabla_x \cdot (\mu(t)\mathcal{X}_t[\mu(t)]) = 0$$

- **Self-attention:** $\mathcal{X}_t[\mu(t)](x) = \mathbb{P}_x V_t \frac{\int y e^{\langle Q_t x, K_t y\rangle} \mu(t)(\mathrm{d}y)}{\int e^{\langle Q_t x, K_t y\rangle} \mu(t)(\mathrm{d}y)}$

Projection onto the tangent space of unit sphere (= LayerNorm)

- $Q_t = K_t = \beta I_d$

$$\dot{x}(t) = \nabla \delta \mathcal{F}[\mu(t)](x(t)) \qquad \mathcal{F}[\mu] = \iint e^{\beta\langle x, y\rangle} \mu(\mathrm{d}x)\mu(\mathrm{d}y)$$

Riemannian gradient over sphere

Fréchet derivative

- (Reverse) **Wasserstein gradient flow** $\Longrightarrow$ Maximizer of energy: $\mu = \delta_x$ **Clustering**

Theorem (Łojaciewicz inequality on a small cap, C.-Lin-Polyanskiy-Rigollet)

If $\mathrm{supp}(\mu) \subset S_+(\sqrt{\beta})$, then there exists $x \in \mathbb{S}^{d-1}$, such that

$$\mathcal{F}[\delta_x] - \mathcal{F}[\mu] \leq \int \|\mathcal{X}[\mu](y)\|_2^2 \, \mu(\mathrm{d}y)$$

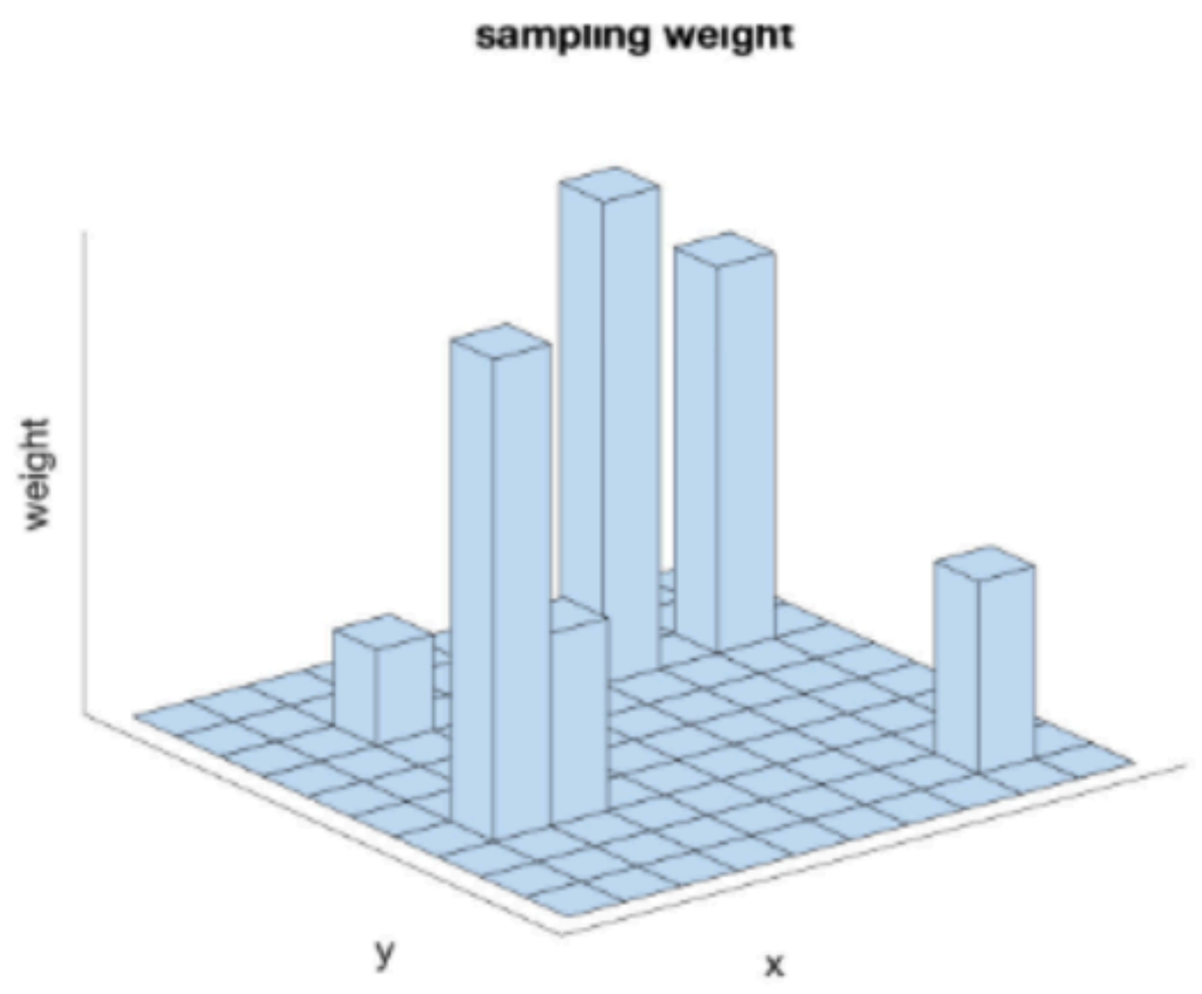
Cap size = $\sqrt{\beta}$

Continuous Nonlinear Adaptive Experimental Design

Ruhui Jin (presenter), Qin Li, Stephen Mussmann, and Stephen Wright

Motivation?

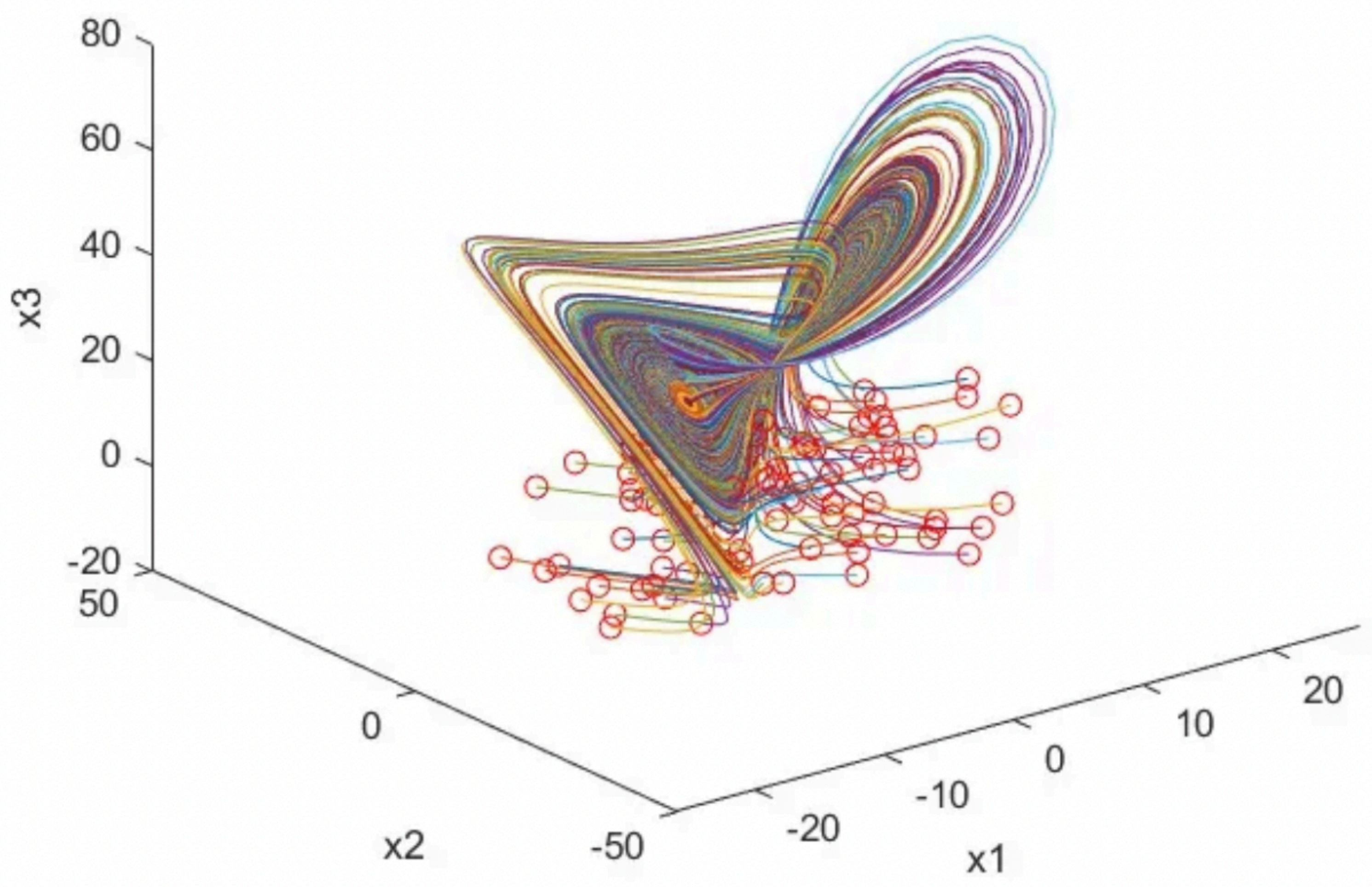
Experimental design identifies the most valuable measurements and provide guidance prior to experiment procedure.



What has been done?

Traditional design approaches assign probability weights to a finite set of design options.

In practice, experimental configurations, such as when to take snapshots, where to place sensors, should be **continuously-indexed** across the domain.



What's novel in this work?

We aim at

- Optimal design on **continuous probability space**.
- Measurements are from **nonlinear** models.

$$\overset{\text{pursued design measure}}{\rho^*} := \arg\min_{\rho\in\mathbf{Pr}_2} \overset{\text{design criterion}}{F}\left[\rho;\ \overset{\text{inference solution}}{\sigma^*[\rho]}\right]$$

