LPENS
LABORATOIRE DE PHYSIQUE
DE L'ÉCOLE NORMALE SUPÉRIEURE

Generative modeling and sampling using transport maps

Summer school

Mathematical methods for high-dimensional data

Sapienza Università di Roma

September 8-12, 2025

Marylou Gabrié

École Normale Supérieure

Two related tasks we will be interested in this course

Generative modelling

training dataset

unknown

$$x^{(1)}, x^{(2)}, \ldots, x^{(k)} \sim \rho_*(x)$$

train

$$\rho_\theta(\cdot)$$

sample

$$x^{\text{new}} \sim \rho_\theta(\cdot)$$

Sampling

target probability density $\rho_*(x)$

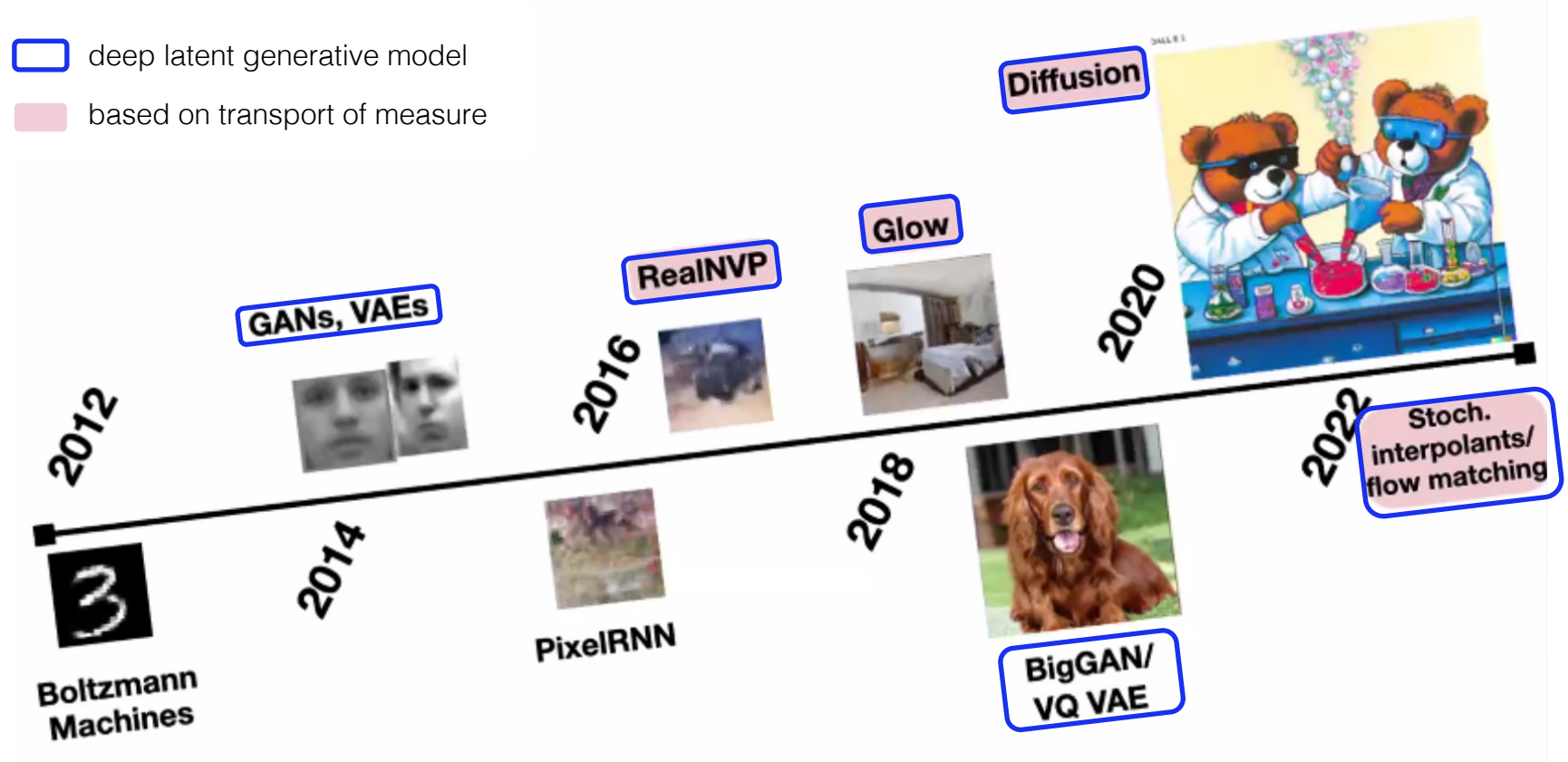
known up to a normalization constant

sample

$$x^{(1)}, x^{(2)}, \ldots, x^{(k)} \sim \rho_*(x)$$

a priori unrelated to learning ...

