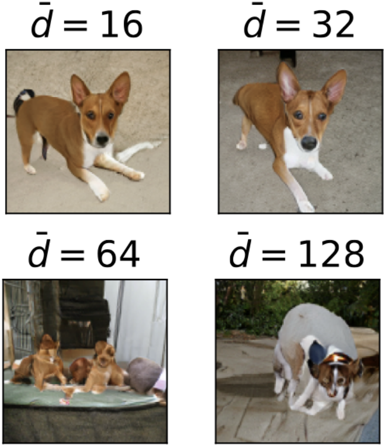
Toward **Physical** Generative Models

Nisha Chandramoorthy

U Chicago
Collaborators: Jeongjin Park (Georgia Tech), Nicole Yang (Emory), Adriaan de Clercq (U Chicago)

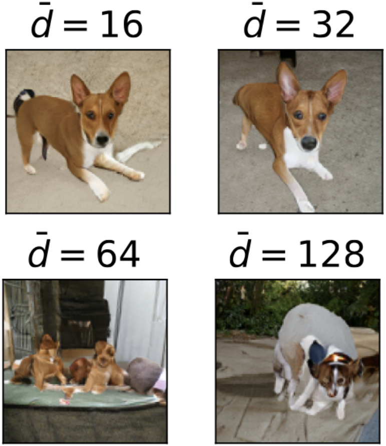
June 12, 2025

Singular measures with densities on an unknown data manifold

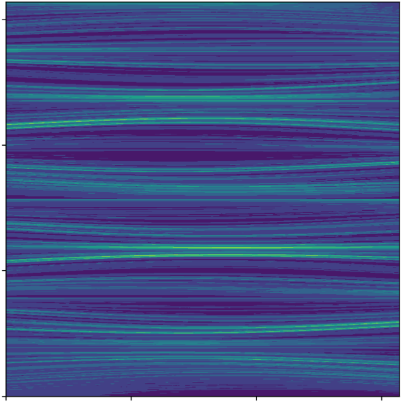


[**Pope et al 2021**]

Singular measures with densities on an unknown data manifold



[**Pope et al 2021**]



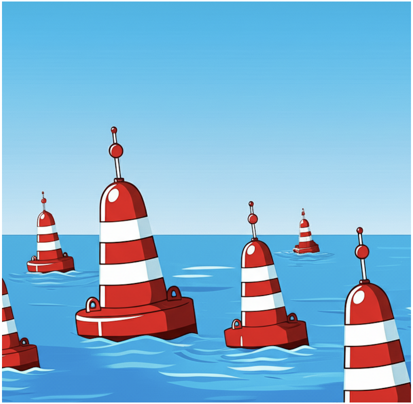
measure continuous on unstable manifold – 1D roughly horizontal curves. [**C and Wang, 2022**]

Downstream task: Bayesian computation (inverse problems/data assimilation)

Hidden orbit: $x, F(x), F^2(x), \cdots ,$

Downstream task: Bayesian computation (inverse problems/data assimilation)

Hidden orbit: $x, F(x), F^2(x), \cdots,$



Target: X_t or parameters of F given observations

Three ways to sample

Generative modeling: when are generative models robust to learning errors?

C and de Clercq, 2025

Learning dynamics: learning statistically accurate chaotic timeseries from data

Park, Yang and C, NeuRIPS 2024

Three ways to sample

Generative modeling: when are generative models robust to learning errors?

C and de Clercq, 2025

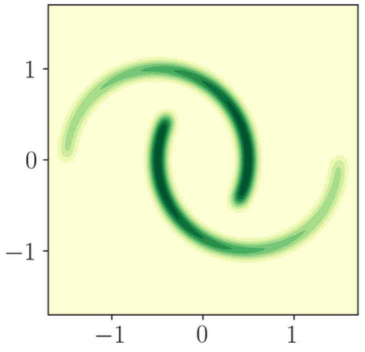
Learning dynamics: learning statistically accurate chaotic timeseries from data

Park, Yang and C, NeuRIPS 2024

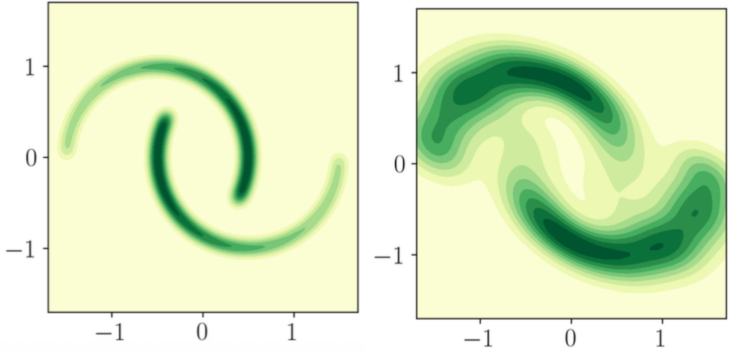
Learning scores + sampling: any dynamical measure transport

C, Schäfer and Marzouk, AISTATS 2024

Robustness of the support: generating from the *data manifold*

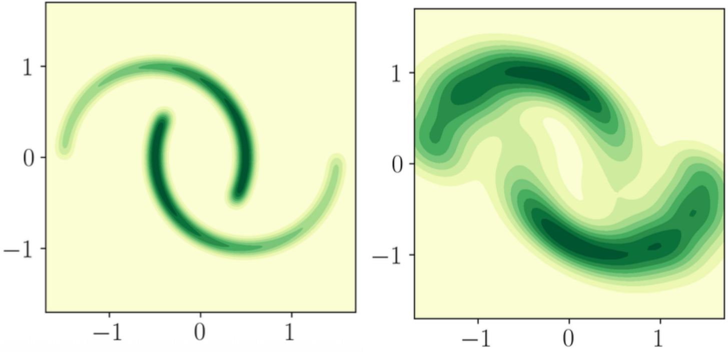


Robustness of the support: generating from the *data manifold*



- Formalize sampling from the data manifold?

Robustness of the support: generating from the *data manifold*



- Formalize sampling from the data manifold?
- Distinguish GMs based on robustness of the predicted support?

Dynamical formulation of generative models

- Given samples $x_1, \cdots, x_m \sim p_{\mathrm{data}}$, $x_i \in \mathbb{R}^D$, generate more samples from p_{data}.

Dynamical formulation of generative models

- Given samples $x_1, \cdots, x_m \sim p_{\mathrm{data}}$, $x_i \in \mathbb{R}^D$, generate more samples from p_{data}.
- Learned vector field: $x \to v_t(x) \in T_x\mathbb{R}^D$

Dynamical formulation of generative models

- Given samples $x_1, \cdots, x_m \sim p_{\mathrm{data}}$, $x_i \in \mathbb{R}^D$, generate more samples from p_{data}.
- Learned vector field: $x \to v_t(x) \in T_x\mathbb{R}^D$
- In score-based diffusion [**Anderson 1982; De Bortoli et al, Song and Ermon 2019, Song et al 2020, Sohl-Dickstein et al 2015**], $v_t(x) \equiv s(x, t)$, scores of densities of a noising process initialized with $x_i \sim p_{\mathrm{data}}$.

Dynamical formulation of generative models

- Given samples $x_1, \cdots, x_m \sim p_{\mathrm{data}}$, $x_i \in \mathbb{R}^D$, generate more samples from p_{data}.
- Learned vector field: $x \to v_t(x) \in T_x\mathbb{R}^D$
- In score-based diffusion [**Anderson 1982; De Bortoli et al, Song and Ermon 2019, Song et al 2020, Sohl-Dickstein et al 2015**], $v_t(x) \equiv s(x, t)$, scores of densities of a noising process initialized with $x_i \sim p_{\mathrm{data}}$.
- In conditional flow matching variants, stochastic interpolant variants [**Liu 2022, Lipman et al 2023, Tong et al 2023, Albergo et al 2023**], flow of $v_t(x)$ transports probability densities from p_0 (easy) to p_{data} (target)

Dynamical formulation of generative models

- Given samples $x_1, \cdots, x_m \sim p_{\mathrm{data}}$, $x_i \in \mathbb{R}^D$, generate more samples from p_{data}.
- Learned vector field: $x \to v_t(x) \in T_x\mathbb{R}^D$
- In score-based diffusion [**Anderson 1982; De Bortoli et al, Song and Ermon 2019, Song et al 2020, Sohl-Dickstein et al 2015**], $v_t(x) \equiv s(x, t)$, scores of densities of a noising process initialized with $x_i \sim p_{\mathrm{data}}$.
- In conditional flow matching variants, stochastic interpolant variants [**Liu 2022, Lipman et al 2023, Tong et al 2023, Albergo et al 2023**], flow of $v_t(x)$ transports probability densities from p_0 (easy) to p_{data} (target)
- $\ell(\theta) = \mathbb{E}_{t, X_t \sim p_t} \| v_{\theta,t}(X_t) - v_t(X_t) \|^2$,

Dynamical formulation of generative models

- Given samples $x_1, \cdots, x_m \sim p_{\mathrm{data}}$, $x_i \in \mathbb{R}^D$, generate more samples from p_{data}.
- Learned vector field: $x \to v_t(x) \in T_x\mathbb{R}^D$
- In score-based diffusion [**Anderson 1982; De Bortoli et al, Song and Ermon 2019, Song et al 2020, Sohl-Dickstein et al 2015**], $v_t(x) \equiv s(x, t)$, scores of densities of a noising process initialized with $x_i \sim p_{\mathrm{data}}$.
- In conditional flow matching variants, stochastic interpolant variants [**Liu 2022, Lipman et al 2023, Tong et al 2023, Albergo et al 2023**], flow of $v_t(x)$ transports probability densities from p_0 (easy) to p_{data} (target)
- $\ell(\theta) = \mathbb{E}_{t, X_t \sim p_t} \| v_{\theta,t}(X_t) - v_t(X_t) \|^2$,
- $\partial p_t / \partial t = -\mathrm{div}(v_t\, p_t)$, with $p_\tau \equiv p_{\mathrm{data}}$.