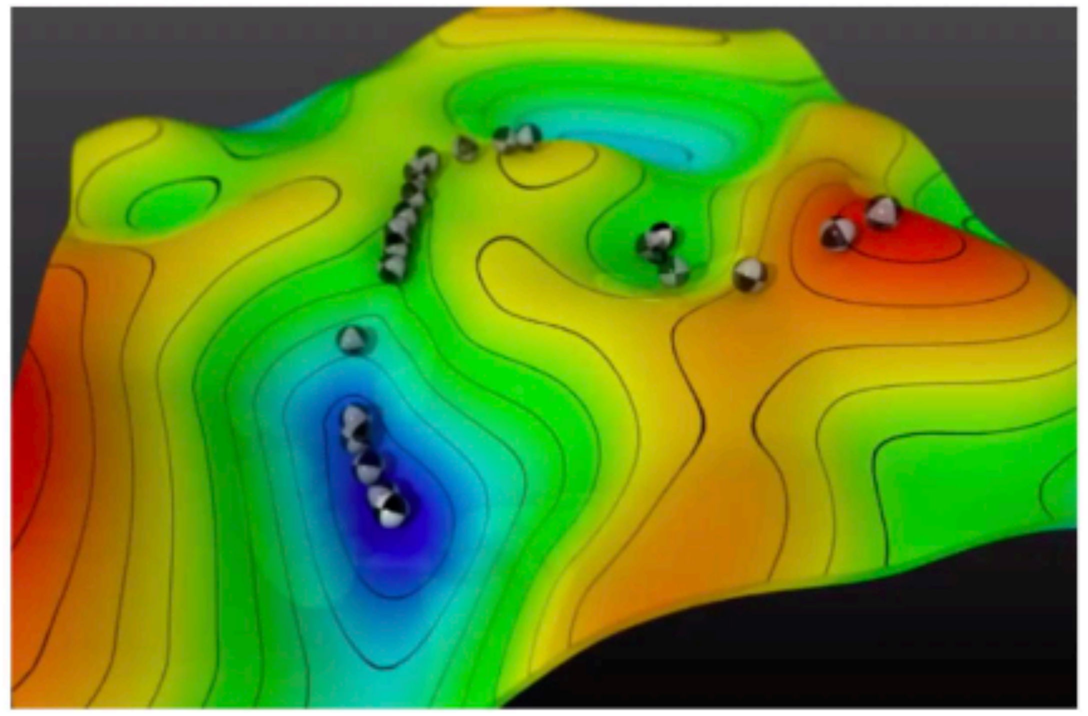
Optimization thru the lens of dynamics,

1. Outer-loop ρ

Gradient flow $\Rightarrow \partial_t\rho$

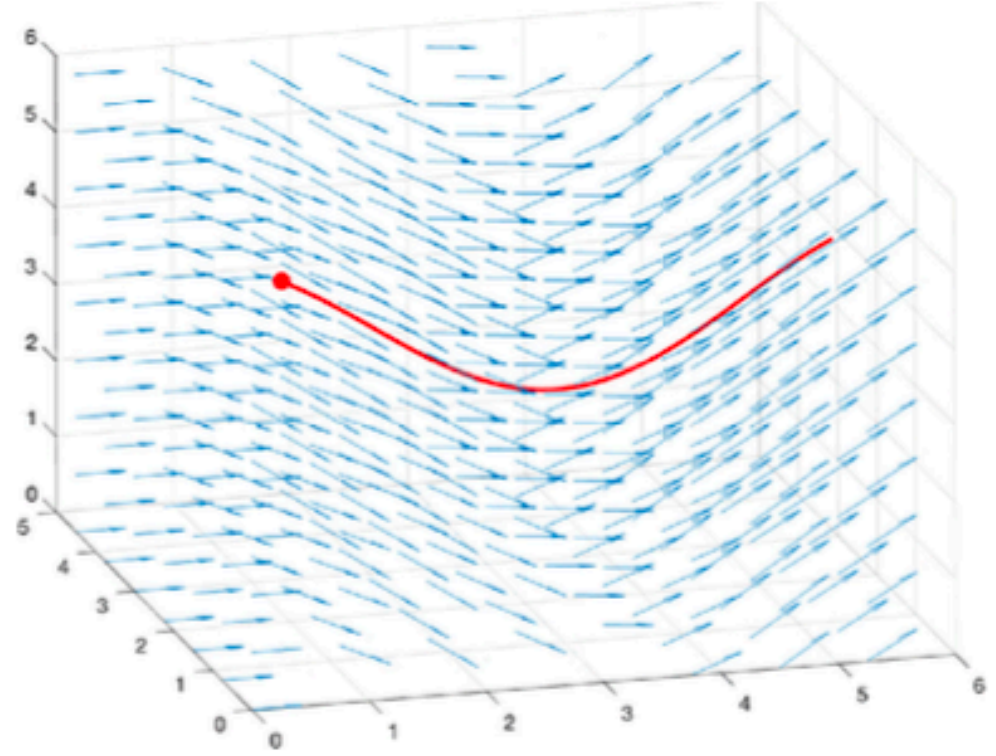
2. Inner-loop $\sigma^*[\rho]$

Dynamics translation $\dot{\sigma}^*[\rho] \Leftarrow \partial_t\rho$



(a) Dynamics of ρ

$\Rightarrow$

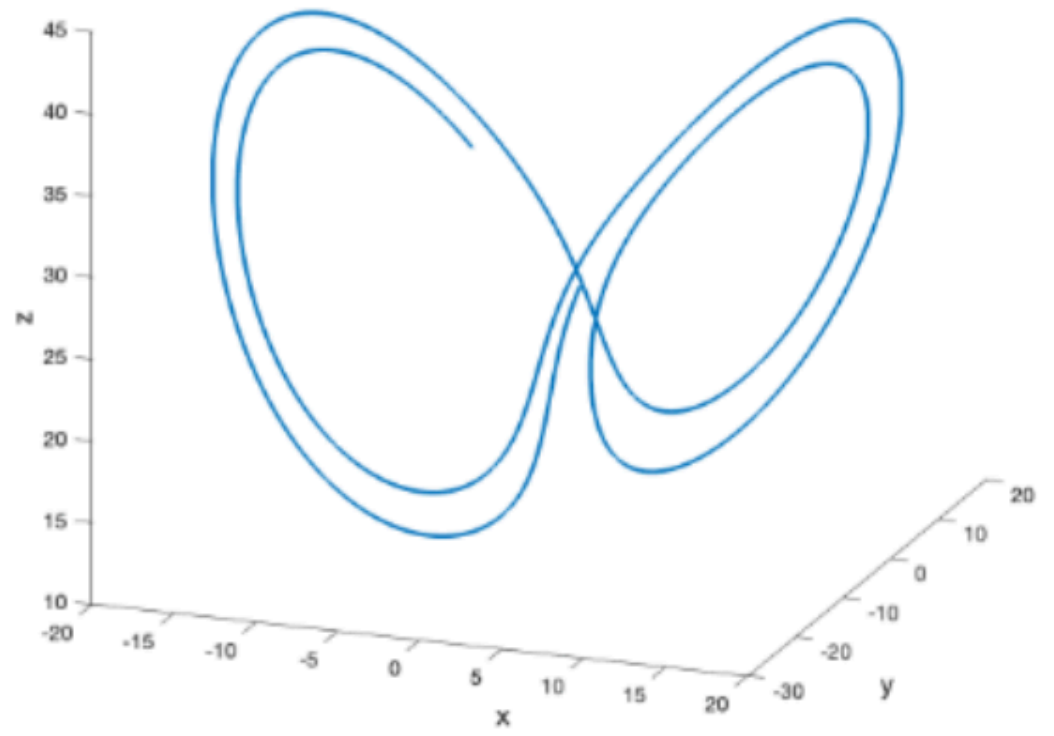


(b) Dynamics of $\boldsymbol{\sigma}^*(\rho)$

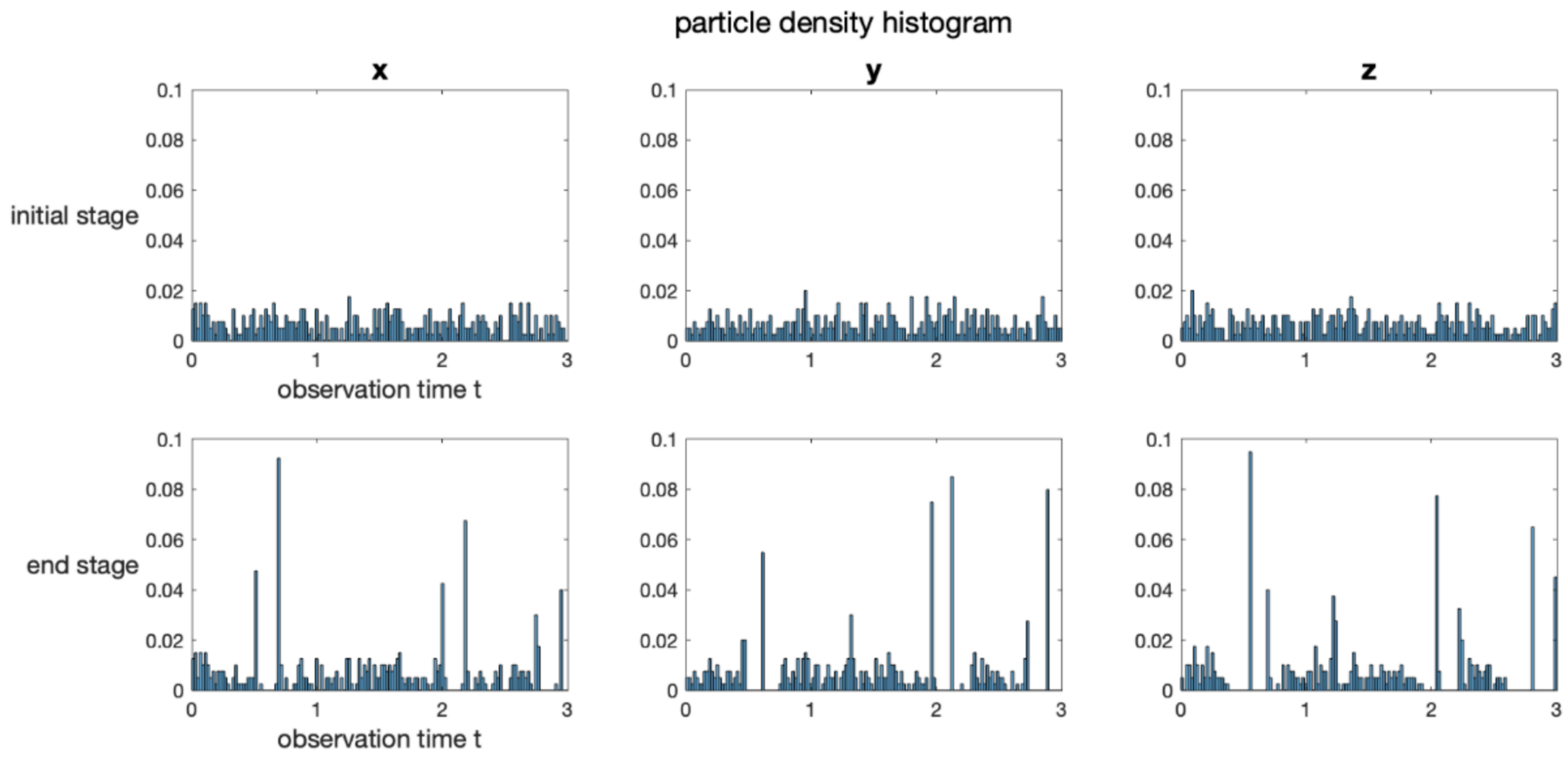
Numerical performances

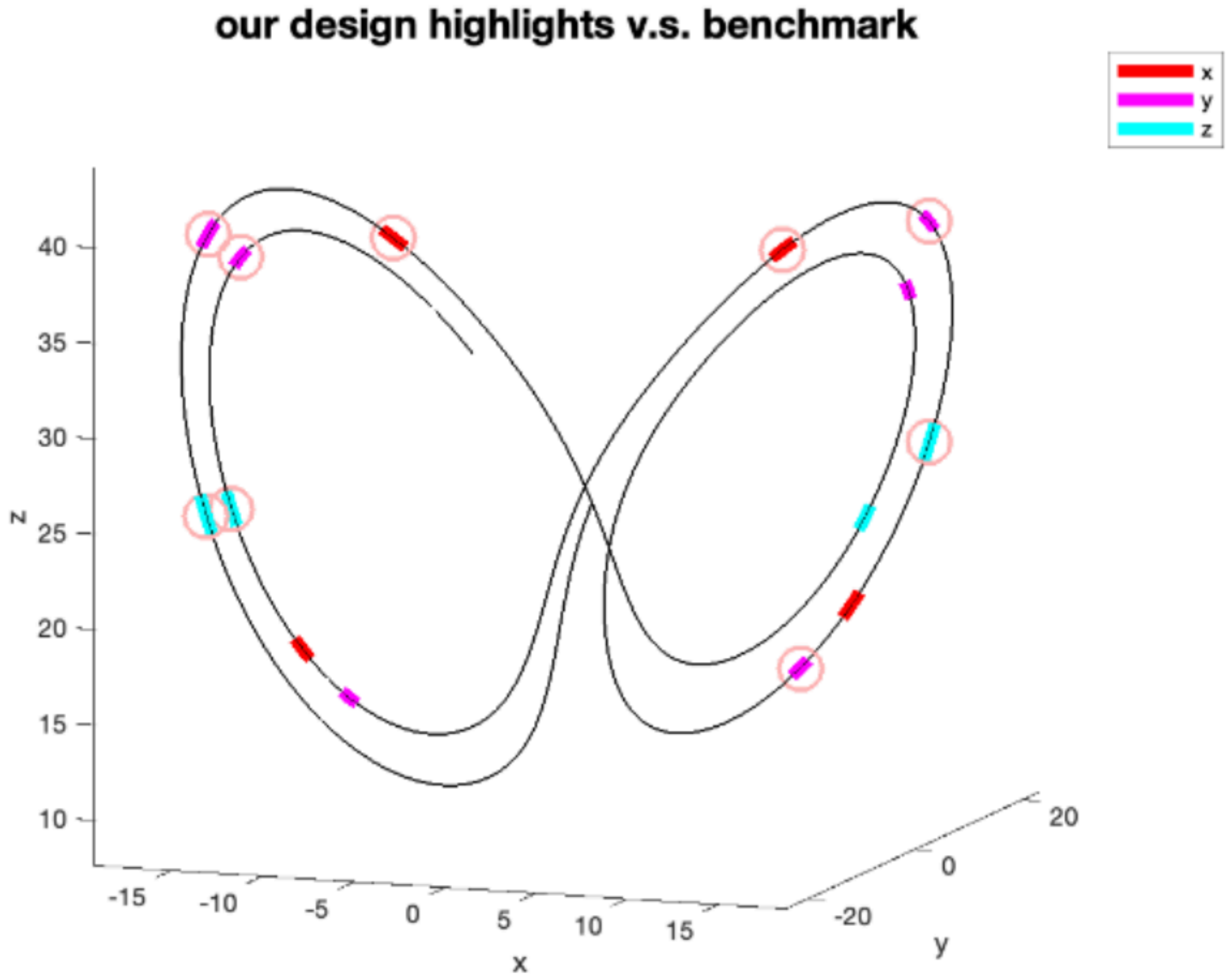
We examine the approach on Lorenz 63 model:

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}\tau} = \alpha(y-x) \\ \dfrac{\mathrm{d}y}{\mathrm{d}\tau} = x(\gamma - z) - y \\ \dfrac{\mathrm{d}z}{\mathrm{d}\tau} = xy - \beta z. \end{cases}$$



Design results on important observation times

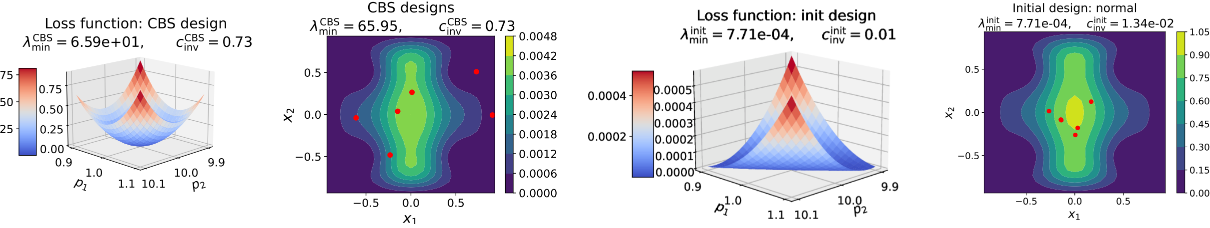




A Sampling Approach to Experimental Design (preprint)

Kathrin Hellmuth, Christian Klingenberg, Qin Li

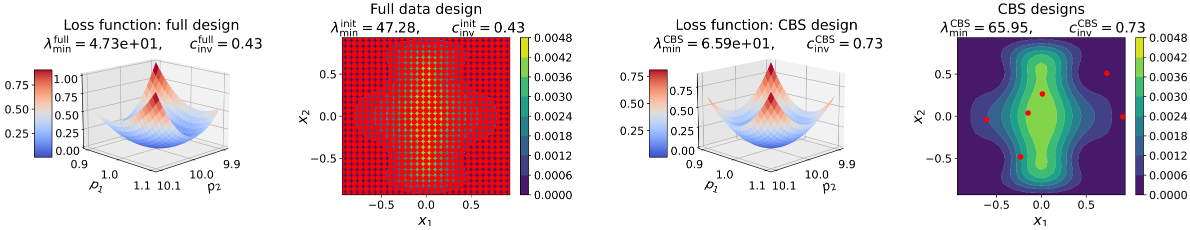
Inverse problem: Infer p from $y = F(p)(+\eta)$



Studienstiftung des deutschen Volkes
Bayerisches Staatsministerium für Wissenschaft und Kunst
UNI WÜ
WISCONSIN UNIVERSITY OF WISCONSIN–MADISON
Caltech

Sampling Approach to Experimental Design

- Starting point: Observe phenomenon everywhere $\Rightarrow p \in \mathbb{R}^P$ reconstructible, full data sensitive
- goal: find small subset of sensitive data points.



Local sensitivity is encoded in sensitivity matrix $J = J_p F(p^*)$, with Random Matrix Multiplication:

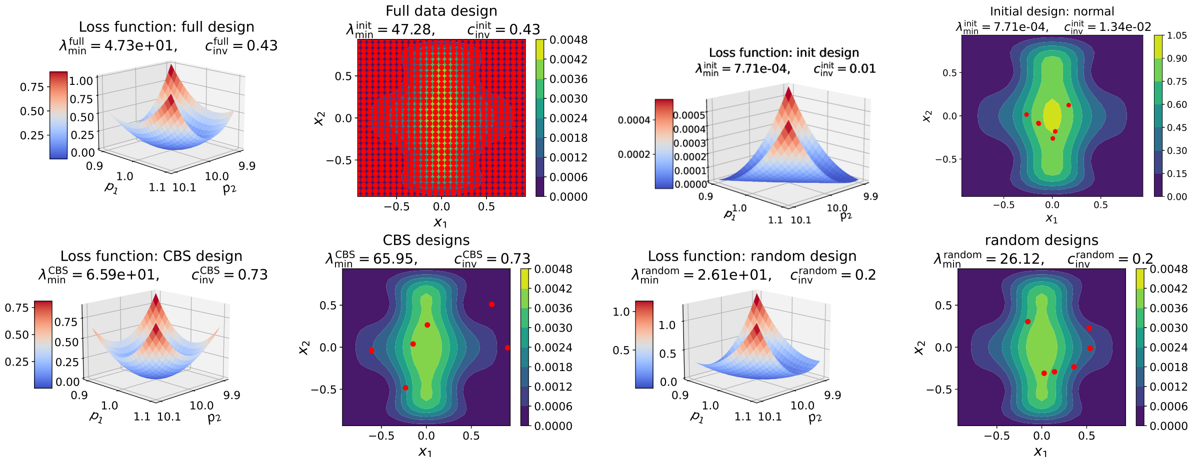
Theorem