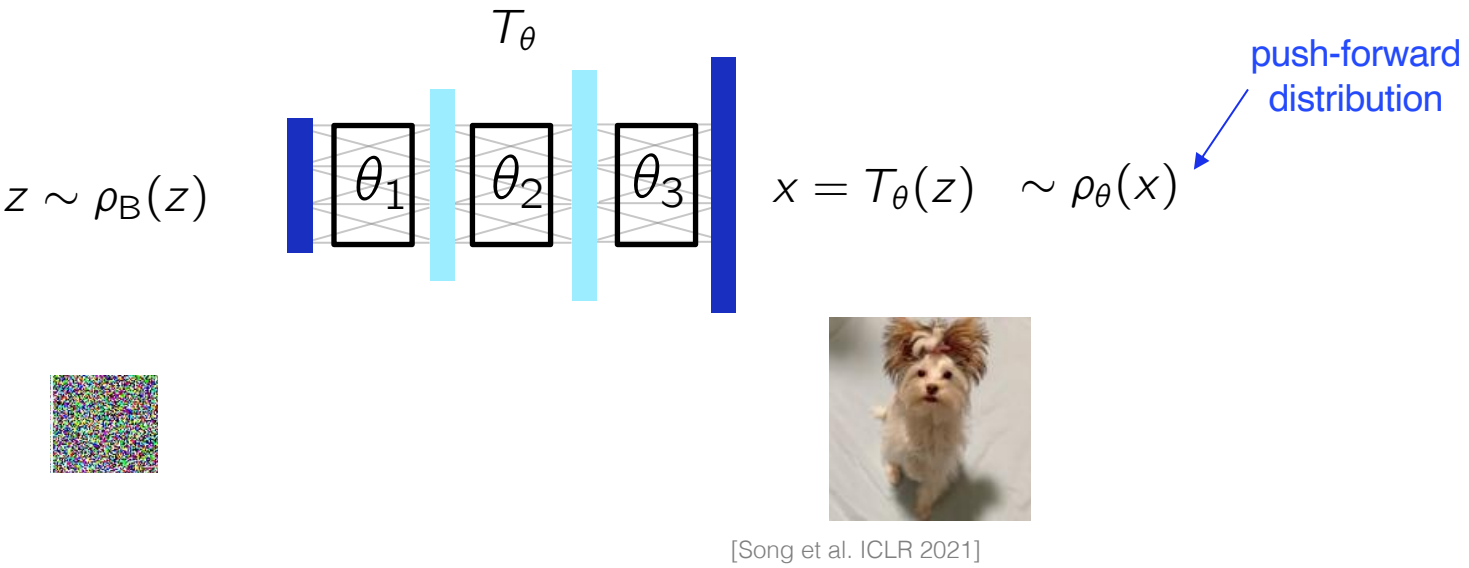
Historical development of generative models



slide credit Michael Albergo

Deep latent generative models

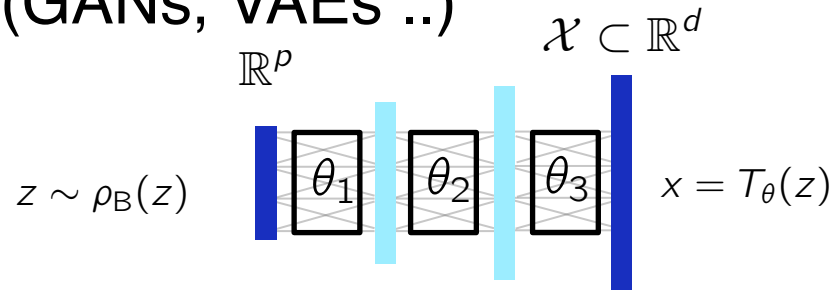
▷ Use transformation $T_\theta : \mathbb{R}^p \to \mathcal{X}$ (deep neural network) from simple base distribution ρ_B :

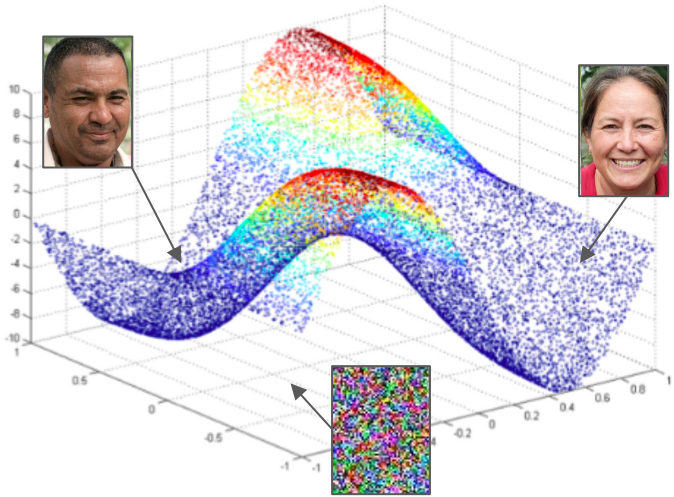


[Song et al. ICLR 2021]

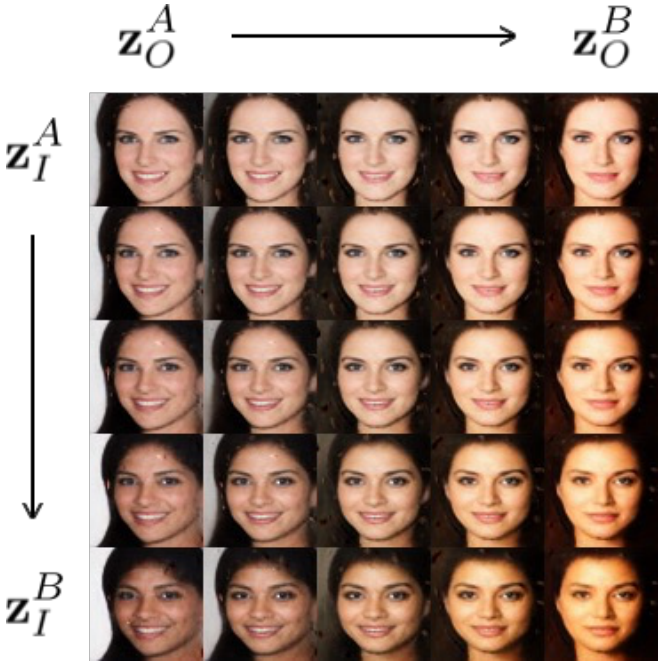
New dog picture for each new base variable!

A small latent space following the manifold hypothesis $p \ll d$ (GANs, VAEs ..)





Interpolating in the latent space:



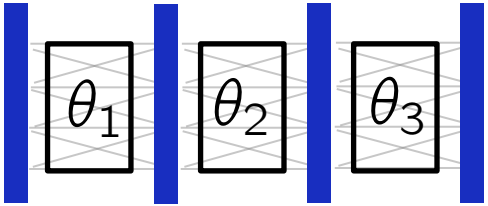
[Donahue et al, ICLR 2018]

▷ As we will see however, modern transport-based generative models will drop this intuition!