Errors in dynamical generative models

Score based generative model [**Song and Ermon, 2021**]

Errors in dynamical generative models

Score based generative model [**Song and Ermon, 2021**]

How do regression errors ($\ell(\theta)$) propagate along v_t?

Errors in dynamical generative models

Score based generative model [**Song and Ermon, 2021**]

How do regression errors ($\ell(\theta)$) propagate along v_t?

Are some generative models more robust to errors?

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.
- $F^{t,W} = F^W_{t-1} \circ \cdots \circ F^W_0$ is a random dynamical system, where at each time t, we choose $F^W_t \sim \pi_t$.

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.
- $F^{t,W} = F^W_{t-1} \circ \cdots \circ F^W_0$ is a random dynamical system, where at each time t, we choose $F^W_t \sim \pi_t$.
- Discrete-time RDS: $W = \{W_t\}$ iid standard normal RVs.

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.
- $F^{t,W} = F^W_{t-1} \circ \cdots \circ F^W_0$ is a random dynamical system, where at each time t, we choose $F^W_t \sim \pi_t$.
- Discrete-time RDS: $W = \{W_t\}$ iid standard normal RVs.
- For SGM: $F^W_t(x) = x + (\delta t) s(x, \tau - t) + \sqrt{\delta t}\, W_t$, $W_t \sim \mathcal{N}(0, \mathrm{Id})$.

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.
- $F^{t,W} = F^W_{t-1} \circ \cdots \circ F^W_0$ is a random dynamical system, where at each time t, we choose $F^W_t \sim \pi_t$.
- Discrete-time RDS: $W = \{W_t\}$ iid standard normal RVs.
- For SGM: $F^W_t(x) = x + (\delta t) s(x, \tau - t) + \sqrt{\delta t}\, W_t$, $W_t \sim \mathcal{N}(0, \mathrm{Id})$.
- Markov chain with time-dependent transition kernel: $\kappa_t(x, A) = \mathbb{P}(W : F^W_t(x) \in A) = \pi_t(f : f(x) \in A)$

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.
- $F^{t,W} = F^W_{t-1} \circ \cdots \circ F^W_0$ is a random dynamical system, where at each time t, we choose $F^W_t \sim \pi_t$.
- Discrete-time RDS: $W = \{W_t\}$ iid standard normal RVs.
- For SGM: $F^W_t(x) = x + (\delta t) s(x, \tau - t) + \sqrt{\delta t}\, W_t$, $W_t \sim \mathcal{N}(0, \mathrm{Id})$.
- Markov chain with time-dependent transition kernel: $\kappa_t(x, A) = \mathbb{P}(W : F^W_t(x) \in A) = \pi_t(f : f(x) \in A)$
- $\rho_{t+1}(x) = \mathcal{L}_t \rho_t(x) = \int p_t(x, y)\ \rho_t(y)\ dy$

Non-autonomous dynamics of score generative models

- Reverse process, more generally, any coupling between p_0 with density ρ_0 and p_{data} (target) with (approximate) density ρ_τ.
- $F^{t,W} = F^W_{t-1} \circ \cdots \circ F^W_0$ is a random dynamical system, where at each time t, we choose $F^W_t \sim \pi_t$.
- Discrete-time RDS: $W = \{W_t\}$ iid standard normal RVs.
- For SGM: $F^W_t(x) = x + (\delta t) s(x, \tau - t) + \sqrt{\delta t}\, W_t$, $W_t \sim \mathcal{N}(0, \mathrm{Id})$.
- Markov chain with time-dependent transition kernel: $\kappa_t(x, A) = \mathbb{P}(W : F^W_t(x) \in A) = \pi_t(f : f(x) \in A)$
- $\rho_{t+1}(x) = \mathcal{L}_t \rho_t(x) = \int p_t(x, y)\ \rho_t(y)\ dy$
- Lyapunov exponents (finite time): perturbation evolutions through $dF^{T,W}$ [**Kifer, Young, Ledrappier, Pesin, Arnold, ...**]

Infinitesimal **perturbations** to nonautonomous systems

Sample Space

Infinitesimal **perturbations** to nonautonomous systems

Sample Space

F_t, a diffeomorphism at time t.

Infinitesimal **perturbations** to nonautonomous systems

Sample Space

F_t, a diffeomorphism at time t.

dF_t, a linear map on tangent space that evolves infinitesimal perturbations

$x \to x + \epsilon u_t(x)$, then,

$$u_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon u_t(x)) - F_t(x)}{\epsilon}$$
$$= dF_t(x)\, u_t(x)$$

Probability space

Infinitesimal **perturbations** to nonautonomous systems

Sample Space

F_t, a diffeomorphism at time t.

dF_t, a linear map on tangent space that evolves infinitesimal perturbations

$x \to x + \epsilon u_t(x)$, then,

$$u_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon u_t(x)) - F_t(x)}{\epsilon}$$
$$= dF_t(x)\, u_t(x)$$

Probability space

Evolution/pushforward operator: $p_{t+1} := F_{t\sharp} p_t$

Infinitesimal **perturbations** to nonautonomous systems

Sample Space

F_t, a diffeomorphism at time t.

dF_t, a linear map on tangent space that evolves infinitesimal perturbations

$x \to x + \epsilon u_t(x)$, then,

$$u_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon u_t(x)) - F_t(x)}{\epsilon} = dF_t(x)\, u_t(x)$$

Probability space

Evolution/pushforward operator: $p_{t+1} := F_{t\sharp} p_t$

When p_t has density ρ_t,

$$\rho_{t+1} = \mathcal{L}_t\, \rho_t := \rho_t \circ F_t^{-1} / |\det dF_t| \circ F_t^{-1}$$

Finite-time **perturbation theory**

Sample Space

Finite-time **perturbation theory**

Sample Space

$F_{t,\epsilon}(x) = F_t(x) + \epsilon\, \chi_t(x)$,
map at time t with learning errors.

Finite-time **perturbation theory**

Sample Space

$F_{t,\epsilon}(x) = F_t(x) + \epsilon\,\chi_t(x)$, map at time t with learning errors.

Perturbed dynamics: $F^t_\epsilon(x) := F_{t,\epsilon} \circ \cdots \circ F_{0,\epsilon}(x)$

Finite-time **perturbation theory**

Sample Space

$F_{t,\epsilon}(x) = F_t(x) + \epsilon\, \chi_t(x)$, map at time t with learning errors.

Perturbed dynamics: $F_\epsilon^t(x) := F_{t,\epsilon} \circ \cdots \circ F_{0,\epsilon}(x)$

Error evolution in tangent space,

$$u_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_{t,\epsilon}(x) - F_t(x)}{\epsilon} = dF_t(x)\, u_t(x) + \chi_t(x)$$

Probability space

Finite-time **perturbation theory**

Sample Space

$F_{t,\epsilon}(x) = F_t(x) + \epsilon\, \chi_t(x)$, map at time t with learning errors.

Perturbed dynamics: $F^t_\epsilon(x) := F_{t,\epsilon} \circ \cdots \circ F_{0,\epsilon}(x)$

Error evolution in tangent space,

$$u_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_{t,\epsilon}(x) - F_t(x)}{\epsilon}$$
$$= dF_t(x)\, u_t(x) + \chi_t(x)$$

Probability space

Perturbed evolution/-pushforward operator: $p_{t+1,\epsilon} := F_{t,\epsilon\sharp} p_{t,\epsilon}$

Finite-time **perturbation theory**

Sample Space

$F_{t,\epsilon}(x) = F_t(x) + \epsilon\, \chi_t(x)$, map at time t with learning errors.

Perturbed dynamics: $F_\epsilon^t(x) := F_{t,\epsilon} \circ \cdots \circ F_{0,\epsilon}(x)$

Error evolution in tangent space,

$$u_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_{t,\epsilon}(x) - F_t(x)}{\epsilon}$$
$$= dF_t(x)\, u_t(x) + \chi_t(x)$$

Probability space

Perturbed evolution/-pushforward operator: $p_{t+1,\epsilon} := F_{t,\epsilon\sharp} p_{t,\epsilon}$

When $p_{t,\epsilon}$ has density ρ_t,

$$\rho_{t+1,\epsilon} = \mathcal{L}_{t,\epsilon}\, \rho_{t,\epsilon}$$
$$:= \rho_{t,\epsilon} \circ F_{t,\epsilon}^{-1} / |\det dF_{t,\epsilon}| \circ F_{t,\epsilon}^{-1}$$

Response theory of diffusion models

- Errors in learning v_t: when $v_t \to v_t + \epsilon\chi_t$, $F_t \to F_{t,\epsilon}$, $\mathcal{L}_t \to \mathcal{L}_{t,\epsilon}$

Response theory of diffusion models

- Errors in learning v_t: when $v_t \to v_t + \epsilon\chi_t$, $F_t \to F_{t,\epsilon}$, $\mathcal{L}_t \to \mathcal{L}_{t,\epsilon}$
- What is the change in $\rho_\tau \equiv p_{\rm data}$?

Response theory of diffusion models

- Errors in learning v_t: when $v_t \to v_t + \epsilon\chi_t$, $F_t \to F_{t,\epsilon}$, $\mathcal{L}_t \to \mathcal{L}_{t,\epsilon}$
- What is the change in $\rho_\tau \equiv p_{\mathrm{data}}$?
- Robustness of support to errors?

