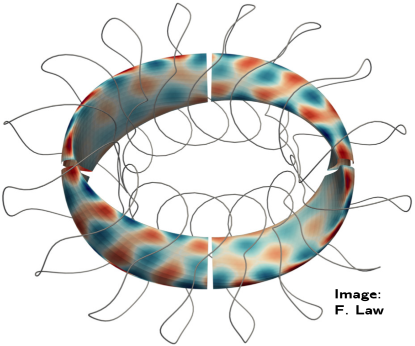
DICE: Discrete inverse continuity equation for learning population dynamics

Benjamin Peherstorfer
Courant Institute of Mathematical Sciences, New York University

with Tobias Blickhan (Courant), Jules Berman (Courant), and Andrew Stuart (Caltech)

June 2025

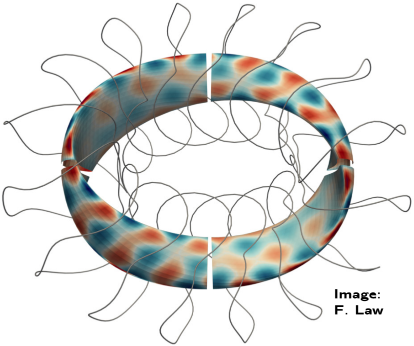
Leveraging reduced dynamics for ...



outer-loop applications

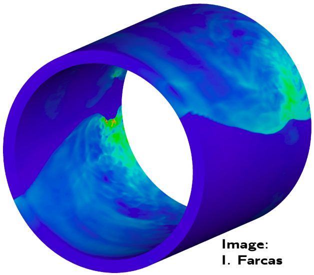
ensemble predictions, control,
optimal design, inverse problems

Leveraging reduced dynamics for ...



outer-loop applications

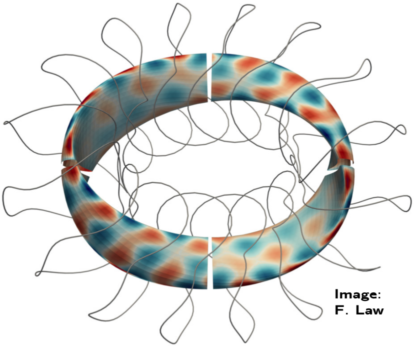
ensemble predictions, control,
optimal design, inverse problems



learning models from data

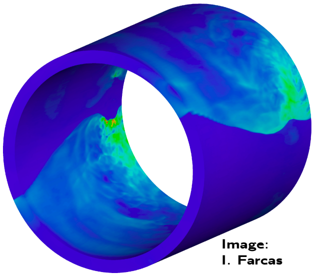
non-intrusive methods, operator learning, context-aware learning

Leveraging reduced dynamics for ...



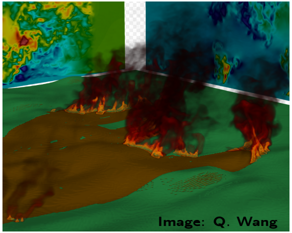
outer-loop applications

ensemble predictions, control,
optimal design, inverse problems



learning models from data

non-intrusive methods, operator learning, context-aware learning

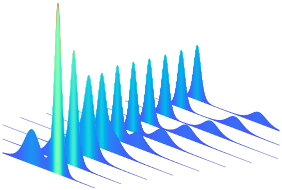


complex decision-making

interactions/coupling,
system-level predictions,
data assimilation

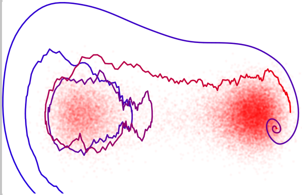
Challenges of reduced modeling

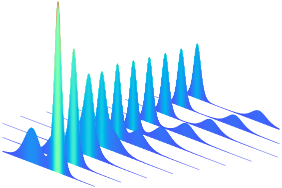
transport problems



Challenges of reduced modeling

transport problems

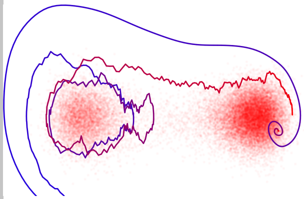




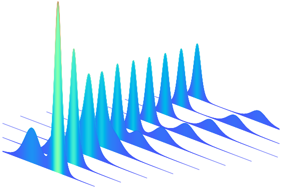
stochastic dynamics

Challenges of reduced modeling

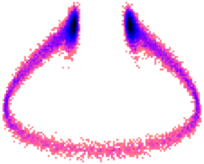
transport problems



chaotic systems

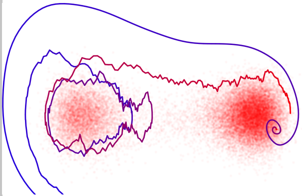


stochastic dynamics

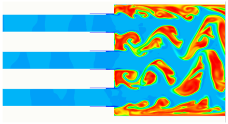


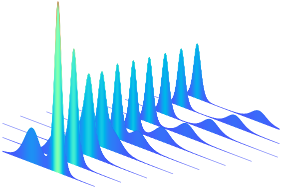
Challenges of reduced modeling

transport problems

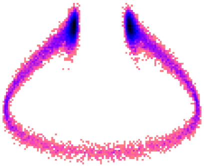


chaotic systems





stochastic dynamics



learning dynamics from data

(increase scope)

Outline

1. Population dynamics
   and generative modeling

2. From continuous to discrete
   loss functions

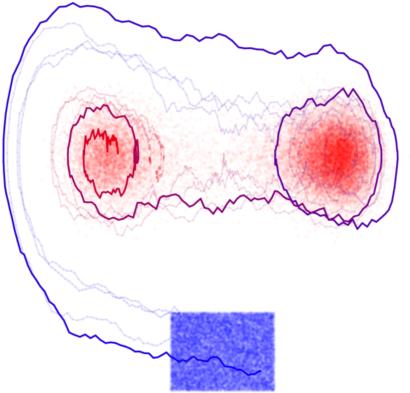
3. DICE: Inferring vector fields for
   learning stochastic systems

4. Numerical experiments

Outline

1. **Population dynamics and generative modeling**
2. From continuous to discrete loss functions
3. DICE: Inferring vector fields for learning stochastic systems
4. Numerical experiments

PD: Stochastic systems and sample trajectories



Stochastic process $X(t;\mu)$ over domain $\mathcal{X} \subseteq \mathbb{R}^d$

- Time $t \in \mathcal{T} \subset \mathbb{R}$
- Parameter $\mu \in \mathcal{D} \subseteq \mathbb{R}^{d'}$

Realization of sample trajectory for $\mu \in \mathcal{D}$

$$X_i(t_1;\mu), X_i(t_2;\mu), \dots, X_i(t_{n_t};\mu) \subset \mathcal{X}$$

Data in form of sample trajectories

$$\mathfrak{X} = \{X_i(t_k;\mu_j) \,|\, i = 1,\dots,n_x, \quad k = 1,\dots,n_t, \quad j = 1,\dots,n_\mu\} \subset \mathcal{X}$$

Goal is rapidly predicting behavior of stochastic process at new parameters μ

PD: Learning DEs from data

Fit right-hand side b of differential equations to data from process X

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t;\mu) = b(t, \boldsymbol{x}(t;\mu);\mu)$$

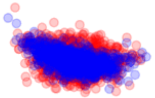
PD: Learning DEs from data

Fit right-hand side b of differential equations to data from process X

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t;\mu) = b(t, \boldsymbol{x}(t;\mu);\mu)$$

Example: Particle moving in potential with friction and stochastic forcing

- Potential
$$\phi(x) = \frac{1}{2}x^2 - \frac{\mu}{4}x^4$$
- Collect sample trajectories and fit $b_\theta : \mathcal{X} \to \mathcal{X}$ via mean-squared loss as is common in, e.g., dynamic mode decomposition, operator inference, operator learning **[Rowley et al., 2009], [Schmid, 2010], [Tu et al., 2014], [Williams et al., 2015], [P., Willcox, 2016], [Qian et al., 2020], [Lu et al., 2021], [Kovachki et al., 2023]**
- Training with mean-squared loss collapses learned models to conditional expectation



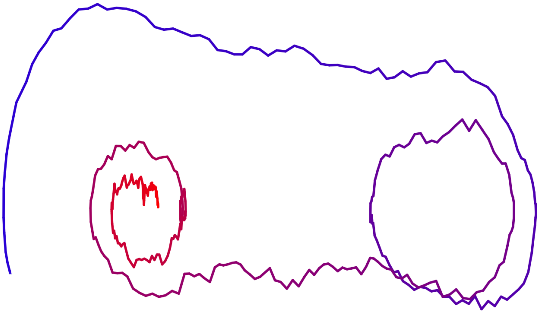
PD: Learning SDEs from data

Describe dynamics of X via stochastic differential equations

$$\mathrm{d}X(t;\mu) = b(t, X(t;\mu);\mu)\mathrm{d}t + \sigma(t;\mu)\mathrm{d}W_t$$

