Diffusion Models

Ellen Zhong
Department of Computer Science
Princeton University

Fall 2026
COS 597Y Week 5



PRINCETON
UNIVERSITY

Diffusion everywhere



Diffusion everywhere





A whimsical and creative image depicting a hybrid creature that is a mix of a waffle and a hippopotamus. This imaginative creature features the distinctive, bulky body of a hippo, but with a texture and appearance resembling a golden-brown, crispy… (Esser et al., 2024)

Diffusion everywhere





A whimsical and creative image depicting a hybrid creature that is a mix of a waffle and a hippopotamus. This imaginative creature features the distinctive, bulky body of a hippo, but with a texture and appearance resembling a golden-brown, crispy… (Esser et al., 2024)



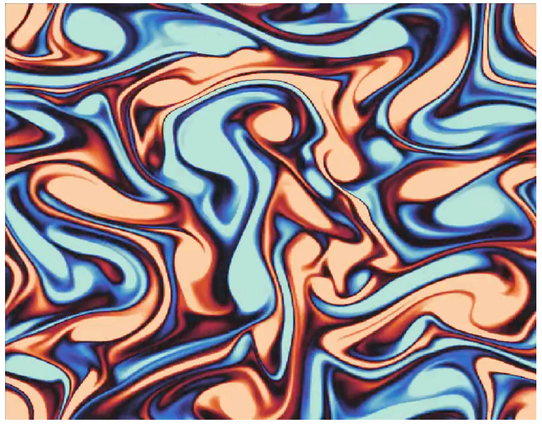
Diffusion everywhere





A whimsical and creative image depicting a hybrid creature that is a mix of a waffle and a hippopotamus. This imaginative creature features the distinctive, bulky body of a hippo, but with a texture and appearance resembling a golden-brown, crispy… (Esser et al., 2024)





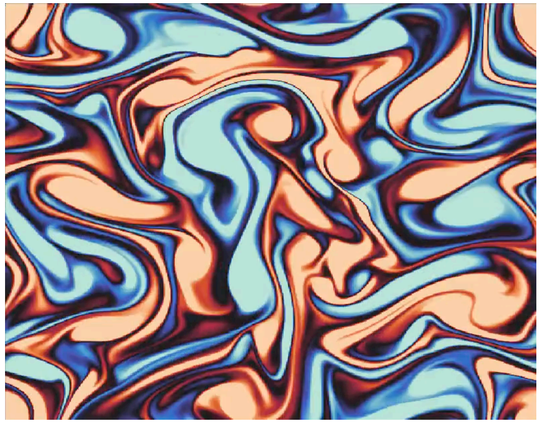
Diffusion everywhere





A whimsical and creative image depicting a hybrid creature that is a mix of a waffle and a hippopotamus. This imaginative creature features the distinctive, bulky body of a hippo, but with a texture and appearance resembling a golden-brown, crispy… (Esser et al., 2024)





Article

De novo design of protein structure and function with RFdiffusion

https://doi.org/10.1038/s41586-023-06415-8

Received: 14 December 2022

Accepted: 7 July 2023

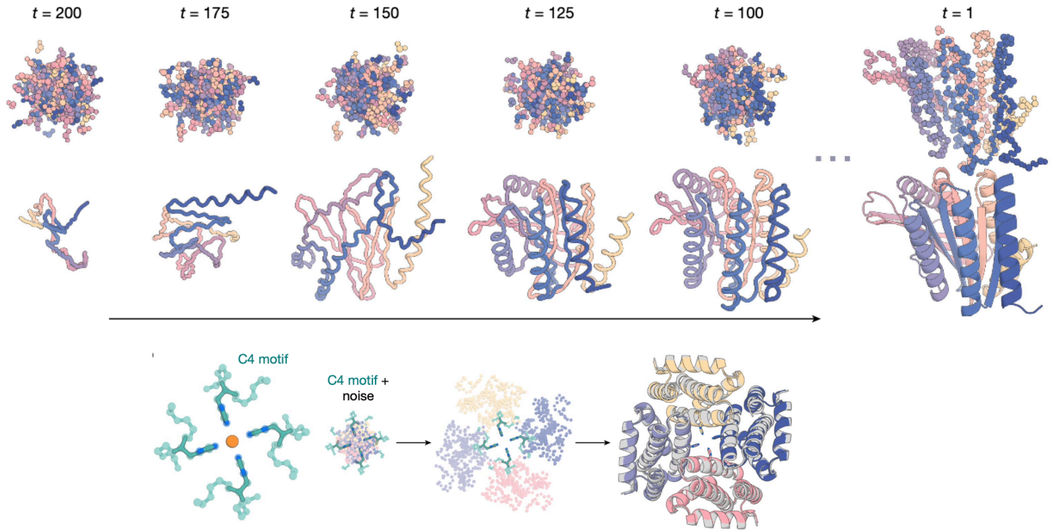
Published online: 11 July 2023

Open access

Check for updates

Joseph L. Watson[1,2,15], David Juergens[1,2,3,15], Nathaniel R. Bennett[1,2,3,15], Brian L. Trippe[2,4,5,15], Jason Yim[2,6,15], Helen E. Eisenach[1,2,15], Woody Ahern[1,2,7,15], Andrew J. Borst[1,2], Robert J. Ragotte[1,2], Lukas F. Milles[1,2], Basile I. M. Wicky[1,2], Nikita Hanikel[1,2], Samuel J. Pellock[1,2], Alexis Courbet[1,2,8], William Sheffler[1,2], Jue Wang[1,2], Preetham Venkatesh[1,2,9], Isaac Sappington[1,2,9], Susana Vázquez Torres[1,2,9], Anna Lauko[1,2,9], Valentin De Bortoli[8], Emile Mathieu[10], Sergey Ovchinnikov[11,12], Regina Barzilay[6], Tommi S. Jaakkola[6], Frank DiMaio[1,2], Minkyung Baek[13] & David Baker[1,2,14]✉





Diffusion everywhere





A whimsical and creative image depicting a hybrid creature that is a mix of a waffle and a hippopotamus. This imaginative creature features the distinctive, bulky body of a hippo, but with a texture and appearance resembling a golden-brown, crispy… (Esser et al., 2024)

Article

De novo design of protein structure and function with RFdiffusion

https://doi.org/10.1038/s41586-023-06415-8

Received: 14 December 2022

Accepted: 7 July 2023

Published online: 11 July 2023

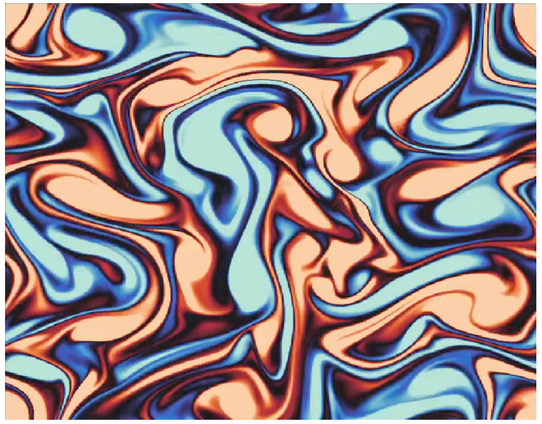
Open access

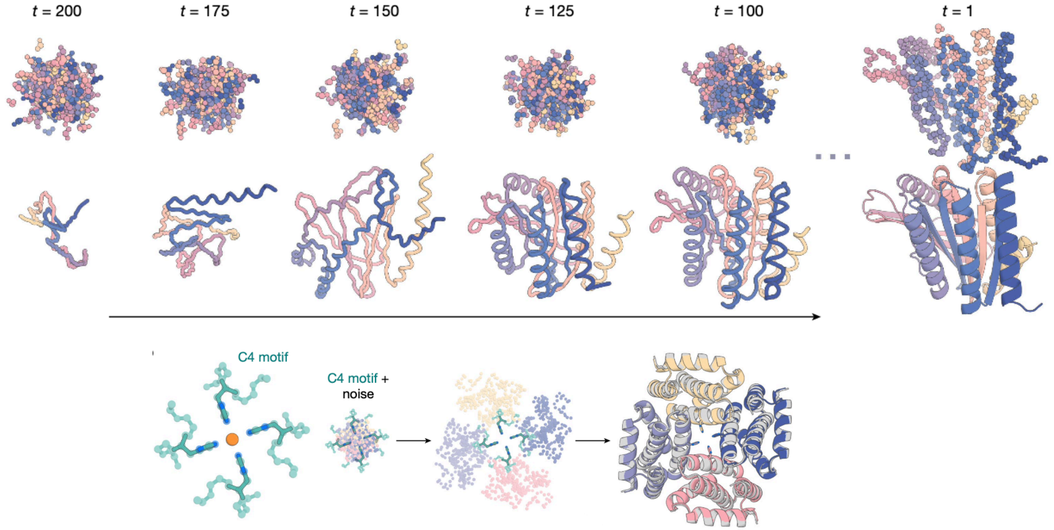
Check for updates

Joseph L. Watson[1,2,15], David Juergens[1,2,3,15], Nathaniel R. Bennett[1,2,3,15], Brian L. Trippe[2,4,5,15], Jason Yim[2,6,15], Helen E. Eisenach[1,2,15], Woody Ahern[1,2,15], Andrew J. Borst[1,2], Robert J. Ragotte[1,2], Lukas F. Milles[1,2], Basile I. M. Wicky[1,2], Nikita Hanikel[1,2], Samuel J. Pellock[1,2], Alexis Courbet[1,2,8], William Sheffler[1,2], Jue Wang[1,2], Preetham Venkatesh[1,2,9], Isaac Sappington[1,2,9], Susana Vázquez Torres[1,2,9], Anna Lauko[1,2,9], Valentin De Bortoli[8], Emile Mathieu[10], Sergey Ovchinnikov[11,12], Regina Barzilay[6], Tommi S. Jaakkola[6], Frank DiMaio[1,2], Minkyung Baek[13] & David Baker[1,2,14]✉









