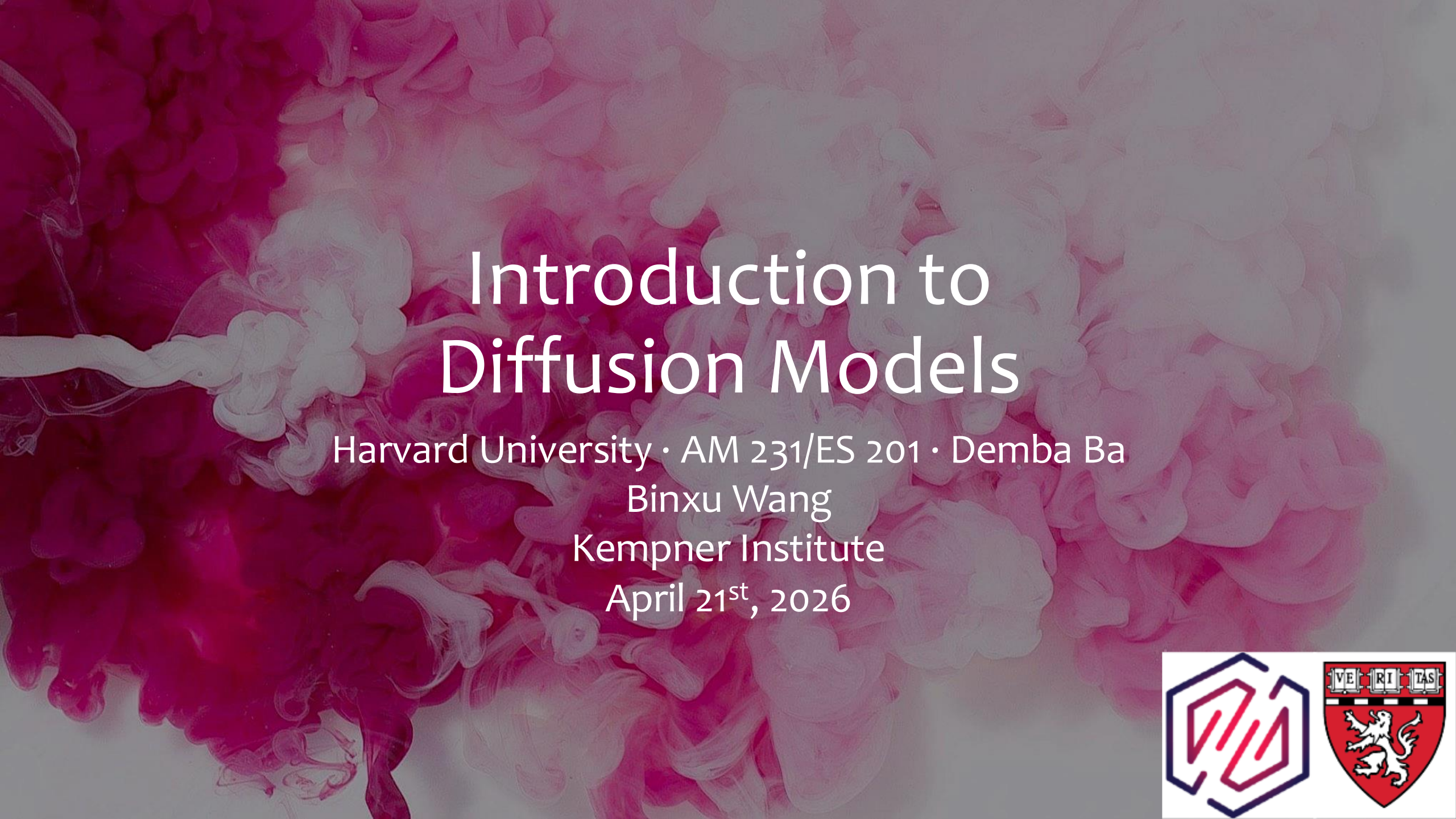


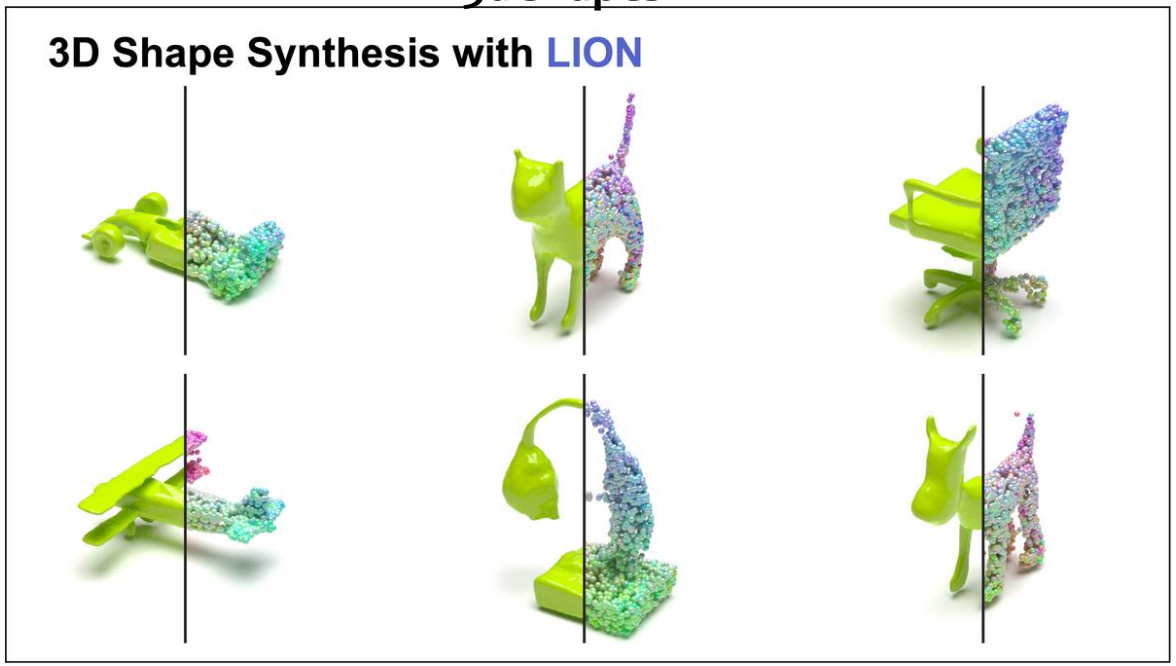
Introduction to Diffusion Models

Harvard University · AM 231/ES 201 · Demba Ba
