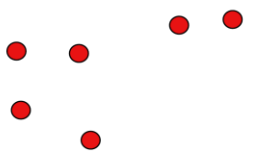
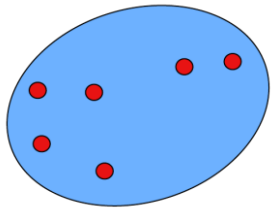
# Learning Where to Learn

Nicolas Guerra

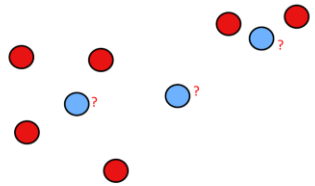
Goal: Find optimal $\nu$ such that $\hat{F}$ performs well on unseen distributions.
($\hat{F}$ learned $F : X \to Y$ using data $x \sim \nu$)



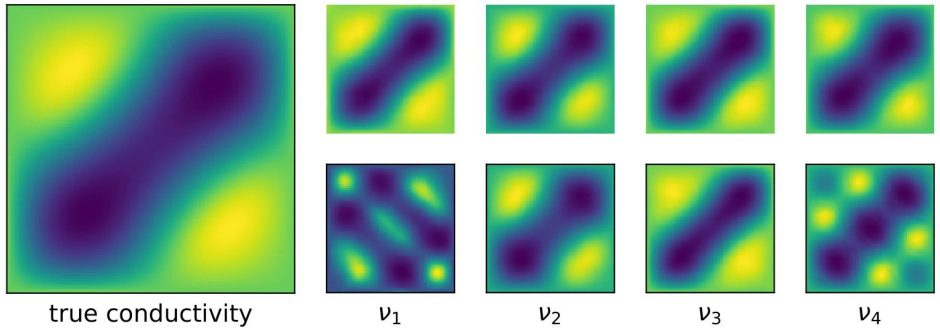
*Family of test distributions.*



*Ideally, train on a large enough distribution (blue).*

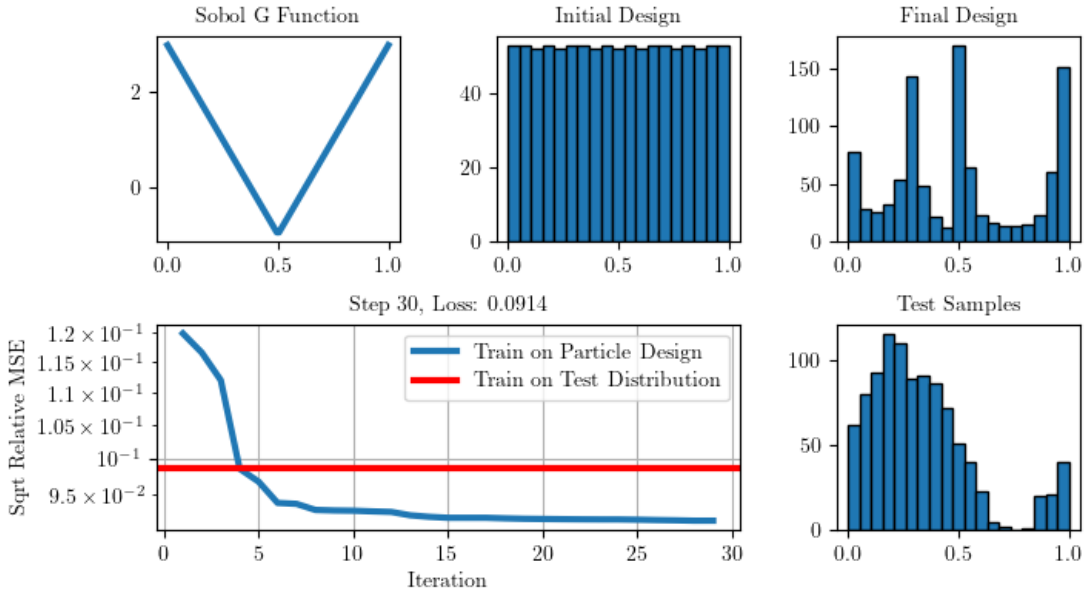


*In practice, constrained.*




Ways to represent $\nu$:

- Parametric, e.g. Gaussian process
- Nonparametric, i.e. $\nu = \sum_{i=1}^{N} w_i \delta_{x_i}$




Applications/Inverse Problems:

- Electrical Impedance Tomography
- Darcy Flow
- Radiative Transfer

Cornell University

# CGKN: Conditional Gaussian Koopman Network

## A Stochastic Digital Twin for Modeling, Forecasting, Data Assimilation, and Uncertainty Quantification

**Nan Chen**[1*], Chuanqi Chen[2], Zhongrui Wang[1] and Jinlong Wu[2]
[1]UW-Madison (Math) [2]UW-Madison (ME) *chennan@math.wisc.edu

Consider a spatiotemporal discretization of a general complex turbulent system (e.g., PDEs). In many cases, only the measurements of part of the state variables are available (e.g., sparse obs):

$$\begin{aligned} \mathbf{u}_1(n+1) &= \mathcal{G}_1(\mathbf{u}_1(n), \mathbf{u}_2(n)), \\ \mathbf{u}_2(n+1) &= \mathcal{G}_2(\mathbf{u}_1(n), \mathbf{u}_2(n)), \end{aligned} \quad \xrightarrow{\text{CGKN}} \quad \begin{aligned} \mathbf{u}_1(n+1) &= \mathbf{F}_1\big(\mathbf{u}_1(n)\big) + \mathbf{G}_1\big(\mathbf{u}_1(n)\big)\mathbf{v}(n) + \boldsymbol{\sigma}_1\boldsymbol{\epsilon}_1(n+1), \\ \mathbf{v}(n+1) &= \mathbf{F}_2\big(\mathbf{u}_1(n)\big) + \mathbf{G}_2\big(\mathbf{u}_1(n)\big)\mathbf{v}(n) + \boldsymbol{\sigma}_2\boldsymbol{\epsilon}_2(n+1), \end{aligned}$$

where $\mathbf{v}(n) = \boldsymbol{\varphi}(\mathbf{u}_2(n))$, while $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{G}_1$, and $\mathbf{F}_2$ contain neural operators.

The CGKN transforms general nonlinear systems into highly nonlinear neural differential equations with conditional Gaussian structures (Chen et. al., *JCP* 2024, *CMAME* 2025).

- **Physics + Machine Learning.** The mathematical justification for the conditional Gaussian nonlinear stochastic systems has been demonstrated in (Chen & Majda 2018). Many well-known complex dynamical systems belong to this framework (Lorenz, Boussinesq, etc).
- **Data Assimilation.** It aims to retain essential nonlinear components while applying systematic simplifications to facilitate the development of analytic formulae for nonlinear DA. This allows for incorporating the DA performance into the deep learning training.
- **Uncertainty Quantification.** Even though the distribution in the latent space is conditional Gaussian, when mapping back to physical space via the decoder, the posterior distribution can be highly non-Gaussian.