Learn drift b to match the sample trajectories

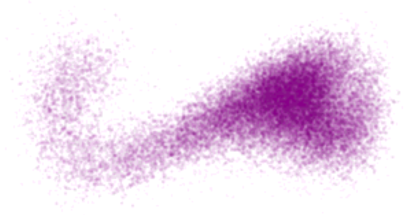
- Neural ordinary differential equations (NODEs) and other methods for stochastic systems **[Chen et al., 2018], [Dupont et al., 2019], [Li et al., 2020], [Kidger et al., 2021], [Salvi et al., 2022], [Chen, Xiu, 2024], ...**
- Model reduction when drift term is known **[Benner, Redmann, 2015], [Redmann, Freitag, 2018], [Freitag, Nicolaus, Redmann, 2024] ...**
- Parametric inference **[Kloeden, Platen, 1992], [Sorensen, 2009], ...** and (discrete) Markov process inference **[Murphy, 2012], ...**



PD: Population dynamics

Samples of $X(t;\mu)$ follow law $\rho(t,\cdot;\mu) : \mathcal{X} \to \mathbb{R}_{\geq 0}$ over time t and parameter μ

- $X(t,\mu)$ are samples from $\rho(t,\cdot;\mu)$
- Dynamics of $\rho(t,\cdot;\mu)$ over time t are the population dynamics



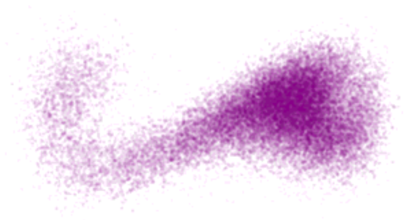
PD: Population dynamics

Samples of $X(t; \mu)$ follow law $\rho(t, \cdot; \mu) : \mathcal{X} \to \mathbb{R}_{\geq 0}$ over time t and parameter μ

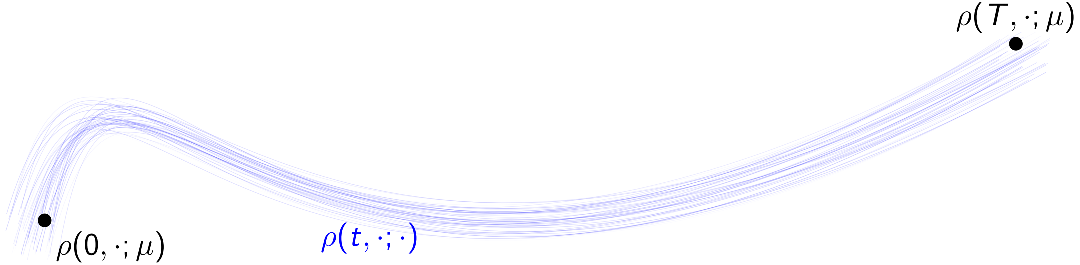
- $X(t, \mu)$ are samples from $\rho(t, \cdot; \mu)$
- Dynamics of $\rho(t, \cdot; \mu)$ over time t are the population dynamics

Sample versus population dynamics

- Sufficient for generating samples from ρ
- Fluid with constant density: Samples complicated, population dynamics constant
- Chaotic/turbulent systems with smooth population dynamics
- Have population dynamics for deterministic and stochastic systems



PD: Generative modeling with conditioning on time



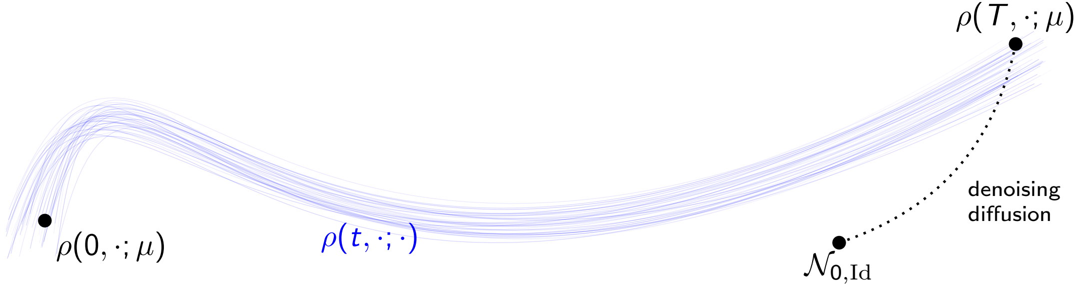
Standard generative modeling learns population dynamics via conditioning

- Denoising diffusion modeling: learn dynamics from Gaussian to target **[Sohl-Dickstein et al., 2015]**, **[Song et al, 2021]**, ...
- Flow-based modeling: learn dynamics between reference and target distribution **[Albergo et al., 2023]**, **[Lipman et al., 2023]**, ...

$\rightsquigarrow$ requires one costly inference step per physical time step

Other works often require parametrizing density or simulations **[Tong et al., 2020]**, **[Bunne et al., 2022]**

PD: Generative modeling with conditioning on time



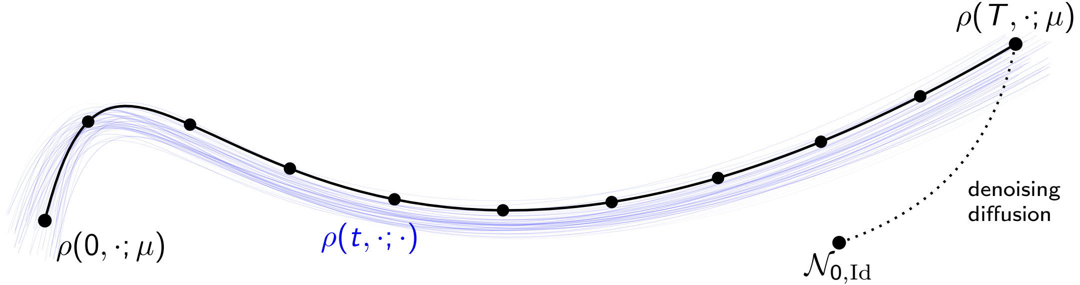
Standard generative modeling learns population dynamics via conditioning

- Denoising diffusion modeling: learn dynamics from Gaussian to target **[Sohl-Dickstein et al., 2015]**, **[Song et al, 2021]**, ...
- Flow-based modeling: learn dynamics between reference and target distribution **[Albergo et al., 2023]**, **[Lipman et al., 2023]**, ...

$\rightsquigarrow$ requires one costly inference step per physical time step

Other works often require parametrizing density or simulations **[Tong et al., 2020]**, **[Bunne et al., 2022]**

PD: Generative modeling with conditioning on time



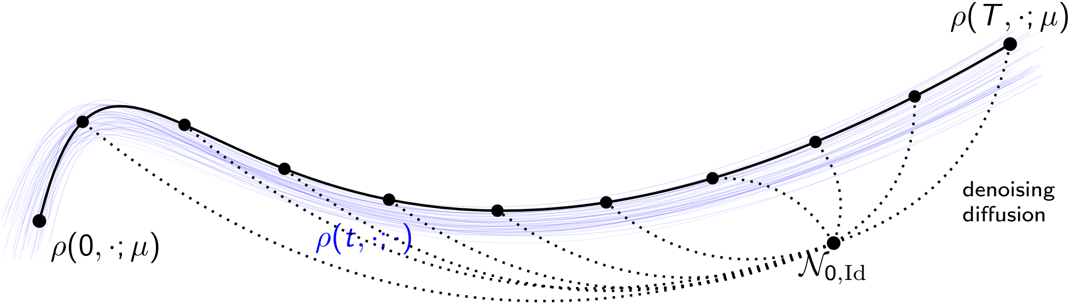
Standard generative modeling learns population dynamics via conditioning

- Denoising diffusion modeling: learn dynamics from Gaussian to target **[Sohl-Dickstein et al., 2015], [Song et al, 2021], ...**
- Flow-based modeling: learn dynamics between reference and target distribution **[Albergo et al., 2023], [Lipman et al., 2023], ...**

$\rightsquigarrow$ requires one costly inference step per physical time step

Other works often require parametrizing density or simulations **[Tong et al., 2020], [Bunne et al., 2022]**

PD: Generative modeling with conditioning on time



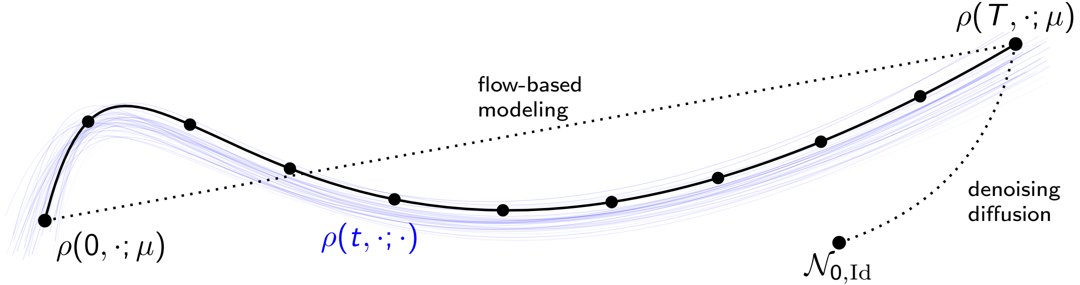
Standard generative modeling learns population dynamics via conditioning

- Denoising diffusion modeling: learn dynamics from Gaussian to target **[Sohl-Dickstein et al., 2015]**, **[Song et al, 2021]**, ...
- Flow-based modeling: learn dynamics between reference and target distribution **[Albergo et al., 2023]**, **[Lipman et al., 2023]**, ...