Response theory of diffusion models

- Errors in learning v_t: when $v_t \to v_t + \epsilon\chi_t$, $F_t \to F_{t,\epsilon}$, $\mathcal{L}_t \to \mathcal{L}_{t,\epsilon}$
- What is the change in $\rho_\tau \equiv p_{\mathrm{data}}$?
- Robustness of support to errors?

$$\langle f, \partial_\epsilon|_{\epsilon=0}\mathbb{E}_W\, \mathcal{L}_\epsilon^{\tau,W}\rho_0\rangle = \langle f, \mathbb{E}_W\, \partial_\epsilon\mathcal{L}_\epsilon^{\tau,W}|_{\epsilon=0}\rho_0\rangle$$

Response theory of diffusion models

- Errors in learning v_t: when $v_t \to v_t + \epsilon\chi_t$, $F_t \to F_{t,\epsilon}$, $\mathcal{L}_t \to \mathcal{L}_{t,\epsilon}$
- What is the change in $\rho_\tau \equiv p_{\mathrm{data}}$?
- Robustness of support to errors?

$$\langle f, \partial_\epsilon|_{\epsilon=0}\mathbb{E}_W\, \mathcal{L}_\epsilon^{\tau,W}\rho_0\rangle = \langle f, \mathbb{E}_W\, \partial_\epsilon \mathcal{L}_\epsilon^{\tau,W}|_{\epsilon=0}\rho_0\rangle$$

$$\partial_\epsilon \rho_{\tau,\epsilon}(x_\tau) = -\rho_\tau(x_\tau) \sum_{t=0}^{T-1} \Big(\mathrm{div}(\chi_t)(x_{t+1}) + \chi_t(x_{t+1}) \cdot s_{t+1}(x_{t+1})\Big)$$

- Can distinguish generative models based on robustness

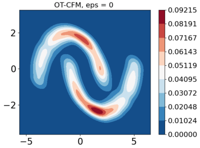
When do inexact generative models still sample the support?

SGM/Diffusion:

Left: unperturbed; Right: perturbed

When do inexact generative models still sample the support?

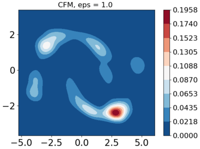
Conditional flow matching:



unperturbed

SGM/Diffusion:

Left: unperturbed; Right: perturbed



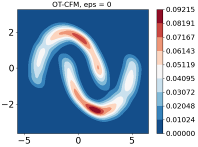
perturbed

When do inexact generative models still sample the support?

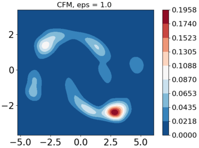
SGM/Diffusion:

Left: unperturbed; Right: perturbed

Conditional flow matching:



unperturbed



perturbed

What dynamics leads to robustness of support?

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon}$$
$$= dF_t(x)\, E_t(x)$$

Construction:

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon}$$
$$= dF_t(x)\, E_t(x)$$

Construction:

- E_0: $d <= D$ random vector fields, normalized

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon} = dF_t(x)\, E_t(x)$$

Construction:

- E_0: $d <= D$ random vector fields, normalized
- $dF_t\, E_t := E_{t+1}\, R_{t+1}$, $t \leqslant \tau$ (QR decomposition)

Sensitive subspaces

> Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon} = dF_t(x)\, E_t(x)$$

Construction:

- E_0: $d <= D$ random vector fields, normalized
- $dF_t\, E_t := E_{t+1}\, R_{t+1}$, $t \leqslant \tau$ (QR decomposition)

- $E_t(x) \in \mathbb{R}^{D \times d}$: orthonormal basis of most sensitive subspace at x.

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon} = dF_t(x)\, E_t(x)$$

Construction:

- E_0: $d <= D$ random vector fields, normalized
- $dF_t\, E_t := E_{t+1}\, R_{t+1}$, $t \leqslant \tau$ (QR decomposition)

- $E_t(x) \in \mathbb{R}^{D \times d}$: orthonormal basis of most sensitive subspace at x.
- Asymptotic convergence to Oseledets subspaces of backward Lyapunov vectors [**Arnold, Random Dynamical Systems; Oseledets, Pesin, Kifer...**]

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon} = dF_t(x)\, E_t(x)$$

Construction:

- E_0: $d <= D$ random vector fields, normalized
- $dF_t\, E_t := E_{t+1}\, R_{t+1}$, $t \leqslant \tau$ (QR decomposition)

- $E_t(x) \in \mathbb{R}^{D \times d}$: orthonormal basis of most sensitive subspace at x.
- Asymptotic convergence to Oseledets subspaces of backward Lyapunov vectors [**Arnold, Random Dynamical Systems; Oseledets, Pesin, Kifer...**]
- In finite time, E_t related to top d eigenvectors of $dF^t\, (dF^t)^\top$ (Cauchy-Green tensor)

Sensitive subspaces

Dynamical GM: $F^t(x) := F_t \circ \cdots \circ F_0(x)$

Tangent dynamics

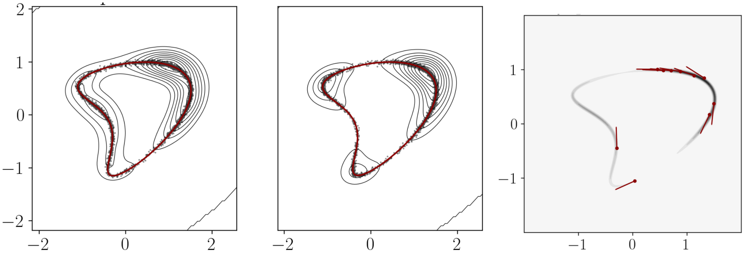
$$E_{t+1}(F_t(x)) := \lim_{\epsilon \to 0} \frac{F_t(x + \epsilon E_t(x)) - F_t(x)}{\epsilon} = dF_t(x)\, E_t(x)$$

Construction:

- E_0: $d <= D$ random vector fields, normalized
- $dF_t\, E_t := E_{t+1}\, R_{t+1}$, $t \leqslant \tau$ (QR decomposition)

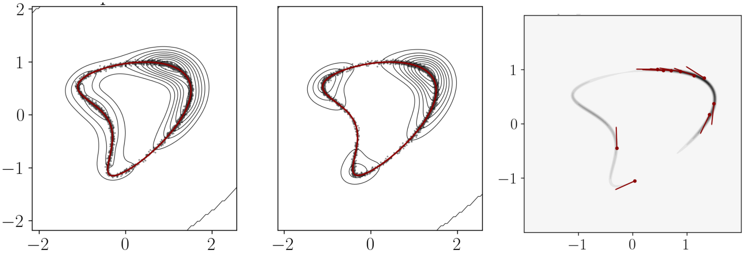
- $E_t(x) \in \mathbb{R}^{D \times d}$: orthonormal basis of most sensitive subspace at x.
- Asymptotic convergence to Oseledets subspaces of backward Lyapunov vectors [**Arnold, Random Dynamical Systems; Oseledets, Pesin, Kifer...**]
- In finite time, E_t related to top d eigenvectors of $dF^t\,(dF^t)^\top$ (Cauchy-Green tensor)
- Mean (over time) Log-diagonal of R_t: top d Lyapunov exponents.

Most sensitive subspaces of diffusion models



Left to right: Unperturbed; Perturbed; Top Lyapunov vector aligns with tangent bundle of target support.

Most sensitive subspaces of diffusion models



Left to right: Unperturbed; Perturbed; Top Lyapunov vector aligns with tangent bundle of target support.

Does reverse process learn data manifold? [**Pidstrigach 2022; Stanczuk et al 2024; Kadkhodaie et al 2024; Chen, Huang, Zhao, and Wang 2023; Lee Lu Tan 2023; Mimikos-Stamatopoulos, Zhang, Katsoulakis 2024**]

Alignment of least stable Lyapunov vectors implies robustness of support

Let $\mathrm{supp}(p_{\mathrm{data}}) = M$, a d-dimensional subset of $\mathbb{R}^D$.

Alignment of least stable Lyapunov vectors implies robustness of support

Let $\mathrm{supp}(p_{\mathrm{data}}) = M$, a d-dimensional subset of $\mathbb{R}^D$.

> **Alignment**: $E_t(x)$ spans $T_x M$ (most sensitive subspace is tangent to the data manifold)

Alignment of least stable Lyapunov vectors implies robustness of support

Let $\mathrm{supp}(p_{\mathrm{data}}) = M$, a d-dimensional subset of $\mathbb{R}^D$.

Alignment: $E_t(x)$ spans $T_x M$ (most sensitive subspace is tangent to the data manifold)

Proposition (informal): a convergent and aligned generative model learns the support of the target.

Alignment of least stable Lyapunov vectors implies robustness of support

Let $\mathrm{supp}(p_{\mathrm{data}}) = M$, a d-dimensional subset of $\mathbb{R}^D$.

Alignment: $E_t(x)$ spans $T_x M$ (most sensitive subspace is tangent to the data manifold)

Proposition (informal): a convergent and aligned generative model learns the support of the target.

Proof sketch:

- $x_1, \cdots, x_m \sim \mathrm{p}_{\mathrm{data}}$ can be mapped to $y_1, \cdots, y_m \sim p_{\tau,\epsilon}$ s.t. $\|x_i - y_i\| \sim O(\epsilon^c)$ whp. (From convergence)

Alignment of least stable Lyapunov vectors implies robustness of support

Let $\mathrm{supp}(p_{\mathrm{data}}) = M$, a d-dimensional subset of $\mathbb{R}^D$.

Alignment: $E_t(x)$ spans $T_x M$ (most sensitive subspace is tangent to the data manifold)

Proposition (informal): a convergent and aligned generative model learns the support of the target.

Proof sketch:

- $x_1, \cdots, x_m \sim \mathrm{p}_{\mathrm{data}}$ can be mapped to $y_1, \cdots, y_m \sim p_{\tau,\epsilon}$ s.t. $\|x_i - y_i\| \sim O(\epsilon^c)$ whp. (From convergence)
- $x_i - y_i$ parallel to $E_t(x_i)$

Alignment of least stable Lyapunov vectors implies robustness of support

Let $\mathrm{supp}(p_{\mathrm{data}}) = M$, a d-dimensional subset of $\mathbb{R}^D$.