**Test Example:**

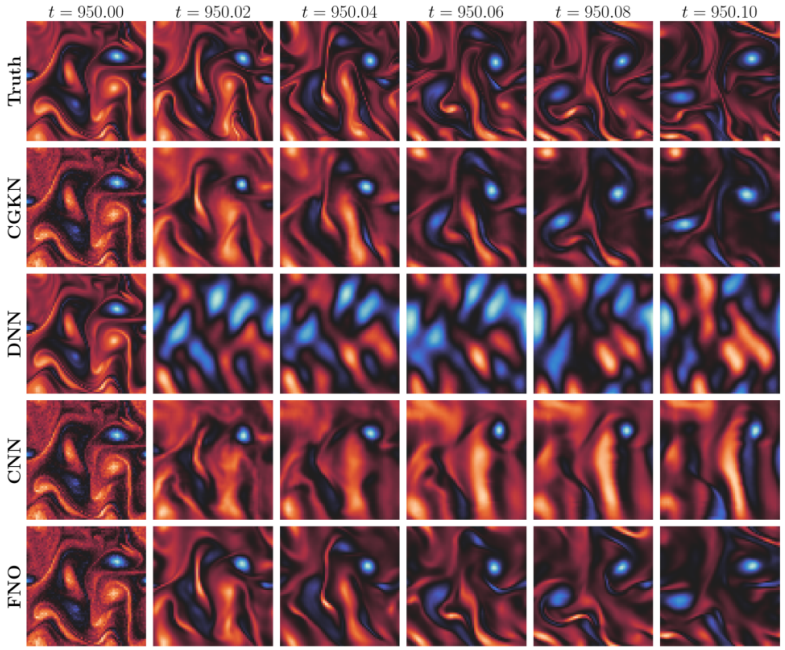
2D Navier-Stokes Equation

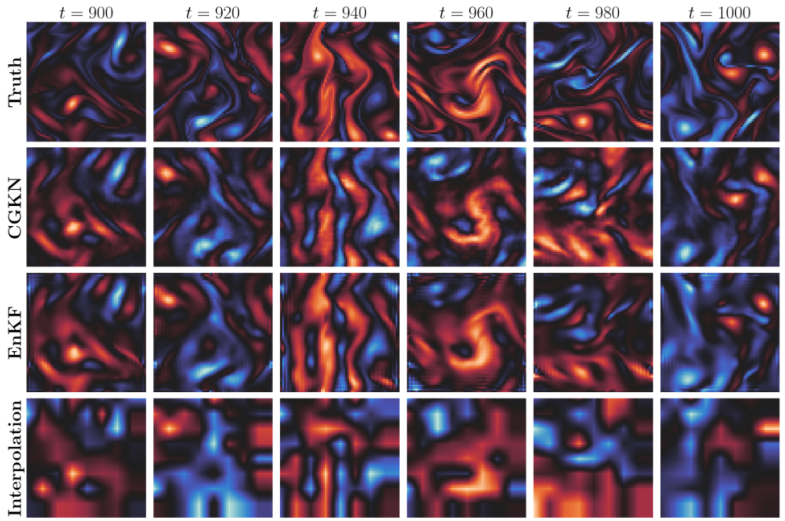
Resolution: $128 \times 128$

Observation: $8 \times 8$

| Methods \ Examples | Navier–Stokes Equations | |
|---|---|---|
| | DA Error | Forecast Error |
| CGKN | 6.0940e+01 | 1.9754e+01 |
| EnKF | 6.9010e+01 | — |
| Interpolation | 1.2844e+02 | — |
| DNN | — | 1.0936e+02 |
| CNN | — | 3.0600e+01 |
| FNO | — | 1.7129e+01 |

(All NNs have similar numbers of hyperparameters.)




Right top: Forecast

Right bottom: Data assimilation

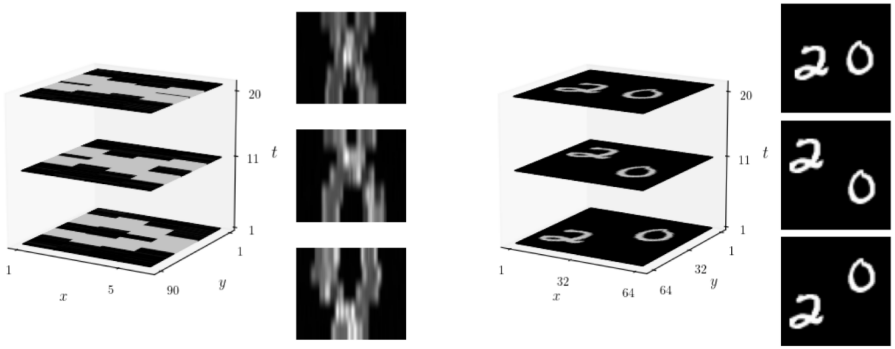
Computational efficiency compared to EnKF:

- 600x faster for Viscus Burgers Equation
- 125x faster for Kuramoto-Sivashinsky Equation
- 300x faster for Navier-Stokes Equation

CGKN has been adopted to recover the ocean field and topography (sea mountains), modeling equatorial Pacific Earth systems, conduct nonlinear Lagrangian data assimilation, incorporate memory in the system (Mori-Zwanzig formalism). An online model correction algorithm has also been developed (Chen, et. al., 2023, *Chaos*; Wang, et. al., 2025).

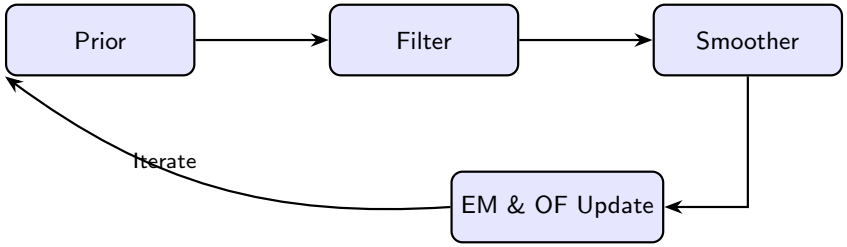
# Motivation & Problem Setup

- **Dynamic X-ray CT** seeks to reconstruct a sequence of images over time from limited projection data.
- **Digital twins** rely on accurate and efficient data assimilation to monitor and predict complex dynamical systems, making scalable sequential reconstruction methods essential for real-time applications.
- Some problems are **ill-posed**, **high-dimensional**, and subject to **radiation exposure limits:**
  - Small measurement noise can cause large reconstruction errors.
  - Only a limited number of projections can be collected safely.
  - Traditional "all-at-once" methods scale poorly with time and memory.
- **Goal:** Develop a *scalable, adaptive framework* that processes data sequentially and doesn't require major parameter tuning, memory, or time.



# Sequential Filtering Framework

- Employs a **Kalman filtering approach** for sequential estimation in time-dependent imaging.
- Uses a **reduced-order model** for scalability.
- Core components:
  - **Kalman Filter & RTS Smoother** — dynamic estimation and temporal refinement
  - **Expectation–Maximization (EM)** — learns noise statistics automatically
  - **Optical Flow (OF)** — estimates and refines motion between frames



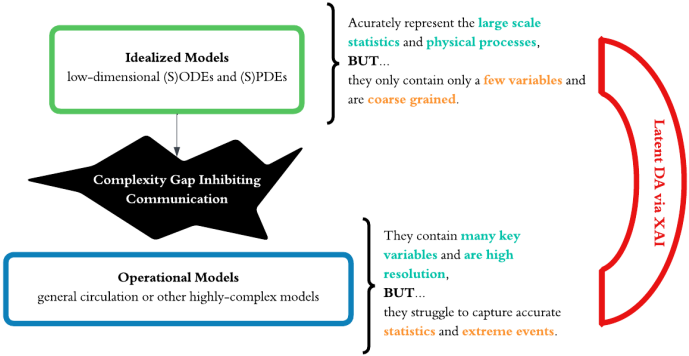

Prior → Filter → Smoother → EM&OF Update

1 2

* crmoser2@wisc.edu

# Bridging Idealized and Operational Models:
## An Explainable AI Framework for Earth System Emulators

Pouria Behnoudfar[1], **Charlotte Moser**[1*], Marc Bocquet[2], Sibo Cheng[2], Nan Chen[1]




### Results of implementation

The framework produces a **bridging model** that can efficiently generate numerous high-quality synthetic datasets that **mimic nature**, facilitate the study of **extreme events**, and **provide training datasets** for many machine learning tasks, including building effective **digital twins**.

