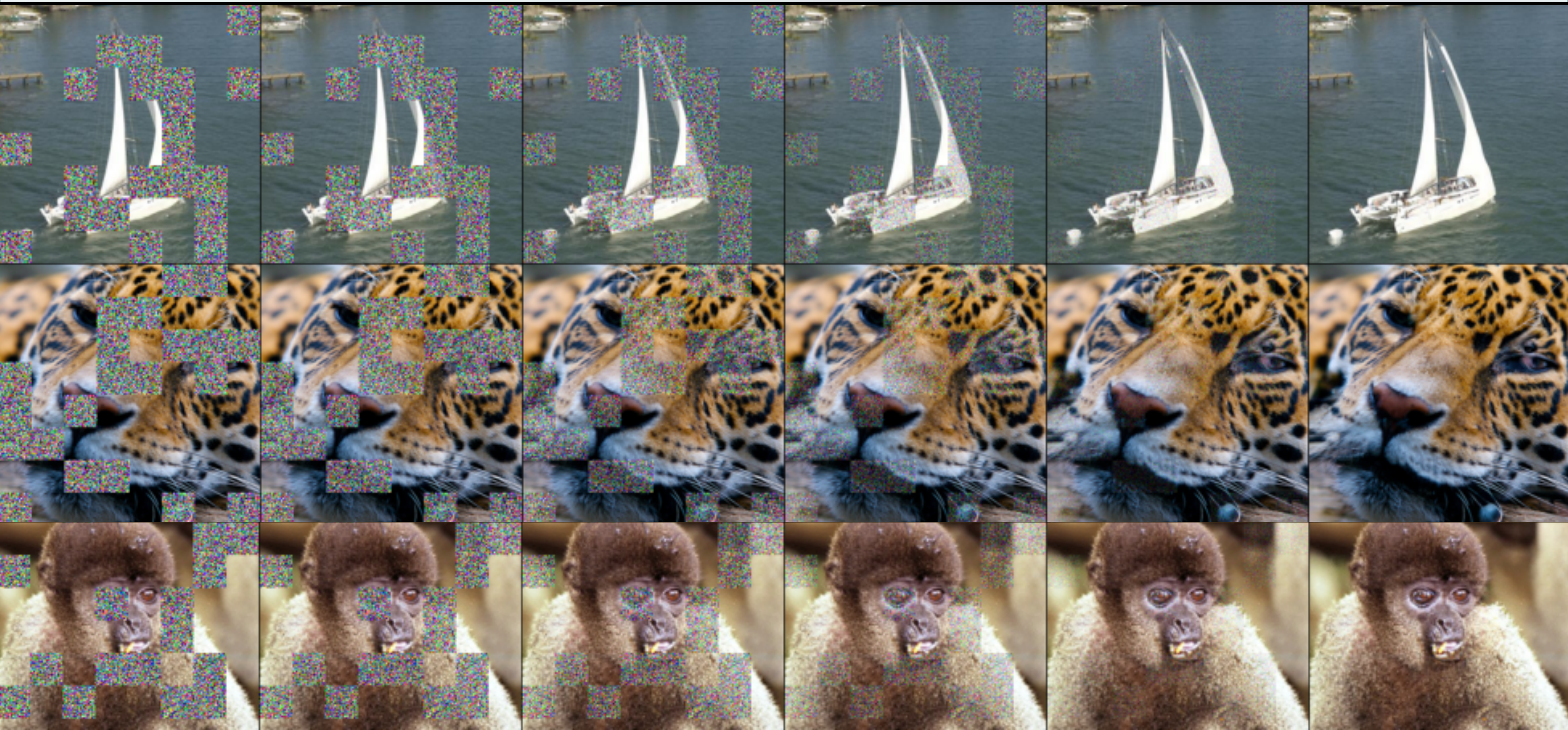
Paris Bachelier Seminar
Institut Henri Poincaré, Paris Oct 3 2025

Courant NYU CFM



# Beyond Diffusions with Stochastic Interpolants

Eric Vanden-Eijnden

*with Michael Albergo, Nick Boffi, Yifan Chen, Florentin Coeurdoux, Mengjian Hua, Mark Goldstein, Willis Ma, Hugo Negrel, Rajesh Ranganath, and Saining Xie*

Black Forest Labs

**Prompt**: Epic artwork of a massive brutalist building floating above a favela in a tropical landscape, the large brutalist building has large wires and cables hanging from it, cinematic art

# De novo design of protein structure and function with RFdiffusion

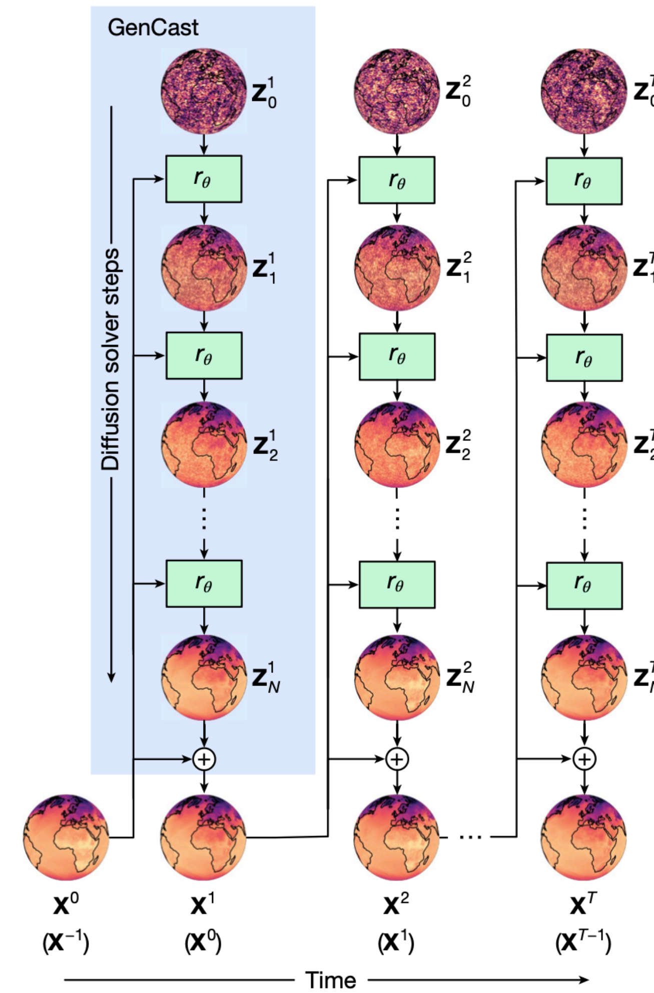
Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, ... David Baker + Show authors

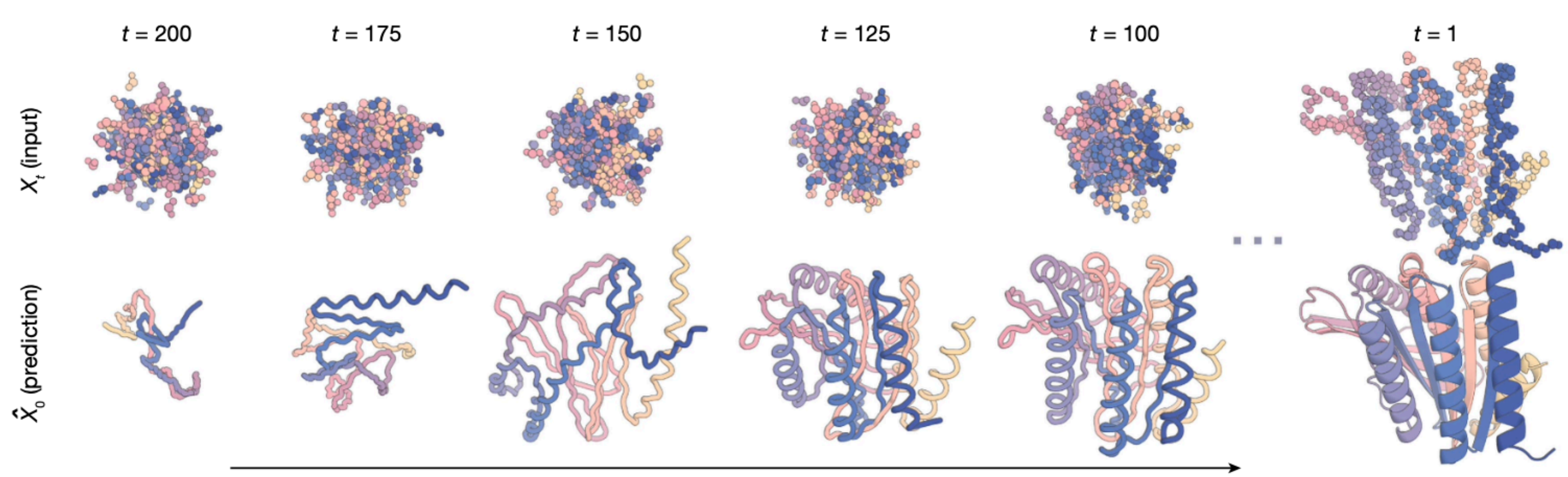
*Nature* **620**, 1089–1100 (2023) | Cite this article

# Probabilistic weather forecasting with machine learning

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam & Matthew Willson

*Nature* **637**, 84–90 (2025) | Cite this article



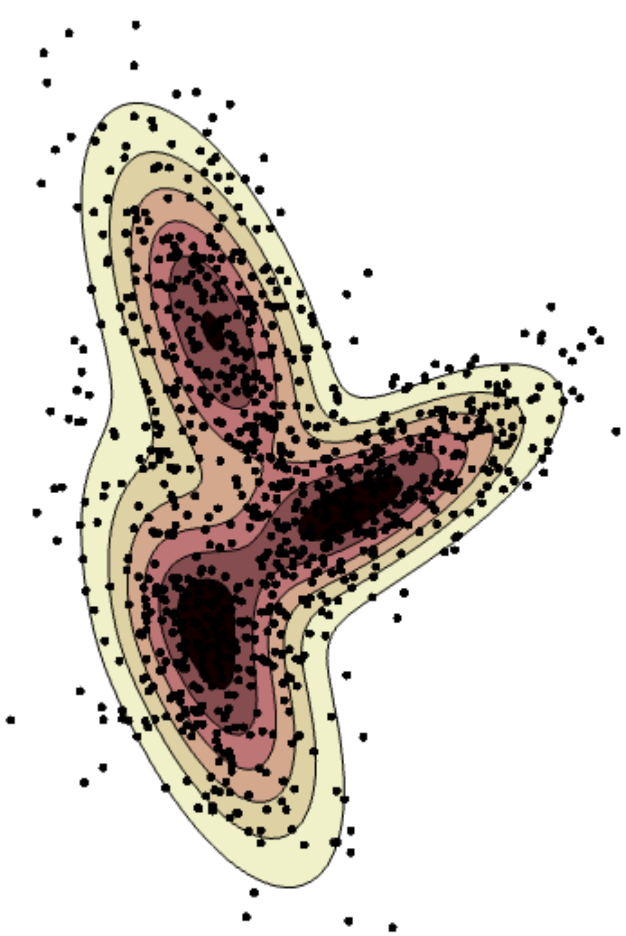



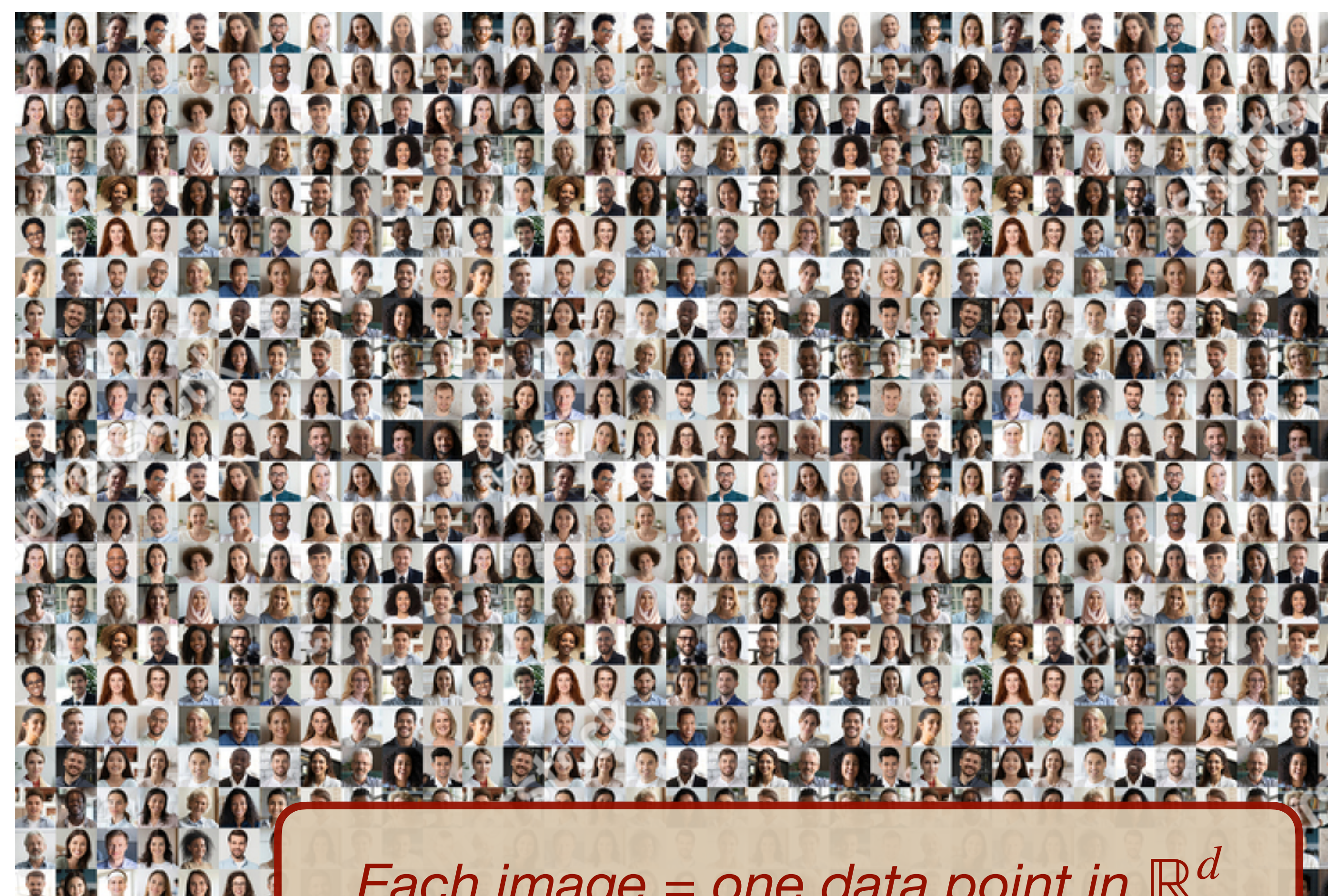
# Probabilistic Approach to Unsupervised Learning

***Working assumption to organize unlabeled data:***

*View the* ***data points*** *as* ***samples*** *from an* ***unknown probability distribution****.*

***Learn this distribution*** *in a way that* ***allows for generation of new samples***





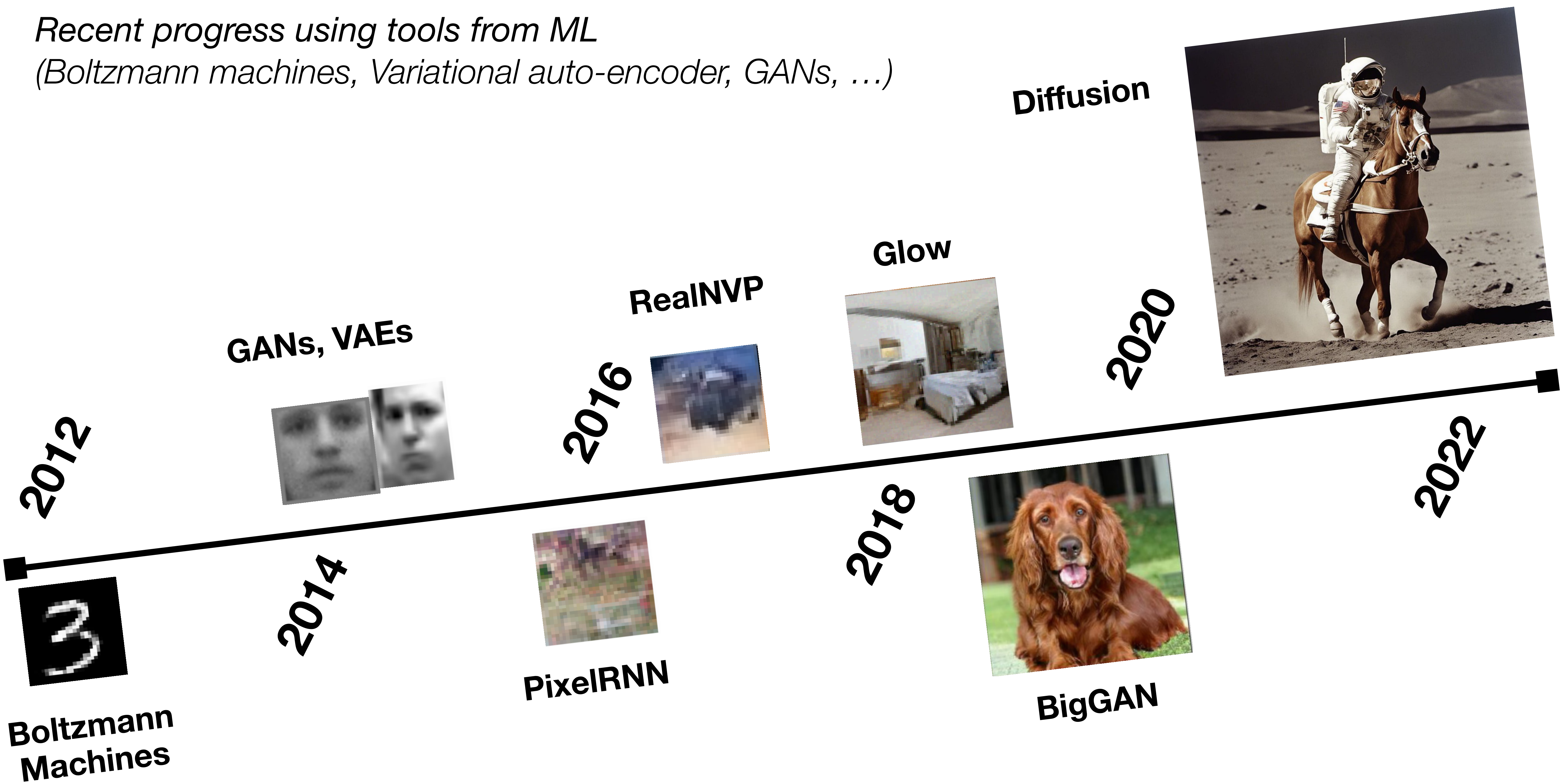
*Each image = one data point in $\mathbb{R}^d$*

***Density estimation in high dimension!***

# Density Estimation

*Old problem, intractable with traditional methods in high dimension*
*(binning, kernel density estimation,…)*

*Recent progress using tools from ML*
*(Boltzmann machines, Variational auto-encoder, GANs, …)*