Diffusion models are fantastic tools for generating data

Diffusion models — Many perspectives

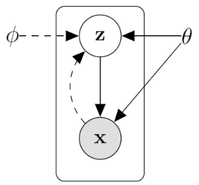
- “Diffusion models are a latent variable model (VAE) with many Z’s”
  - Variational Inference — DDPM
- “Diffusion models are a sequence of score-based models”
  - Score matching — Score-based generative models $s_\theta(x, t) \approx \nabla_x \log p_t(x)$
- “Diffusion models are about thermodynamics and time reversal”
  - Stochastic processes — SDE perspective
- “Generative modeling is transporting one distribution into another”
  - Optimal transport / flow matching — ODE perspective

Timeline

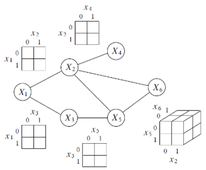
- **2015 Deep Unsupervised Learning using Nonequilibrium Thermodynamics (Sohl-Dickstein et al.)**
  Introduced the foundational framework for diffusion probabilistic models, drawing inspiration from non-equilibrium statistical physics to iteratively destroy and reconstruct data structures.
- **2019/2020 — Generative Modeling by Estimating Gradients of the Data Distribution (Song & Ermon) & DDPM (Ho et al.)**
  Yang Song and Stefano Ermon popularized score-based generative modeling via Langevin dynamics, while Ho et al. introduced Denoising Diffusion Probabilistic Models (DDPM), establishing the equivalence between score-matching objectives and noise prediction.
- **2021 — Score-Based SDE Unification (Song et al.)**
  Unified discrete diffusion and score-matching methods into a single continuous-time Stochastic Differential Equation (SDE) framework, introducing deterministic probability flow ODEs.
- **2022 — Latent Diffusion Models (Stable Diffusion) (Rombach et al.)**
  Shifted the diffusion denoising process from pixel space into a lower-dimensional latent space, drastically reducing compute requirements and enabling high-resolution consumer-grade generation.
- **2022-now— Flow Matching, Discrete Diffusion Models**
  Expansion of diffusion-style generative modeling to new dynamics and state spaces.

Recap: Generative modeling

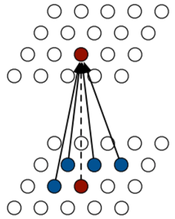
Likelihood-based models



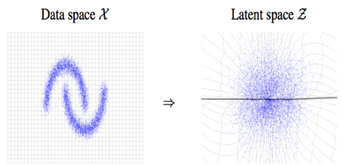
Bayesian networks
(e.g., VAEs)



MRF

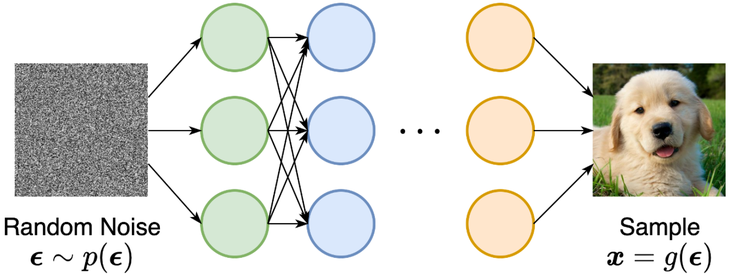


Autoregressive
models



Flow models

Implicit generative models



Random Noise
$\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$

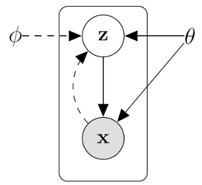
Sample
$\boldsymbol{x} = g(\boldsymbol{\epsilon})$

GANs

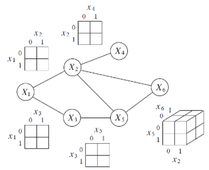
* In generative modeling, there are usually two opposing forces: tractability and flexibility

Recap: Generative modeling

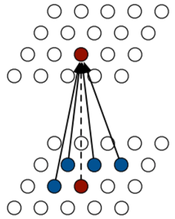
Likelihood-based models



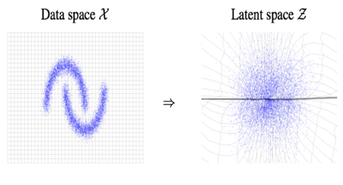
Bayesian networks (e.g., VAEs)



MRF



Autoregressive models



Flow models

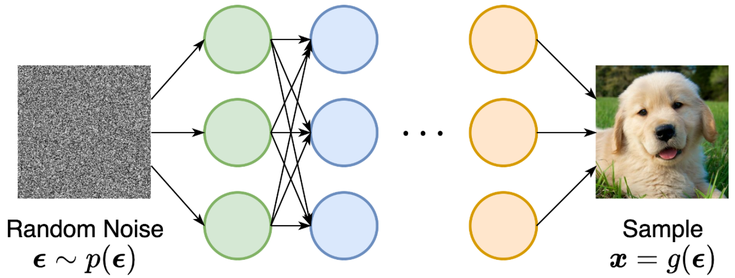
* Typically make strong assumptions to ensure tractability of likelihood

* In generative modeling, there are usually two opposing forces: tractability and flexibility

Recap: Generative modeling

* Adversarial training is often unstable
* Mode dropping
* Cannot compute likelihood of a sample

Implicit generative models

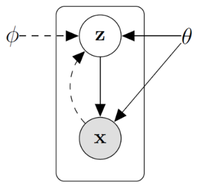


GANs

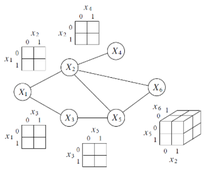
* In generative modeling, there are usually two opposing forces: tractability and flexibility

Recap: Generative modeling

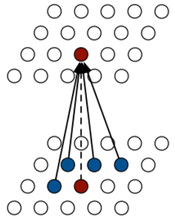
Likelihood-based models



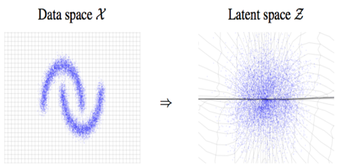
Bayesian networks
(e.g., VAEs)



MRF

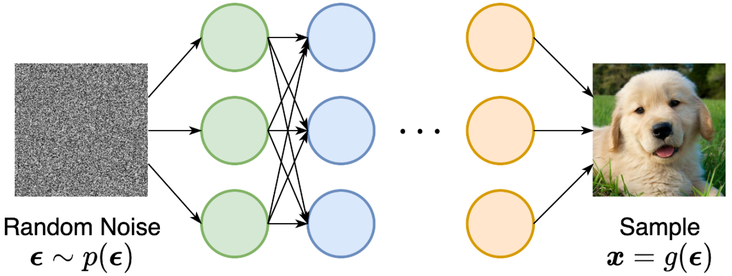


Autoregressive
models



Flow models

Implicit generative models



Random Noise
$\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$

Sample
$\boldsymbol{x} = g(\boldsymbol{\epsilon})$

GANs

* In generative modeling, there are usually two opposing forces: tractability and flexibility
* Diffusion/score-based generative models are both tractable and flexible