- The framework is highly efficient, featuring a **reconfigured latent data assimilation** technique specifically designed to deal with large dimension discrepancies between **idealized** and **operational** models.
- The resulting **computationally efficient bridging model** is high resolution like the **operational model** but is dynamically and statistically improved through assimilation with pseudo observations of the **idealized model**.
- The process is explainable, allowing us to trace how specific features from the **idealized model** correct specific biases in the **operational model**.

# Implementation for Representing the Dominant Interannual Phenomena: ENSO

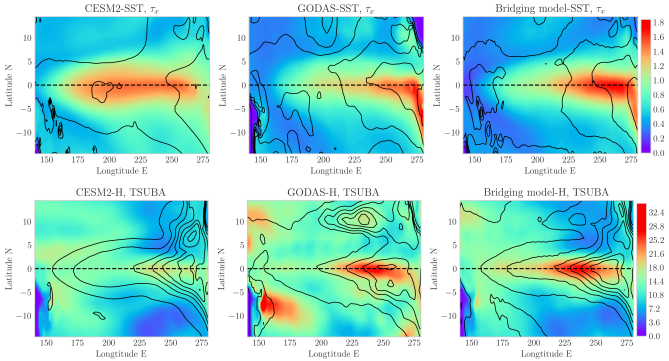



Figure: Standard deviation of different physical fields.

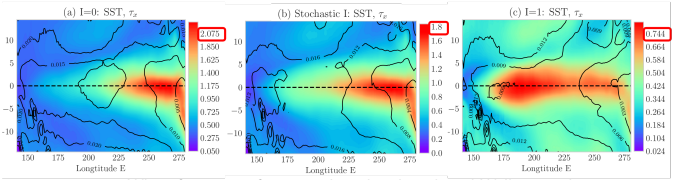



Figure: What-if scenarios for strengthened and weakened Walker circulation.

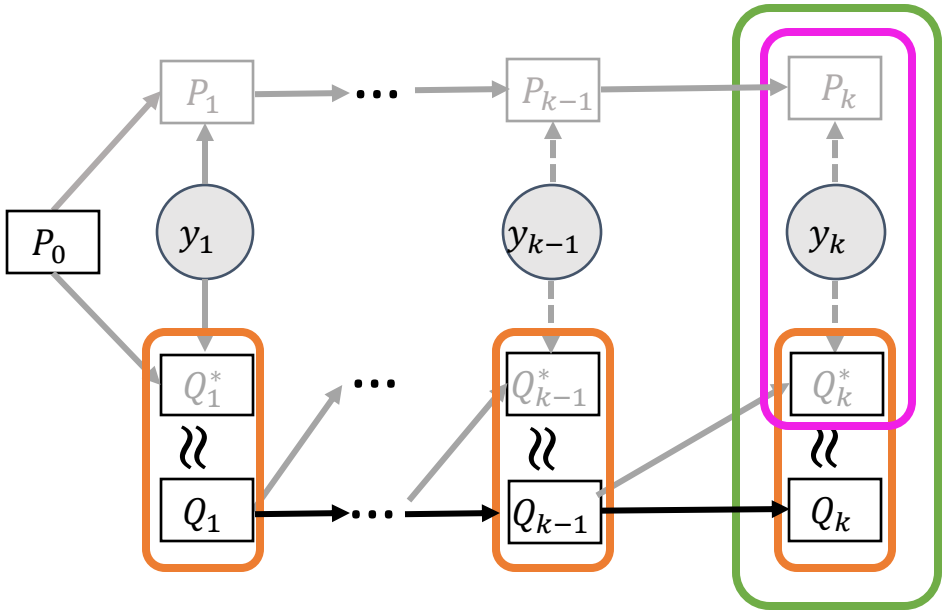
- The framework corrects both on equator and off equator biases in the **operational model** for both **pseudo-observed** and unobserved variables.
- By manipulating the long term background forcing in the **idealized model** the **bridging model** can effectively produce high resolution representations of different scenarios that mimic observed decadal variability.
- The **bridging model** provides a new way to build physics-assisted digital twins, which can efficiently test what-if scenarios.

# Towards understanding the errors in online Bayesian data assimilation

Liliang Wang and Alex Gorodetsky, University of Michigan

## Background & Motivation

- Many real-world tasks require assimilating data incrementally for real-time prediction and decision making :
  - Robotics navigation
  - Dynamic soaring of UAVs
- Practical Online Bayesian data assimilation methods are based on approximations: true posterior $P_k$ is approximated with $Q_k$
- Gap between analysis theory development and methodology development
  - The majority of the existing theoretical work are:
    - Asymptotic results
    - Developed for a specific method
  - ***Non-asymptotic*** analysis theory for ***general*** online Bayesian data assimilation methods is needed




: prior-to-posterior map

learning error $d(P_k, Q_k)$ comes from two sources:

Incremental approximation error $d(Q_k^*, Q_k)$

prior perturbation error $d(P_k, Q_k^*)$

$d(Q_k^*, Q_k)$: fairly studied $d(Q_k^*, Q_k)$: ***not well studied***

# Main Results [1]

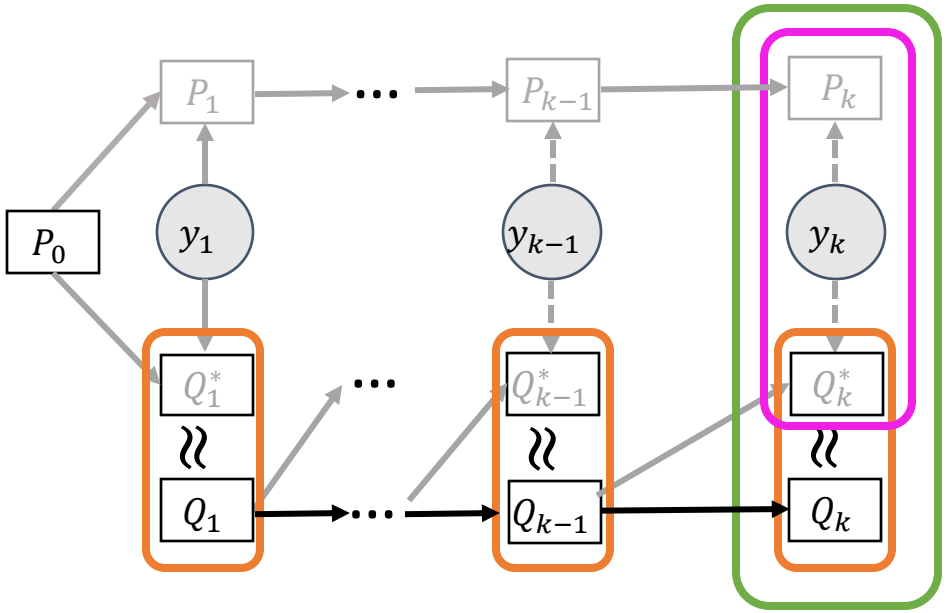
- The prior perturbation error $d(P_k, Q_k^*)$ is caused by the prior error $d(P_{k-1}, Q_{k-1})$. What is the relation between the two?