Binxu Wang
Kempner Institute
April 21st, 2026

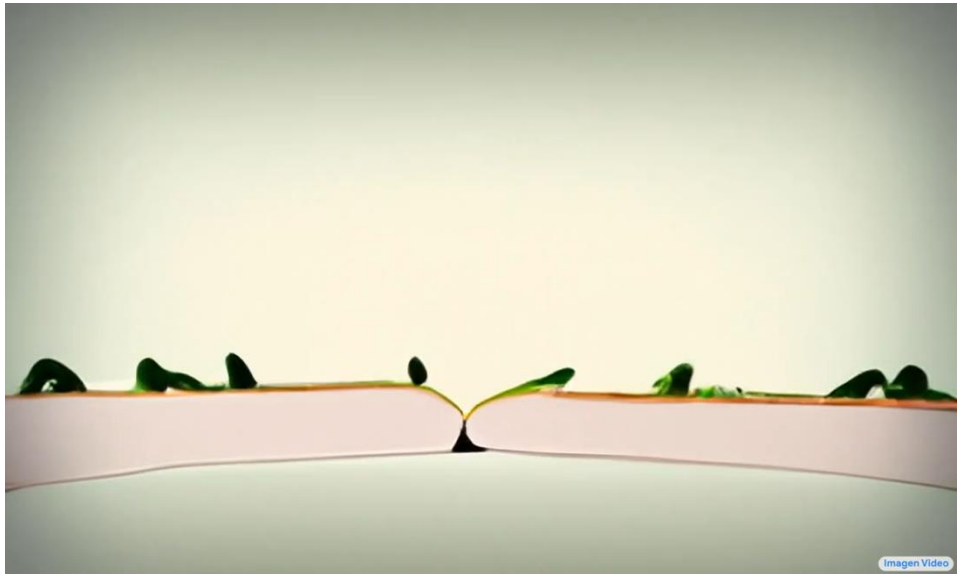
Diffusion models are leading the charge…