Transport based deep generative models

learnable diffeomorphism
= transport map

$T_\theta : \Omega \mapsto \Omega$

$z \sim \rho_B(z)$  $x = T_\theta(z) \quad \sim \rho_\theta(x)$

[Song et al. ICLR 2021]

We will see:

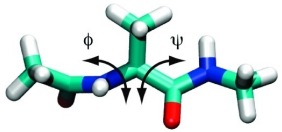
- The transport map can be parametrized in different ways:
  - Explicit parametrization of the map (normalizing flows)
  - Parametrization of a velocity or drift field to be included in an ODE/SDE (continuous normalizing flows, diffusion models, flow matchings/stochastic interpolants)

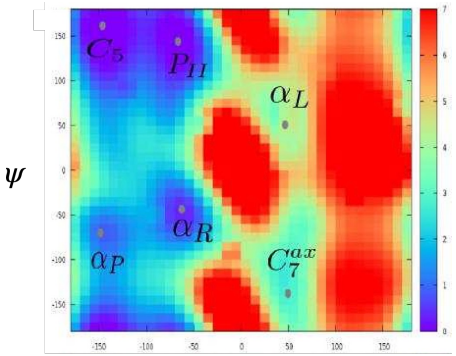
- Transport based generative models are also well suited for sampling!

Examples of scientific applications of sampling

▷ Statistical mechanics:
Gibbs-Boltzmann distribution

$$\rho_*(x) = \frac{1}{\mathcal{Z}_\beta} e^{-\beta U(x)}$$

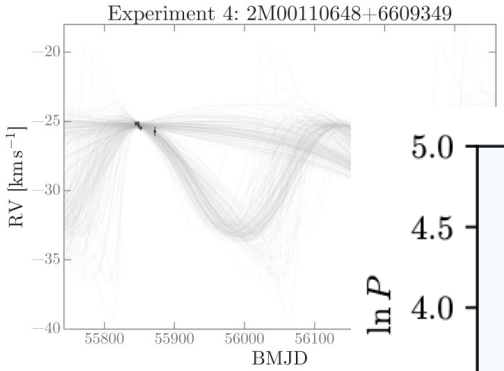




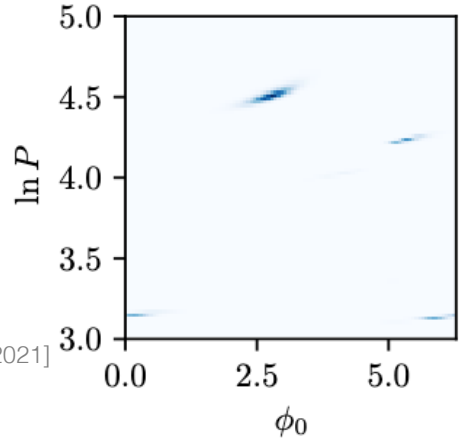
[Jaillet et al,
J Comp Chem 2011]

▷ Data analysis: Bayesian posteriors

$$\rho_*(\theta) \propto \ell(\mathcal{D}|\theta)\rho_0(\theta)$$



[Price-Whelan et al.
The Astrophysical Journal 2017]



[Gabrié et al, ICML workshop 2021]

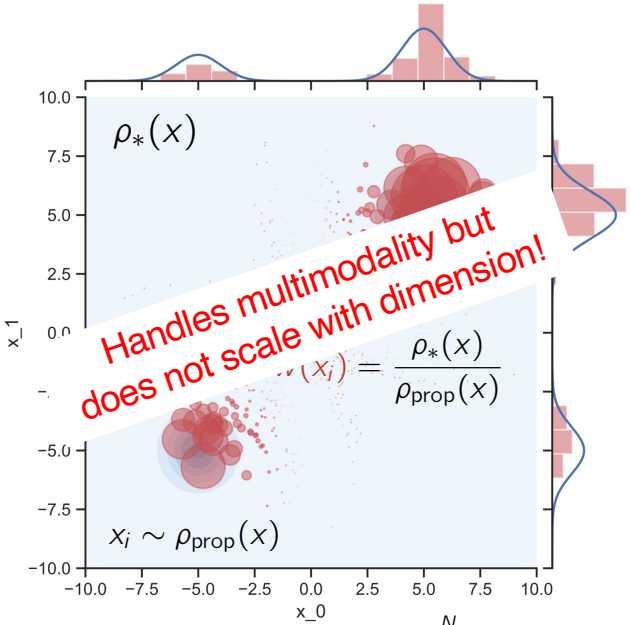
▷ Distribution is known up to a normalization constant and to retrieve information from it,

first sample $x^{(1)}, x^{(2)}, \ldots, x^{(k)} \sim \rho_*(x)$ then build Monte Carlo estimators $\frac{1}{k}\sum_{i=1}^{k} f(x^i) \approx \mathbb{E}[f(x)]$

Why sampling multimodal distributions is hard?

Two fundamental approaches to sampling:

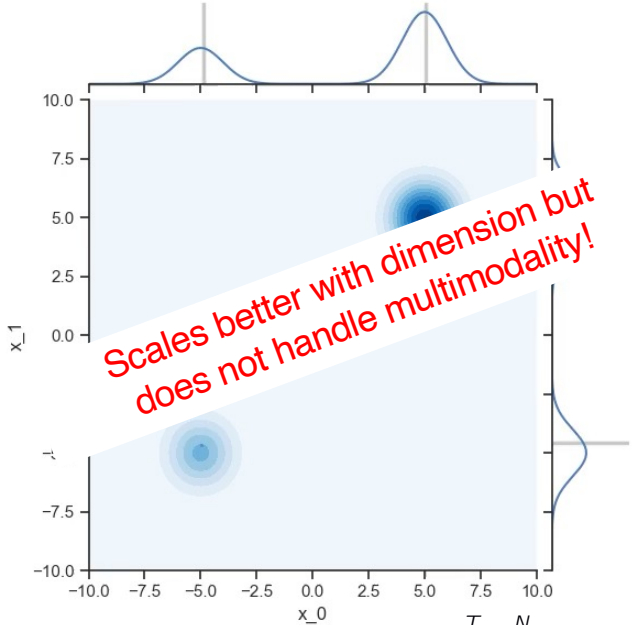
- Shoot and reject/reweight algorithms:
  (e.g. Importance Sampling IS)



$$\mathbb{E}_{\rho_*}[f(x)] = \int_\Omega f(x)\rho_*(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i)f(x_i)$$

High variance!