$\rightsquigarrow$ requires one costly inference step per physical time step

Other works often require parametrizing density or simulations **[Tong et al., 2020]**, **[Bunne et al., 2022]**

PD: Generative modeling with conditioning on time



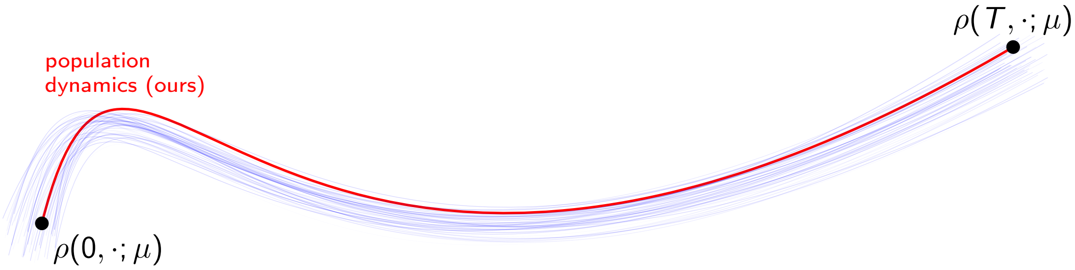
Standard generative modeling learns population dynamics via conditioning

- Denoising diffusion modeling: learn dynamics from Gaussian to target **[Sohl-Dickstein et al., 2015]**, **[Song et al, 2021]**, ...
- Flow-based modeling: learn dynamics between reference and target distribution **[Albergo et al., 2023]**, **[Lipman et al., 2023]**, ...

$\rightsquigarrow$ requires one costly inference step per physical time step

Other works often require parametrizing density or simulations **[Tong et al., 2020]**, **[Bunne et al., 2022]**

PD: Inferring population dynamics from sample trajectories



We aim to learn an approximation of the dynamics of ρ over time t

- Avoids conditioning on t for faster inference; one inference step gives one trajectory
- No need to learn the density function ρ, just its dynamics
- Builds on known loss functions but their empirical estimation has been challenging

[Berman, Blickhan, P., NeurIPS, 2024.], [Blickhan, Berman, Stuart, P., *upcoming*]

Outline

1. Population dynamics and generative modeling

2. From continuous to discrete loss functions

3. DICE: Inferring vector fields for learning stochastic systems

4. Numerical experiments

Outline

1. Population dynamics and generative modeling
2. **From continuous to discrete loss functions**
3. DICE: Inferring vector fields for learning stochastic systems
4. Numerical experiments

Cont: Continuity equation describes population dynamics

Consider the (unknown) governing SDE of $X(t)$ as

$$\mathrm{d}X(t) = b(t, X(t))\mathrm{d}t + \sigma(t)\mathrm{d}W_t$$

Cont: Continuity equation describes population dynamics

Consider the (unknown) governing SDE of $X(t)$ as

$$\mathrm{d}X(t) = b(t, X(t))\mathrm{d}t + \sigma(t)\mathrm{d}W_t$$

Continuity equation gives (population) dynamics of law $X(t) \sim \rho(t, \cdot)$

$$\partial_t \rho(t, x) = -\nabla \cdot (\rho(t, x)v(t, x)), \qquad \text{for all } x \in \mathcal{X}, t \in [0, T]$$

Cont: Continuity equation describes population dynamics

Consider the (unknown) governing SDE of $X(t)$ as

$$\mathrm{d}X(t) = b(t, X(t))\mathrm{d}t + \sigma(t)\mathrm{d}W_t$$

Continuity equation gives (population) dynamics of law $X(t) \sim \rho(t, \cdot)$

$$\partial_t \rho(t, x) = -\nabla \cdot (\rho(t, x) v(t, x)), \qquad \text{for all } x \in \mathcal{X}, t \in [0, T]$$

Vector field v of continuity equation can be derived from SDE as

$$v(t, x) = b(t, x) - \frac{\sigma(t)^2}{2} \nabla \log \rho(t, x)$$

Cont: Continuity equation describes population dynamics

Consider the (unknown) governing SDE of $X(t)$ as

$$\mathrm{d}X(t) = b(t, X(t))\mathrm{d}t + \sigma(t)\mathrm{d}W_t$$

Continuity equation gives (population) dynamics of law $X(t) \sim \rho(t, \cdot)$

$$\partial_t \rho(t, x) = -\nabla \cdot (\rho(t, x)v(t, x)), \qquad \text{for all } x \in \mathcal{X}, t \in [0, T]$$

Vector field v of continuity equation can be derived from SDE as

$$v(t, x) = b(t, x) - \frac{\sigma(t)^2}{2}\nabla \log \rho(t, x)$$

But there are many vector fields v that lead to the same population dynamics

- E.g., adding divergence-free field w/ρ with $\nabla \cdot w = 0$ leaves dynamics of ρ unchanged
- Different to the SDE, where changing the drift/diffusion lead to other sample dynamics

Cont: Vector field for sample generation

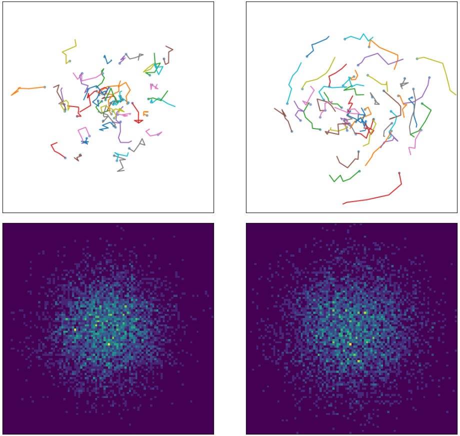
Given a vector field $\hat{v}$ that is compatible so that the population dynamics are given by

$$\partial_t \rho(t,x) = -\nabla \cdot (\rho(t,x)\hat{v}(t,x))$$

Generate new samples $\hat{X}(t,x) \sim \rho(t,\cdot)$ using ODE/SDE formulation

$$\mathrm{d}\hat{X}(t) = \hat{v}(t,\hat{X}(t))\mathrm{d}t$$

- New samples $\hat{X}(t)$ follow the same law $\rho(t,\cdot)$
- But sample trajectories can be starkly different to sample trajectories of original SDE with drift b



Cont: Vector field for sample generation

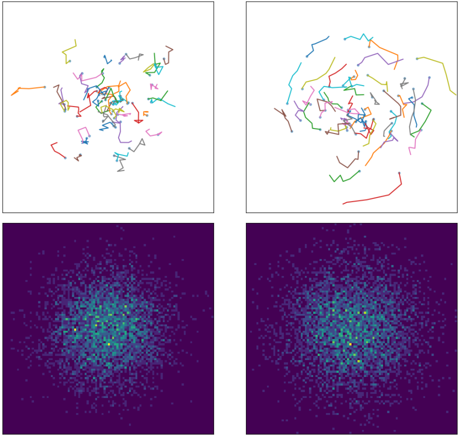
Given a vector field $\hat{v}$ that is compatible so that the population dynamics are given by

$$\partial_t \rho(t,x) = -\nabla \cdot (\rho(t,x)\hat{v}(t,x))$$

Generate new samples $\hat{X}(t,x) \sim \rho(t,\cdot)$ using ODE/SDE formulation

$$\mathrm{d}\hat{X}(t) = \hat{v}(t,\hat{X}(t))\mathrm{d}t + \sigma(t)\mathrm{d}W_t$$

- New samples $\hat{X}(t)$ follow the same law $\rho(t,\cdot)$
- But sample trajectories can be starkly different to sample trajectories of original SDE with drift b