Random sampling of data according to a sensitivity informed distribution and reweighting yields a sensitive design, with high probability.

ch

Application to Schrödinger potential reconstruction

$$(-\Delta + p)u_p = 10^4 \text{ in } X := [-1,1]^2,$$
$$u_p = 0 \text{ on } \partial X.$$



Möbius inversion meets tensors: inference by Edgeworth series

Stephen Huan[1] Andreas Robertson[2] Youssef Marzouk[3] Florian Schäfer[4]

[1]Carnegie Mellon University
Advisors: Andrej Risteski, Nick Boffi
https://cgdct.moe

[2]Sandia [3]MIT [4]Georgia Tech

ProbSciML 2025-06-10

The problem

Generative modeling from samples

Formalism of measure transport: seek *transport map* T pushing base distribution $z \sim \mathcal{N}(0, \mathsf{Id})$ to data $T(z) \sim X$

Previous works: minimize loss over parametric family

This work: approximate transport from sample cumulants

Cumulant matching

Motivation: data scarcity in scientific applications

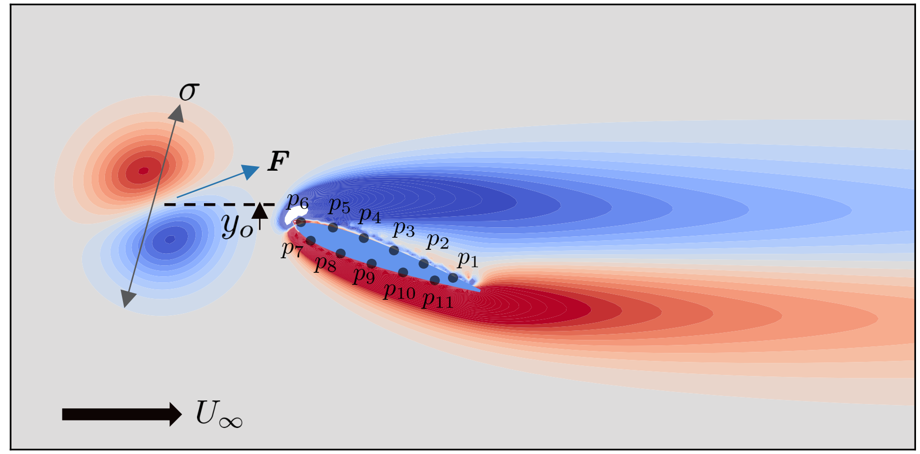
Modernization of [Cornish and Fisher 1938], [McCullagh 1987]

Separation of combinatorics (`sympy`) + numerics (`jax`)

Generalization of methods used in practice (Edgeworth series, moment closure, 2-point statistics in material microstructures)

Machine-learning-driven flow reconstruction and uncertainty quantification using sparse observables

- Unsteady aerodynamic phenomena (gusts)
- Flow changes are imprinted on pressure sensors
- Estimating complex transient aerodynamic using sparse pressure sensors



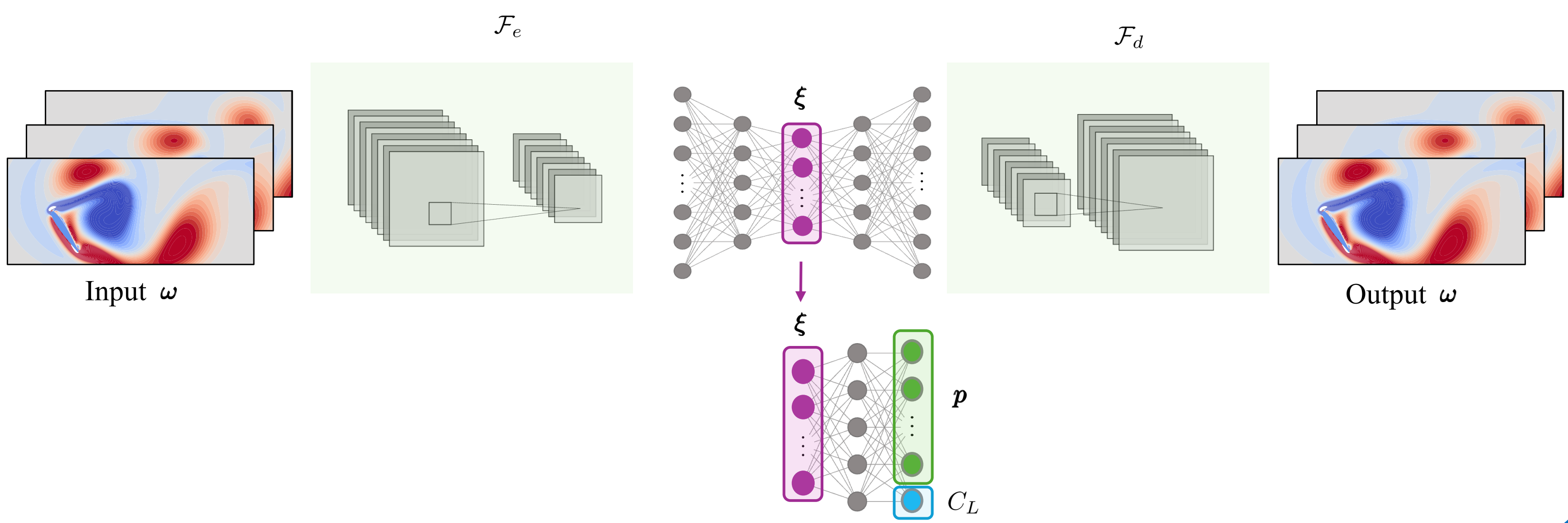
How can we use noisy sparse pressure to estimate flow response?