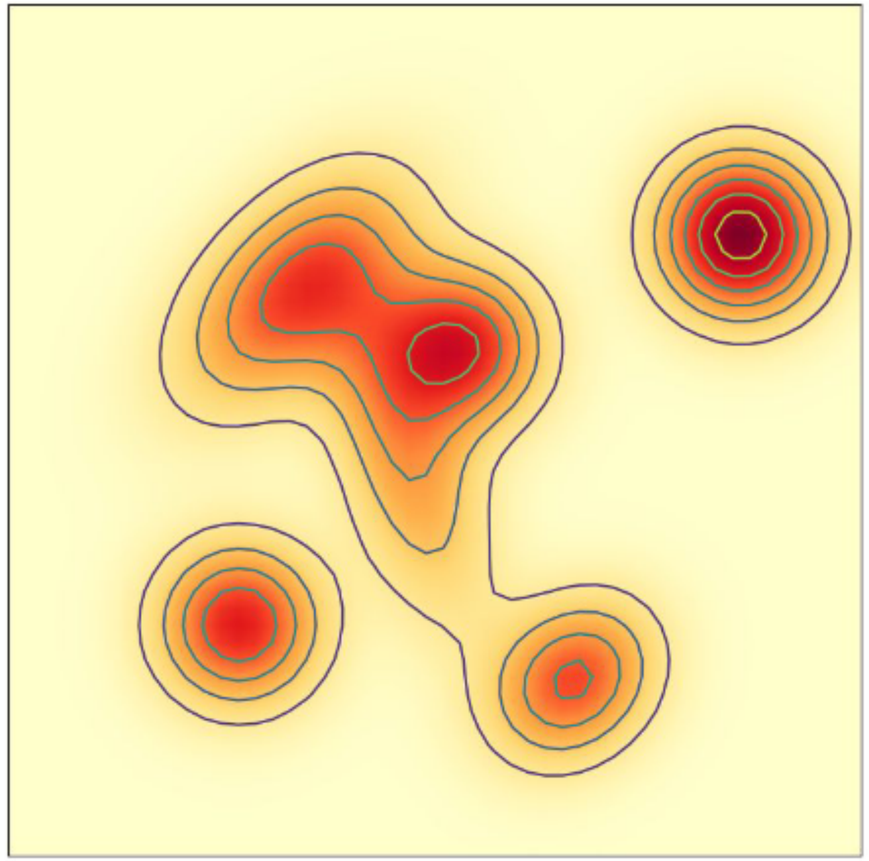
Diffusion models — Many perspectives

- “Diffusion models are a latent variable model (VAE) with many Z’s”
  - Variational Inference — DDPM
- **“Diffusion models are a sequence of score-based models”**
  - Score matching — Score-based generative models $s_\theta(x, t) \approx \nabla_x \log p_t(x)$
- “Diffusion models are about thermodynamics and time reversal”
  - Stochastic processes — SDE perspective
- “Generative modeling is transporting one distribution into another”
  - Optimal transport / flow matching — ODE perspective

Score Functions = Gradients of the log-likelihood

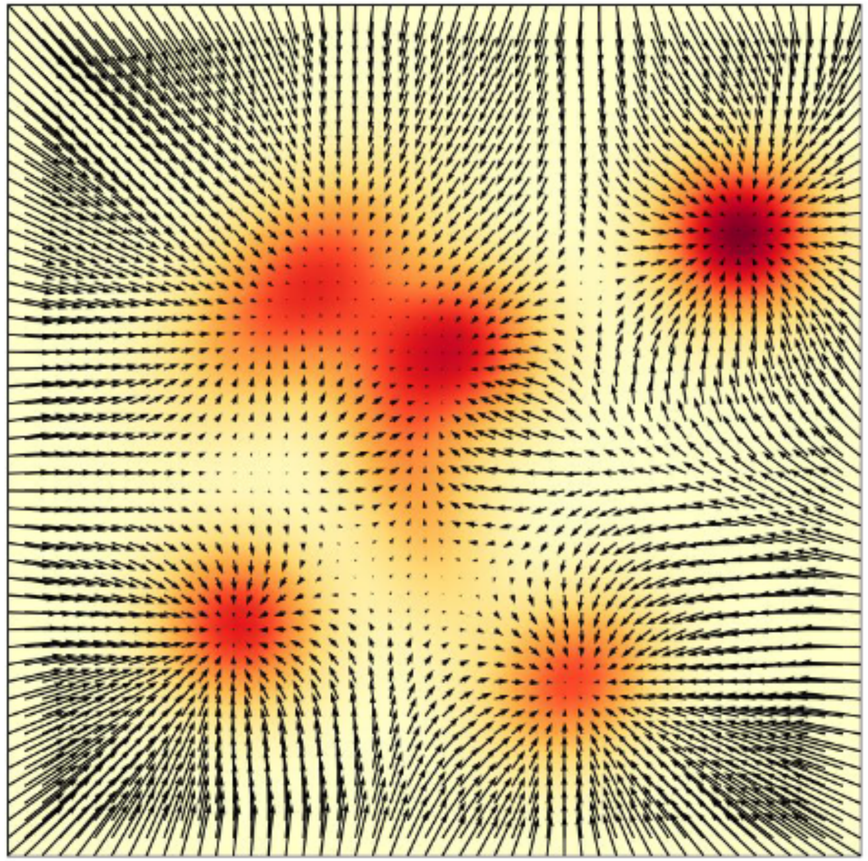
Log-likelihood:

$$\log p(x)$$



Score function:

$$s(x) = \nabla_x \log p(x)$$



Score-based generative models

- A model $s_\theta(x)$, which models the score function explicitly, is a score-based generative model

$$s_\theta(x) \approx \nabla_x \log p(x),$$

- Score-based models do not need a tractable normalizing constant
- And can be learned by minimizing the Fisher divergence:

$$\mathbb{E}_{p_{data}(x)} \left[\| s_\theta(x) - \nabla_x \log p_{data}(x) \|^2 \right]$$

- Rewrite as the score-matching objective to train via SGD:

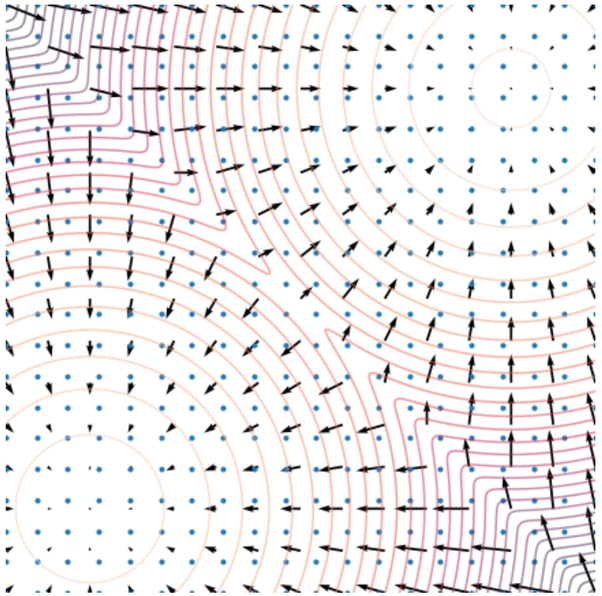
$$\mathscr{L}_{\text{SM}} = \mathbb{E}_{p_{\text{data}}(x)} \left[\frac{1}{2} \| s_\theta(x) \|^2 + \nabla_x \cdot s_\theta(x) \right]$$

Sampling with langevin dynamics

- After training, we can use an iterative procedure called **Langevin dynamics** to draw samples
- Langevin dynamics provides an MCMC procedure to sample from $p(x)$ using only $\nabla_x \log p(x)$

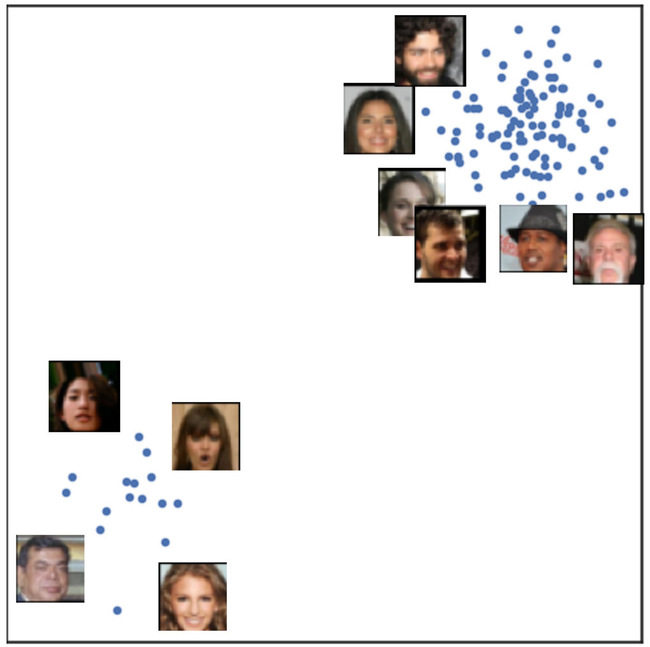
$$x_{i+1} \leftarrow x_i + \epsilon \nabla_x \log p(x) + \sqrt{2\epsilon}\, z_i \qquad i = 0,1,\ldots,K \qquad z_i \sim \mathcal{N}(0,I)$$

- Converges to a sample in $p(x)$ when $\epsilon \to 0$ and $K \to \infty$
- We can plug in our score-based model $s_\theta(x)$