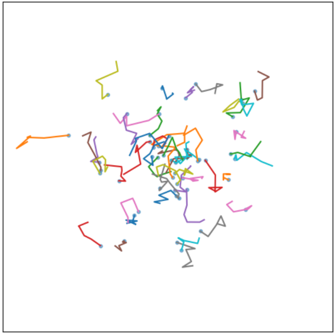


Cont: Vector fields with minimal kinetic energy

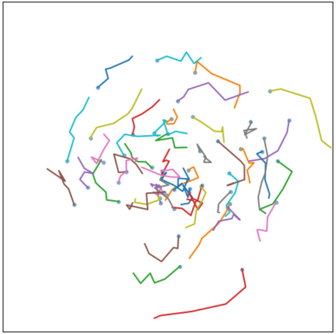
Aim to find the vector field that minimizes the kinetic energy

$$E_t(v) = \mathbb{E}_{x\sim\rho(t,\cdot)}\left[\frac{1}{2}|v(t,x)|^2\right]$$

Minimizes the average energy ("movement") of samples



minimal kinetic energy



other vector field

Cont: Dynamic transport

Optimization problem **[Benamou, Brenier, 2000]**

$$\min_{v} \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)} \left[\frac{1}{2}|v|^2\right] \mathrm{d}t \qquad \text{such that} \qquad \partial_t \rho + \nabla \cdot (\rho v) = 0$$

Cont: Dynamic transport

Optimization problem [Benamou, Brenier, 2000]

$$\min_v \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)}\left[\frac{1}{2}|v|^2\right]\mathrm{d}t \qquad \text{such that} \quad \partial_t\rho + \nabla\cdot(\rho v) = 0$$

Formulation with Lagrange multiplier shows that optimum has to be gradient field

$$v = \nabla s$$

Cont: Dynamic transport

Optimization problem [Benamou, Brenier, 2000]

$$\min_v \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)}\left[\frac{1}{2}|v|^2\right]\mathrm{d}t \qquad \text{such that} \qquad \partial_t\rho + \nabla\cdot(\rho v) = 0$$

Formulation with Lagrange multiplier shows that optimum has to be gradient field

$$v = \nabla s$$

Sufficient to seek gradient field $s : \mathcal{T}\times\mathcal{X}\to\mathbb{R}$ with $v = \nabla s$ that solves

$$\min_s \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)}\left[\frac{1}{2}|\nabla s|^2\right] - \int_{\mathcal{X}} \partial_t\rho\, s\,\mathrm{d}x\,\mathrm{d}t$$

- Not a loss because depends on $\partial_t\rho$, which is unavailable if only sample trajectories given
- Trying to directly turn this objective into a loss (action matching [Neklyudov et al., 2023]) can lead to unstable loss functions [Blickhan, Berman, P., 2024]

Cont: Deriving loss via integration by parts

Recall: Sufficient to seek gradient field $s : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ with $v = \nabla s$ that solves

$$\min_s \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)} \left[\frac{1}{2}|\nabla s|^2\right] - \int_{\mathcal{X}} \partial_t \rho \, s \, \mathrm{d}x \, \mathrm{d}t$$

Cont: Deriving loss via integration by parts

Recall: Sufficient to seek gradient field $s : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ with $v = \nabla s$ that solves

$$\min_s \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)} \left[\frac{1}{2}|\nabla s|^2\right] - \int_{\mathcal{X}} \partial_t \rho \, s \, \mathrm{d}x \, \mathrm{d}t$$

Integration by parts leads to the action-matching loss [Neklyudov et al., 2023]

$$\mathrm{L}_{\mathrm{AM}}(\hat{s}) = \int_0^T \mathbb{E}_{x\sim\rho(t,\cdot)} \left[\frac{1}{2}|\nabla \hat{s}(t,x)|^2 + \partial_t s(t,x)\right] \mathrm{d}t - \mathbb{E}_{x\sim\rho(t,\cdot)}[\hat{s}(t,x)]\Big|_{t=0}^{t=T}$$

- Moved time derivative from ρ to s
- This is a loss function because all terms can be estimated from samples

Cont: Deriving loss via integration by parts

Recall: Sufficient to seek gradient field $s : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ with $v = \nabla s$ that solves

$$\min_s \int_0^T \mathbb{E}_{x \sim \rho(t,\cdot)} \left[\frac{1}{2} |\nabla s|^2 \right] - \int_{\mathcal{X}} \partial_t \rho \, s \, \mathrm{d}x \, \mathrm{d}t$$

Integration by parts leads to the action-matching loss [Neklyudov et al., 2023]

$$\mathrm{L}_{\mathrm{AM}}(\hat{s}) = \int_0^T \mathbb{E}_{x \sim \rho(t,\cdot)} \left[\frac{1}{2} |\nabla \hat{s}(t,x)|^2 + \partial_t s(t,x) \right] \mathrm{d}t - \mathbb{E}_{x \sim \rho(t,\cdot)}[\hat{s}(t,x)] \Big|_{t=0}^{t=T}$$

- Moved time derivative from ρ to s
- This is a loss function because all terms can be estimated from samples

Controls the error

$$\int_0^T \frac{1}{2} \|\nabla \hat{s}(t,\cdot) - \nabla s(t,\cdot)\|_{L^2(\rho(t))} \mathrm{d}t = \mathrm{L}_{\mathrm{AM}}(\hat{s}) + c(s)$$

Various loss functions of this type have been derived in the literature [Neklyudov et al., 2023], [Lavenant et al., 2024], [Berman, Blickhan, P., 2024]; also [Otto, Villani, 2000], [Reich, 2010]

Cont: Discretization matters

Discretize in time (naively):

$$\hat{\mathrm{L}}_{\mathrm{AM}}(\hat{s}) = \sum_{i=1}^{n_t} w_i \mathbb{E}_{x\sim\rho(t_i,\cdot)}\left[\frac{1}{2}|\nabla\hat{s}(t_i,x)|^2 + \partial_t\hat{s}(t_i,x)\right] - \mathbb{E}_{x\sim\rho(t,\cdot)}\left[\hat{s}(t,x)\right]\Big|_{t=0}^{t=T}$$

- Quadrature rule given by weights w_i and time points t_i for $i = 1, \ldots, n_t$
- Typically, just Monte Carlo with $w_i = T/n_t$ and t_i uniform in $[0, T]$ used **[Neklyudov et al., 2023]**

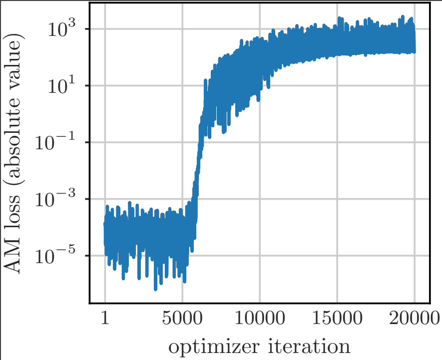
Discrete empirical AM loss violates key invariance to space-constant functions $f : [0, T] \to \mathbb{R}$

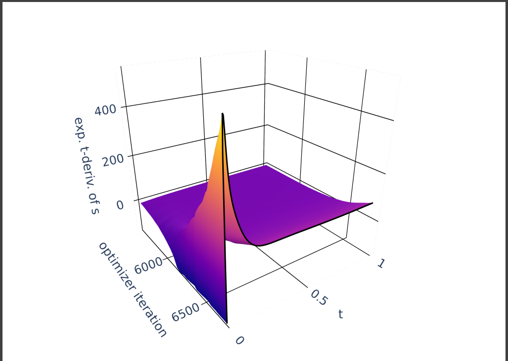
$$\hat{\mathrm{L}}_{\mathrm{AM}}(\hat{s} + f) = \hat{\mathrm{L}}_{\mathrm{AM}}(\hat{s}) + \underbrace{\frac{T}{n_t}\sum_i \partial_t f(t_i) - f(T) + f(0)}_{\text{residual term due to time discretization}}$$

Unstable training: The residual term can grow and change during training so that $(t,x) \mapsto \hat{s}(t,x) + f(t)$ becomes arbitrarily rough, which amplifies the residual term again

[Berman, Blickhan, P., Parametric model reduction of mean-field and stochastic systems via higher-order action matching, NeurIPS 2024.]