- Local exploration Markov chain Monte Carlo (MCMC) *(e.g. Metropolis Adjusted Langevin)*



$$\mathbb{E}_{\rho_*}[f(x)] = \int_\Omega f(x)\rho_*(x)\mathrm{d}x \approx \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N} f(x_i^t)$$

High bias!

Enhanced samplers for multimodal distributions

▷ Decades of research on how to best tackle multimodal distributions

▷ **Annealing methods:** a path of distributions bridging an easy to sample distribution to the target

e.g. Parallel tempering/replica exchange, Annealed Importance Sampling (AIS), Sequential Monte Carlo (SMC)

[Marinari & Parisi (1992), Geyer & Thomson (1995), Neal (1998), Del Moral, Doucet & Ajay (2006) etc.]

▷ **Enhanced samplers with "collective variables":** drive exploration along a low dimensional projection

e.g. Metadynamics, Adaptive Biasing Force, Umbrella Sampling

[Fu et al. "Enhanced Sampling Based on Collective Variables." 2023]

▷ **Samplers assisted by generative models:** Train a model to approximate the target and use it as a helper

mainly with Autoregressive Models, Normalizing Flows

[Rezende & Mohamed ICML 2015, Albergo et al PRD 2019, Wu et al. PRL 2019, Noé et al. Science 2019, Gabrié et al. PNAS 2019 etc.]

& Diffusion Models

[Vargas et al. arXiv:2302.13834, RDMC - Huang et al. 2307.02037, Berner et al. arXiv:2307.01198,
Vargas et al. arXiv:arXiv:2307.01050, SLIPS – Grenioux et al. 2402.10758, Akhound-Sadegh et al. 2402.06121]

Outline

I. Normalizing flows

A. Original formulation
   i. construction
   ii. training
   iii. use for sampling

B. Continuous normalizing flows
   i. construction
   ii. likelihood computation

II. Diffusion models & flow matchings

A. Diffusion models
   i. construction
   ii. training
   iii. use for sampling

B. Flow matchings/Stochastic interpolants
   i. construction
   ii. training

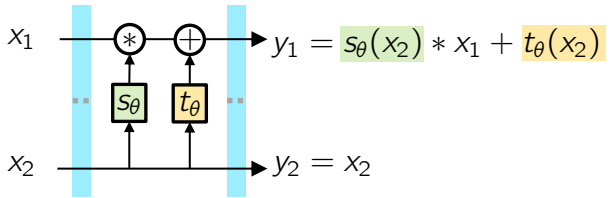
I.A.i A special type of Deep Generative Models
Normalizing Flows (NF): Invertible networks

▷ Parametrized invertible map $T_\theta : \Omega \mapsto \Omega \qquad \Omega \subset \mathbb{R}^d$

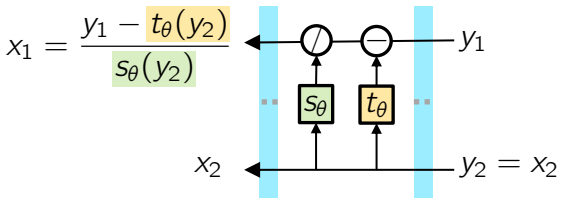
- Base distribution $z \sim \rho_B(z)$
- Push-forward distribution $x = T_\theta(z) \quad \sim \rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \det \left|\nabla_x T_\theta^{-1}\right|$

▷ e.g. “Coupling layers”: easy-to-compute inverse and Jacobian

Affine coupling layer $T_\theta(x)$



Inverse layer $T_\theta^{-1}(y)$



Block diagonal Jacobian: $\nabla_x T_\theta(x) = \begin{bmatrix} s_\theta(x_2) I_{d/2} & X \\ 0 & I_{d/2} \end{bmatrix}$

[Tabak & Vanden Eijnden Commun. Math. Sci. 2010, Dinh, L. et al ICLR 2017, Papamakarios, G et al JMLR 2021]

I.A.iii. Normalizing flows (NF) for sampling

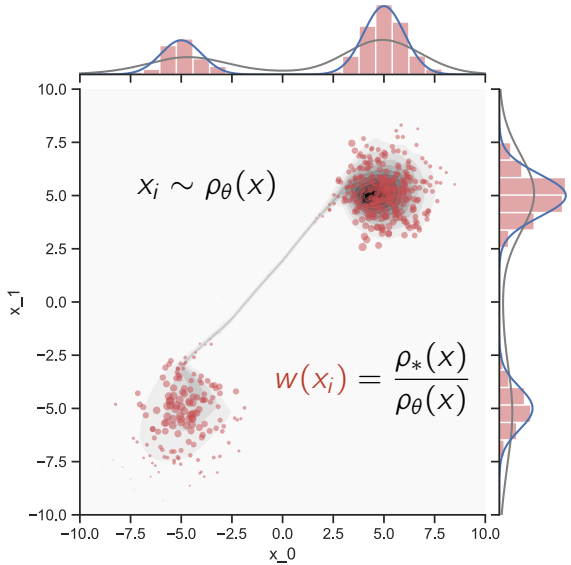
▷ Normalizing flows are a type of generative models with tractable likelihood with $T_\theta : \Omega \mapsto \Omega$ a diffeomorphism: $\rho_\theta(x) = \rho_B(T_\theta^{-1}(x)) \det \left| \nabla_x T_\theta^{-1} \right|$

[Tabak & Vanden-Eijnden Com. Math. Sci. 2010, Dinh et al ICLR 2017, Papamakarios et al. JMLR 2021]