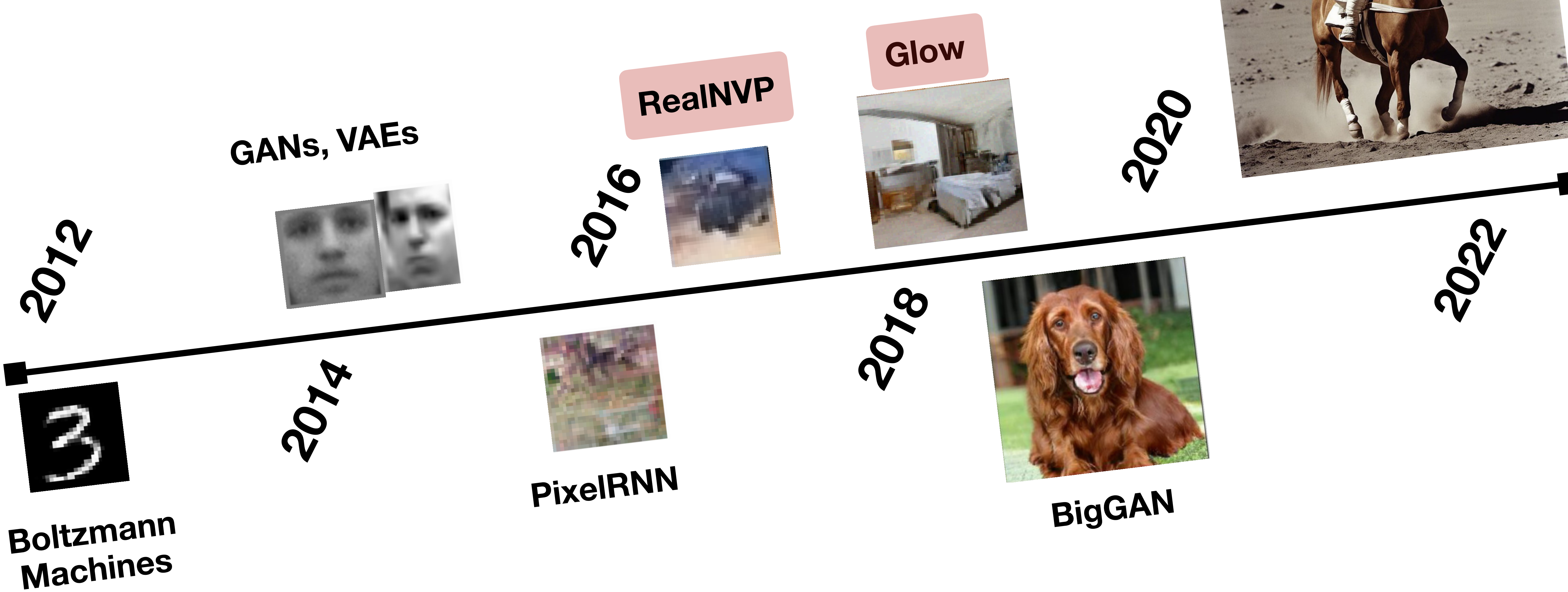


# Density Estimation and Measure Transport

*Old problem, intractable with traditional methods in high dimension*
*(binning, kernel density estimation,…)*

*Recent progress using tools from ML*
*(Boltzmann machines, Variational auto-encoder, GANs, …)*

***Breakthrough via transportation of measure***

# Generation with Flows and Diffusions

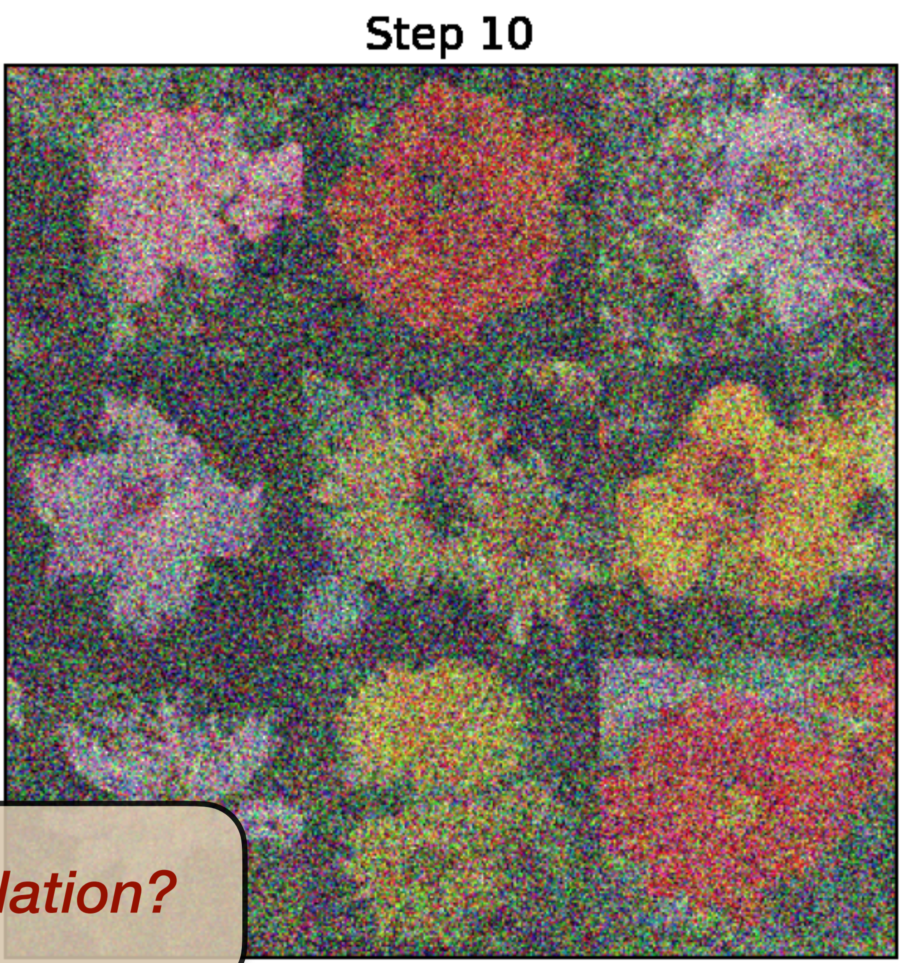
**Aim:** Construct an S/ODE ***Flow/diffusion matching***

$$dX_t = b_t(X_t)dt + \sigma_t dW_t$$

such that: *if* $X_{t=0} \sim \mu_0$ = *(simple) base distribution, then* $X_{t=1} \sim \mu_1$ = *target distribution.*

***Dynamical transport of probability distributions***

*Benamou-Brenier, …*




*Well-suited for **generation** and **sampling**:*

- *draw a sample from the base $\mu_0$;*
- *propagate it through the S/ODE;*
- *get a sample from the target $\mu_1$.*

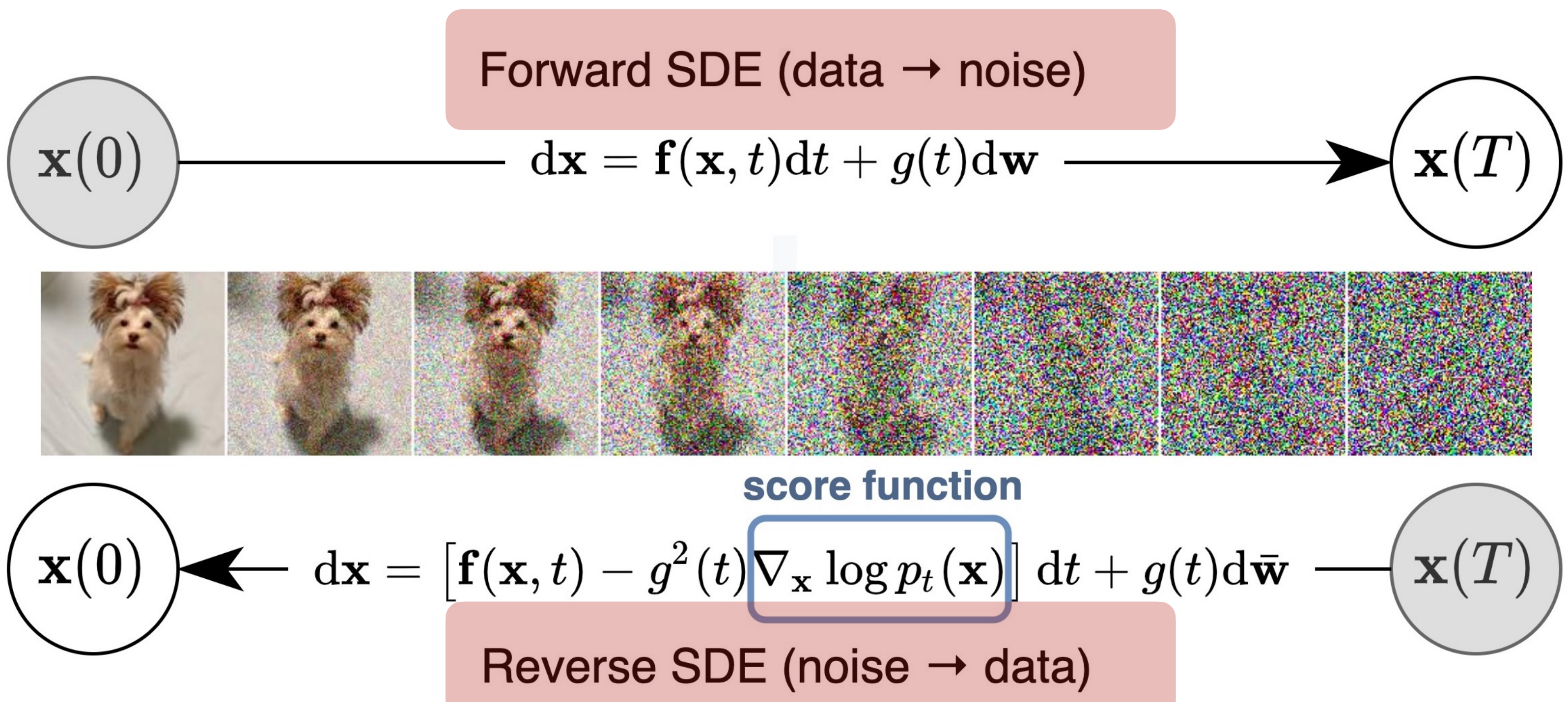
***How can we find $b_t(x)$ using a tractable variational formulation?***

*Sohl-Dickstein et al. arXiv:1503.03585 (2015);*
*Ho et al. arXiv:2006.11239 (2020);*
*Song et al. arXiv:2011.13456 (2021)*

# Score-Based Diffusion Models

**Given samples from the data distribution $\mu_1$:**

- *Devolve them into Gaussian noise* using e.g. an Ornstein-Ulhenbeck process;
- *Time-reverse the SDE* to generate new samples from $\mu_1$ from samples from $N(0,Id)$;




*From Song's blog*

*Builds a **path in distribution space** between $\mu_1$ and $N(0,Id)$;*
*Reduces problem to the **simulation-free regression of the score**.*

*Hyvärinen JMLR **6** (2005);*
*Vincent, Neural Comp. **23**, 1661 (2011)*

# Stochastic Interpolants

*Albergo & V.-E. arXiv:2209.15571 (2022);*
*Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).*

*See also: Liu et al. arXiv:2209.03003 (2022);*
*Lipman et al. arXiv:2210.02747 (2022):*

***Key idea: Build a process that connects a base distribution to the target***

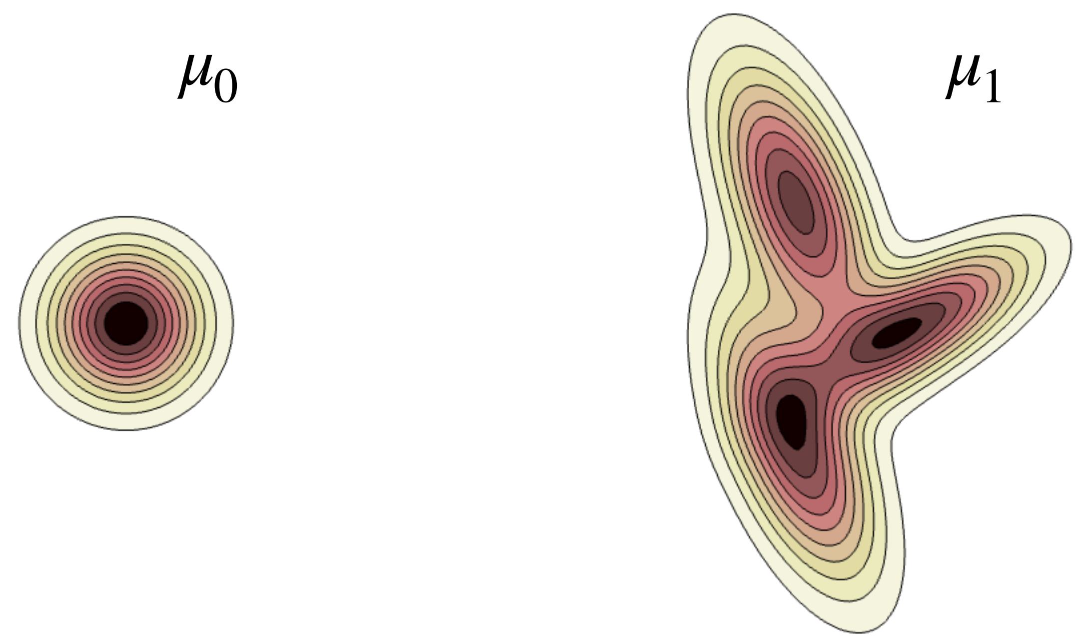
The *stochastic interpolant* $I_t$ is the process:

$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

with: $x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0,Id), \quad z \perp (x_0, x_1)$

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$

e.g. $I_t = (1-t)x_0 + t x_1$ with $(x_0, x_1) \sim \mu_0 \otimes \mu_1$




***By definition:*** $I_0 = x_0 \sim \mu_0$ **,** $I_1 = x_1 \sim \mu_1$

$I_t$ ***easy to sample at all*** $t \in [0,1]$ ***using data***

# Stochastic Interpolants

Albergo & V.-E. arXiv:2209.15571 (2022);
Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).

See also: Liu et al. arXiv:2209.03003 (2022);
Lipman et al. arXiv:2210.02747 (2022):

***Key idea: Build a process that connects a base distribution to the target***

The *stochastic interpolant* $I_t$ is the process:

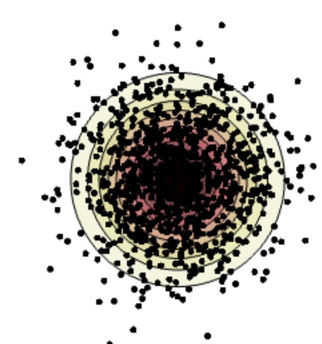
$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

with: $$x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0, Id), \quad z \perp (x_0, x_1)$$

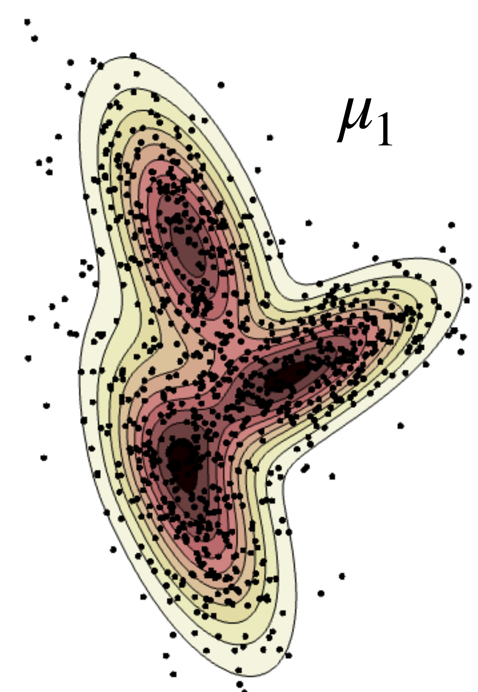
$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$

e.g. $I_t = (1-t)x_0 + tx_1$ with $(x_0, x_1) \sim \mu_0 \otimes \mu_1$

$\mu_0$



$\mu_1$



with samples

$x_0 \sim \mu_0$

$x_1 \sim \mu_1$

***By definition:*** $I_0 = x_0 \sim \mu_0$ **,** $I_1 = x_1 \sim \mu_1$

$I_t$ ***easy to sample at all*** $t \in [0,1]$ ***using data***

# Stochastic Interpolants

*Albergo & V.-E. arXiv:2209.15571 (2022);*
*Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).*

*See also: Liu et al. arXiv:2209.03003 (2022);*
*Lipman et al. arXiv:2210.02747 (2022):*

***Key idea: Build a process that connects a base distribution to the target***

The *stochastic interpolant* $I_t$ is the process:

$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

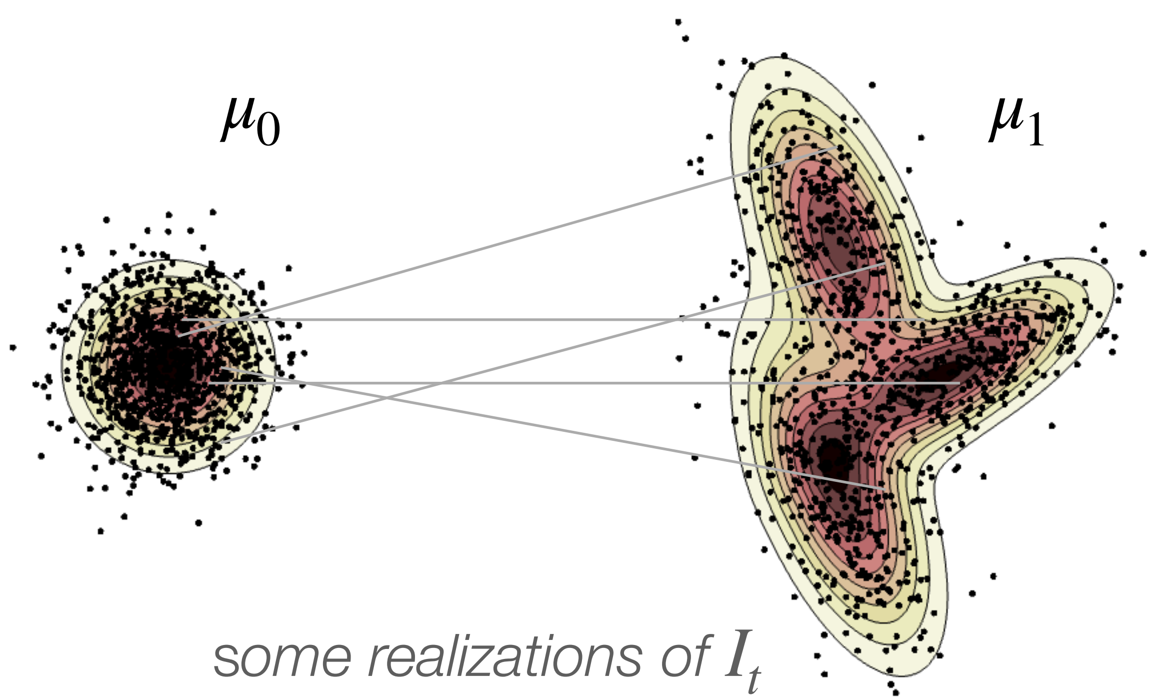
with: $x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0,Id), \quad z \perp (x_0, x_1)$

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$

e.g. $I_t = (1-t)x_0 + tx_1$ with $(x_0, x_1) \sim \mu_0 \otimes \mu_1$

***By definition:*** $I_0 = x_0 \sim \mu_0$ **,** $I_1 = x_1 \sim \mu_1$

$I_t$ ***easy to sample at all*** $t \in [0,1]$ ***using data***



# Stochastic Interpolants

*Albergo & V.-E. arXiv:2209.15571 (2022);*
*Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).*

*See also: Liu et al. arXiv:2209.03003 (2022);*
*Lipman et al. arXiv:2210.02747 (2022):*

***Key idea: Build a process that connects a base distribution to the target***

The *stochastic interpolant* $I_t$ is the process:

$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

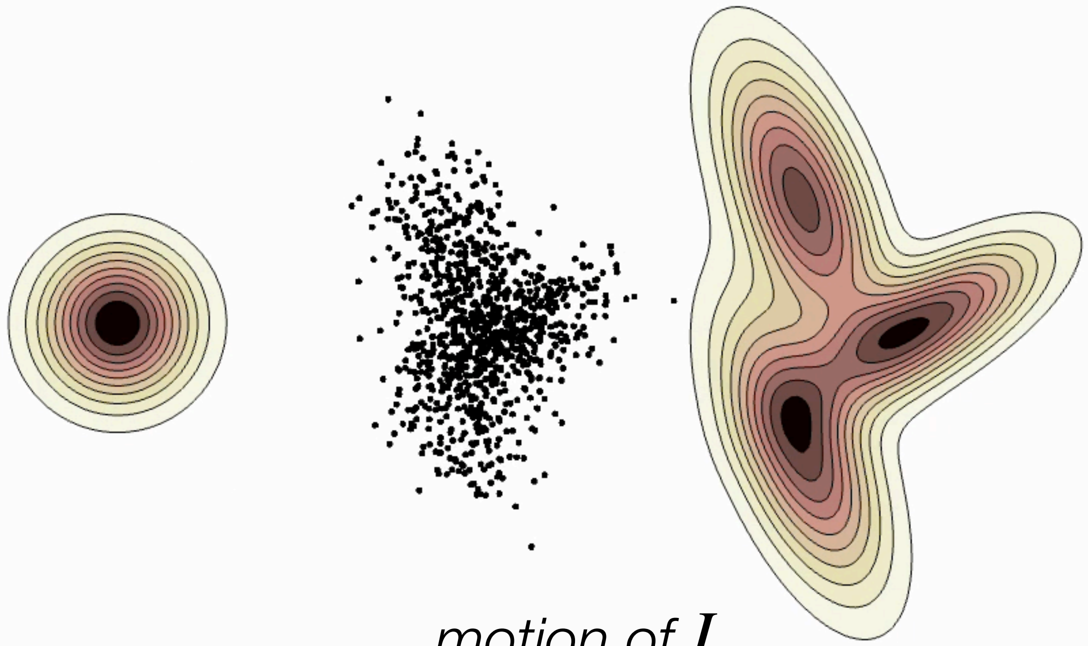
with: $$x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0, Id), \quad z \perp (x_0, x_1)$$