Cont: Experiment on toy example





Set $X(t) \sim \mathcal{N}(0, 10^{-2})$ so that $\rho(t)$ remains constant over time t

- At about 5000 iterations, the action matching loss starts to explode
- Optimizer found an f with sharp kink in $t \mapsto \mathbb{E}[\partial_t(s + f)] \rightsquigarrow$ large time integration error

Outline

1. Population dynamics
   and generative modeling

2. From continuous to discrete
   loss functions

3. DICE: Inferring vector fields for
   learning stochastic systems

4. Numerical experiments

Outline

1. Population dynamics and generative modeling
2. From continuous to discrete loss functions
3. **DICE: Inferring vector fields for learning stochastic systems**
4. Numerical experiments

DICE: Discretize time first: Weak form in discrete time

We only have data at discrete time points $0 = t_0 < t_1 < \cdots < t_K = T$

$$X_i(t_0) \sim \rho(t_0), \ldots, X_i(t_K) \sim \rho(t_K)\,, \qquad i = 1, \ldots, n_x$$

DICE: Discretize time first: Weak form in discrete time

We only have data at discrete time points $0 = t_0 < t_1 < \cdots < t_K = T$

$$X_i(t_0) \sim \rho(t_0), \ldots, X_i(t_K) \sim \rho(t_K)\,, \qquad i = 1, \ldots, n_x$$

Consider the weak form of the continuity equation at the discrete time points

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{x\sim\rho(t_j)}[\varphi(x)] = \mathbb{E}_{x\sim\rho(t_j)}[\nabla s(t_j, x) \cdot \nabla\varphi(x)]\,, \qquad \forall\varphi\,, \quad j = 0, \ldots, K$$

DICE: Discretize time first: Weak form in discrete time

We only have data at discrete time points $0 = t_0 < t_1 < \cdots < t_K = T$

$$X_i(t_0) \sim \rho(t_0), \ldots, X_i(t_K) \sim \rho(t_K)\,, \qquad i = 1, \ldots, n_x$$

Consider the weak form of the continuity equation at the discrete time points

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{x\sim\rho(t_j)}[\varphi(x)] = \mathbb{E}_{x\sim\rho(t_j)}[\nabla s(t_j, x) \cdot \nabla\varphi(x)]\,, \qquad \forall\varphi\,, \quad j = 0, \ldots, K$$

Formulate discrete-time weak form with central finite-difference approximations $\hat{\delta}_j$

$$\mathbb{E}[\varphi(x)\hat{\delta}_j\rho(t_j)] = \mathbb{E}_{x\sim\rho(t_j)}[\nabla\hat{s}(t_j, x) \cdot \nabla\varphi(x)]\,, \qquad \forall\varphi\,, \quad j = 0, \ldots, K$$

DICE: Discretize time first: Weak form in discrete time

We only have data at discrete time points $0 = t_0 < t_1 < \cdots < t_K = T$

$$X_i(t_0) \sim \rho(t_0), \ldots, X_i(t_K) \sim \rho(t_K)\,, \qquad i = 1, \ldots, n_x$$

Consider the weak form of the continuity equation at the discrete time points

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}_{x\sim\rho(t_j)}[\varphi(x)] = \mathbb{E}_{x\sim\rho(t_j)}[\nabla s(t_j, x) \cdot \nabla\varphi(x)]\,, \qquad \forall\varphi\,, \quad j = 0, \ldots, K$$

Formulate discrete-time weak form with central finite-difference approximations $\hat{\delta}_j$

$$\mathbb{E}[\varphi(x)\hat{\delta}_j\rho(t_j)] = \mathbb{E}_{x\sim\rho(t_j)}[\nabla \hat{s}(t_j, x) \cdot \nabla\varphi(x)]\,, \qquad \forall\varphi\,, \quad j = 0, \ldots, K$$

Goal is now finding a loss that has discrete-time weak form as Euler-Lagrange equations

[Blickhan, Berman, Stuart, P., DICE: Inference with the discrete inverse continuity equation, *upcoming*]

DICE: Loss with minimizers satisfying discrete weak form

Define sequence of problems that are coupled via source term

$$\min_{\hat{s}_j \in \mathcal{S}_j} \quad \|\nabla \hat{s}_j\|_{L^2(\rho(t_j))} - \langle \hat{\delta}_j \rho(t_j), \hat{s}_j \rangle_{L^2(\mathrm{d}x)}\,, \qquad j = 0, \dots, K$$

over spaces $\mathcal{S}_j = \{s \in L^2(\rho(t_j)) \,|\, \nabla s \in L^2(\rho(t_j))\}$

DICE: Loss with minimizers satisfying discrete weak form

Define sequence of problems that are coupled via source term

$$\min_{\hat{s}_j \in \mathcal{S}_j} \quad \|\nabla \hat{s}_j\|_{L^2(\rho(t_j))} - \langle \hat{\delta}_j \rho(t_j), \hat{s}_j \rangle_{L^2(\mathrm{d}x)}\,, \qquad j = 0, \dots, K$$

over spaces $\mathcal{S}_j = \{s \in L^2(\rho(t_j)) \,|\, \nabla s \in L^2(\rho(t_j))\}$

Lead to loss $E^K : \mathcal{S}_0 \times \cdots \times \mathcal{S}_K \to \mathbb{R}$ with

$$E^K(\bar{s}) = \|\nabla \bar{s}\|_{L^2(\rho(t_0)\cdots\rho(t_K))} - \langle \bar{\rho}, \bar{s} \rangle_{L^2(\mathrm{d}x\cdots\mathrm{d}x)}$$

- Vector of functions $\bar{s} = [\hat{s}_0, \dots, \hat{s}_K] \in \mathcal{S}_0 \times \cdots \times \mathcal{S}_K$
- Inner product

$$\langle \bar{s}, \bar{w} \rangle_{L^2(\rho(t_0)\cdots\rho(t_K))} = \sum_{j=0}^{K} \int_{\mathcal{X}} \frac{t_{j+1} - t_{j-1}}{2} \hat{s}_j(x) \hat{w}_j(x) \rho(t_j) \mathrm{d}x$$

DICE: Loss with minimizers satisfying discrete weak form

Define sequence of problems that are coupled via source term

$$\min_{\hat{s}_j \in \mathcal{S}_j} \quad \|\nabla \hat{s}_j\|_{L^2(\rho(t_j))} - \langle \hat{\delta}_j \rho(t_j), \hat{s}_j \rangle_{L^2(\mathrm{d}x)}\,, \qquad j = 0, \dots, K$$

over spaces $\mathcal{S}_j = \{s \in L^2(\rho(t_j)) \,|\, \nabla s \in L^2(\rho(t_j))\}$

Lead to loss $E^K : \mathcal{S}_0 \times \cdots \times \mathcal{S}_K \to \mathbb{R}$ with

$$E^K(\bar{s}) = \|\nabla \bar{s}\|_{L^2(\rho(t_0)\cdots\rho(t_K))} - \langle \bar{\rho}, \bar{s} \rangle_{L^2(\mathrm{d}x\cdots\mathrm{d}x)}$$

- Vector of functions $\bar{s} = [\hat{s}_0, \dots, \hat{s}_K] \in \mathcal{S}_0 \times \cdots \times \mathcal{S}_K$
- Inner product

$$\langle \bar{s}, \bar{w} \rangle_{L^2(\rho(t_0)\cdots\rho(t_K))} = \sum_{j=0}^{K} \int_{\mathcal{X}} \frac{t_{j+1} - t_{j-1}}{2} \hat{s}_j(x) \hat{w}_j(x) \rho(t_j) \mathrm{d}x$$

Minimizers $\bar{s}^*$ of E^K solve the time-discrete weak form of the continuity equation

[Blickhan, Berman, Stuart, P., DICE: Inference with the discrete inverse continuity equation, *upcoming*]

DICE: The DICE loss function

Consider extension of $\mathcal{S}_0 \times \cdots \times \mathcal{S}_K$ over time interval $[0, T]$ as

$$\mathcal{S} = \{s : [0, T] \times \mathcal{X} \to \mathbb{R} \,|\, s(t_j, \cdot) \in \mathcal{S}_j \text{ for } j = 0, \dots, K\}$$

A function $\hat{s}^* \in \mathcal{S}$ that minimizes the discrete inverse continuity equation (DICE) loss

$$\begin{aligned}\mathrm{L}_{\mathrm{DICE}}(\hat{s}) = \sum_{j=1}^{K} \Bigg(\frac{t_j - t_{j-1}}{2} \left(\mathbb{E}_{x\sim\rho(t_j)} \left[\frac{1}{2}|\nabla \hat{s}(t_j, x)|^2\right] + \mathbb{E}_{x\sim\rho(t_{j-1})} \left[\frac{1}{2}|\nabla \hat{s}(t_{j-1}, x)|^2\right]\right) \\ - \frac{1}{2} \left(\mathbb{E}_{x\sim\rho(t_j)} [\hat{s}(t_j, x) + \hat{s}(t_{j-1}, x)] - \mathbb{E}_{x\sim\rho(t_{j-1})} [\hat{s}(t_j, x) + \hat{s}(t_{j-1}, x)]\right) \Bigg)\end{aligned}$$

is also a solution of the time-discrete weak form of the continuity equation

- The function $\mathrm{L}_{\mathrm{DICE}}$ is a loss because all terms can be estimated from data (sample traj.)

$$X_i(t_0) \sim \rho(t_0), \cdots, X_i(t_K) \sim \rho(t_K)\,, \qquad i = 1, \dots, n_x$$

- No need to evaluate $\rho(t_0), \dots, \rho(t_K)$; only require samples $X_i(t_j) \sim \rho(t_j)$ for $j = 0, \dots, K$

[Blickhan, Berman, Stuart, P., DICE: Inference with the discrete inverse continuity equation, *upcoming*]

DICE: Properties of the DICE loss

There exists a unique minimizer of $\mathrm{L}_{\mathrm{DICE}}$ in $\mathcal{S}$ in the sense that for two minimizers $\hat{s}^*, \hat{s}^{**}$ have

$$\|\nabla \hat{s}^*(t_j, \cdot) - \nabla \hat{s}^{**}(t_j, \cdot)\|_{L^2(\rho(t_j))} = 0\,, \qquad j = 0, \dots, K$$