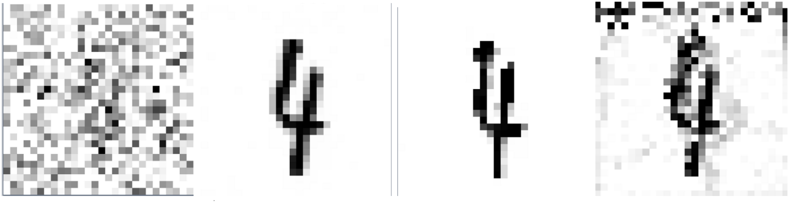
Alignment: $E_t(x)$ spans $T_x M$ (most sensitive subspace is tangent to the data manifold)

Proposition (informal): a convergent and aligned generative model learns the support of the target.

Proof sketch:

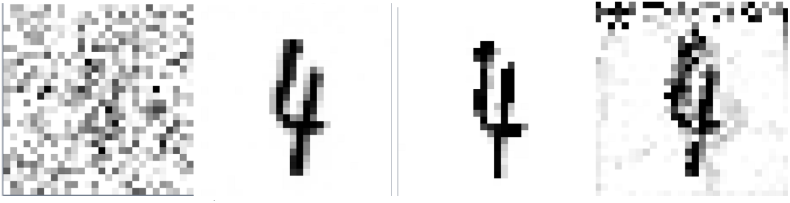
- $x_1, \cdots, x_m \sim \mathrm{p}_{\mathrm{data}}$ can be mapped to $y_1, \cdots, y_m \sim p_{\tau,\epsilon}$ s.t. $\|x_i - y_i\| \sim O(\epsilon^c)$ whp. (From convergence)
- $x_i - y_i$ parallel to $E_t(x_i)$
- Margin of one-class classifier learned on x_i does not change on y_i

Aligned generative models can learn the support



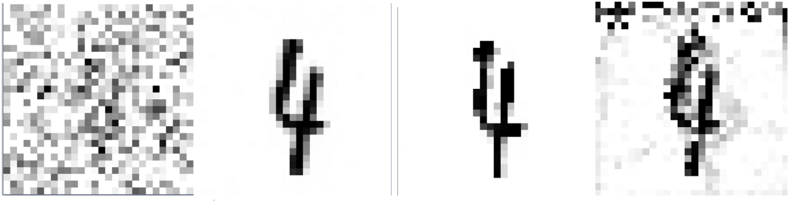
Source sample

Aligned generative models can learn the support



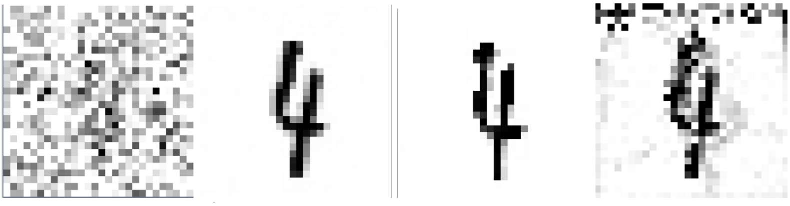
Source sample Predicted

Aligned generative models can learn the support



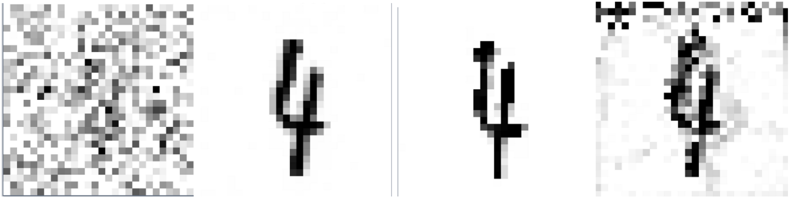
Source sample   Predicted   + most sensitive LV

Aligned generative models can learn the support

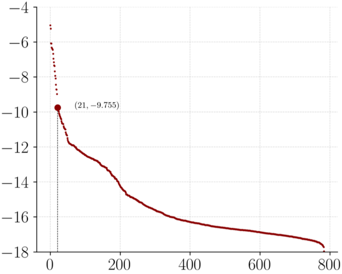


Source sample | Predicted | + most sensitive LV | + 100th LV

Aligned generative models can learn the support



Source sample Predicted + most sensitive LV + 100th LV



Index

Lyapunov exponents

When can we expect alignment?

Theorem (informal): If F^τ is compressive overall, v_t is uniformly compressive for t close to τ, and v_t has small cross-derivatives, alignment holds.

When can we expect alignment?

Theorem (informal): If F^τ is compressive overall, v_t is uniformly compressive for t close to τ, and v_t has small cross-derivatives, alignment holds.

Evolution of score components, $s_t := d\ \log p_t$, along most sensitive subspaces:

$$(s_{t+1}\ E_{t+1}) \circ F_t = s_t\ E_t\ R_t^{-1} - \mathrm{tr}((dF_t^{-1} d^2 F_t)\ E_t R_t^{-1}.$$

When can we expect alignment?

Theorem (informal): If F^{τ} is compressive overall, v_t is uniformly compressive for t close to τ, and v_t has small cross-derivatives, alignment holds.

Evolution of score components, $s_t := d\, \log p_t$, along most sensitive subspaces:

$$(s_{t+1}\, E_{t+1}) \circ F_t = s_t\, E_t\, R_t^{-1} - \mathrm{tr}((dF_t^{-1} d^2 F_t)\, E_t R_t^{-1}.$$

- $s_{\tau} \in (TM)^{\perp}$ for singular targets

When can we expect alignment?

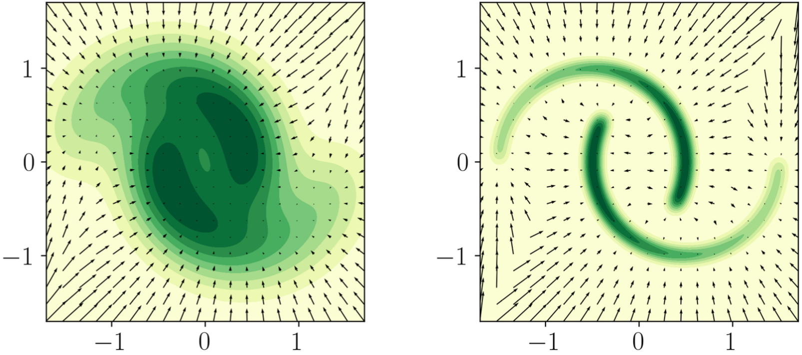
Theorem (informal): If F^τ is compressive overall, v_t is uniformly compressive for t close to τ, and v_t has small cross-derivatives, alignment holds.

Evolution of score components, $s_t := d \log p_t$, along most sensitive subspaces:

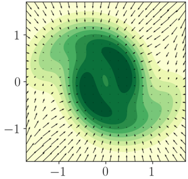
$$(s_{t+1}\, E_{t+1}) \circ F_t = s_t\, E_t\, R_t^{-1} - \mathrm{tr}((dF_t^{-1} d^2 F_t)\, E_t R_t^{-1}.$$

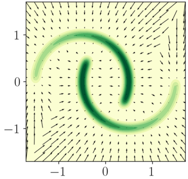
- $s_\tau \in (TM)^\perp$ for singular targets
- Compression $\implies |\det R_t| < 1$.

The dynamics of alignment: the vector field is a uniform attractive force at the end

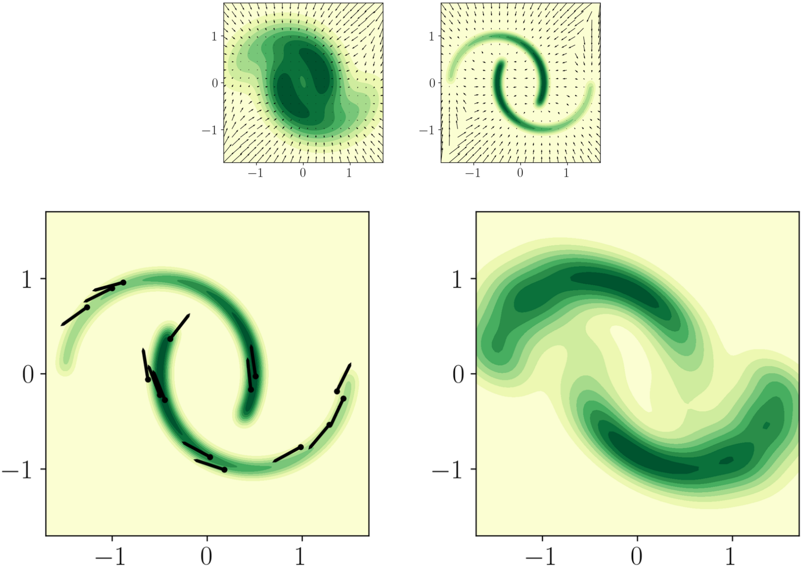


The dynamics of alignment: the vector field is a uniform attractive force at the end

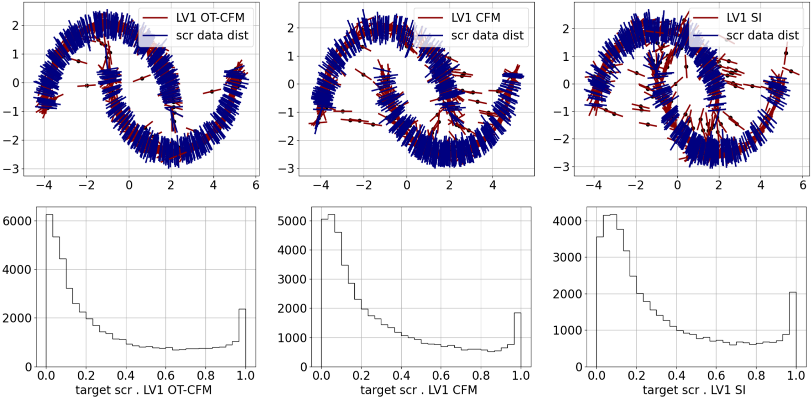




The dynamics of alignment: the vector field is a uniform attractive force at the end

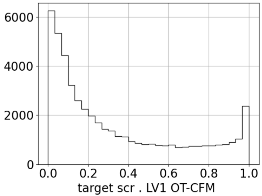


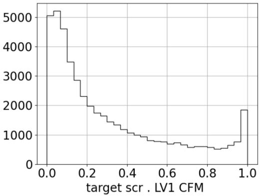
Less alignment leads to less robustness

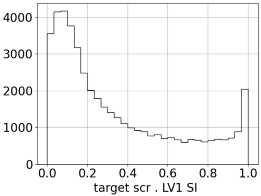


Histograms of angles b/w top LV (most sensitive subspace) and target score for OT-CFM (left), CFM (center) and Stochastic Interpolants (right).

