State of generative modeling and the sciences

According to me I guess!

Michael Albergo ML Sampling Workshop, Bonn October 24, 2024

How does one even begin to summarize this?

I'm supposed to give you an overview of generative models…

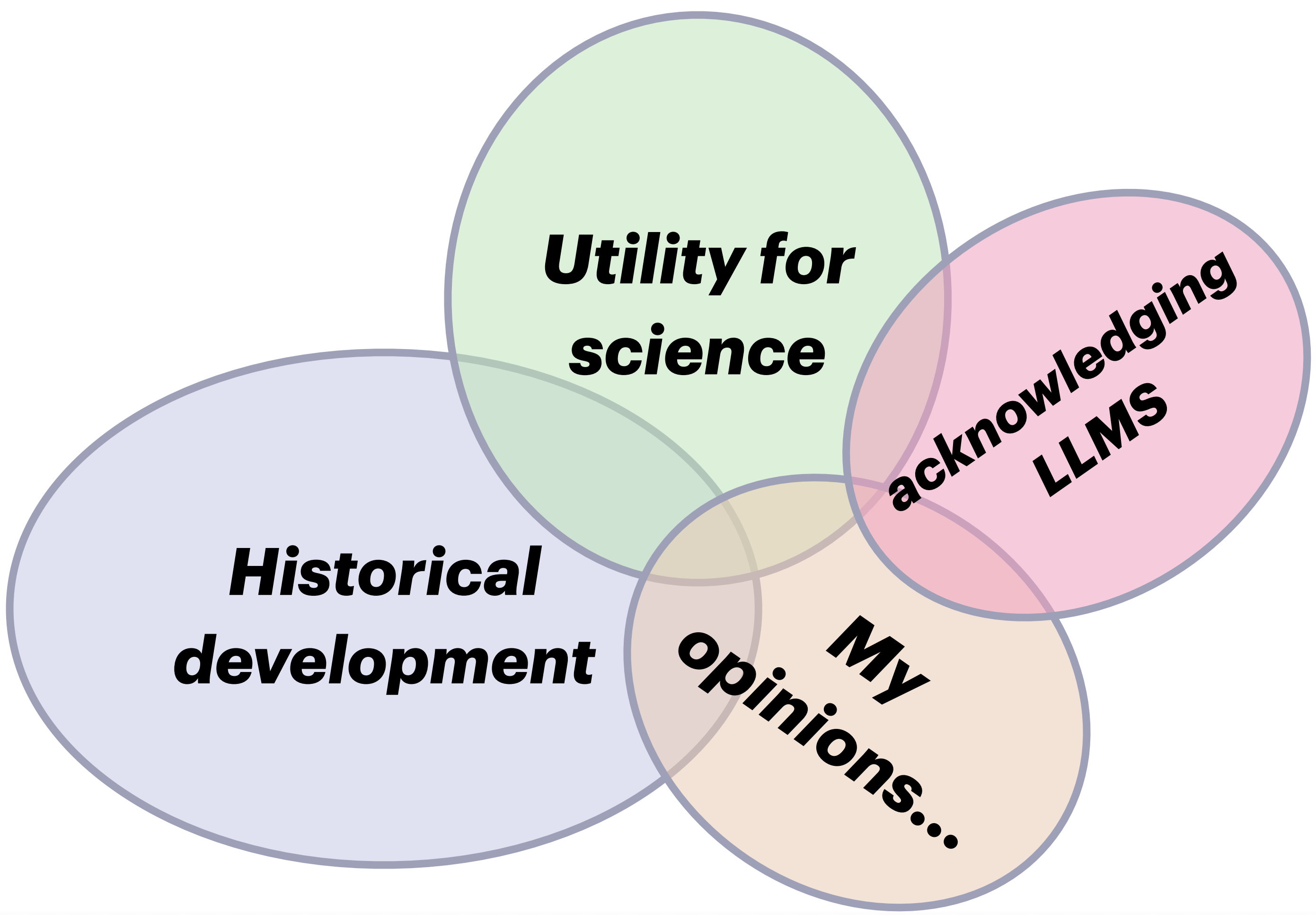
- *Of course this will be biased by my opinions!*
- *I will caveat any claims by this fact :) hopefully spurs some discussion*

How does one even begin to summarize this?

I'm supposed to give you an overview of generative models…

- *Of course this will be biased by my opinions!*
- *I will caveat any claims by this fact :) hopefully spurs some discussion*

The various factors influencing me how to do this

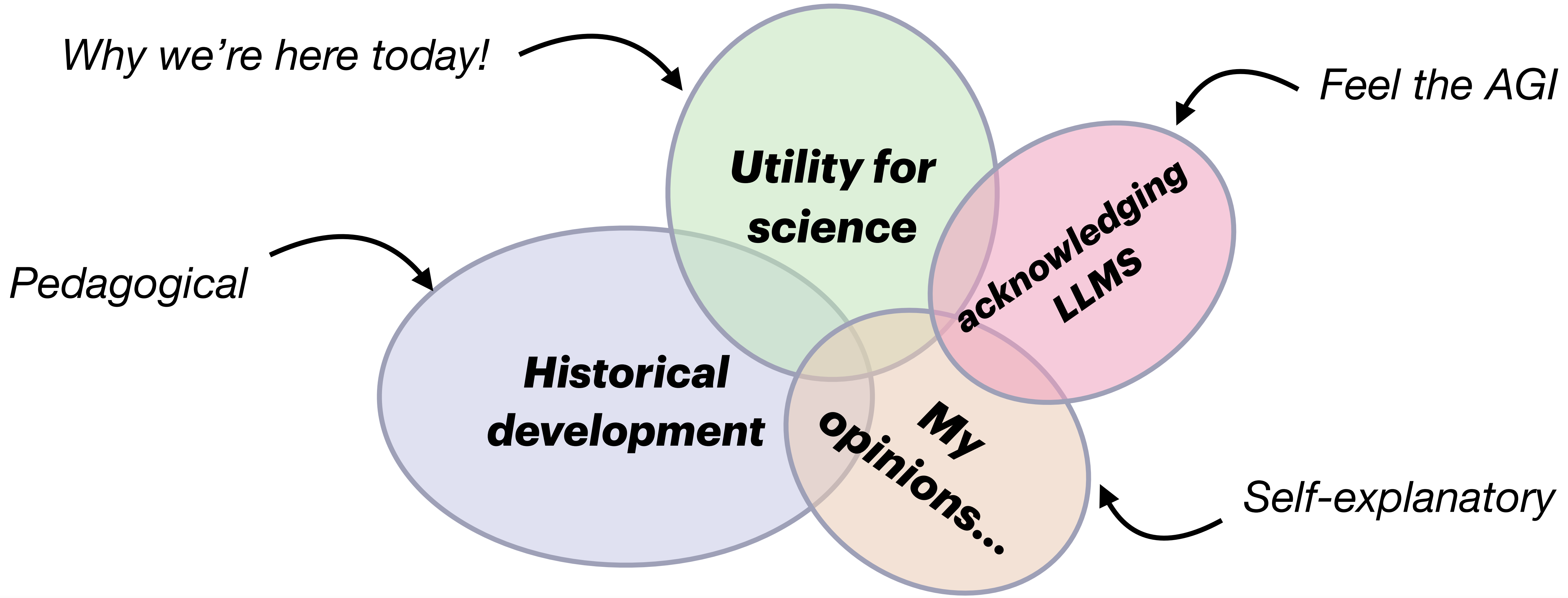


How does one even begin to summarize this?

I'm supposed to give you an overview of generative models…

- *Of course this will be biased by my opinions!*
- *I will caveat any claims by this fact :) hopefully spurs some discussion*

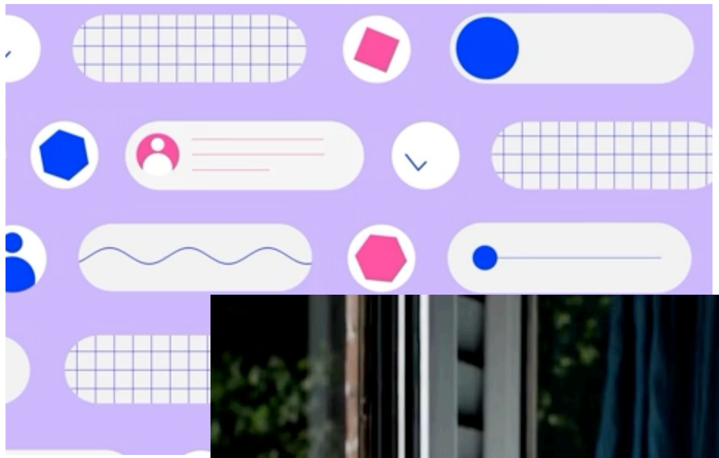
The various factors influencing me how to do this



Problem Motivation: Complexity all around

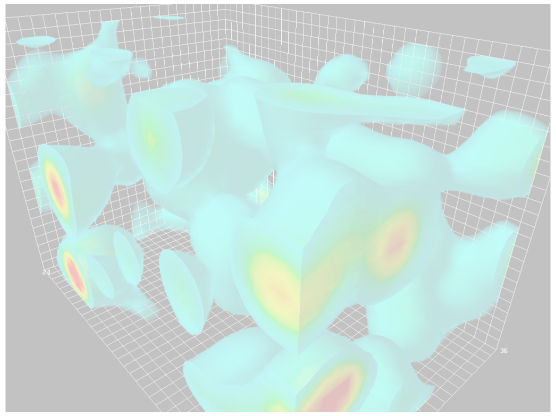
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

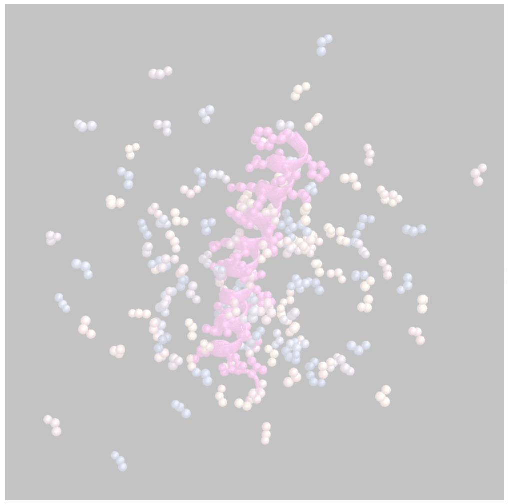


Sora (2024): "A flower growing out on the windowsill"

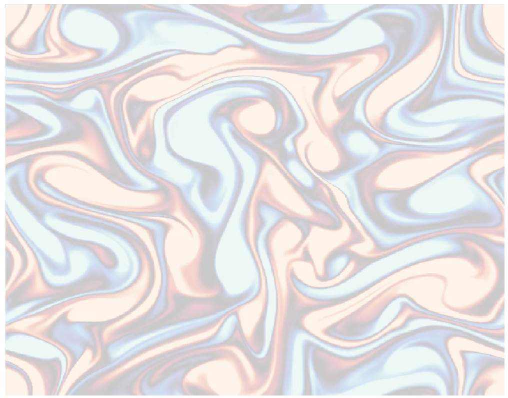
Natural: limited data, but theory



Quantum Theory



Molecular conformation

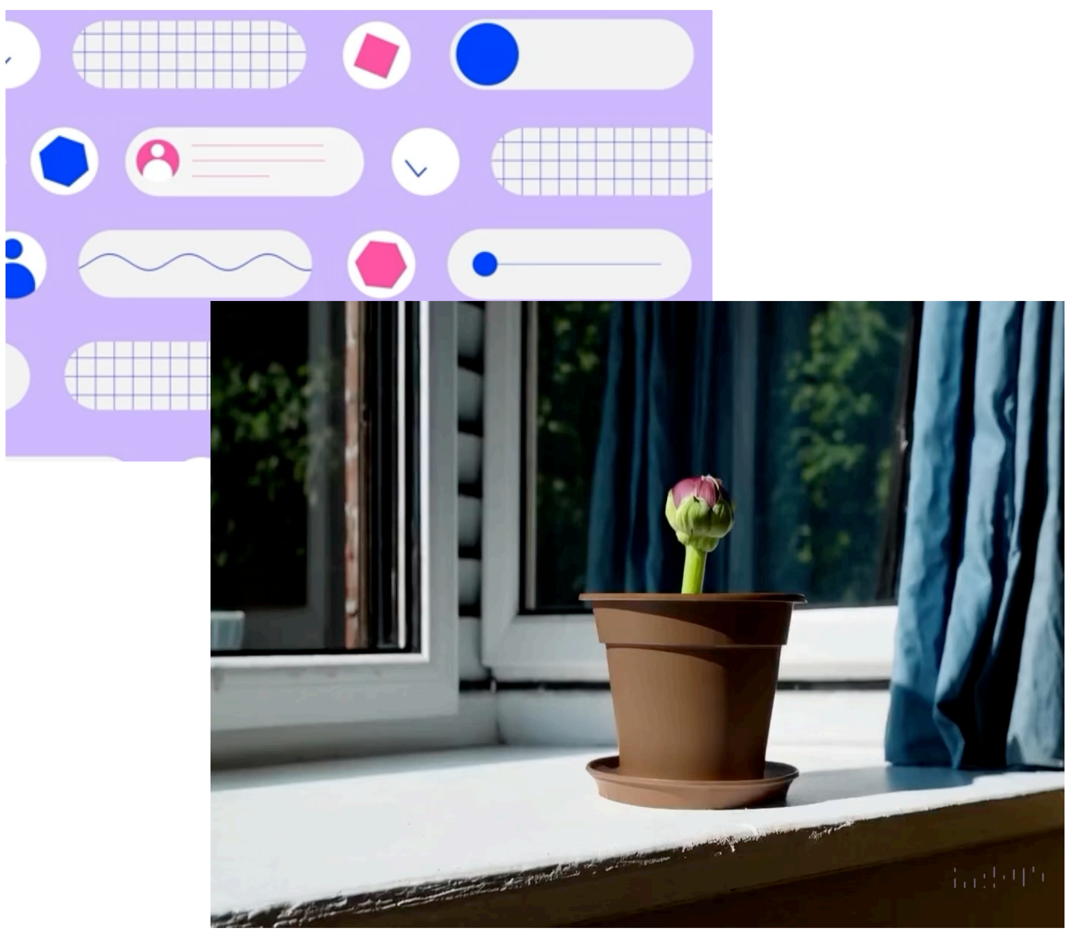


Forecasting

Problem Motivation: Complexity all around

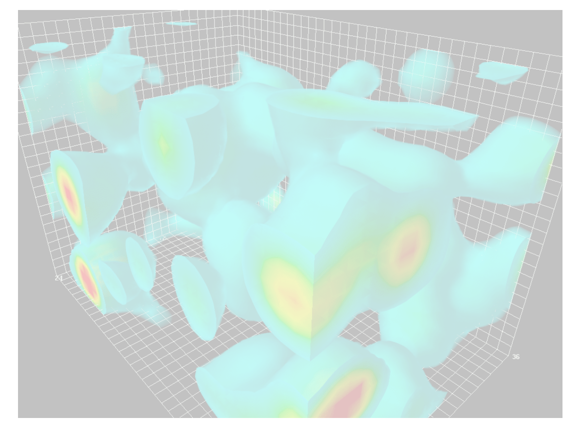
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

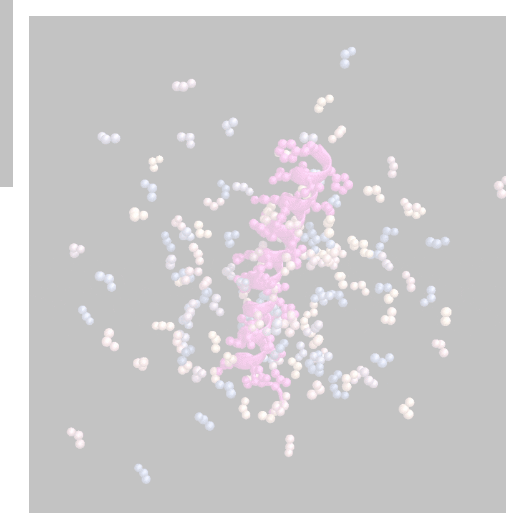


Sora (2024): “A flower growing out on the windowsill”

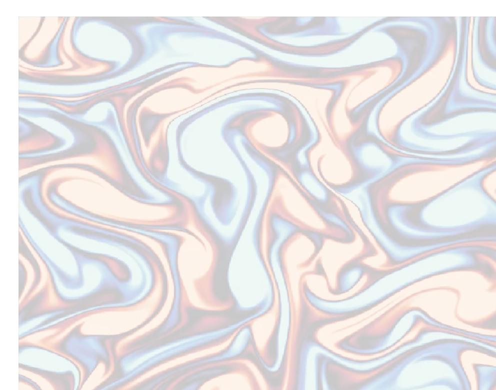
Natural: limited data, but theory



Quantum Theory



Molecular conformation

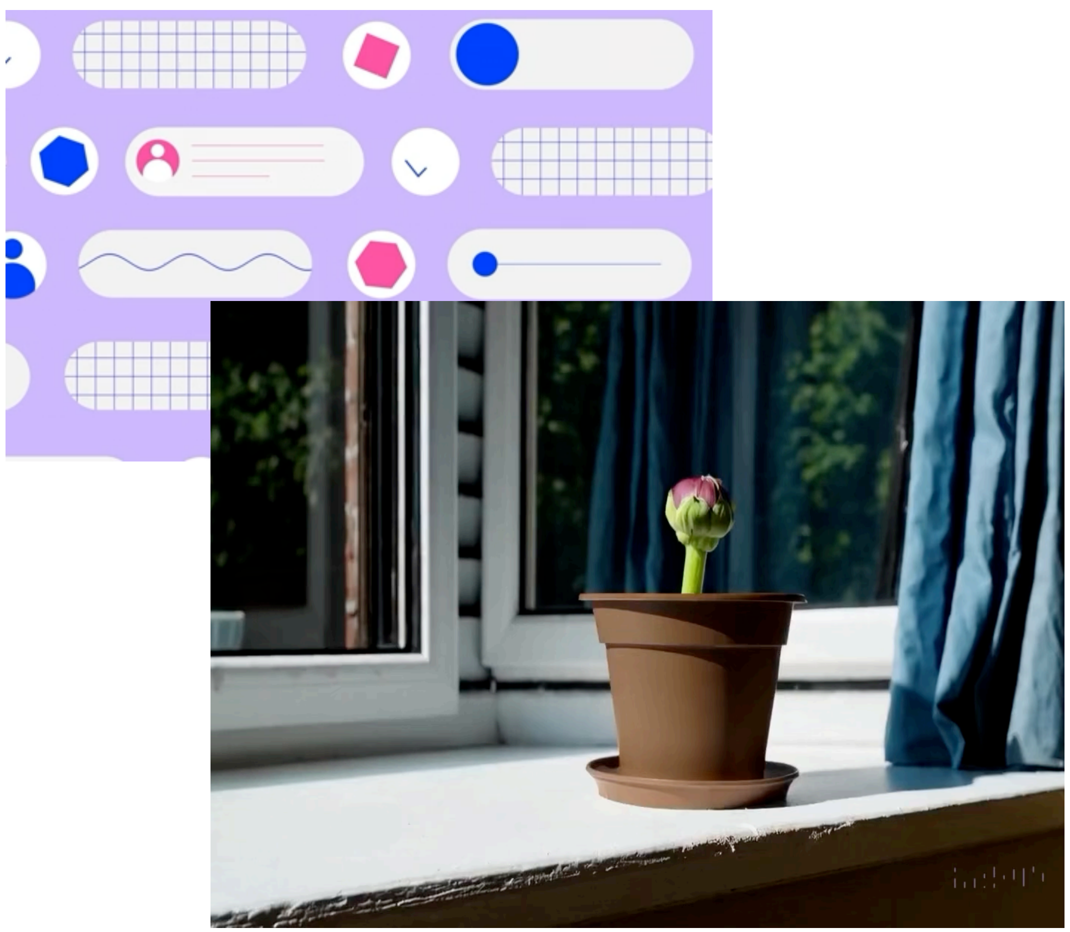


Forecasting

Problem Motivation: Complexity all around

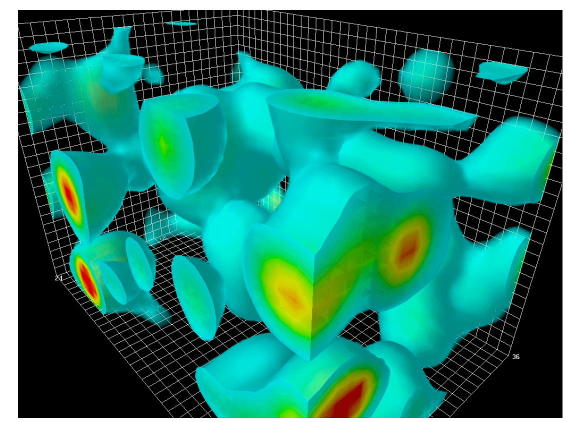
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

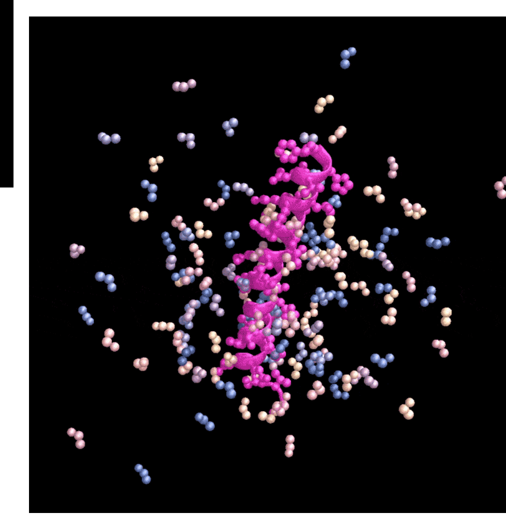


Sora (2024): “A flower growing out on the windowsill”

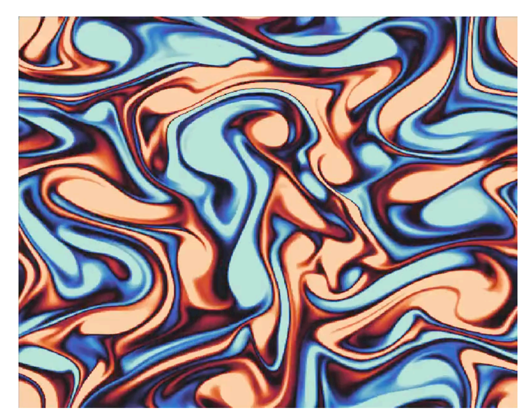
Natural: limited data, but theory



Quantum Theory



Molecular conformation

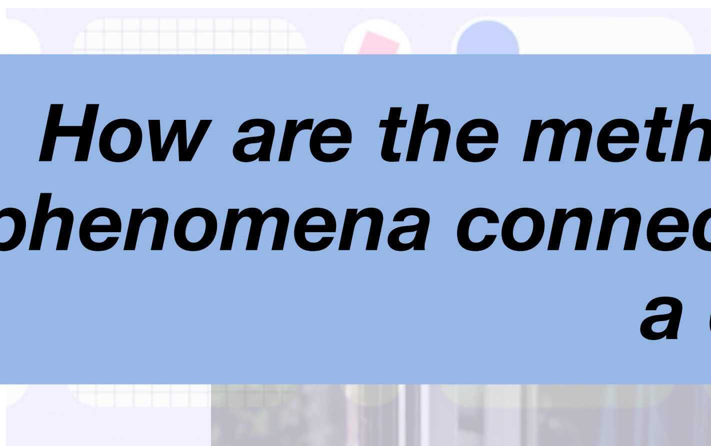


Forecasting

Problem Motivation: Complexity all around

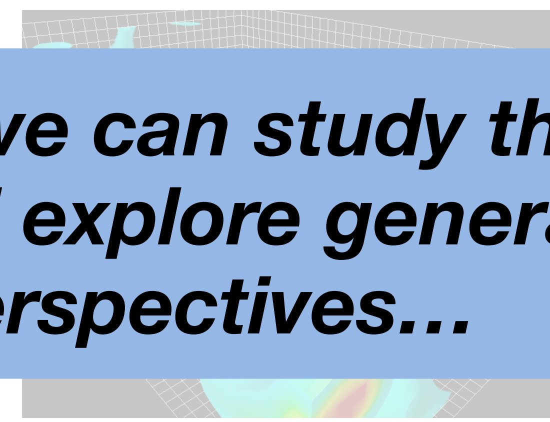
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

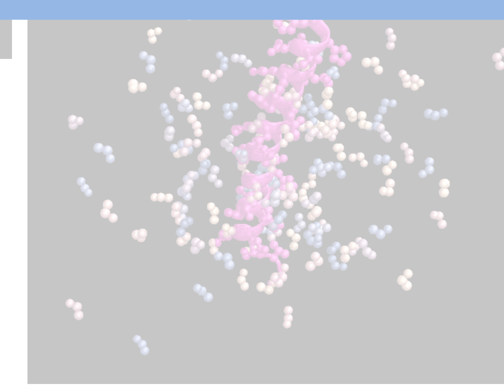


Sora (2024): "A flower growing out on the windowsill"

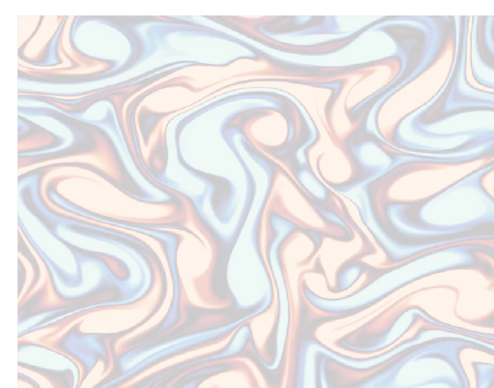
Natural: limited data, but theory



Quantum Theory



Molecular conformation



Forecasting

How are the methods through which we can study these disparate phenomena connected, if at all? We will explore generative models as a dual lens to these perspectives…