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$

e.g. $I_t = (1-t)x_0 + tx_1$ with $(x_0, x_1) \sim \mu_0 \otimes \mu_1$

***By definition:*** $I_0 = x_0 \sim \mu_0$ **,** $I_1 = x_1 \sim \mu_1$

$I_t$ ***easy to sample at all*** $t \in [0,1]$ ***using data***



*motion of* $I_t$

# Stochastic Interpolants

*Albergo & V.-E. arXiv:2209.15571 (2022);*
*Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).*

*See also: Liu et al. arXiv:2209.03003 (2022);*
*Lipman et al. arXiv:2210.02747 (2022):*

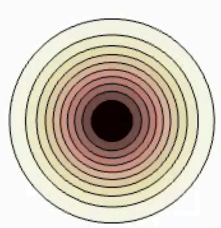
***Key idea: Build a process that connects a base distribution to the target***

The *stochastic interpolant* $I_t$ is the process:

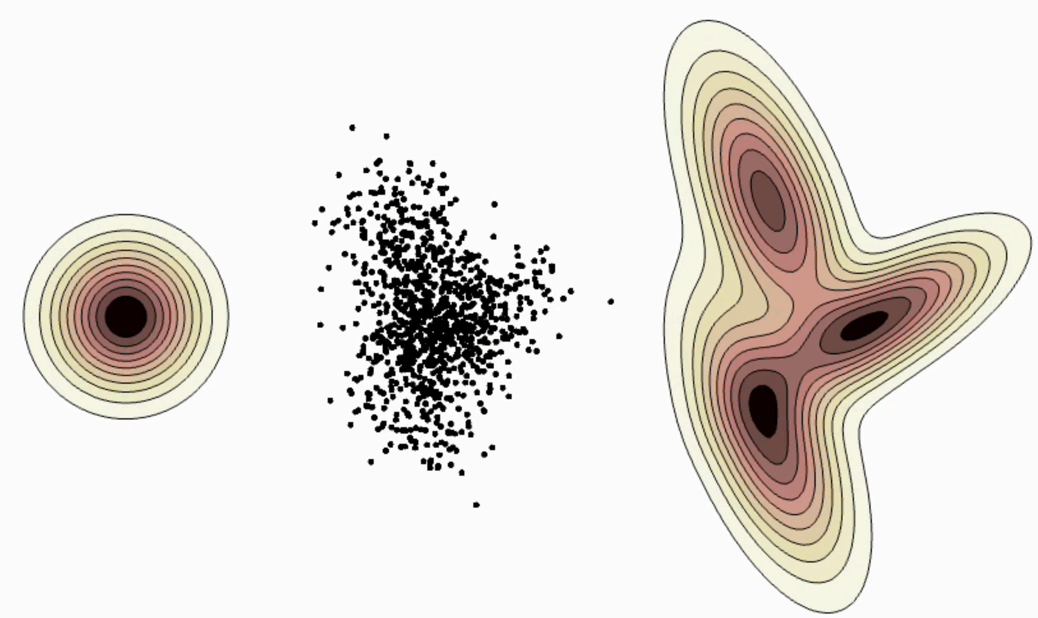
$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

with: $$x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0,Id), \quad z \perp (x_0, x_1)$$

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_0 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$



*motion of* $\mu_t =$ *law*$(I_t)$



*motion of* $I_t$

# Stochastic Interpolants

*Albergo & V.-E. arXiv:2209.15571 (2022);*
*Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).*

*See also: Liu et al. arXiv:2209.03003 (2022);*
*Lipman et al. arXiv:2210.02747 (2022):*

The *stochastic interpolant* $I_t$ is the process:

$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

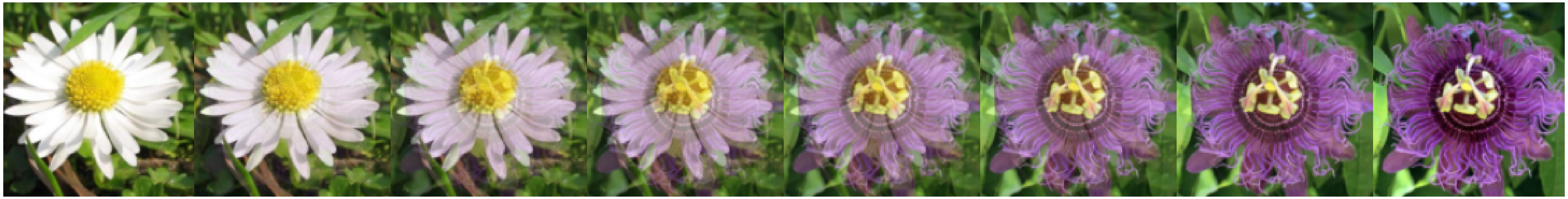
with:

$$x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0, Id), \quad z \perp (x_0, x_1)$$

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_1 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$

***Note that:***

- ***$I_t$ is not necessarily a diffusion;***
- ***base distribution does not need to be Gaussian;***
- ***$(x_0, x_1)$ can be correlated.***



# Stochastic Interpolants

Albergo & V.-E. arXiv:2209.15571 (2022);
Albergo, Boffi, & V.-E. arXiv:2303.08797 (2023).

See also: Liu et al. arXiv:2209.03003 (2022);
Lipman et al. arXiv:2210.02747 (2022):

The *stochastic interpolant* $I_t$ is the process:

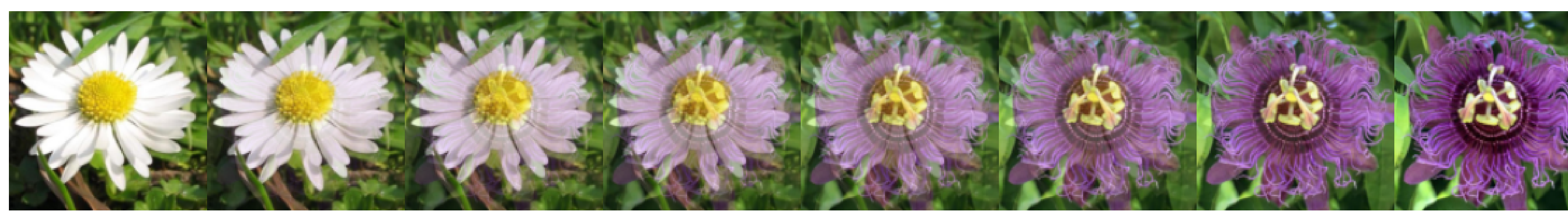
$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad t \in [0,1]$$

with: $$x_0 \sim \mu_0, \quad x_1 \sim \mu_1, \quad z \sim N(0, Id), \quad z \perp (x_0, x_1)$$

$$\alpha_0 = \beta_1 = 1, \quad \alpha_1 = \beta_1 = \gamma_0 = \gamma_1 = 0, \qquad \alpha, \beta, \gamma \in C^1(0,1).$$

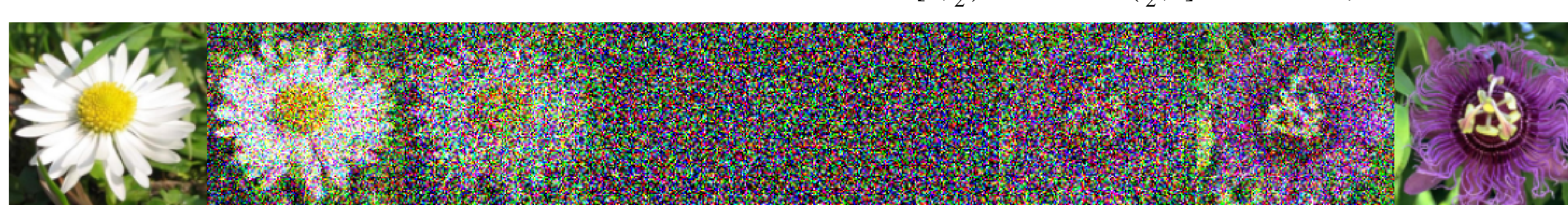
*Different $\alpha_t$, $\beta_t$, and $\gamma_t$ give different processes.*

*Without latent variable* $\qquad x_t = (1-t)x_0 + tx_1$



*With latent variable* $\qquad x_t = (1-t)x_0 + tx_1 + \sqrt{2t(1-t)}z$

*Gaussian encoding-decoding* $\qquad x_t = \cos^2(\pi t)(1_{[0,\frac{1}{2})}(t)x_0 + 1_{(\frac{1}{2},1]}(t)x_1) + \sqrt{2t(1-t)}z$



# Stochastic Interpolants and Transport

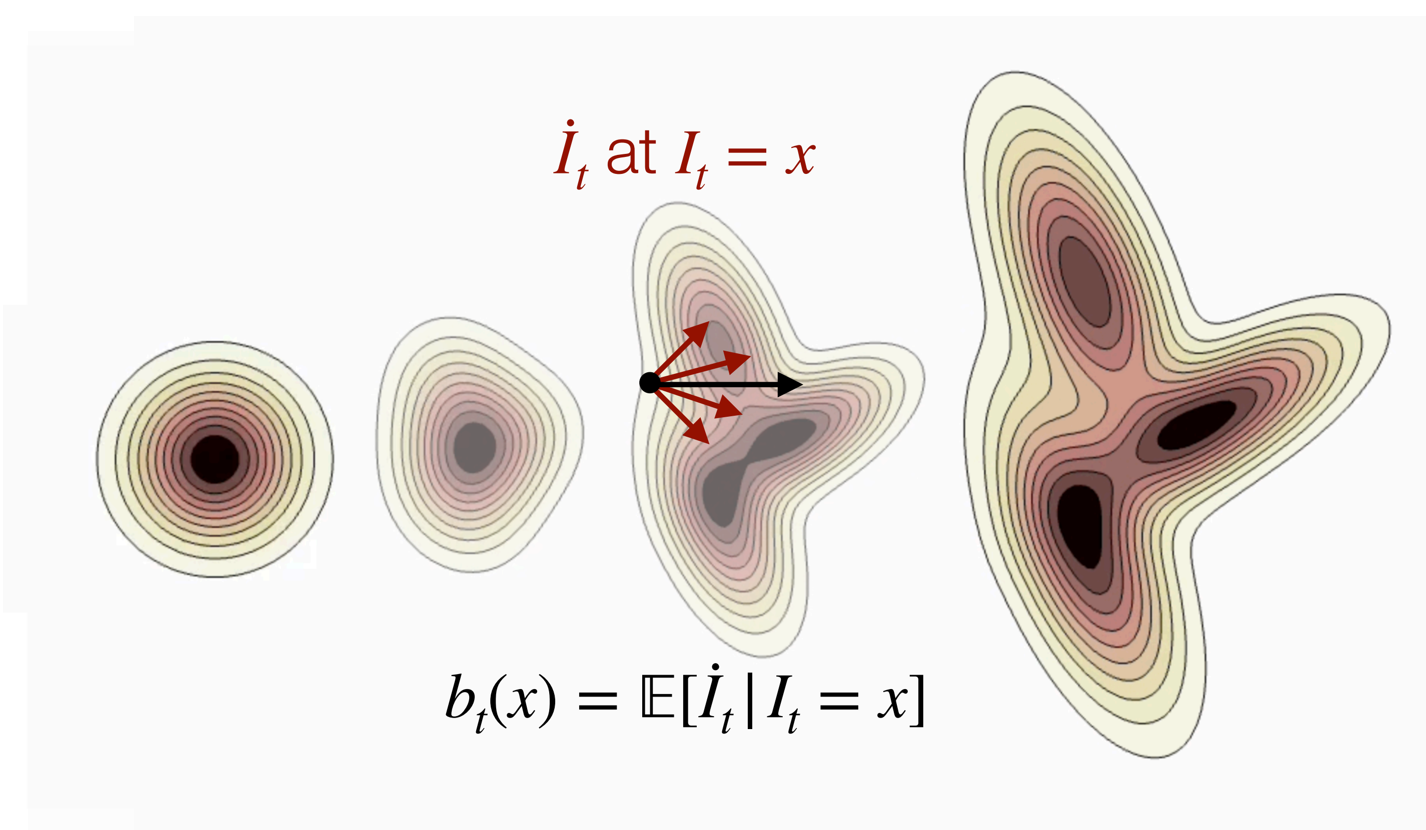
**Thm:** the law of $I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z$ is the same as the law of the solution to

$$\dot{X}_t = b_t(X_t), \qquad X_0 \sim \mu_0$$

*Probability flow ODE*

with the velocity $b_t(x)$ given by the conditional expectation

$$b_t(x) = \mathbb{E}\left[\dot{I}_t \,\middle|\, I_t = x\right] = \underset{\hat{b}_t}{argmin}\, \mathbb{E}\left[\,|\hat{b}_t(I_t) - \dot{I}_t|^2\right]$$



# Stochastic Interpolants and Transport

**Thm:** the law of $I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z$ is the same as the law of the solution to

$$\dot{X}_t = b_t(X_t), \qquad X_0 \sim \mu_0$$

*Probability flow ODE*

with the velocity $b_t(x)$ given by the conditional expectation

$$b_t(x) = \mathbb{E}\left[\dot{I}_t \,\middle|\, I_t = x\right] = \underset{\hat{b}_t}{argmin}\, \mathbb{E}\left[|\hat{b}_t(I_t) - \dot{I}_t|^2\right]$$

**Proof:** if $\mu_t$ is the distribution of $I_t$ and $\phi$ is a test function, we have

$$\int_{\mathbb{R}^d} \phi(x)\mu_t(dx) = \mathbb{E}[\phi(I_t)]$$

and so:

$$\begin{array}{ccc} \dfrac{d}{dt}\mathbb{E}[\phi(I_t)] & = & \mathbb{E}[\dot{I}_t \cdot \nabla\phi(I_t)] \\ \shortparallel & & \shortparallel \\ \displaystyle\int_{\mathbb{R}^d} \phi(x)\partial_t\mu_t(dx) & = & \displaystyle\int_{\mathbb{R}^d} \underbrace{\mathbb{E}[\dot{I}_t | I_t = x]}_{=b_t(x)} \cdot \nabla\phi(x)\mu_t(dx) \end{array}$$

# Stochastic Interpolants and Transport

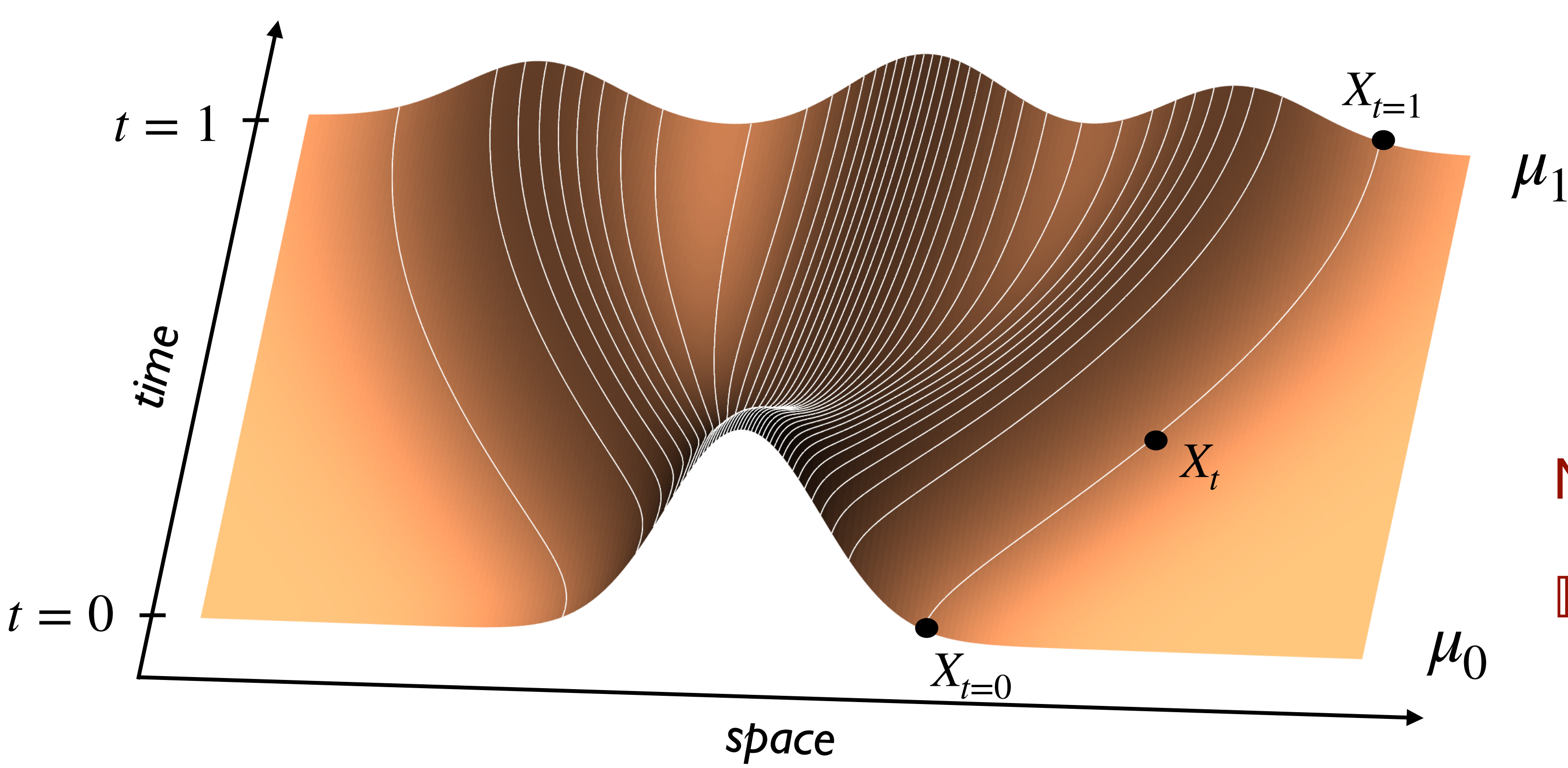
**Thm:** the law of $I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z$ is the same as the law of the solution to

$$\dot{X}_t = b_t(X_t), \qquad X_0 \sim \mu_0$$

*Probability flow ODE*

with the velocity $b_t(x)$ given by the conditional expectation

$$b_t(x) = \mathbb{E}\left[\dot{I}_t \,\middle|\, I_t = x\right] = \underset{\hat{b}_t}{argmin}\, \mathbb{E}\left[|\hat{b}_t(I_t) - \dot{I}_t|^2\right]$$




***Gives a generative model:***

$$X_{t=0} \sim \mu_0 \quad \Leftrightarrow \quad X_{t=1} \sim \mu_1$$

Not OT but finite path length in $W_2$:

$$\mathbb{E}\left[|X_{t=1}(x_0) - x_0|^2\right] \leq \int_0^1 \mathbb{E}\left[|\dot{I}_t|^2\right] dt < \infty$$

# Stochastic Interpolants and Transport

**Thm:** the law of $I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z$ is the same as the law of the solution to

$$\dot{X}_t = b_t(X_t), \qquad X_0 \sim \mu_0$$

*Probability flow ODE*

with the velocity $b_t(x)$ given by the conditional expectation

$$b_t(x) = \mathbb{E}\left[\dot{I}_t \,\middle|\, I_t = x\right] = \underset{\hat{b}_t}{argmin}\, \mathbb{E}\left[|\hat{b}_t(I_t) - \dot{I}_t|^2\right]$$

$$b = \underset{\hat{b}}{argmin} \int_0^1 \mathbb{E}\left[|\hat{b}_t(I_t) - \dot{I}_t|^2\right] dt$$

***Estimation of $b$ = simulation-free regression problem***

- Objective $L_b(\hat{b})$ and its gradient can be evaluated empirically using the samples $I_t$;
- Velocity $b_t(x)$ can be approximated e.g. by deep neural network (DNN);
- Minimization can be performed by SGD.