Images

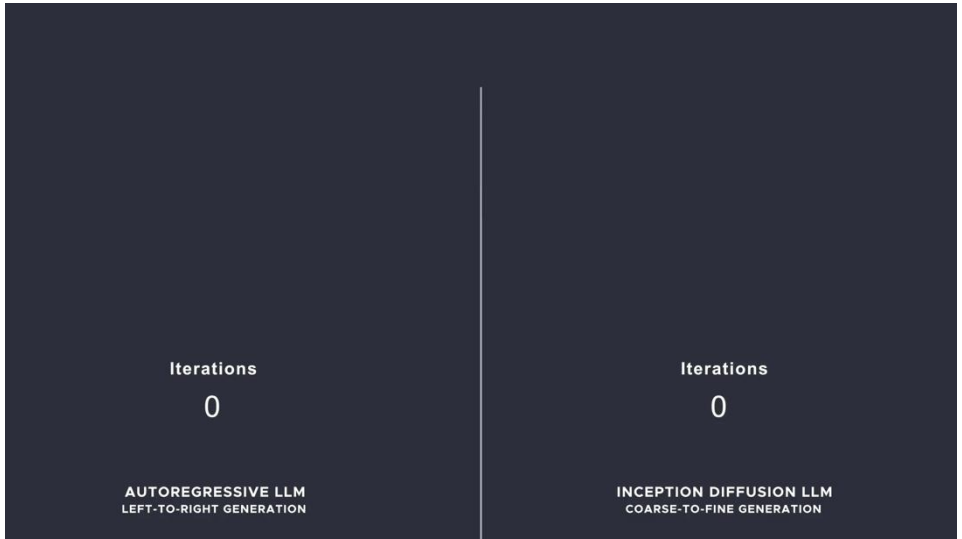
3d shapes



Videos



Language / Code



Roadmaps

- Mathematical foundation
  - Sampling, Training
- Connection to other models
  - VP, VE, Flow Matching, DDPM, VAE, (Flow)
- Practical details
  - Architecture, Training, Efficient Generation
- Conditioning Diffusion
  - Text, Class, Spatial

Mathematical Foundation of Diffusion Models

The Two Pillars of Diffusion

Pillar I Sampling

Probability Flow ODE (PF)

How to transform data distribution → Gaussian, and back?

Pillar II Training

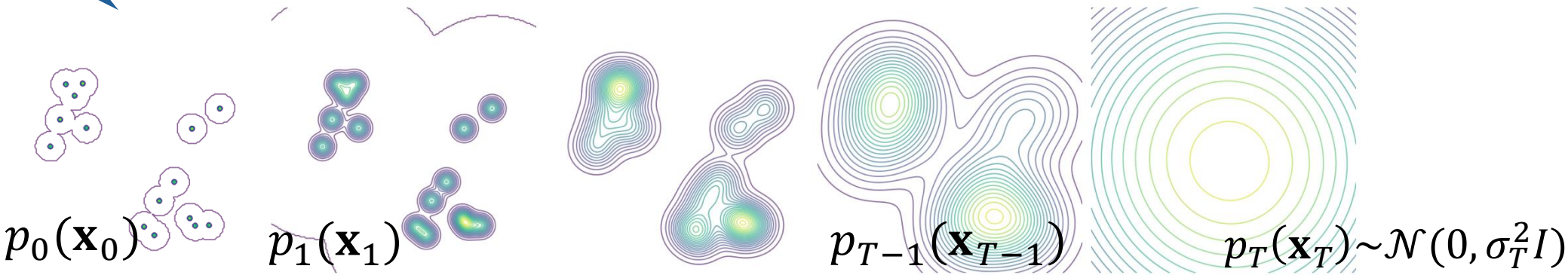
Denoising Score Matching (DSM)

How to learn this process?

Sampling: linking data and noise

- **Noising process**
  - Turn data $p(\mathbf{x}_0)$ into noise

Clean sample $\mathbf{x}_0$ $\mathbf{x}_1$ $\mathbf{x}_{T-1}$ $\mathbf{x}_T$ *Pure noise*



Smoothing the data density with levels of noise.