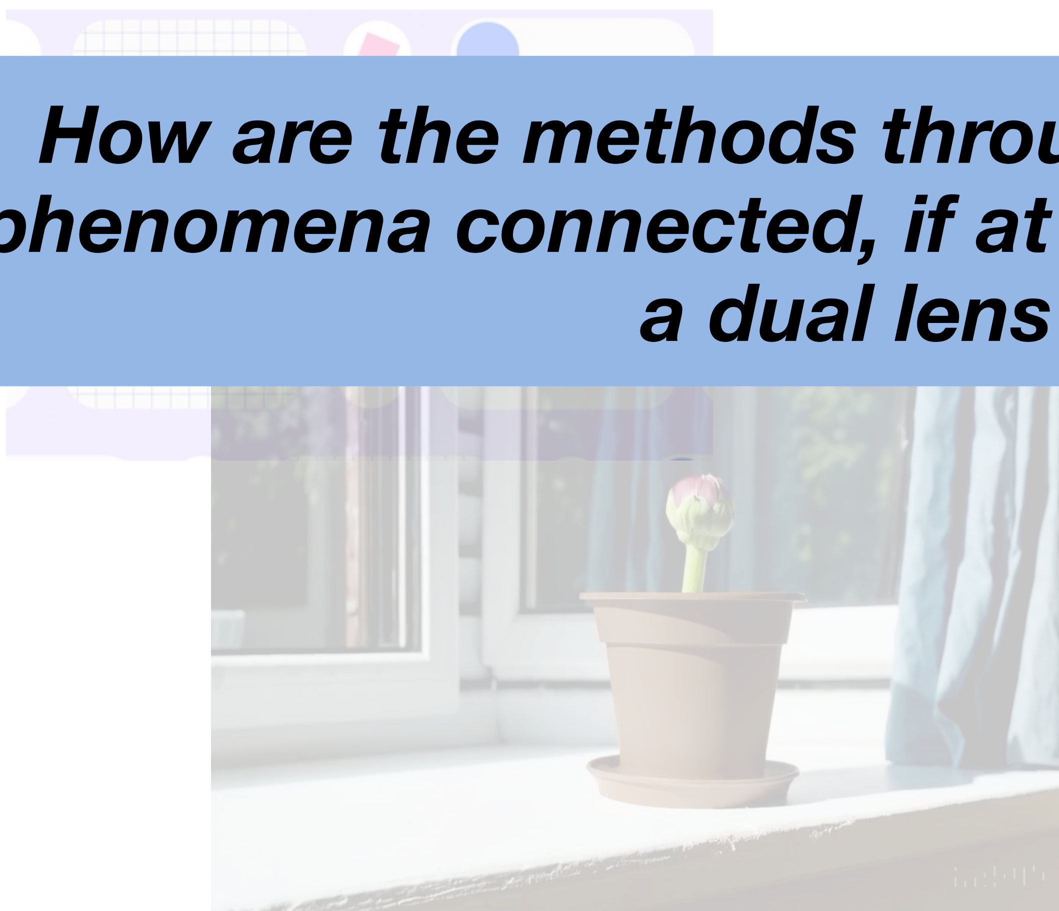
Problem Motivation: Complexity all around

The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

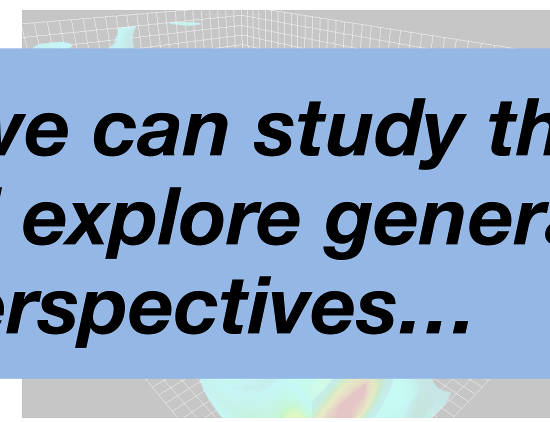
Social: abundance of data

Natural: limited data, but theory

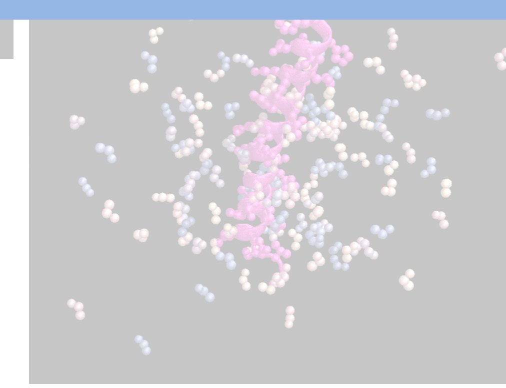
How are the methods through which we can study these disparate phenomena connected, if at all? We will explore generative models as a dual lens to these perspectives…



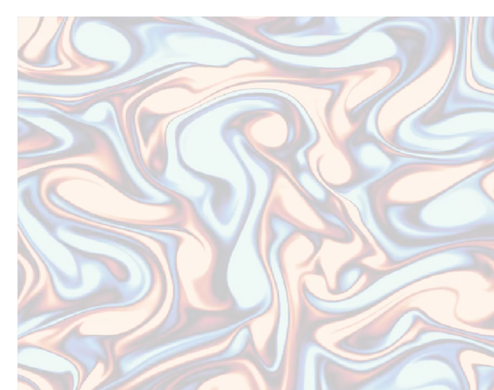
Sora (2024): "A flower growing out on the windowsill"



Quantum Theory



Molecular conformation

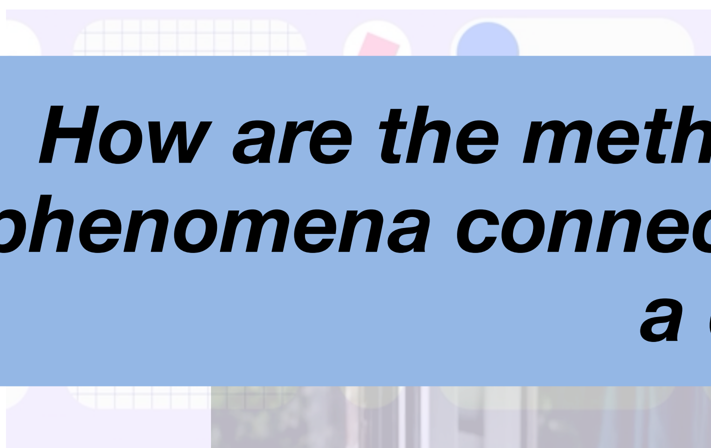


Forecasting

Problem Motivation: Complexity all around

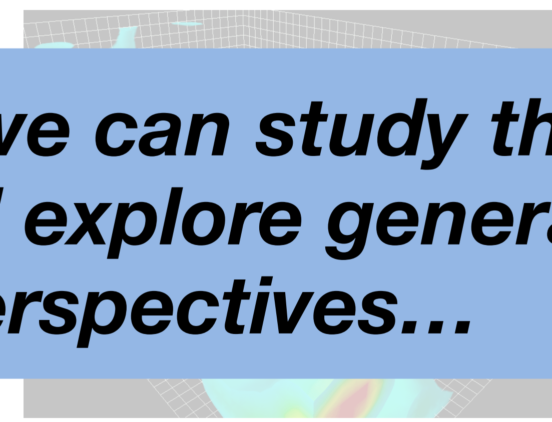
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

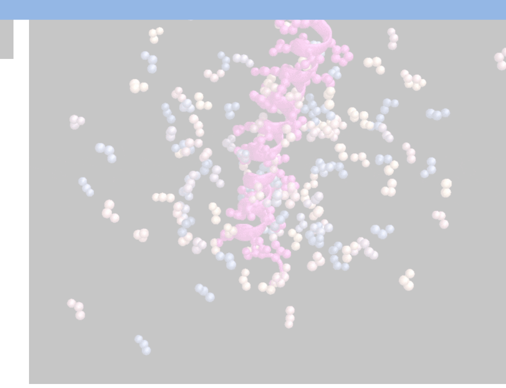


Sora (2024): "A flower growing out on the windowsill"

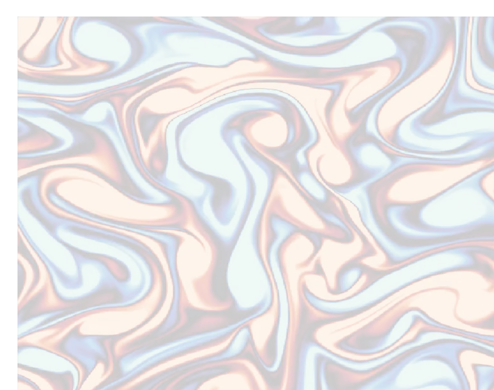
Natural: limited data, but theory



Quantum Theory



Molecular conformation



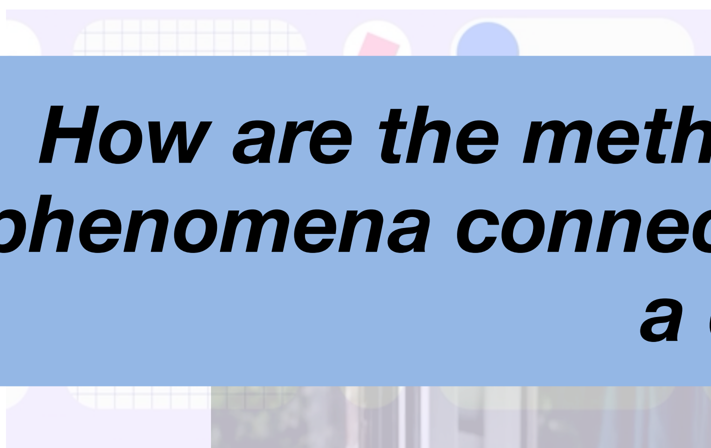
Forecasting

How are the methods through which we can study these disparate phenomena connected, if at all? We will explore generative models as a dual lens to these perspectives…

Problem Motivation: Complexity all around

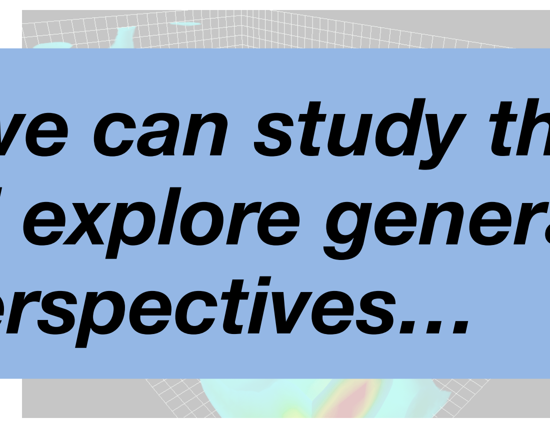
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

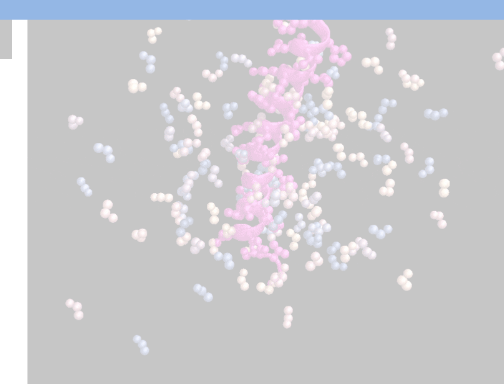


Sora (2024): “A flower growing out on the windowsill”

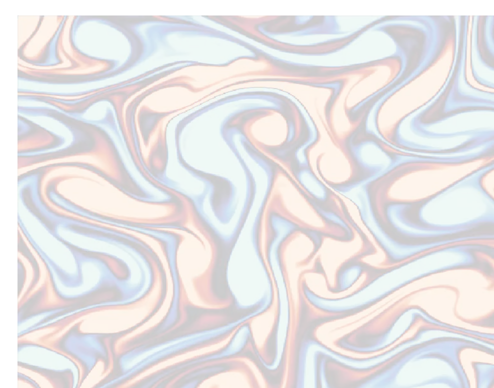
Natural: limited data, but theory



Quantum Theory



Molecular conformation



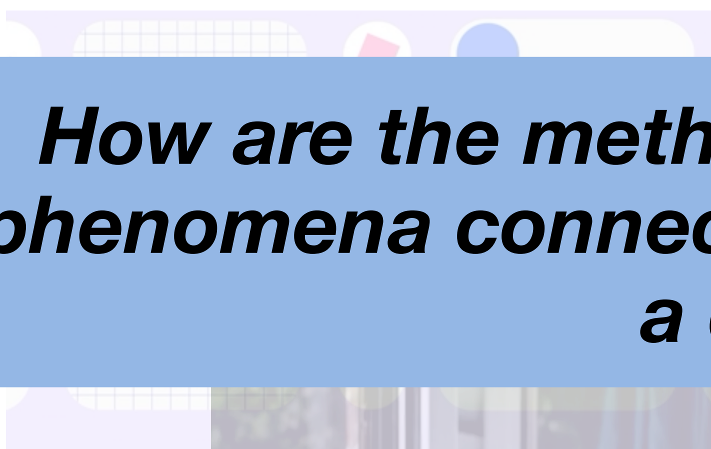
Forecasting

How are the methods through which we can study these disparate phenomena connected, if at all? We will explore generative models as a dual lens to these perspectives…

Problem Motivation: Complexity all around

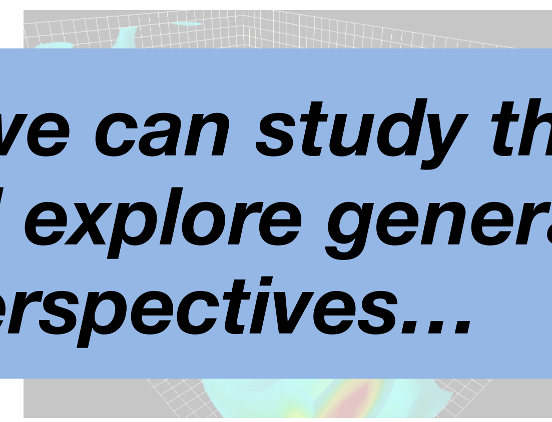
The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

Social: abundance of data

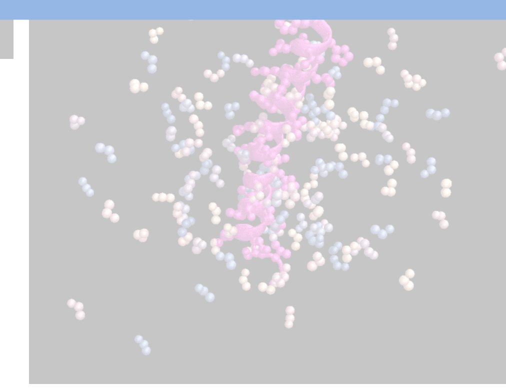


Sora (2024): "A flower growing out on the windowsill"

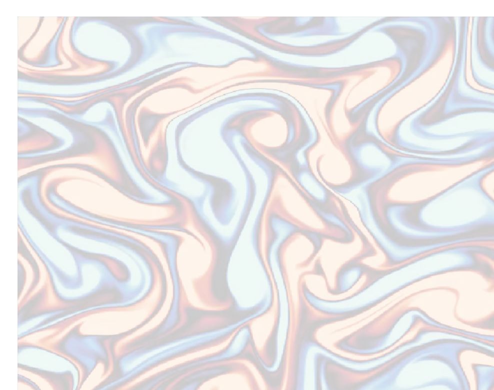
Natural: limited data, but theory



Quantum Theory



Molecular conformation



Forecasting

How are the methods through which we can study these disparate phenomena connected, if at all? We will explore generative models as a dual lens to these perspectives…

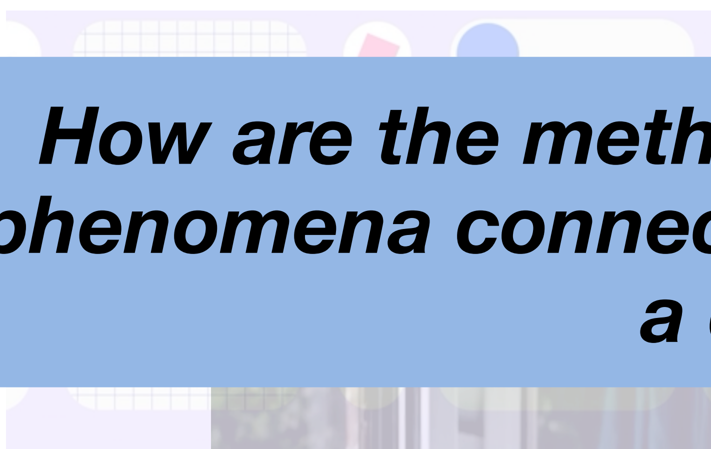
Problem Motivation: Complexity all around

The social and natural worlds are replete with complex structure that often has a probabilistic interpretation

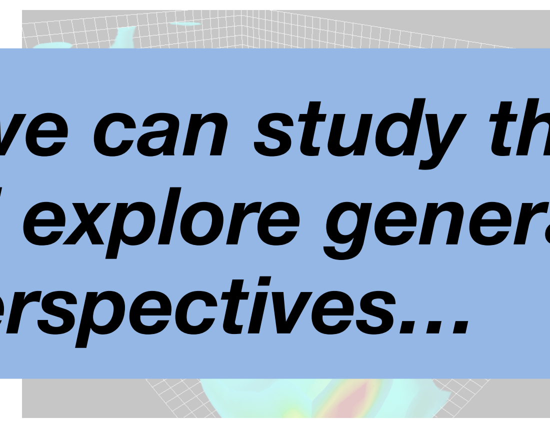
Social: abundance of data

Natural: limited data, but theory

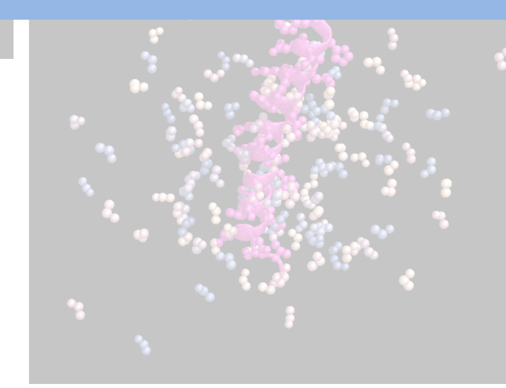
How are the methods through which we can study these disparate phenomena connected, if at all? We will explore generative models as a dual lens to these perspectives…



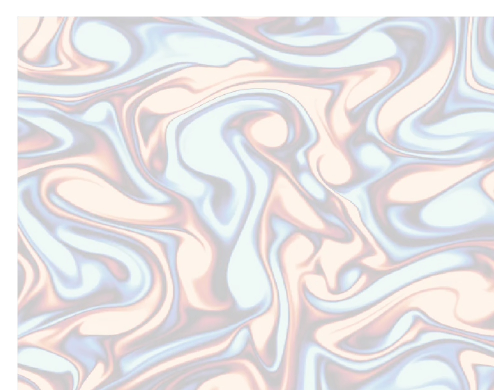
Sora (2024): "A flower growing out on the windowsill"



Quantum Theory



Molecular conformation



Forecasting

Framing the dual probabilistic problems

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

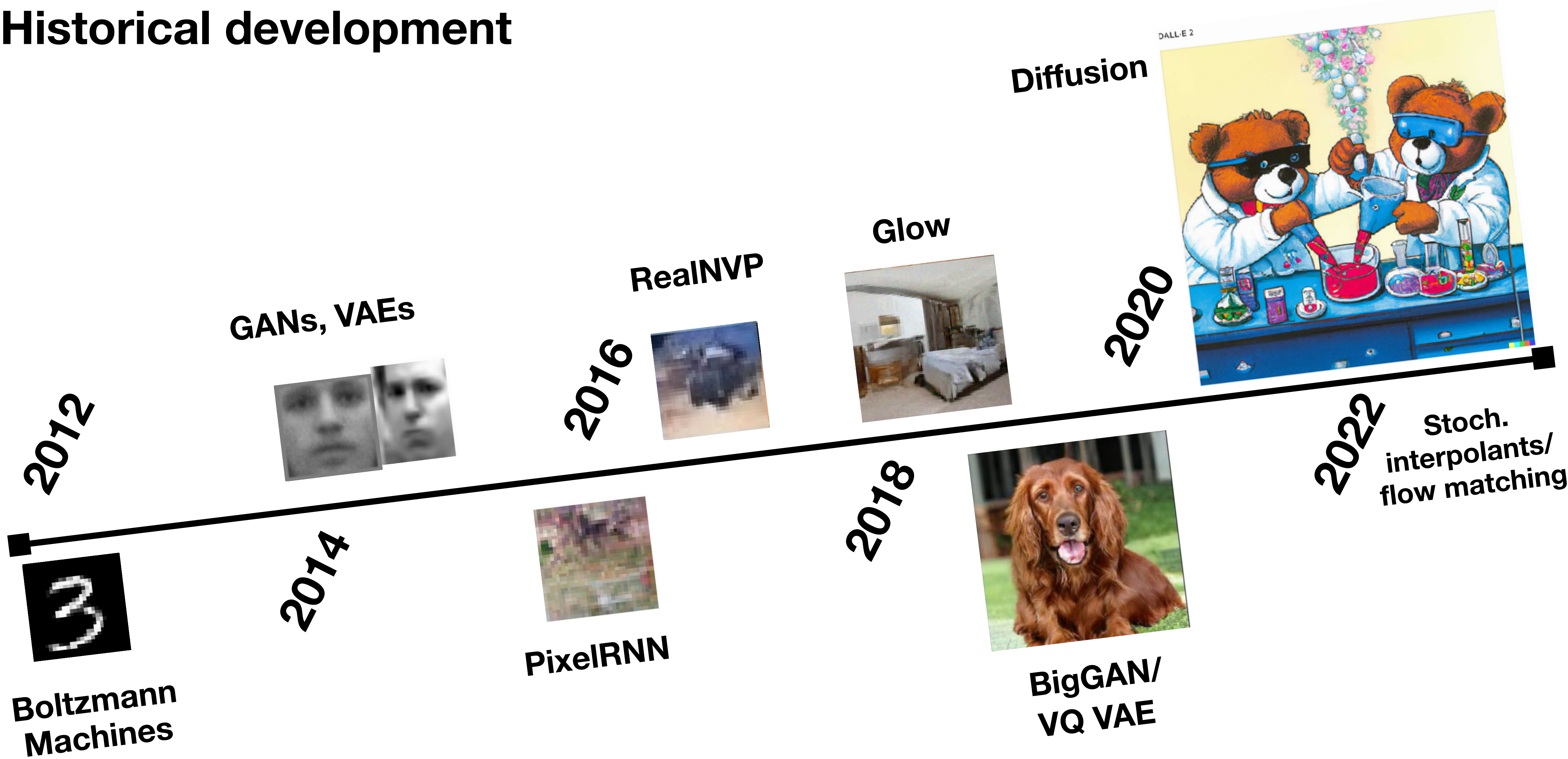
1. sample data $\{x_i\}_{i=1}^{n}$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Framing the dual probabilistic problems

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development

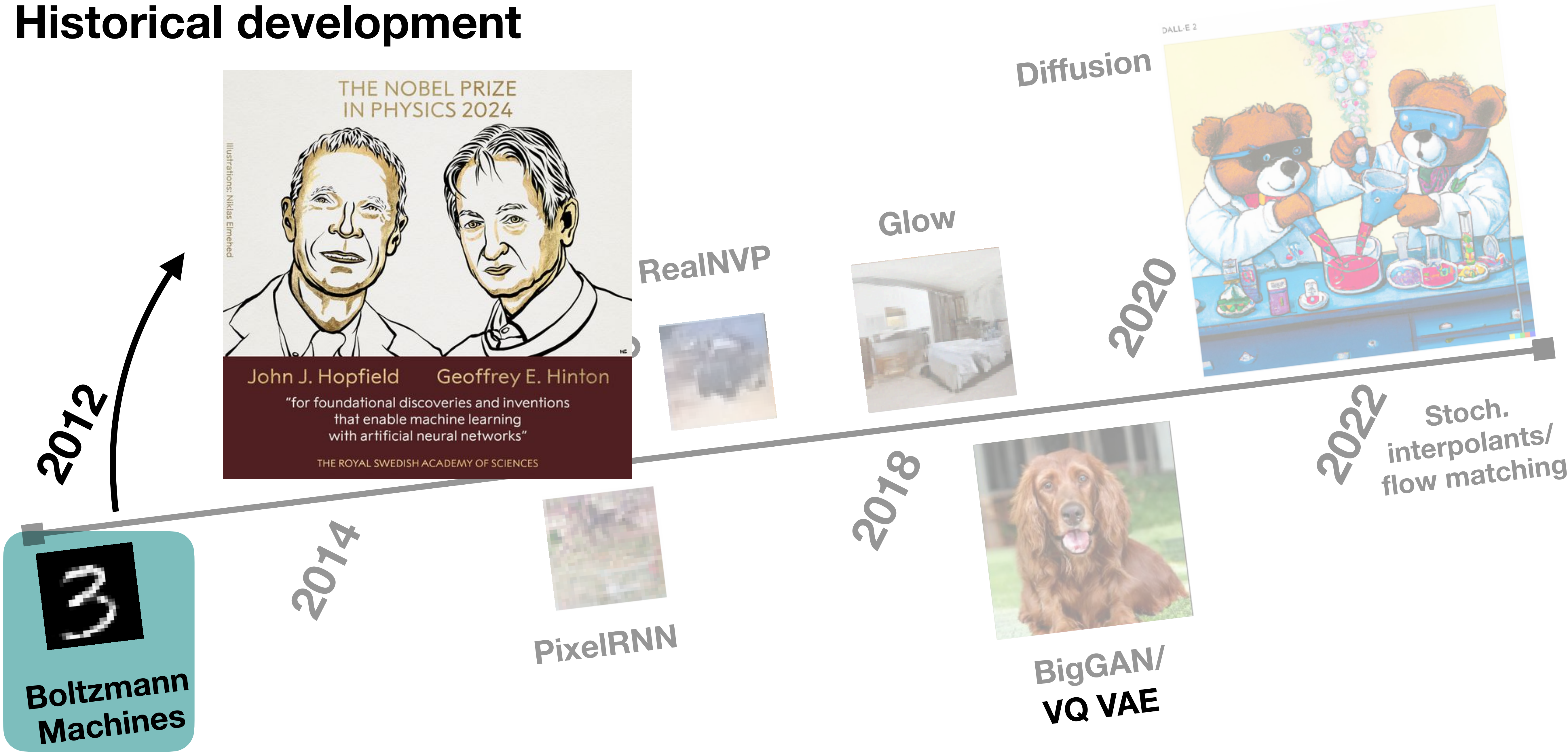


Framing the dual probabilistic problems

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^{n}$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development