# Score and diffusions

**Thm:** The score $s_t(x) = \nabla \log[d\mu_t/dx]$ of the PDF of $I_t$ is given for all $t \in (0,1)$ by

$$s_t(x) = -\gamma_t^{-1}\mathbb{E}[z \mid I_t = x]$$

*Stein's identity*

In addition, it is the unique minimizer of

$$L_s(\hat{s}) = \int_0^1 \mathbb{E}\left[ |\hat{s}_t(I_t)|^2 + 2\gamma_t^{-1} z \cdot \hat{s}_t(I_t) \right] dt$$

**Corr:** For any $\epsilon_t \geq 0$, $\mu_t(x)$ solves:

$$\partial_t \mu_t + \nabla \cdot ([b_t(x) + \epsilon_t s_t(x)]\mu_t) = \epsilon_t \Delta \mu, \qquad \mu_{t=0} = \mu_0,$$

and the solutions to the SDE

$$dX_t^F = b_t(X_t^F)dt + \epsilon_t s_t(X_t^F)dt + \sqrt{2\epsilon_t}dW_t$$

are such that

$$X_{t=0}^F \sim \mu_0 \quad \Rightarrow \quad X_{t=1}^F \sim \mu_1$$

*Diffusion coefficient $\epsilon_t$ adjustable post training*

Proof: Using $\nabla \cdot (s_t \mu_t) = \Delta \mu_t$ , we see that this is the same equation as $\partial_t \mu_t + \nabla \cdot (b_t \mu_t) = 0$

# Score and diffusions

**Corr:** For any $\epsilon_t \geq 0$, the solutions to the SDE

$$dX_t^F = b_t(X_t^F)dt + \epsilon_t s_t(X_t^F)dt + \sqrt{2\epsilon_t}dW_t$$

are such that

$$X_{t=0}^F \sim \mu_0 \quad \Rightarrow \quad X_{t=1}^F \sim \mu_1$$

*Diffusion coefficient $\epsilon_t$ adjustable post training*

**Lem:** If $\quad I_t = \alpha_t x_0 + \beta_t x_1, \quad x_0 \sim N(0,\mathrm{Id}), \quad x_1 \sim \mu_1, \quad x_0 \perp x_1,$

then

$$\alpha_t s_t(x) = \frac{\beta_t b_t(x) - \dot\beta_t x}{\alpha_t \dot\beta_t - \dot\alpha_t \beta_t}$$

*Only need to learn $b_t(x)$*

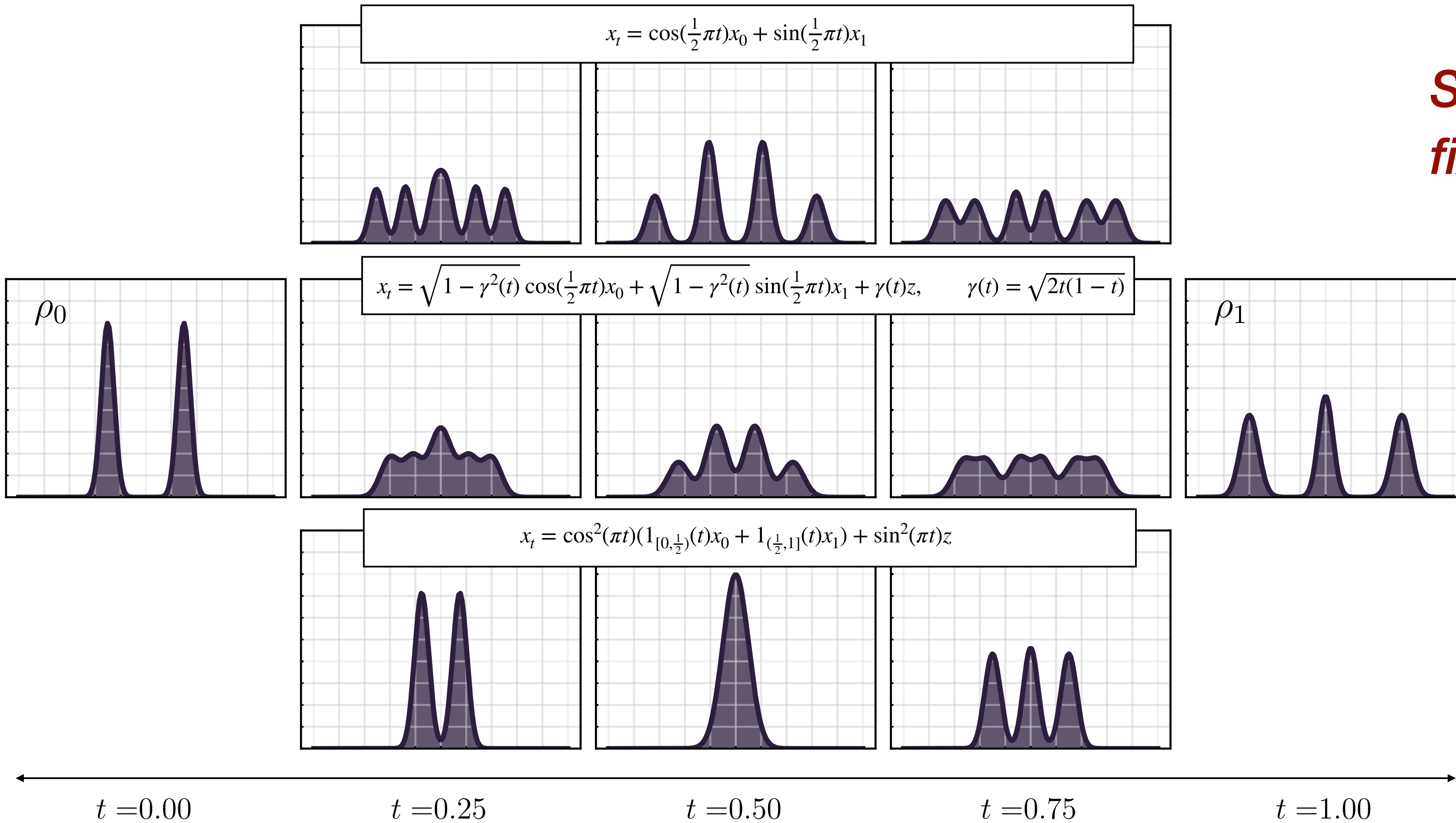
Proof: Use

$$b_t(x) = \mathbb{E}[\dot I_t \,|\, I_t = x] = \dot\alpha_t \mathbb{E}[x_0 \,|\, I_t = x] + \dot\beta_t \mathbb{E}[x_1 \,|\, I_t = x]$$

$$s_t(x) = -\alpha_t^{-1}\mathbb{E}[x_0 \,|\, I_t = x]$$

together with $\quad x = \mathbb{E}[I_t \,|\, I_t = x] = \alpha_t \mathbb{E}[x_0 \,|\, I_t = x] + \beta_t \mathbb{E}[x_1 \,|\, I_t = x]$

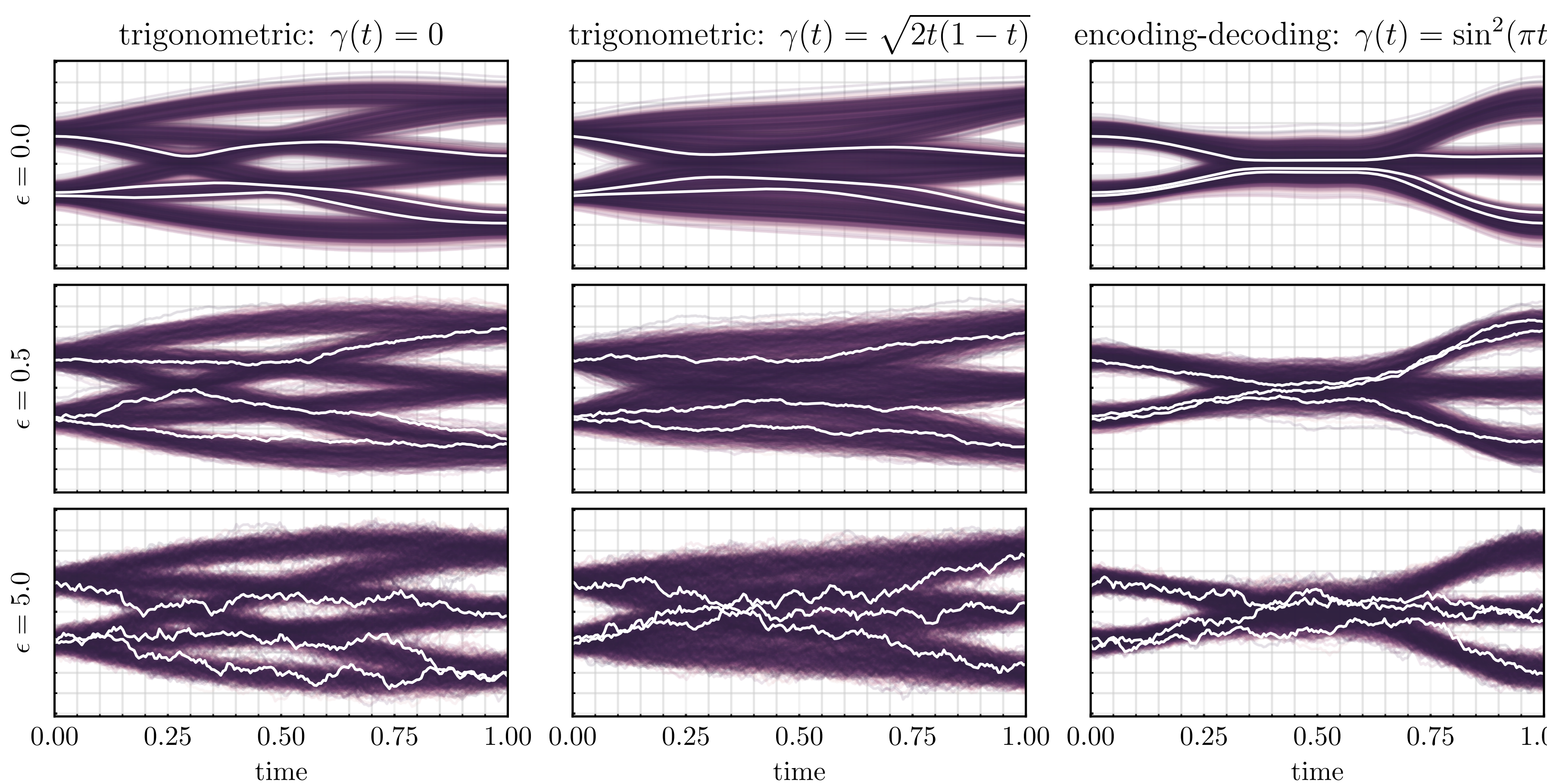
*Stochastic interpolant*
*fixes the connecting distribution $\mu_t$*

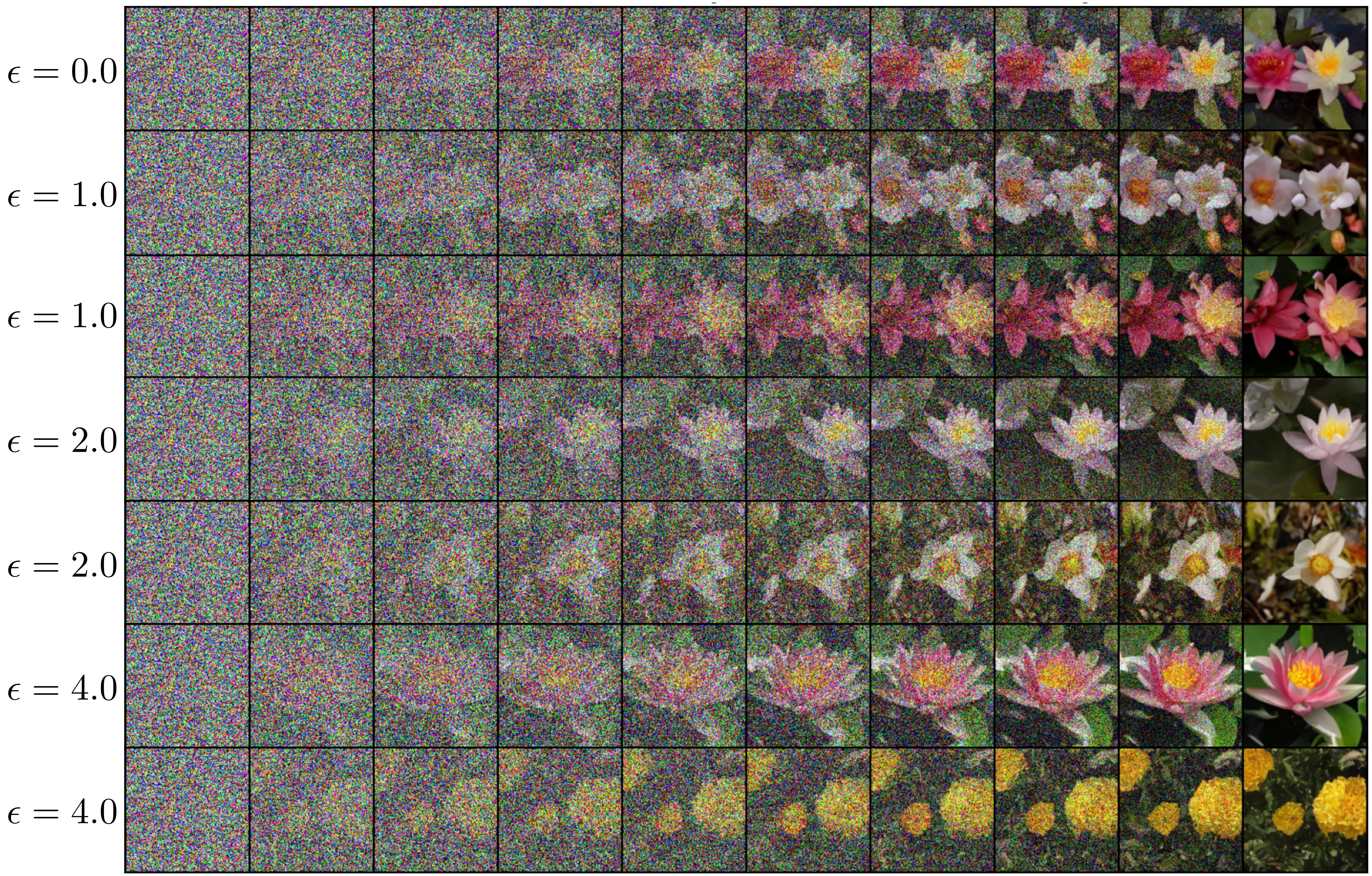



*Diffusion coefficient $\epsilon_t$*
*controls the way to sample $\mu_t$*

*ODE and SDE sample*
*the **same** $\mu_t$ in different ways*

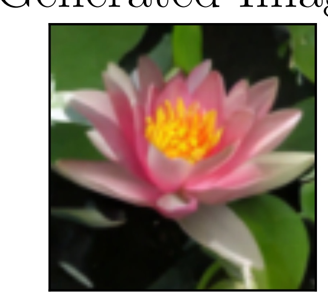
*ODE gives one-to-one map;*
*SDE samples $\mu_1$ more broadly*
*from any $x_0 \sim \mu_0$*



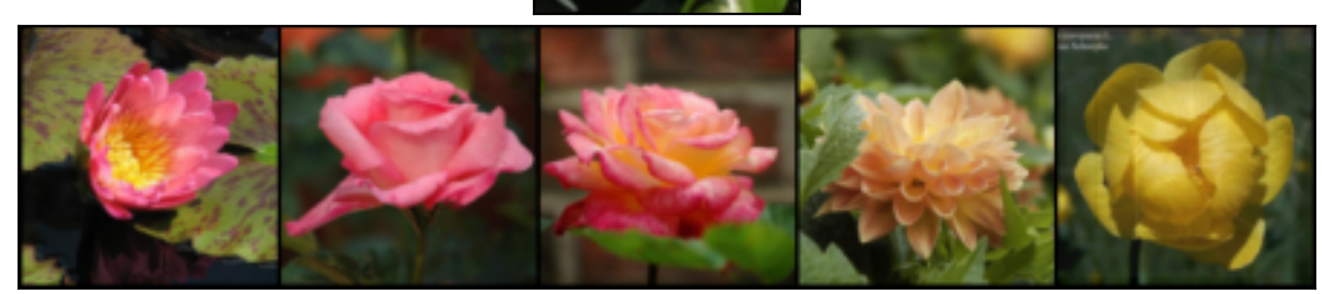



*More noise in SDE = more diversity in outputs from same $x_0 \sim \mu_0$*

Generated Image



*Does better than memorizing the data set!*



5 Nearest neighbors in training set

# Scalable Interpolant Transformers

*with Ma, Albergo, Boffi, Goldstein & Xie (2023)*

https://scalable-interpolant.github.io/

| 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** |

*ImageNet 512x512*

*ImageNet 256x256*

Technology used by Stability AI, BlackForest Labs, Meta Movie Gen, etc..

# Conditional Generation

**Thm:** Given $(x_0, x_1, \xi) \sim \mu(dx_0, dx_1, d\xi)$, let

*$\xi$ = conditioning variables*

$$I_t = \alpha_t x_0 + \beta_t x_1 + \gamma_t z, \qquad z \sim N(0,\text{Id}), \quad z \perp (x_0, x_1, \xi),$$

and define $b_t(x, \xi) = \mathbb{E}\left[\dot\alpha_t x_0 + \dot\beta_t x_1 + \dot\gamma_t z \,|\, I_t = x, \xi\right]$ and $s_t(x, \xi) = -\gamma_t^{-1}\mathbb{E}\left[z \,|\, I_t = x, \xi\right]$

Then, for any $\epsilon_t \geq 0$ the solutions to

$$dX_t = b_t(X_t, \xi)dt + \epsilon_t s_t(X_t, \xi)dt + \sqrt{2\epsilon_t}dW_t, \qquad X_{t=0} \sim \mu_0(dx_0 \,|\, \xi)$$

are such that

$$X_{t=1} \sim \mu(dx_1 \,|\, \xi) = \textit{conditional measure of } x_1 \textit{ given } \xi$$

In addition, $b_t(x, \xi)$ is the unique minimizer of

$$L_b(\hat b) = \int_0^1 \mathbb{E}\left[\,|\hat b_t(I_t, \xi)|^2 - 2(\dot\alpha_t x_0 + \dot\beta_t x_1 + \dot\gamma_t z)\cdot \hat b_t(I_t, \xi)\right] dt$$

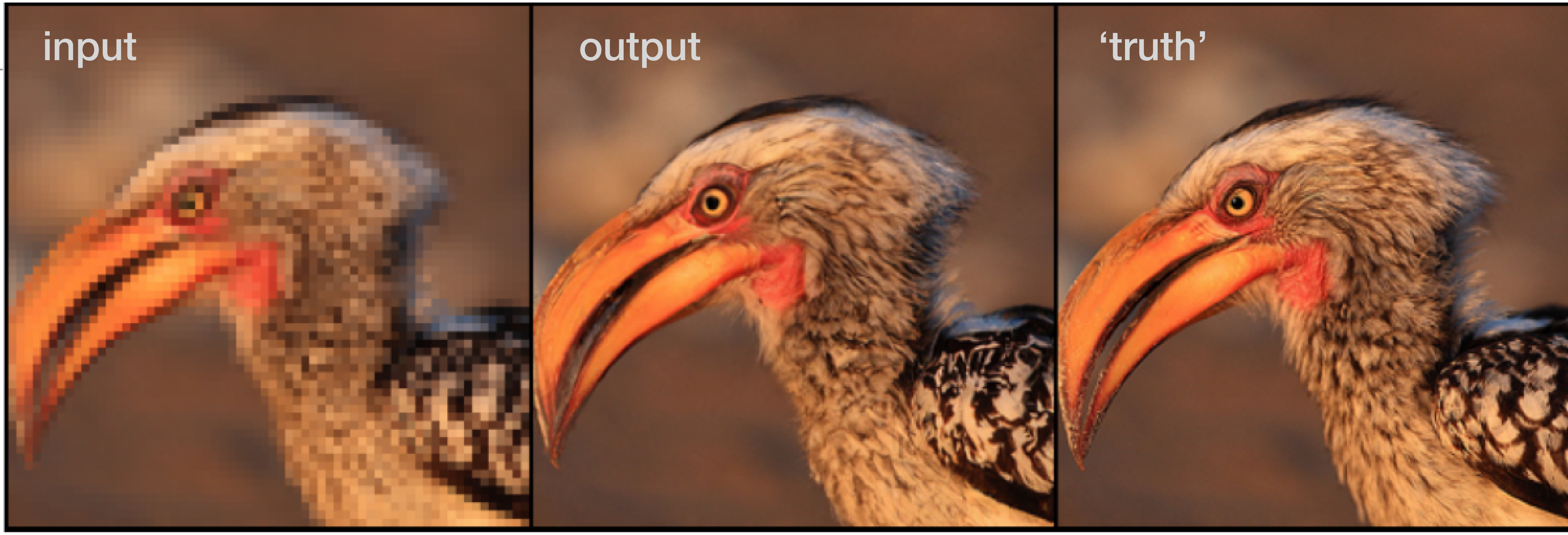
***Similar story as before, with $\epsilon_t$ adjustable post-training***