- with key assumptions that $\rho(t, \cdot)$ exists and is absolutely continuous w.r.t. Lebesgue measure,
- the spaces $L^2(\rho(t))$ admit a Poincaré inequality

Under the same assumptions as above, the DICE loss is lower bounded

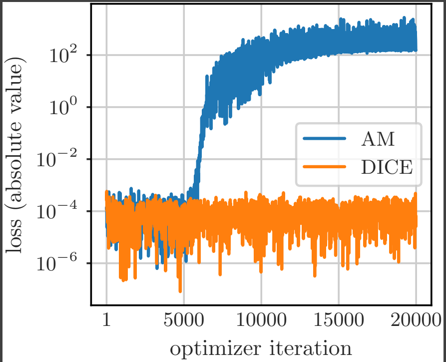
$$\mathrm{L}_{\mathrm{DICE}}(s) \geq C > -\infty\,, \qquad s \in \mathcal{S}$$

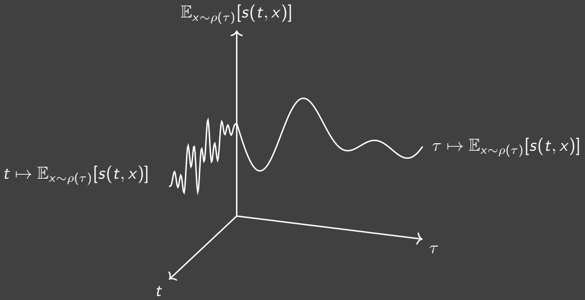
The DICE loss is invariant with respect to functions $f : [0, T] \to \mathbb{R}$ that are constant in space

$$\mathrm{L}_{\mathrm{DICE}}(s + f) = \mathrm{L}_{\mathrm{DICE}}(s)\,, \qquad s \in \mathcal{S}$$

[Blickhan, Berman, Stuart, P., DICE: Inference with the discrete inverse continuity equation, *upcoming*]

DICE: Experiment on toy example (cont'd)





Set $X(t) \sim \mathcal{N}(0, 10^{-2})$ so that $\rho(t)$ remains constant over time t

- DICE loss is invariant to spurious constants
- Training with DICE loss remains stable over many iterations

DICE: Continuous time limit

Assume abs. cont. densities $\rho(t,\cdot)$ and boundedness of $\rho(0,\cdot)$

$$0 < \underline{\rho}_0 \leq \rho(0,x) \leq \overline{\rho}_0\,, \qquad x \in \mathcal{X}$$

If $(\rho, \nabla s)$ are compatible and $\hat{s}$ is a minimizer of $\mathrm{L}_{\mathrm{DICE}}$ with respect to $\{\rho(t_j)\}_{j=0}^{K}$, then

$$\|\nabla s(t,\cdot) - \nabla \hat{s}(t,\cdot)\|_{L^2(\rho(t))} \leq C \max_{i=1,\dots,K} |t_{i-1} - t_i|$$

- Constant C depends on time derivatives of ρ and the Lipschitzness of $\hat{s}$
- Can take limit $K \to \infty$ so that $\max_i |t_{i-1} - t_i| \to 0$ and thus

$$\|\nabla s(t,\cdot) - \nabla \hat{s}(t,\cdot)\|_{L^2(\rho(t))} \to 0\,, \qquad t \in [0, T]$$

We control the error with respect to ∇s via a time-discrete (empirical) loss

[Blickhan, Berman, Stuart, P., DICE: Inference with the discrete inverse continuity equation, *upcoming*]

DICE: Generalizing over parameters μ

Recall that the stochastic process $X(t;\mu)$ was depending on $\mu \in \mathcal{D}$ with

$$\partial_t \rho(t,x;\mu) = -\nabla \cdot (\rho(t,x;\mu) v(t,x;\mu))$$

Correspondingly have data samples

$$\mathfrak{X} = \{X_i(t_k;\mu_j) \,|\, i = 1,\dots,n_x, \quad k = 1,\dots,n_t, \quad j = 1,\dots,n_\mu\} \subset \mathcal{X}$$

Optimize for $\hat{s} : \mathcal{T} \times \mathcal{X} \times \mathcal{D} \to \mathbb{R}$ with the loss

$$\min_{\hat{s}} \frac{1}{n_\mu} \sum_{i=1}^{n_\mu} \mathrm{L}_{\mathrm{DICE}}(\hat{s}(\cdot,\cdot;\mu_i))$$

[Blickhan, Berman, Stuart, P., DICE: Inference with the discrete inverse continuity equation, *upcoming*]

DICE: Predictions with learned gradient field

Offline phase: Learning gradient field $\hat{s}$ and obtaining approximate population dynamics

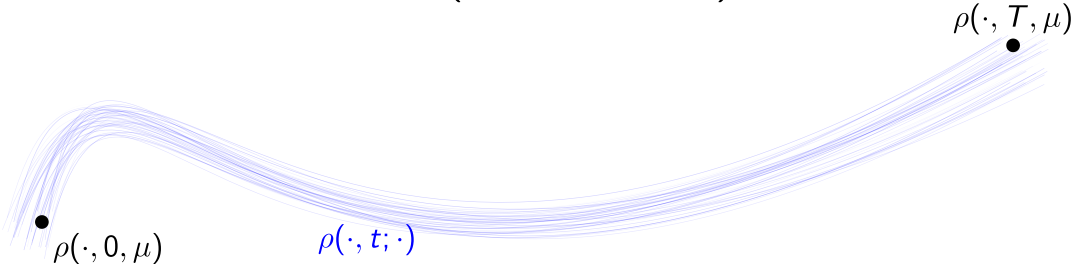
$$\partial_t \hat{\rho}(t, x; \mu) = -\nabla \cdot (\hat{\rho}(t, x; \mu) \nabla \hat{s}(t, x; \mu))$$

Online/Inference phase: Sampling from corresponding process $\hat{X}(t; \mu) \sim \hat{\rho}(t, \cdot; \mu)$ via (S)DE

$$\frac{\mathrm{d}}{\mathrm{d}t} \hat{X}(t; \mu) = \nabla \hat{s}(t, \hat{X}(t; \mu); \mu)$$

- Initial condition $\hat{X}(0; \mu) \sim \rho(0, \cdot; \mu)$
- Sampling time and physical time of process $\hat{X}$ are the same
- Obtain SDE if we use energy with entropy term

DICE: Rapid predictions (inference step)

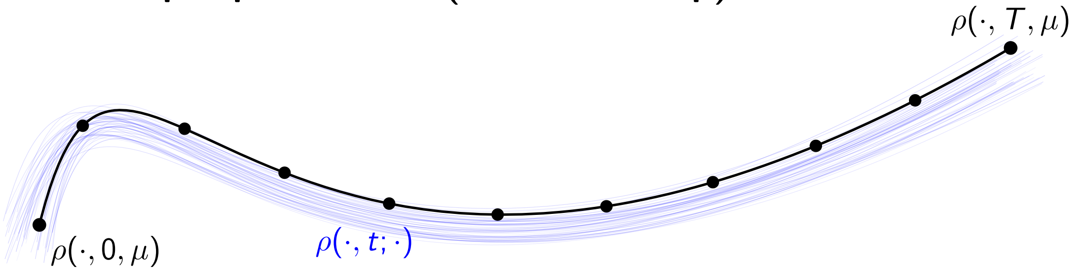


In the inference step of our approach, the sampling time is the physical time

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{X}(t;\mu) = \nabla \hat{s}(\hat{X}(t;\mu), t; \mu)\,, \qquad \hat{X}(0;\mu) \sim \rho(0;\mu)$$

- One inference step provides a whole time trajectory
- In stark contrast, conditional diffusion- and flow-based modeling require one inference step per time step

DICE: Rapid predictions (inference step)

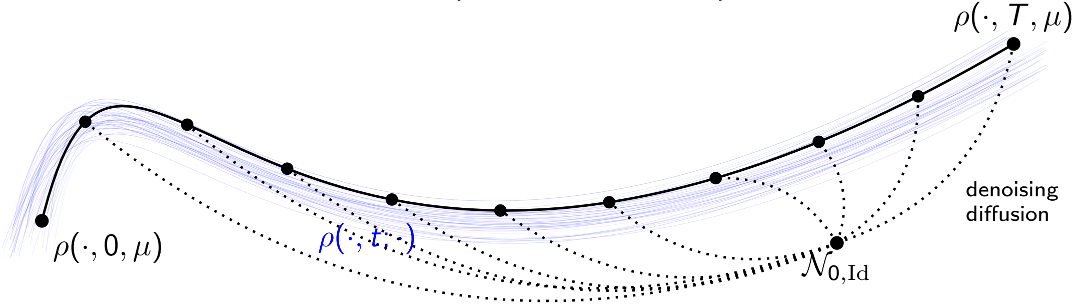


In the inference step of our approach, the sampling time is the physical time

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{X}(t;\mu) = \nabla \hat{s}(\hat{X}(t;\mu), t; \mu)\,, \qquad \hat{X}(0;\mu) \sim \rho(0;\mu)$$