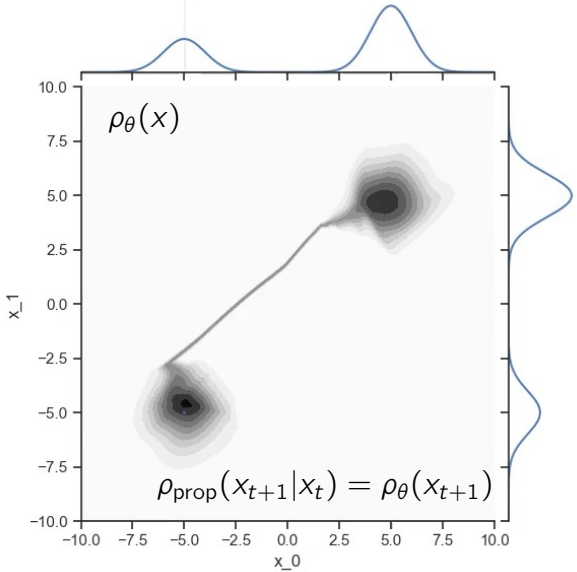
▷ Use approximate NF model as a proposal in Monte Carlo strategies to obtain exact samples

Pioneering works in physics: [Rezende & Mohamed ICML 2015, Albergo et al. PRD 2019, Wu et al. PRL 2019, Noé et al. Science 2019]

Importance sampling



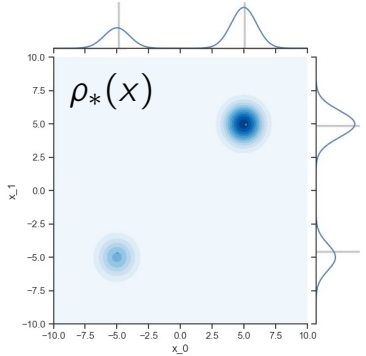
Metropolis-Hastings



I.A.iii. Adaptive training: Learning a NF while sampling

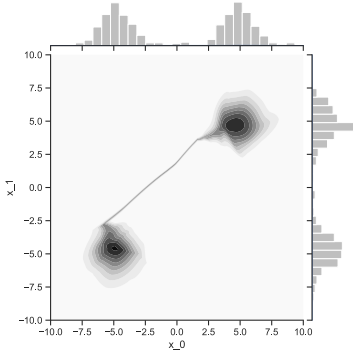
▷ flowMC loops over 3 steps: [MG, Rotskoff, Vanden-Eijnden PNAS 2022]

1. local sampler



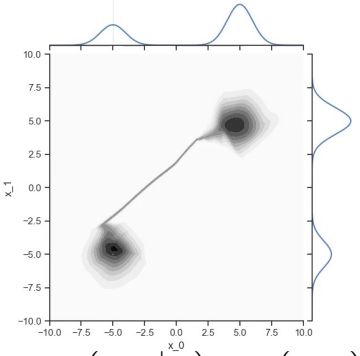
$$x^i_{t+1} \sim \pi_{\text{local}}(x^i_{t+1}|x^i_t)$$

2. maximum likelihood



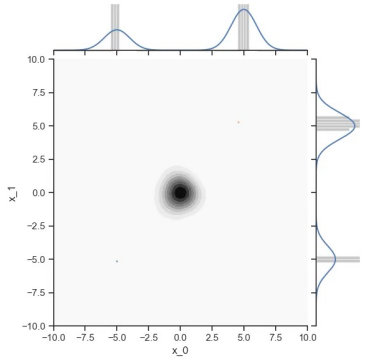
$$\theta^* = \arg\max_\theta \sum_{i,t} \log \rho_\theta(x^i_t)$$

3. non-local sampler



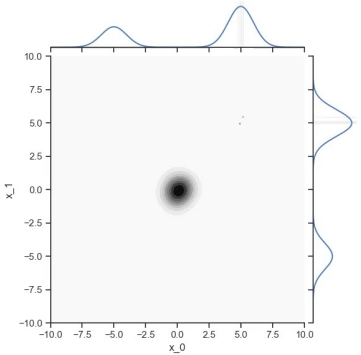
$$\rho_{\text{prop}}(x_{t+1}|x_t) = \rho_\theta(x_{t+1})$$

▷ Converging all steps in parallel:



▷ Requires the knowledge of the modes location!

a.k.a no free lunch!



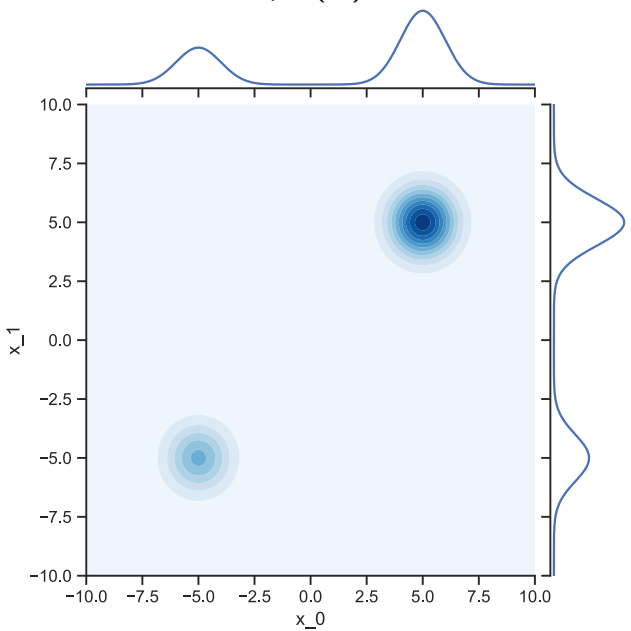
[MG, Rotskoff, Vanden-Eijnden PNAS 2022, Parno & Marzouk 2018 SIAM Journal of Uncertainty; Naesseth et al. Neurips 2020]

I.A.iii. Variational training of NFs for sampling

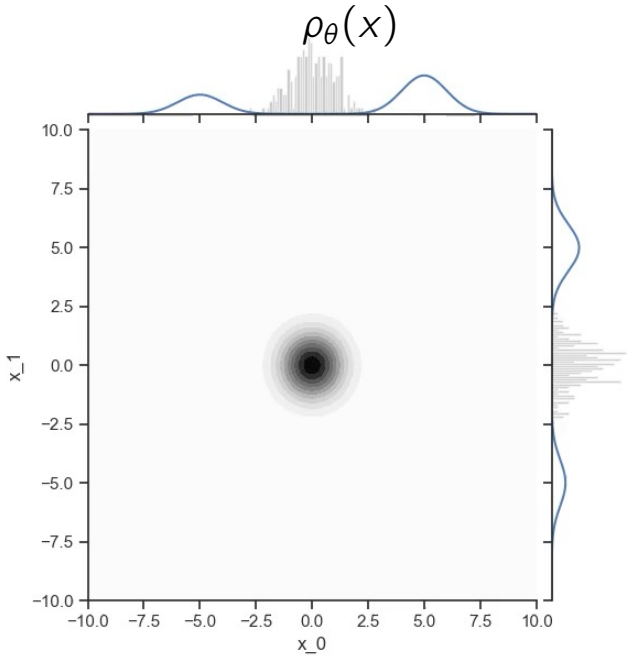
▷ A data-free learning objective: the (reverse) Kullback-Leibler divergence $D_{\mathrm{KL}}(\rho_\theta \| \rho_*)$