***Pointwise global Lipschitz prior-to-posterior stability*** :
$\forall\, P_{k-1}$, we have

$$d(P_k, Q_k^*) \leq K(y_k, P_{k-1})\, d(P_{k-1}, Q_{k-1}), \quad \forall\, Q_{k-1} \text{ (global)}$$

Holds for $d$ being
- total variation distance
- Hellinger distance
- Wasserstein-1 distance

Holds in three contexts
- inverse problems
- state estimation
- joint state-parameter estimation




: prior-to-posterior map

- Two ***upper bounds*** on the learning error $d(P_k, Q_k)$:

$$d(P_k, Q_k) \leq \sum_{j=1}^{k-1} C_1(\mathcal{Y}_k, P_{j:k-1})\, d(Q_j^*, Q_j) + d(Q_k^*, Q_k)$$

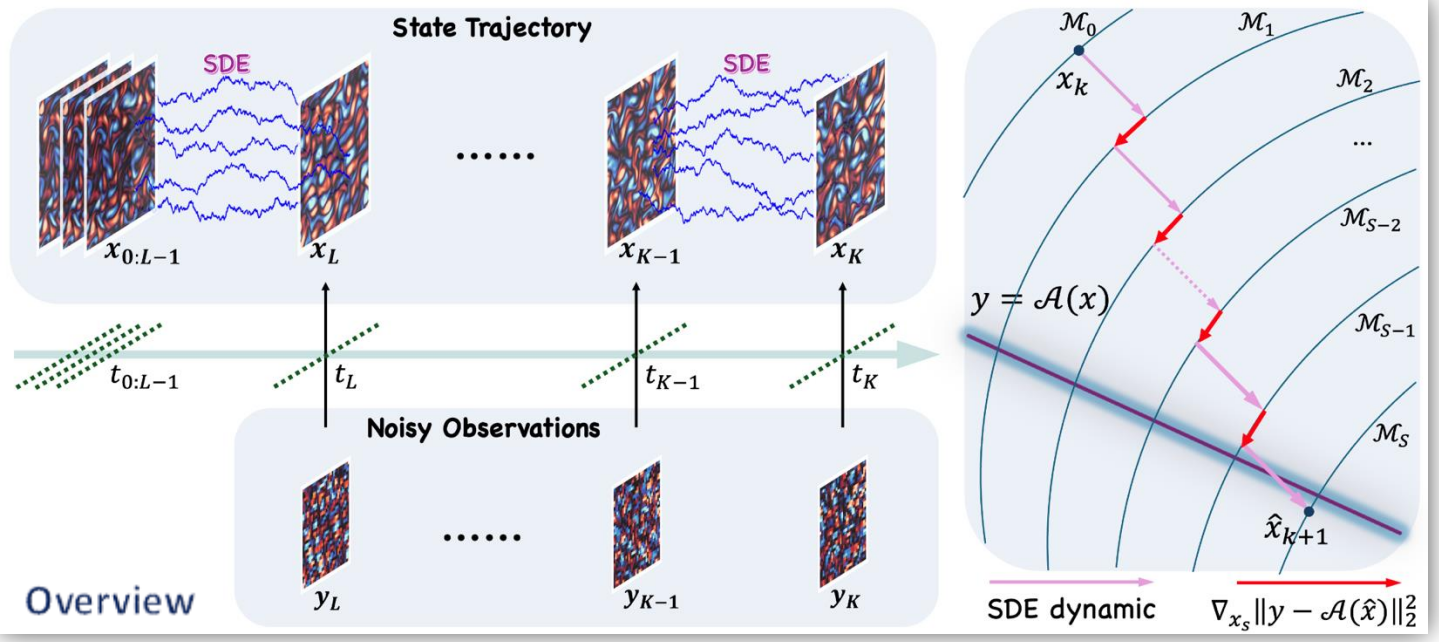
$$d(P_k, Q_k) \leq \sum_{j=1}^{k-1} C_2(\mathcal{Y}_k, Q_{j:k-1})\, d(Q_j^*, Q_j) + d(Q_k^*, Q_k)$$

- Sufficient conditions for ***learning error decay,*** i.e.,

$$d(P_k, Q_k) \leq d(P_{k-1}, Q_{k-1})$$

[1] Wang, L. and Gorodetsky, A. A. (2025). A global Lipschitz stability perspective for understanding approximate approaches in Bayesian sequential learning. arXiv preprint arXiv:2507.20379

# FlowDAS: A Stochastic Interpolants-based Framework for Data Assimilation




$$\tilde{I}_s^k = \alpha_s \boldsymbol{x}_k + \beta_s \boldsymbol{x}_{k+1} + \sqrt{s}\sigma_s \boldsymbol{z}_k$$
$$\boldsymbol{R}_s^k = \dot{\alpha}_s \boldsymbol{x}_k + \dot{\beta}_s \boldsymbol{x}_{k+1} + \sqrt{s}\dot{\sigma}_s \boldsymbol{z}_k,$$

$$\mathcal{L}_b^{\text{emp}}(\hat{\boldsymbol{b}}) = \frac{1}{K'} \sum_{k \in \mathcal{B}_{K'}} \int_0^1 \ell(\hat{\boldsymbol{b}}_s(\tilde{I}_s^k, \boldsymbol{x}_k), \boldsymbol{R}_s^k)\, \mathrm{d}s,$$

**Algorithm 1** Training

1: **Input:** Dataset $\boldsymbol{x}_{0:K}$; minibatch size $K' \leq K$; coefficients $\alpha_s$, $\beta_s$, $\sigma_s$
2: **repeat**
3: Compute the empirical loss $\mathcal{L}_b^K[\hat{\boldsymbol{b}}]$ in Eq. 14
4: Take the gradient step on $\mathcal{L}_b^K[\hat{\boldsymbol{b}}]$ to update $\hat{\boldsymbol{b}}_s$
5: **until** converged
6: **return** drifts $\hat{\boldsymbol{b}}_s$

**Algorithm 2** Sampling

1: **Input:** Observation $\boldsymbol{y}_{1:K}$, the measurement map $\mathcal{A}$, initial state $\boldsymbol{x}_0$, model $\hat{\boldsymbol{b}}_s(\boldsymbol{X}, \boldsymbol{X}_0)$, noise coefficient $\sigma_s$, grid $s_0 = 0 < s_1 < \cdots < s_N = 1$, i.i.d. $\boldsymbol{z}_n \sim \mathcal{N}(0, I_D)$ for $n = 0 : N-1$, step size $\zeta_n$, Monte Carlo sampling times $J$
2: Set $\hat{\boldsymbol{x}}_0 \leftarrow \boldsymbol{x}_0$
3: Set the $(\Delta s)_n = s_{n+1} - s_n, n = 0 : N-1$
4: **for** $k = 0$ **to** $K-1$ **do**
5: $\boldsymbol{X}_{s_0}, \boldsymbol{y} \leftarrow \hat{\boldsymbol{x}}_k, \boldsymbol{y}_{k+1}$
6: **for** $n = 0$ **to** $N-1$ **do**
7: $\boldsymbol{X}'_{s_{n+1}} = \boldsymbol{X}_{s_n} + \hat{\boldsymbol{b}}_s(\boldsymbol{X}_{s_n}, \boldsymbol{X}_{s_0})(\Delta s)_n + \sigma_{s_n}\sqrt{(\Delta s)_n}\boldsymbol{z}_n$
8: $\{\hat{\boldsymbol{X}}_1^{(j)}\}_{j=1}^J \leftarrow$ Posterior estimation $(\hat{\boldsymbol{b}}_s, s_n, \boldsymbol{X}_0, \boldsymbol{X}_{s_n})$
9: $\{w_j\}_{j=1}^J \leftarrow$ Softmax function$(\{\left\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{X}}_1^{(j)})\right\|_2^2\}_{j=1}^J)$
10: $\boldsymbol{X}_{s_{n+1}} = \boldsymbol{X}'_{s_{n+1}} - \zeta_n \nabla_{\boldsymbol{X}_{s_n}} \sum_{j=1}^J w_j \left\|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{X}}_1^{(j)})\right\|_2^2$
11: **end for**
12: $\hat{\boldsymbol{x}}_{k+1} \leftarrow \boldsymbol{X}_{s_N}$
13: **end for**
14: **return** $\{\hat{\boldsymbol{x}}_k\}_{k=1}^K$