Sample from a mixture of two Gaussians

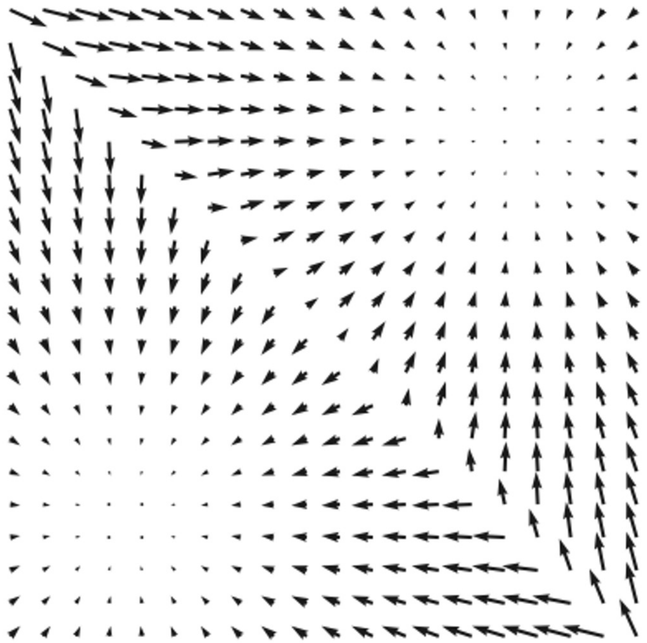
Naive score-based generative modeling



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$$

score matching



Scores

$$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Langevin dynamics



New samples

https://yang-song.net/blog/2021/score/

Aside: Langevin dynamics for proteins

- Basis of molecular dynamics simulations

$$dX_t = \frac{1}{2}\nabla_x \log p(X_t)\,dt + dW_t$$

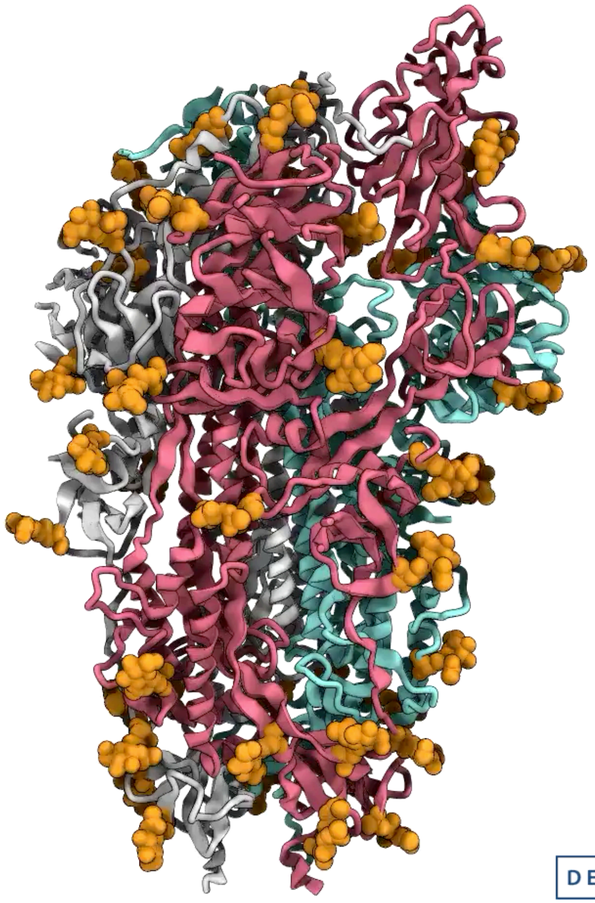
- Sample from a distribution (i.e. Boltzmann) given its score function (i.e. force field in MD)

$$p(x) = p_{\text{Boltzmann}}(x) = \frac{1}{Z}e^{-\beta U(x)}$$

- The score is the negative gradient of the energy:

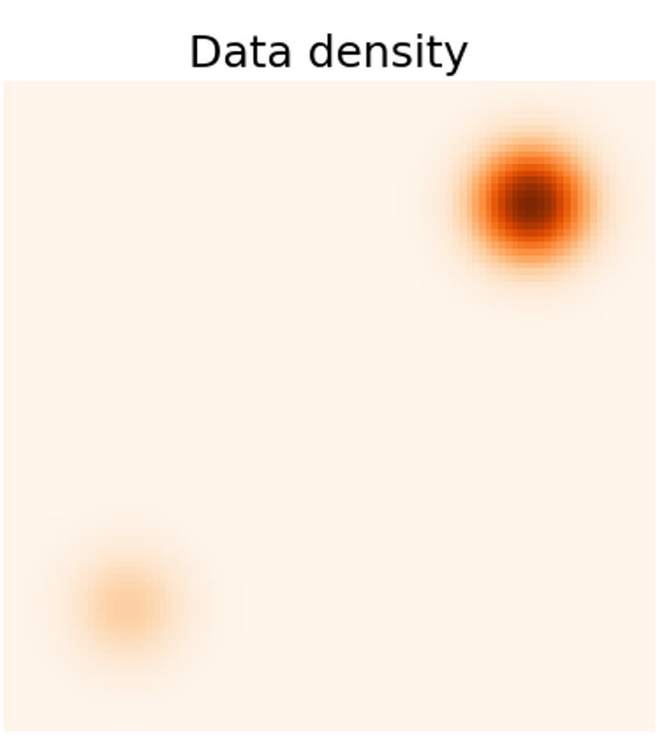
$$s(x) = \nabla_x \log p_{\text{B}}(x) = -\nabla_x U(x)$$

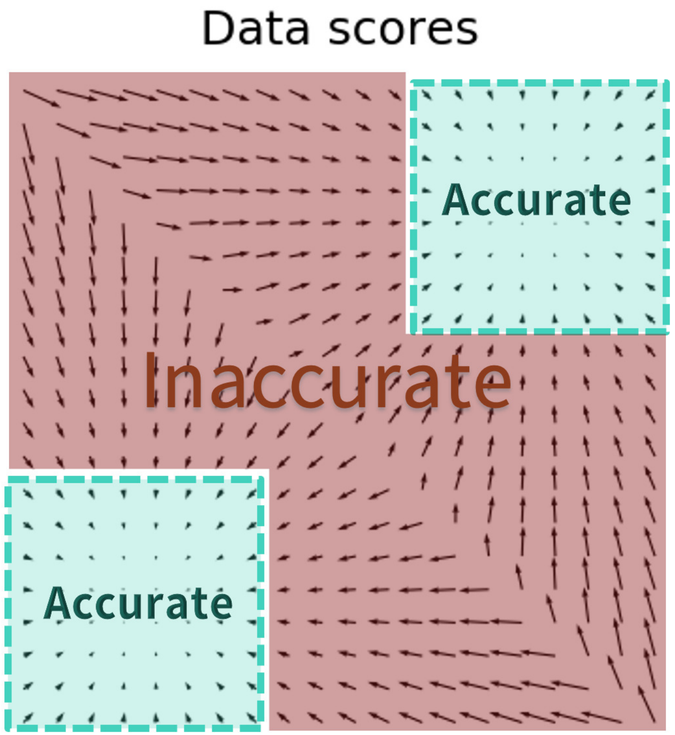
- Langevin dynamics combines the gradient with random thermal noise
- Caveat: Slowing mixing times