- One inference step provides a whole time trajectory
- In stark contrast, conditional diffusion- and flow-based modeling require one inference step per time step

DICE: Rapid predictions (inference step)

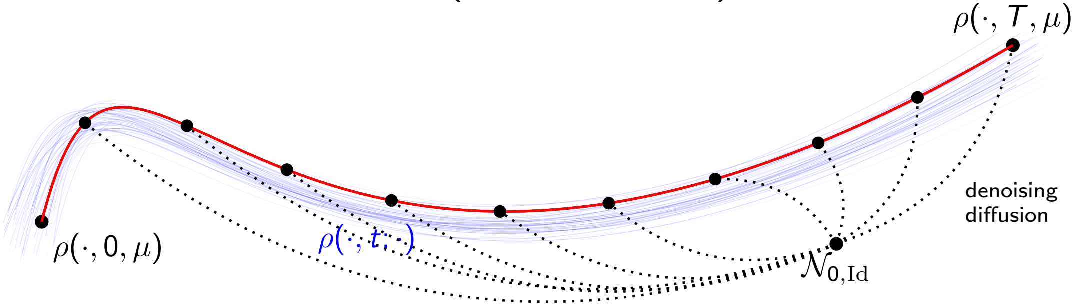


In the inference step of our approach, the sampling time is the physical time

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{X}(t;\mu) = \nabla\hat{s}(\hat{X}(t;\mu), t; \mu)\,, \qquad \hat{X}(0;\mu) \sim \rho(0;\mu)$$

- One inference step provides a whole time trajectory
- In stark contrast, conditional diffusion- and flow-based modeling require one inference step per time step

DICE: Rapid predictions (inference step)



In the inference step of our approach, the sampling time is the physical time

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{X}(t;\mu) = \nabla \hat{s}(\hat{X}(t;\mu), t;\mu)\,, \qquad \hat{X}(0;\mu) \sim \rho(0;\mu)$$

- One inference step provides a whole time trajectory
- In stark contrast, conditional diffusion- and flow-based modeling require one inference step per time step

Outline

1. Population dynamics and generative modeling
2. From continuous to discrete loss functions
3. DICE: Inferring vector fields for learning stochastic systems
4. Numerical experiments

Outline

1. Population dynamics
   and generative modeling

2. From continuous to discrete
   loss functions

3. DICE: Inferring vector fields for
   learning stochastic systems

4. **Numerical experiments**

Experiments: Oscillator example

Particle in potential ϕ with stochastic forcing

- Potential function with $\mu = 1/5$ fixed

$$\phi(x) = \frac{1}{2}x^2 - \frac{\mu}{4}x^4$$

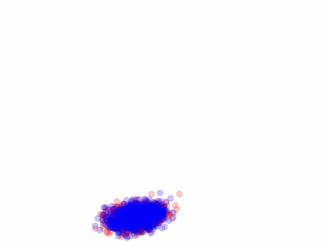
- Stochastic forcing term

$$\sigma(t;\mu) = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$$

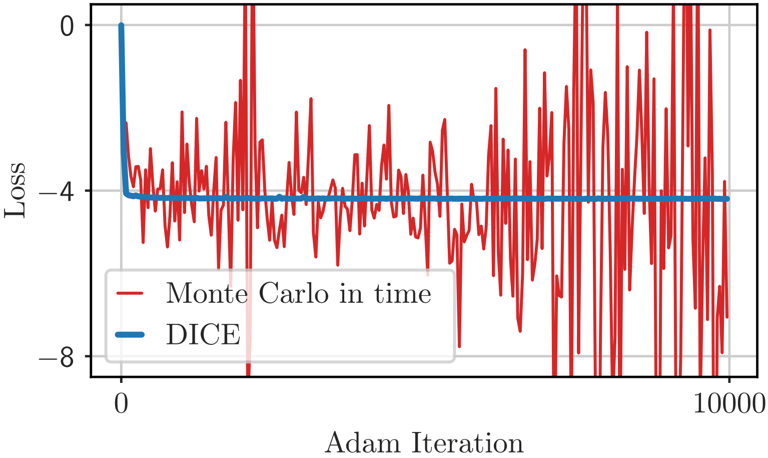
- Collect $n_x = 1400$ samples over $t \in [0, 14]$
- Initial condition

$$X(0) \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ -10 \end{bmatrix}, \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \right)$$

We match population dynamics rather than just mean behavior as deterministic methods

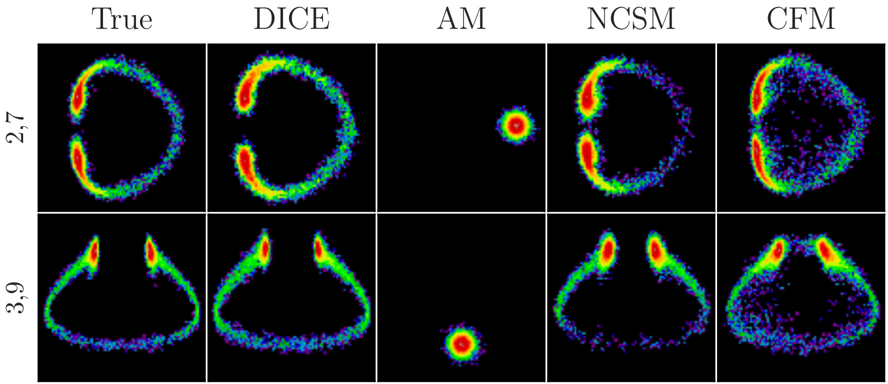


Experiments: Estimation of loss during optimization



- Instabilities introduced in training when estimating AM loss with Monte Carlo in time
- Proper time discretization as in the DICE loss stabilizes training behavior

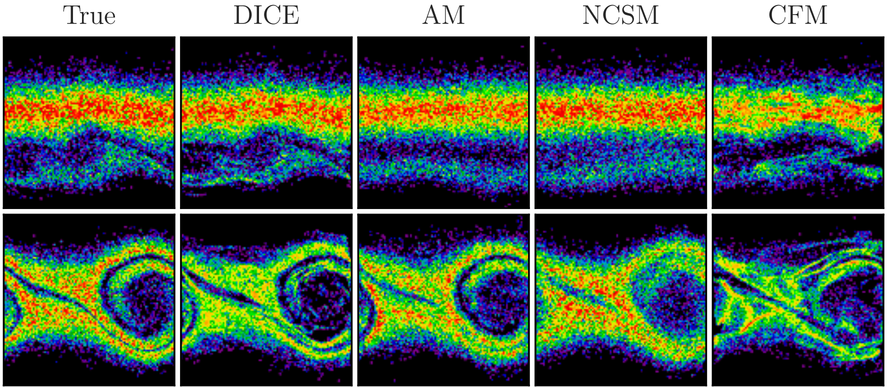
Experiments: Extended chaotic Lorenz system in 9D



- Sample trajectories from extended Lorenz system in 9 dimensions [Reiterer et al., 1998]
- Learn gradient field $\hat{s}$ and generate new samples for reduced Rayleigh number
- Our approach captures fine details well

CFM: [Albergo et al., 2023] [Lipman et al., 2023], NCSM: [Song et al., 2019], AM: [Neklyudov et al., 2023]

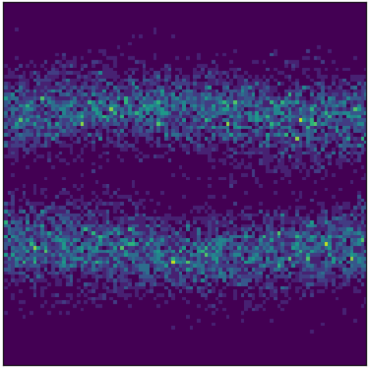
Experiments: Particle instabilities



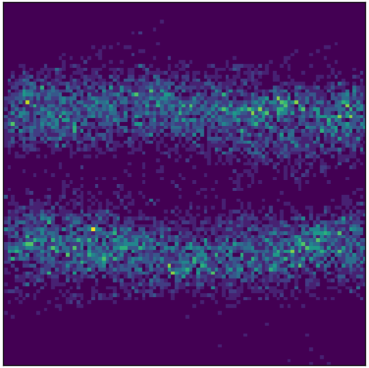
- Instabilities governed by Vlasov-Poisson equation [Tyranowsk, 2021]
- Data generated with particle-in-cell methods
- Parameter μ controls wave length