Algorithm

Paper

GitHub

- Consider a stochastic process $\boldsymbol{X}_s$ defined over the interval $s \in [0, 1]$.
  - Initial state: $\boldsymbol{X}_0 \sim \pi(\boldsymbol{X}_0)$
  - Final state: $\boldsymbol{X}_1 \sim q(\boldsymbol{X}_1 | \boldsymbol{X}_0)$
- A *Stochastic Interpolant* can be described as:
  $$I_s = \alpha_s \boldsymbol{X}_0 + \beta_s \boldsymbol{X}_1 + \sigma_s \boldsymbol{W}_s,$$
  - $\boldsymbol{W}_s$ is a Wiener process. The time-varying coefficients are defined to satisfy with boundary conditions: $\alpha_s = 1 - s$, $\beta_s = s^2$, $\sigma_s = 1 - s$.
- The velocity of the interpolant path, $\boldsymbol{R}_s$, is given by:
  $$\boldsymbol{R}_s = \dot{\alpha}_s \boldsymbol{X}_0 + \dot{\beta}_s \boldsymbol{X}_1 + \dot{\sigma}_s \boldsymbol{W}_s$$
- Consider the following SDE:
  $$\mathrm{d}\boldsymbol{X}_s = \boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{X}_0)\,\mathrm{d}s + \sigma_s\,\mathrm{d}\boldsymbol{W}_s$$
- The drift term $\boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{X}_0)$ is defined as the minimizer of the cost function:
  $$\mathcal{L}_b(\hat{\boldsymbol{b}}_s) = \int_0^1 \mathrm{E}\left[\left\|\hat{\boldsymbol{b}}_s(I_s, \boldsymbol{X}_0) - \boldsymbol{R}_s\right\|^2\right] \mathrm{d}s$$
- Then, one can prove that: $\mathrm{Law}(I_s | \boldsymbol{X}_0) = \mathrm{Law}(\boldsymbol{X}_s)$.

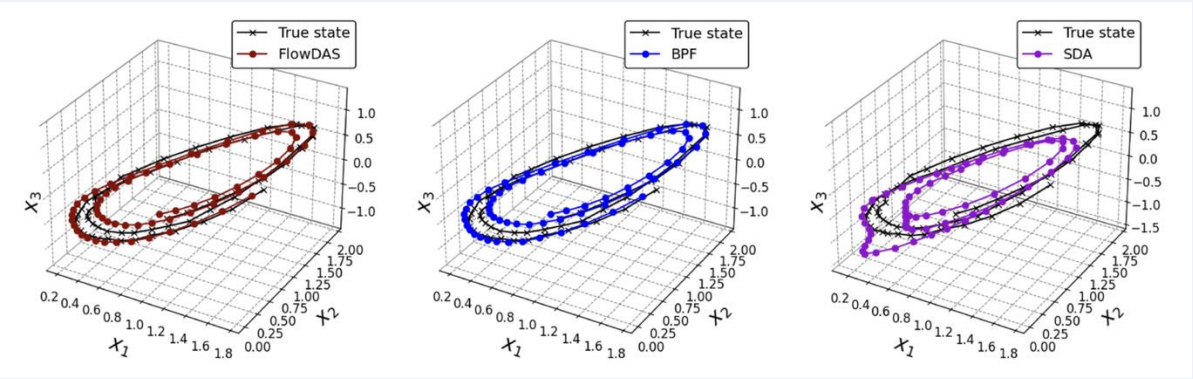
Predictor

- Derive the observation-aware drift function:
  $$\boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{y}, \boldsymbol{X}_0) = \boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{X}_0) + \frac{\nabla \log p(\boldsymbol{y} | \boldsymbol{X}_s, \boldsymbol{X}_0)}{\lambda_s \beta_s}$$
  $$= \boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{X}_0) + \frac{\sum_{j=1}^J \omega_j \nabla \log p(\boldsymbol{y} | \boldsymbol{X}_1^{(j)})}{\lambda_s \beta_s}$$
- **How to estimate the posterior likelihood $\nabla \log p(\boldsymbol{y} | \boldsymbol{X}_1^{(j)})$?**
  - **Acceleration** of $\boldsymbol{X}_1$ estimation:
    - First-order Euler-Milstein method:
      $$\hat{\boldsymbol{X}}_1 = \boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{X}_0)(1 - s) + \int \sigma_s\,\mathrm{d}\boldsymbol{W}_s$$
    - Second-order stochastic Runge-Kutta method:
      $$\hat{\boldsymbol{X}}_1{}' = \frac{\boldsymbol{b}_s(\boldsymbol{X}_s, \boldsymbol{X}_0) + \boldsymbol{b}_1(\hat{\boldsymbol{X}}_1, \boldsymbol{X}_0)}{2}(1 - s) + \int \sigma_s\,\mathrm{d}\boldsymbol{W}_s$$