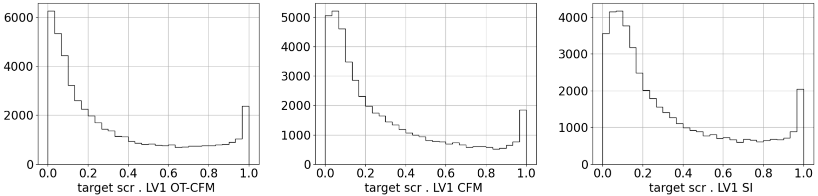
Robustness of non-aligned models

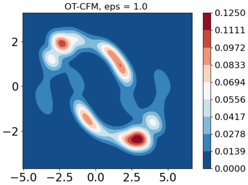




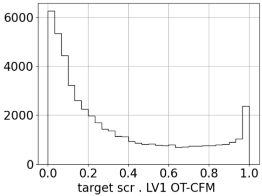


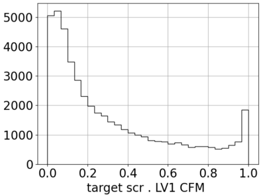
Robustness of non-aligned models

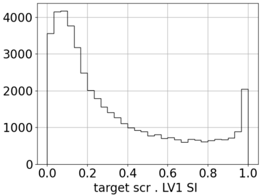


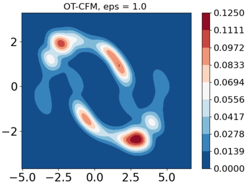


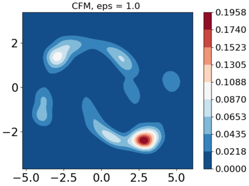
Robustness of non-aligned models



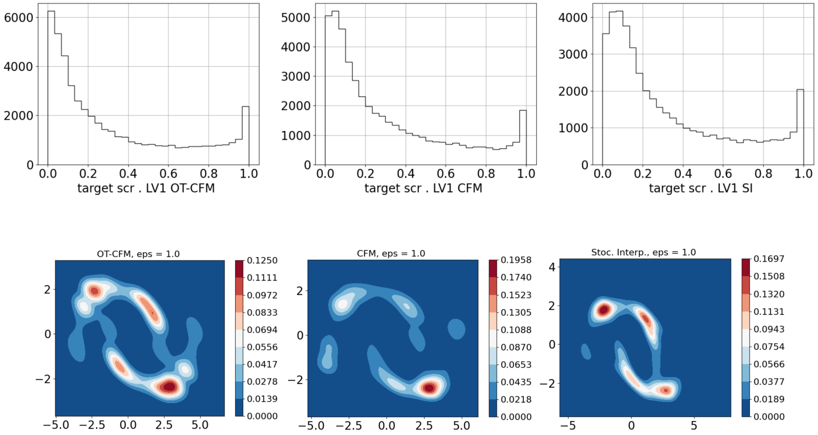








Robustness of non-aligned models



The dynamics of robust generative models

- Alignment of most sensitive LVs with tangent spaces of the data manifold leads to robustness

The dynamics of robust generative models

- Alignment of most sensitive LVs with tangent spaces of the data manifold leads to robustness
- Some compressive dynamics can lead to alignment

The dynamics of robust generative models

- Alignment of most sensitive LVs with tangent spaces of the data manifold leads to robustness
- Some compressive dynamics can lead to alignment
- Aligned and convergent GMs are manifold learners

The dynamics of robust generative models

- Alignment of most sensitive LVs with tangent spaces of the data manifold leads to robustness
- Some compressive dynamics can lead to alignment
- Aligned and convergent GMs are manifold learners

> Lemma: Alignment property is regular.

The dynamics of robust generative models

- Alignment of most sensitive LVs with tangent spaces of the data manifold leads to robustness
- Some compressive dynamics can lead to alignment
- Aligned and convergent GMs are manifold learners

> Lemma: Alignment property is regular.

- An aligned GM retains alignment under perturbations.

Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\mathrm{nn}}(x_t) \approx x_{t+1} = F(x_t)$?

Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\mathrm{nn}}(x_t) \approx x_{t+1} = F(x_t)$?
- ... while also ensuring F_{nn} is *physical*?

Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\mathrm{nn}}(x_t) \approx x_{t+1} = F(x_t)$?
- ... while also ensuring F_{nn} is *physical*?
- That is, does F_{nn} sample from μ?

Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\mathrm{nn}}(x_t) \approx x_{t+1} = F(x_t)$?
- ... while also ensuring F_{nn} is *physical*?
- That is, does F_{nn} sample from μ?
- Optimize over neural representations F_{nn}
  $\ell(x, F_{\mathrm{nn}}) = \|F(x) - F_{\mathrm{nn}}(x)\|^2$

Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\rm nn}(x_t) \approx x_{t+1} = F(x_t)$?
- ... while also ensuring $F_{\rm nn}$ is *physical*?
- That is, does $F_{\rm nn}$ sample from μ?
- Optimize over neural representations $F_{\rm nn}$ $\ell(x, F_{\rm nn}) = \|F(x) - F_{\rm nn}(x)\|^2$
- Training loss: $R_S(F_{\rm nn}) = (1/m) \sum_{x \in S} \ell(x, F_{\rm nn})$, $S \sim \mu^m$.

Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\rm nn}(x_t) \approx x_{t+1} = F(x_t)$?
- ... while also ensuring $F_{\rm nn}$ is *physical*?
- That is, does $F_{\rm nn}$ sample from μ?
- Optimize over neural representations $F_{\rm nn}$ $\ell(x, F_{\rm nn}) = \|F(x) - F_{\rm nn}(x)\|^2$
- Training loss: $R_S(F_{\rm nn}) = (1/m) \sum_{x \in S} \ell(x, F_{\rm nn})$, $S \sim \mu^m$.
- Neural ODE [**Chen et al 2018**]:

$$\frac{d}{dt} \varphi^t_{\rm nn}(x) = v_{\rm nn}(\varphi^t_{\rm nn}(x)), \quad x \in \mathbb{R}^d. \tag{1}$$

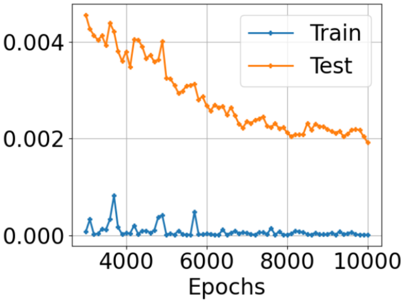
Generative models for chaotic systems: setup

- Given m samples, $\{(x_t, F(x_t))\}_{t \leqslant m}$, can we learn $F_{\mathrm{nn}}(x_t) \approx x_{t+1} = F(x_t)$?
- ... while also ensuring F_{nn} is *physical*?
- That is, does F_{nn} sample from μ?
- Optimize over neural representations F_{nn}
  $\ell(x, F_{\mathrm{nn}}) = \|F(x) - F_{\mathrm{nn}}(x)\|^2$
- Training loss: $R_S(F_{\mathrm{nn}}) = (1/m) \sum_{x \in S} \ell(x, F_{\mathrm{nn}})$, $S \sim \mu^m$.
- Neural ODE [**Chen et al 2018**]:

$$\frac{d}{dt} \varphi^t_{\mathrm{nn}}(x) = v_{\mathrm{nn}}(\varphi^t_{\mathrm{nn}}(x)), \quad x \in \mathbb{R}^d. \tag{1}$$

- Fix some δt and set $F \equiv \varphi^{\delta t}$, where $d\varphi^t(x)/dt = v(\varphi^t(x))$

Physical neural parameterization via minimizing MSE



Good “generalization” performance.

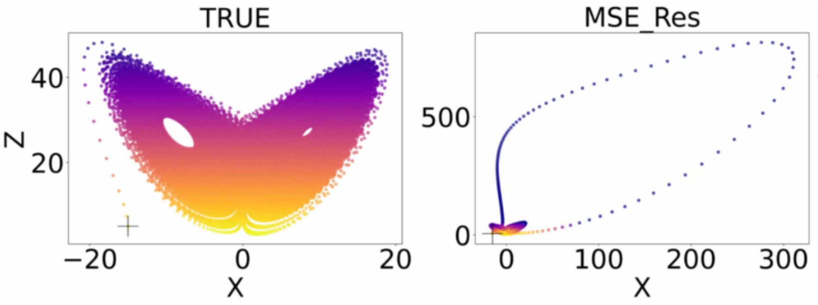
Several different architectures and hyperparameter choices produce acceptable generalization error $= E_{x \sim \mu} \ell(x, F_{\mathrm{nn}})$.

Generalization $\implies$ learning dynamics?

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$

Generalization $\implies$ learning dynamics?