Use online sequential filters with learned models

UCLA Samueli Mechanical & Aerospace Engineering

Hanieh Mousavi, Anya Jones, and Jeff Eldredge

Low-order Representation of Flow



Learn forecast and observation operators with surrogate models

Forecast: $\boldsymbol{\xi}_t^f = \tilde{f}(\boldsymbol{\xi}_{t-1}^a) + \boldsymbol{w}_t \longrightarrow \pi(\boldsymbol{\xi}_t|\boldsymbol{\xi}_{t-1})$

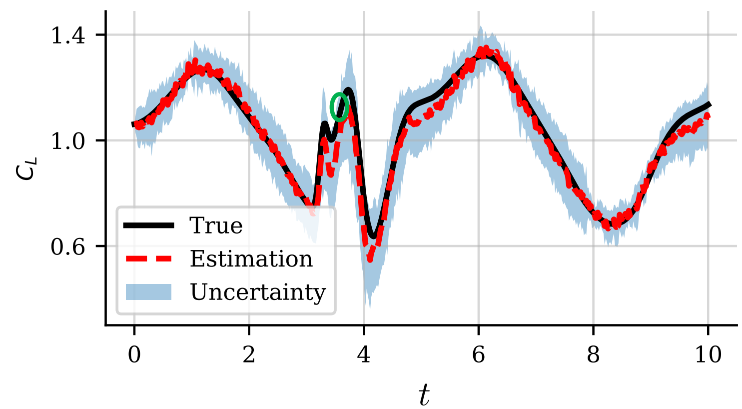
Observation: $\boldsymbol{p}_t = h(\boldsymbol{\xi}_t) + \boldsymbol{\eta}_t \longrightarrow \pi(\boldsymbol{p}_t|\boldsymbol{\xi}_t)$

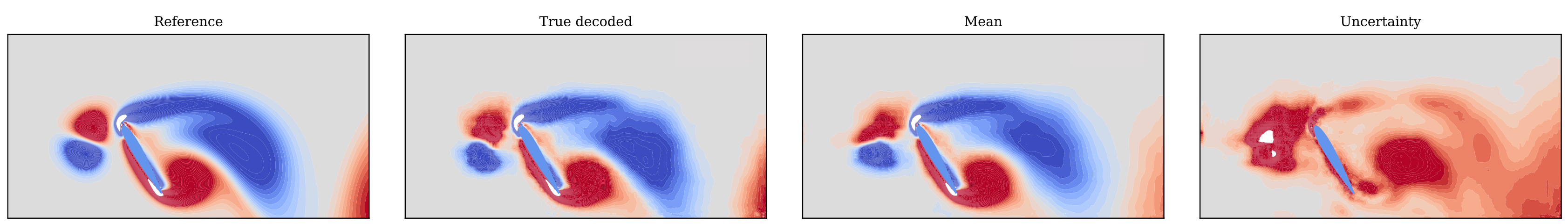
$\Rightarrow$ $\pi(\boldsymbol{\xi}_t|\boldsymbol{p}_{0:t})$ through $\boldsymbol{\xi}_t^a = \boldsymbol{\xi}_t^f + \boldsymbol{K}_t(\boldsymbol{p}_t - h(\boldsymbol{\xi}_t^f))$

Results:

- ❑ Learned surrogates enable efficient, real-time, uncertainty-aware flow estimation from limited measurements.
- ❑ The real-time state estimation in the reduced dimension is fast and efficient.
- ❑ Identifies which sensor combinations contribute the most to correcting the predicted states
- ❑ Useful for control, design, and flight prediction of unsteady aerodynamic systems

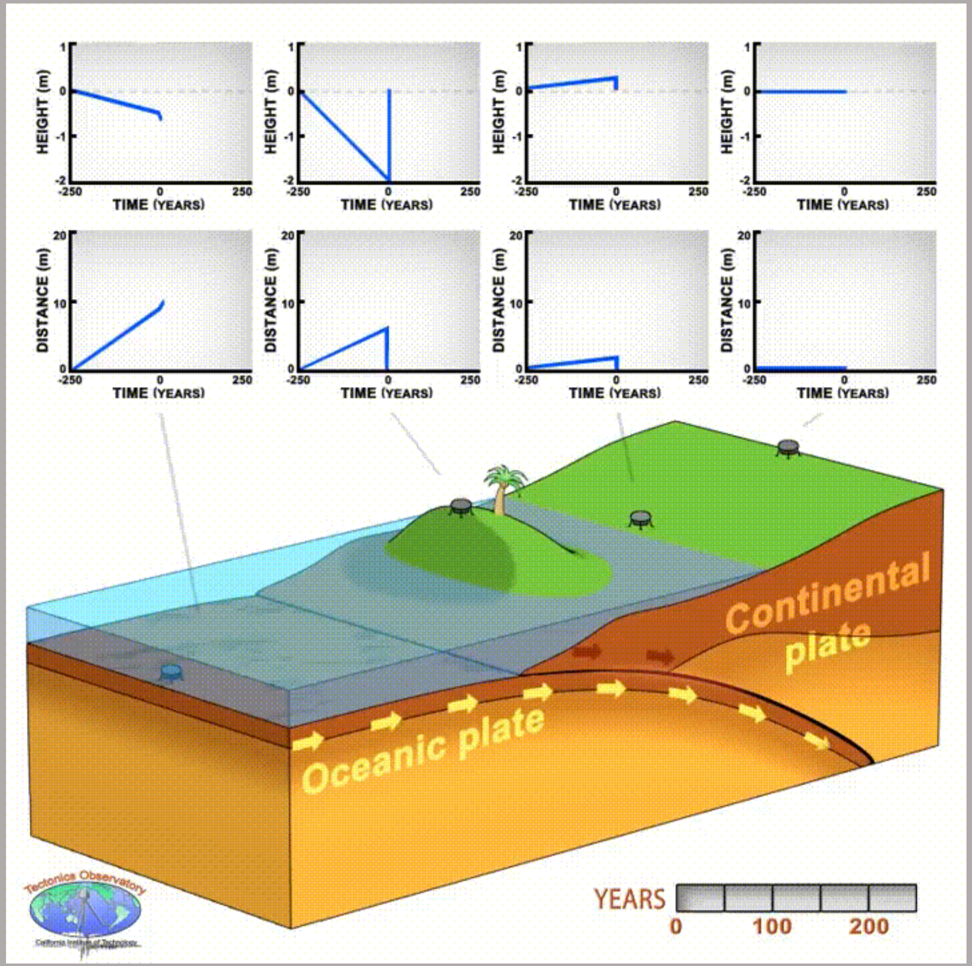
$\alpha = 60°, \ t_o = 3.3$





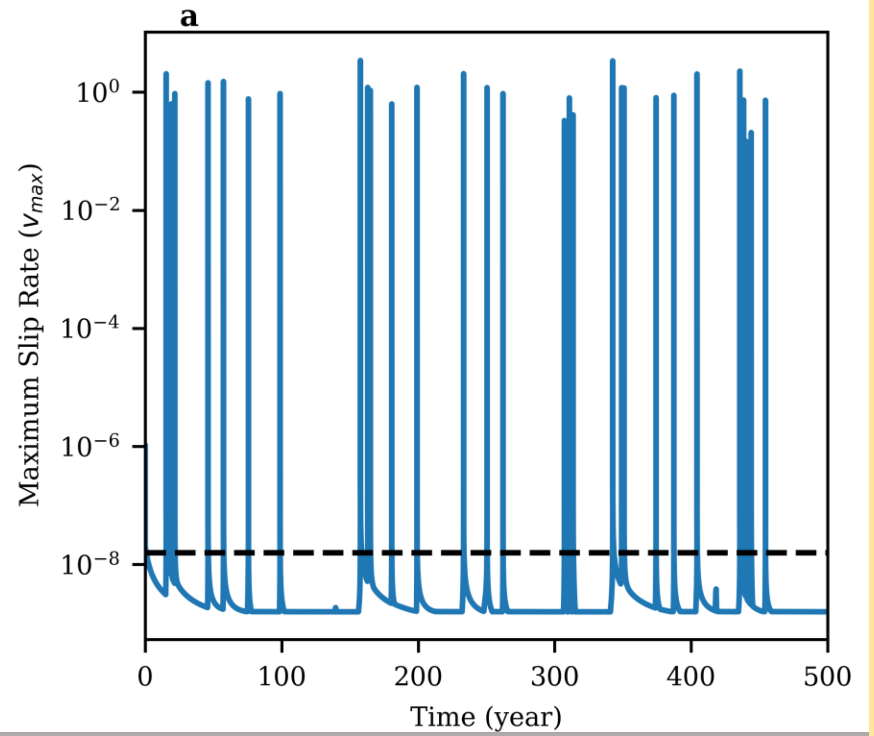
Data Assimilation in a Machine-Learned Reduced-Order Model of a Multiscale Chaotic Earthquake Model

Hojjat Kaveh, Andrew Stuart, Jean Philippe Avouac

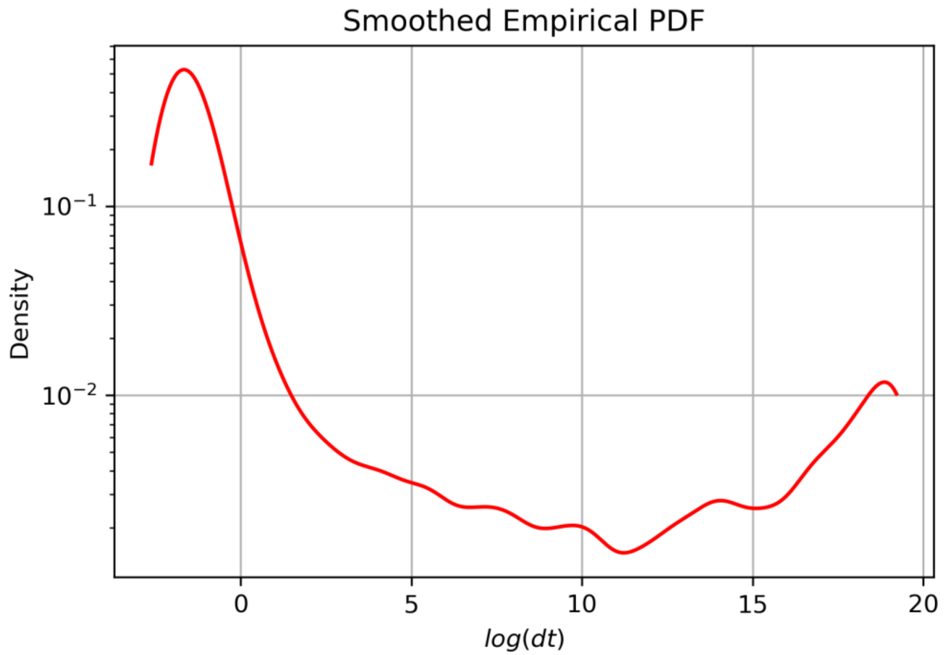


Caltech

Behavior of the PDE: chaotic, multiscale, with extreme events



Time-series of ($||v||_\infty$) shows chaotic evolution with extreme events over different scales.