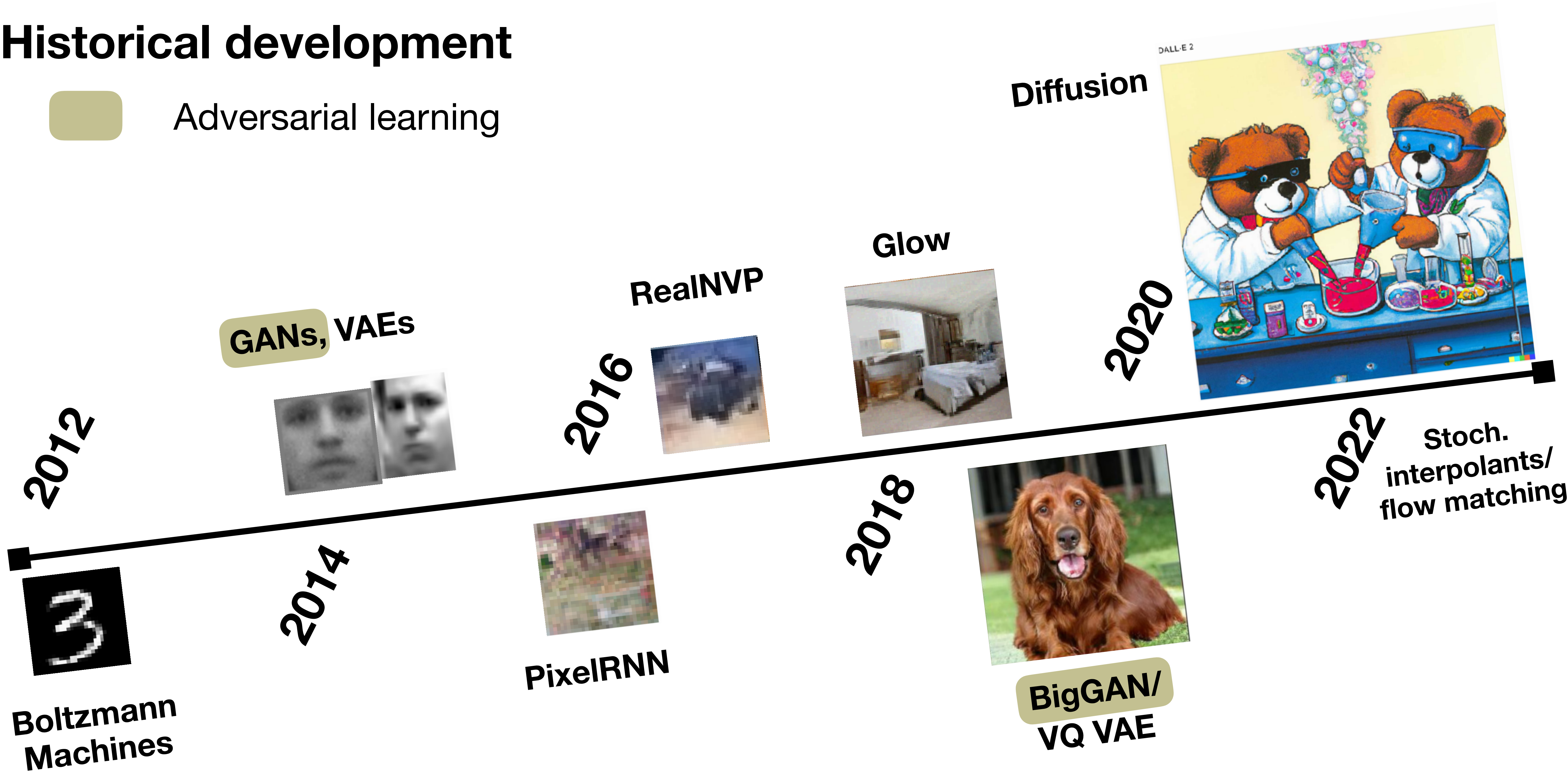
Framing the dual probabilistic problems

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ **(Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development

Adversarial learning



Framing the dual probabilistic problems

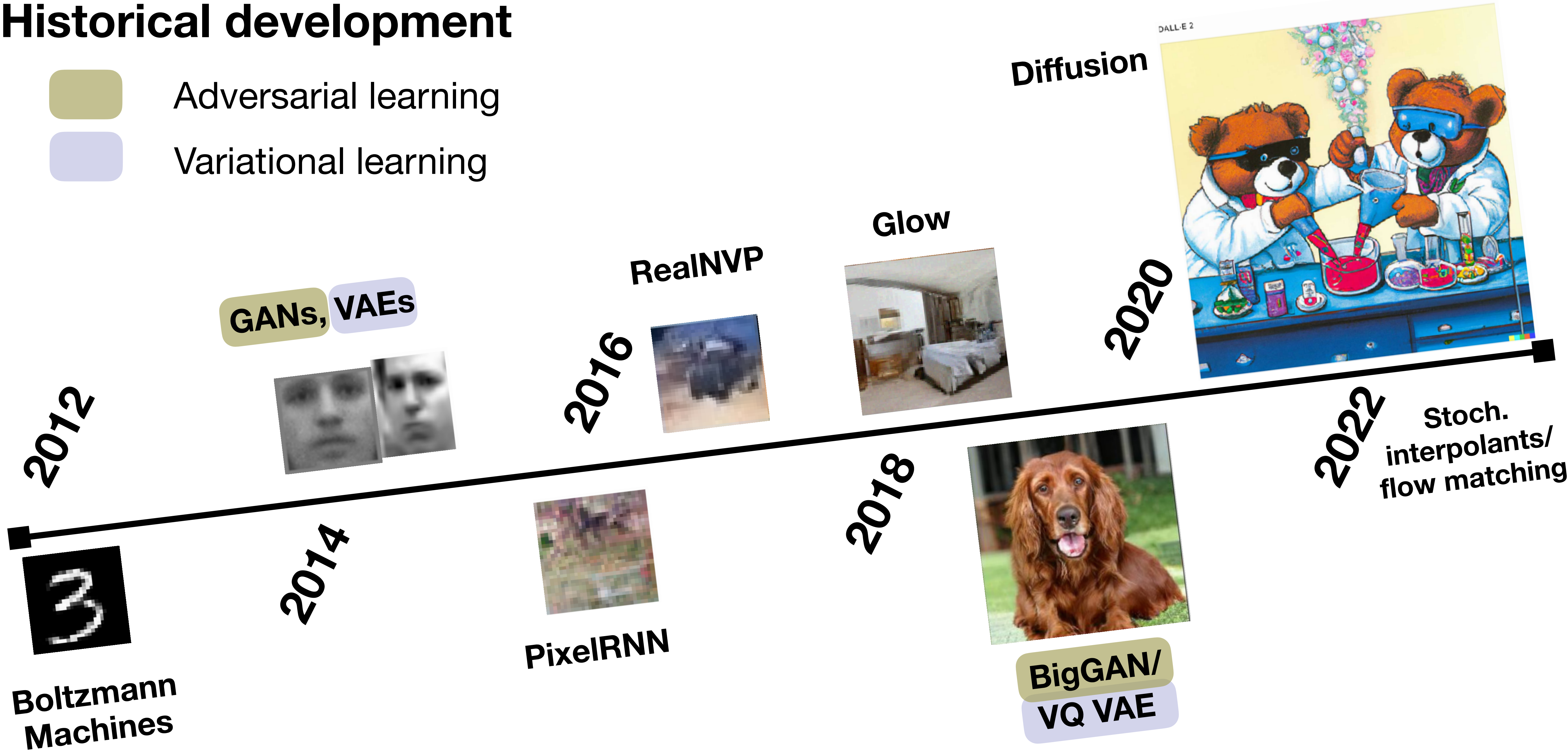
Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development

Adversarial learning

Variational learning

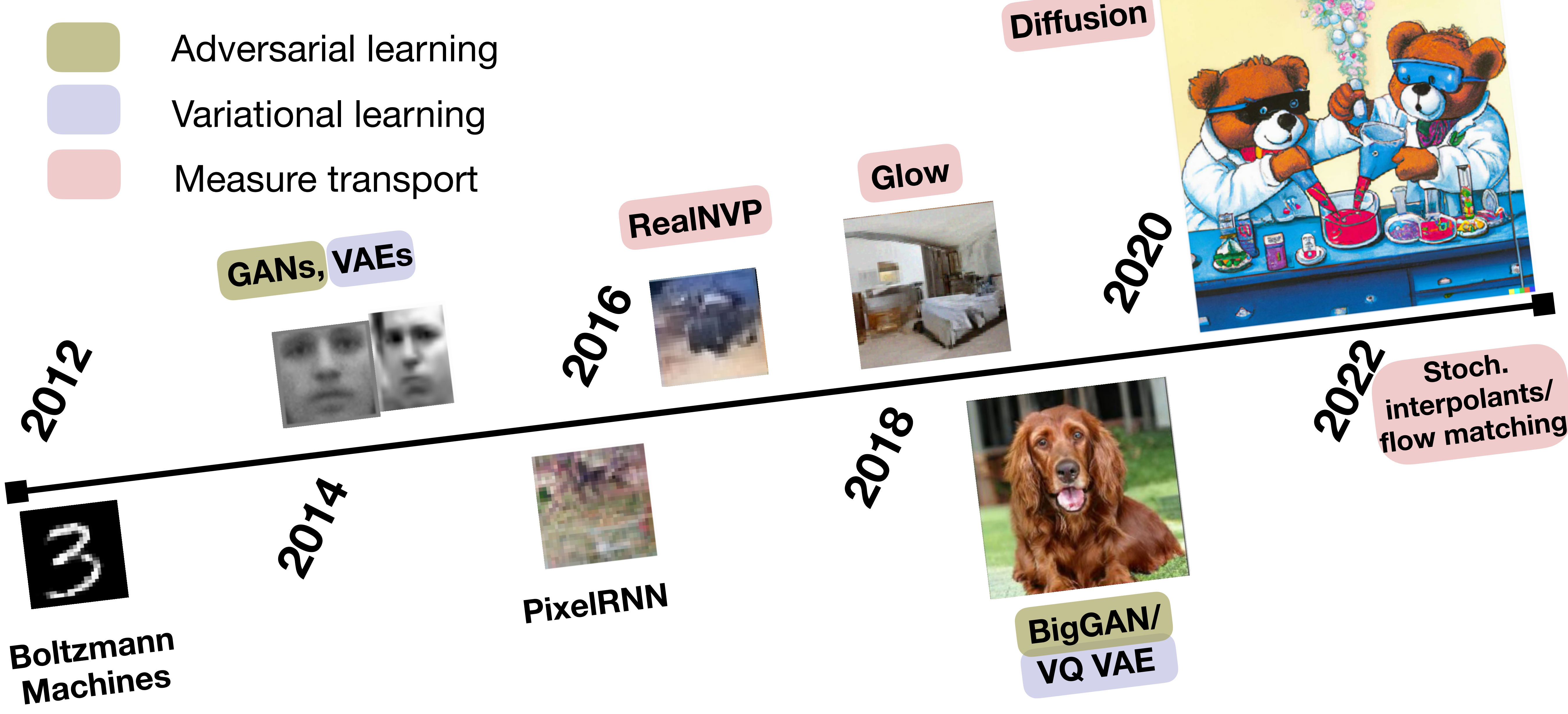


Framing the dual probabilistic problems

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development

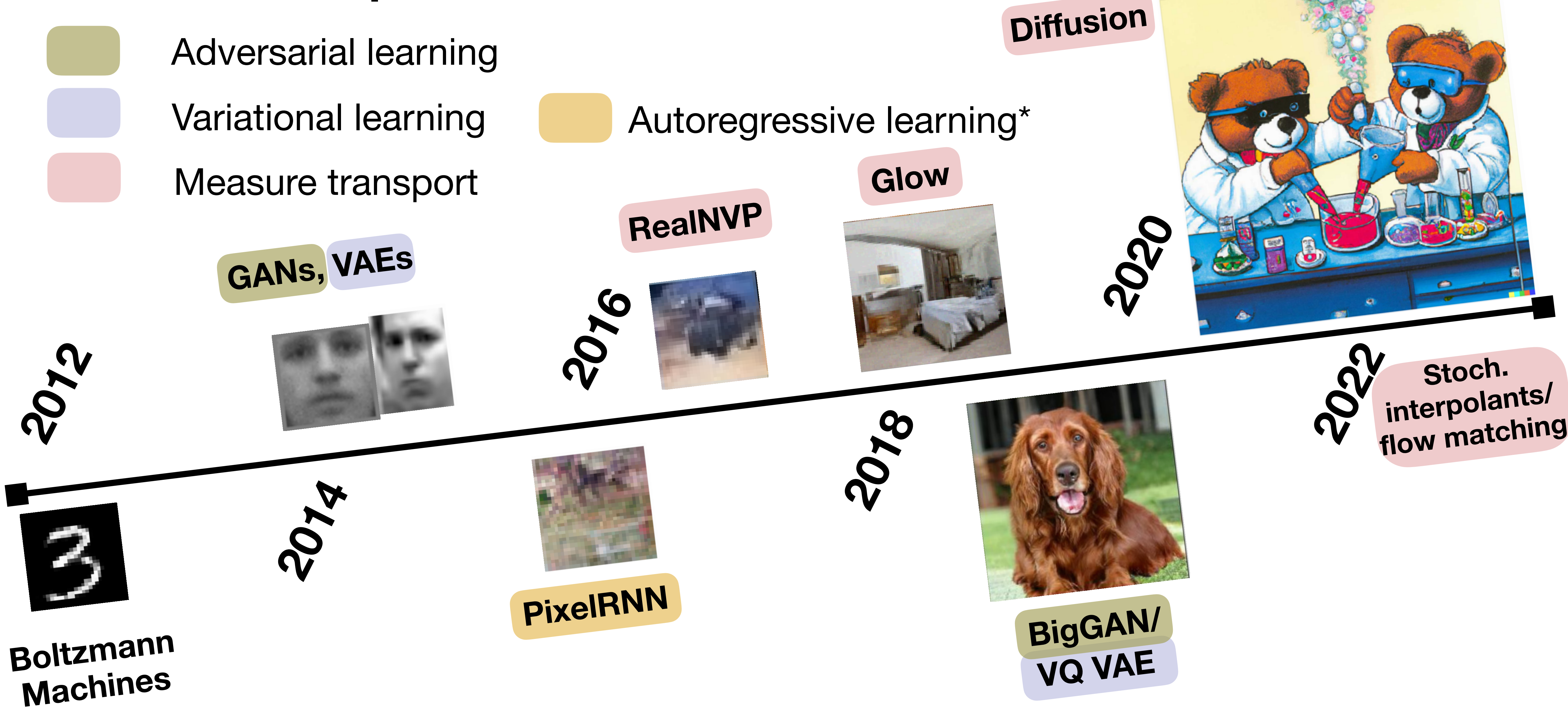


Framing the dual probabilistic problems

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathcal{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development



4 perspectives that dominate contemporary GM

Agenda

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ (**Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development

Adversarial learning

Variational learning

Measure transport

Autoregressive learning*

Agenda

Goal: estimate the unknown *probability density function* $\rho_1 \in \mathscr{D}(\Omega)$ either through:

1. sample data $\{x_i\}_{i=1}^n$ **(Generative modeling)**
2. query access to the unnormalized log likelihood **(Sampling)**

Historical development

Adversarial learning

Variational learning

Measure transport

Autoregressive learning*

A quick introduction to each of these topics

- a retrospective on the pros/cons of each, and what we've learned from these various perspectives
- how aspects of each of these tools are used today, in form or another!

My claim: ultimately, we evaluate these methods on measure theoretic quantities, and we should therefore being building tools from the measure transport perspective. There's a lot of evidence of this now!

Generative Adversarial Learning (2014)

Implicit Generative Model

Picture this: It's 2014 and standard approaches to optimizing your generative models (maximum likelihood estimation) are hard!

Two player game idea: what if I instead have two neural networks train each other?

learn to sample ρ_1 with generator $\hat{G}(z) = \hat{x}_1 \sim \hat{\rho}_1$

learn to discriminate real samples from fake $\hat{D}(x$ or $\hat{x})$

$$G : \mathbb{R}^n \rightarrow \mathbb{R}^d \qquad D : \mathbb{R}^d \rightarrow [0,1]$$

Generative Adversarial Learning (2014)

Implicit Generative Model

Picture this: It's 2014 and standard approaches to optimizing your generative models (maximum likelihood estimation) are hard!

Two player game idea: what if I instead have two neural networks train each other?

learn to sample ρ_1 with generator $\hat{G}(z) = \hat{x}_1 \sim \hat{\rho}_1$

learn to discriminate real samples from fake $\hat{D}(x$ or $\hat{x})$

Learning:

$$L[\hat{G}, \hat{D}] = \min_{\hat{G}} \max_{\hat{D}} \mathbb{E}_{\rho_1}[\log \hat{D}(x_1)] + \mathbb{E}_{\hat{\rho}_1}[\log(1 - D(\hat{x}_1))]$$

Discriminator maximizes: wants $\hat{D}(x_1) = 1$ and $\hat{D}(\hat{x}_1) = 0$

Generator minimizes: wants $\hat{D}(\hat{x}_1) = 1$ (tricks discriminator)

Generative Adversarial Learning (2014)

Implicit Generative Model

Learning:

$$L[\hat{G}, \hat{D}] = \min_{\hat{G}} \max_{\hat{D}} \mathbb{E}_{\rho_1}[\log \hat{D}(x_1)] + \mathbb{E}_{\hat{\rho}_1}[\log(1 - D(\hat{x}_1))]$$

Discriminator maximizes: wants $\hat{D}(x_1) = 1$ and $\hat{D}(\hat{x}_1) = 0$

Generator minimizes: wants $\hat{D}(\hat{x}_1) = 1$ (tricks discriminator)

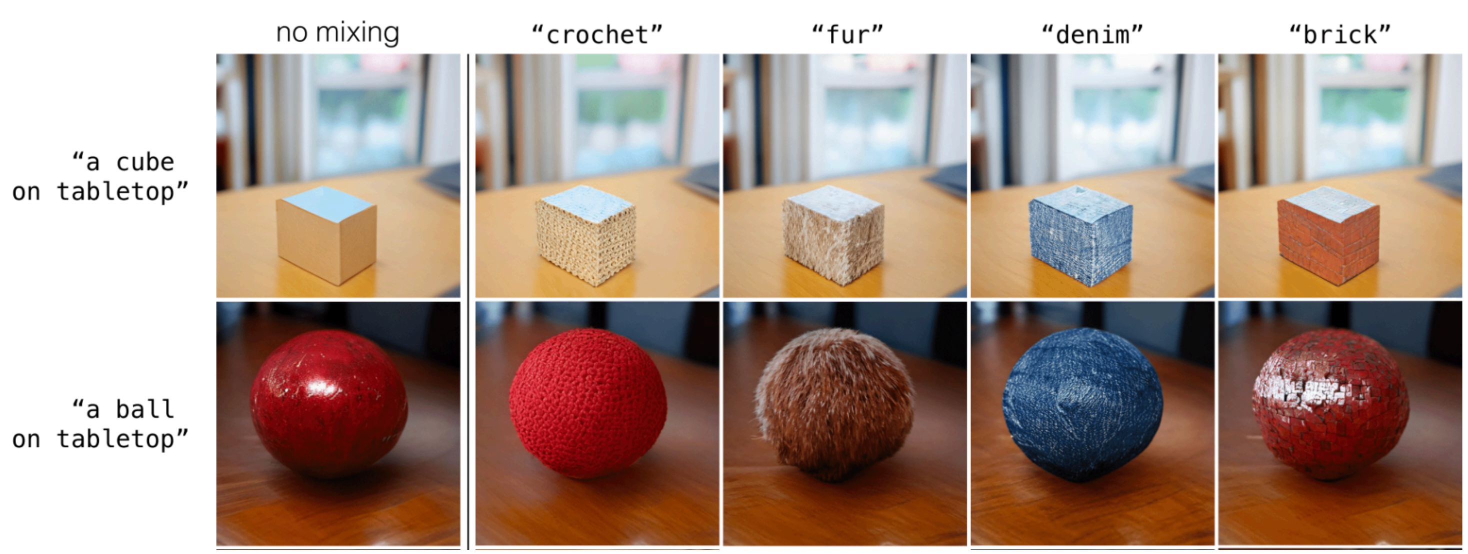
A theoretically motivated minimax game:

- If $\hat{D}$ can represent any function, then finding G^* amounts to minimizing a Jensen-Shannon divergence (like symmetrized KL)
- Lots of research into changing the "$\log$" functions to minimize other divergences!
- Allows scale for probabilistic modeling "**without likelihoods**"

Benefits and Challenges in GAN learning

Fast, expressive sampling — One step, unstructured maps

Interpretable latent space — Not diffeomorphisms, so latent space meaningfully lower dimensional



Minimax optimization — Learning can be unstable because of sensitivity of equilibria in two-player game

Lots of follow-up research into this!

No explicit likelihood — Likelihoods are preferable for science!

GAN Outlook

Nonetheless, can still be remarkably powerful when tuned carefully

https://mingukkang.github.io/GigaGAN/ (2023)

A photo of a ramen taken from an angle, with some background.

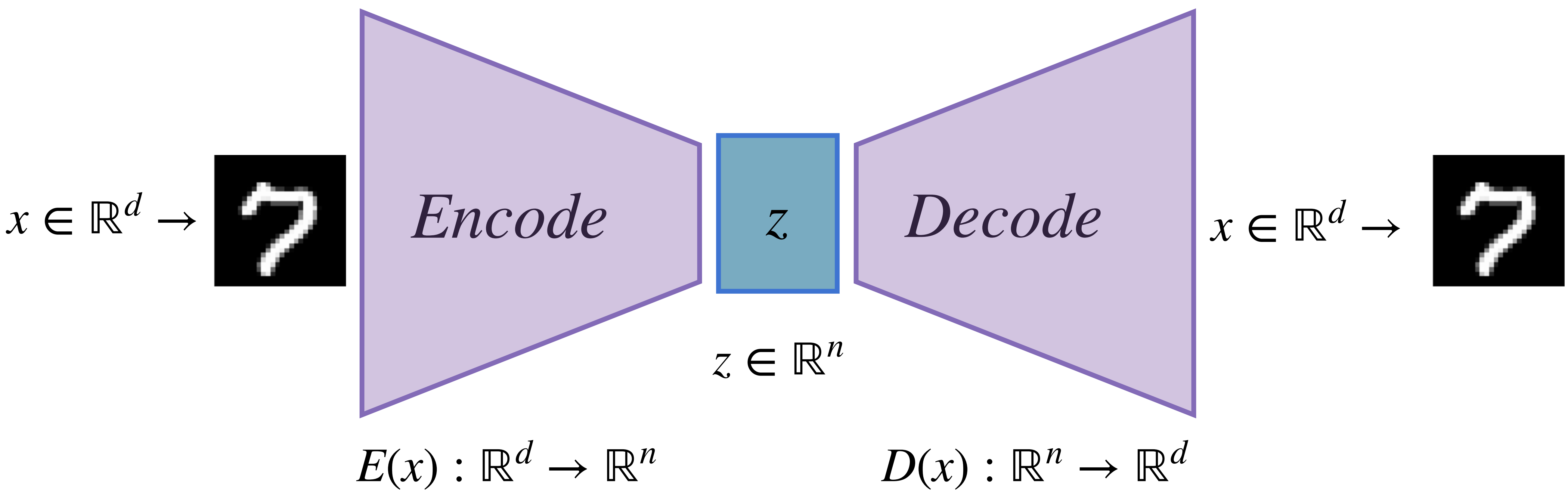
Images generated in 0.13 seconds!

Variational Learning

Variational Autoencoders: Making auto-encoding probabilistic!

Representation Learning *Generative modeling*

Autoencoding framework: encode images to a lower dim representation z



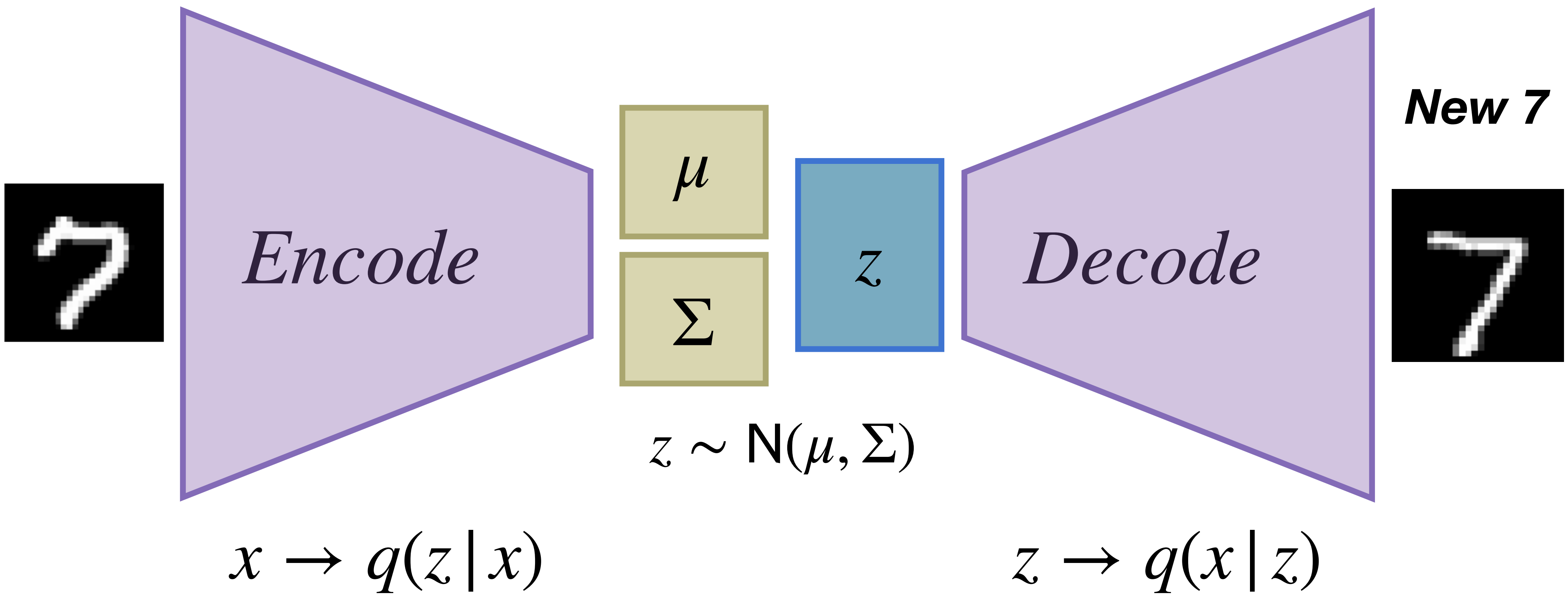
Useful for representation learning! ***How to make it probabilistic?***

Variational Learning

Variational Autoencoders: Making auto-encoding probabilistic!