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_*) = \int \log \frac{\rho_\theta(x)}{\rho_*(x)} \rho_\theta(x) \mathrm{d}x \approx \frac{1}{N} \sum_{i=1}^{N} \log \frac{\rho_B(z_i) \det |\nabla_{z_i} T_\theta|}{\rho_*(T_\theta(z_i))} \qquad z_i \sim \rho_B(z)$$

$\rho_*(x)$



$\rho_\theta(x)$



Mode collapse!
Adhoc fixes in these first papers (annealing and adding data!)

Weiss, P. (1907). L'hypothèse du champ moléculaire et la propriété ferromagnétique.
Rezende et al. ICML 205, Wu et al. PRL 2019, Albergo et al PRD 2019

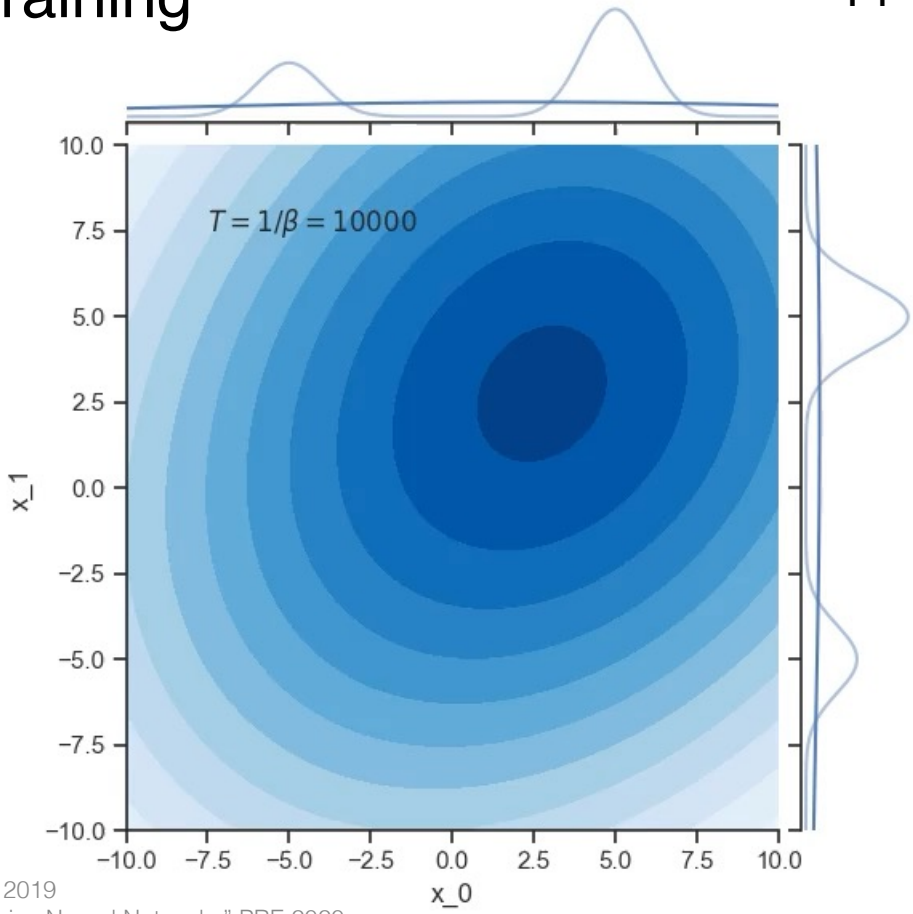
I.A.iii. Sequential tempering for training

▷ Consider a tempering path of distributions:

$$\rho_{*,\beta}(x) = \rho_*(x)^\beta$$

$$\beta_0 < \beta_1 \ldots < \beta_N = 1$$

▷ Train either
- adaptively (sampling MCMC chains in parallel)
- variationally (minimize the “reverse” KL)



Wu et al. “Solving Statistical Mechanics Using Variational Autoregressive Networks.” PRL 2019
McNaughton et al. “Boosting Monte Carlo Simulations of Spin Glasses Using Autoregressive Neural Networks” PRE 2020
Hackett et al. “Flow-Based Sampling for Multimodal Distributions in Lattice Field Theory.” arXiv:2107.00734.
Ciarella et al. “Machine-Learning-Assisted Monte Carlo Fails at Sampling Computationally Hard” MLST 2023