- **Denoising generation process**
  - Turn noise $p_T(\mathbf{x}_T)$ into data

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

Forward process
Adding noise and diffusion equation

Dynamic view

Evolution of particle
Brownian motion (SDE)

$$d\mathbf{x}_t = d\boldsymbol{W}_t$$

Evolution of density
Diffusion eq. (PDE)

$$\partial_t p(\mathbf{x}; t) = \frac{1}{2}\nabla^2 p(\mathbf{x}; t)$$

Static view

Adding noise to sample

$$\mathbf{x}_t = \mathbf{x}_0 + \sqrt{t}\,\boldsymbol{z}$$

$$\boldsymbol{z} \sim \mathcal{N}(0, I_{\boldsymbol{d}})$$

$$\mathbf{x}_0 \sim p_0$$

$$p(\mathbf{x}; t) = p(\mathbf{x}; 0) * \mathcal{N}(0, t\, I_d) \quad \longleftrightarrow \quad p(\mathbf{x}_t) = p(\mathbf{x}_0) * \mathcal{N}(0, t\, I_d)$$

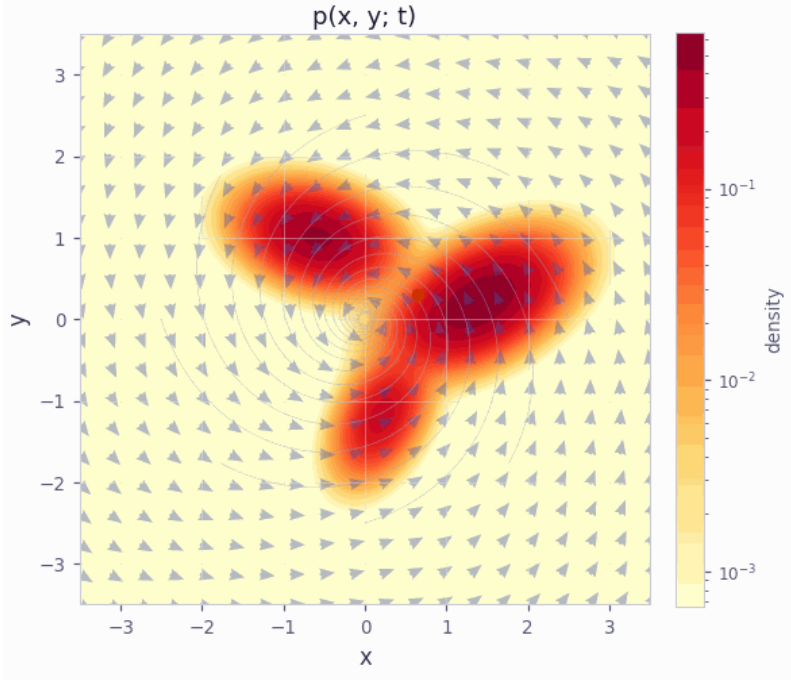
How to reverse a noising process?

Probability flow and Liouville Equation

Sample evolution (ODE/SDE) $d\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t; t)\, dt$

Density evolution (PDE)
Liouville Eq. $\dfrac{\partial p(\mathbf{x}; t)}{\partial t} = -\nabla \cdot (\boldsymbol{f}(\mathbf{x}; t)\, p(\mathbf{x}; t))$

Evolution of density $p(x, y; t)$ under a vector field



Derivation

Reverse noising process via score function

Sample evolution (ODE)

$$d\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t; t)\, dt$$

$$\boldsymbol{f}(\mathbf{x}; t) := \frac{1}{2} \nabla \log p(\mathbf{x}; \mathrm{t})$$

Score function

Density evolution (PDE)

$$\frac{\partial p(\mathbf{x}; t)}{\partial t} = -\nabla \cdot (\boldsymbol{f}(\mathbf{x}; t)\, p(\mathbf{x}; t))$$

$$= -\nabla \cdot \left(\frac{1}{2} \nabla \log p(\mathbf{x}; \mathrm{t})\, p(\mathbf{x}; t) \right)$$

$$= -\frac{1}{2} \nabla \cdot \left(\frac{\nabla p(\mathbf{x}; \mathrm{t})}{p(\mathbf{x}; t)}\, p(\mathbf{x}; t) \right)$$

$$\partial_t p(\mathbf{x}; t) = -\frac{1}{2} \nabla^2\, p(\mathbf{x}; t)$$

Equivalent to time reversal $t \to -t$

Diffusion / Brownian motion

$$d\mathbf{x}_t = d\boldsymbol{W}_t$$

$$\partial_t p(\mathbf{x}; t) = \frac{1}{2} \nabla^2 p(\mathbf{x}; t)$$

Probability flow and Liouville & Fokker Planck Equation

Sample evolution (ODE/SDE) $d\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t; t)\, dt$

Density evolution (PDE)
Liouville Eq. $\dfrac{\partial p(\mathbf{x}; t)}{\partial t} = -\nabla \cdot (\boldsymbol{f}(\mathbf{x}; t)\, p(\mathbf{x}; t))$