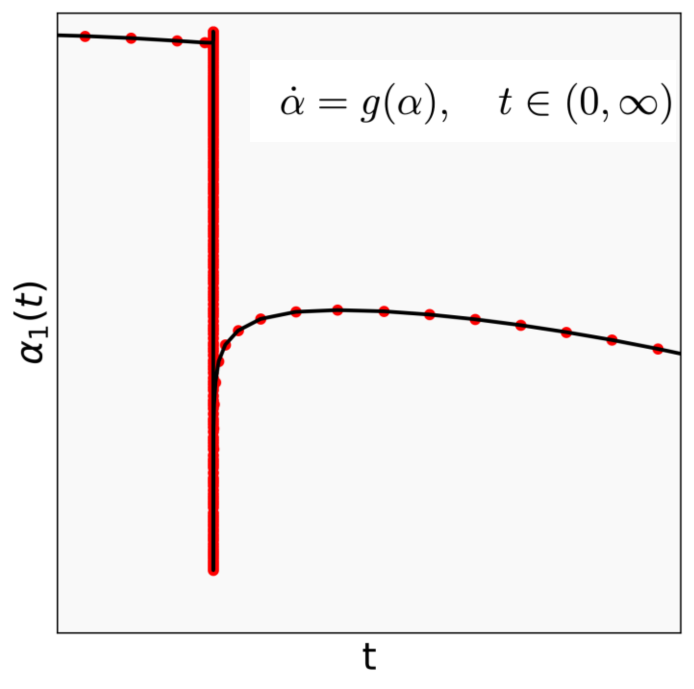
Empirical distribution of the simulation time step dt that ensures certain numerical accuracy.

What will I present in this poster?

Find a reduced coordinate

Find a neural ODE for time-series that appear to be not differentiable

Solve a data assimilation problem using the machine learned forward model



Error Analysis for Learning Time-stepping Operators of Evolution PDEs

Statistical and Computational Challenges in Probabilistic SciML

Meenakshi Krishnan

Joint Work with Ke Chen, Haizhao Yang

IMSI

9th June, 2025

University of Maryland College Park

Introduction

Classical numerical solvers for time-dependent PDE often suffer from stability restrictions/require iterative solvers for complex non-linear systems. Deep Neural Networks (DNNs) potentially offer a fast and stable alternative by learning time-stepping operators.

- Theoretical analysis for learning numerical solvers is severely lacking. Current analysis for solution operators of continuous formulations do not readily extend to their approximations.

Goal: Provide explicit error estimates for learning the time-stepping operator with feed forward neural networks (FNN), offering guidance for efficient design of networks for various classes of PDE.

- Study how the network width, depth, and number of training data impact the learning error.

Reaction Diffusion Equation

Example: Consider a class of reaction-diffusion equations of the form,

$$\partial_t u + \Delta u = f(u) \text{ for } x \in D \subseteq \mathbb{R}^d,\ t \in [0, T],\ u(x, 0) = u_0(x), x \in D \quad (1)$$

where $u_0(x) : C(D)$, the reaction function $f \in C^1(\mathbb{R})$. Assume homogeneous Dirichlet boundary conditions and smooth boundary ∂D.

The semi-discrete form of equation (1) obtained by discretizing in time using the implicit Euler scheme, with step-size Δt:

$$u^1 = [1 - \Delta t \Delta]^{-1} \left(u^0 + f(u^1) \right) =: \Phi_{BE}(u^1, u^0, f). \quad (2)$$

Use fixed point iterations to approximate the solution to the algebraic equation:

$$u^{(0)} = u^0, u^{(i)} = \Phi_{BE}(u^{(i-1)}, u^0, f) \text{ for } i = 1, \ldots, m,\ u^{(m)} =: \Phi_{\mathrm{P}}(u^0, f). \quad (3)$$

$\Phi_{\mathrm{P}} : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{X}_1$ with $\mathcal{X}_1 = (C(D), \|\cdot\|_\infty)$, $\mathcal{X}_2 = (C^1(D), \|\cdot\|_{C^1})$ for multi-input operator learning.

Use encoder-decoders: $E^n_{\mathcal{X}_1 \times \mathcal{X}_2}(u, f) = \left[E^n_{\mathcal{X}_1}(u), E^n_{\mathcal{X}_2}(f) \right]$ where $E^n_{\mathcal{X}_i} : \mathcal{X}_i \to \mathbb{R}^{d_{\mathcal{X}_i}}$ for $i = 1, 2$. Similarly, define the decoder $D^n_{\mathcal{X}_1 \times \mathcal{X}_2}$.

Generalization Error

Consider the target operator to be learned $\Phi_{\mathrm{P}} : C(D) \times C^1(\mathbb{R}) \to C(D)$ defined in (3). Denote $L_{\mathrm{P}} = 1/(1 - L_r \Delta t)$ assuming $L_r \Delta t < 1$ for L_r a uniform Lipschitz constant for reaction function space.

Let Γ_{NN} be the network minimizing training loss among $\mathcal{F}_{\mathrm{NN}}$ architecture with width p, depth $\mathcal{O}(mL)$, maximum norm bounded by M. Then with:

$$Lp = \Omega\left((d_{\mathcal{X}_1} + d_{\mathcal{X}_2})^{\frac{1}{4}} n^{\frac{1}{4}}\right), M \geq \mathcal{O}(\sqrt{\ell_{\max}}), \text{for } \ell_{\max} = (d_{\mathcal{X}_2} + 1) d_{\mathcal{X}_1} + d_{\mathcal{X}_2},$$

$$\boxed{\mathcal{E}_{\mathrm{gen}} \lesssim L_{\mathrm{P}}^2 \log(L_{\mathrm{P}}) (\ell_{\max})^{5/2} n^{-\frac{1}{2}} \log n + \mathcal{E}_{\mathrm{proj}}\,.}$$

The constants in $\lesssim$ solely depend on $p, \ell_{\max}$ and Lipschitz constants of encoder-decoders.

No Curse of Dimensionality!

- In practice, Newton's method is used to solve the non-linearity due to the step-size restriction of Picard's method.
- Can also extend results to other classes of evolution equations like parabolic equations with forcing terms, conservation laws.

References i

K. Chen, C. Wang, and H. Yang.
Deep operator learning lessens the curse of dimensionality for pdes.
arXiv preprint arXiv:2301.12227, 2023.

H. Liu, H. Yang, M. Chen, T. Zhao, and W. Liao.
Deep nonparametric estimation of operators between infinite dimensional spaces.
Journal of Machine Learning Research, 25(24):1–67, 2024.

Thank You

Error Analysis of OT Filters - Conditional OT

Reference measure: $\eta_U \in \mathbb{P}(\mathcal{U})$. Target measure: $\nu(\cdot \mid y) \in \mathbb{P}(\mathcal{U})$.

Brenier potential: There exists a convex potential $\phi^\dagger : \mathcal{Y} \times \mathcal{U} \mapsto \mathbb{R}$,

such that $T^\dagger(y, \cdot) = \nabla_u \phi^\dagger(y, \cdot)$, and $T^\dagger(y, \cdot)\#\eta_{\mathcal{U}} = \nu(\cdot \mid y)$

Empirical dual objective: Sample $(y_i, v_i)_{i=1}^N \sim \eta$ and $(y_i, u_i)_{i=1}^N \sim \nu$.

$$\hat{\phi} = \arg\min_{\phi \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N \phi(y_i, v_i) + \phi^*(y_i, u_i).$$

Error bound:

Slow rate: $\mathbb{E}\|\nabla_u \hat{\phi} - \nabla_u \phi^\dagger\|^2_{L^2_{\eta_{\mathcal{U}}}} \lesssim$ **Bias** $+O(n^{-1/2})$

Fast rate: $\mathbb{E}\|\nabla_u \hat{\phi} - \nabla_u \phi^\dagger\|^2_{L^2_{\eta_{\mathcal{U}}}} \lesssim$ **Bias** $+O(n^{-1})$

OT Filter & Transport Viewpoint

Hidden State: $U_\tau \in \mathbb{R}^n$. Observation: $Y_\tau \in \mathbb{R}^m$.

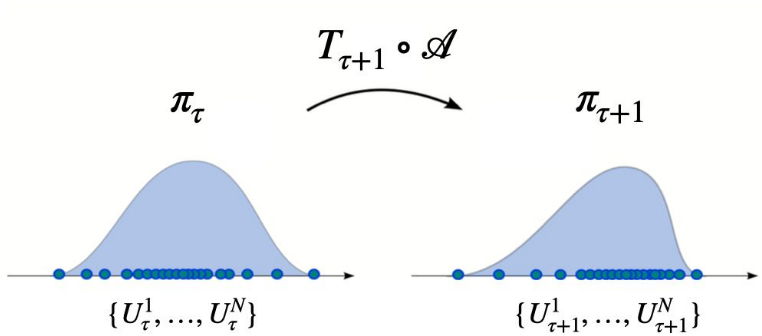
Posterior: Find the posterior $\pi_\tau(\cdot) = P(U_\tau \in \cdot \mid Y_1, \ldots, Y_\tau)$.