Representation Learning *Generative modeling*

Variational framework: encode a posterior distribution $q(z \mid x)$ for each input x



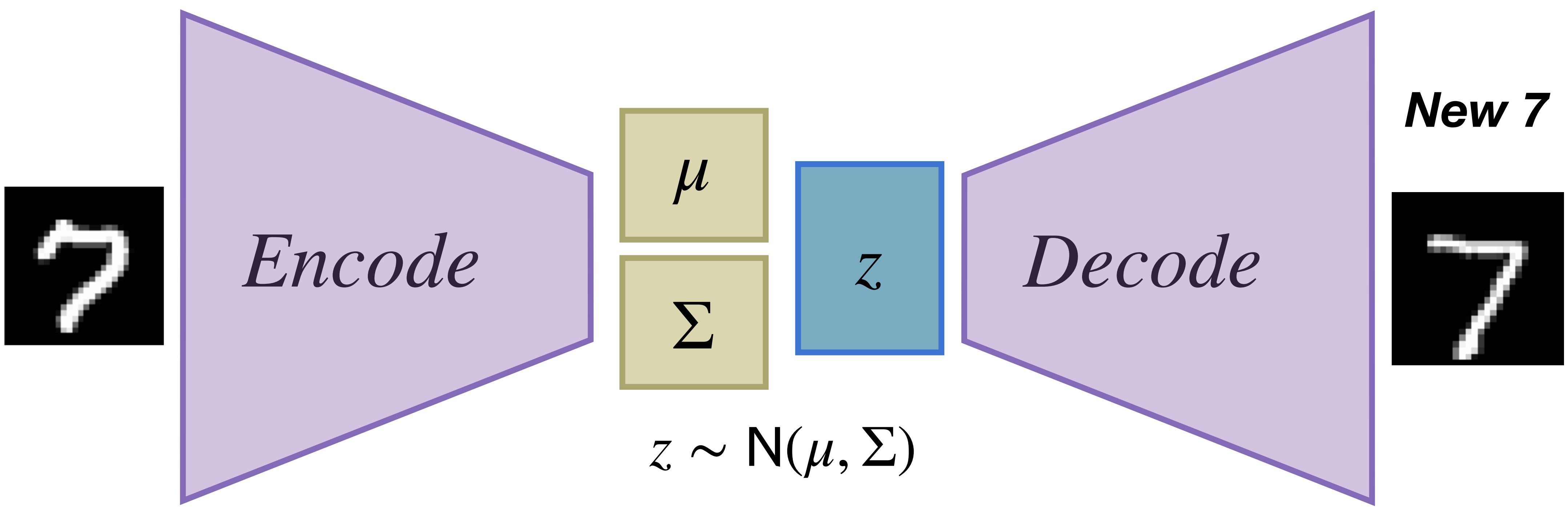
Reconstruct original input, but **regularize** latent space to be **Gaussian** so you can sample a space with structure!

Variational Learning

Variational Autoencoders: Making auto-encoding probabilistic!

Representation Learning *Generative modeling*

$$\min \mathbb{E}_{q(z|x)}[\,|x - D(z)|^2\,] + D_{KL}\big(q(z\,|\,x)\,||\,p(z\,|\,x)\big)$$



$z \sim \mathrm{N}(\mu, \Sigma)$

$x \rightarrow q(z\,|\,x)$ $z \rightarrow q(x\,|\,z)$

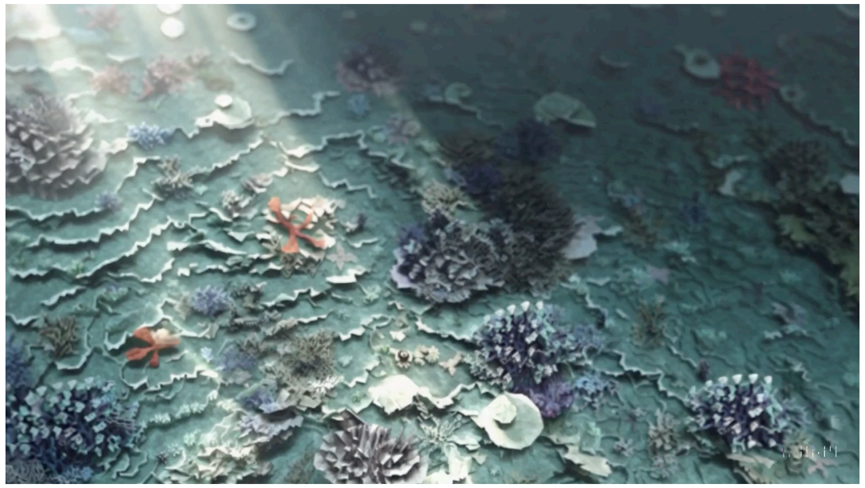
Reconstruct original input, but **regularize** latent space to be **Gaussian** so you can sample a space with structure!

Benefits and challenges

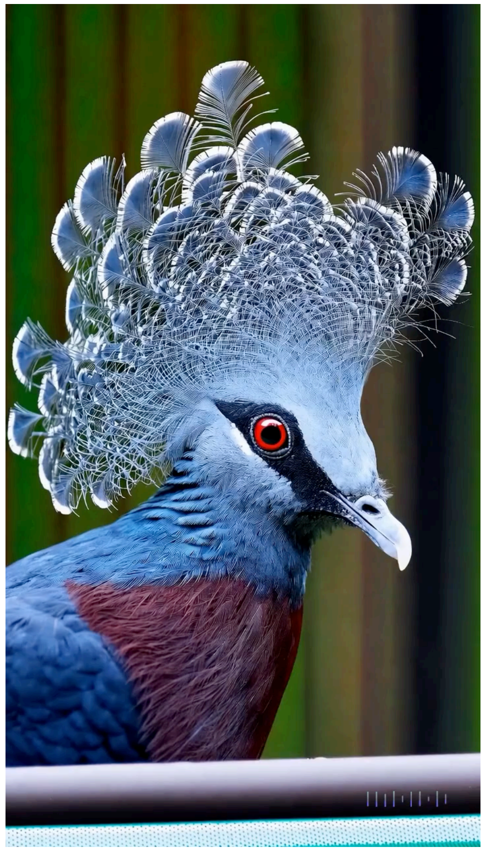
Rich latent representations

Generative modeling in latent space an essential ingredient for large scale methods

tons of research into improving latent representations



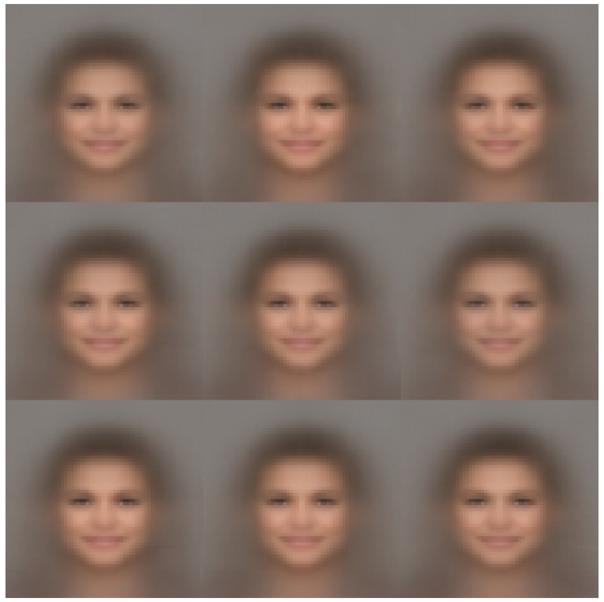
Sora: Origami sea creatures



Sora: Victoria-crowned pigeon

Subpar generative models on their own

trade-offs between image reconstruction (expressivity) and latent space structure

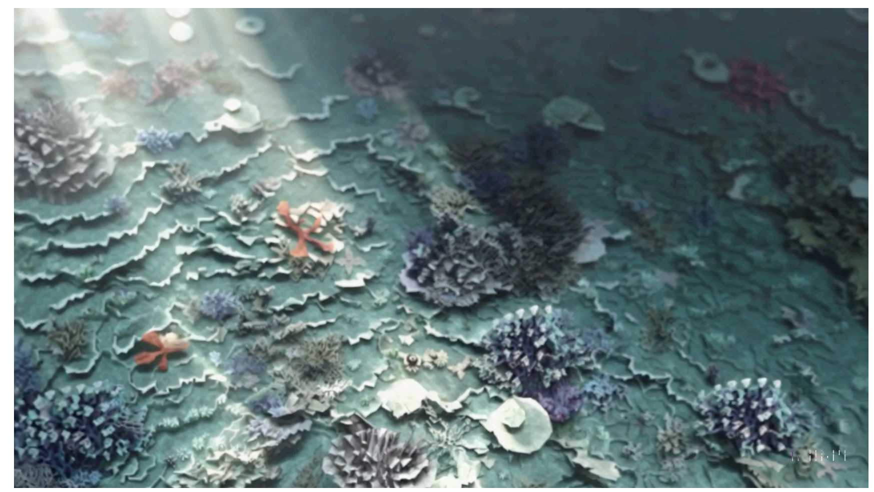


Benefits and challenges

Rich latent representations

Generative modeling in latent space an essential ingredient for large scale methods

tons of research into improving latent representations

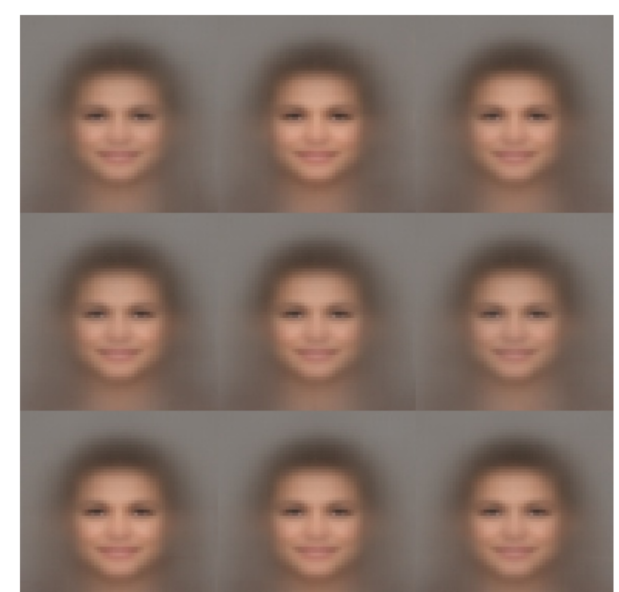


Sora: Origami sea creatures

Sora: Victoria-crowned pigeon

Subpar generative models on their own

trade-offs between image reconstruction (expressivity) and latent space structure

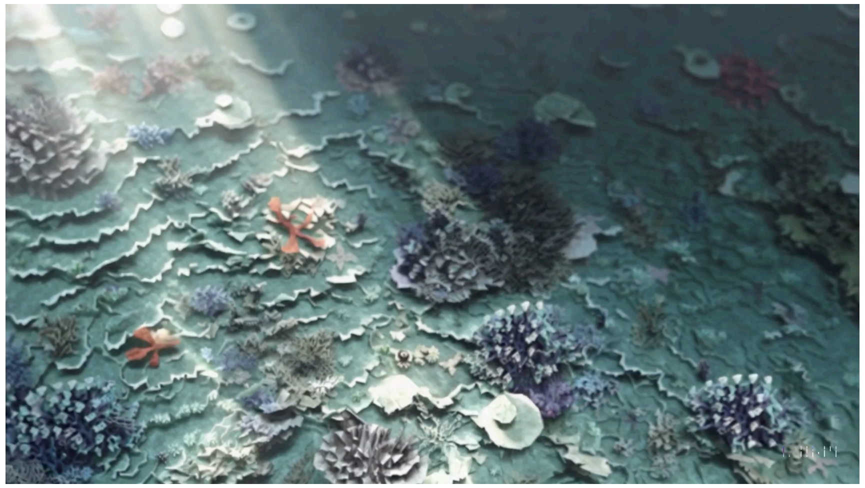


Benefits and challenges

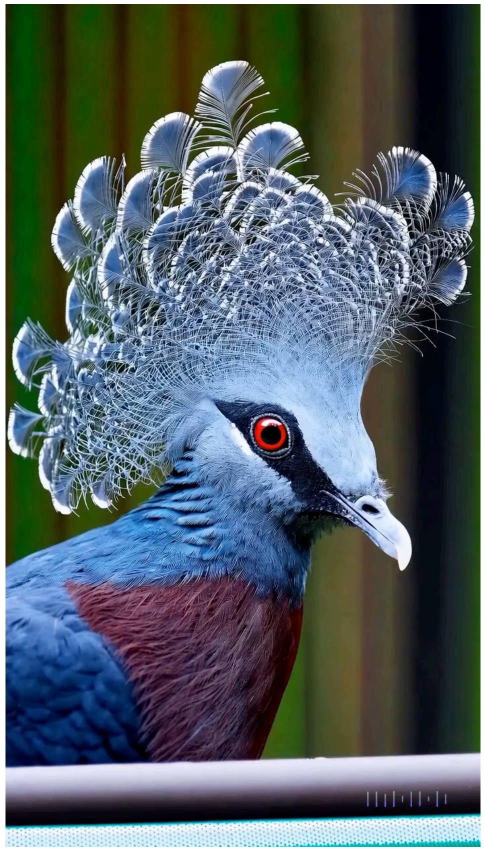
Rich latent representations

Generative modeling in latent space an essential ingredient for large scale methods

tons of research into improving latent representations



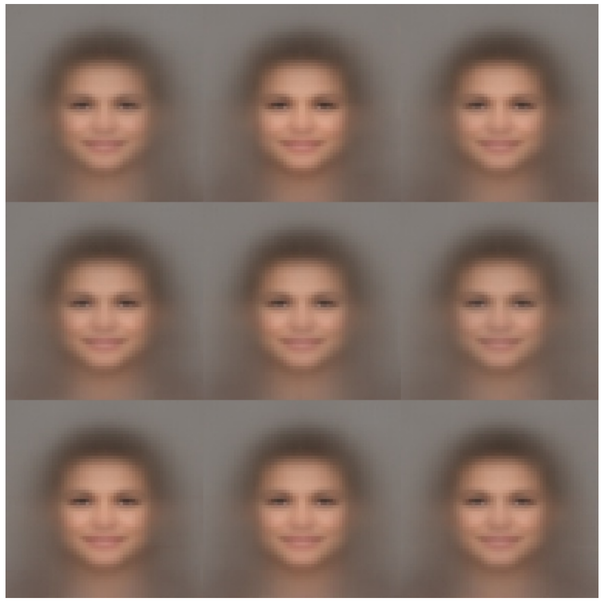
Sora: Origami sea creatures



Sora: Victoria-crowned pigeon

Subpar generative models on their own

trade-offs between image reconstruction (expressivity) and latent space structure



Benefits and challenges

Rich latent representations

Generative modeling in latent space an essential ingredient for large scale methods

We should be thinking about when and how to best use latent generative modeling in science — structuring these latent spaces is really different in these domains, and under explored!

Sora: Origami sea creatures

Subpar generative models on their own

trade-offs between image reconstruction (expressivity) and latent space structure

Benefits and challenges

Rich latent representations

Generative modeling in latent space an essential ingredient for large scale methods

We should be thinking about when and how to best use latent generative modeling in science — structuring these latent spaces is really different in these domains, and under explored!

Sora: Origami sea creatures

Subpar generative models on their own

trade-offs between image reconstruction (expressivity) and latent space structure

Benefits and challenges

Rich latent representations

Generative modeling in latent space an essential ingredient for large scale methods

We should be thinking about when and how to best use latent generative modeling in science — structuring these latent spaces is really different in these domains, and under explored!

Sora: Origami sea creatures

Subpar generative models on their own

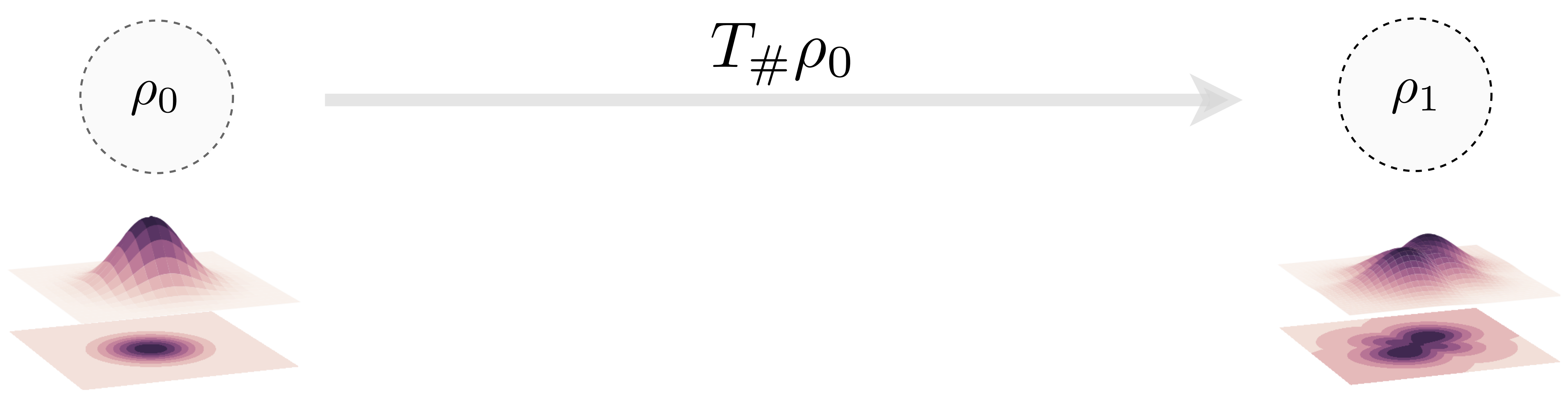
trade-offs between image reconstruction (expressivity) and latent space structure

Flows and Diffusions: Problem Setup

A direct maximum likelihood approach?

The transport framework

- Take a simple *base density* ρ_0 (e.g. Gaussian) and;
- Build a (reversible) map $T : \Omega \to \Omega$ such that the *pushforward of* ρ_0 *by* T *is* ρ_1*:* $T\sharp\rho_0 = \rho_1$

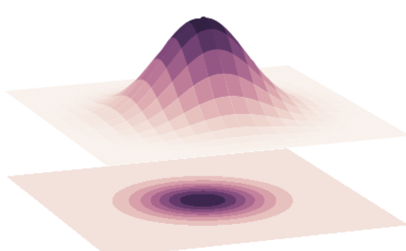


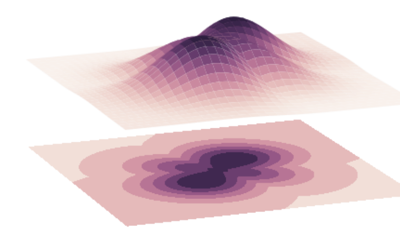
Likelihood under $\rho(1)$ given by: $\rho_1(x_1) = \rho_0(T^{-1}(x))\,\det[\nabla T^{-1}(x)]$

The transport framework

- Build a (reversible) map $T : \Omega \to \Omega$ such that the *pushforward of* $\rho(0)$ *by* T *is* $\rho(1)$*:* $T\sharp\rho(0) = \rho(1)$

$$\rho_0 \xrightarrow{T_{\#}\rho_0} \rho_1$$





Likelihood: $\rho_1(x) = \rho_0(T^{-1}(x)) \det[\nabla T^{-1}(x)]$

For parametric $\hat{T}(x)$ to be useful

- $\det[\nabla \hat{T}^{-1}(x)]$ to be **tractable**
- $\hat{T}(x)$ **maximally unconstrained**

Tradeoff!

Problem Setup

The transport framework

- Build a (reversible) map $T : \Omega \to \Omega$ such that the *pushforward of* $\rho(0)$ *by* T *is* $\rho(1)$: $T\sharp\rho(0) = \rho(1)$

$$\rho_0 \xrightarrow{T_\#\rho_0} \rho_1$$

How do we harness measure transport for these various tasks in probabilistic modeling? How do we learn these maps?

Ex. Image generation
Ex. Statistical physics

Ex. Translation

Ex. Climate/weather
Ex. Dynamical systems

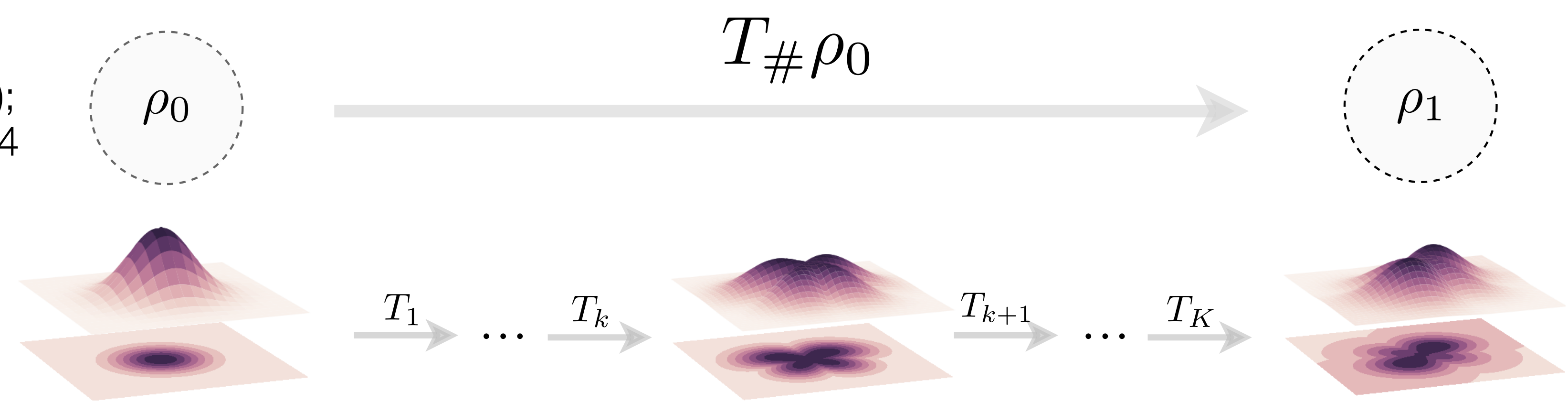
Brief history on transport realizations

Series of discrete transforms

$\det[\nabla T^{-1}(x)]$ tractable, but too constrained?

T_k learned sequentially

Chen & Gopinath, NeurIPS 13 (2000);
Tabak & V.-E., Commun. Math. Sci. 8: 217-233 (2010);
Tabak & Turner, Comm. Pure App. Math LXVI, 145-164 (2013).



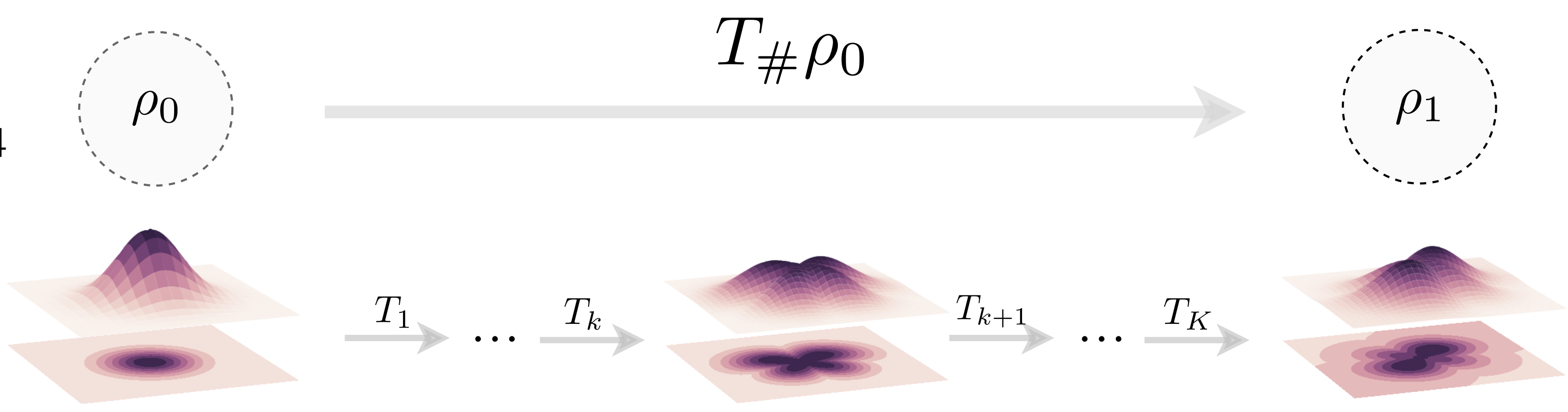
T_k structured invertible NNs

NICE: Dinh *et al.* arXiv:1410.8516 (2014);
Real NVP: Dinh *et al.* arXiv:1605.08803 (2016)
Rezende *et al.*, arXiv:1505.05770 (2015);
Papamakarios *et al.* arXiv:1912.02762 (2019); …

Brief history on transport realizations

Series of discrete transforms

$\det[\nabla T^{-1}(x)]$ tractable, but too constrained?



T_k learned sequentially

Chen & Gopinath, NeurIPS 13 (2000);
Tabak & V.-E., Commun. Math. Sci. 8: 217-233 (2010);
Tabak & Turner, Comm. Pure App. Math LXVI, 145-164 (2013).