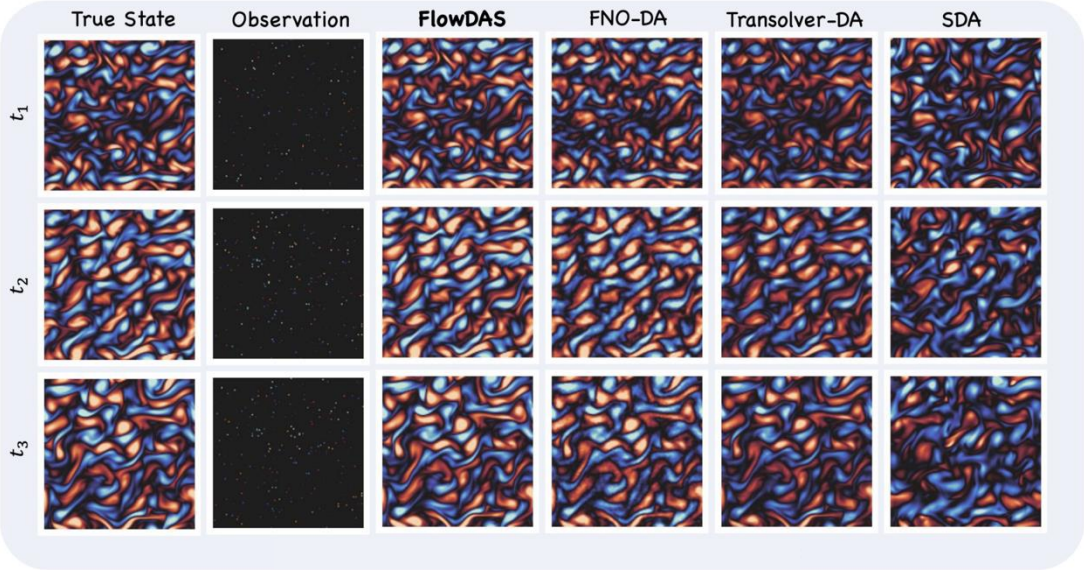
Corrector

**FlowDAS** has been accepted by the **NeurIPS 2025**!

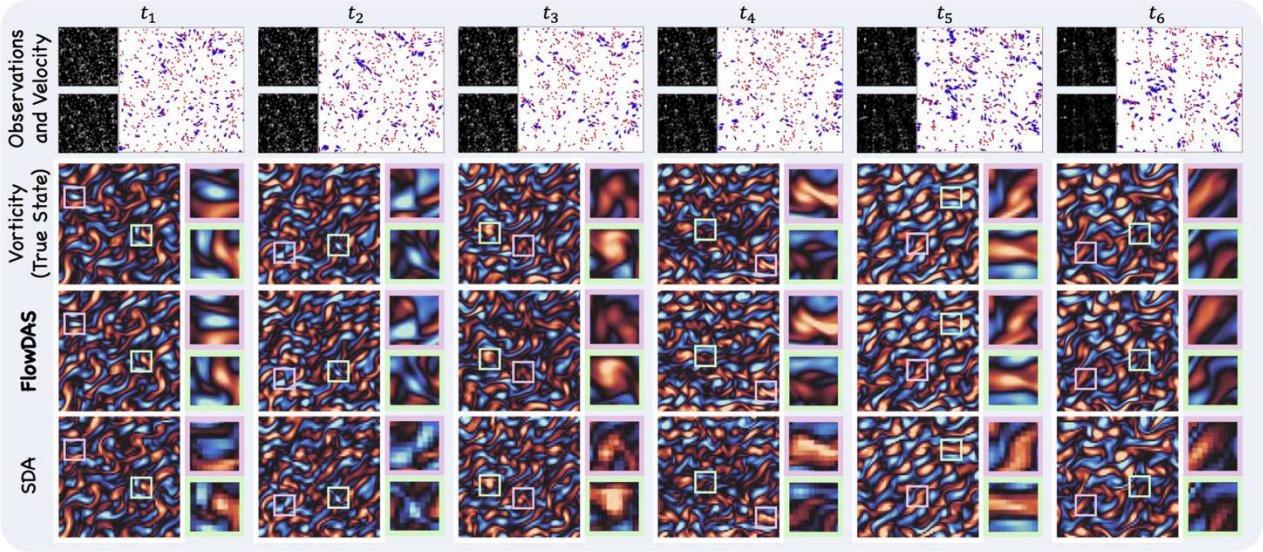
# FlowDAS: A Stochastic Interpolants-based Framework for Data Assimilation



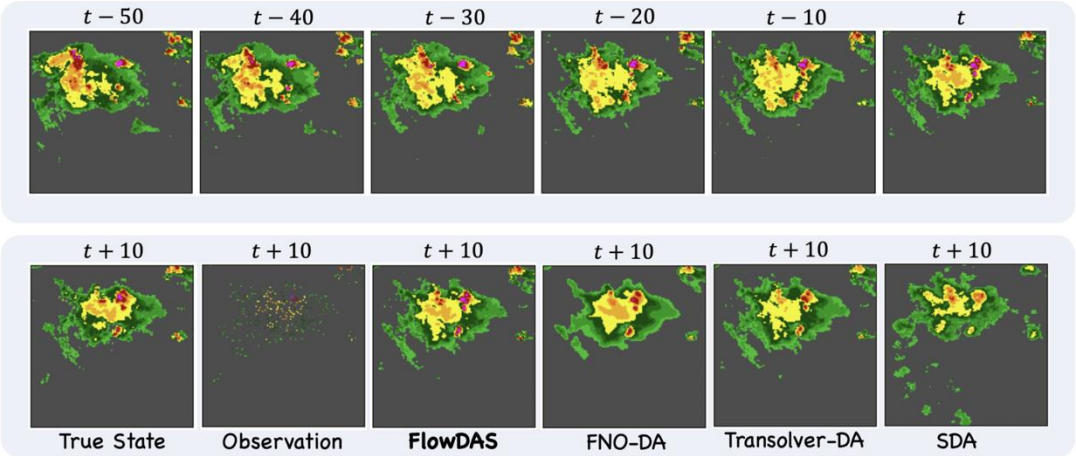

Experiments on **Lorenz-63** data assimilation task




Experiments on **Navier-Stokes** sparse observation task



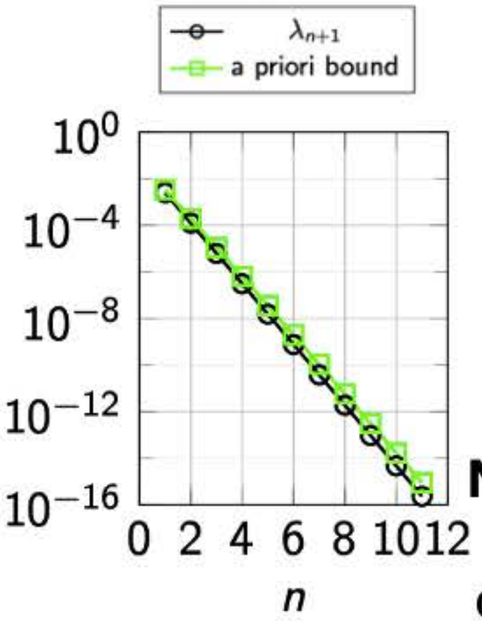

Experiments on **Particle Image Velocimetry**




Experiments on **SEVIR weather forecasting** task

# Probabilistic Error Analysis of a Randomized Proper Orthogonal Decomposition


- Kathrin Smetana[1], Tommaso Taddei[2], Marissa Whitby[1] and Zhiyu Yin[1]
- [1] Stevens Institute of Technology [2] Inria Bordeaux South-West


M = 80 samples

**Main goals:** To approximate the range of a bounded nonlinear function $u : P \to H$ using randomized POD and derive error bounds that do not rely on the dimension of the ambient space.

**Current Result:** We have a nonasymptotic probabilistic error bound of the covariance estimator used in randomized POD/PCA for bounded random variables.

**Current Direction:** We are currently using arguments from [Reiß, Wahl 20] to derive a priori error bounds for randomized POD/PCA for bounded random variables.

**Motivation:** For a more restrictive subclass of sub-Gaussian random variables, which excludes some bounded random variables, [Reiß, Wahl 20] shows if a covariance operator has exponentially decaying eigenvalues, then the number of samples needed for POD/PCA to produce (quasi-)optimal results scales linearly in the reduced space.

**Applications:**

- PCA of gradient of log-likelihood function for Bayesian Inverse Problems [Zahm, Cui, Law, Spantini, Marzouk 22]
- Construction of local multiscale ansatz spaces [Smetana, Taddei 23]

Supported by NSF Award # 2145364

# The Problem of High-Dimensional Summation

## Why Naive Summation Fails for Low-Rank Tucker Tensors

### The Problem: Rank Inflation

Summing low-rank Tucker tensors, $\mathcal{C} = \sum_{i=1}^{d} \mathcal{X}^{(i)}$, is computationally challenging.