Recursive update:

(Propagation) $\pi_\tau \mapsto \mathscr{A}\pi_\tau = \tilde{\pi}_{\tau+1} := \int_{\mathscr{U}} a(\cdot \mid u)\, d\pi_\tau(u),$

(Conditioning) $\tilde{\pi}_{\tau+1} \mapsto \mathscr{B}_y \tilde{\pi}_{\tau+1} = \pi_{\tau+1} := \dfrac{h(y \mid \cdot)\tilde{\pi}_{\tau+1}(u)}{\int_{\mathscr{U}} h(y \mid u)\, d\tilde{\pi}_{\tau+1}(u)}.$

Conditional OT approach:



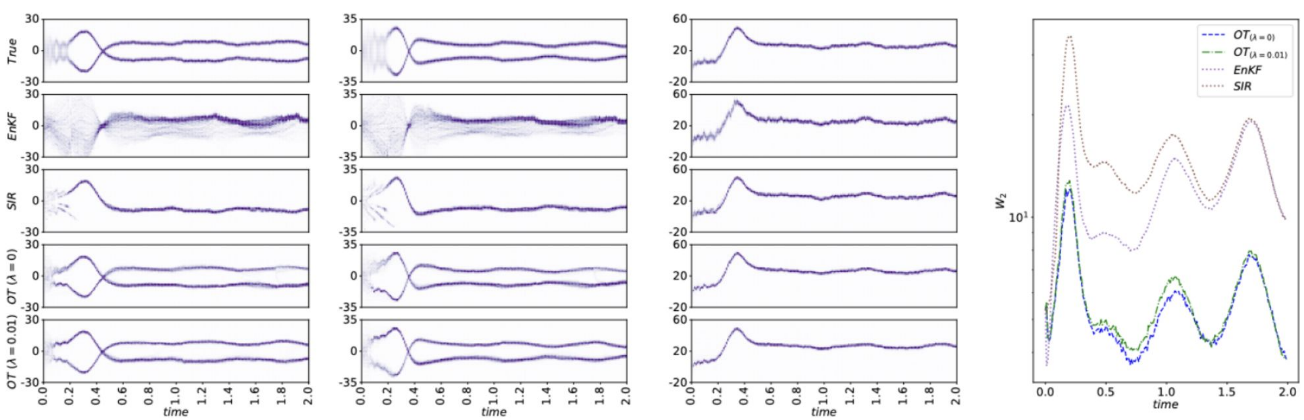
Filtering Result & Numerics

Error bound:

Slow rate: $$\mathbb{E}D(\hat{\nu}^t \| \nu^t) \lesssim \sum_{\tau=1}^{t} \textbf{Bias}_\tau + O(n^{-1/2})$$

Fast rate: $$\mathbb{E}D(\hat{\nu}^t \| \nu^t) \lesssim \sum_{\tau=1}^{t} \textbf{Bias}_\tau + O\left(\left(\frac{\log n}{n}\right)^{\frac{2}{2+\gamma}}\right), \gamma \in [0,1)$$

Numerics: Lorenz 63 model with observing the first and third states.



(a) $U_t(1)$ (b) $U_t(2)$ (c) $U_t(3)$ (d) W_2 distance.

Unifying and extending Diffusion Models through PDEs for solving Inverse Problems

From Probabilistic Inference to Conditional Generation

Authors:

Agnimitra Dasgupta,
Alexsander Marciano da Cunh,
Ali Fardisi,
Mehrnegar Aminy,
Brianna Binder,
Bryan Shaddy,
Assad A Oberai

Probabilistic Inverse Problem

Let $\boldsymbol{X} \in \mathbb{R}^{N_x}$ be the vector to be inferred

Let P_X denote prior information (can be non-Gaussian, samples)

Let $\boldsymbol{Y} \in \mathbb{R}^{N_y}$ be the measurement vector

Let $P_{Y|X}$ (can be black-box) denote the composition of the measurement and forward map

Given $P_X, P_{Y|X}$ and an instance of $\boldsymbol{Y} = \boldsymbol{y}$, characterize $P_{X|Y}(\boldsymbol{x}\,|\,\boldsymbol{y})$

Conditional Generative Problem

Generate samples $\boldsymbol{x}_i$ from P_X

Generate samples $\boldsymbol{y}_i$ using $P_{Y|X}(\boldsymbol{y}\,|\,\boldsymbol{x}_i)$ - apply the measurement and forward models

The dataset $\mathcal{S} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ is drawn from P_{XY}

Given $\mathcal{S} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, and an instance of $\boldsymbol{Y} = \boldsymbol{y}$, generate samples from $P_{X|Y}(\boldsymbol{x}\,|\,\boldsymbol{y})$

Develop conditional diffusion models to accomplish this

Conditional Diffusion Models

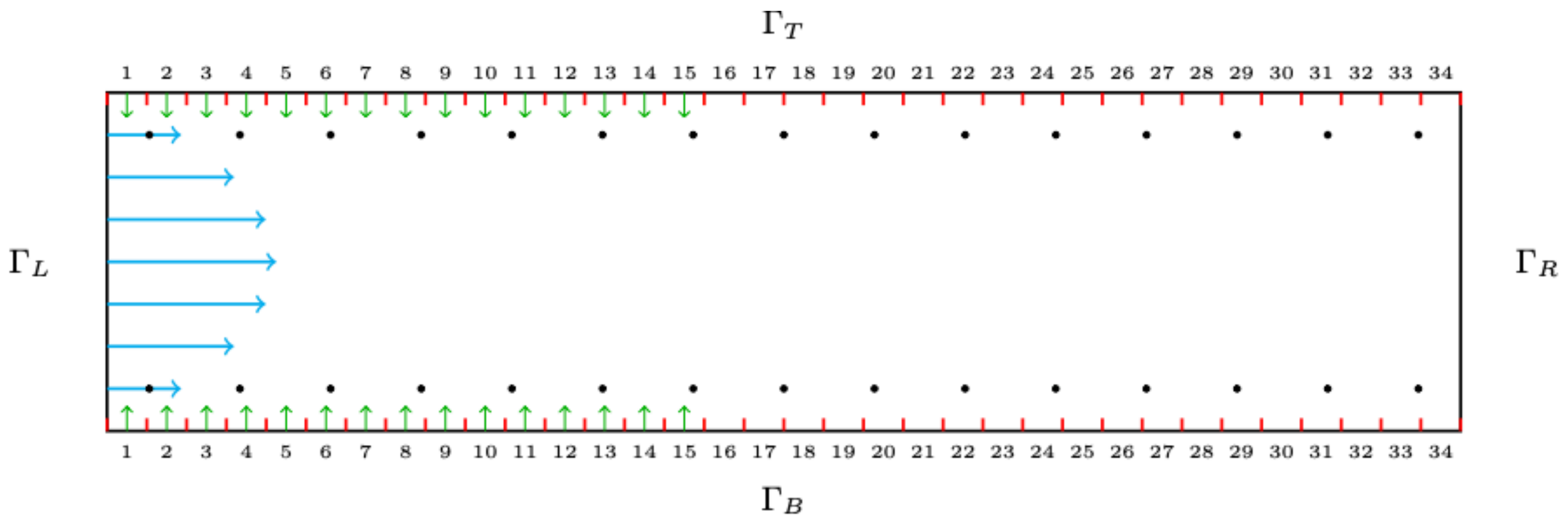
Conditional diffusion models

- Basic idea: sample from a Gaussian distribution and transform to samples from the desired conditional distribution
- Existing approaches: DDPM, NCSN, Stochastic differential equations
- Our approach: work directly at the probability density function (pdf) level, and use elementary concepts from linear PDEs.