T_k structured invertible NNs

NICE: Dinh *et al.* arXiv:1410.8516 (2014);
Real NVP: Dinh *et al.* arXiv:1605.08803 (2016)
Rezende *et al.*, arXiv:1505.05770 (2015);
Papamakarios *et al.* arXiv:1912.02762 (2019); …

$k \to \infty$

T solution of *continuous time flow*

FFJORD: Grathwohl *et al.* arXiv:1810.01367 (2018)

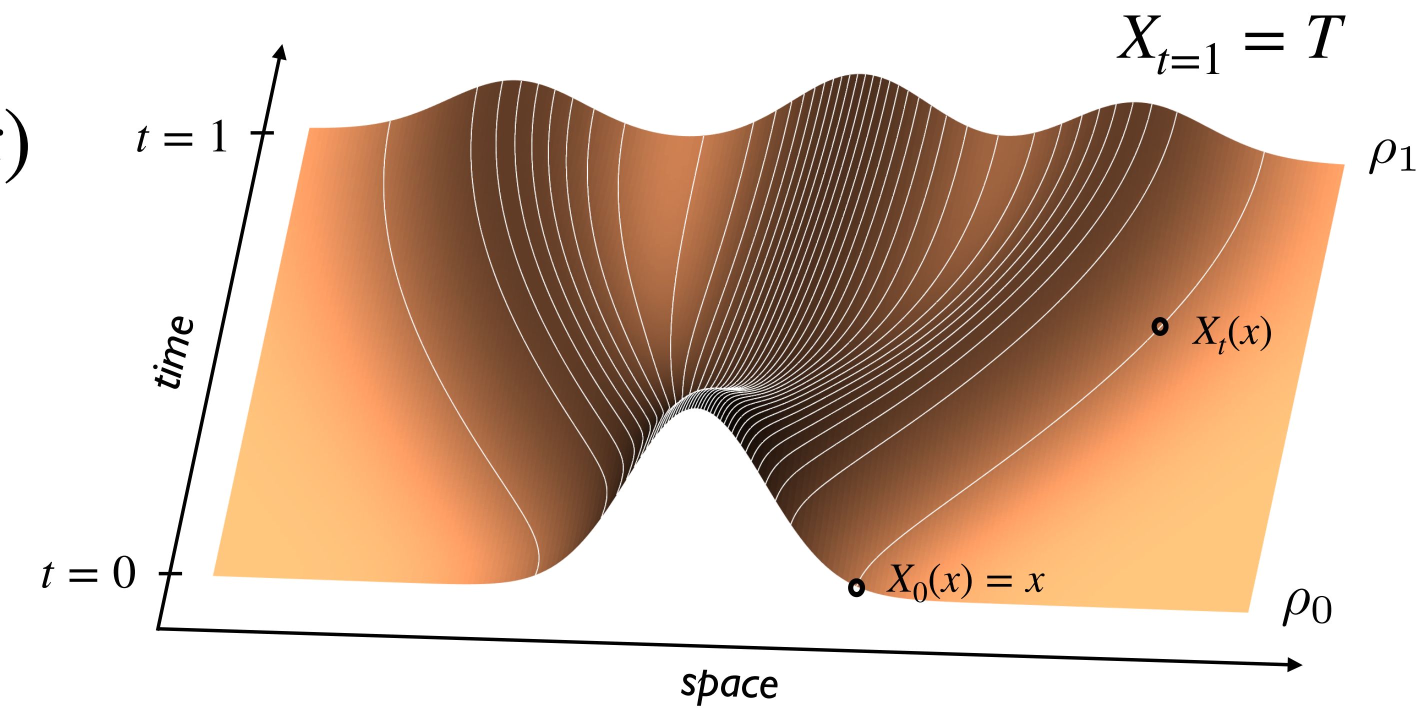
- $\det[\nabla T^{-1}(x)] \to \mathrm{Tr}[\frac{\partial b_t}{\partial x(t)}]$
- estimable via Skilling-Hutchinsion $O(D)$
- integrable with Neural ODEs

The continuous time picture

X_t flow map given by velocity field $b(t, x)$

$$X_{t=0}(x) = x \in \mathbb{R}^d$$

$$\dot{X}_t(x) = b(t, X_t(x))$$

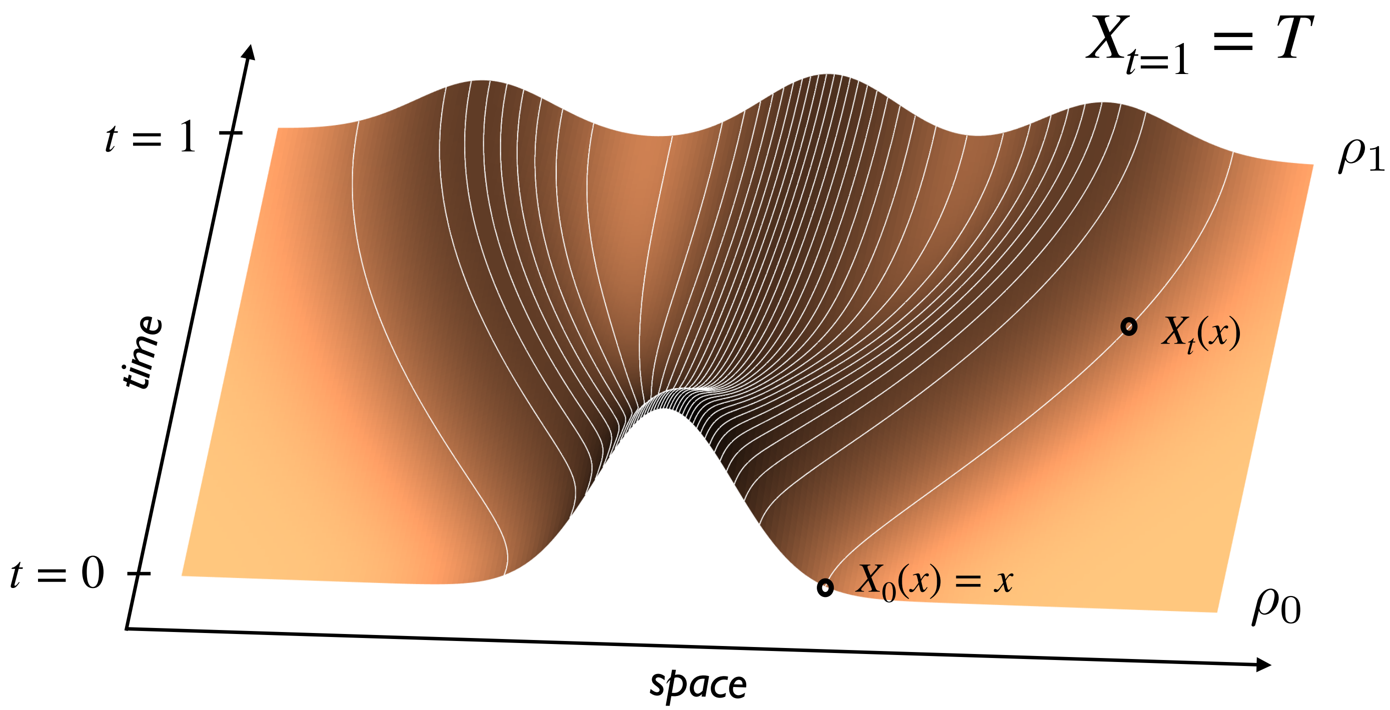


The continuous time picture

X_t flow map given by velocity field $b(t, x)$

$$X_{t=0}(x) = x \in \mathbb{R}^d$$

$$\dot{X}_t(x) = b(t, X_t(x))$$



At the level of the of the distribution, how does $\rho(t, x)$ evolve?

Transport equation

$$\partial_t \rho(t, x) + \nabla \cdot \big(b(t, x)\rho(t, x)\big) = 0, \quad \rho(t = 0, \cdot) = \rho_0$$

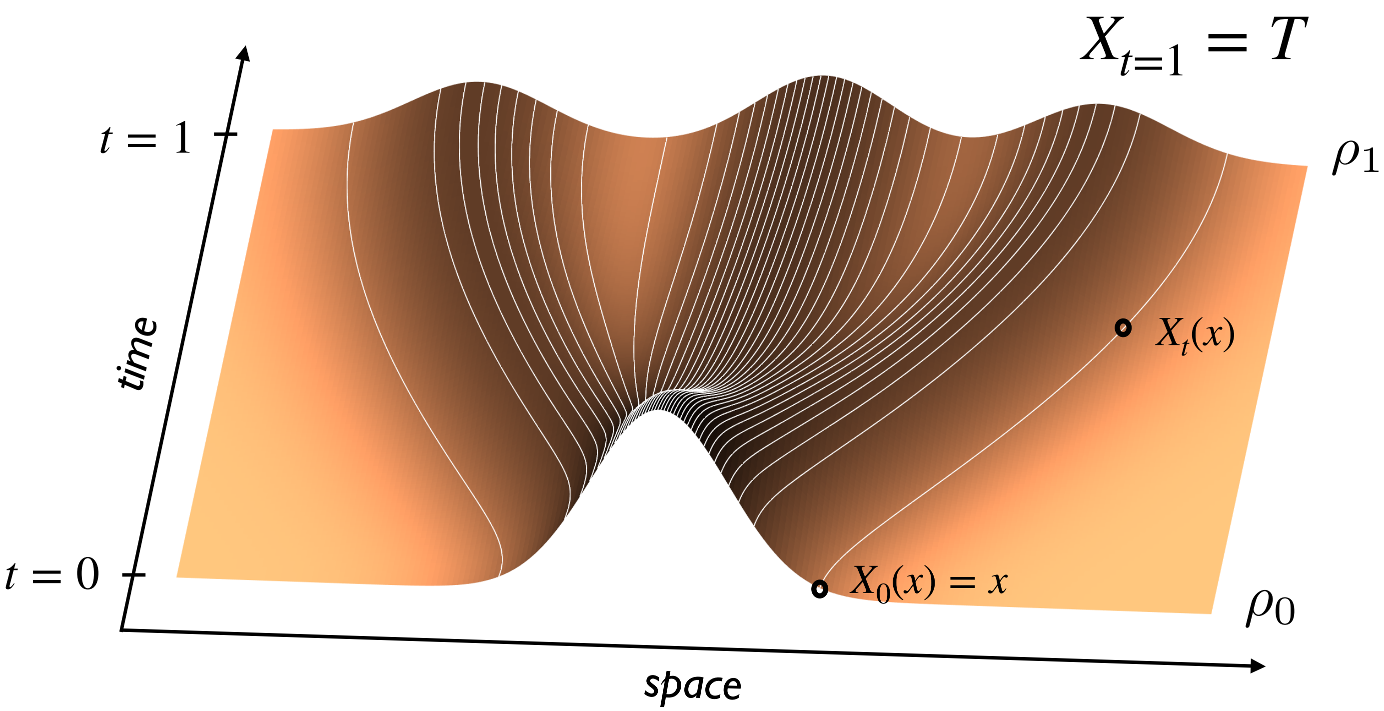
If $\rho(t)$ solves TE, **then** $\rho(t = 1, \cdot) = \rho_1$

The continuous time picture

X_t flow map given by velocity field $b(t,x)$

$$X_{t=0}(x) = x \in \mathbb{R}^d$$

$$\dot{X}_t(x) = b(t, X_t(x))$$



At the level of the of the distribution, how does $\rho(t,x)$ evolve?

Transport equation

$$\partial_t \rho(t,x) + \nabla \cdot \big(b(t,x)\rho(t,x)\big) = 0, \quad \rho(t=0,\cdot) = \rho_0$$

If $\rho(t)$ solves TE, **then** $\rho(t=1,\cdot) = \rho_1$

Benamou-Brenier theory says that $b(t,x)$ exists (assuming Lipschitz)

How to find a sufficient $b(t,x)$ to map ρ_0 to ρ_1?

Direct maximum likelihood

One approach: find $b(t, x)$ via maximum likelihood

FFJORD: Grathwohl *et al.* arXiv:1810.01367 (2018)

$$\rho(1, X_1(x)) = \rho_0(x) \exp\left(-\int_0^1 \nabla \cdot b(t, X_t(x)) dt \right)$$

$$\min_b KL(\rho_1 || \rho(1)) = \min \mathbb{E}_{\rho_1}\left[\log \frac{\rho_1(x)}{\rho(1, x)}\right]$$

$$= \min -\mathbb{E}_{\rho_1}\Big[\log \rho(1, x)\Big] + C$$

- $b(t, x)$ parametrized as neural network
- adjoint method (Neural ODE) allows for gradient wrt parameters of b

Direct maximum likelihood

One approach: find $b(t,x)$ via maximum likelihood

FFJORD: Grathwohl *et al.* arXiv:1810.01367 (2018)

$$\rho(1,X_1(x)) = \rho_0(x)\exp\Big(-\int_0^1 \nabla\cdot b(t,X_t(x))dt\Big)$$

$$\min_b KL(\rho_1\,||\,\rho(1)) = \min \mathbb{E}_{\rho_1}\Big[\log\frac{\rho_1(x)}{\rho(1,x)}\Big]$$

$$= \min -\mathbb{E}_{\rho_1}\Big[\log\rho(1,x)\Big] + C$$

- $b(t,x)$ parametrized as neural network
- adjoint method (Neural ODE) allows for gradient wrt parameters of b

Loss involves integrating the ODE

Many paths from ρ_0 to ρ_1

Direct maximum likelihood

One approach: find $b(t,x)$ via maximum likelihood

FFJORD: Grathwohl *et al.* arXiv:1810.01367 (2018)

$$\rho(1,X_1(x)) = \rho_0(x)\exp\Big(-\int_0^1 \nabla\cdot b(t,X_t(x))dt\Big)$$

$$\min_b KL(\rho_1\,||\,\rho(1)) = \min \mathbb{E}_{\rho_1}\Big[\log\frac{\rho_1(x)}{\rho(1,x)}\Big]$$

$$= \min -\mathbb{E}_{\rho_1}\Big[\log\rho(1,x)\Big] + C$$

- $b(t,x)$ parametrized as neural network
- adjoint method (Neural ODE) allows for gradient wrt parameters of b

Loss involves integrating the ODE

Many paths from ρ_0 to ρ_1

Is there a simpler paradigm for learning $b(t,x)$?

Solving for $b(t,x)$ solves the transport

Is there a simple paradigm for learning $b(t,x)$?

Dream scenario: figure out a way to perform regression on the velocity field

$$\min_{\hat{b}} \int_{t=0}^{t=1} |\, b(t,x) - \hat{b}(t,x)\,|^2 \rho(t,x) dx dt$$

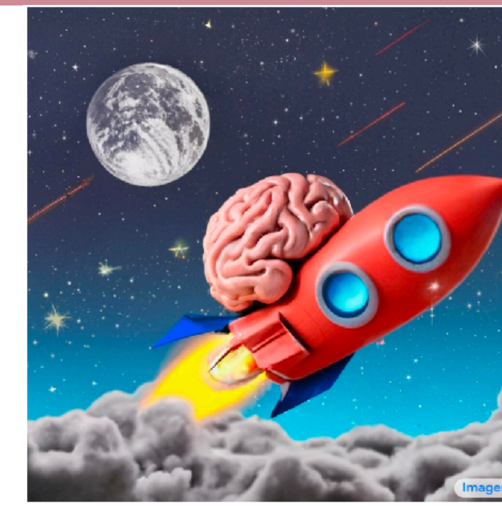
Problems:

- Don't have a fixed $b(t,x)$ to regress on
- Don't have a $\rho(t,x)$ to sample from!

How can we work exactly on $t \in [0,1]$ with arbitrary ρ_0 and ρ_1, build a connection between them, and get the velocity $b(t,x)$ directly?

Inspiration: Score-based diffusion

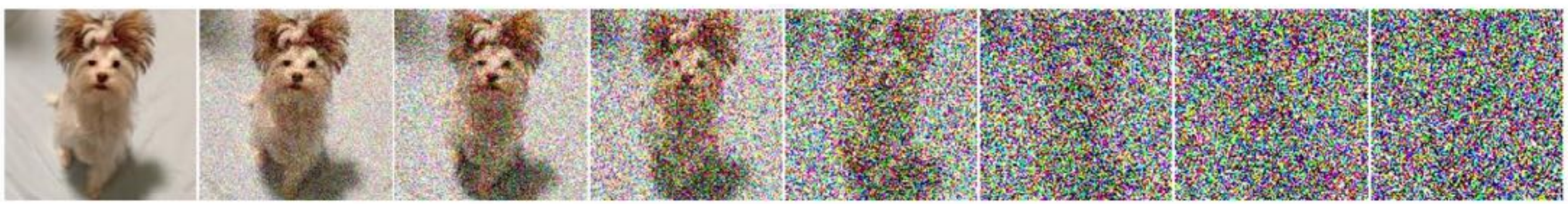
Song et al. arXiv:2011.13456 (2021);
Sohl-Dickstein et al arXiv:1503.03585 (2021);
Hyvärinen JMLR **6** (2005);
Vincent, Neural Comp. **23**, 1661 (2011)



"A brain riding a rocket ship headed toward the moon." Imagen, Saharia et al 2205.11487

Map $x_1 \sim \rho_1$ to Gaussian ρ_0 via Ornstein-Uhlenbeck (OU) process

$$dX_t = -X\,dt + \sqrt{2}\,dW_t, \quad X_0 = x_1$$



SDE $\quad dX_t^B = -X_t\,dt + \boxed{\nabla \log \rho(t, X_t)}\,dt + \sqrt{2}\,dW_t, \quad X_0 = x_0$

ODE $\quad b(t, x) = x - \nabla \log \rho(t, x)$

Access to the score $s(t, x) = \nabla \log \rho(t, x)$ allows one to simulate the reverse process as a generative model

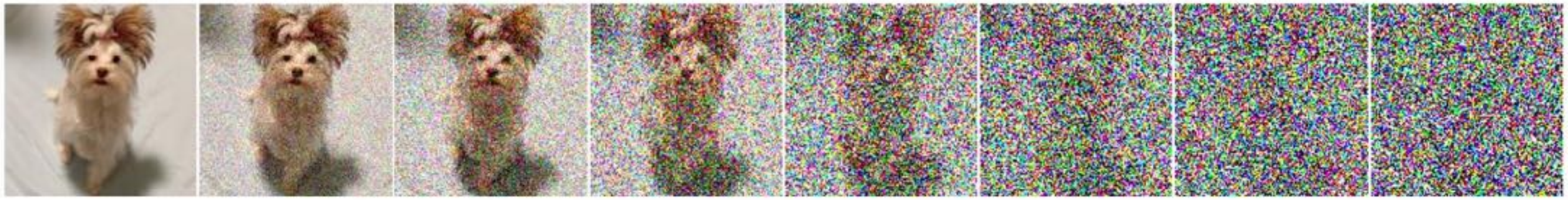
Inspiration: Score-based diffusion

Song et al. arXiv:2011.13456 (2021);
Sohl-Dickstein et al arXiv:1503.03585 (2021);
Hyvärinen JMLR **6** (2005);
Vincent, Neural Comp. **23**, 1661 (2011)

"A brain riding a rocket ship headed toward the moon." Imagen, Saharia et al 2205.11487

Map $x_1 \sim \rho_1$ to Gaussian ρ_0 via Ornstein-Uhlenbeck (OU) process

$$dX_t = -X\,dt + \sqrt{2}\,dW_t, \quad X_0 = x_1$$



SDE $$dX_t^B = -X_t\,dt + \boxed{\nabla \log \rho(t, X_t)}\,dt + \sqrt{2}\,dW_t, \quad X_0 = x_0$$

ODE $$b(t,x) = x - \nabla \log \rho(t,x)$$

We can regress using the Ornstein-Uhlenbeck path. But this path emerges from a carefully chosen SDE. Can we do something simpler?

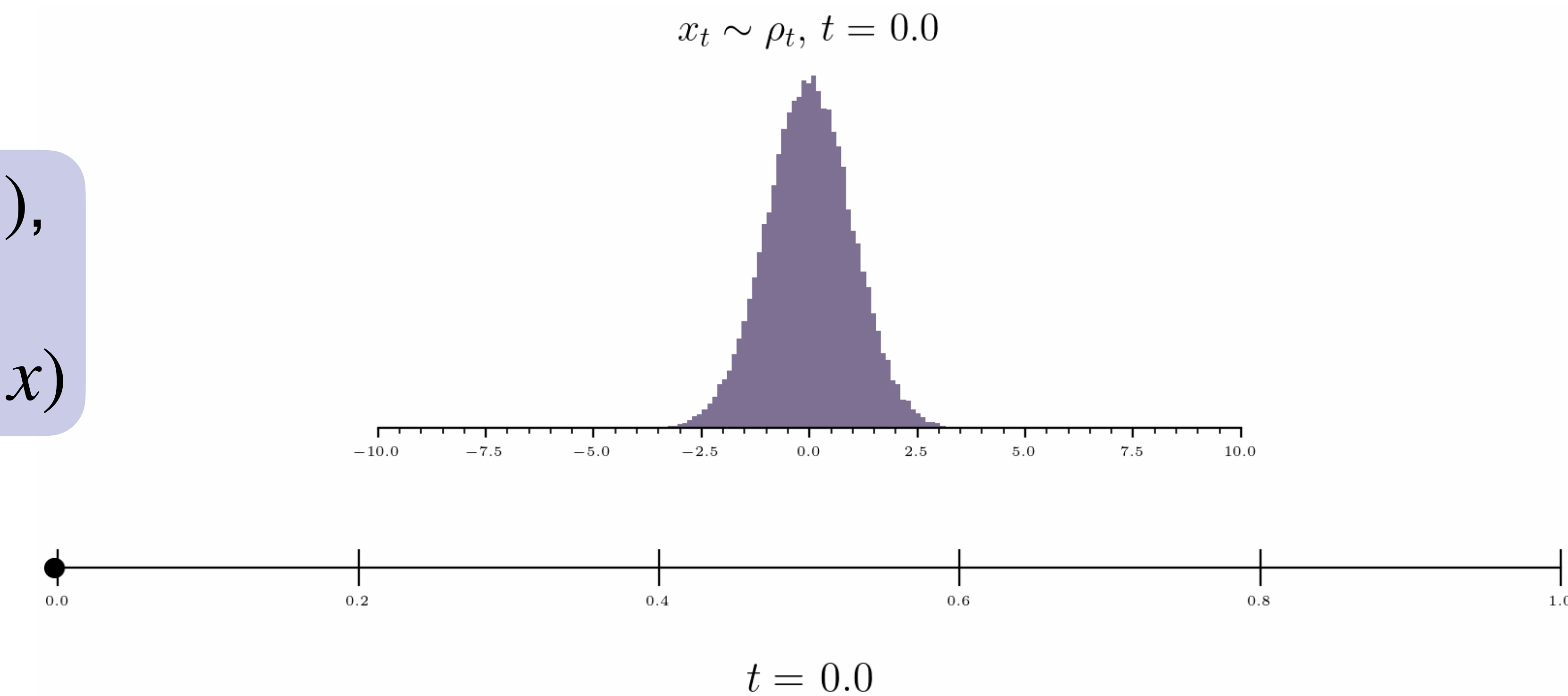
Stochastic Interpolants

MSA *& Vanden-Eijnden arXiv:2209.15571 (2022);*

Interpolant Function $I(t, x_0, x_1)$

- A function of x_0, x_1, and time t with b.c.'s: $I_{t=0} = x_0$ and $I_{t=1} = x_1$
- Example: $I(t, x_0, x_1) = (1 - t)x_0 + tx_1$

If x_0, x_1 drawn from some $\rho(x_0, x_1)$, then $I(t, x_0, x_1)$ is a **stochastic process which samples $I_t \sim \rho(t, x)$**



Interpolant Density

$$\rho(t, x) = \mathbb{E}_{\rho(x_0, x_1)}\left[\delta\big(x - I(t, x_0, x_1)\big)\right]$$

What fixes $\rho(t, x)$?

1. Choice of **coupling**: how to sample x_0, x_1

   simple example: $\rho(x_0, x_1) = \rho_0(x_0)\rho_1(x_1)$

2. Choice of **interpolant** $I(t, x_0, x_1)$:

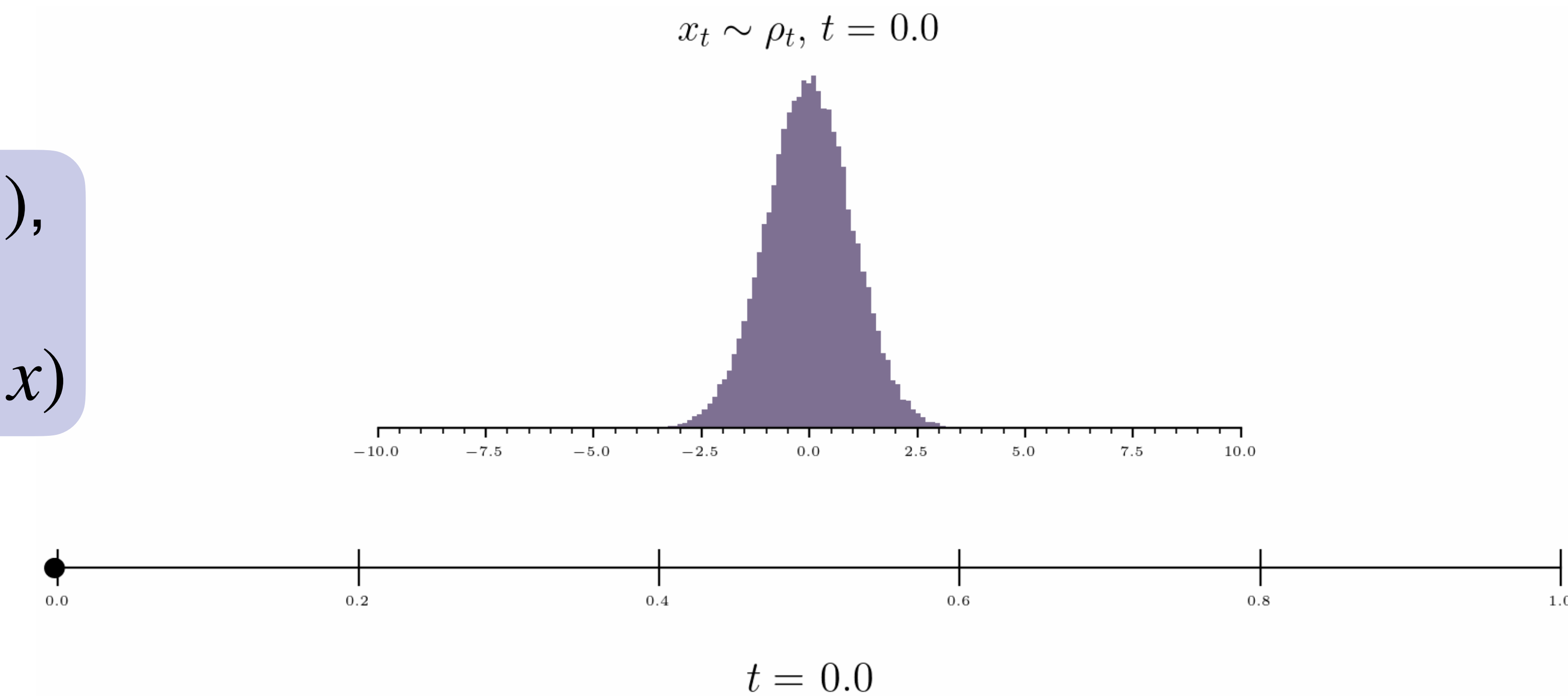
Stochastic Interpolants

MSA & Vanden-Eijnden arXiv:2209.15571 (2022);

Interpolant Function $I(t, x_0, x_1)$

- A function of x_0, x_1, and time t with b.c.'s: $I_{t=0} = x_0$ and $I_{t=1} = x_1$
- Example: $I(t, x_0, x_1) = (1 - t)x_0 + tx_1$

If x_0, x_1 drawn from some $\rho(x_0, x_1)$, then $I(t, x_0, x_1)$ is a **stochastic process which samples $I_t \sim \rho(t, x)$**



Interpolant Density

$$\rho(t, x) = \mathbb{E}_{\rho(x_0, x_1)}\left[\delta\big(x - I(t, x_0, x_1)\big)\right]$$

What fixes $\rho(t, x)$?