Sample evolution (SDE) $d\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t; t)\, dt + g(t) dW_t$ $dW_t \sim \mathcal{N}(0, dt\, I_d)$

Injecting white noise into the dynamics

Density evolution (PDE)
Fokker Planck Eq. $\dfrac{\partial p(\mathbf{x}; t)}{\partial t} = -\nabla \cdot \left(\boldsymbol{f}(\mathbf{x}; t)\, p(\mathbf{x}; t)\right) + \dfrac{1}{2} g^2(t) \nabla^2 p(\mathbf{x}; t)$

Derivation

Reverse noising process via score function (SDE)

Sample evolution (SDE)

$$d\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t; t)\, dt + g(t) dW_t$$

Density evolution (PDE)

$$\frac{\partial p(\mathbf{x}; t)}{\partial t} = -\nabla \cdot (\boldsymbol{f}(\mathbf{x}; t)\, p(\mathbf{x}; t)) + \frac{1}{2} g^2(t) \nabla^2 p(\mathbf{x}; t)$$

$$\boldsymbol{f}(\mathbf{x}; t) \coloneqq \frac{1}{2}(1 + g^2(t)) \nabla \log p(\mathbf{x}; \mathrm{t})$$

$$= -\frac{1}{2}\big(1 + g^2(t)\big)\nabla^2\, p(\mathbf{x}; t) + \frac{1}{2} g^2(t) \nabla^2 p(\mathbf{x}; t)$$

There is a parametric family $(g(t))$ of SDE that all reverse the noising process!

$$\partial_t p(\mathbf{x}; t) = -\frac{1}{2} \nabla^2\, p(\mathbf{x}; t)$$

Equivalent to time reversal $t \to -t$

Diffusion / Brownian motion

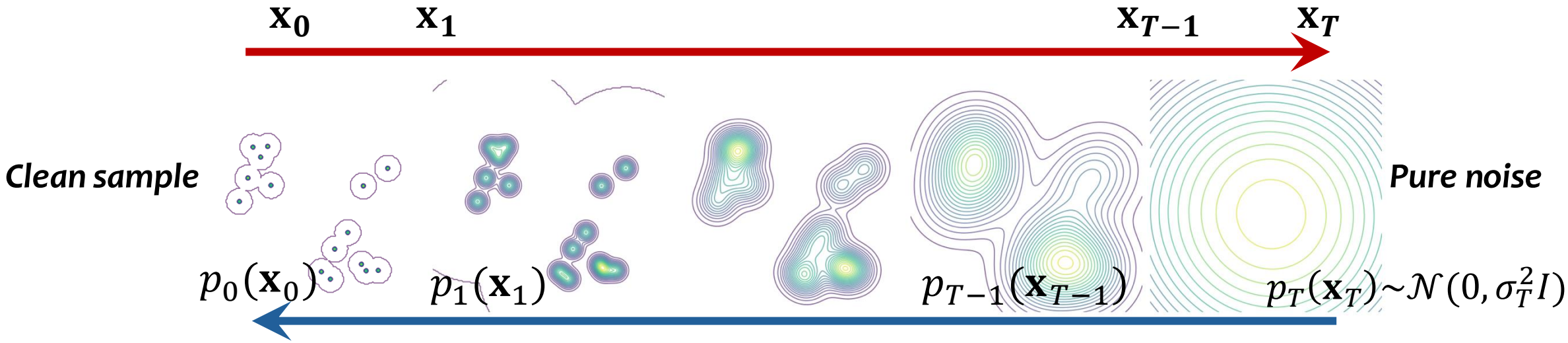
$$d\mathbf{x}_t = d\boldsymbol{W}_t$$

$$\partial_t p(\mathbf{x}; t) = \frac{1}{2} \nabla^2 p(\mathbf{x}; t)$$

Summary of sampling process

Noising process (Dynamic or Static)

$$d\mathbf{x}_t = d\boldsymbol{W}_t \qquad \mathbf{x}_t = \mathbf{x}_0 + \sqrt{t}\,\boldsymbol{z}$$



Key:

score function of each noise levels, enables sampling from the distribution.