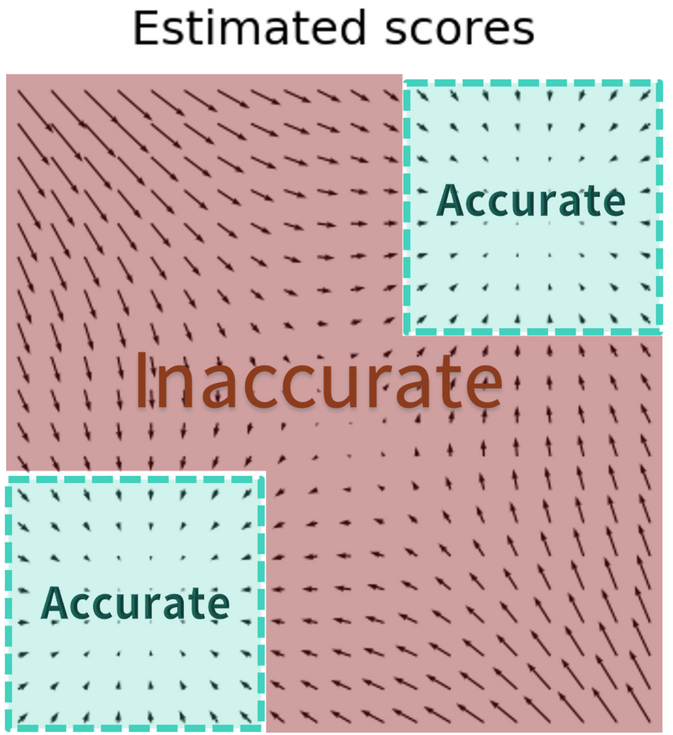


D E Shaw Research

Pitfalls



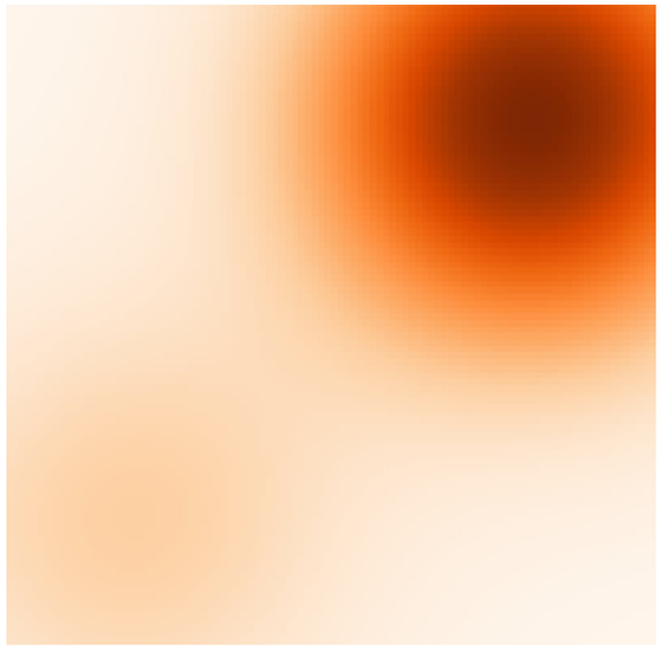




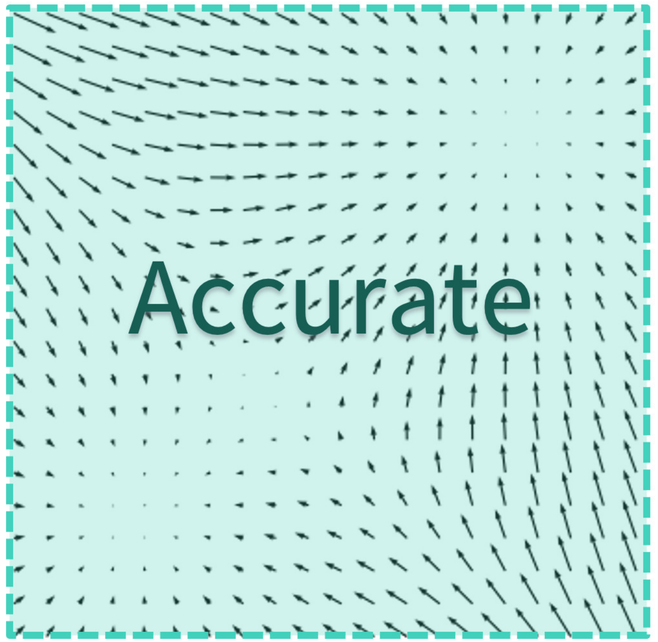
https://yang-song.net/blog/2021/score/

Solution

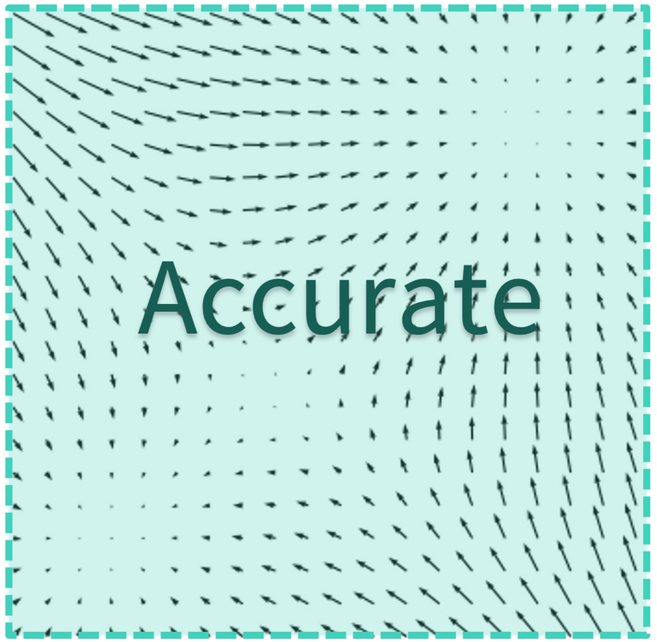
Perturbed density



Perturbed scores



Estimated scores



Perturb the data with Gaussian noise:

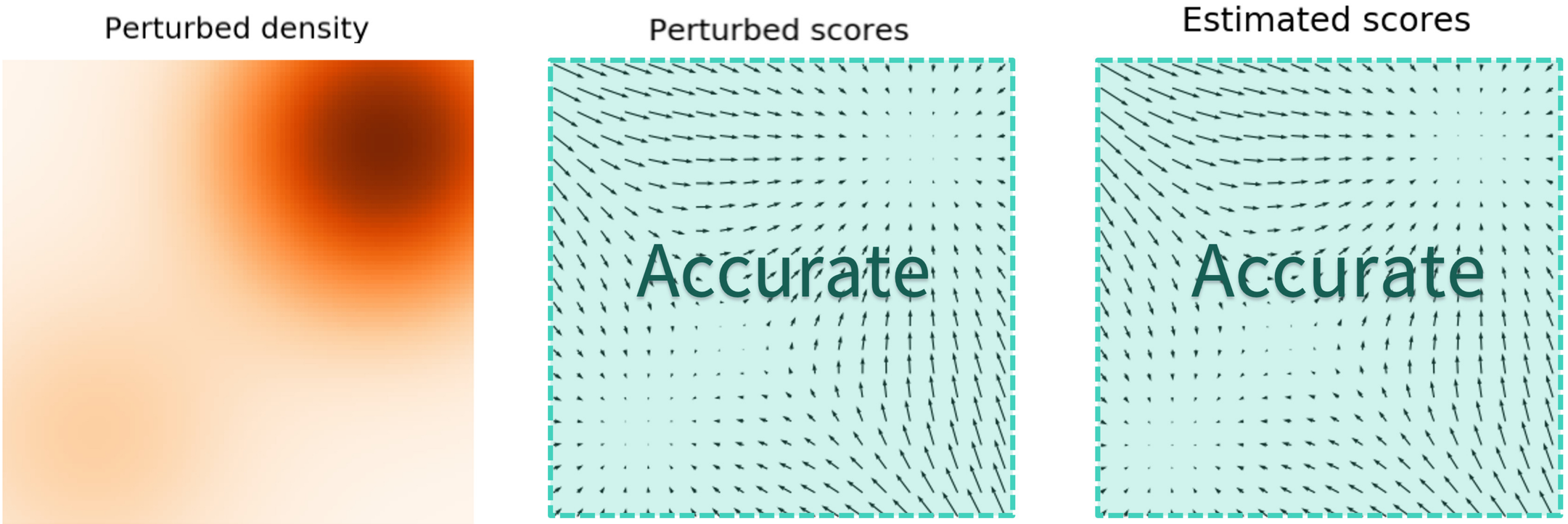
$$x_t = \alpha_t x_0 + \sigma_t \epsilon$$

Denoising objective:

$$\mathscr{L}_{\text{DSM}} = \mathbb{E}\left[\left\| s_\theta(x_t, t) + \frac{\epsilon}{\sigma_t} \right\|^2 \right]$$