1. Choice of **coupling**: how to sample x_0, x_1

   simple example: $\rho(x_0, x_1) = \rho_0(x_0)\rho_1(x_1)$

2. Choice of **interpolant** $I(t, x_0, x_1)$:

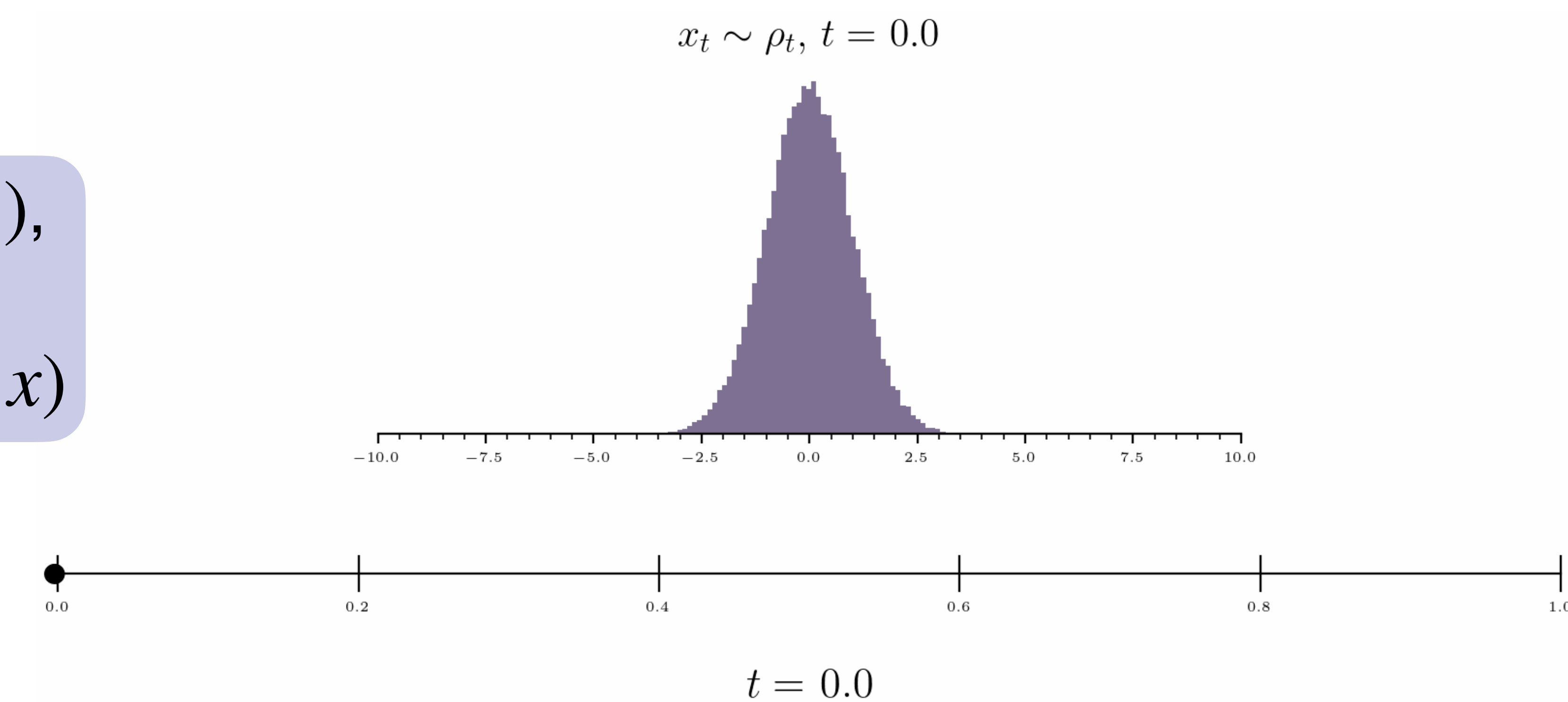
Stochastic Interpolants

MSA *& Vanden-Eijnden arXiv:2209.15571 (2022);*

Interpolant Function $I(t, x_0, x_1)$

- A function of x_0, x_1, and time t with b.c.'s: $I_{t=0} = x_0$ and $I_{t=1} = x_1$
- Example: $I(t, x_0, x_1) = (1-t)x_0 + t x_1$

If x_0, x_1 drawn from some $\rho(x_0, x_1)$, then $I(t, x_0, x_1)$ is a **stochastic process which samples $I_t \sim \rho(t, x)$**



Interpolant Density

$$\rho(t, x) = \mathbb{E}_{\rho(x_0, x_1)}\left[\delta\big(x - I(t, x_0, x_1)\big)\right]$$

Can sample $\rho(t, x)$!

$$\min_{\hat{b}} \int_{t=0}^{t=1} |\, b(t, x) - \hat{b}(t, x)\,|^2 \rho(t, x) dx dt$$

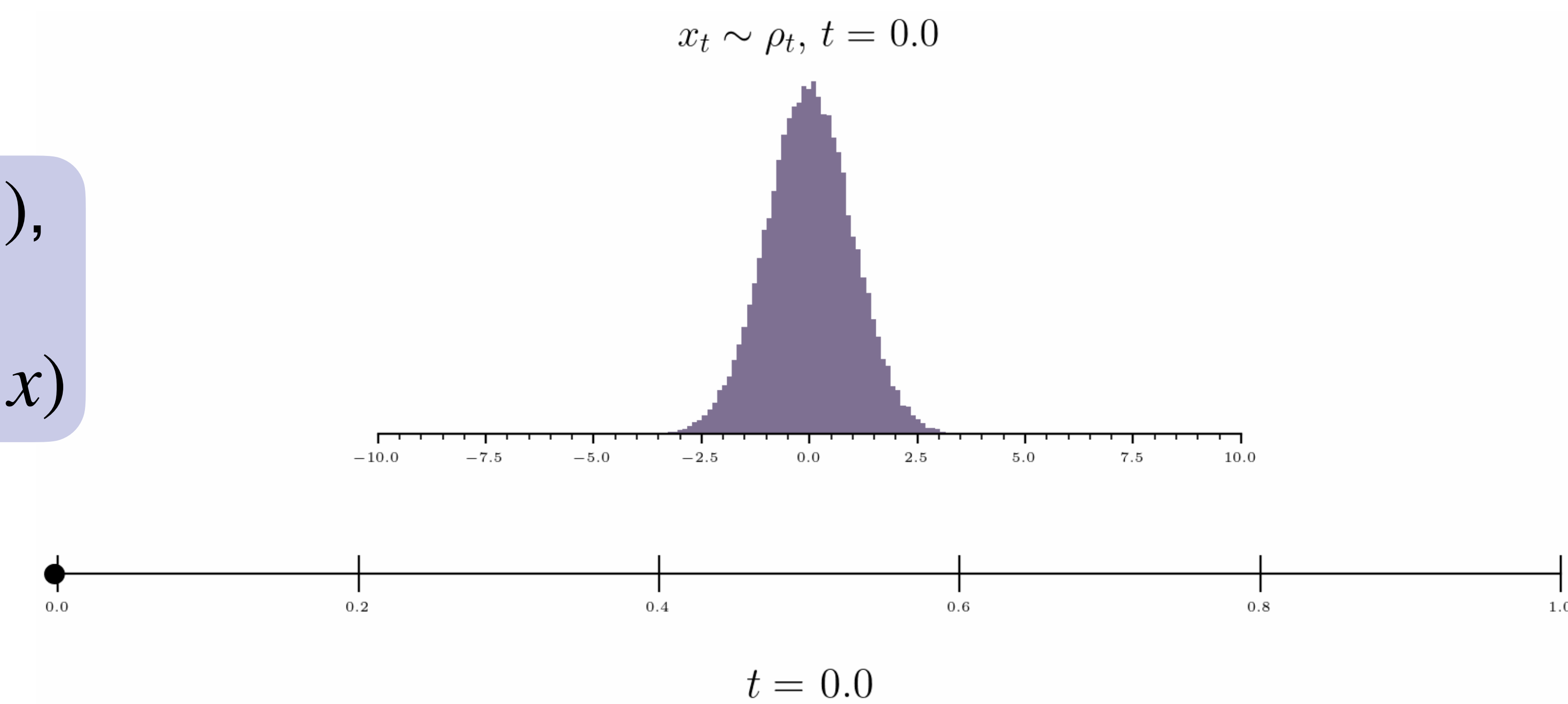
Stochastic Interpolants

MSA *& Vanden-Eijnden arXiv:2209.15571 (2022);*

Interpolant Function $I(t, x_0, x_1)$

- A function of x_0, x_1, and time t with b.c.'s: $I_{t=0} = x_0$ and $I_{t=1} = x_1$
- Example: $I(t, x_0, x_1) = (1-t)x_0 + tx_1$

If x_0, x_1 drawn from some $\rho(x_0, x_1)$, then $I(t, x_0, x_1)$ is a **stochastic process which samples** $I_t \sim \rho(t, x)$



Interpolant Density

$$\rho(t, x) = \mathbb{E}_{\rho(x_0, x_1)}\left[\delta\left(x - I(t, x_0, x_1)\right)\right]$$

Can sample $\rho(t, x)$!

$$\min_{\hat{b}} \int_{t=0}^{t=1} |\, b(t,x) - \hat{b}(t,x)\,|^2 \rho(t,x)\,dx\,dt$$

Stochastic Interpolants: what is $b(t, x)$?

Interpolant Function $I(t, x_0, x_1)$

- Example: $I(t, x_0, x_1) = (1 - t)x_0 + tx_1$

$$\min_{\hat{b}} \int_{t=0}^{t=1} |\, b(t, x) - \hat{b}(t, x) \,|^2 \rho(t, x) dx dt$$

- when $x_0, x_1 \sim \rho(x_0, x_1),\ \ I_t \sim \rho(t)$

We have samples $I_t \sim \rho(t, x)$ via the interpolant, but what is $b(t, x)$?

Definition

The $\rho(t, \cdot\,)$ of x_t satisfies a transport equation

$$\partial_t \rho + \nabla \cdot \big(b(t, x)\rho\big) = 0, \quad \rho(t = 0, \cdot) = \rho_0$$

and $b(t, x)$ is given as the conditional expectation

$$b(t, x) = \mathbb{E}[\partial_t I(t) \,|\, I(t) = x]$$

prove with characteristic function, sketch in backup slides.

Stochastic Interpolants: Simple Objective

$$\min_{\hat{b}} \int_{t=0}^{t=1} |\hat{b}(t,x) - b(t,x)|^2 \rho(t,x) dx dt$$

plug in definition of $b(t,x)$

$$\min_{\hat{b}} \int_{t=0}^{t=1} \int_{\mathbb{R}^d} |\mathbb{E}[\partial_t I(t) \,|\, I(t) = x] - \hat{b}(t,x)|^2 \rho(t,x) dx dt$$

$$\int_{\mathbb{R}^d} \mathbb{E}[\partial_t I(t) \,|\, I(t) = x] \rho(t,x) = \mathbb{E}_{\rho(x_0,x_1)}[\partial_t I(t)]$$

Note: definition of conditional expectation

Prop.

$b(t,x)$ is the minimizer of

$$L[\hat{b}] = \int_0^1 \mathbb{E}_{\rho(x_0,x_1)} \left[|\hat{b}(t, x(t)) - \partial_t I(t)|^2 \right] dt$$

using shorthand $I(t) = I(t, x_0, x_1)$

Stochastic Interpolants: Generative Model

"Flow matching"

MSA *& Vanden-Eijnden arXiv:2209.15571 (2022);*
Liu et al. arXiv:2209.03003 (2022);
Lipman et al. arXiv:2210.02747 (2022)

Prop.

$b(t,x)$ is the minimizer of

$$L[\hat{b}] = \int_0^1 \mathbb{E}_{\rho(x_0,x_1)}\left[\,|\,\hat{b}(t,x(t)) - \partial_t I(t)\,|^2\right] dt$$

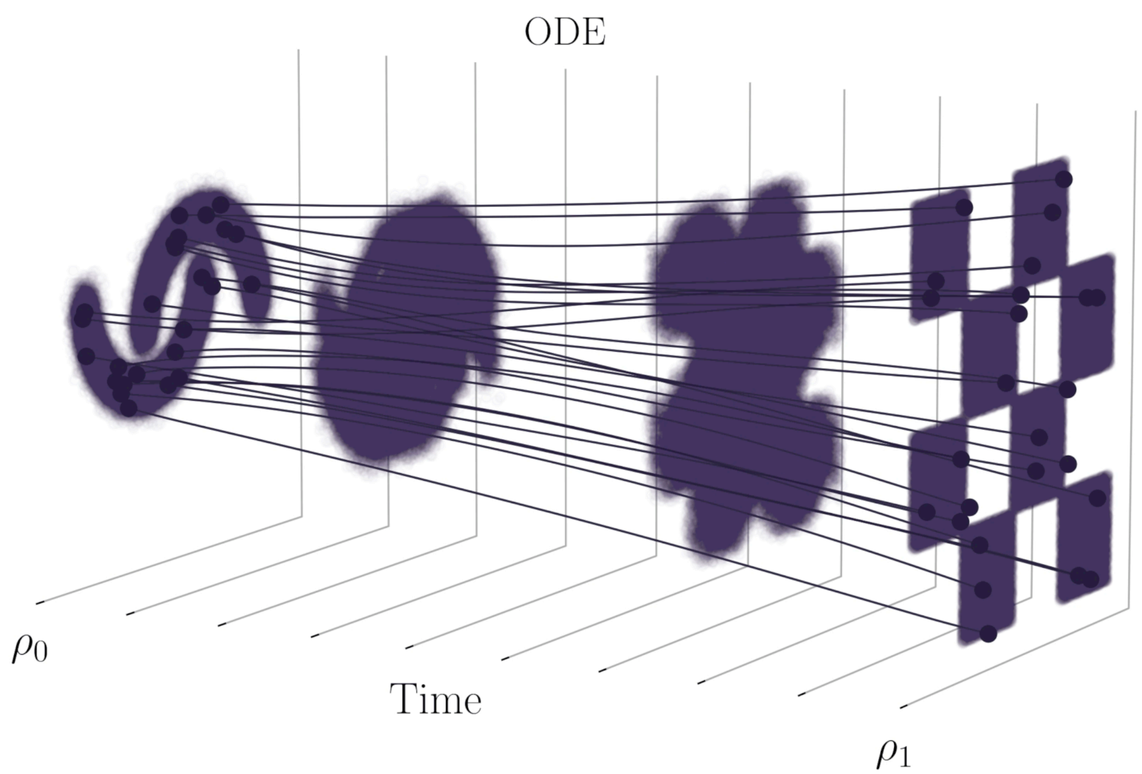
using shorthand $I(t) = I(t,x_0,x_1)$

- Loss is directly estimable over ρ_0, ρ_1
- Generative model connects *any* two densities
- Likelihood and sampling available via fast ODE integrators
- Loss bounds Wasserstein-2 between $\rho(1,x)$ and ρ_1 (Gronwall)

Generative model $\dot{X}_t(x) = b\big(t, X_t(x)\big)$

Correspondence between deterministic and stochastic maps

Why go through this derivation? To stress that the mathematics of learning flows and diffusions by regression is the same, and learning one often defines learning the other

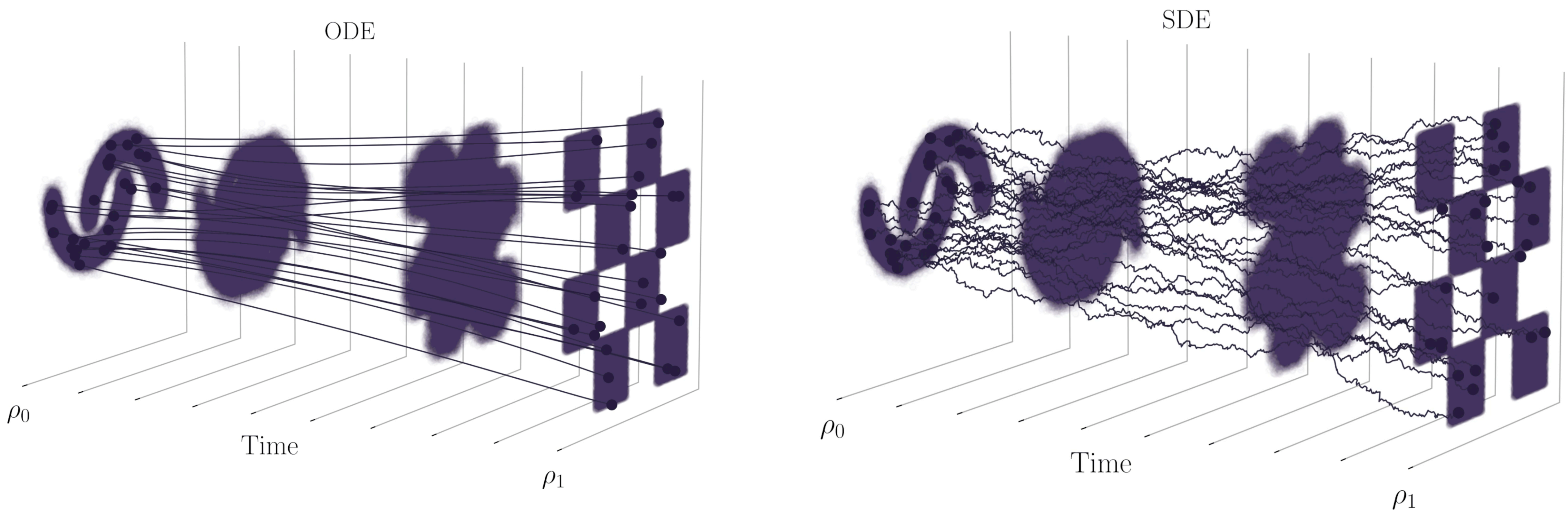


Deterministic

Both processes have the same distribution in law, how are they different?

Correspondence between deterministic and stochastic maps

Why go through this derivation? To stress that the mathematics of learning flows and diffusions by regression is the same, and learning one often defines learning the other



Deterministic **Stochastic**

Both processes have the same distribution in law, how are they different?

Unifying flow-based and diffusion-based generative models

MSA *& Vanden-Eijnden (ICLR 2023) 2209.15571*
MSA *& Boffi, Vanden-Eijnden (JMLR 2024) 2303.08797*

Unifying flow-based and diffusion-based generative models

MSA *& Vanden-Eijnden (ICLR 2023) 2209.15571*
MSA *& Boffi, Vanden-Eijnden (JMLR 2024) 2303.08797*

Transport equation

$$\partial_t \rho + \nabla \cdot (b\rho) = 0$$

ODE

$$\frac{d}{dt} X_t = b\left(t, X_t\right)$$

Learn $\hat{b}$

Unifying flow-based and diffusion-based generative models

MSA *& Vanden-Eijnden (ICLR 2023) 2209.15571*
MSA *& Boffi, Vanden-Eijnden (JMLR 2024) 2303.08797*

Transport equation

$$\partial_t \rho + \nabla \cdot (b\rho) = 0$$

ODE

$$\frac{d}{dt} X_t = b\left(t, X_t\right)$$

Learn $\hat{b}$

Fokker-Planck Equations

$$\partial_t \rho + \nabla \cdot (b^{\mathrm{F/B}} \rho) = \epsilon \Delta \rho$$

where $b^{\mathrm{F/B}} = b \pm \epsilon s$

SDE

$$dX_t^{\mathrm{F/B}} = b_{\mathrm{F/B}}\left(t, X_t^{\mathrm{F}}\right) dt + \sqrt{2\epsilon}\, dW_t^{\mathrm{F/B}}$$

Learn $\hat{b}_{\mathrm{F/B}}$

Bounding the KL between ρ and $\hat{\rho}$

Bounding the KL between ρ and $\hat\rho$

If $\hat\rho$ the density pushed by *estimated* deterministic dynamics $\hat b$, then

$$\partial_t \hat\rho + \nabla \cdot (\hat b \hat\rho) = 0$$

$$\mathrm{KL}(\rho(1) \| \hat\rho(1)) = \int_0^1 \int_{\mathbb{R}^d} (\nabla \log \hat\rho - \nabla \log \rho) \cdot (\hat b - b) \rho \, dx \, dt$$

matching b's does not bound KL, Fisher is uncontrolled by small error in $\hat b - b$

Bounding the KL between ρ and $\hat{\rho}$

If $\hat{\rho}$ the density pushed by *estimated* deterministic dynamics $\hat{b}$, then

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b}\hat{\rho}) = 0$$

$$\mathrm{KL}(\rho(1) \| \hat{\rho}(1)) = \int_0^1 \int_{\mathbb{R}^d} (\nabla \log \hat{\rho} - \nabla \log \rho) \cdot (\hat{b} - b)\rho \, dx \, dt$$

matching b's does not bound KL, Fisher is uncontrolled by small error in $\hat{b} - b$

If $\hat{\rho}$ the density pushed by *estimated* stochastic dynamics $\hat{b}_{\mathrm{F}} = \hat{b} + \epsilon s$, then

$$\partial_t \hat{\rho} + \nabla \cdot (b^{\mathrm{F}}\hat{\rho}) = \epsilon \Delta \hat{\rho}$$

$$\mathrm{KL}(\rho(1) \| \hat{\rho}(1)) \leq \frac{1}{4\epsilon} \int_0^1 \int_{\mathbb{R}^d} \left| \hat{b}_{\mathrm{F}} - b_{\mathrm{F}} \right|^2 \rho \, dx dt$$

$\hat{b}_{\mathrm{F}} - b_{\mathrm{F}}$ does control KL divergence

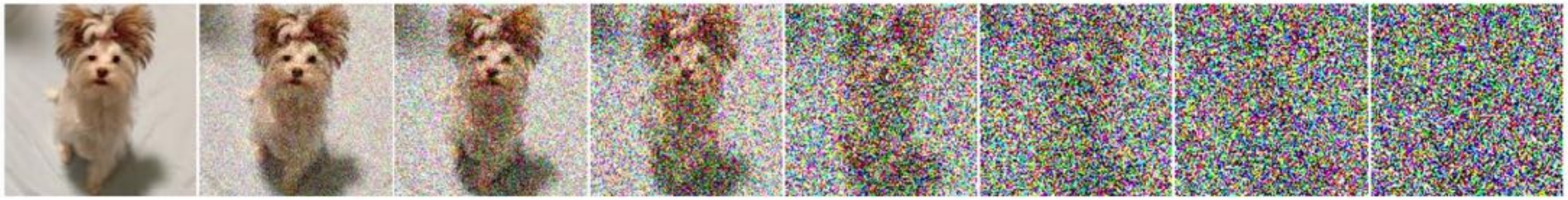
Benefits and Challenges of dynamical measure transport

Access to likelihoods

Regression objectives

Essential for many scientific applications

Contemporary losses are functionally convex!



Iterative sampling can be slow

Formulation for discrete data?

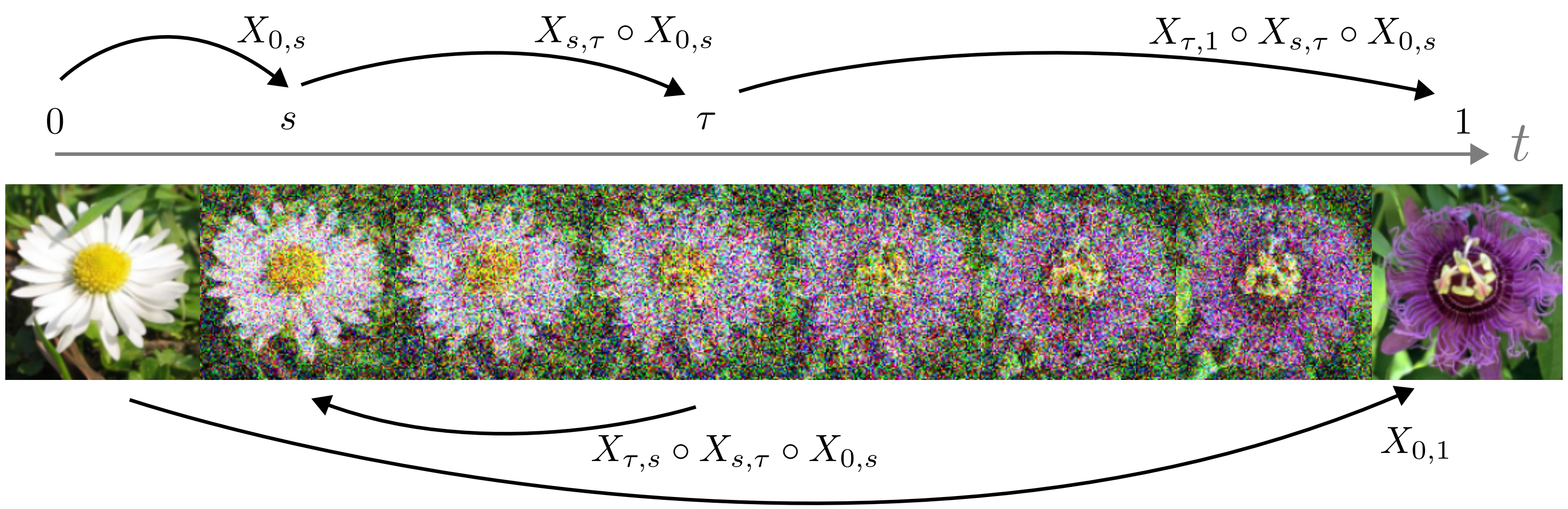
One to few sampling would be ideal

Many proposals, no final picture

Map matching a discrete diffusion

Directly learning the 1 to few step flow map

Can be done with a two-time flow map



$$X_{s,t}(x_s) = x_t$$

"consistency models"

"map matching"

Discrete diffusion:

What's the best way to parameterize a discrete time markov process?