# Superresolution

*target = high-res image*
*base = low-res image + noise*

*velocity conditioned on low-res image*

Albergo et al. arXiv:2310.03725 (2023)

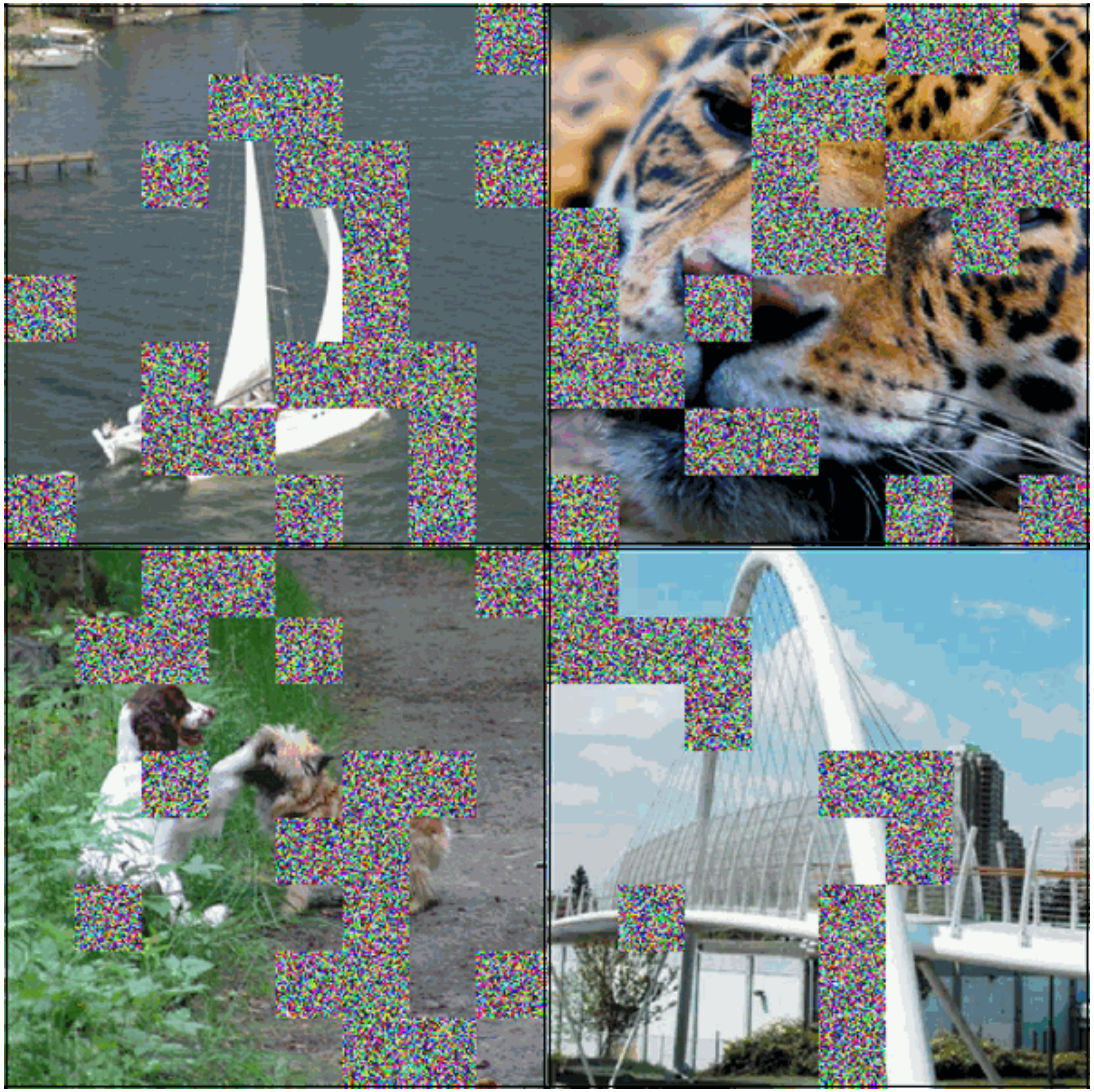



# Un-masking

*Albergo et al. arXiv:2310.03725 (2023)*

*target = full image*
*base = maskeds image*

*velocity conditioned on mask position*

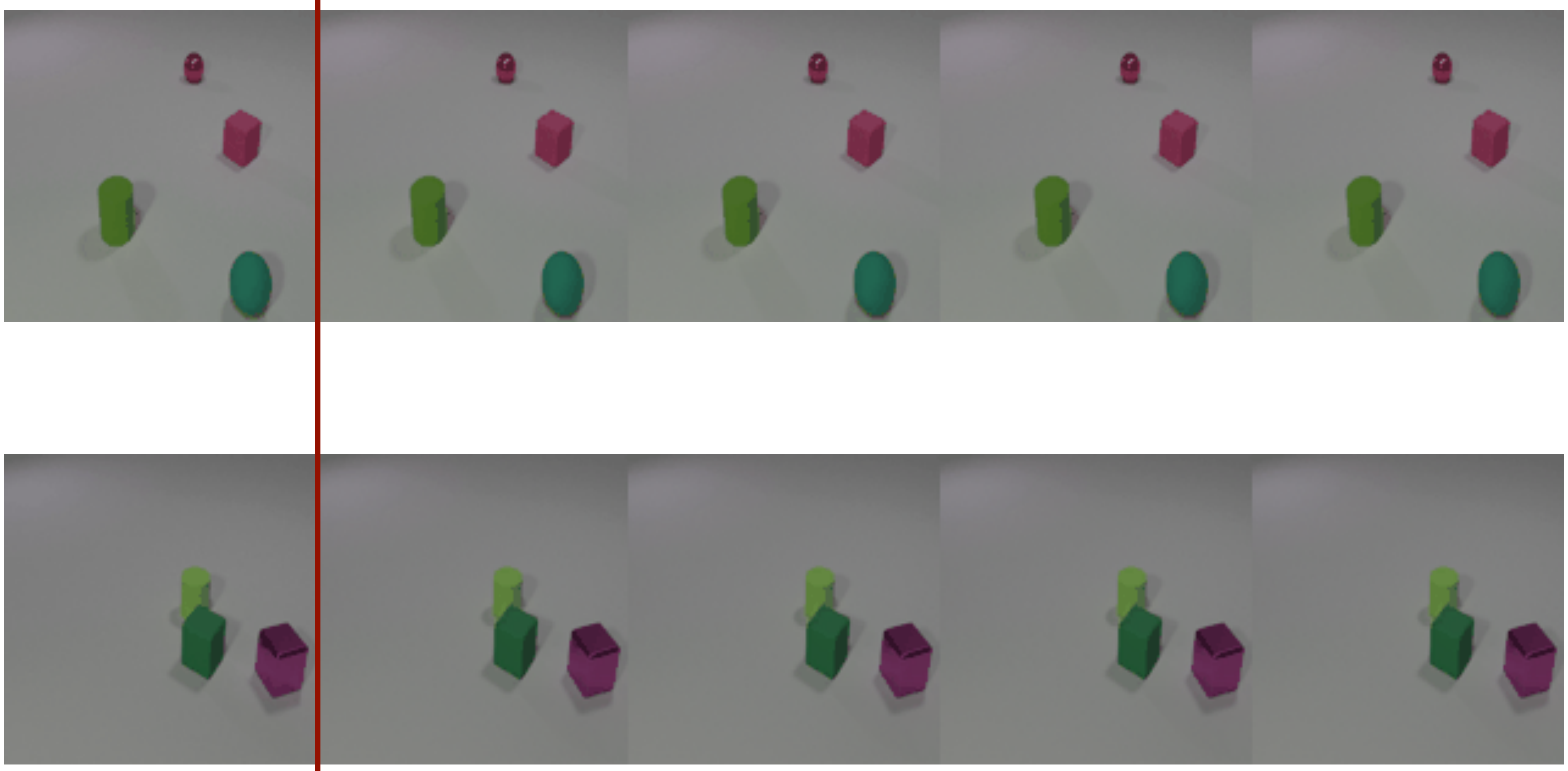


# Probabilistic video generation by roll-out

*Chen, Goldstein, Albergo, Boffi, Albergo & V.-E. arXiv:2403.13724 (2024)*

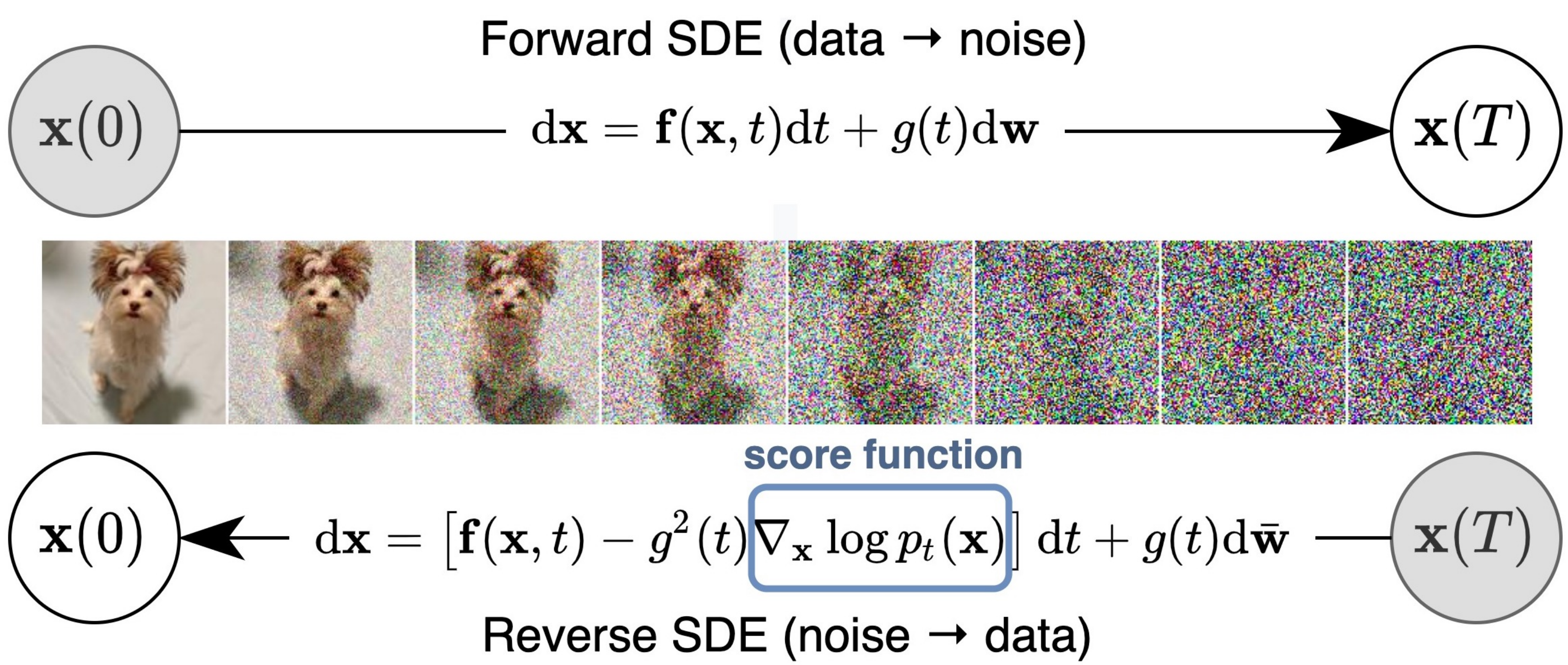
***Video forecasting (frame-by-frame, with rollout)***

original generated



*Set up of: Davtyan, Proc. IEEE/CVF (2023).*

# Link with Score-Based Diffusion Models



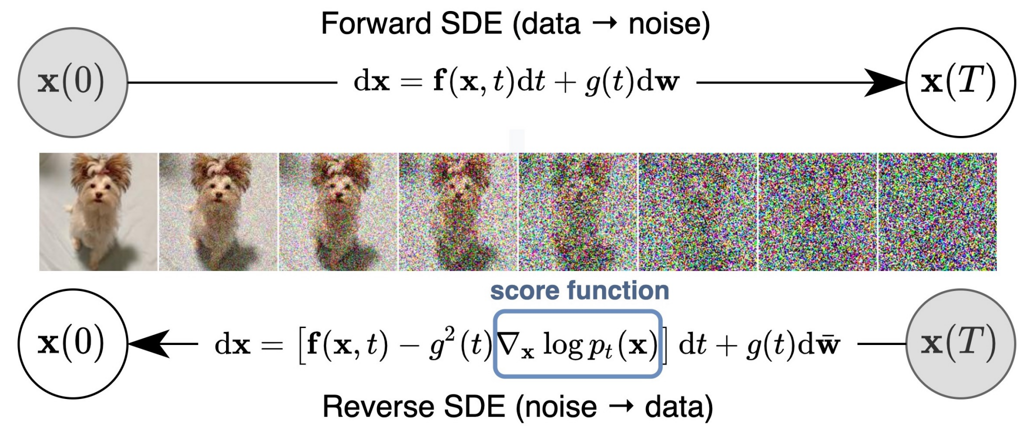

*From Song's paper*

OU process: $dX_\tau = -X_\tau d\tau + dW_\tau$ $\qquad X_{\tau=0} \sim \mu_1$= data distribution

$$X_\tau = x_1 e^{-\tau} + \int_0^\tau e^{-\tau+\tau'} dW_{\tau'} \overset{d}{=} x_1 e^{-\tau} + \sqrt{1 - e^{-2\tau}}\, z$$

# Score-Based Diffusion Models vs Interpolants

$$X_\tau = x_1 e^{-\tau} + \int_0^\tau e^{-\tau+\tau'} dW_{\tau'} \overset{d}{=} x_1 e^{-\tau} + \sqrt{1 - e^{-2\tau}}\, z$$




Take: $t = e^{-\tau} \quad \Rightarrow \quad X_{-\log t} \overset{d}{=} I_t = z\sqrt{1 - t^2} + x_1 t$

***Time rescaling***

***Stochastic interpolant with*** $\alpha_t = \sqrt{1 - t^2},\ \beta_t = t$

***Stochastic interpolants*** *disentangle two operations:*

*1. building a time-dependent* $\mu_t$ *connecting the base and target distributions, and;*
*2. sampling this* $\mu_t$ *using an ODE or an SDE with adjusted diffusion coefficient.*

## *Leveraging the flexibility of the formalism*

# Multitask Learning with Operator Interpolants

*Négrel, Coeurdoux, Albergo & V.-E. arXiv:2508.04605 (2025)*

**Two observations:**

- *The coefficients $\alpha_t, \beta_t$ in $I_t = \alpha_t x_0 + \beta_t x_1$ do not need to be scalar functions of time*
- *There is no reason to choose them beforehand (i.e. before training).*

The *operator interpolant* $I(\alpha, \beta)$ is the process:

$$I(\alpha, \beta) = \alpha x_0 + \beta x_1 \ , \quad x_0 \sim \mu_0 \ , \quad x_1 \sim \mu_1$$

where $x_0, x_1 \in \mathscr{H}$ (e.g. $\mathbb{R}^d$) and

$\alpha, \beta$ *are* ***linear operators*** *in that space*.

The *multitask drifts* $\eta_{0,1}(\alpha, \beta, x)$ *are the vector fields*:

$$\eta_0(\alpha, \beta, x) = \mathbb{E}[x_0 \,|\, I(\alpha, \beta) = x], \qquad \eta_1(\alpha, \beta, x) = \mathbb{E}[x_1 \,|\, I(\alpha, \beta) = x]$$

# Multitask Learning with Operator Interpolant

Négrel, Coeurdoux, Albergo & V.-E. arXiv:2508.04605 (2025)

The *operator interpolant* $I(\alpha, \beta)$ is the process:

$$I(\alpha, \beta) = \alpha x_0 + \beta x_1 \,, \quad x_0 \sim \mu_0 \,, \quad x_1 \sim \mu_1$$

where $x_0, x_1 \in \mathscr{H}$ (e.g. $\mathbb{R}^d$) and

$\alpha, \beta$ *are* ***linear operators*** *in that space*.

The ***multitask drifts*** $\eta_0(\alpha, \beta, x)$ ***and*** $\eta_1(\alpha, \beta, x)$ ***are the vector fields***:

$$\eta_0(\alpha, \beta, x) = \mathbb{E}[x_0 \,|\, I(\alpha, \beta) = x], \qquad \eta_1(\alpha, \beta, x) = \mathbb{E}[x_1 \,|\, I(\alpha, \beta) = x]$$

***Estimation of $\eta_0, \eta_1$ = simulation-free regression problem***

$$\eta_0 = \underset{\hat{\eta}_0}{argmin}\, \mathbb{E}_{(\alpha,\beta)\sim\nu}\mathbb{E}_{(x_0,x_1)\sim\mu}\left[\,|\,\hat{\eta}_0(\alpha, \beta, I(\alpha, \beta)) - x_0\,|^2\right]$$

***e.g.*** $\beta = 1 - \alpha$ ***diagonal,*** $I(\alpha) = \alpha \odot x_0 + (1-\alpha) \odot x_1$ ***, and*** $\nu = U([0,1]^d)$

# Multitask Learning with Operator Interpolant

Négrel, Coeurdoux, Albergo & V.-E. arXiv:2508.04605 (2025)

**Thm:** Given any path $(\alpha_t, \beta_t)_{t\in[0,1]}$ the law of $I(\alpha_t, \beta_t)$ is the same as the law of the solution to

$$\dot{X}_t = \dot{\alpha}_t \eta_0(\alpha_t, \beta_t, X_t) + \dot{\beta}_t \eta_1(\alpha_t, \beta_t, X_t), \qquad X_0 \stackrel{d}{=} I(\alpha_0, \beta_0)$$

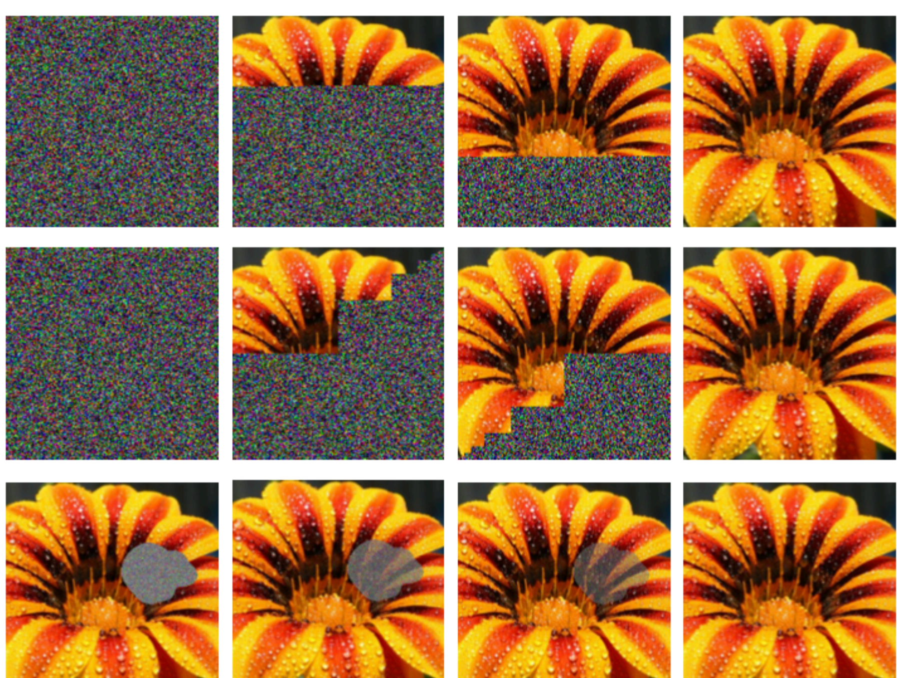
***If we can sample $I(\alpha_0, \beta_0)$ we can generate sample along any ray emanating from it.***

$I(\alpha) = \alpha \odot x_0 + (1-\alpha) \odot x_1\,; \quad \nu = U([0,1]^d)$

AR

Morton

In-paint



***Other choices of operators $\alpha, \beta$***

Gaussian

Motion

# Multitask Learning with Operator Interpolant

*Négrel, Coeurdoux, Albergo & V.-E. arXiv:2508.04605 (2025)*

**Thm:** Given any path $(\alpha_t, \beta_t)_{t\in[0,1]}$ the law of $I(\alpha_t, \beta_t)$ is the same as the law of the solution to

$$\dot{X}_t = \dot{\alpha}_t \eta_0(\alpha_t, \beta_t, X_t) + \dot{\beta}_t \eta_1(\alpha_t, \beta_t, X_t), \qquad X_0 \overset{d}{=} I(\alpha_0, \beta_0)$$