- The core issue is **Rank Inflation**.
- The rank of the sum becomes the sum of individual ranks: $R_k \approx \sum_i r_k^{(i)}$.
- This causes exponential growth in storage and computational cost.

### The Conventional, Inefficient Solution

This leads to a costly process:

1. **Sum:** Ranks additively combine, creating a high-rank tensor.
2. **Round:** An expensive decomposition is required to compress the result.

**The question:** Can we find the final low-rank sum *without* forming the high-rank intermediates?

# Breaking the Cycle with Structured Sketching

## A Shortcut to the Sum

### Proposed Solution: Sketch, then Sum

Our approach bypasses the need for rounding.

- We create small, compressed **sketches** of each tensor's unfolding: $\mathbf{Y}_k^{(i)} = \mathbf{X}_{(k)}^{(i)} \mathbf{S}_k$.
- We sum these small sketches to find an approximate basis, $\{\mathbf{U}_k\}$, for the final sum.
- **Reconstruction:** We project each original core tensor onto this new basis and sum the results to form the final core $\tilde{\mathcal{G}}$.
- **The result:** An accurate, low-rank approximation $\tilde{\mathcal{C}} = \tilde{\mathcal{G}} \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N$ with none of the expensive detours.

### The Theoretical Core & Next Frontier

**The Key Insight:** Method efficacy hinges on choosing the sketch size. Our **Sub-rank Selection** heuristic provides a principled way to balance accuracy and cost.

**Interesting discussions:**

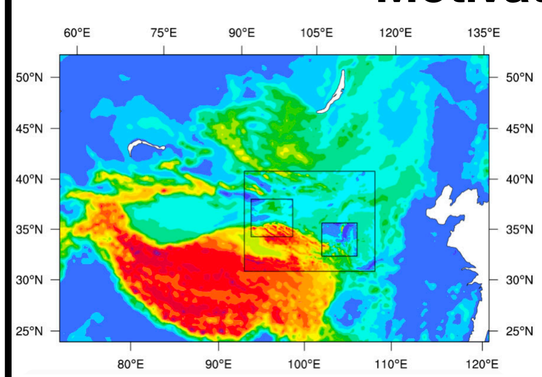
- How do we find the optimal sketch size?
- Can this process be made fully adaptive?
- What are the next steps for scalable tensor computations; a hierarchical Approach?

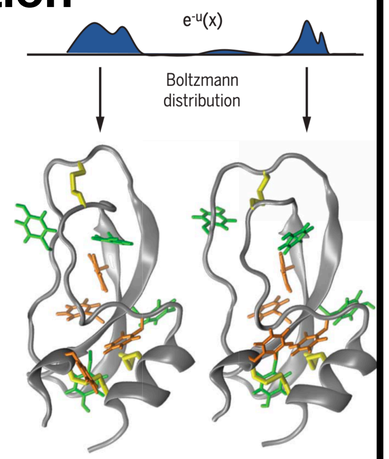
# Identification of Paths for Dynamic Measure Transport

Aimee Maurais, Bamdad Hosseini, and Youssef Marzouk

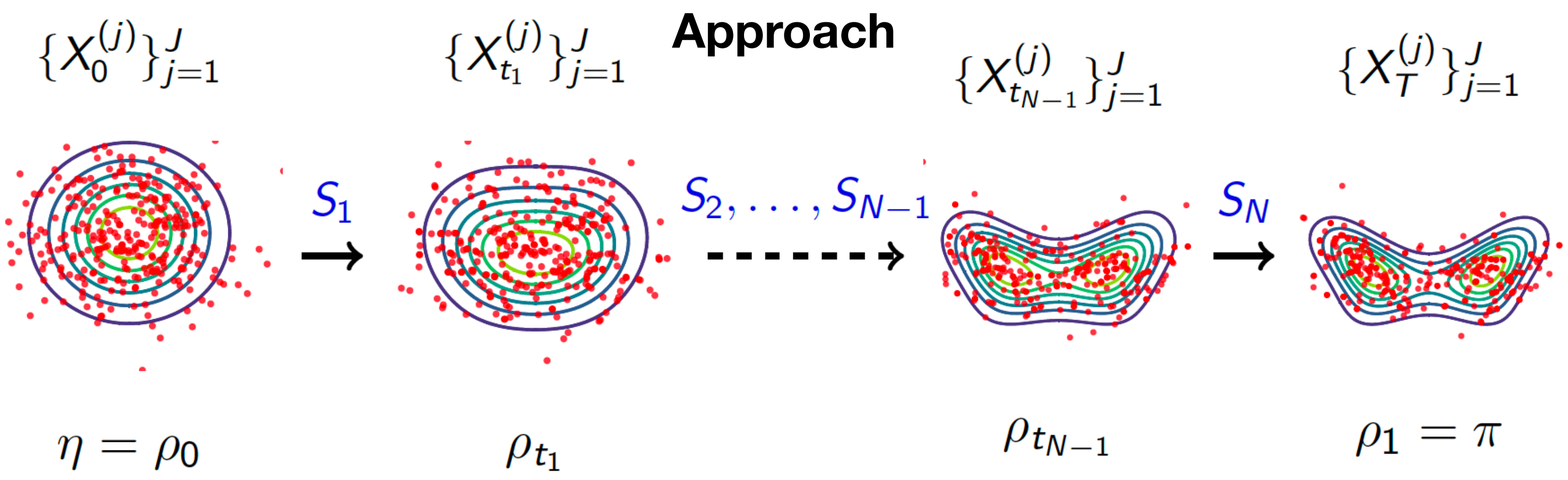
**Task: Sample from a distribution $\pi$ on $\mathbb{R}^d$**