Graph? **Masking?**

Iterative denoising?

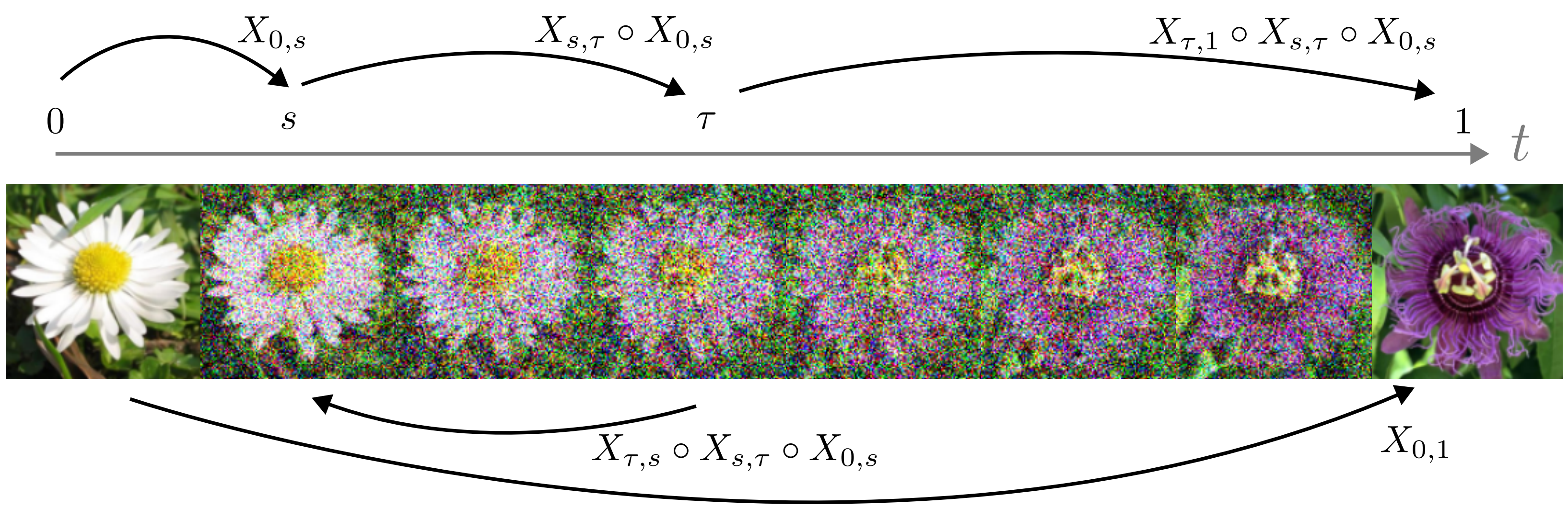
```
def binary_search(arr, x):





    # If x is greater

    # If x is smaller

    else:

```

Gat et al arXiv:2407.15595

Map matching a discrete diffusion

Directly learning the 1 to few step flow map

Can be done with a two-time flow map



$$X_{s,t}(x_s) = x_t$$

"consistency models"

"map matching"

Discrete diffusion:

What's the best way to parameterize a discrete time markov process?

Graph? **Masking?**

Iterative denoising?

```
def binary_search(arr, x):




        # If x is greater

        # If x is smaller

        else:

```

Gat et al arXiv:2407.15595

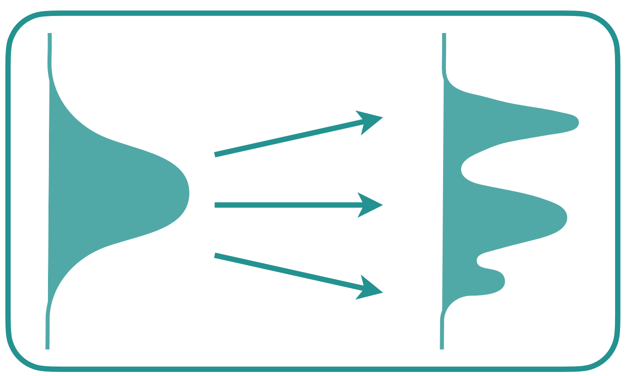
Thank you!

Backup slides

Interpolant applications backup slides

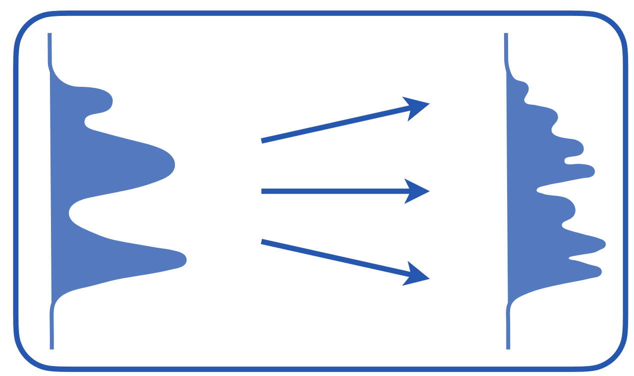
Summary of Context and Applications

Generative modeling



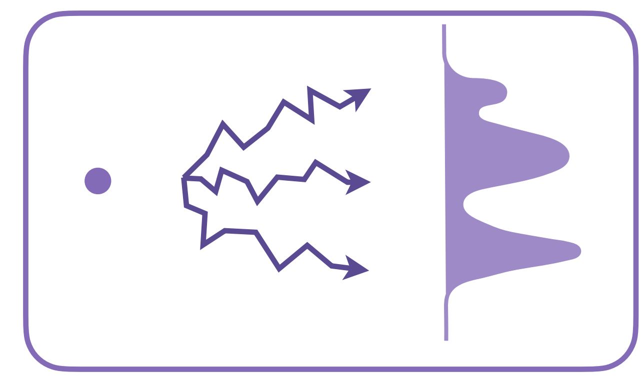
Ex. Image generation
Ex. Statistical physics

Domain Adaptation



Ex. Translation
Ex. Superresolution

Forecasting



Ex. Climate/weather
Ex. Dynamical systems

*We will use the **design flexibility of the interpolant** and the **coupling between** x_0, x_1 to approach various problems*

Example: Interpolants for image generation

MSA *& EVE (ICLR 2023) 2209.15571;*
NM, MG, ***MSA,*** *NB, EVE, SX (ECCV 2024) 2401.08740*

Freedom to choose α, β in:

$$x(t) = \alpha(t)x_0 + \beta(t)x_1$$

to reduce transport cost:

$$C[b] = \int_0^1 \mathbb{E}[\,|\,b(t,x)\,|^2\,]dt$$

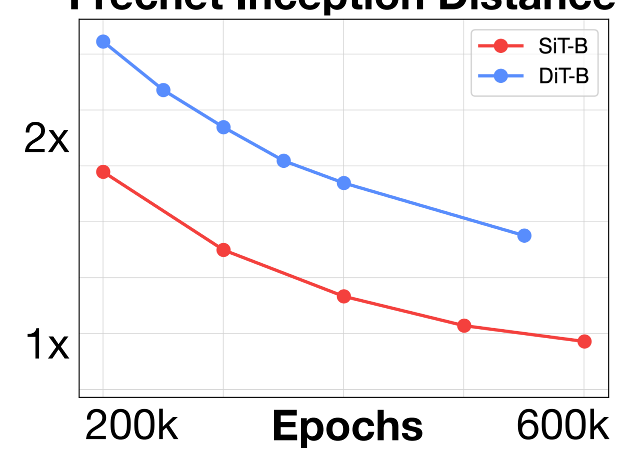
Freedom to choose $\epsilon(t)$ in:

$$dX_t^{\mathrm{F}} = b_{\mathrm{F}}\,dt + \sqrt{2\epsilon(t)}\,dW_t^{\mathrm{F}}$$

to tighten bounds on:

$$\mathrm{D}_{KL}(\hat{\rho}_1\,||\,\rho_1)$$

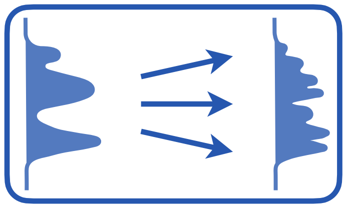
Frechet Inception Distance



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**

Systematic improvements to methods underlying, e.g. Sora (OpenAI, 2024)

Example: Data-dependent coupling



***MSA**, MG, NB, RR, EVE (ICML 2024 Spotlight) 2310.03725*
***MSA**, NB, ML, EVE (ICLR 2024) 2310.03695*

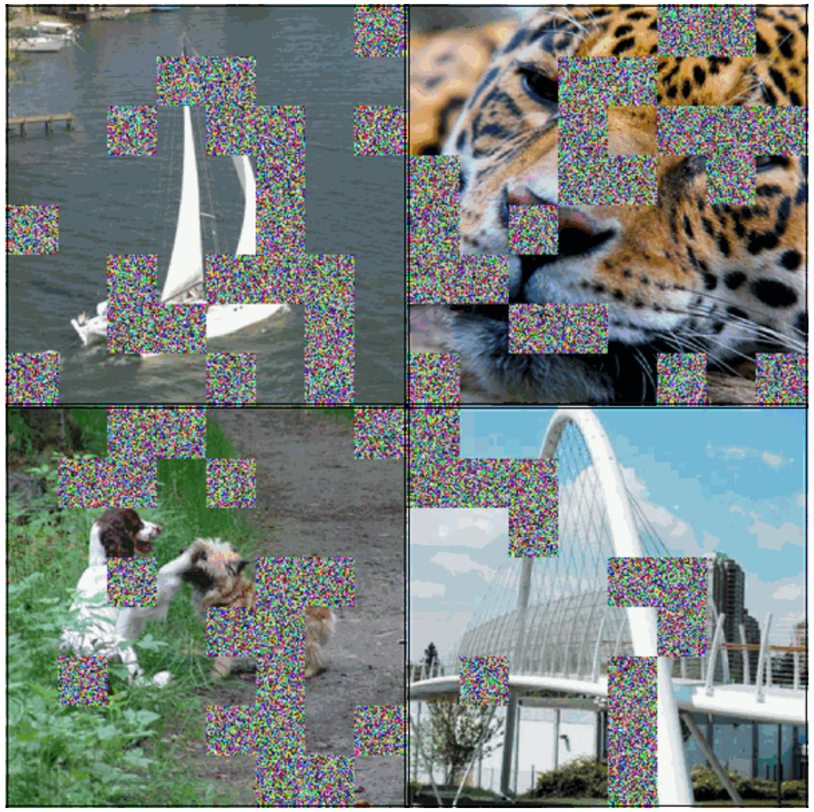
What if one x_0 is coupled to another x_1?

$$\rho(x_0, x_1) = \rho_1(x_1)\rho_0(x_0 \mid x_1)$$

In-painting

x_0 a masked image

$b(t, x)$ invariant in unmasked areas



Super-resolution

x_0 a low-res image

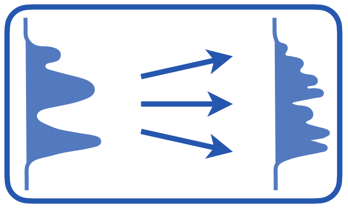
x_0 now *proximal* to its target

Frechet Inception Distance

Model	Train	Valid
Improved DDPM (Nichol & Dhariwal, 2021)	12.26	–
SR3 (Saharia et al., 2022)	11.30	5.20
ADM (Dhariwal & Nichol, 2021)	7.49	3.10
Cascaded Diffusion (Ho et al., 2022a)	4.88	4.63
I^2SB (Liu et al., 2023a)	–	2.70
Dependent Coupling (**Ours**)	**2.13**	**2.05**

More efficient and better performance across tasks

Example: Data-dependent coupling



***MSA**, MG, NB, RR, EVE (ICML 2024 Spotlight) 2310.03725*
***MSA**, NB, ML, EVE (ICLR 2024) 2310.03695*

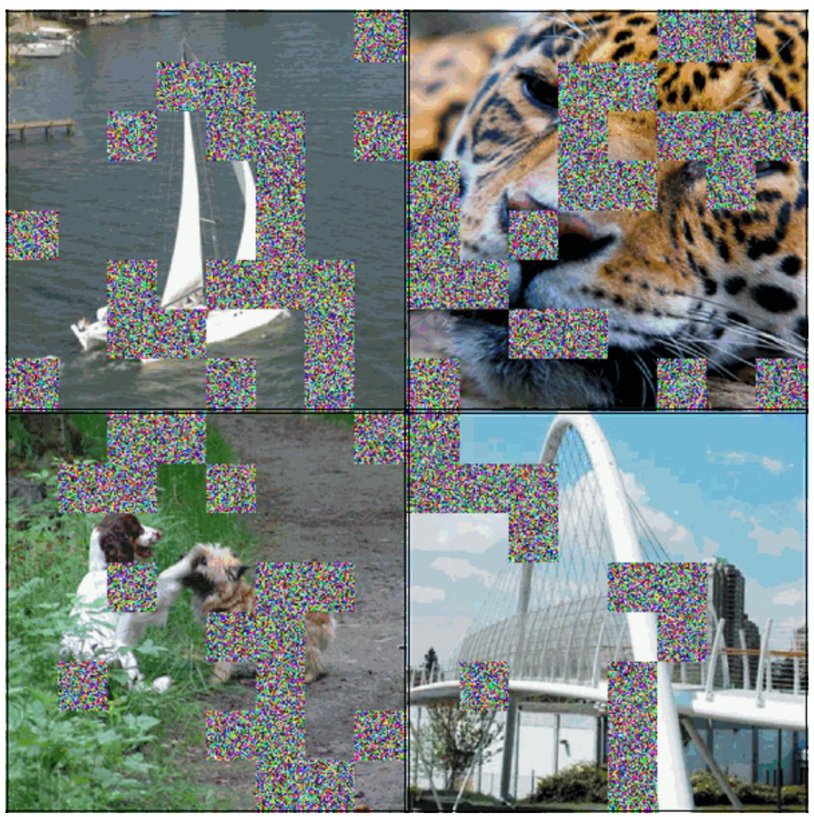
What if one x_0 is coupled to another x_1?

$$\rho(x_0, x_1) = \rho_1(x_1)\rho_0(x_0 \,|\, x_1)$$

In-painting

x_0 a masked image

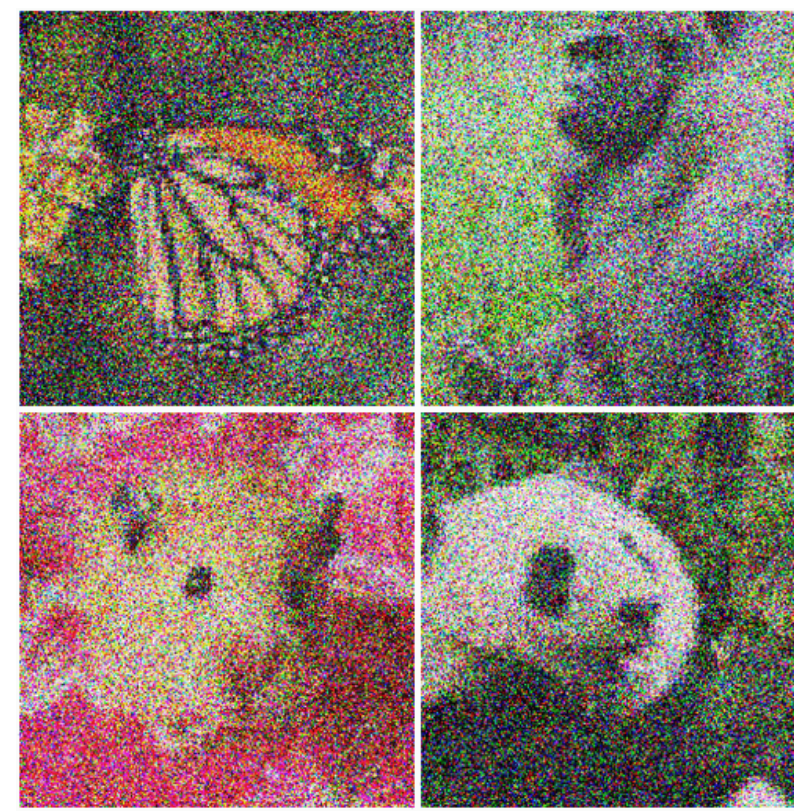
$b(t, x)$ invariant in unmasked areas



Super-resolution

x_0 a low-res image

x_0 now *proximal* to its target

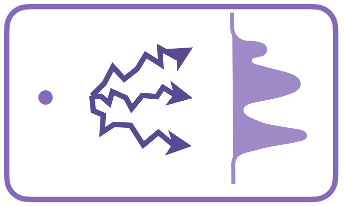


Frechet Inception Distance

Model	Train	Valid
Improved DDPM (Nichol & Dhariwal, 2021)	12.26	–
SR3 (Saharia et al., 2022)	11.30	5.20
ADM (Dhariwal & Nichol, 2021)	7.49	3.10
Cascaded Diffusion (Ho et al., 2022a)	4.88	4.63
I^2SB (Liu et al., 2023a)	–	2.70
Dependent Coupling (**Ours**)	**2.13**	**2.05**

More efficient and better performance across tasks

Example: Probabilistic forecasting



*YC, MG, MH, **MSA**, NB, EVE arXiv:2402. (2024)*

Interpolants for ensembles of future events

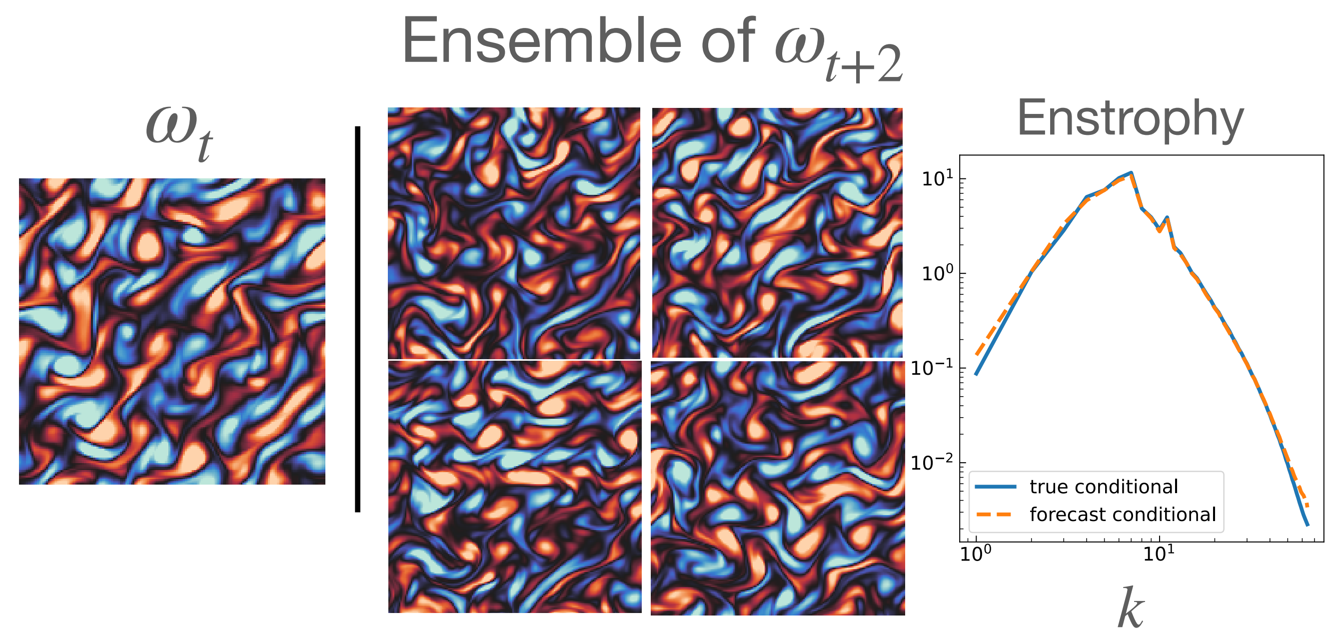
$$\rho(x_0, x_1) = \rho_0(x_0)\rho_1(x_1 \,|\, x_0)$$

Navier Stokes

Evolution of the vorticity ω

Map ω_t to distribution $\rho(\omega_{t+\tau} \,|\, \omega_t)$

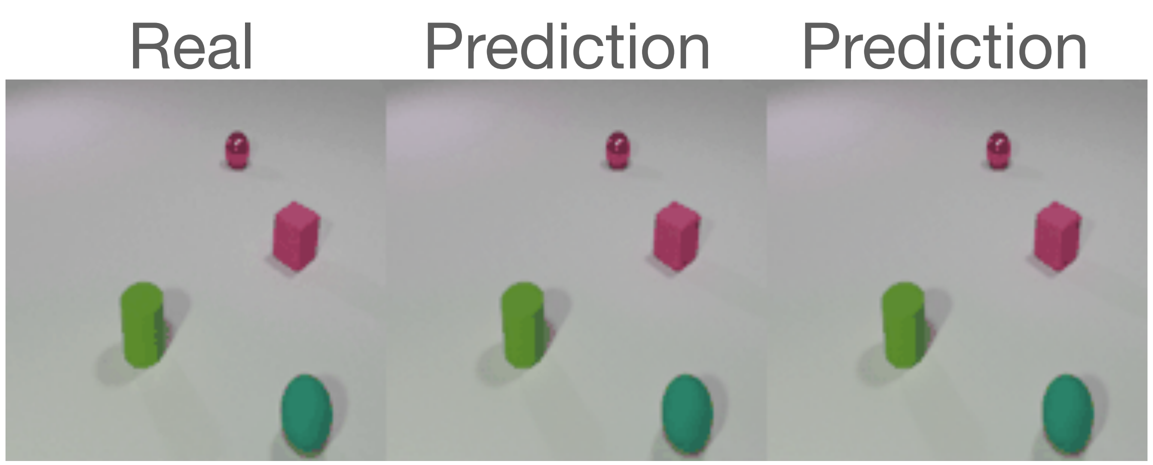
Choose NS w/ random forcing that has invariant measure



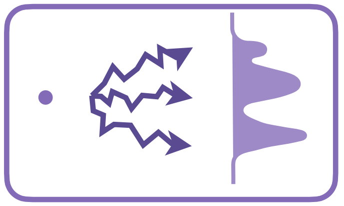
Video completion

Map x_t to distribution $\rho(x_{t+1} \,|\, x_{t-\tau:t})$

Roll out subsequent frames



Example: Probabilistic forecasting



*YC, MG, MH, **MSA**, NB, EVE arXiv:2402. (2024)*

Interpolants for ensembles of future events

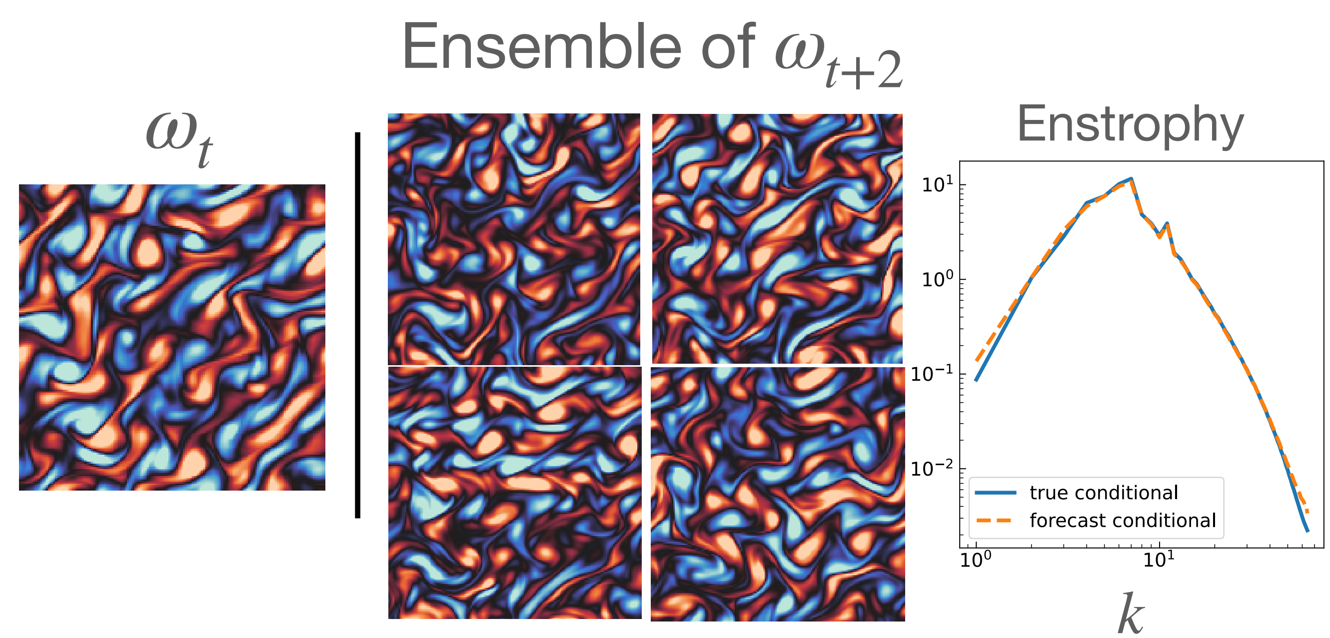
$$\rho(x_0, x_1) = \rho_0(x_0)\rho_1(x_1 \,|\, x_0)$$