***If we can sample $I(\alpha_0, \beta_0)$ we can generate sample along any ray emanating from it.***

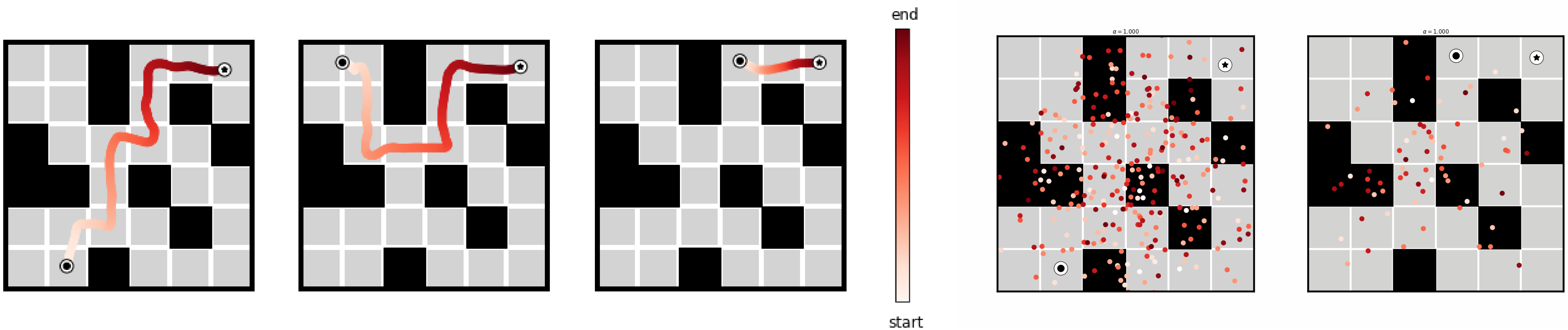
***Approach can be seen as a way of amortizing learning over a variety of tasks: it enables generation strategies to be defined, optimized, or modified dynamically at inference time without retraining.***

***If $x_0 \sim N(0, Id)$ & $x_0 \perp x_1$ we can use the SDE***

$$dX_t = (\dot{\alpha}_t - \epsilon_t \alpha_t^{-1})\eta_0(\alpha_t, \beta_t, X_t)dt + \dot{\beta}_t \eta_1(\alpha_t, \beta_t, X_t)dt + \sqrt{2\epsilon_t}\, dW_t, \qquad X_0 \overset{d}{=} I(\alpha_0, \beta_0)$$

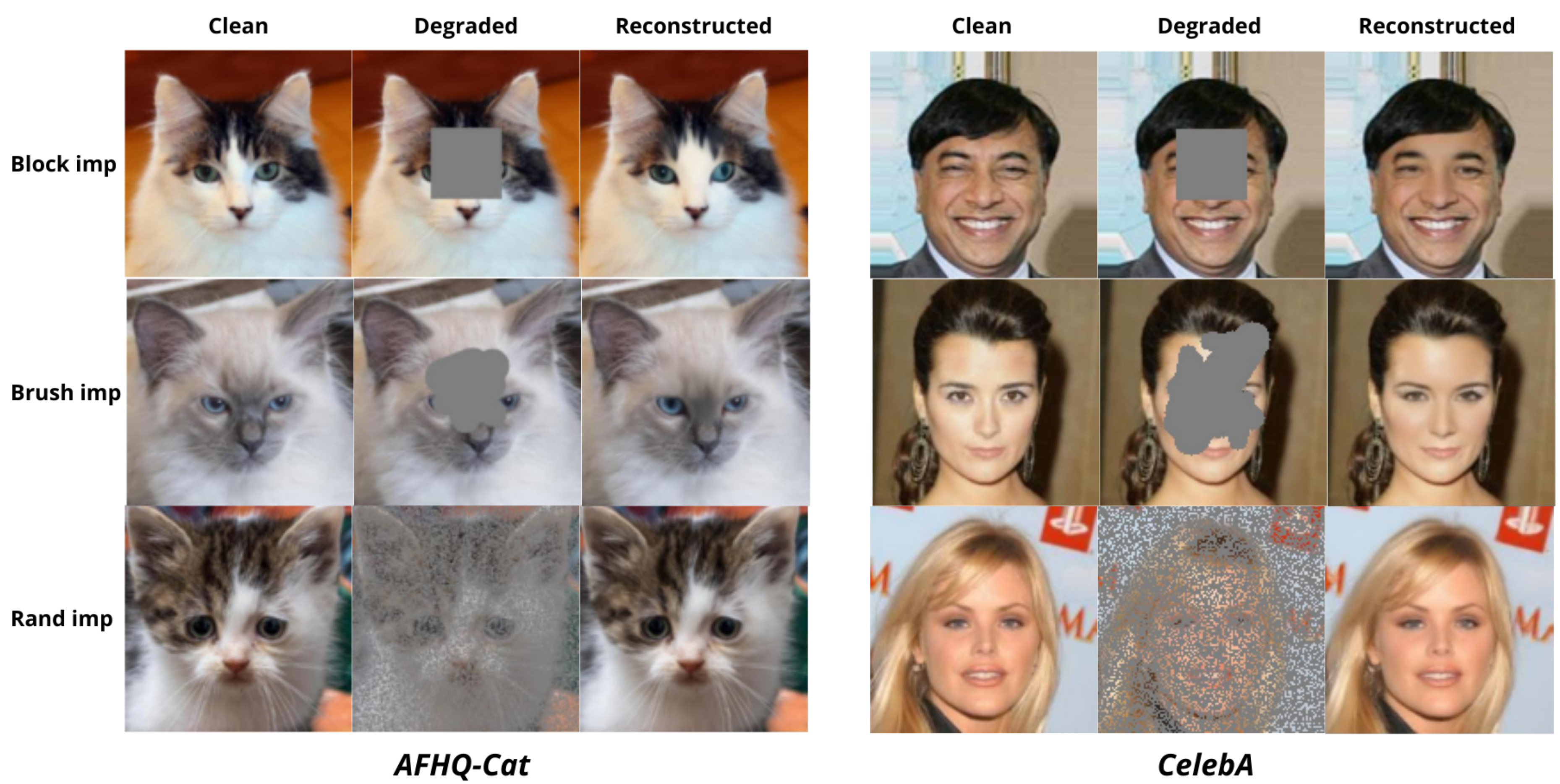
# Multitask Learning with Operator Interpolant

*Négrel, Coeurdoux, Albergo & V.-E.  arXiv:2508.04605 (2025)*




*Robotics application*

*Inpainting, editing, denoising, etc.*



# Posterior Sampling and Fine-Tuning

*Négrel, Coeurdoux, Albergo & V.-E. arXiv:2508.04605 (2025)*

**Objective:** Sample the posterior distribution

$$\mu_1^r(dx) = Z^{-1} e^{r(x)} \mu_1(dx)$$

where:

- $\mu_1(dx)$ is the prior distribution (sampleable)
- $r(x) = \frac{1}{2}\langle x, Ax\rangle + \langle b, x\rangle$ is the likelihood (aka reward) function (given)
- $Z = \int e^{r(x)} \mu_1(dx) < \infty$ is the evidence / partition function (unknown)

**Thm:** Let $I(\alpha, \beta) = \alpha x_0 + \beta x_1$ and $I_r(\alpha, \beta) = \alpha x_0 + \beta x_1^r$, where $x_0 \sim \mu_0, x_1 \sim \mu_1$ , $x_1^r \sim \mu_1^r$

and

$$\eta_0(\alpha, \beta, x) = \mathbb{E}[x_0 \,|\, I(\alpha, \beta) = x], \qquad \eta_1(\alpha, \beta, x) = \mathbb{E}[x_1 \,|\, I(\alpha, \beta) = x] \qquad \textit{prior}$$

$$\eta_0^r(\alpha, \beta, x) = \mathbb{E}[x_0 \,|\, I_r(\alpha, \beta) = x], \qquad \eta_1^r(\alpha, \beta, x) = \mathbb{E}[x_1^r \,|\, I_r(\alpha, \beta) = x] \qquad \textit{posterior}$$

Then

$$\eta_0^r(\alpha, \beta, x) = \alpha^{-1} \beta \beta_r^{-1} \alpha_r \eta_0(\alpha_r, \beta_r, x_r) + \alpha^{-1}(x - \beta \beta_r^{-1} x_r)$$

$$\eta_1^r(\alpha, \beta, x) = \eta_1(\alpha_r, \beta_r, x_r)$$

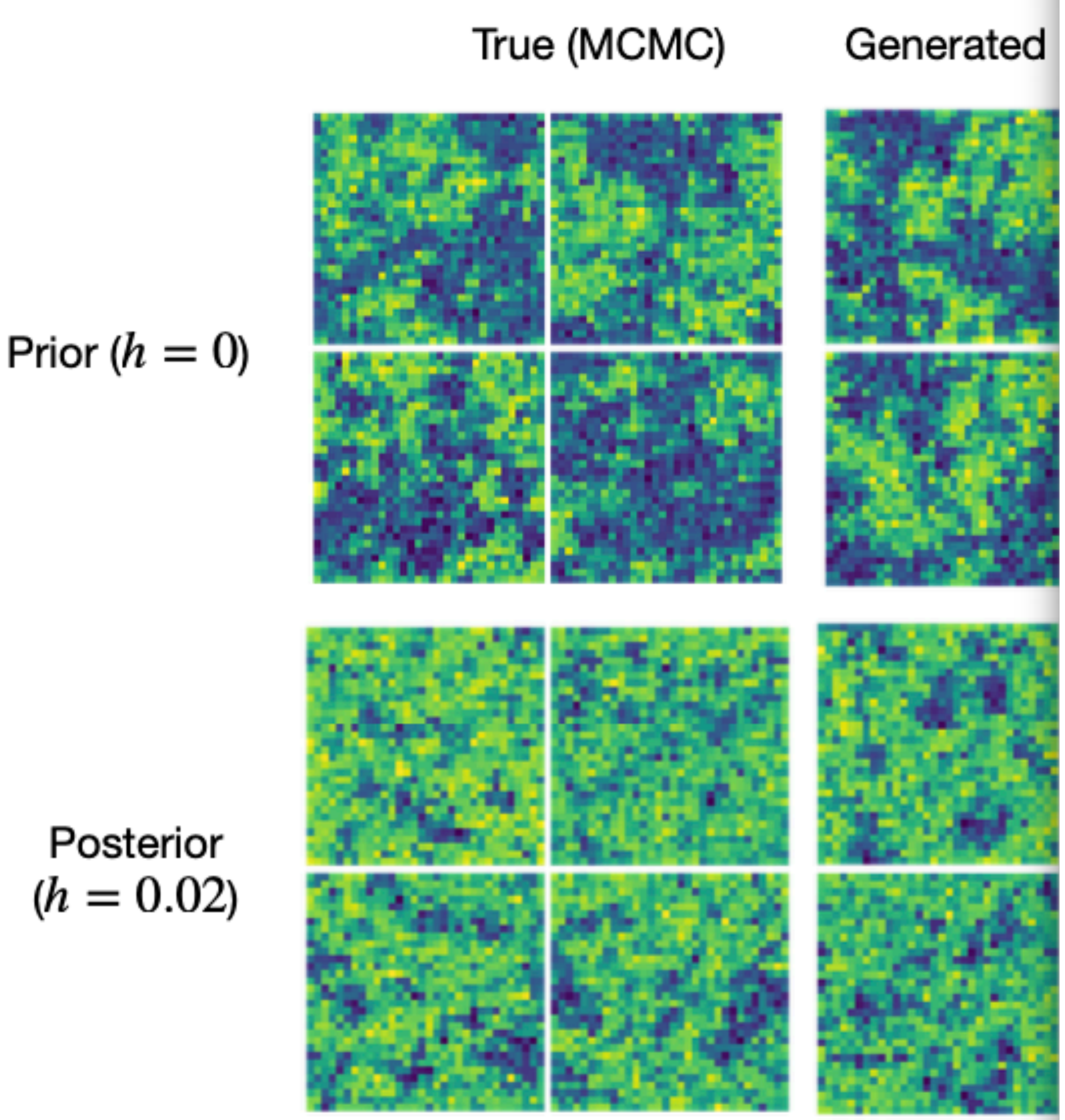
if $\quad \beta_r^T \alpha_r^{-T} \alpha_r^{-1} \beta_r = \beta^T \alpha^{-T} \alpha^{-1} \beta - A, \qquad x_r = \alpha_r \alpha_r^T \beta_r^{-T} \left(\beta^T \alpha^{-T} \alpha^{-1} x + b\right).$

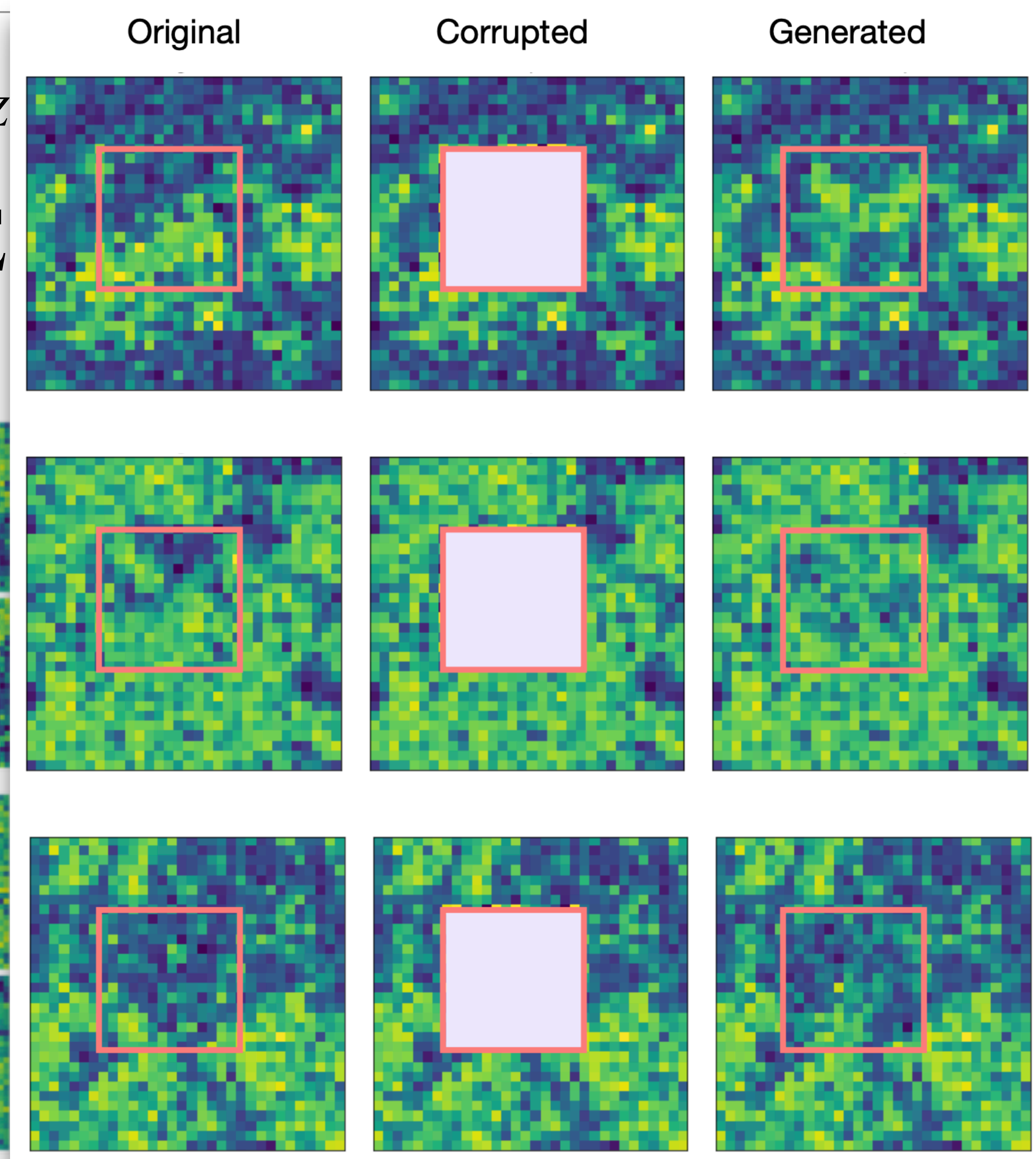
# Posterior Sampling and Fine-Tuning

*Négrel, Coeurdoux, Albergo & V.-E.  arXiv:2508.04605 (2025)*

***Lattice $\phi^4$ model:*** $\mu(d\varphi) = Z$

$$E(\varphi) = \frac{1}{2}\chi \sum_{a\sim b} |\varphi(a) - \varphi(b)|^2 + \frac{1}{2}\kappa \sum_{a}$$



# Likelihood control

***Why use the SDE rather than the ODE?***

***The drift $b_t(x)$ and the score $s_t(x)$ are only known approximately!***

*The ODE only offers control of the Wasserstein distance from the target,*
*whereas the SDE allows for control of the Kullback-Leibler divergence from the target.*

**Thm:** Let $\hat{\mu}_t(x)$ be the solution to the FPE

$$\partial_t \hat{\mu}_t + \nabla \cdot ((\hat{b}_t + \epsilon \hat{s}_t)\hat{\mu}_t) = \epsilon \Delta \hat{\mu}_t, \qquad \hat{\mu}_{t=0} = \mu_0,$$

Then:

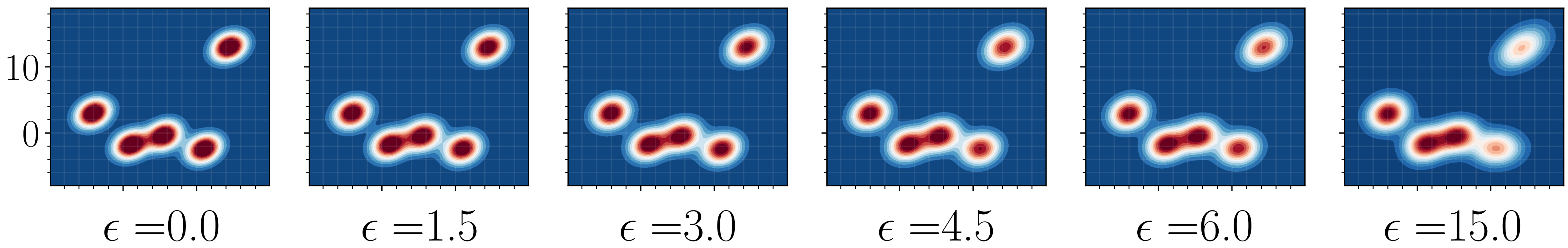
$$\mathrm{KL}(\mu_1 \| \hat{\mu}_1) \leq \frac{1}{2\epsilon}\big(L_b(\hat{b}) - \min_{\hat{b}} L_b(\hat{b})\big) + \frac{\epsilon}{2}\big(L_s(\hat{s}) - \min_{\hat{s}} L_s(\hat{s})\big)$$

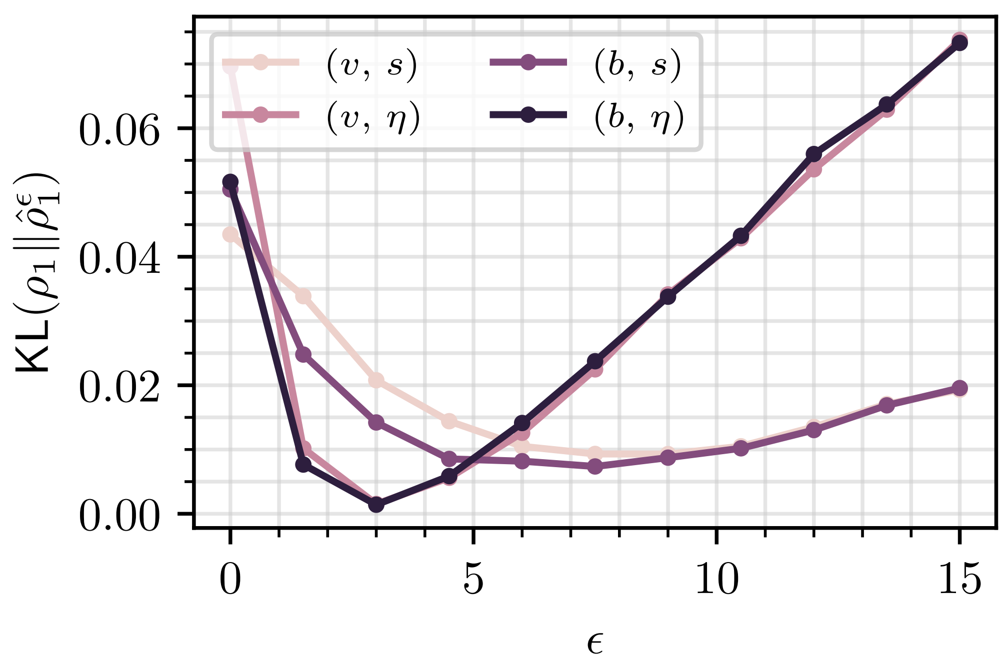
*Proof = consequence of Girsanov theorem*