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$



Jacobian-matching loss

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$
+Jacobian info	$[0.9022, -0.0024, -14.4803]$

- Modified loss:
  $\ell(x, F_{\mathrm{nn}}) = \|F_{\mathrm{nn}}(x) - F(x)\|^2 + \lambda\|dF_{\mathrm{nn}}(x) - dF(x)\|^2$

Jacobian-matching loss

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$
+Jacobian info	$[0.9022, -0.0024, -14.4803]$

- Modified loss:
  $\ell(x, F_{\mathrm{nn}}) = \|F_{\mathrm{nn}}(x) - F(x)\|^2 + \lambda \|dF_{\mathrm{nn}}(x) - dF(x)\|^2$
- With modified loss, statistical moments (correlations, LEs) are accurate

Jacobian-matching loss

	Lyapunov Exponent
True LE	$\approx$ **[0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$
+Jacobian info	$[0.9022, -0.0024, -14.4803]$

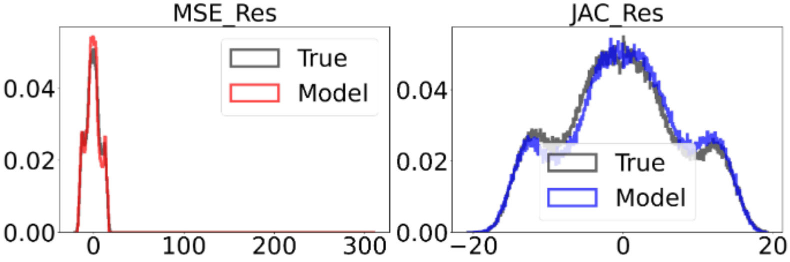
- Modified loss:
  $\ell(x, F_{\mathrm{nn}}) = \|F_{\mathrm{nn}}(x) - F(x)\|^2 + \lambda \|dF_{\mathrm{nn}}(x) - dF(x)\|^2$
- With modified loss, statistical moments (correlations, LEs) are accurate
- Physical measure: $\mathrm{Unif}(x_0, x_1, \cdots, x_t) \xrightarrow[t \to \infty]{} \mu$ for Leb a.e. x_0.

Jacobian-matching loss

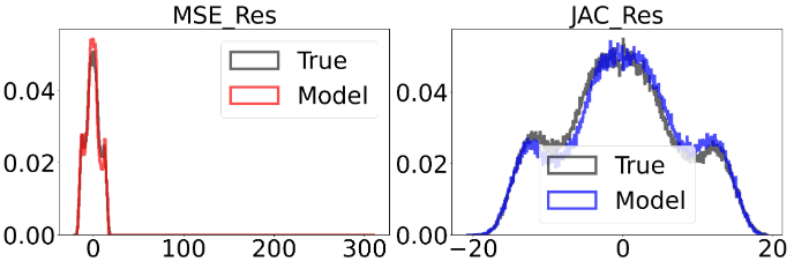
	Lyapunov Exponent
True LE	**$\approx$ [0.9, 0, -14.5]**
Neural ODE	$[0.8926, -0.0336, -6.0616]$
+Jacobian info	$[0.9022, -0.0024, -14.4803]$

- Modified loss:
  $\ell(x, F_{\mathrm{nn}}) = \|F_{\mathrm{nn}}(x) - F(x)\|^2 + \lambda \|dF_{\mathrm{nn}}(x) - dF(x)\|^2$
- With modified loss, statistical moments (correlations, LEs) are accurate
- Physical measure: $\mathrm{Unif}(x_0, x_1, \cdots, x_t) \xrightarrow{t \to \infty} \mu$ for Leb a.e. x_0.
- $(1/T) \sum_{t \leqslant T} J(x_t) \xrightarrow{t \to \infty} \mathbb{E}_{x \sim \mu} J(x)$, for Leb a.e. x_0.

C^1 matching of vector field leads to learning physical measure

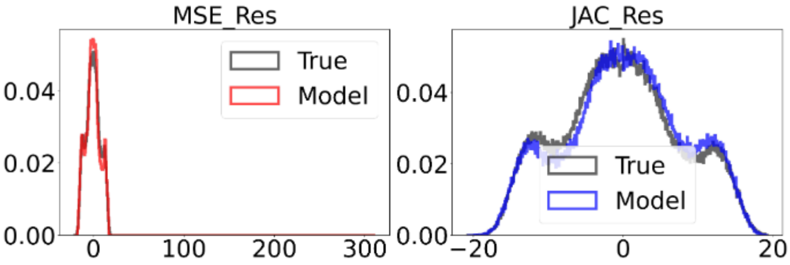


C^1 matching of vector field leads to learning physical measure



- is Jacobian-matching always enough to learn physical dynamics?

C^1 matching of vector field leads to learning physical measure



- is Jacobian-matching always enough to learn physical dynamics?
- comparison against generative modeling of the physical measure?

Generalization and shadowing

- Good generalization with Jacobian loss ($\mathbb{E}_{x\sim\mu}\ell(x, F_{\rm nn})$) $\implies$ shadowing holds w.h.p.

Generalization and shadowing

- Good generalization with Jacobian loss ($\mathbb{E}_{x\sim\mu}\ell(x, F_{\rm nn})$) $\implies$ shadowing holds w.h.p.
- Jacobian-matching leads to learning shadowing orbits w.h.p.

Generalization and shadowing

- Good generalization with Jacobian loss ($\mathbb{E}_{x\sim\mu}\ell(x, F_{\rm nn})$) $\implies$ shadowing holds w.h.p.
- Jacobian-matching leads to learning shadowing orbits w.h.p.
- No shadowing guarantee for MSE loss.

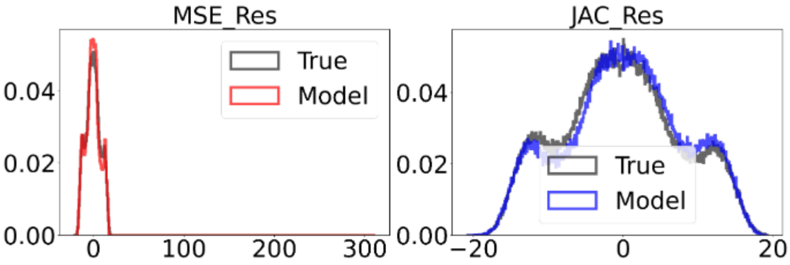
Generalization and shadowing

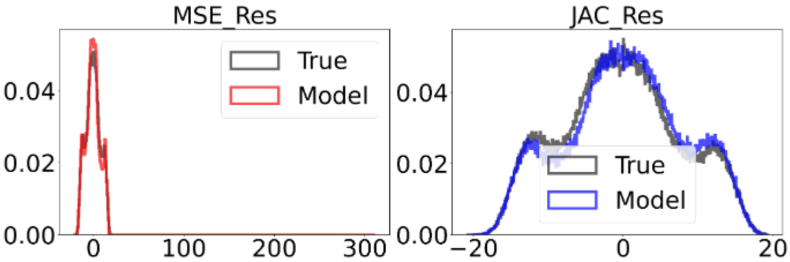
- Good generalization with Jacobian loss ($\mathbb{E}_{x\sim\mu}\ell(x, F_{\rm nn})$) $\implies$ shadowing holds w.h.p.
- Jacobian-matching leads to learning shadowing orbits w.h.p.
- No shadowing guarantee for MSE loss.
- $\mu_m^{\rm sh}(x) = {\rm Unif}(x, F(x), \cdots, F^m(x))$.

Generalization and shadowing

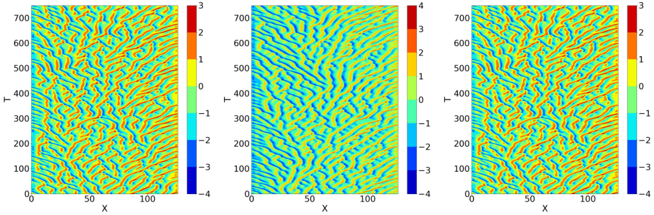
- Good generalization with Jacobian loss ($\mathbb{E}_{x\sim\mu}\ell(x, F_{\rm nn})$) $\implies$ shadowing holds w.h.p.
- Jacobian-matching leads to learning shadowing orbits w.h.p.
- No shadowing guarantee for MSE loss.
- $\mu_m^{\rm sh}(x) = {\rm Unif}(x, F(x), \cdots, F^m(x))$.
- $\mu_m(x) = {\rm Unif}(x, F_{\rm nn}(x), \cdots, F_{\rm nn}^m(x))$.

> Let $F_{\rm nn}$ be a model of F that satisfies i)C^1 strong generalization and ii) $\lim_{m\to\infty} W^1(\mu_m^{\rm sh}(x), \mu) \leqslant \epsilon_2$ w.h.p. Then, w.h.p., $\lim_{m\to\infty} W^1(\mu_m^{\rm nn}, \mu) \approx 0$.

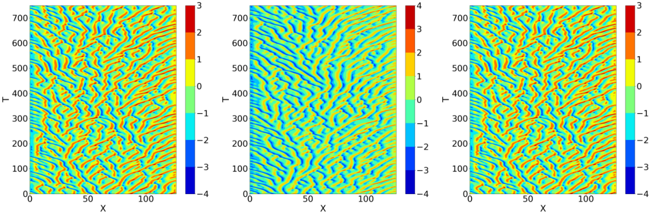

MSE_Res
True
Model
0.04
0.02
0.00
0
100
200
300
JAC_Res
True
Model
0.04
0.02
0.00
−20
0
20



		Norm Difference		
Model	Loss	$W^1(\hat{\mu}_{500}, \mu_{\mathrm{NN},500})$	$\|\Lambda - \Lambda_{\mathrm{NN}}\|$	$\|\langle x\rangle_{500} - \langle x\rangle_{500,\mathrm{NN}}\|$
MLP	MSE	18.9711	9.6950	15.2220
MLP	JAC	0.6800	0.0118	0.6524
ResNet	MSE	1.3567	10.8516	0.7760
ResNet	JAC	**0.1433**	**0.0106**	**0.0559**
FNO	MSE	10.5409	22.1600	9.4270
FNO	JAC	1.3076	0.0505	0.9748



Left: KS solutions; Center: NN network based on MSE loss; Right: Jacobian-matching loss



Left: KS solutions; Center: NN network based on MSE loss; Right: Jacobian-matching loss

Rössler	[0.0665, -0.0004 -5.4112]	[0.0008,-0.0285 -1.4108]	**[0.0609, -0.0004 -5.3808]**
Hyperchaos	[4.0039, 0.0082 -19.9972, -48.0205]	[4.1393, 0.0955 -15.2120, -29.9480]	**[4.3789, -0.1617 -19.9974, -48.0205]**
Kuramoto-Sivashinsky	[0.3036, 0.2733, 0.2592, 0.2257, 0.2050, 0.1888, 0.1649, 0.1496, 0.1288, 0.1128, 0.0992, 0.0776, 0.0646, 0.0492, 0.0342]	[0.1652, 0.1647, 0.1540, 0.1524, 0.1443, 0.1411, 0.1336, 0.1262, 0.1236, 0.1143, 0.1141, 0.1091, 0.1045, 0.0971, 0.0985]	**[0.2904, 0.2622, 0.2293, 0.1990, 0.1701, 0.1584, 0.1320, 0.1071, 0.0912, 0.0724, 0.0591, 0.0442, 0.0306, 0.0157, 0.0023,]**

Only 2 out of first 64 LEs predicted with $< 10\%$ error

Score learning to sample from chaotic systems?