## Motivation

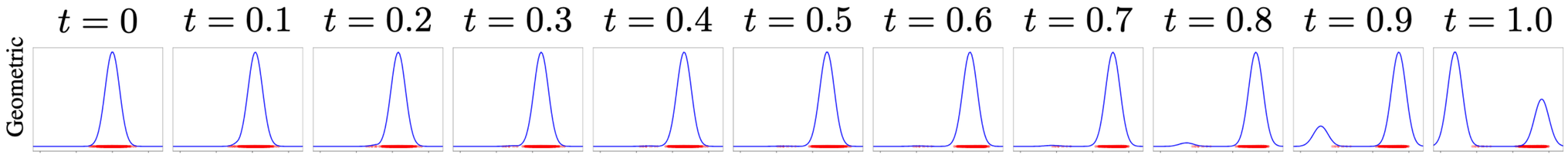



## Approach




**The commonly used geometric mixture $\rho(\,\cdot\,, t) \propto \eta^{1-t}\pi^t$ may be problematic!**



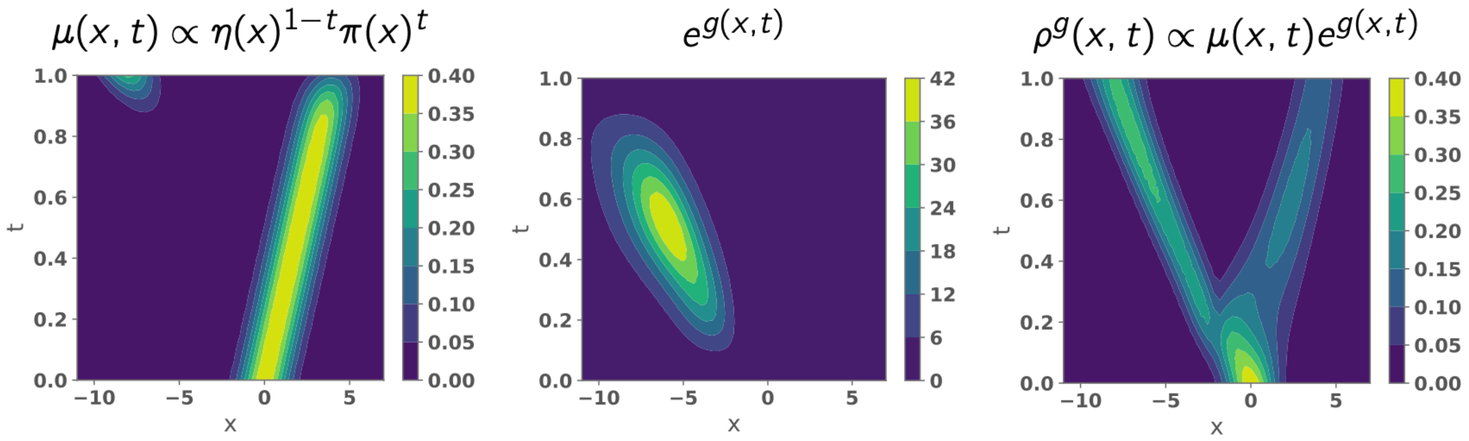

**Can we identify a path $\rho$ for which a corresponding velocity field is "nice"?**

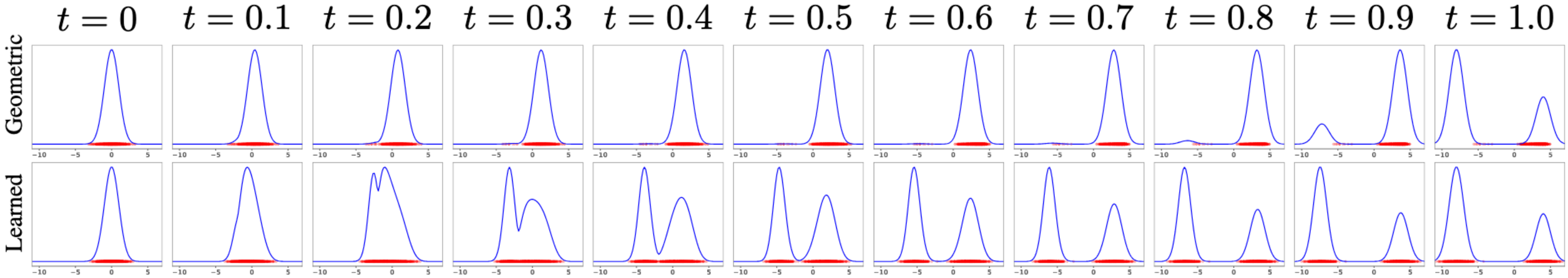
# Approach: Identify a *tilting* via control

**We seek a *new* path $\rho^g \propto \rho_{\text{ref}} e^g$, where $\rho_{\text{ref}}$ is a reference path, by solving**

$$\inf_{v \in \mathscr{V}, g \in \mathscr{G}} \|v\|_{\mathscr{V}}^2 + \lambda_g \|g\|_{\mathscr{G}}^2 \quad \text{s.t.} \quad -\nabla \cdot (v\rho^g) = \rho^g(\partial_t \log \rho^g), \quad \rho^g \propto \rho^{\text{ref}} e^g, \quad g(\cdot,0) = g(\cdot,1) \equiv 0$$

- **Formalizes previous approaches to fix the geometric mixture**
- **Flexible framework enables promotion of *smoothness***
- **Many implementations possible!**



# Bayesian Inference for Latent Gaussian Models Governed by PDEs

Sonia Reilly and Georg Stadler
Courant Institute, NYU

Hierarchical model with Gaussian prior:

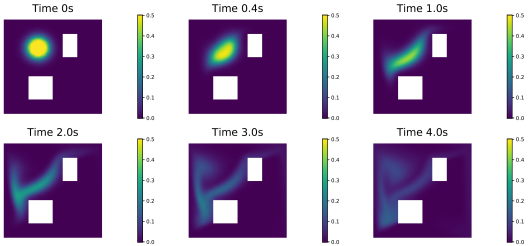
$$\boldsymbol{\theta} \sim \pi_{\text{hyp}}(\boldsymbol{\theta})$$
$$\boldsymbol{m}|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{pr}}(\boldsymbol{\theta}), \boldsymbol{Q}_{\text{pr}}^{-1}(\boldsymbol{\theta}))$$
$$\boldsymbol{y}|\boldsymbol{m}, \boldsymbol{\theta} \sim \pi_{\text{like}}(\boldsymbol{y}|\boldsymbol{m}, \boldsymbol{\theta})$$

Linear Gaussian Bayesian inverse problem as LGM:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{m} + \boldsymbol{\varepsilon} \quad \text{with} \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{Q}_{\varepsilon}(\boldsymbol{\theta})),$$

with $\boldsymbol{A}$ a discretization of a linear PDE.
Want to characterize $\pi(\boldsymbol{m}|\boldsymbol{y})$.

E.g. Advection-Diffusion




Given later time observations $\boldsymbol{y}$ and prior with unknown hyperparameters $\boldsymbol{\theta}$, find posterior of initial condition $\boldsymbol{m}$

# Fast Sampling Using Hyperparameter Marginal $\pi(\boldsymbol{\theta}|\boldsymbol{y})$

Sample $\boldsymbol{m}^* \sim \pi(\boldsymbol{m}|\boldsymbol{y})$ by

1. sampling $\boldsymbol{\theta}^* \sim \pi(\boldsymbol{\theta}|\boldsymbol{y})$ (low-dimensional, so can use MCMC)
2. sampling $\boldsymbol{m}^* \sim \pi(\boldsymbol{m}|\boldsymbol{\theta}^*, \boldsymbol{y})$ (Gaussian, since it is the posterior of a linear Gaussian Bayesian inverse problem)

Need to compute quickly:

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{\pi(\boldsymbol{m}, \boldsymbol{\theta}, \boldsymbol{y})}{\pi(\boldsymbol{m}|\boldsymbol{\theta}, \boldsymbol{y})} = \frac{\pi_{\text{like}}(\boldsymbol{y}|\boldsymbol{m}, \boldsymbol{\theta})\pi_{\text{pr}}(\boldsymbol{m}|\boldsymbol{\theta})\pi_{\text{hyp}}(\boldsymbol{\theta})}{\pi(\boldsymbol{m}|\boldsymbol{\theta}, \boldsymbol{y})}$$

$$\propto \left(\frac{|\boldsymbol{Q}_{\text{pr}}||\boldsymbol{Q}_{\varepsilon}|}{|\boldsymbol{Q}_{\text{post}}|}\right)^{1/2} \exp\left(-\frac{1}{2}\left[||\boldsymbol{y}||_{\boldsymbol{Q}_{\varepsilon}} + ||\boldsymbol{\mu}_{\text{pr}}||_{\boldsymbol{Q}_{\text{pr}}} - ||\boldsymbol{\mu}_{\text{post}}||_{\boldsymbol{Q}_{\text{post}}}\right]\right)\pi_{\text{hyp}}(\boldsymbol{\theta})$$

Idea: low rank approximation of $\boldsymbol{Q}_{\text{post}} - \boldsymbol{Q}_{\text{pr}} = \boldsymbol{A}^T\boldsymbol{Q}_{\varepsilon}\boldsymbol{A}$ (two ways)