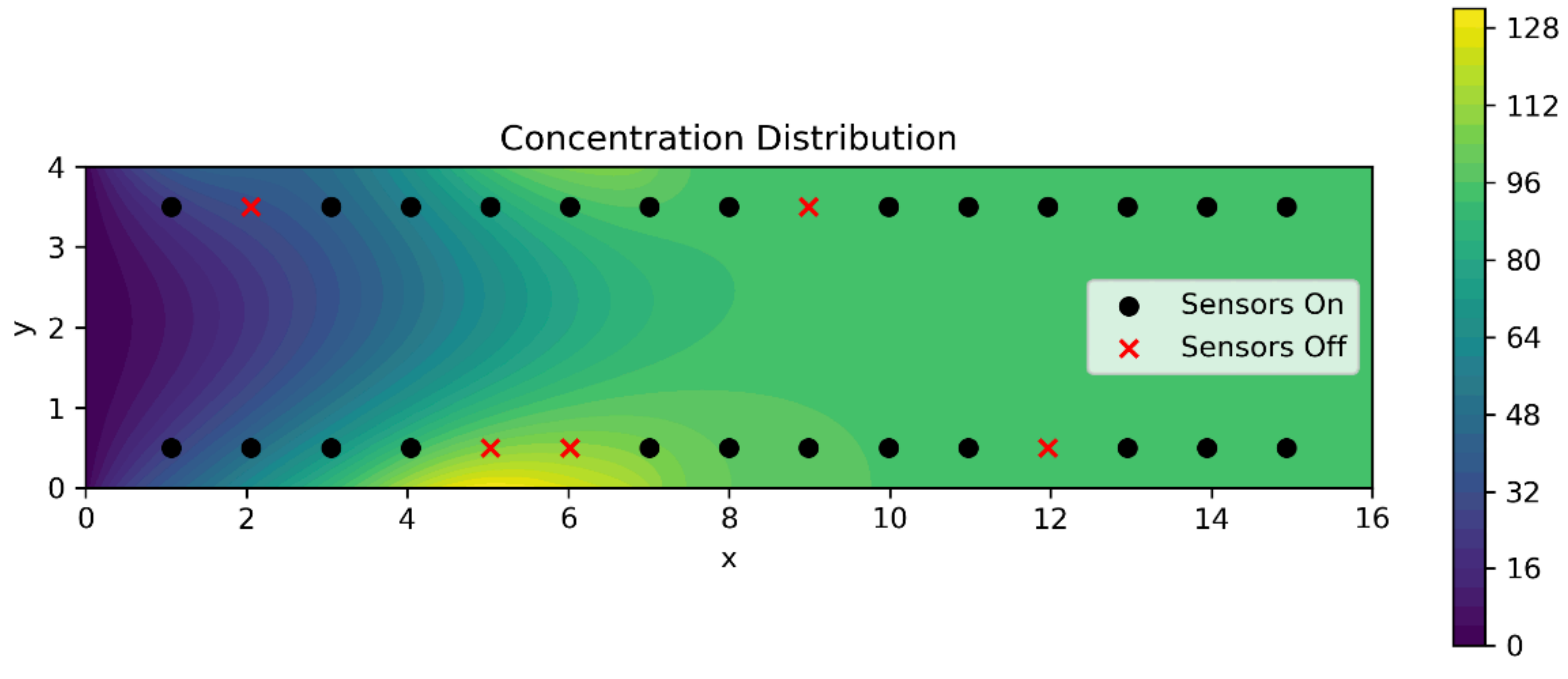
Contributions of our work

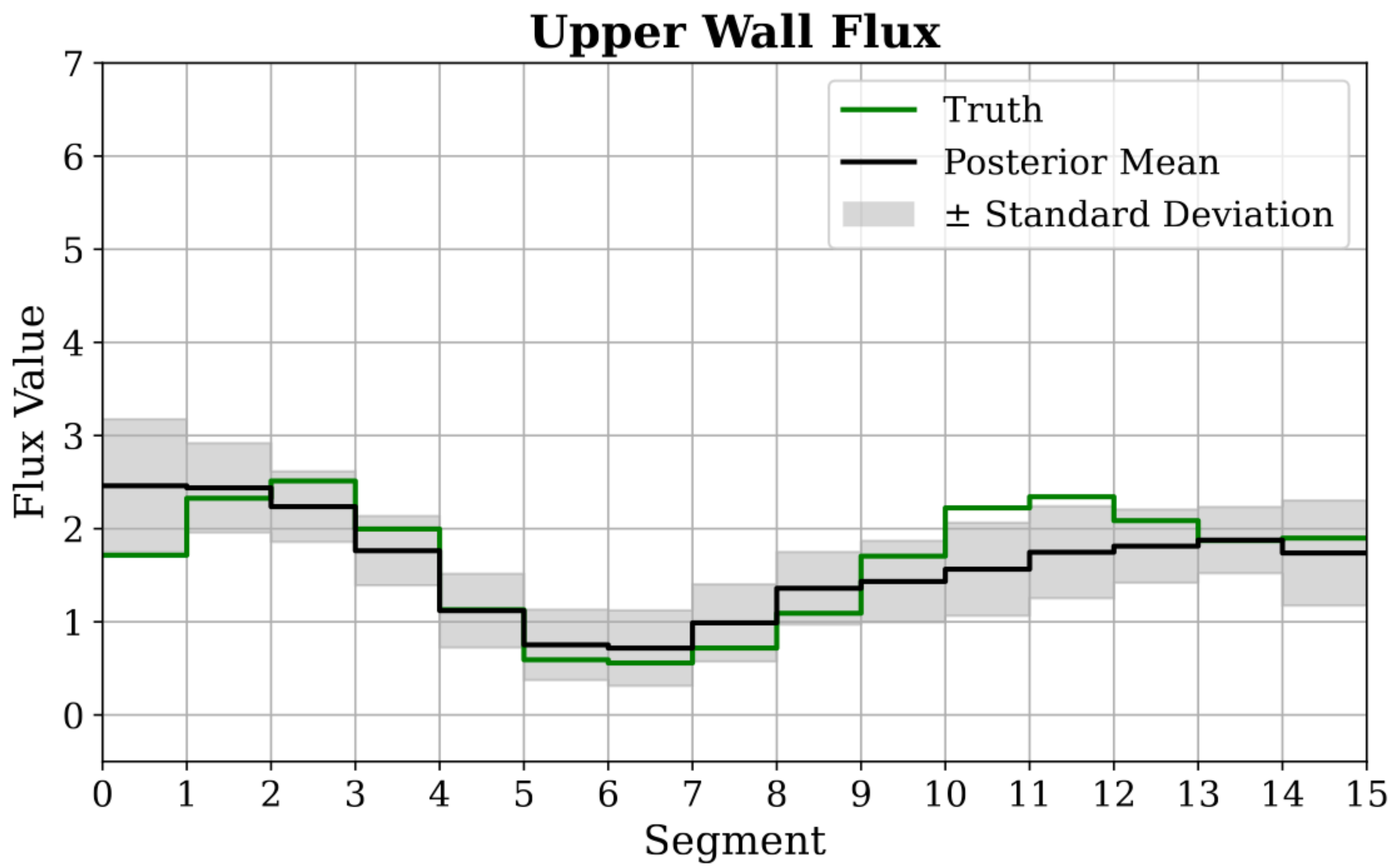
1. Provides a unified exposition of variance exploding and variance-preserving diffusion models
2. Identify and introduce a new class of variance-preserving diffusion formulations
3. The unified framework also includes a family of sampling strategies — including Langevin Monte Carlo, probability flow ODEs, and SDE-based samplers as special cases
4. Also, by conditioning the diffusion model on both the measurement and a vector parameterizing the measurement operator, we enable a single model to solve inverse problems involving multiple measurement operators.

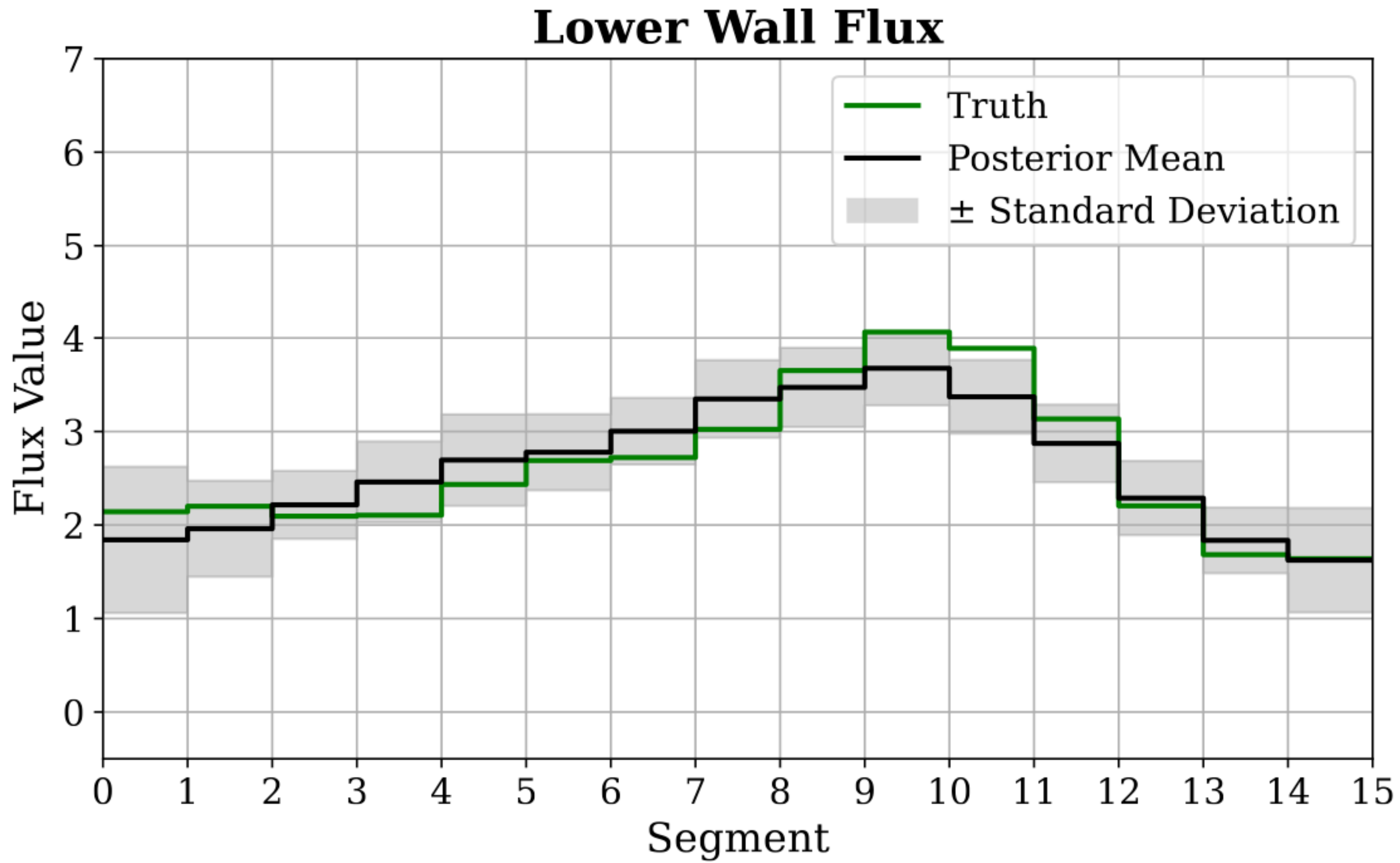
Numerical Example



$$\nabla \cdot (\mathbf{a}u) - \kappa \nabla^2 u = 0$$







Problem Setup

Observations (for each $m = 1, \ldots, M$):

$$\left\{ \left(x_i^m,\ u_m(x_i^m)\right) \right\}_{i=1}^{N_m} \quad \text{(typically } N_m \ll \text{size of a fine mesh)}$$

Assumed governing PDE:

$$\mathcal{P}(u_m)\,(x) = f_m(x), \qquad x \in \Omega, \tag{1}$$

$$B(u_m)\,(x) = g_m(x), \qquad x \in \partial\Omega \quad \text{(or initial data)}. \tag{2}$$

Model parameterisation:

$$\mathcal{P}(u_m)(x) = P\big(S(x, u_m)\big), \quad S(x, u_m) = \big(x,\ u_m(x),\ (L_1 u_m)(x), \ldots, (L_k u_m)(x)\big).$$

We fix S, and want to learn the function P.

Goals

1. Recover the possibly nonlinear operator $\mathcal{P}(u)(x) = P(S(x, u))$.
2. Reconstruct u_m on a fine mesh with high accuracy.

Notes

- If **(1)** is solved, plug $\widehat{\mathcal{P}}$ into a standard PDE solver to achieve **(2)**.
- If **(2)** is solved first, **(1)** reduces to a (possibly nonlinear) regression problem on $\big\{S(x_i^m, u_m), f_m(x_i^m)\big\}$.

Our approach

Two–step (e.g. SINDy):

1. Regress to obtain $\hat{u}_m$.
2. *Fix* $\hat{u}_m$ and regress for $\hat{P}$ on $S(x, \hat{u}_m)$.

Issue: ignores coupling between state and operator inference.

Proposed simultaneous scheme:
Estimates should match the observed data:

$$\hat{u}_m(x_i^m) \approx u_m(x_i^m),$$

and satisfy the PDE on sampled collocation points

$$\hat{P}\big(S(x_j, \hat{u}_m)\big) \approx f_m(x_j).$$

Joint optimisation problem

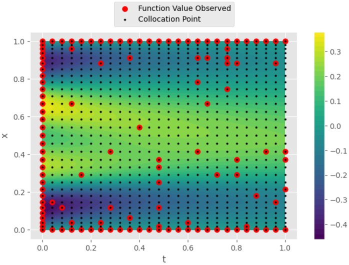
$$\begin{aligned}\min_{\substack{\hat{P}\in\mathcal{Q}\\ \hat{u}_m\in\mathcal{U}}} \ & \frac{1}{2}\|\hat{P}\|_{\mathcal{Q}}^2 + \frac{\lambda}{2}\sum_{m=1}^{M}\|\hat{u}_m\|_{\mathcal{U}}^2 \\ & + \frac{c_1}{2}\sum_{m,i}\big(\hat{u}_m(x_i^m) - u_m(x_i^m)\big)^2 \\ & + \frac{c_2}{2}\sum_{m,j}\Big(\hat{P}\big(S(x_j, \hat{u}_m)\big) - f_m(x_j)\Big)^2 \\ & + \frac{c_3}{2}\sum_{m,b}\big(B(\hat{u}_m)(x_b) - g_m(x_b)\big)^2.\end{aligned}$$

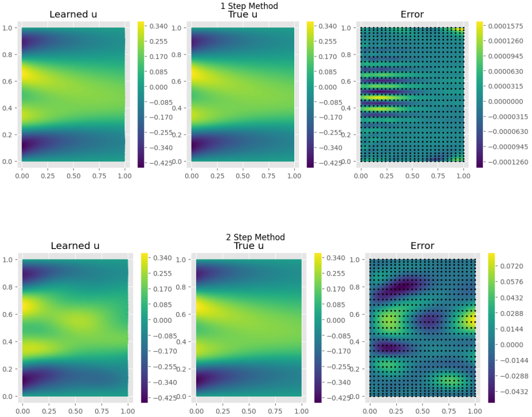
Algorithmic recipe

- Choose RKHSs $\mathcal{U}$ (states) and $\mathcal{Q}$ (operators).
- Representer theorem gives a finite basis for $\hat{u}_m, \hat{P}$.
- Solve resulting nonlinear least–squares via Levenberg–Marquardt.