*In the context of SBDM:*
*Chen et al. arXiv:2209.11215 (2022);*
*Lee et al. arXiv:2206.06227 (2022).*

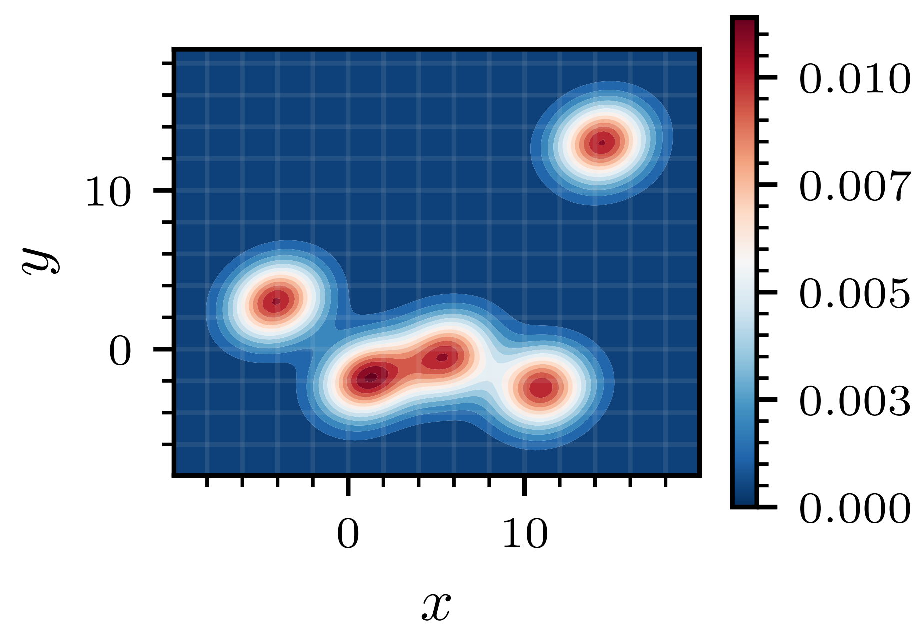
*Empirical verification:*

*Gaussian mixture density in d=128*




*Target*




$$\mathsf{KL}(\rho_1 \| \hat{\rho}_1) \leq \frac{1}{2\epsilon}\big(L_b(\hat{b}) - \min_{\hat{b}} L_b(\hat{b})\big) + \frac{\epsilon}{2}\big(L_s(\hat{s}) - \min_{\hat{s}} L_s(\hat{s})\big)$$

***Diffusion coefficient can be adjusted post-training!***

# Adjusting the Diffusion Coefficient

***Base = point mass***

**Thm:** for all $t \in [0,1]$ and any $\epsilon_t$ with $\epsilon_0 = \sigma_0 > 0$:
the law of $I_t = \beta_t x_1 + \sigma_t W_t$ is the same as the law of the solution to

$$dX_t = b_t(X_t)dt + \tfrac{1}{2}(\epsilon_t^2 - \sigma_t^2)s_t(X_t)dt + \epsilon_t dW_t, \qquad X_0 = 0$$

where the drift $b_t(x)$ and the score $s_t(x)$ are given explicitly by

$$b_t(x) = \mathbb{E}\left[\dot\beta_t x_1 + \dot\sigma_t W_t \,\middle|\, I_t = x\right] \qquad\qquad s_t(x) = \frac{\beta_t b_t(x) - \dot\beta_t x}{t\sigma_t(\dot\beta_t \sigma_t - \beta_t \dot\sigma_t)}$$

*Chen, Goldstein, Albergo, Boffi, Albergo & V.-E. arXiv:2403.13724 (2024)*

# Minimizing the impact of the estimation error

**Thm:** Let

$$dX_t = b_t(X_t)dt + \frac{1}{2}(\epsilon_t^2 - \sigma_t^2)s_t(X_t)dt + \epsilon_t dW_t$$ = ***exact process***

$$d\hat{X}_t = \hat{b}_t(\hat{X}_t)dt + \frac{1}{2}(\epsilon_t^2 - \sigma_t^2)\hat{s}_t(\hat{X}_t)dt + \epsilon_t dW_t$$ = ***estimated process***

Then the ***path KL divergence*** $D_{\mathrm{KL}}(P_{\hat{X}} \| P_X)$ is ***minimized*** when $\epsilon_t = \epsilon_t^F$ with

$$\epsilon_t^F = \left| 2t\sigma_t(\beta_t^{-1}\dot{\beta}_t\sigma_t - \dot{\sigma}_t) - \sigma_t^2 \right|^{1/2}$$

***Explicit expression for $\epsilon_t$ independent of the drift and the target distribution!***

# Connection with Föllmer processes

**Thm:** Let

$$dX_t = b_t(X_t)dt + \tfrac{1}{2}(\epsilon_t^2 - \sigma_t^2)s_t(X_t)dt + \epsilon_t dW_t$$ ***= exact process***

$$d\hat{X}_t = \hat{b}_t(\hat{X}_t)dt + \tfrac{1}{2}(\epsilon_t^2 - \sigma_t^2)\hat{s}_t(\hat{X}_t)dt + \epsilon_t dW_t$$ ***= estimated process***

Then the ***path KL divergence*** $D_{\mathrm{KL}}(P_{\hat{X}} \| P_X)$ is ***minimized*** when $\epsilon_t = \epsilon_t^F$ with

$$\epsilon_t^F = \left| 2t\sigma_t(\beta_t^{-1}\dot{\beta}_t\sigma_t - \dot{\sigma}_t) - \sigma_t^2 \right|^{1/2}$$

**Thm:** The process $X^F = (X_t^F)_{t\in[0,1]}$ obtained with $\epsilon_t = \epsilon_t^F$ is a

***Föllmer process = Schrödinger bridge between $\delta_0$ and $\mu_1$***

***Different interpretation/derivation of the Föllmer process, through minimization of the impact of estimation error.***

## *Some Scientific Applications*

# Probabilistic forecasting and superresolution

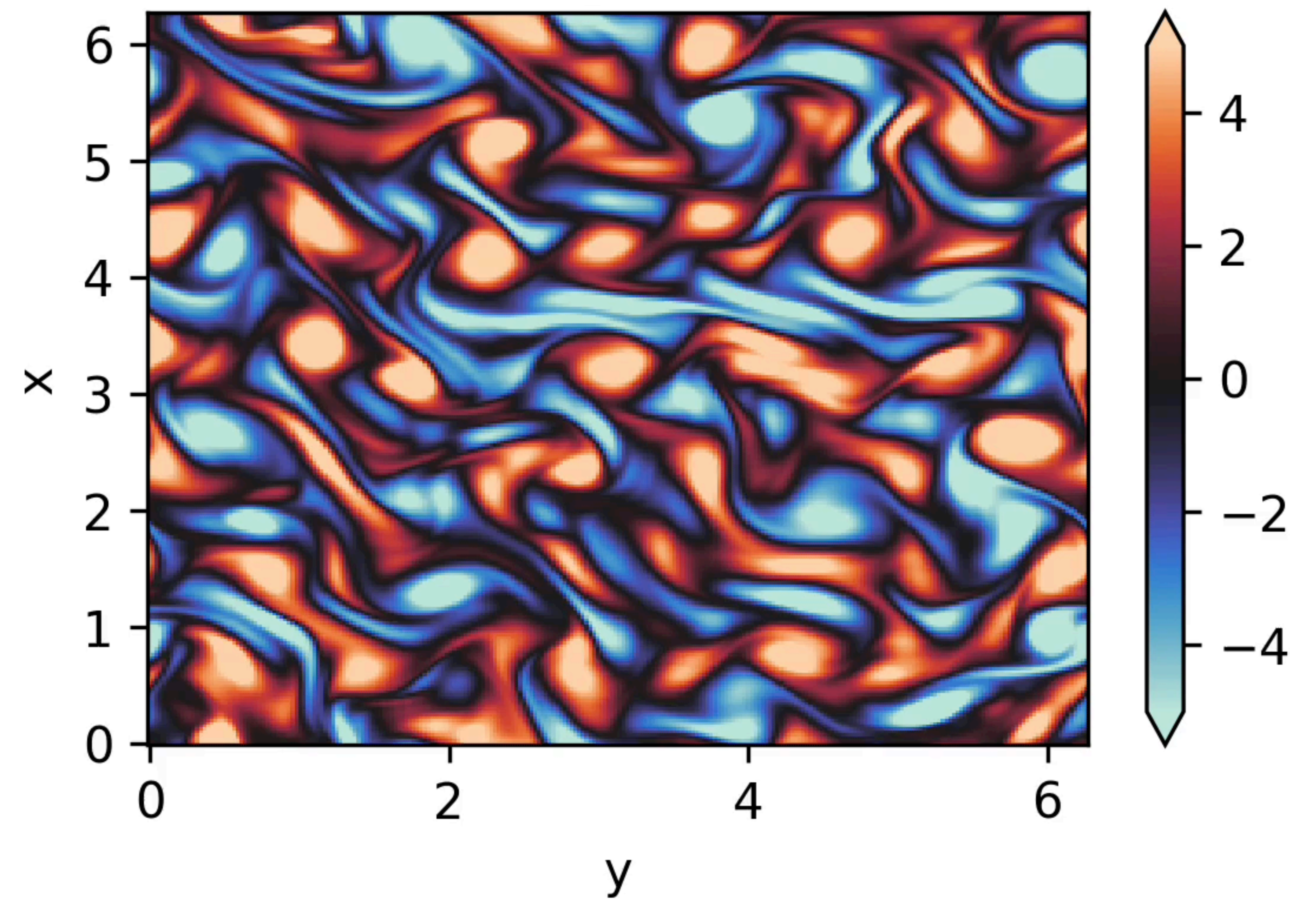
*Chen, Goldstein, Albergo, Boffi, Albergo & V.-E.  arXiv:2403.13724 (2024)*

***2D Navier-Stokes equation  with random forcing on the torus***

$$d\omega + v \cdot \nabla \omega dt = \nu \Delta \omega dt - \alpha \omega dt + \varepsilon d\eta$$

$$v = \nabla^{\perp}\psi = (-\partial_y \psi, \partial_x \psi), \qquad -\Delta\psi = \omega$$

$d\eta$ = white-in-time forcing acting on a few Fourier modes



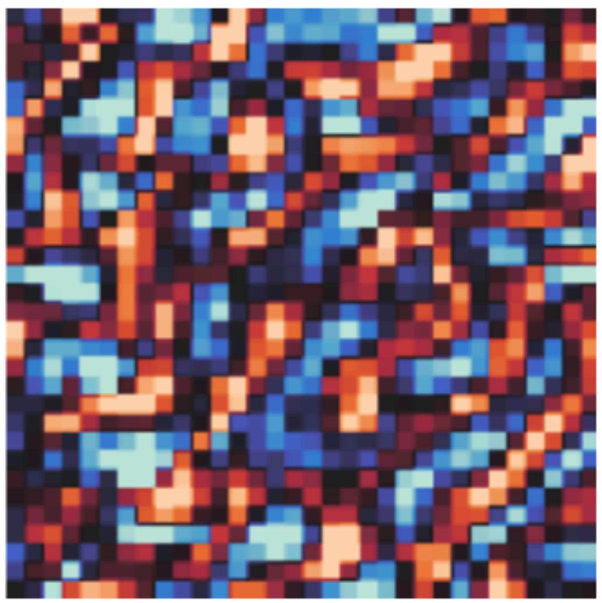

*Set-up of Hairer & Mattingly (2006) for which NSE is provable ergodic with a unique IM.*

***Aim:*** *Given $\omega_t$ in full- or low-resolution, forecast the ensemble of $\omega_{t+\tau}$ with $\tau > 0$ at full-resolution.*

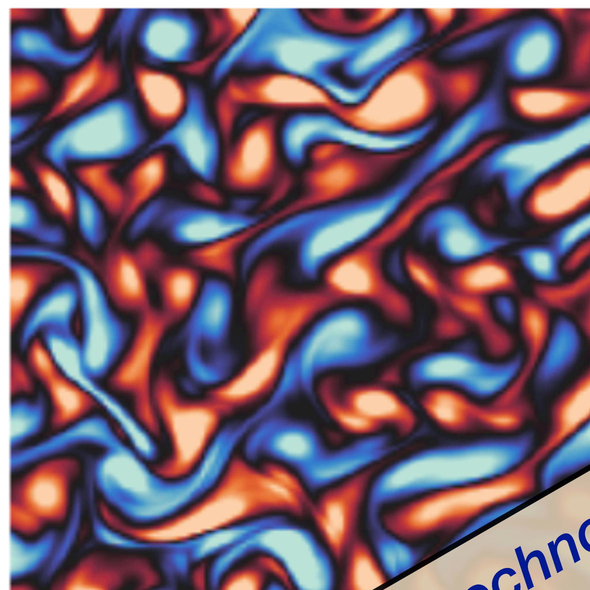
# Probabilistic forecasting/downscaling

*Chen, Goldstein, Albergo, Boffi, Albergo & V.-E. arXiv:2403.13724 (2024)*

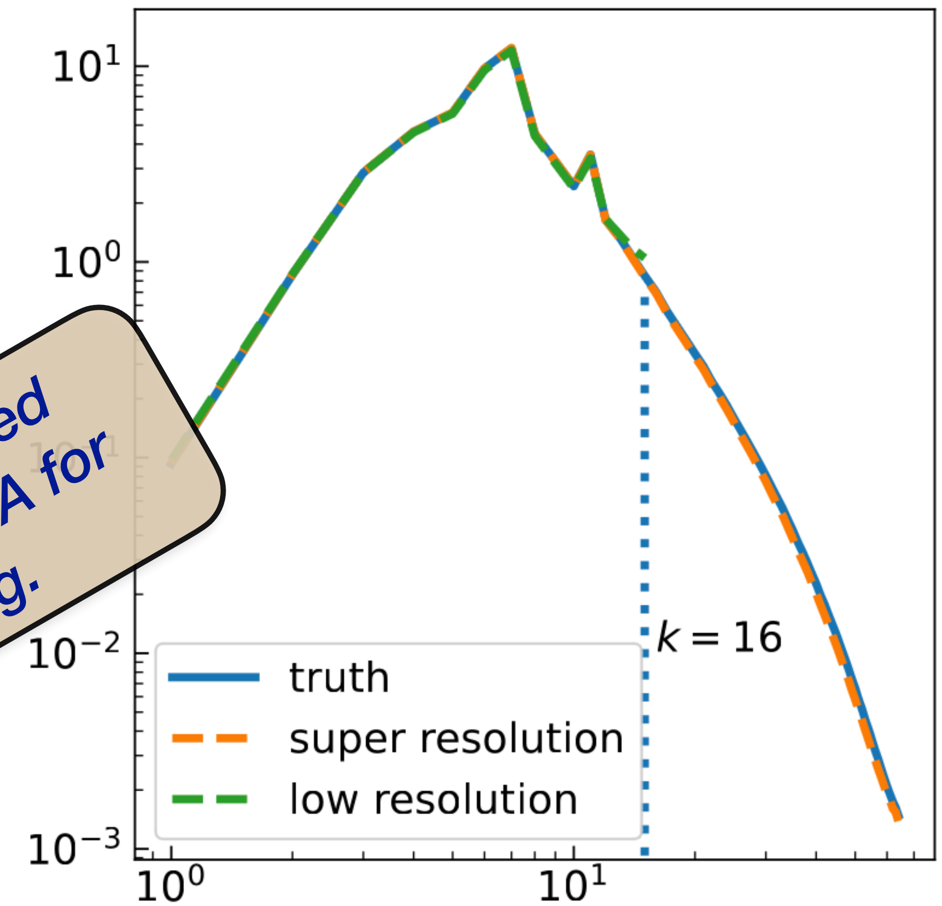
***Forecasting and superresolution in 2D Navier-Stokes equation with random forcing on the torus***



***Low res data at t***

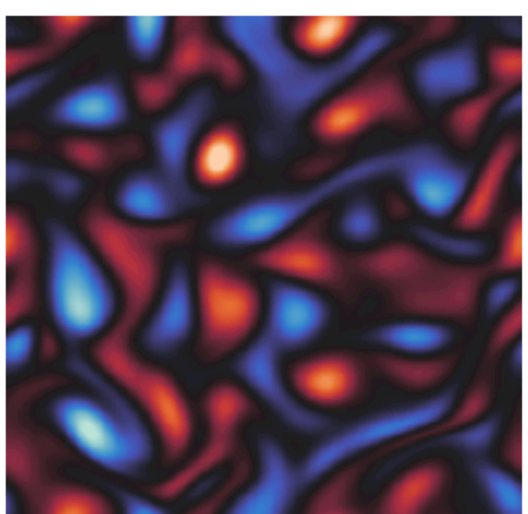


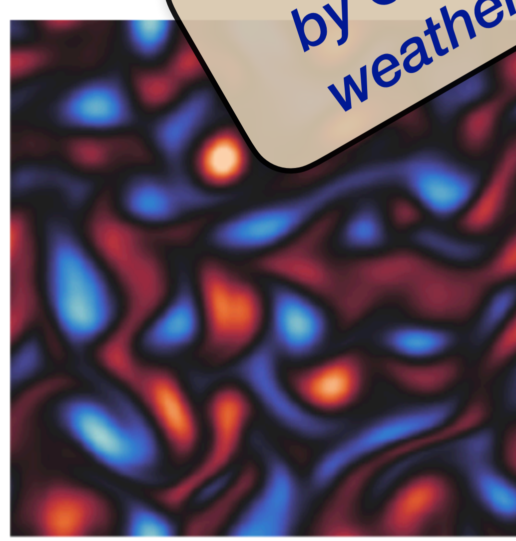
***High res forecast at $t+\tau$***




enstrophy spectrum (v.s. $k$)

Similar technology used by Google and NVDIA for weather forecasting.