https://yang-song.net/blog/2021/score/

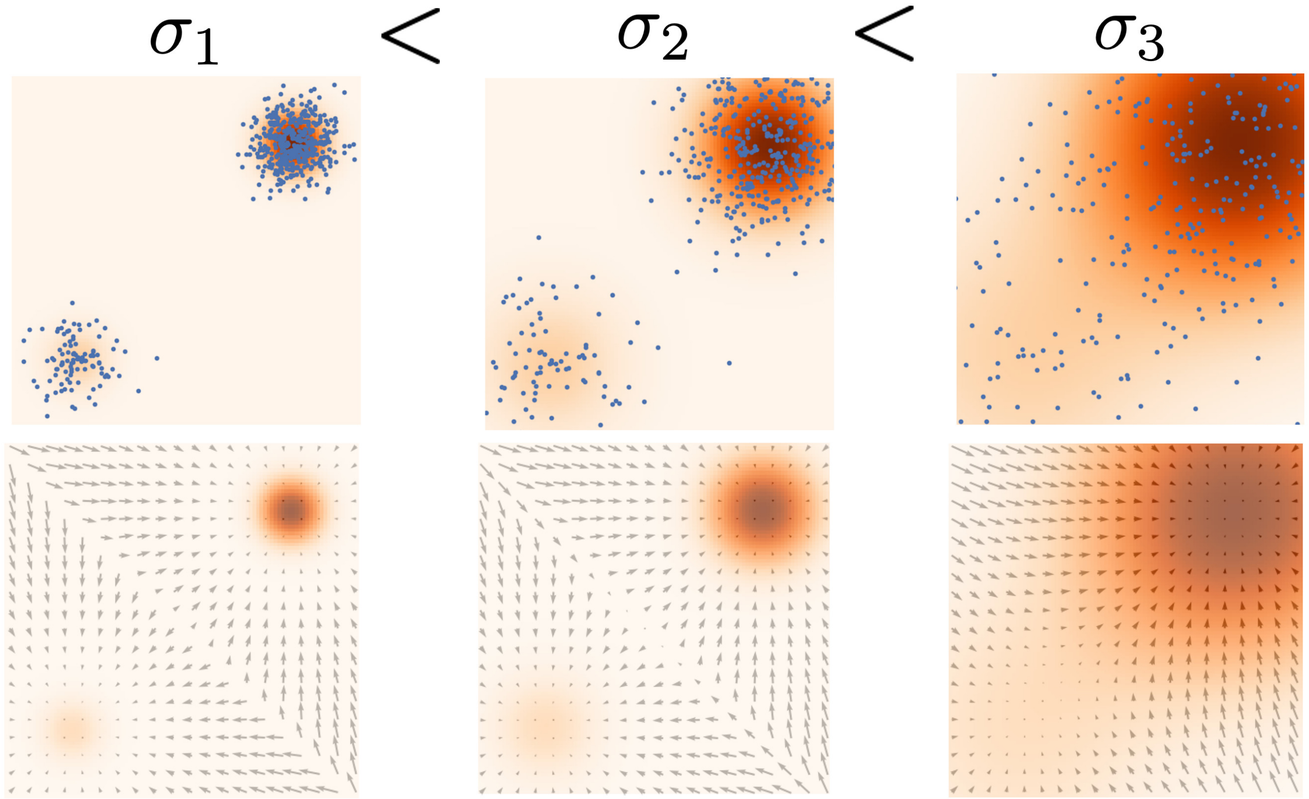
Solution



How do we choose an appropriate noise scale for the perturbation process?

https://yang-song.net/blog/2021/score/

Noise Conditional Score-based Model



Noise Conditional Score Network (NCSN)

$$s_\theta(x, i) \approx \nabla_x \log p_{\sigma_i}(x)$$

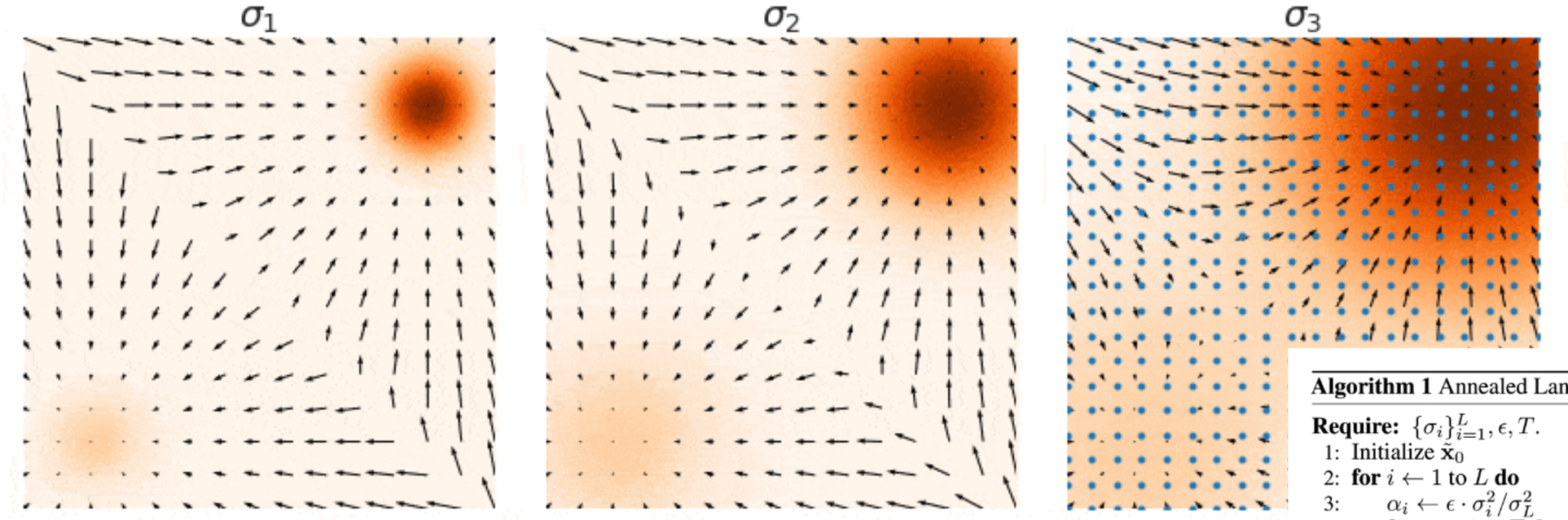
for all $i = 1,2,...,L$



https://yang-song.net/blog/2021/score/

Sampling with annealed Langevin dynamics

- Annealed Langevin dynamics combine a sequence of Langevin chains with gradually decreasing noise scales.



Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

Initialize $\tilde{\mathbf{x}}_0$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
  for $t \leftarrow 1$ to T **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
  end for
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
end for
return $\tilde{\mathbf{x}}_T$

- Produce samples from it by running Langevin dynamics for $i = L, L-1, \ldots, 1$ in sequence

https://yang-song.net/blog/2021/score/

Hyperparmeters circa 2021

- Choose $\sigma_1 < \sigma_2 < \ldots < \sigma_L$ as a geometric progression
- L typically in the hundreds or thousands
- Parameterize $s_\theta(x, i)$ with U-Net skip connections
- Apply EMA on model weights when used at test time

- .... Why hardcode noise levels?

Score-based models with SDEs

- Adding noise is important, but why ‘hardcode’?
  - What if we could take just one step per noise level?
- By generalizing to infinite noise scales, we can:
  - get higher quality samples
  - exact log-likelihood computation
  - controllable generation with inverse problem solving
- A **stochastic process** defines a process of generating infinite noise scales

Continuous time! (Math warmup)

- Just two bits of math, for our purposes:

- $dX = f(x,t)dt \quad \rightarrow \quad X_{t+\delta} = X_t + \delta f(x,t)$
- $dX = g(t)dW_t \quad \rightarrow \quad X_{t+\delta} = X_t + g(t)\sqrt{\delta}\epsilon, \qquad \epsilon \sim \mathcal{N}(0,I)$

Wiener process / Brownian motion

- W_t is Wiener process:
  - $W_0 = 0$
  - $W_{t+\delta} - W_t$ is Gaussian and independent of past, variance δ
  - Some sample path continuity we won't discuss here
  - Basically, a running-sum-so-far for many many Gaussians
  - dW_t is heuristically $W_{t+\delta} - W_t$ for small δ
  - $dX = g(t)dW_t$ called ``integration w.r.t. Brownian motion"

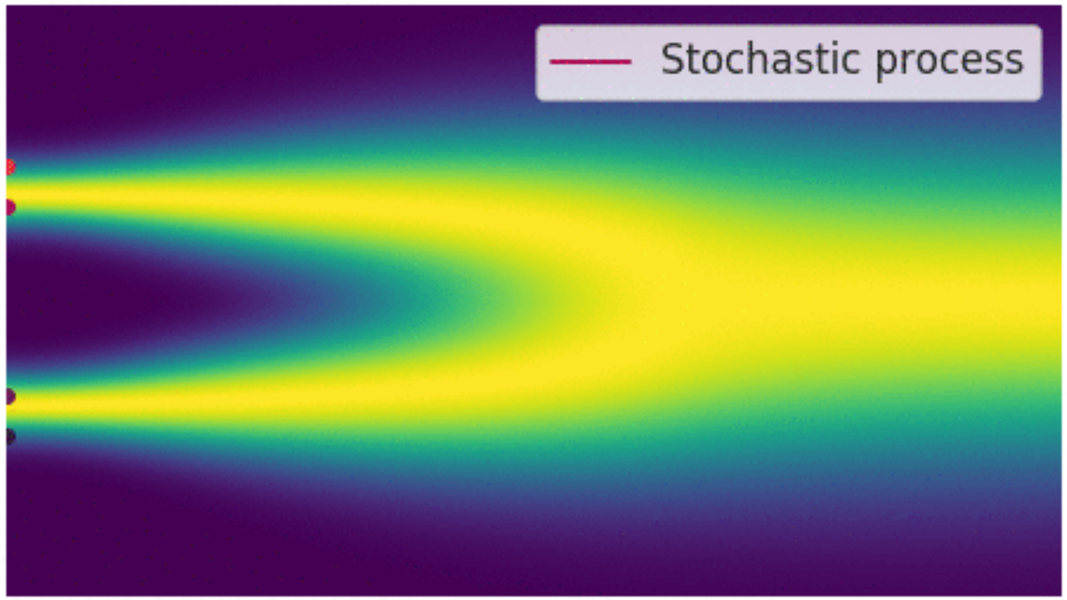
Wiener process / Brownian motion

- W_t is Wiener process:
  - $W_0 = 0$
  - $W_{t+\delta} - W_t$ is Gaussian and independent of past, variance δ
  - Some sample path continuity we won't discuss here
  - Basically, a running-sum-so-far for many many Gaussians
  - dW_t is heuristically $W_{t+\delta} - W_t$ for small δ
  - $dX = g(t)dW_t$ called ``integration w.r.t. Brownian motion"