Outline

I. Normalizing flows

A. Original formulation
   i. construction
   ii. training
   iii. use for sampling

B. Continuous normalizing flows
   i. construction
   ii. likelihood computation

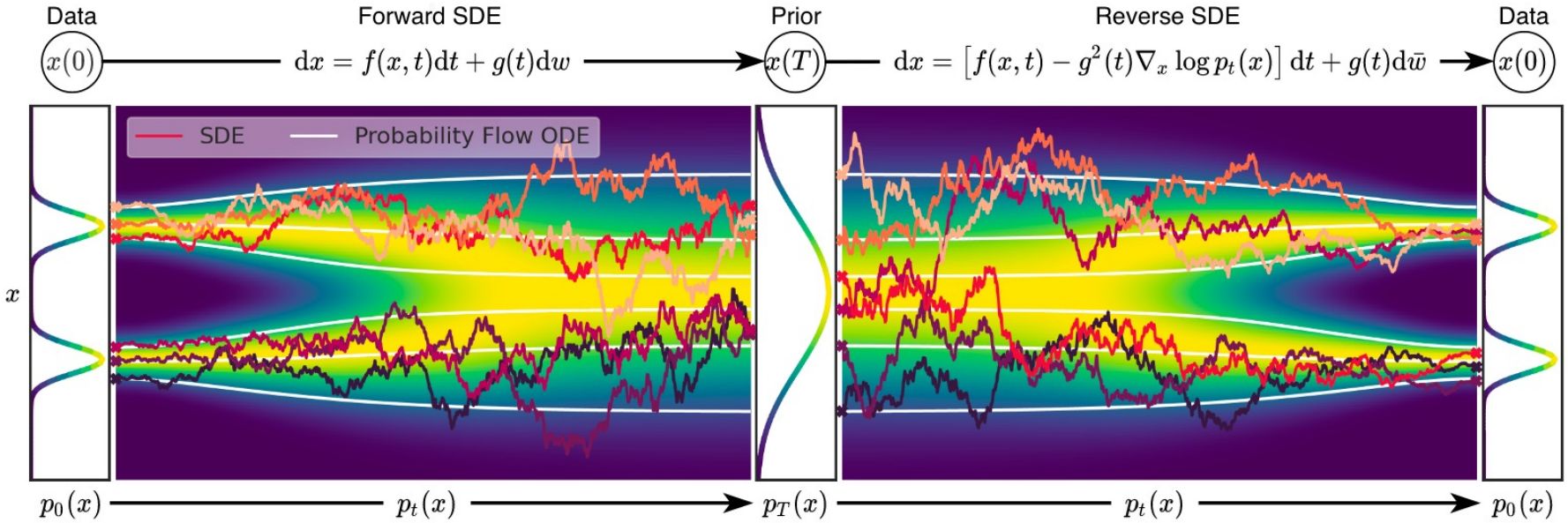
II. Diffusion models & flow matchings

A. Diffusion models
   i. construction
   ii. training
   iii. use for sampling

B. Flow matchings/Stochastic interpolants
   i. construction
   ii. training

Diffusion models

▷ An example of noising/denoising paths for a 1D mixture of Gaussian distribution



[Song et al. ICLR 2021]

Brian D O Anderson. Reverse-time diffusion equation models. Stochastic Process. Appl., 12(3): 313–326, May 1982.
Sohl-Dickstein et al. “Deep Unsupervised Learning Using Nonequilibrium Thermodynamics.” ICML 2015,
Ho et al. “Denoising Diffusion Probabilistic Models.” In Neural Information Processing Systems 2020
Song et al. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021

Diffusion models quickly became state of the art

Figure 6: Samples from BigGAN-deep with truncation 1.0 (FID 6.95, left) vs samples from our diffusion model with guidance (FID 4.59, middle) and samples from the training set (right).

[Diffusion Models Beat GANs on Image Synthesis, Dhariwal & Nichol, NeurIPS 2021]

II.A.iii. Diffusion for sampling: Investigated strategies to estimate the score without data

▷ **Non parametric:** Monte Carlo estimation of the score in Monte Carlo diffusion samplers

Saremi et al. "Chain of Log-Concave Markov Chains," ICLR 2024
Huang et al. "Reverse Diffusion Monte Carlo," ICLR 2024
Grenioux*, Noble*, MG, and Oliviero Durmus. "Stochastic Localization via Iterative Posterior Sampling." ICML 2024

$$\nabla \log p_t(y) = -\frac{y - e^{-t}\,\mathbb{E}[X_0|X_t = y]}{(1 - e^{-2t})} \quad \text{where} \quad X_t \overset{\mathcal{L}}{=} X_0 e^{-t} + \sqrt{1 - e^{-2t}}\,Z$$

Computing the score = computing an expectation with respect to the posterior $q_t(x_0|y) \propto p_t(y|x_0)\,\rho_*(x_0)$

$\mathcal{N}\big(y; e^{-t}x_0, (1 - e^{-2t})\,I_d\big)$

▷ **Parametric:** Variational inference meets diffusion models

Zhang, et al. "Path Integral Sampler: A Stochastic Control Approach For Sampling." ICLR 2022
Vargas, et al. "Denoising Diffusion Samplers." ICLR 2023
Richter, et al. "Improved sampling via learned diffusions." ICLR 2024
Noble*, Grenioux* et MG and Oliviero Durmus. "Learned Reference-based Diffusion Sampler for multi-modal distributions" , ICLR 2025

path measures

$\mathbb{P}_*$ →
- fwd: $X_0 \sim \rho_*$ $\quad \mathrm{d}X_t = -X_t \mathrm{d}t + \sqrt{2}\mathrm{d}W_t$ *(e.g. Ornstein-Uhlenbeck)*
- bwd: $Y_0 \sim \mathcal{N}(0, I)$ $\quad \mathrm{d}Y_t = (Y_t + \nabla \log p_{T-t}(Y_t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$

$\mathbb{P}_\theta$ → bwd: $Y_0 \sim \mathcal{N}(0, I)$ $\quad \mathrm{d}Y_t = (Y_t + s_t^\theta(Y_t))\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$

Path space VI problem

$$\min_\theta \mathrm{KL}(\mathbb{P}_\theta || \mathbb{P}_*)$$

Outline

I. Normalizing flows

- A. Original formulation
  - i. construction
  - ii. training
  - iii. use for sampling
- B. Continuous normalizing flows
  - i. construction
  - ii. likelihood computation

II. Diffusion models & flow matchings

- A. Diffusion models
  - i. construction
  - ii. training
  - iii. use for sampling
- B. Flow matchings/Stochastic interpolants
  - i. construction
  - ii. training