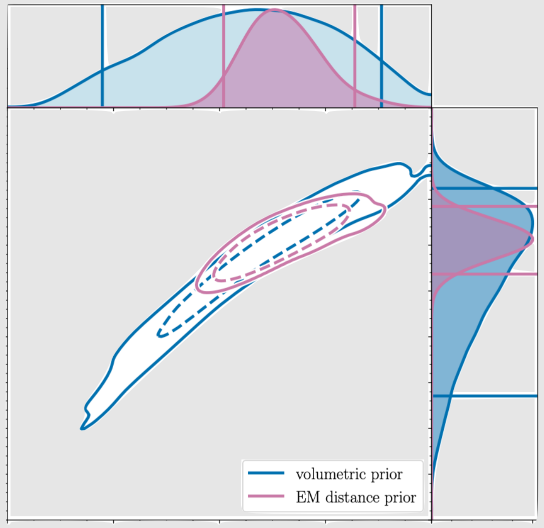
***True vs forecasted conditional mean***

***Probabilistic aspect is key!***

***Roll-out possible.***

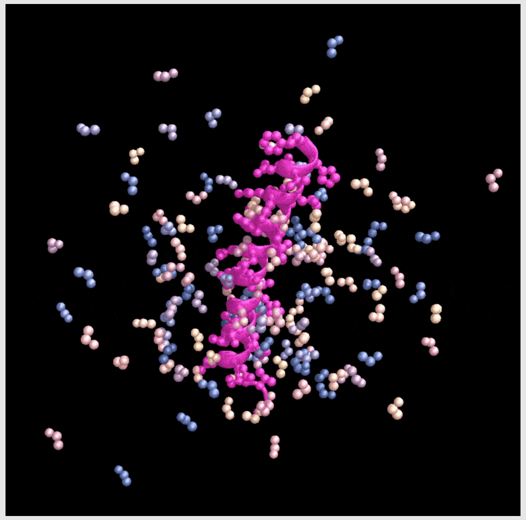
# Assisting Monte-Carlo Sampling

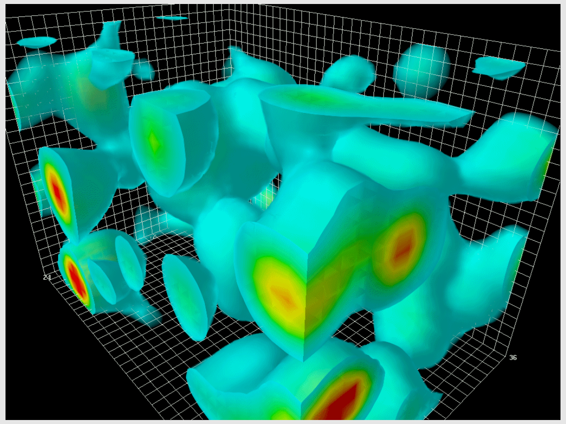
- *Generic method in **Statistical Mechanics, Bayesian Inference, Uncertainty Quantification,** etc.*
- *Aim at **sampling a target distribution** known up to a normalized constant.*
- *Plagued by **slow convergence** - hard to propose good new samples.*




**Bayesian inference**

**molecular conformations**





**quantum field theory**

# Assisting Monte-Carlo Sampling

- *Generic method in **Statistical Mechanics, Bayesian Inference, Uncertainty Quantification,** etc.*
- *Aim at **sampling a target distribution** known up to a normalized constant.*
- *Plagued by **slow convergence** - hard to propose good new samples.*

***Idea: learn generative models to get better samples.***


Rezende *et al.*, arXiv:1505.05770; ….
Noé *et al.*, Science 365 eaaw1147 (2019);
Albergo, Kanwar, Shanahan, Phys. Rev. D 100, 034515 (2019);
Gabrié, Rotskoff & V.-E. PNAS 119, e2109420119 (2021);
Albergo & V.-E. arXiv:2410.02711 (2024)


***Different set-up than standard ML:***

***Problems with model but no data initially (as opposed to data but no model)***

***Allow for infinite data generation with validation — verifiable AI !***

The Unreasonable Effectiveness of Mathematics in the Natural Sciences

Richard Courant Lecture in Mathematical Sciences delivered at New York University, May 11, 1959

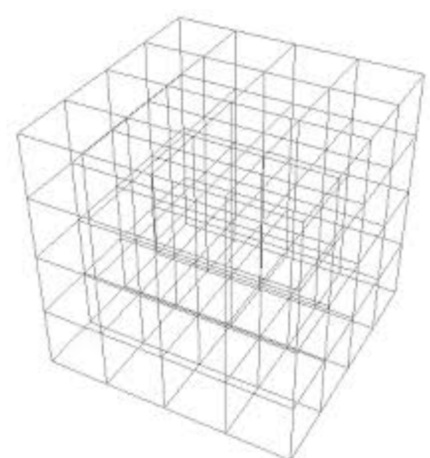
EUGENE P. WIGNER

# The Unreasonable Effectiveness of Machine Learning

**Curses of Dimensionality (CoD):**

The number of operations/parameters needed to optimize/integrate/approximate Lipschitz functions to precision $\delta$ depends exponentially on the input dimension $d$, $O(\delta^{-d})$.

[Bellman, 61]



***Gridding does not scale:***

*2 points in* $d = 1$;

$2^2 = 4$ *points in* $d = 2$;

$\ldots$

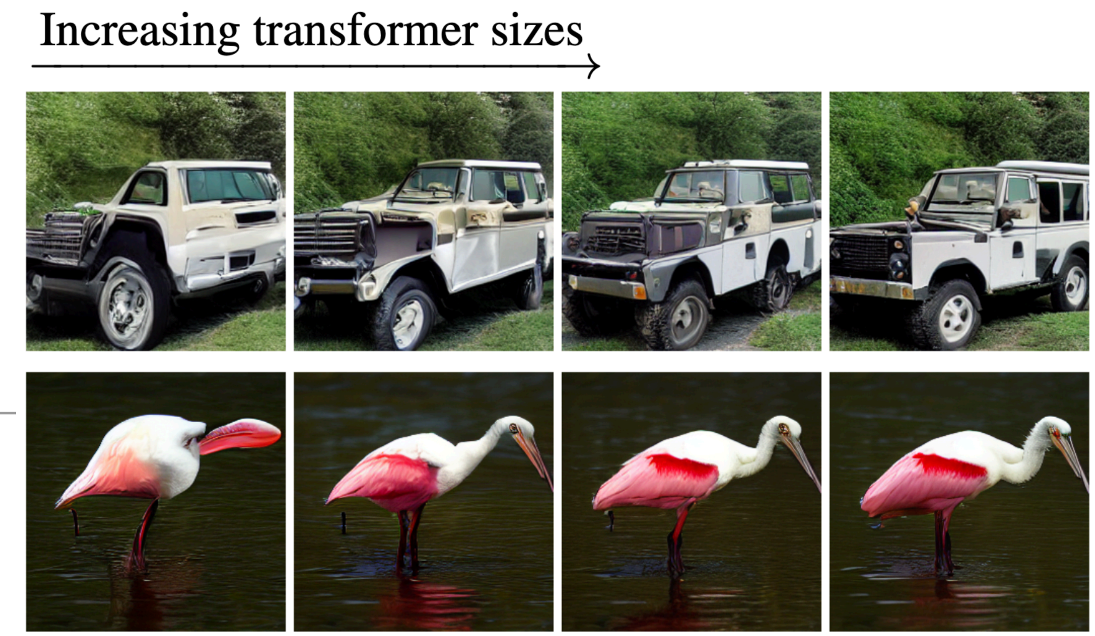
$2^{1000} = 10^{300}$ *points in* $d = 1000$

***How come we can learn to generate images or texts which a priori live in very high dimensional spaces?***

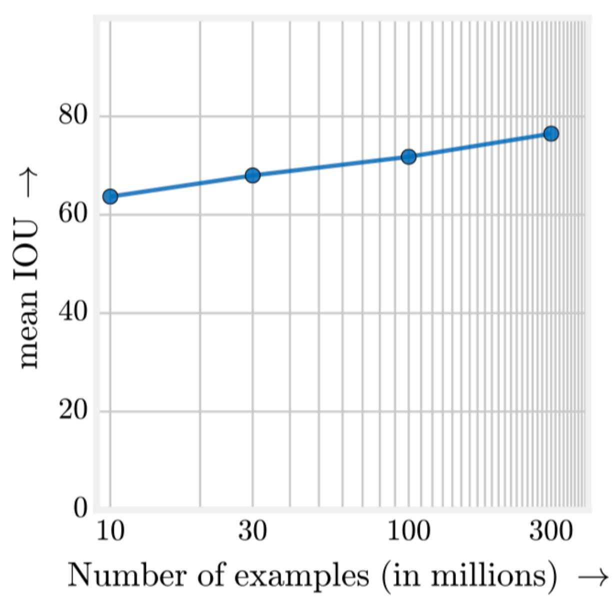
*When, how, and why can neural networks approximate high dimensional functions?*

# Need for Theory

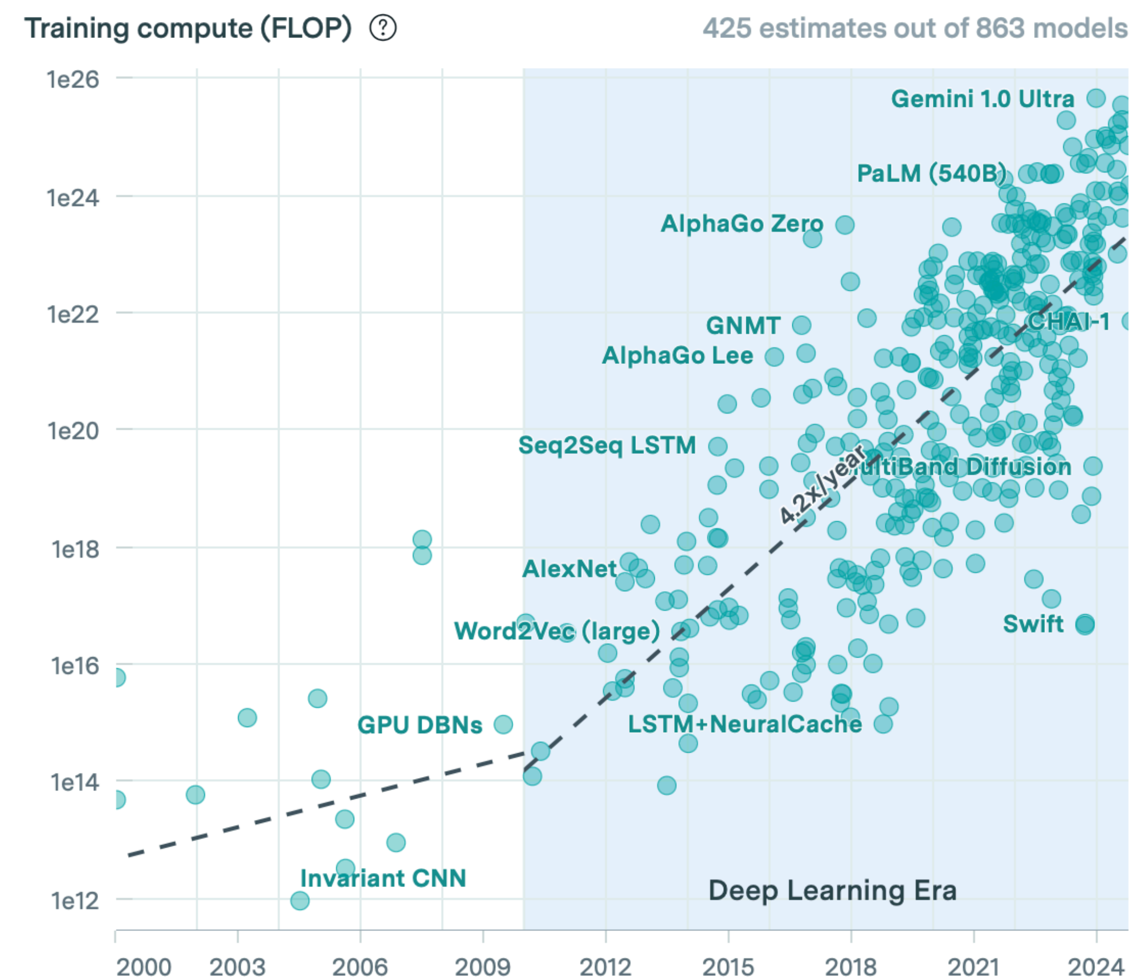
DL is very costly in terms of compute and data.
Brute-force approach is not sustainable.




| Initialization | mIOU |
|---|---|
| ImageNet | 73.6 |
| 300M | 75.3 |
| ImageNet+300M | **76.5** |




[Sun et al ICCV 2017]




[Sevilla & Roldán, epoch.ai blog 2025]

***Performance increases logarithmically with data volume***

***AI quadruples its compute every year***


C'EST TOUT
POUR AUJOUR
D'HUI

Computer Science > Machine Learning

*[Submitted on 30 Sep 2022 (v1), last revised 20 Oct 2022 (this version, v2)]*

# Building Normalizing Flows with Stochastic Interpolants

Michael S. Albergo, Eric Vanden-Eijnden

A simple generative model based on a continuous-time normalizing flow between any pair of base and target probability densities is proposed. The velocity field of this flow is inferred from the probability current of a time-dependent density that interpolates between the base and the target in finite time. Unlike conventional normalizing flow inference methods based the maximum likelihood principle, which require costly backpropagation through ODE solvers, our interpolant approach leads to a simple quadratic loss for the velocity itself which is expressed in terms of expectations that are readily amenable to empirical estimation. The flow can be used to generate samples from either the base or target, and to estimate the likelihood at any time along the interpolant. In addition, the flow can be optimized to minimize the path length of the interpolant density, thereby paving the way for building optimal transport maps. The approach is also contextualized in its relation to diffusions. In particular, in situations where the base is a Gaussian density, we show that the velocity of our normalizing flow can also be used to construct a diffusion model to sample the target as well as estimating its score. This allows one to map methods based on stochastic differential equations to those using ordinary differential equations, simplifying the mechanics of the model, but capturing equivalent dynamics. Benchmarking on density estimation tasks illustrates that the learned flow can match and surpass maximum likelihood continuous flows at a fraction of the conventional ODE training costs.

Subjects: **Machine Learning (cs.LG)**; Machine Learning (stat.ML)
Cite as: arXiv:2209.15571 **[cs.LG]**
(or arXiv:2209.15571v2 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2209.15571

## Submission history

From: Michael Albergo [view email]
[v1] Fri, 30 Sep 2022 16:30:31 UTC (3,383 KB)
**[v2]** Thu, 20 Oct 2022 14:57:06 UTC (4,361 KB)

[Submitted on 15 Mar 2023]

# Stochastic Interpolants: A Unifying Framework for Flows and Diffusions

Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden

We introduce a class of generative models based on the stochastic interpolant framework proposed in Albergo & Vanden-Eijnden (2023) that unifies flow-based and diffusion-based methods. We first show how to construct a broad class of continuous-time stochastic processes whose time-dependent probability density function bridges two arbitrary densities exactly in finite time. These `stochastic interpolants' are built by combining data from the two densities with an additional latent variable, and the specific details of the construction can be leveraged to shape the resulting time-dependent density in a flexible way. We then show that the time-dependent density of the stochastic interpolant satisfies a first-order transport equation as well as a family of forward and backward Fokker-Planck equations with tunable diffusion; upon consideration of the time evolution of an individual sample, this viewpoint immediately leads to both deterministic and stochastic generative models based on probability flow equations or stochastic differential equations with a tunable level of noise. The drift coefficients entering these models are time-dependent velocity fields characterized as the unique minimizers of simple quadratic objective functions, one of which is a new objective for the score of the interpolant density. Remarkably, we show that minimization of these quadratic objectives leads to control of the likelihood for generative models built upon stochastic dynamics; by contrast, we show that generative models based upon a deterministic dynamics must, in addition, control the Fisher divergence between the target and the model. Finally, we construct estimators for the likelihood and the cross-entropy of interpolant-based generative models, and demonstrate that such models recover the Schrödinger bridge between the two target densities when explicitly optimizing over the interpolant.

Subjects: **Machine Learning (cs.LG)**; Disordered Systems and Neural Networks (cond-mat.dis-nn); Probability (math.PR)

Cite as: arXiv:2303.08797 **[cs.LG]**
(or arXiv:2303.08797v1 **[cs.LG]** for this version)

## Submission history

From: Michael Albergo [view email]
**[v1]** Wed, 15 Mar 2023 17:43:42 UTC (5,381 KB)

## Download:

- PDF
- Other formats

(license)

Current browse context:
**cs.LG**
< prev | next >
new | recent | 2303

Change to browse by:
cond-mat
cond-mat.dis-nn
cs
math
math.PR

References & Citations
- NASA ADS
- Google Scholar
- Semantic Scholar

**Export Bibtex Citation**

Bookmark

reenshot

[Submitted on 5 Oct 2023]

# Stochastic interpolants with data-dependent couplings

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, Eric Vanden-Eijnden

Generative models inspired by dynamical transport of measure -- such as flows and diffusions -- construct a continuous-time map between two probability densities. Conventionally, one of these is the target density, only accessible through samples, while the other is taken as a simple base density that is data-agnostic. In this work, using the framework of stochastic interpolants, we formalize how to \textit{couple} the base and the target densities. This enables us to incorporate information about class labels or continuous embeddings to construct dynamical transport maps that serve as conditional generative models. We show that these transport maps can be learned by solving a simple square loss regression problem analogous to the standard independent setting. We demonstrate the usefulness of constructing dependent couplings in practice through experiments in super-resolution and in-painting.

Subjects: **Machine Learning (cs.LG)**; Machine Learning (stat.ML)
Cite as: arXiv:2310.03725 **[cs.LG]**
(or arXiv:2310.03725v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2310.03725

## Submission history

From: Mark Goldstein [view email]
**[v1]** Thu, 5 Oct 2023 17:46:31 UTC (16,569 KB)

## Access Paper:

- Download PDF
- PostScript
- Other Formats

(view license)

Current browse context:
**cs.LG**
< prev | next >
new | recent | 2310

Change to browse by:
cs
stat
stat.ML

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

Export BibTeX Citation

Bookmark

[Submitted on 16 Jan 2024]

# SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, Saining Xie

We present Scalable Interpolant Transformers (SiT), a family of generative models built on the backbone of Diffusion Transformers (DiT). The interpolant framework, which allows for connecting two distributions in a more flexible way than standard diffusion models, makes possible a modular study of various design choices impacting generative models built on dynamical transport: using discrete vs. continuous time learning, deciding the objective for the model to learn, choosing the interpolant connecting the distributions, and deploying a deterministic or stochastic sampler. By carefully introducing the above ingredients, SiT surpasses DiT uniformly across model sizes on the conditional ImageNet 256x256 benchmark using the exact same backbone, number of parameters, and GFLOPs. By exploring various diffusion coefficients, which can be tuned separately from learning, SiT achieves an FID-50K score of 2.06.

| | |
|---|---|
| Comments: | Code available: this https URL |
| Subjects: | **Computer Vision and Pattern Recognition (cs.CV)**; Machine Learning (cs.LG) |
| Cite as: | arXiv:2401.08740 **[cs.CV]** |
| | (or arXiv:2401.08740v1 **[cs.CV]** for this version) |

## Submission history

From: Nanye Ma [view email]

**[v1]** Tue, 16 Jan 2024 18:55:25 UTC (16,564 KB)

## Access Paper:

- Download PDF
- HTML (experimental)
- Other Formats



Current browse context:
**cs.CV**

< prev | next >

new | recent | 2401

Change to browse by:

cs

cs.LG

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

**Export BibTeX Citation**

Bookmark

[Submitted on 20 Mar 2024]

# Probabilistic Forecasting with Stochastic Interpolants and Föllmer Processes

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden

We propose a framework for probabilistic forecasting of dynamical systems based on generative modeling. Given observations of the system state over time, we formulate the forecasting problem as sampling from the conditional distribution of the future system state given its current state. To this end, we leverage the framework of stochastic interpolants, which facilitates the construction of a generative model between an arbitrary base distribution and the target. We design a fictitious, non-physical stochastic dynamics that takes as initial condition the current system state and produces as output a sample from the target conditional distribution in finite time and without bias. This process therefore maps a point mass centered at the current state onto a probabilistic ensemble of forecasts. We prove that the drift coefficient entering the stochastic differential equation (SDE) achieving this task is non-singular, and that it can be learned efficiently by square loss regression over the time-series data. We show that the drift and the diffusion coefficients of this SDE can be adjusted after training, and that a specific choice that minimizes the impact of the estimation error gives a Föllmer process. We highlight the utility of our approach on several complex, high-dimensional forecasting problems, including stochastically forced Navier-Stokes and video prediction on the KTH and CLEVRER datasets.

Subjects: **Machine Learning (cs.LG)**; Machine Learning (stat.ML)

Cite as: arXiv:2403.13724 **[cs.LG]**

(or arXiv:2403.13724v1 **[cs.LG]** for this version)

https://doi.org/10.48550/arXiv.2403.13724

## Submission history

From: Mark Goldstein [view email]

[v1] Wed, 20 Mar 2024 16:33:06 UTC (5,485 KB)

## Access Paper:

- View PDF
- HTML (experimental)
- TeX Source
- Other Formats

view license

Current browse context:
**cs.LG**

< prev | next >

new | recent | 2403

Change to browse by:

cs

stat

stat.ML

## References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

**Export BibTeX Citation**

## Bookmark

Computer Science > Machine Learning

[Submitted on 6 Aug 2025]

# Multitask Learning with Stochastic Interpolants

Hugo Negrel, Florentin Coeurdoux, Michael S. Albergo, Eric Vanden-Eijnden

We propose a framework for learning maps between probability distributions that broadly generalizes the time dynamics of flow and diffusion models. To enable this, we generalize stochastic interpolants by replacing the scalar time variable with vectors, matrices, or linear operators, allowing us to bridge probability distributions across multiple dimensional spaces. This approach enables the construction of versatile generative models capable of fulfilling multiple tasks without task-specific training. Our operator-based interpolants not only provide a unifying theoretical perspective for existing generative models but also extend their capabilities. Through numerical experiments, we demonstrate the zero-shot efficacy of our method on conditional generation and inpainting, fine-tuning and posterior sampling, and multiscale modeling, suggesting its potential as a generic task-agnostic alternative to specialized models.

Subjects: **Machine Learning (cs.LG)**; Dynamical Systems (math.DS)
Cite as: arXiv:2508.04605 **[cs.LG]**
(or arXiv:2508.04605v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2508.04605

## Submission history

From: Michael Albergo [view email]
reenshot ed, 6 Aug 2025 16:25:19 UTC (15,701 KB)