Generative modeling: want samples from ν given $x_1, \cdots, x_m \sim \nu$

Score learning to sample from chaotic systems?

Generative modeling: want samples from ν given $x_1, \cdots, x_m \sim \nu$

Goal: exploit dynamical systems theory for dimension reduction

Sampling: want samples from a target measure ν given **score**, $\nabla \log \rho^\nu$

C, Schäfer and Marzouk, AISTATS 2024; **C and Wang** SIAM J. Appl. Dyn. Sys 2022

Score learning to sample from chaotic systems?

Generative modeling: want samples from ν given $x_1, \cdots, x_m \sim \nu$

Goal: exploit dynamical systems theory for dimension reduction

Sampling: want samples from a target measure ν given **score**, $\nabla \log \rho^\nu$

C, Schäfer and Marzouk, AISTATS 2024; **C and Wang** SIAM J. Appl. Dyn. Sys 2022

Sampling via measure transport

- Target measure: μ with density ρ^{μ}.
- Tractable source measure ν with density ρ^{ν}.
- $\mathrm{supp}(\nu) = \mathbb{X}$ and $\mathrm{supp}(\mu) = \mathbb{Y}$.

Sampling via measure transport

- Target measure: μ with density ρ^μ.
- Tractable source measure ν with density ρ^ν.
- $\mathrm{supp}(\nu) = \mathbb{X}$ and $\mathrm{supp}(\mu) = \mathbb{Y}$.

A transport map $T : \mathbb{X} \to \mathbb{Y}$ is an invertible transformation such that $T_\sharp \nu = \mu$.

The **score** of a probability measure μ with density ρ^μ is $\nabla \log \rho^\mu$.

Sampling via measure transport

- Target measure: μ with density ρ^μ.
- Tractable source measure ν with density ρ^ν.
- $\mathrm{supp}(\nu) = \mathbb{X}$ and $\mathrm{supp}(\mu) = \mathbb{Y}$.

A transport map $T : \mathbb{X} \to \mathbb{Y}$ is an invertible transformation such that $T_\sharp \nu = \mu$.

The **score** of a probability measure μ with density ρ^μ is $\nabla \log \rho^\mu$.

Known prior score $\implies$ known target score

$$\underbrace{\nu}_{\text{target}} = \underbrace{\ell(y,\cdot)}_{\text{likelihood}} \times \underbrace{F_\sharp \mu}_{\text{prior}} / Z$$

The score operator

Change of variables/pushforward operation:

$$\rho^{\mu} = \frac{\rho^{\nu} \circ T^{-1}}{|\det \nabla T| \circ T^{-1}}$$

Pushforward operation on **scores**:

$$\begin{aligned} \mathcal{G}(s, U) &= \left(s(\nabla U)^{-1} - \nabla \log |\det \nabla U| (\nabla U)^{-1}\right) \circ U^{-1} \\ &= \left(s\,(\nabla U)^{-1} - \operatorname{tr}\left((\nabla U)^{-1} \nabla^2 U\right) (\nabla U)^{-1}\right) \circ U^{-1}, \end{aligned}$$

Score operator conditioned on unstable manifolds

$$\mathcal{G}(s^{\mu}, F) = s^{\mu}\, (\nabla^u F)^{-1} - \mathrm{tr}((\nabla^u F)^{-1} \nabla^{u^2} F)(\nabla^u F)^{-1}) \circ F^{-1}$$

- If $F_\sharp \mu = \mu$, and F is chaotic, $\mathcal{U}(\cdot, F)$ is a contraction with s^{μ} as fixed point [**C and Wang SIAM Appl. Dyn. Sys 2022, Ni 2022**]

Score operator conditioned on unstable manifolds

$$\mathcal{G}(s^{\mu}, F) = s^{\mu}\,(\nabla^u F)^{-1} - \mathrm{tr}((\nabla^u F)^{-1} \nabla^{u^2} F)(\nabla^u F)^{-1}) \circ F^{-1}$$

- If $F_{\sharp}\mu = \mu$, and F is chaotic, $\mathcal{U}(\cdot, F)$ is a contraction with s^{μ} as fixed point [**C and Wang SIAM Appl. Dyn. Sys 2022, Ni 2022**]
- Fixed point iteration produces target score s^{μ} anywhere with exponential precision.

Score operator conditioned on unstable manifolds

$$\mathcal{G}(s^{\mu}, F) = s^{\mu}\,(\nabla^u F)^{-1} - \mathrm{tr}((\nabla^u F)^{-1} \nabla^{u^2} F)(\nabla^u F)^{-1}) \circ F^{-1}$$

- If $F_\sharp \mu = \mu$, and F is chaotic, $\mathcal{U}(\cdot, F)$ is a contraction with s^{μ} as fixed point [**C and Wang SIAM Appl. Dyn. Sys 2022, Ni 2022**]
- Fixed point iteration produces target score s^{μ} anywhere with exponential precision.
- With target score, can use any score-based sampling algorithm in reduced dimension.

A dynamical view of sampling and generative modeling

Dimension reduction in sampling/GM via projections on unstable manifolds

A dynamical view of sampling and generative modeling

Dimension reduction in sampling/GM via projections on unstable manifolds

Jacobian-matching leads to statistical accuracy in chaotic systems

A dynamical view of sampling and generative modeling

Dimension reduction in sampling/GM via projections on unstable manifolds

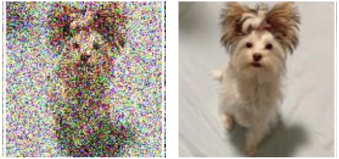
Jacobian-matching leads to statistical accuracy in chaotic systems

For robust GMs, finite time Lyapunov vectors of the generating process span the tangent bundle of the data manifold

References: C and de Clercq, 2025 (submitted); Park, Yang and C, NeuRIPS 2024; C, Schafer, Marzouk AISTATS 2024.

Operator approximation to replace score learning for the reverse process

- Approximate $\{T_{W_i}\}$ for different noise paths W_i using only forward process
- Solve for a map T_W for a new W using operator-valued kernel regression



Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- Given samples $x_1, \cdots, x_m \sim \nu$, generate more samples from ν.
- A deterministic nonparametric transport method derived with an operator root-finding principle.

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- Given samples $x_1, \cdots, x_m \sim \nu$, generate more samples from ν.
- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Newton-like features: unstable, converges fast

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- Given samples $x_1, \cdots, x_m \sim \nu$, generate more samples from ν.
- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- Given samples $x_1, \cdots, x_m \sim \nu$, generate more samples from ν.
- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails
- Next steps: nonparametric PDE solves e.g. particle vortex methods, smooth particle hydrodynamics, PINNs etc.

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- Given samples $x_1, \cdots, x_m \sim \nu$, generate more samples from ν.
- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails
- Next steps: nonparametric PDE solves e.g. particle vortex methods, smooth particle hydrodynamics, PINNs etc.
- Low-rank approximations of elliptic PDE solution?

Score Operator Newton construction: can be used for sampling, generative modeling, Bayesian inference and filtering in chaotic systems

- Given samples $x_1, \cdots, x_m \sim \nu$, generate more samples from ν.
- A deterministic nonparametric transport method derived with an operator root-finding principle.
- Newton-like features: unstable, converges fast
- Global nature of elliptic PDE helps i) avoid mode collapse and ii) capture tails
- Next steps: nonparametric PDE solves e.g. particle vortex methods, smooth particle hydrodynamics, PINNs etc.
- Low-rank approximations of elliptic PDE solution?
- Projected Bayesian filtering in chaotic systems

C, Schäfer, Marzouk AISTATS 2024

Bayesian data assimilation

- Dynamical system: $X_{t+1} = \varphi(X_t)$

Bayesian data assimilation

- Dynamical system: $X_{t+1} = \varphi(X_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$

Bayesian data assimilation

- Dynamical system: $X_{t+1} = \varphi(X_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$

Bayesian data assimilation

- Dynamical system: $X_{t+1} = \varphi(X_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$
- At observation time t, want to sample
  $\nu_t \equiv \mathbb{P}(X_t | Y_1, \cdots, Y_t) \propto \ell(y_t, \cdot)\ \varphi_\sharp \nu_{t-1}$, filtering distribution

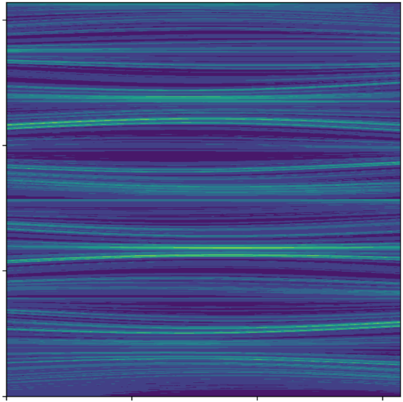
Bayesian data assimilation

- Dynamical system: $X_{t+1} = \varphi(X_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$
- At observation time t, want to sample $\nu_t \equiv \mathbb{P}(X_t | Y_1, \cdots, Y_t) \propto \ell(y_t, \cdot)\ \varphi_\sharp \nu_{t-1}$, filtering distribution
- **Pushforward**: $\pi = \varphi_\sharp \mu = \mu \circ \varphi^{-1}$ or if $x \sim \mu$, $\varphi(x) \sim \pi$.