Stochastic processes via SDEs

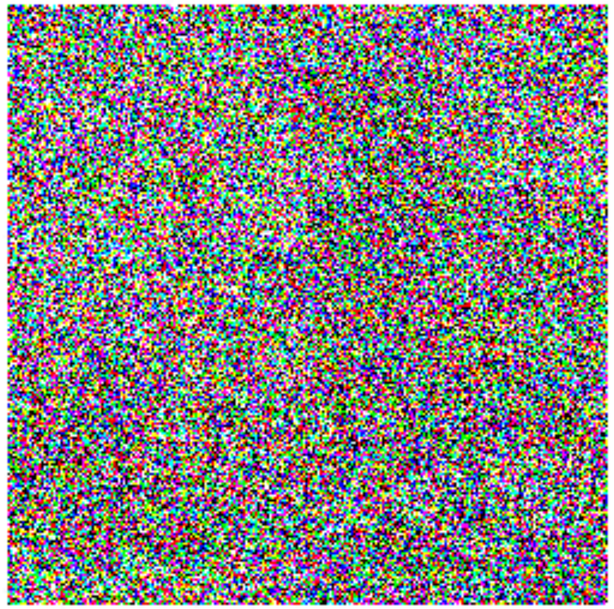
- A stochastic process can be defined in terms of (solution of) a stochastic differential equation:
  - $dX = f(X, t)dt + g(t)dW_t$
- For certain choices of f, g, $X_t \mid X_0$ is simple

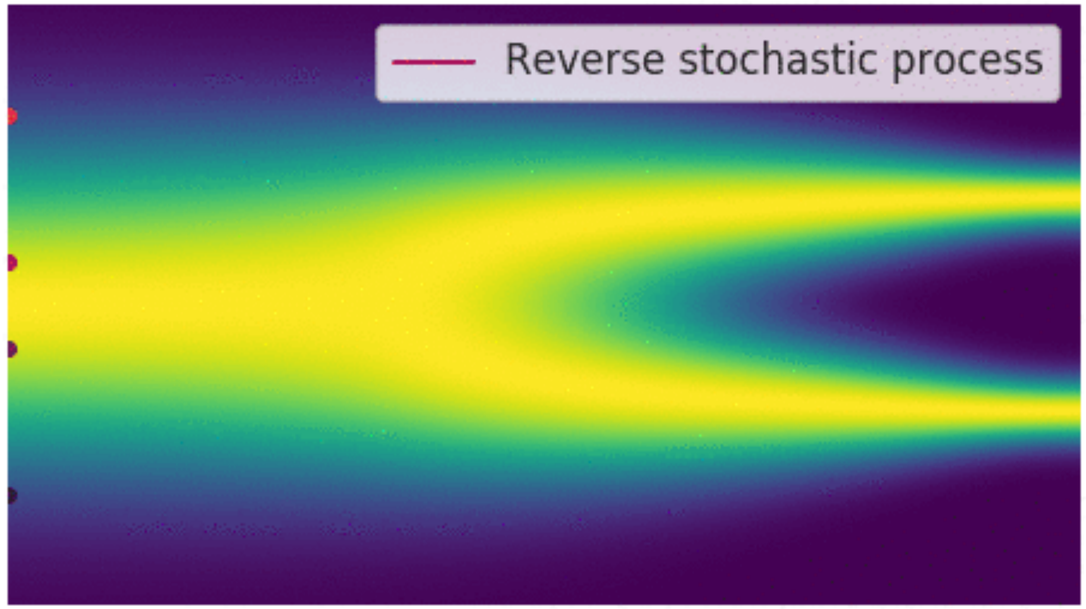




Reversing the SDE for sample generation

- For any SDE there is a reverse SDE, which corresponds to the reverse trajectories:
- $dX_t = \left[f(X_t, t) - g(t)^2 \nabla_x \log p_t(X_t) \right] dt + g(t)\, dW_t$





- Solving a reverse SDE yields a score-based generative model

Reversing the SDE for sample generation

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$



score function

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

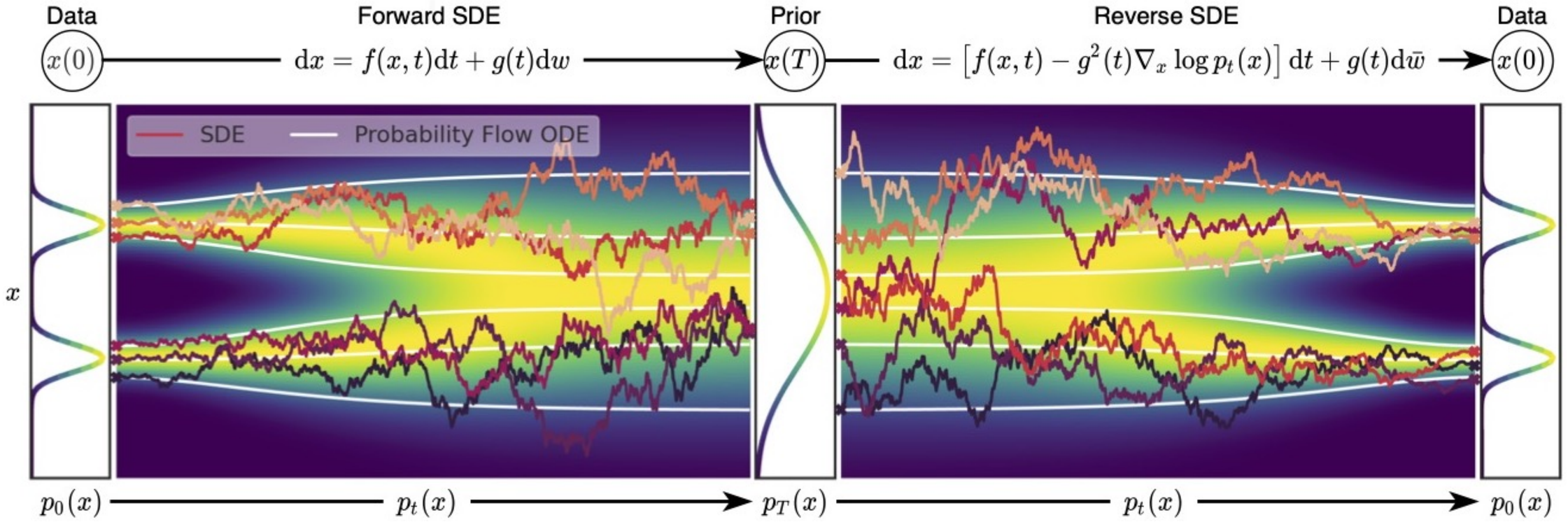
- Analogously learned with denoising score matching / continuous weighted combination of Fisher divergences
- Reverse SDE solved with numerical SDE solvers (e.g. Euler-Maruyama method)

Probability flow ODE

- We can convert SDE into a corresponding ODE without changing its marginal distributions $\{p_t(x)\}_{t\in[0,T]}$

$$dX_t = [f(X_t, t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(X_t)]dt$$

- Allows for exact log-likelihood computation



Flow matching: Learn the velocity field directly

Flow Model

Initialize: $X_0 \sim p_{\text{init}},$

ODE: $\mathrm{d}X_t = u_t^\theta(X_t)\mathrm{d}t$

E.g. Gaussian

Neural network vector field

Diffusion coeff.