Navier Stokes

Evolution of the vorticity ω

Map ω_t to distribution $\rho(\omega_{t+\tau} \,|\, \omega_t)$

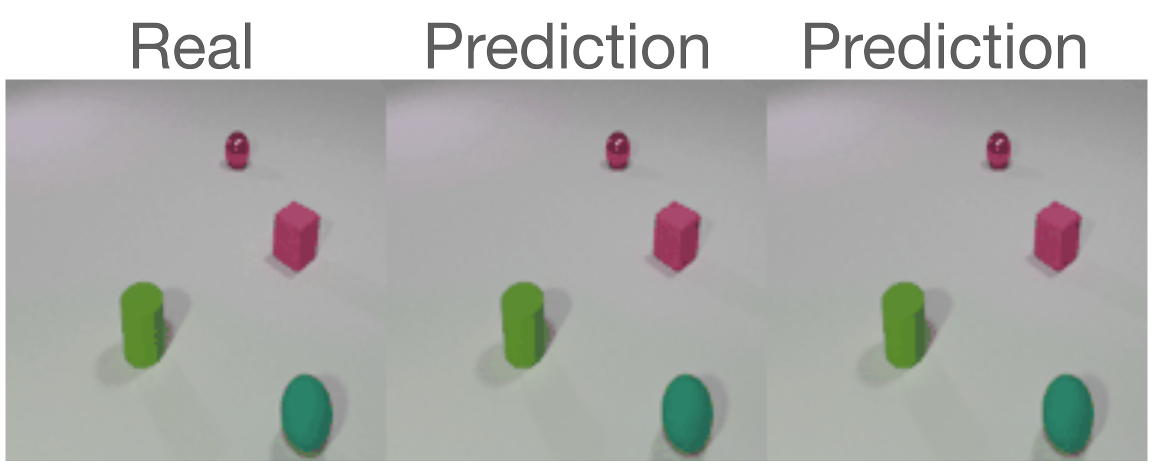
Choose NS w/ random forcing that has invariant measure



Video completion

Map x_t to distribution $\rho(x_{t+1} \,|\, x_{t-\tau:t})$

Roll out subsequent frames



Example: Probabilistic forecasting

*YC, MG, MH, **MSA**, NB, EVE arXiv:2402. (2024)*

Interpolants for ensembles of future events

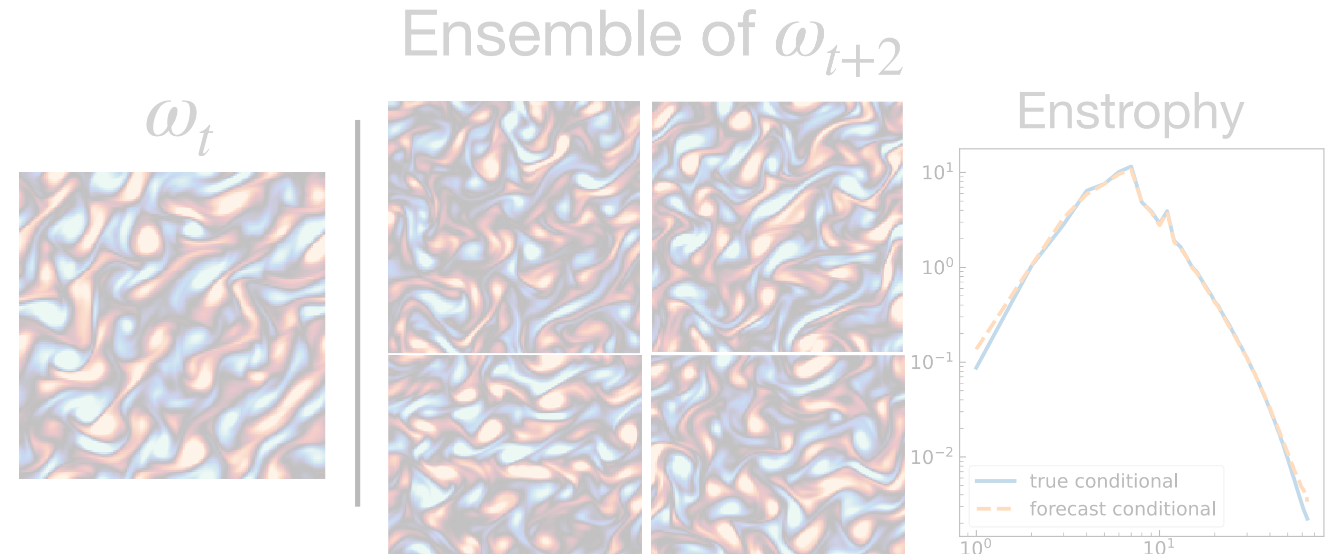
$$\rho(x_0, x_1) = \rho_0(x_0)\rho_1(x_1 \,|\, x_0)$$

Navier Stokes

Evolution of the vorticity ω

Map ω_t to distribution $\rho(\omega_{t+\tau} \,|\, \omega_t)$

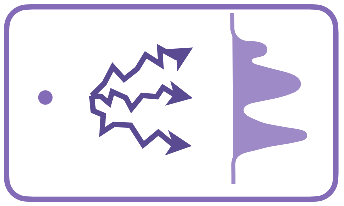
Choose NS w/ random forcing that has invariant measure



Introduces a new family of interpolant Follmer processes — least cost stochastic transport with respect to a reference measure.

Gives tighter control on KL-divergence

Example: Probabilistic forecasting



*YC, MG, MH, **MSA**, NB, EVE arXiv:2402. (2024)*

Interpolants for ensembles of future events

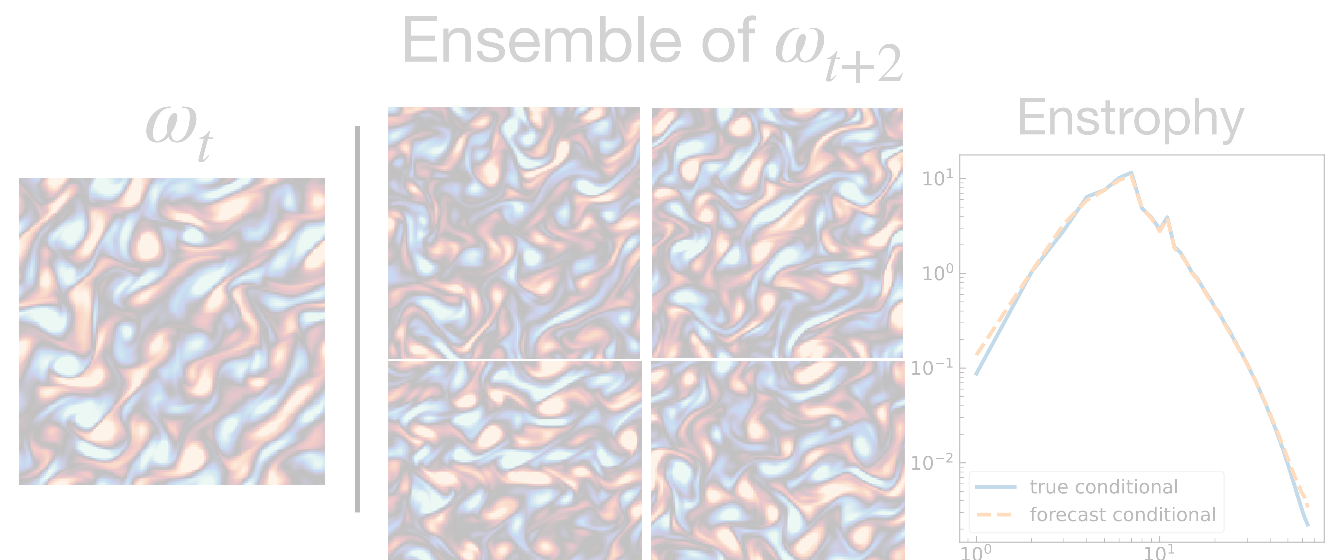
$$\rho(x_0, x_1) = \rho_0(x_0)\rho_1(x_1 \,|\, x_0)$$

Navier Stokes

Evolution of the vorticity ω

Map ω_t to distribution $\rho(\omega_{t+\tau} \,|\, \omega_t)$

Choose NS w/ random forcing that has invariant measure



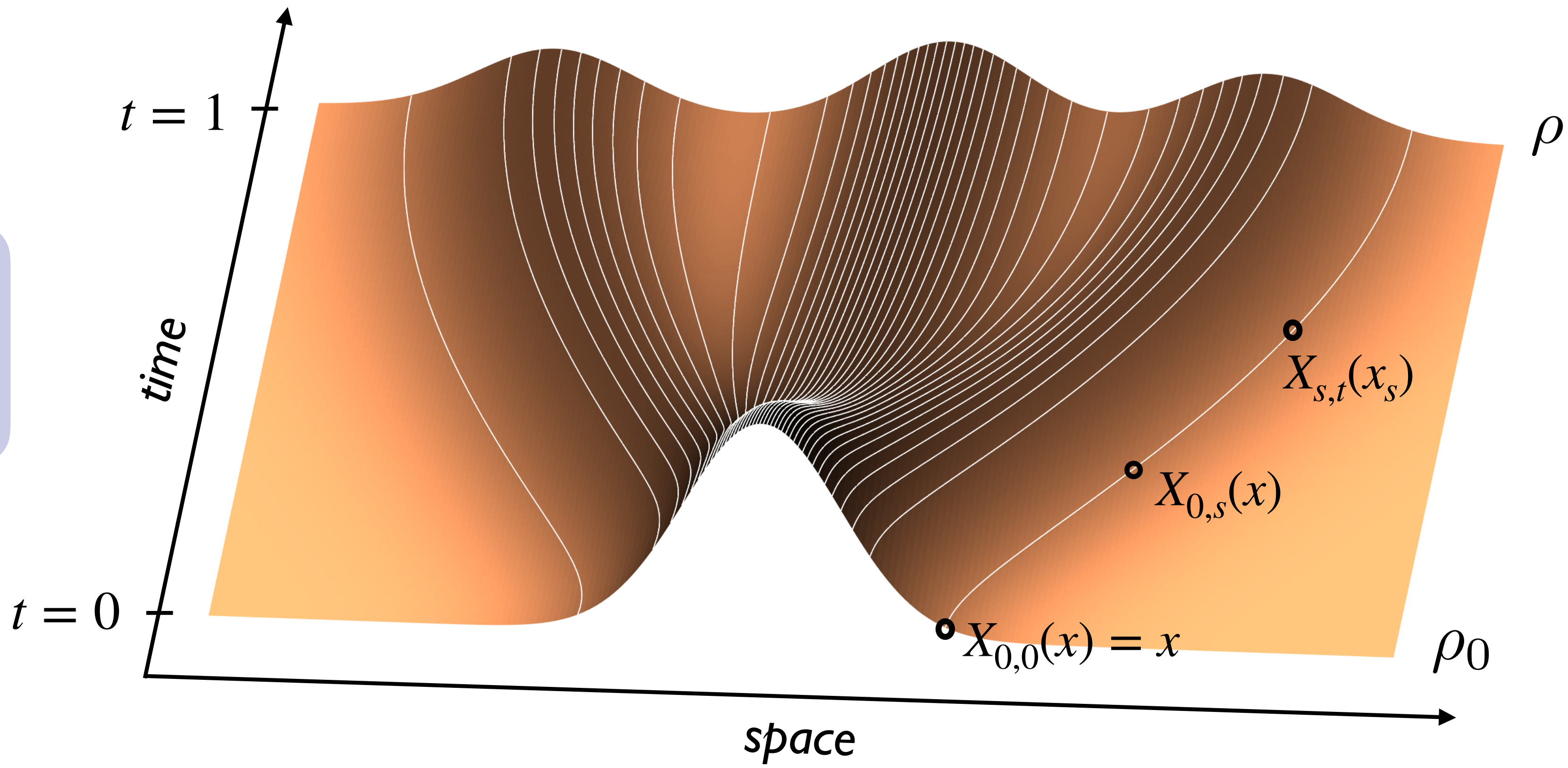
Introduces a new family of interpolant Follmer processes — least cost stochastic transport with respect to a reference measure.

Gives tighter control on KL-divergence

Map Matching Backup slides

Making sense of the flow map

The two-time flow map $X_{s,t}$



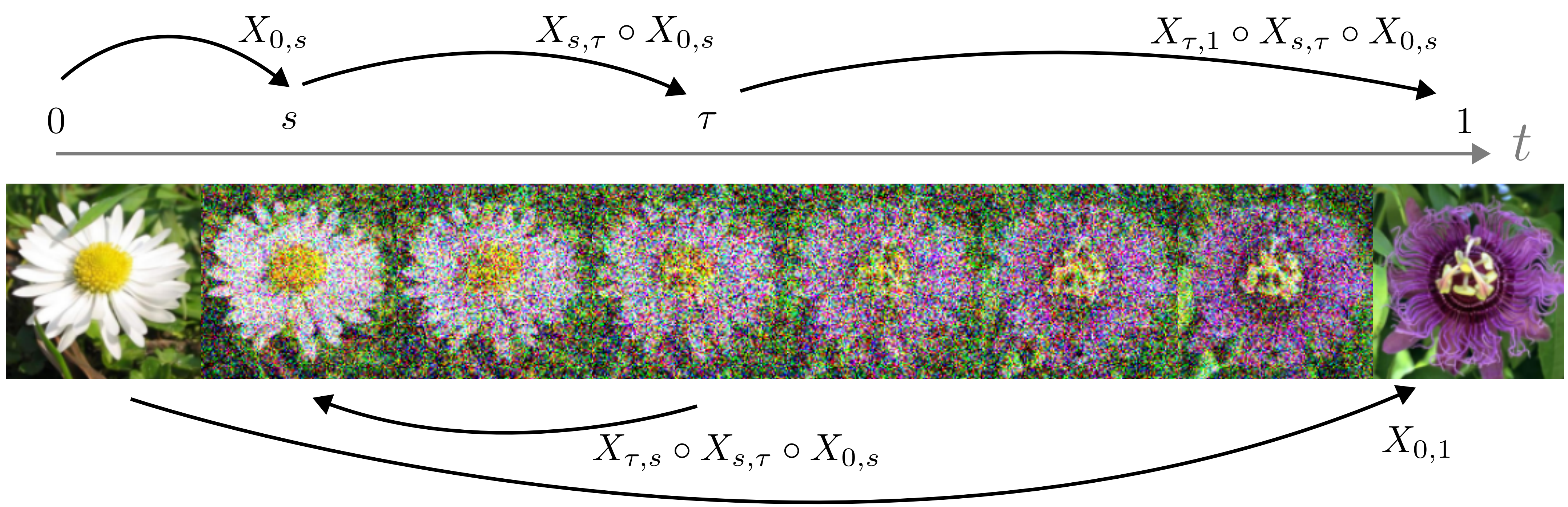
Given an ordinary differential equation of the form

$$\dot{x}_t = b_t\left(x_t\right), \quad x_{t=0} = x_0 \sim \rho_0$$

The two-time flow map is an *arbitrary integrator* from s to t

$$X_{s,t}(x_s) = x_t$$

Properties of the flow map



Semi-group property $\quad X_{t,\tau}(X_{s,t}(x)) = X_{s,\tau}(x)$ **composable**

$\longrightarrow \quad X_{t,s}(X_{s,t}(x)) = x$ **invertible**

$\longrightarrow \quad X_{s,s}(x) = x$ **stationarity**

What dynamical equations does the flow map satisfy?

Lagrangian Equation ($\frac{\partial}{\partial t}$)

$$\partial_t X_{s,t}(x_s) = \dot{x}_t = b_t(X_{s,t}(x))$$

$X_{s,t}(x)$ is the unique solution of

$$\partial_t X_{s,t}(x) = b_t(X_{s,t}(x))$$
$$X_{s,s}(x) = x$$

Eulerian Equation ($\frac{\partial}{\partial s}$)

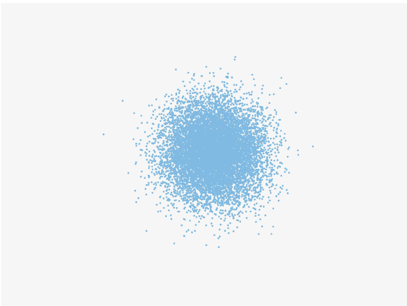
$$\begin{aligned}\frac{d}{ds}X_{s,t}((X_{t,s}(x)) &= 0 \\ &= \partial_s X_{s,t}(X_{t,s}(x)) \\ &+ b_t(X_{s,t}(X_{t,s}(x)) \cdot \nabla X_{s,t}(X_{t,s}(x))\end{aligned}$$

$$\partial_s X_{s,t}(x) + \nabla X_{s,t}(x) \cdot b_t(x) = 0$$
$$X_{t,t}(x) = x$$

Can we use these equation to design objectives for learning $X_{s,t}$?

Map Matching

*Boffi, **MSA,** Vanden-Eijnden arXiv:2406.07507*



$b_t(x)$

Learning the flow map

x_1

Distillation

Direct learning

Learn from existing $b_t(x)$

Lagrangian Map Distillation (LMD)

Eulerian Map Distillation (EMD)

Learn from data $x_1 \sim \rho_1$

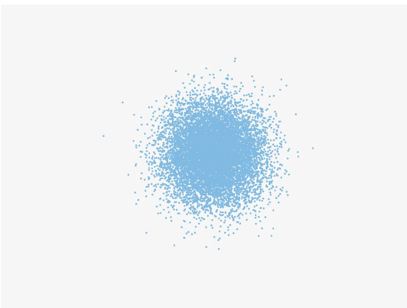
Flow Map Matching (FMM)

Can we use these equation to design objectives for learning $X_{s,t}$?

Map Matching

Boffi, **MSA,** Vanden-Eijnden arXiv:2406.07507

Learning the flow map



$b_t(x)$

x_1

Distillation

Direct learning

Learn from existing $b_t(x)$

Lagrangian Map Distillation (LMD)

Eulerian Map Distillation (EMD)

Learn from data $x_1 \sim \rho_1$

Flow Map Matching (FMM)

Can we use these equation to design objectives for learning $X_{s,t}$?

Lagrangian Map Distillation (LMD)

Prop.

The flow map $X_{s,t}$ is the global minimizer of

$$L_{LMD}(\hat{X}) = \int_{[0,T]^2} \int_{\mathbb{R}^d} \left| \partial_t \hat{X}_{s,t}(x) - b_t\left(\hat{X}_{s,t}(x)\right) \right|^2 \rho_s(x) dx\, ds\, dt$$

subject to $X_{s,s}(x) = x$.

- PINN loss - minimized only when integrand is zero
- $b_t(x)$ any known drift, for example previous trained flow model

Ansatz

$$\hat{X}_{s,t}(x) = (1 - (t - s))x + (t - s) f^{\theta}_{s,t}(x)$$

Tutorial!

https://tinyurl.com/lagrangian-map

Eulerian Map Distillation (EMD)

Prop.

The flow map $X_{s,t}$ is the global minimizer of

$$L_{EMD}(\hat{X}) = \int_{[0,T]^2} \int_{\mathbb{R}^d} \left| \partial_s \hat{X}_{s,t}(x) + b_s(x) \cdot \nabla \hat{X}_{s,t}(x) \right|^2 \rho_s(x)\, dx\, ds\, dt$$

subject to $X_{s,s}(x) = x$.

- PINN loss - minimized only when integrand is zero
- $b_t(x)$ any known drift, for example previous trained flow model

Ansatz

$$\hat{X}_{s,t}(x) = (1 - (t - s))x + (t - s)f^{\theta}_{s,t}(x)$$

Tutorial!

Coming soon….

Flow map matching (FMM)

Prop.

The flow map $X_{s,t}$ is the global minimizer of

$$L_{FMM}[\hat{X}] = \int_{[0,1]^2} \left(\mathbb{E}\left[\left| \partial_t \hat{X}_{s,t}\left(\hat{X}_{t,s}\left(I_t\right)\right) - \dot{I}_t \right|^2 \right] + \mathbb{E}\left[\left| \hat{X}_{s,t}\left(\hat{X}_{t,s}\left(I_t\right)\right) - I_t \right|^2 \right] \right) dsdt$$

where I_t is an interpolant with $\text{Law}(I_t) = \rho_t$.

- Depends solely on $\hat{X}_{s,t}$ and interpolant I_t
- First term ensures Lagrangian equation, second term semigroup.

Ansatz

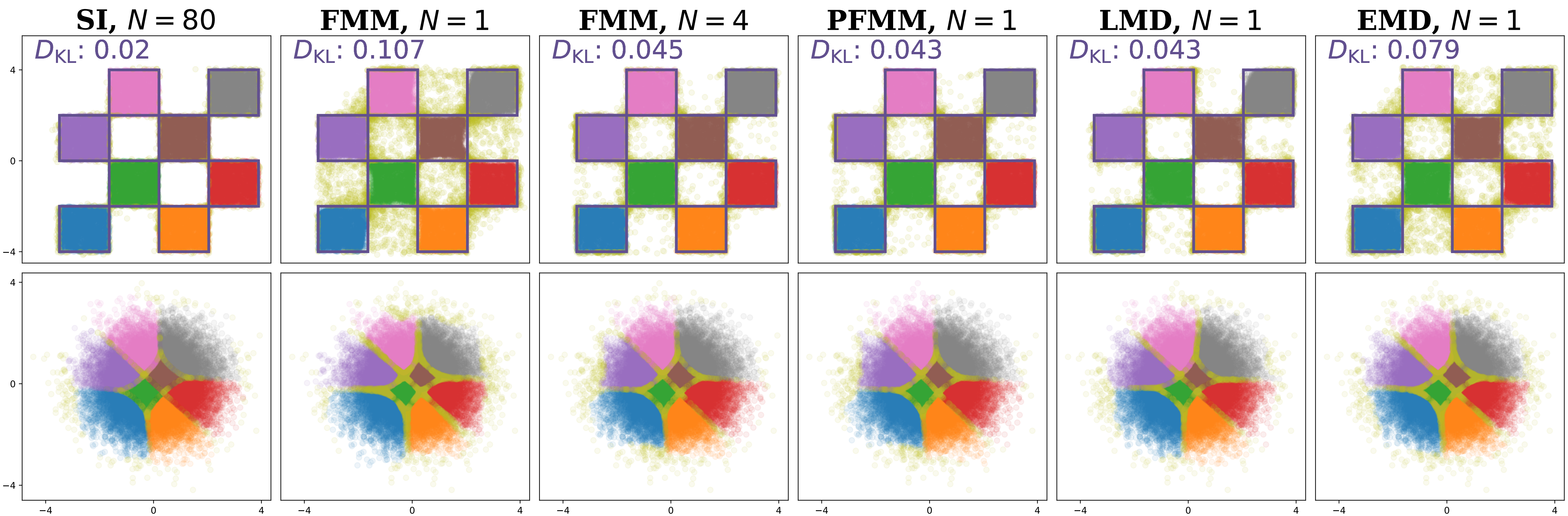
$$\hat{X}_{s,t}(x) = (1 - (t-s))x + (t-s)f^{\theta}_{s,t}(x)$$

Tutorial!

https://tinyurl.com/map-match

How do they compare?

2D checkerboard distribution

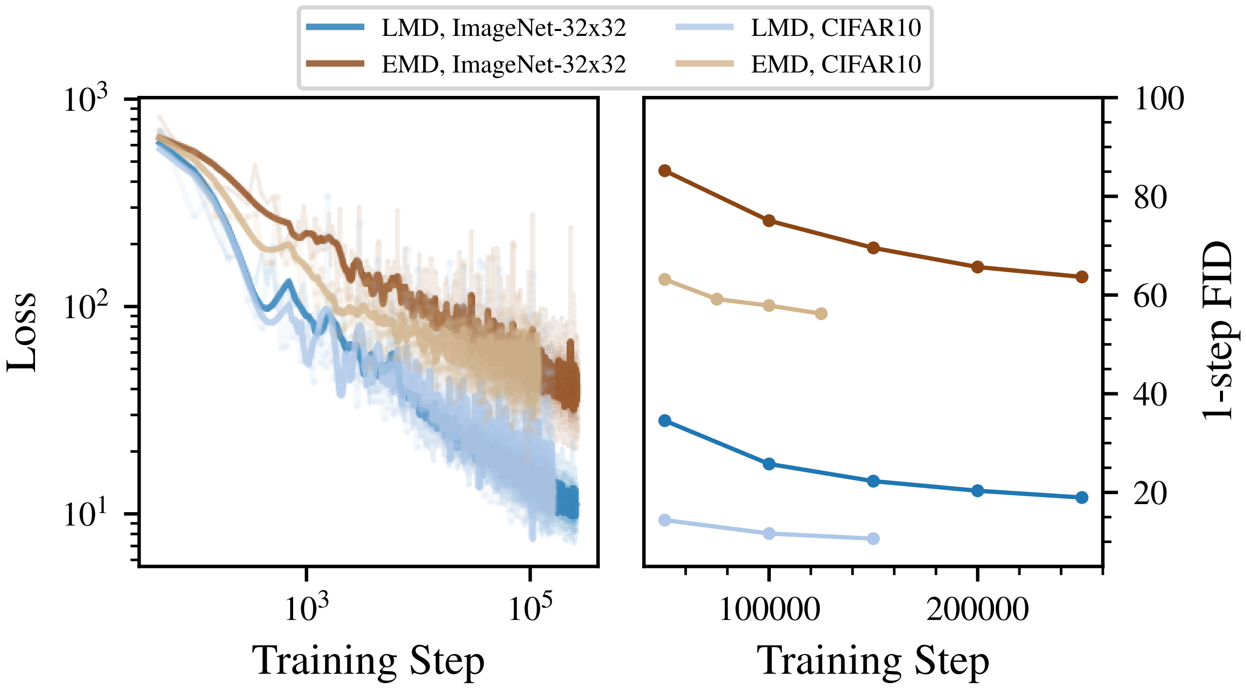


One to few step map matching and Lagrangian distillation on par with 80-step interpolant

Eulerian Map Distillation struggles

How do they compare?

Produce samples in much fewer steps than through solving the ODE (SI)



Lagrangian distillation converges faster than Eulerian

Does this make sense theoretically? What can we say about the loss functions

Wasserstein Control on Distillation Losses

Let $\rho_1^b = X_{0,1}\#\rho_0$ and $\hat{\rho}_1 = \hat{X}_{0,1}\#\rho_0$. Then the squared Wasserstein distance $W_2^2(\rho_1^b, \hat{\rho}_1)$ satisfies

Lagrangian Bound

$$W_2^2(\rho_1^b, \hat{\rho}_1) \leqslant e^{1+2\int_0^1 |C_t| dt} L_{LMD}(\hat{X})$$

Eulerian Bound

$$W_2^2(\rho_1^b, \hat{\rho}_1) \leqslant e L_{EMD}(\hat{X})$$

Eulerian bound much tighter!

- Bringing L_{LMD} and L_{EMD} to same value would imply better learning for EMD
- But empirically, *optimization is harder!* Bounds useful, but don't tell whole story.