II.B.i Flow matching / stochastic interpolant models

▷ Build a bridge between two arbitrary distributions

$\rho_1(x_1)$ data distribution

$\rho_0(x_0)$ arbitrary base distribution

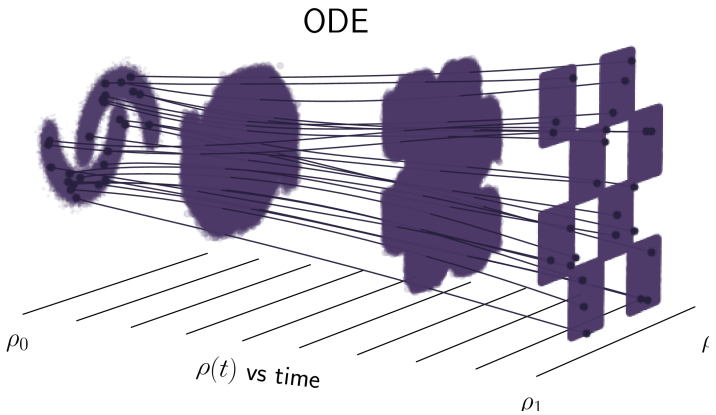
Interpolation function $I_t(x_0, x_1) = x_t = tx_1 + (1-t)x_0$

Instantaneous distribution $\rho_t(x) = \int \mathrm{d}x_0 \int \mathrm{d}x_1 \delta\big(x - I_t(x_0, x_1)\big)\rho_0(x_0)\rho_1(x_1)$

▷ The distribution $\rho_t(x)$ is also generated by an ODE $\dfrac{\mathrm{d}x_t}{\mathrm{d}t} = v(x_t, t)$

with $v = \arg\min \int_0^1 \mathrm{d}t\, \mathbb{E}_{\rho_0,\rho_1}\left[\|v(x_t) - \partial_t I_t(x_0, x_1)\|^2\right]$

Loss function for learning the velocity



https://github.com/malbergo/stochastic-interpolants

▷ Close cousins to diffusion models, differences/advantages

- Base distribution can be arbitrary
- The process bridges the two distribution exactly between times 0 and 1
- But requires to choose an interpolant (how?)

Albergo et al, “Building Normalizing Flows with Stochastic Interpolants”, ICLR 2022
Lipman et al, “Flow Matching for Generative Modeling,” ICLR 2022
Albergo et al, “Stochastic Interpolants: A Unifying Framework for Flows and Diffusions”, JMLR 2024

image generation

Model	Params(M)	Training Steps	FID ↓
DiT-S	33	400K	68.4
SiT-S	33	400K	57.6
DiT-B	130	400K	43.5
SiT-B	130	400K	33.5
DiT-L	458	400K	23.3
SiT-L	458	400K	18.8
DiT-XL	675	400K	19.5
SiT-XL	675	400K	17.2
DiT-XL	675	7M	9.6
SiT-XL	675	7M	8.6
DiT-XL (cfg-1.5)	675	7M	2.27
SiT-XL (cfg-1.5)	675	7M	2.06

Interpolants beat diffusion models in large-scale image generation!

Ma, Goldstein, Albergo, NMB, Vanden-Eijnden, and Xie. arXiv:2401.08740 (2024)

Slide by Nick Boffi

Outline

I. Normalizing flows

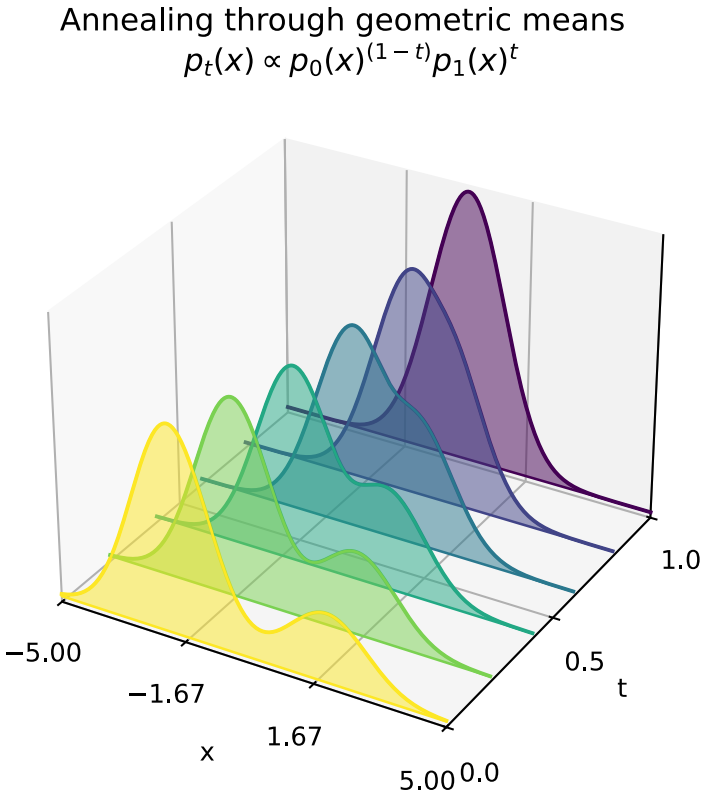
- A. Original formulation
  - i. construction
  - ii. training
  - iii. use for sampling
- B. Continuous normalizing flows
  - i. construction
  - ii. likelihood computation

II. Diffusion models & flow matchings

- A. Diffusion models
  - i. construction
  - ii. training
  - iii. use for sampling
- B. Flow matchings/Stochastic interpolants
  - i. construction
  - ii. training

III. Neural ODE for sampling

III. Neural ODE for sampling



▷ **Annealing methods:** a path of distributions bridging an easy to sample distribution to the target

e.g. Parallel tempering/replica exchange,
Annealed Importance Sampling (AIS),
Sequential Monte Carlo (SMC)

[Marinari & Parisi (1992), Geyer & Thomson (1995), Neal (1998), Del Moral, Doucet & Ajay (2006) etc.]

▷ Idea:

- Fix a path of distributions with density known up to normalization $(p_t(.))_{t\in[0,1]}$
- Learn a velocity field realizing the desired path of distributions $\frac{\partial p_t(x)}{\partial t} = -\nabla(p_t(x) v_t^*(x))$

Vaikuntanathan and Jarzynski. “Escorted Free Energy Simulations: Improving Convergence by Reducing Dissipation.” PRL 2008
Vargas et al “Transport Meets Variational Inference: Controlled Monte Carlo Diffusions.” ICLR 2024.
Albergo and Vanden Eijnden. “NETS: A Non-Equilibrium Transport Sampler.”ICML 2025