Diffusion Model

Initialize: $X_0 \sim p_{\text{init}},$

SDE: $\mathrm{d}X_t = u_t^\theta(X_t)\mathrm{d}t + \sigma_t \mathrm{d}W_t$

To get samples, simulate ODE/SDE from t=0 to t=1 and return X_1

Different interpretations/conventions

“Flow time convention”:

- Data: t = 1
- Noise: t = 0

Flow matching, rectified flows, stochastic interpolants

“Diffusion time convention”:

- Data t = 0
- Noise: t → infinity

Score-based diffusion models with SDEs

“Discrete time”:

- Use discrete time steps instead of continuous time steps
- No ODE or SDE but Markov chain
- DDIM ~= Probability flow ODE

DDPM
DDIM

Different interpretations/conventions

Probability path (here):

$$p_t(x|z) = \mathcal{N}(\alpha_t z, \beta_t^2 I_d)$$

Flow matching, rectified flows

Interpolant function:

$$I_t(\epsilon, z) = \alpha_t \epsilon + \beta_t z$$

Stochastic interpolants

"Forward" diffusion process:

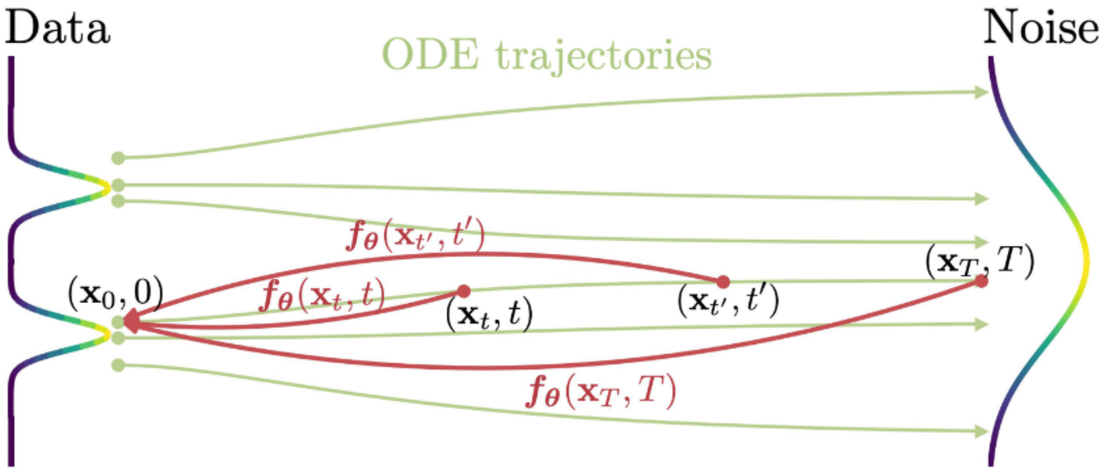
$$\mathrm{d}X_t = a_t(X_t)\mathrm{d}t + \sigma_t \mathrm{d}W_t$$

Denoising diffusion models

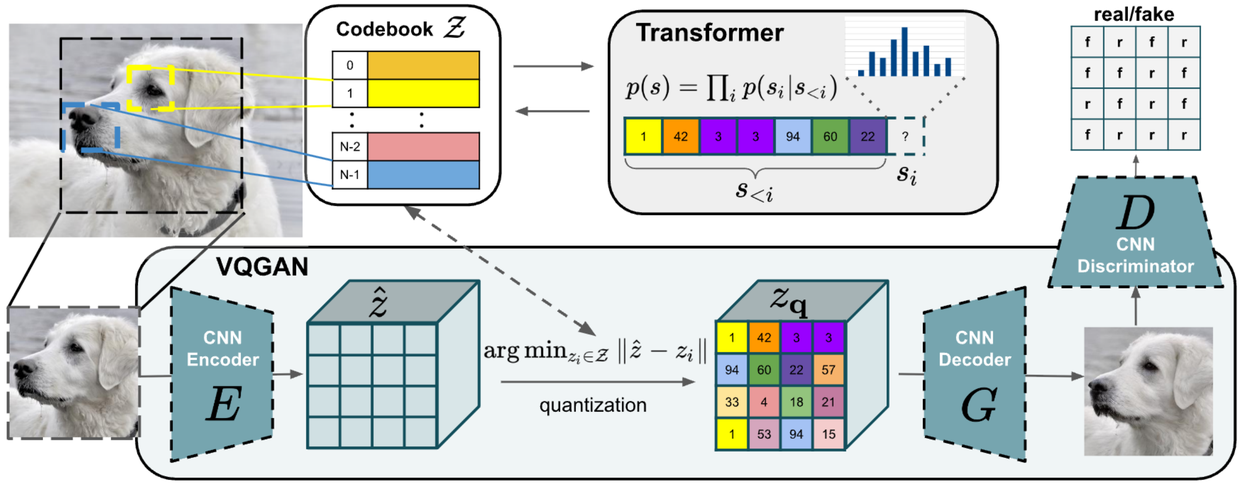
Note: For Gaussian probability paths, the above procedures are equivalent.

Active research directions

- Why do diffusion models generalize? (i.e. produce new stuff)
- Faster generation. How to train diffusion models to need fewer inference steps?
- Discrete diffusion. Diffusion modeling vs autoregressive generation



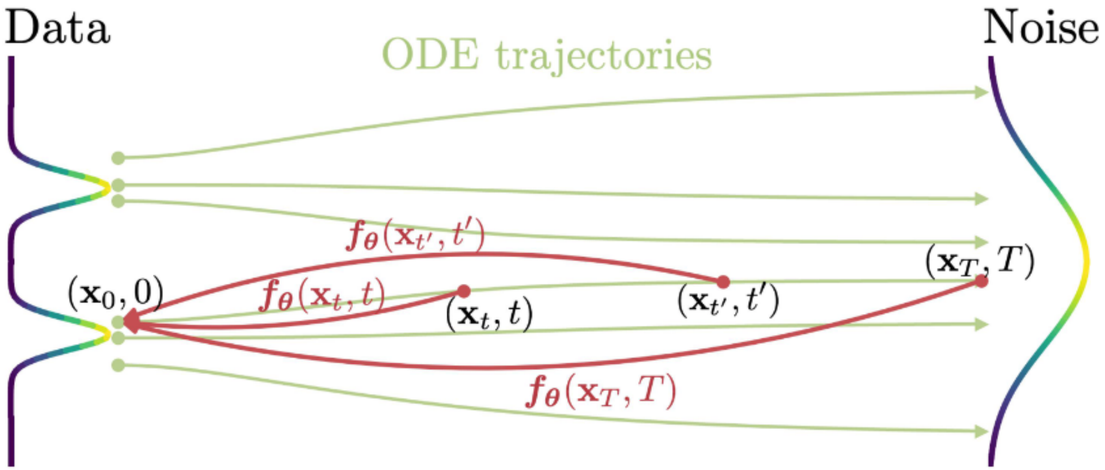
Consistency Models / Flow Maps



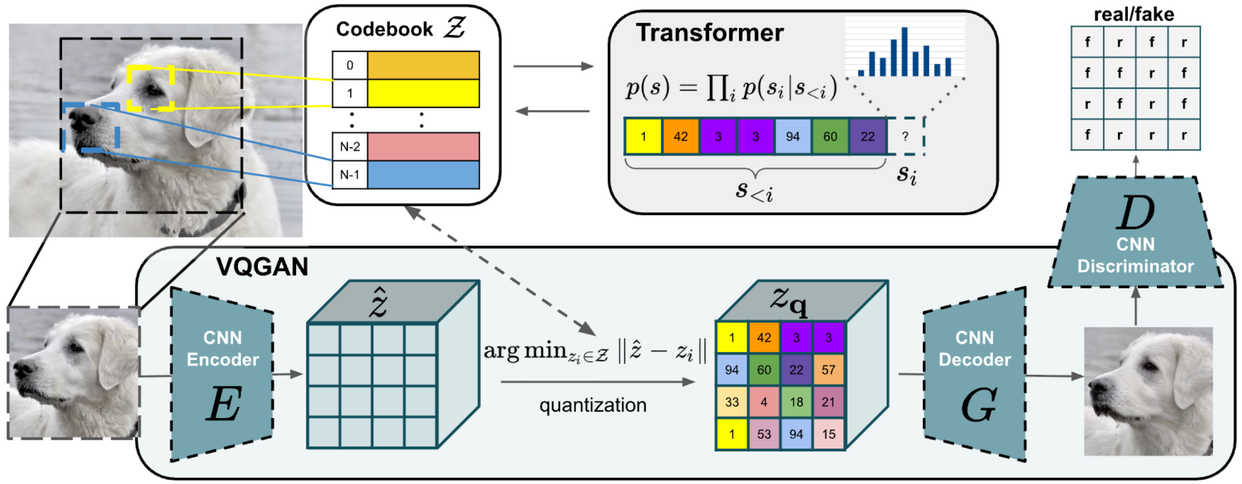
Vector-Quantization approach: Esser et al, 2021, Ramesh et al, 2021

Active research directions

- Better conditioning and controllability (Next week)
- Structured / geometric diffusion (Chroma, Bowen Jing seminar)
- Scientific generative modeling. Data representations, physical constraints, discovery, etc.



Consistency Models / Flow Maps



Vector-Quantization approach: Esser et al, 2021, Ramesh et al, 2021

Next Week

- Inverse problems / (diffusion) generative models for scientific inference
- How to use diffusion models for science?
- Readings:
  - Proteina-complexa
    - Scaling Atomistic Protein Binder Design with Generative Pretraining and Test-Time Compute (Didi et al. ICLR 2026)
  - CryoBoltz
    - Multiscale Guidance of Protein Structure Prediction with Heterogeneous Cryo-EM Data (Raghu et al. NeurIPS 2025)

Presentations

- DDPM (Binxu and Haoyi)
- Chroma (Yury and Jiashi)

- Post-lecture feedback (anonymous to the presenters, due at 11:59pm):
  - What worked well?
  - What could be improved?
  - What is one question, idea, or takeaway you had?
- Feedback is a gift and giving good feedback is a skill!
- Good feedback is specific, actionable, and constructive.