Results on Burgers Equation

$$u_t = \frac{1}{2}\, u\, u_x + 0.01\, u_{xx}$$





Can we trust Gradient Descent?

in $\mathscr{P}_2$

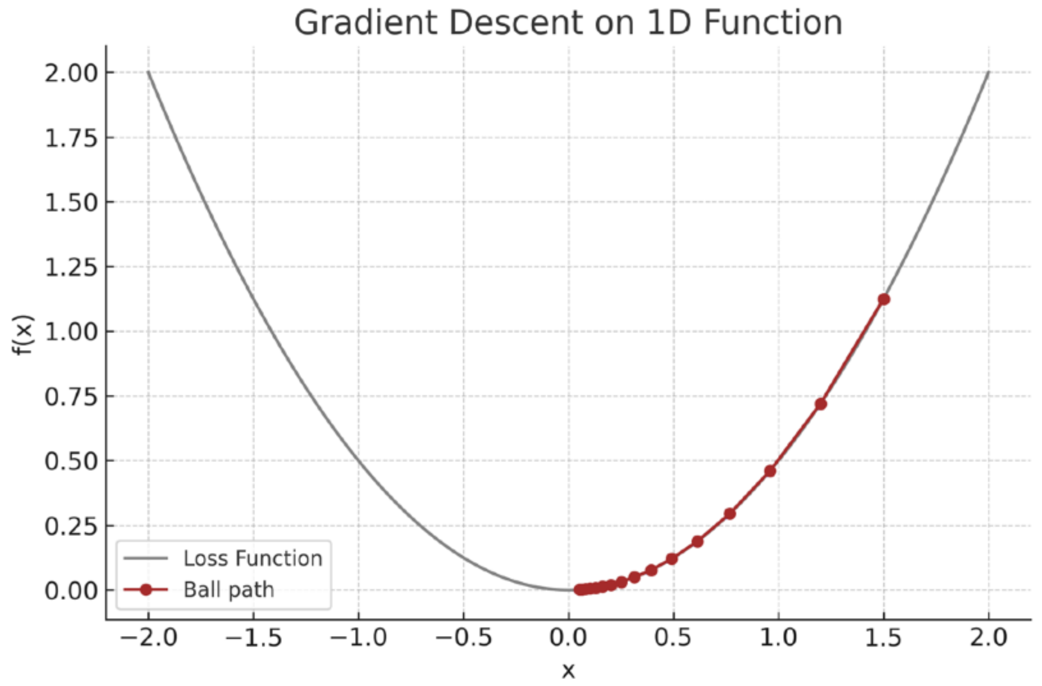
Yewei Xu, Qin Li

Can we trust gradient descent in $\mathscr{P}_2$

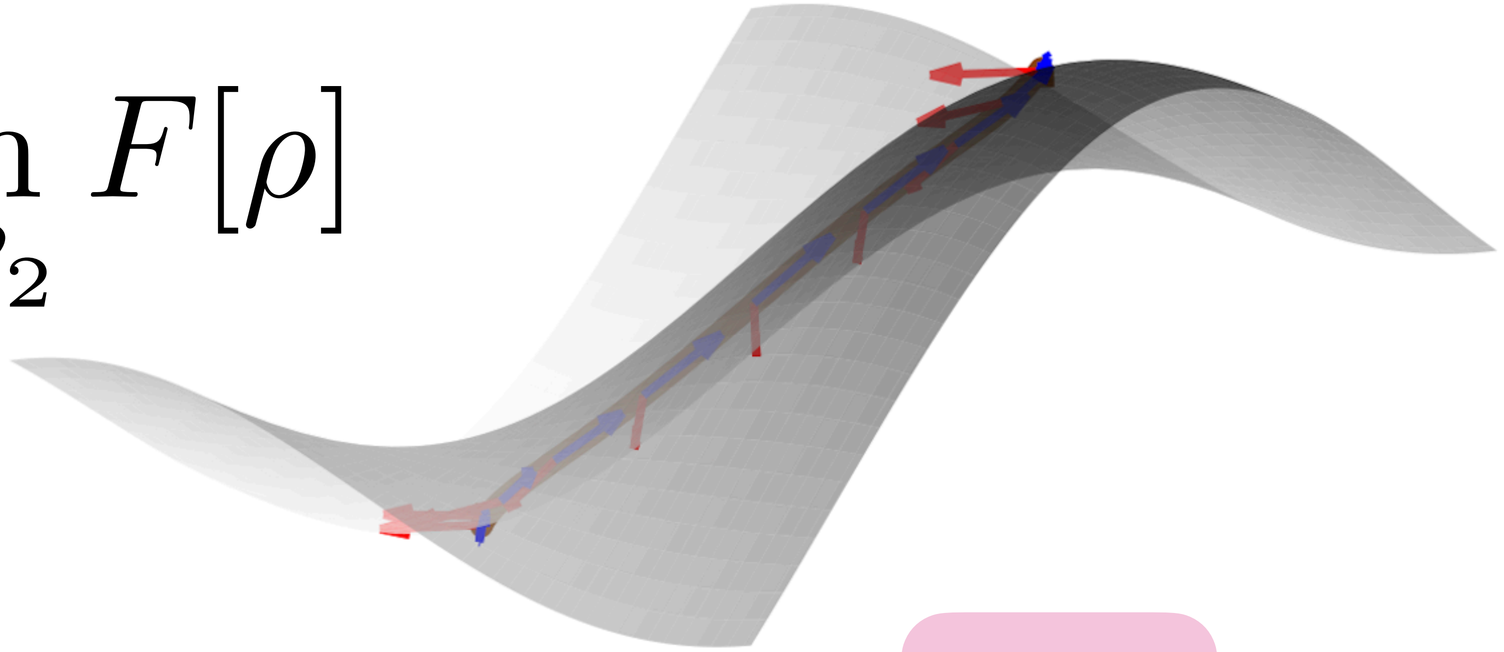
$$\min f(x)$$

$$\dot{x} = -\nabla_x f$$

$$x_{n+1} = x_n - h\nabla_x f(x_n)$$



$$\min_{\rho\in\mathcal{P}_2} F[\rho]$$



$$\partial_t\rho = -\nabla_{W_2}F = \nabla_x\cdot\left(\rho\nabla\frac{\delta F}{\delta\rho}\right)$$

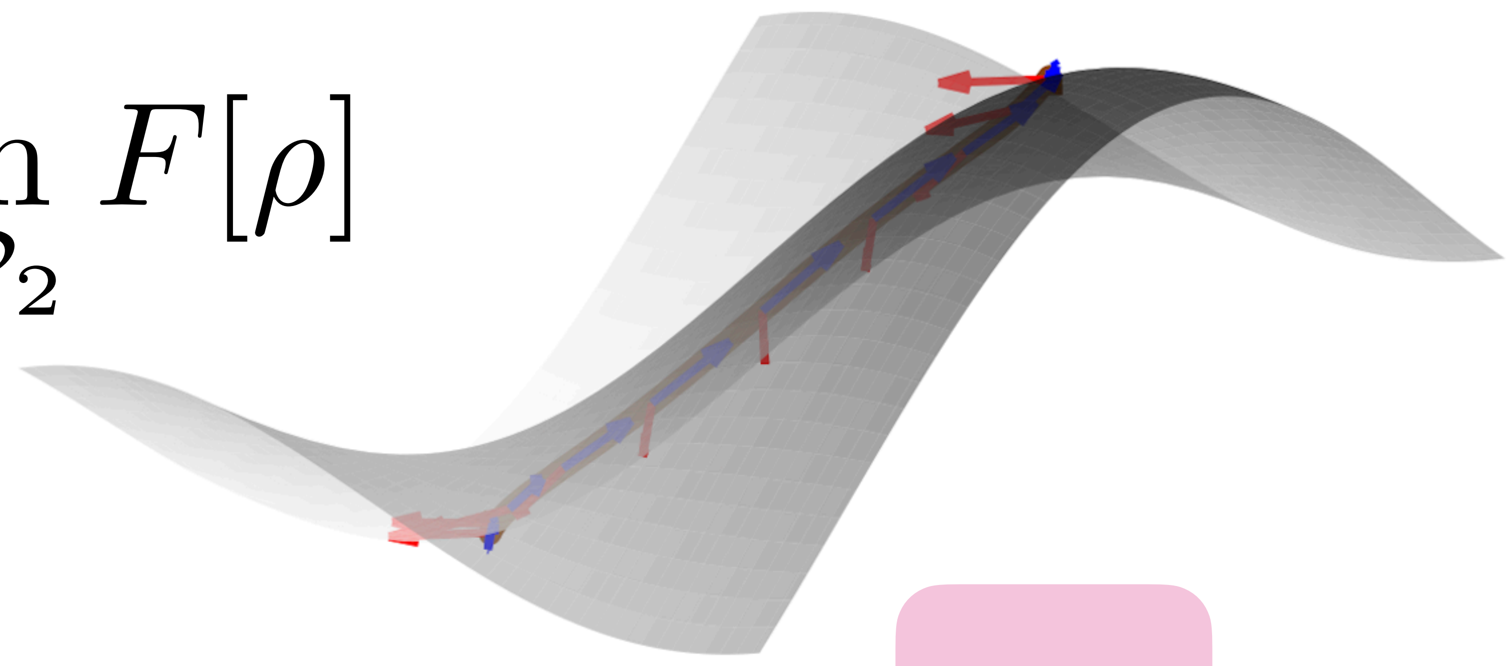
$$\rho_{n+1} = (\mathsf{I} - h\mathsf{T}[\rho_n])_{\#}\rho_n$$

Can we trust gradient descent in $\mathscr{P}_2$

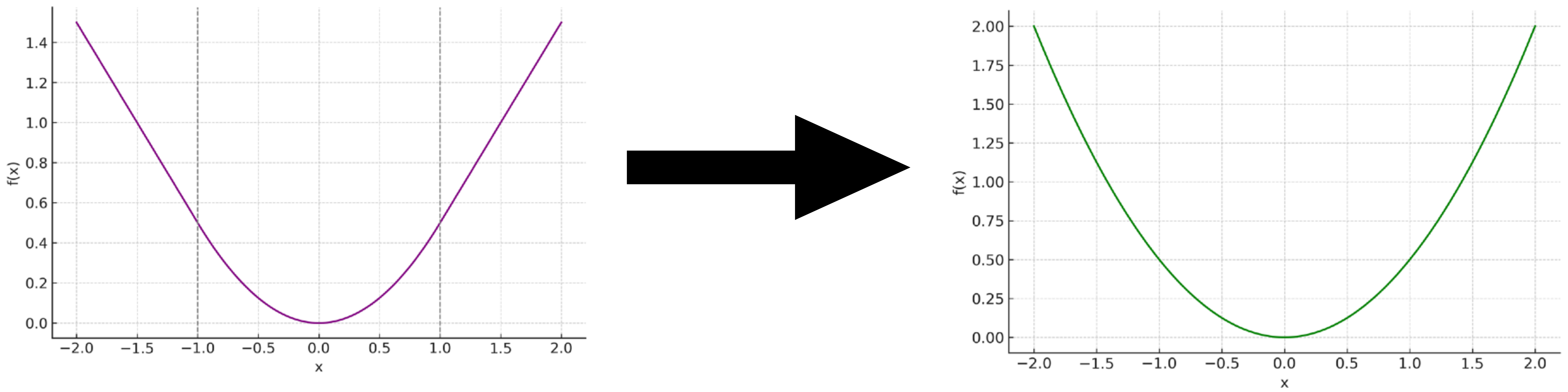
Theorem: for $F[\rho] = \mathrm{KL}(\rho \,|\, \mathcal{N}(0,1))$

never converge

$$\min_{\rho \in \mathcal{P}_2} F[\rho]$$



$$\partial_t \rho = -\nabla_{W_2} F = \nabla_x \cdot \left(\rho \nabla \frac{\delta F}{\delta \rho}\right)$$



$$\rho_{n+1} = (\mathsf{I} - h\mathsf{T}[\rho_n])_{\#}\rho_n$$

- $\nabla \dfrac{\delta F}{\delta \rho}$ may exist even if the gradient does not
- If both exist, they coincide

Forward-Euler time-discretization for W-GF can be wrong. Y. Xu and Q. L, arxiv: 2406.08209