Bayesian data assimilation

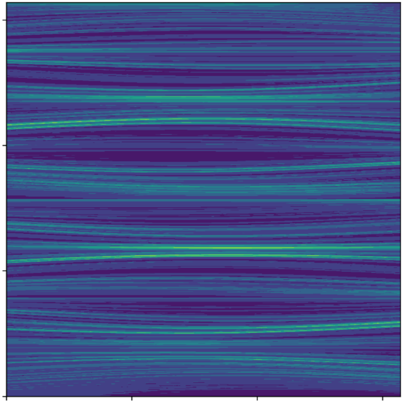
- Dynamical system: $X_{t+1} = \varphi(X_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$
- At observation time t, want to sample $\nu_t \equiv \mathbb{P}(X_t | Y_1, \cdots, Y_t) \propto \ell(y_t, \cdot)\ \varphi_\sharp \nu_{t-1}$, filtering distribution
- **Pushforward**: $\pi = \varphi_\sharp \mu = \mu \circ \varphi^{-1}$ or if $x \sim \mu$, $\varphi(x) \sim \pi$.

Absolutely continuous conditional structure



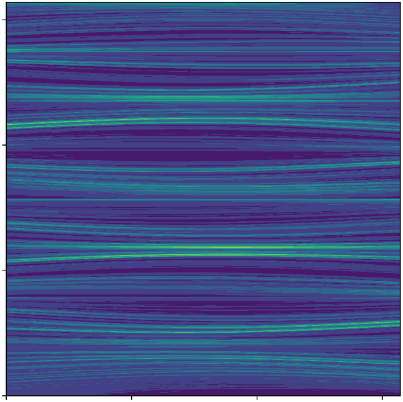
- Empirical density of a perturbed Baker’s map

Absolutely continuous conditional structure



- Empirical density of a perturbed Baker's map
- Unstable manifold – 1D roughly horizontal curves.

Absolutely continuous conditional structure



- Empirical density of a perturbed Baker's map
- Unstable manifold – 1D roughly horizontal curves.
- SRB measure may be singular

Operator approximation to replace score learning for the reverse process

- Need $s_t(x) = \nabla \rho_{1-t}(x)$ to simulate reverse process

Operator approximation to replace score learning for the reverse process

- Need $s_t(x) = \nabla \rho_{1-t}(x)$ to simulate reverse process
- Approximate $\{T_{W_i}\}$ for different noise paths W_i using only forward process (using Koopman operator methods)

Operator approximation to replace score learning for the reverse process

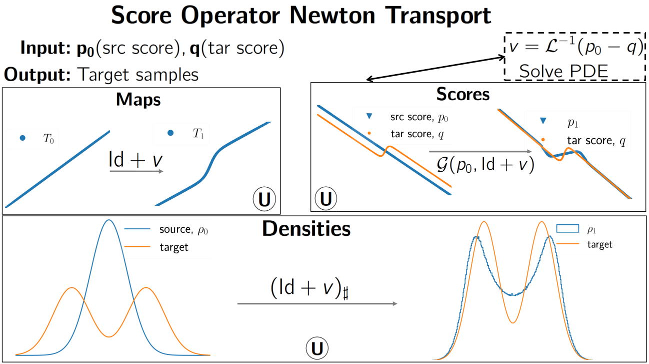
- Need $s_t(x) = \nabla \rho_{1-t}(x)$ to simulate reverse process
- Approximate $\{T_{W_i}\}$ for different noise paths W_i using only forward process (using Koopman operator methods)
- Solve for a map T_W for a new W using operator-valued kernel regression

Projected SCONE

- Similar derivation for operator $\mathcal{U}$

Projected SCONE

- Similar derivation for operator $\mathcal{U}$
- $v \in (E^u)^*$ solutions of projected SCONE iteration



Generative modeling of physical measure

- Latent SDE/ODE approach [**Kidger et al 2022**]:
  $F_{\mathrm{ls}} := f_\theta \circ \Phi^t_\phi \circ g_\phi,$

Generative modeling of physical measure

- Latent SDE/ODE approach [**Kidger et al 2022**]:
  $F_{\rm ls} := f_\theta \circ \Phi^t_\phi \circ g_\phi$,
- $g_\phi : \mathbb{R}^d \to \mathbb{R}^{d_l}$, an embedding from the data to latent space ($\mathbb{R}^{d_l}$), with $g_{\phi\sharp}\mu = q_{\phi,0}$.

Generative modeling of physical measure

- Latent SDE/ODE approach [**Kidger et al 2022**]: $F_{\mathrm{ls}} := f_\theta \circ \Phi_\phi^t \circ g_\phi$,
- $g_\phi : \mathbb{R}^d \to \mathbb{R}^{d_l}$, an embedding from the data to latent space $(\mathbb{R}^{d_l})$, with $g_{\phi\sharp}\mu = q_{\phi,0}$.
- Latent dynamics: $\Phi_\phi^t : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$; pushforward distributions $\Phi^t_{\phi\sharp} q_{\phi,0} = q_{\phi,t}$.

Generative modeling of physical measure

- Latent SDE/ODE approach [**Kidger et al 2022**]:
  $F_{\mathrm{ls}} := f_\theta \circ \Phi^t_\phi \circ g_\phi$,
- $g_\phi : \mathbb{R}^d \to \mathbb{R}^{d_l}$, an embedding from the data to latent space ($\mathbb{R}^{d_l}$), with $g_{\phi\sharp}\mu = q_{\phi,0}$.
- Latent dynamics: $\Phi^t_\phi : \mathbb{R}^{d_l} \to \mathbb{R}^{d_l}$; pushforward distributions $\Phi^t_{\phi\sharp} q_{\phi,0} = q_{\phi,t}$.
- Decoder: $f_\theta : \mathbb{R}^{d_l} \to \mathbb{R}^d$ $f_{\theta\sharp} q_{\phi,t} = p_\theta(\cdot|Z_t)$.
-

$$\ell_{\mathrm{ls}}(X_{1:m},(\phi,\theta)) := \sum_{t=1}^{T} \mathbb{E}_{z_t \sim q_{\phi,t}(\cdot|X_{1:T})}[-\log p_\theta(x_t|z_t)] + \mathrm{KL}(q_{\phi,0}(\cdot|X_{1:T}) \| p_{Z_0}),$$

An efficient algorithm to sample from a high-dimensional target distribution (e.g., a Bayesian posterior) when *partially* specified

An efficient algorithm to sample from a high-dimensional target distribution (e.g., a Bayesian posterior) when *partially* specified

$$\underbrace{\nu}_{\text{target}} = \underbrace{L(y,\cdot)}_{\text{likelihood}} \times \underbrace{F_\sharp \mu}_{\text{prior}} / Z$$

An efficient algorithm to sample from a high-dimensional target distribution (e.g., a Bayesian posterior) when *partially* specified

$$\underbrace{\nu}_{\text{target}} = \underbrace{L(y,\cdot)}_{\text{likelihood}} \times \underbrace{F_\sharp \mu}_{\text{prior}} / Z$$

Goal: to sample efficiently from ν

An efficient algorithm to sample from a high-dimensional target distribution (e.g., a Bayesian posterior) when *partially* specified

$$\underbrace{\nu}_{\text{target}} = \underbrace{L(y,\cdot)}_{\text{likelihood}} \times \underbrace{F_\sharp \mu}_{\text{prior}} / Z$$

Goal: to sample efficiently from ν

A statistically consistent sampler $\Longrightarrow$ reliable estimates of state given past observations

Motivating problem: Bayesian data assimilation/Bayesian inference

- Dynamical system: $x_{t+1} = \varphi(x_t)$

Motivating problem: Bayesian data assimilation/Bayesian inference

- Dynamical system: $x_{t+1} = \varphi(x_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$

Motivating problem: Bayesian data assimilation/Bayesian inference

- Dynamical system: $x_{t+1} = \varphi(x_t)$
- Hidden orbit: x, $\varphi(x)$, $\varphi^2(x), \cdots$,
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$

Motivating problem: Bayesian data assimilation/Bayesian inference

- Dynamical system: $x_{t+1} = \varphi(x_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$
- At observation time t, $\nu_t \propto \ell(y_t, \cdot)\ \varphi_\sharp \nu_{t-1}$, **filtering distribution**

Motivating problem: Bayesian data assimilation/Bayesian inference

- Dynamical system: $x_{t+1} = \varphi(x_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots ,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$
- At observation time t, $\nu_t \propto \ell(y_t, \cdot)\ \varphi_\sharp \nu_{t-1}$, **filtering distribution**
- Physical measure: $\mathrm{Unif}(x_0, x_1, \cdots, x_t) \xrightarrow{t \to \infty} \mu = \nu_0$ for Leb a.e. x_0.

Motivating problem: Bayesian data assimilation/Bayesian inference

- Dynamical system: $x_{t+1} = \varphi(x_t)$
- Hidden orbit: $x, \varphi(x), \varphi^2(x), \cdots,$
- Observation: $Y_t = h(X_t) + \epsilon_t$, Likelihood: $x \to \ell(y, x)$
- At observation time t, $\nu_t \propto \ell(y_t, \cdot)\ \varphi_\sharp \nu_{t-1}$, **filtering distribution**
- Physical measure: $\mathrm{Unif}(x_0, x_1, \cdots, x_t) \xrightarrow{t \to \infty} \mu = \nu_0$ for Leb a.e. x_0.