Experiments: Our gradient fields minimize kinetic energy

full (numerical) model

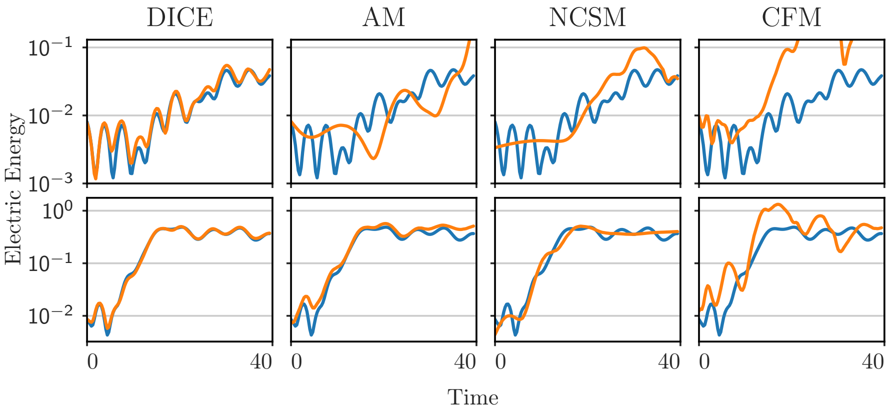


ours



- Population induced by learned field matches well the population from full model
- Learned field minimizes kinetic energy and so particles move less in learned dynamics

Experiments: Electric energy



- Quantity of interest is electric energy over time at wave lengths μ
- Predictions with our approach capture transient regime well

Experiments: Inference time

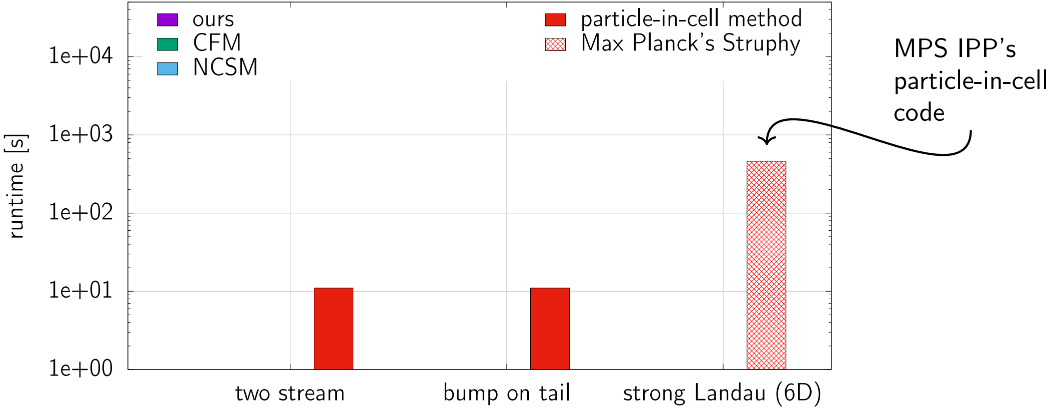
example:	**two-stream**		**bump-on-tail**		**strong Landau**		**9D chaos**	
metric:	error	r.t. [s]	error	r.t. [s]	error	r.t. [s]	sinkhorn	r.t. [s]
CFM	5.52	141	1.44	139	0.629	161	0.259	36
NCSM	0.626	1133	0.245	1142	4.06	4531	0.869	1109
AM	0.892	6	0.275	6	NaN	-	80.1	7
DICE (ours)	**0.283**	6	**0.070**	6	**0.463**	7	**0.200**	7

Our approach achieves orders of magnitude lower inference times compared to diffusion- (NCSM) and flow-based (CFM) modeling

- Sampling time is physical time: One sample trajectory is obtained in one inference step
- DICE's proper time discretization is key for generalization over μ as AM fails to be predictive

CFM: [Albergo et al., 2023] [Lipman et al., 2023], NCSM: [Song et al., 2019], AM: [Neklyudov et al., 2023]

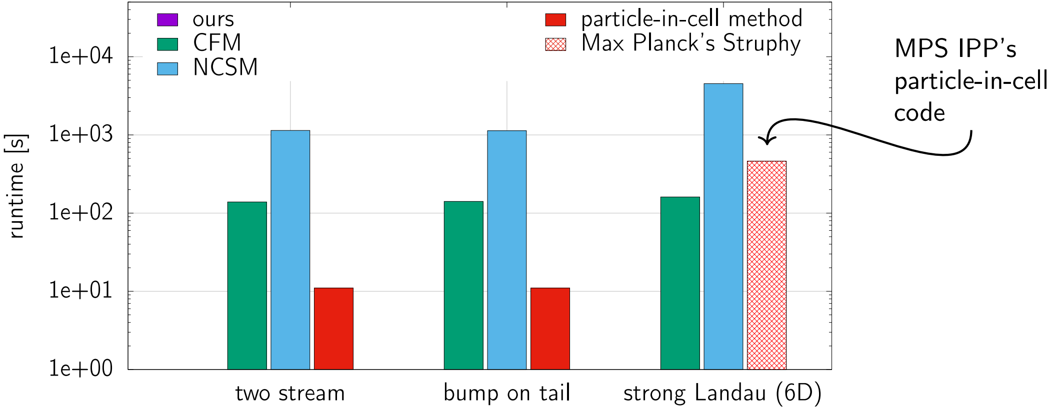
Experiments: Speedups over particle-in-cell methods



Speedups over numerical models

- Critically, we learn reduced model from data; no need to re-implement codes
- Speedups of more than one order of magnitude for 6D Vlasov-Poisson (Landau damping) problem compared to Max Planck Society's Struphy MPI particle-in-cell code **[Possanner et al., 2023]**

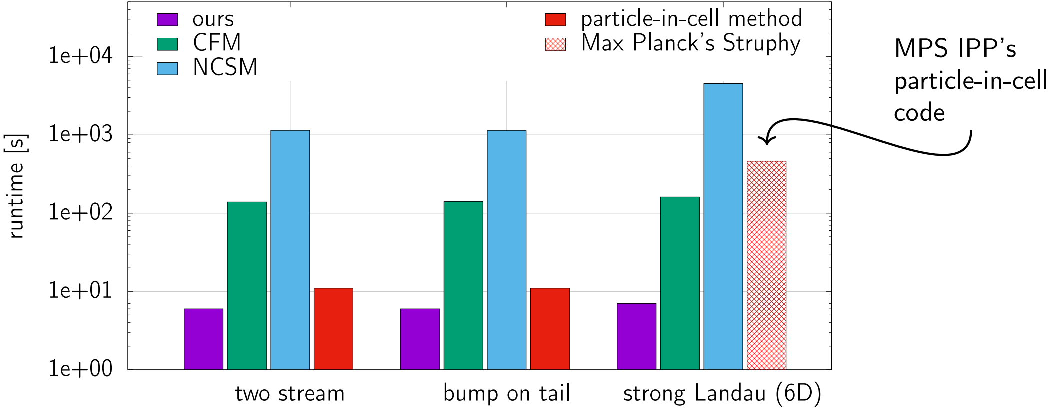
Experiments: Speedups over particle-in-cell methods



Speedups over numerical models

- Critically, we learn reduced model from data; no need to re-implement codes
- Speedups of more than one order of magnitude for 6D Vlasov-Poisson (Landau damping) problem compared to Max Planck Society's Struphy MPI particle-in-cell code **[Possanner et al., 2023]**

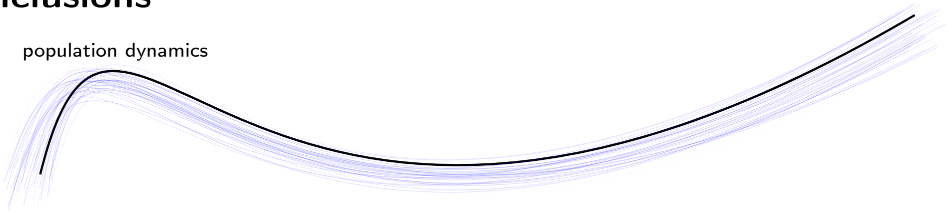
Experiments: Speedups over particle-in-cell methods



Speedups over numerical models

- Critically, we learn reduced model from data; no need to re-implement codes
- Speedups of more than one order of magnitude for 6D Vlasov-Poisson (Landau damping) problem compared to Max Planck Society's Struphy MPI particle-in-cell code **[Possanner et al., 2023]**

Conclusions



- Population dynamics vs. sample dynamics: Enables learning reduced models of systems that are meaningful in statistical sense such as chaotic and stochastic systems.
- Careful discretization of loss functions is crucial for stable training: Bringing tools from numerical analysis such as finite differences can be helpful.
- Inference step of models of population dynamics is fast: Speedups over traditional generative modeling and classical numerical solvers.

References:

- Berman, Blickhan, P., Parametric model reduction of mean-field and stochastic systems via higher-order action matching, NeurIPS, 2024.
- Blickhan, Berman, Stuart, P., DICE: Discrete inverse continuity equation for learning population dynamics, *upcoming*.