Reverse denoising process (ODE & SDE)

$$d\mathbf{x}_t = -\frac{1}{2}\nabla \log p(\mathbf{x}; \mathrm{t})\, dt$$

$$d\mathbf{x}_t = -\frac{1}{2}(1 + g(t)^2)\nabla \log p(\mathbf{x}; \mathrm{t})\, dt + g(t)\, d\boldsymbol{W}_t$$

Training

How to obtain the score function?

$$\mathbf{s}(\mathbf{x}, \sigma) := \nabla_{\mathbf{x}} \log p(\mathbf{x}; \sigma)$$

$$p(\mathbf{x}; \sigma) = p_0(\mathbf{x}) * \mathcal{N}(0, \sigma^2 I_d)$$

Denoising and Tweedie's formula

$$\mathbf{x}_\sigma = \mathbf{x}_0 + \sigma\,\mathbf{z} \quad \mathbf{z} \sim \mathcal{N}(0, I_d)$$

Tweedie's formula $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_\sigma] = \mathbf{x}_\sigma + \sigma^2 \nabla \log p(\mathbf{x}_\sigma; \sigma)$

Expectation of clean sample, given the noised sample. (***Denoiser***)

$\mathbf{D}(\mathbf{x}_\sigma, \sigma)$

Noised sample

Score function

$\mathbf{s}(\mathbf{x}_\sigma, \sigma)$

Proof of Tweedie's formula

$$\nabla_{\mathbf{x}_\sigma} \log p(\mathbf{x}_\sigma; \sigma) = \frac{\nabla_{\mathbf{x}_\sigma} p(\mathbf{x}_\sigma)}{p(\mathbf{x}_\sigma)}$$

$$= \frac{\nabla_{\mathbf{x}_\sigma} \int p(\mathbf{x}_\sigma|\mathbf{x}_0) p(\mathbf{x}_0)\, d\mathbf{x}_0}{p(\mathbf{x}_\sigma)}$$

$$= \frac{\int \left(\nabla_{\mathbf{x}_\sigma} p(\mathbf{x}_\sigma|\mathbf{x}_0)\right) p(\mathbf{x}_0)\, d\mathbf{x}_0}{p(\mathbf{x}_\sigma)}$$

$$\nabla_{\mathbf{x}_\sigma} p(\mathbf{x}_\sigma|\mathbf{x}_0)$$
$$= \nabla_{\mathbf{x}_\sigma} \mathcal{N}(\mathbf{x}_\sigma|\mathbf{x}_0, \sigma^2 I_d)$$
$$= \left(-\frac{\mathbf{x}_\sigma - \mathbf{x}_0}{\sigma^2}\right) \mathcal{N}(\mathbf{x}_\sigma|\mathbf{x}_0, \sigma^2 I_d)$$
$$= \left(-\frac{\mathbf{x}_\sigma - \mathbf{x}_0}{\sigma^2}\right) p(\mathbf{x}_\sigma|\mathbf{x}_0)$$

$$= \frac{\int \left(-\frac{\mathbf{x}_\sigma - \mathbf{x}_0}{\sigma^2}\right) p(\mathbf{x}_\sigma|\mathbf{x}_0) p(\mathbf{x}_0)\, d\mathbf{x}_0}{p(\mathbf{x}_\sigma)}$$

$$= -\frac{\mathbf{x}_\sigma \int p(\mathbf{x}_\sigma|\mathbf{x}_0) p(\mathbf{x}_0)\, d\mathbf{x}_0}{\sigma^2 p(\mathbf{x}_\sigma)} + \frac{\int \mathbf{x}_0 p(\mathbf{x}_\sigma|\mathbf{x}_0) p(\mathbf{x}_0)\, d\mathbf{x}_0}{\sigma^2 p(\mathbf{x}_\sigma)}$$

$$= -\frac{\mathbf{x}_\sigma p(\mathbf{x}_\sigma)}{\sigma^2 p(\mathbf{x}_\sigma)} + \frac{\int \mathbf{x}_0 p(\mathbf{x}_0|\mathbf{x}_\sigma) d\mathbf{x}_0}{\sigma^2}$$

$$= -\frac{\mathbf{x}_\sigma}{\sigma^2} + \frac{1}{\sigma^2} \mathbb{E}[\mathbf{x}_0|\mathbf{x}_\sigma]$$

Denoising score matching

Minimize MSE to estimate denoiser

General Lemma $\mathcal{L} = \mathbb{E}_{p(X,Y)} \|\boldsymbol{f}(X) - Y\|^2$

$$\min \mathcal{L} \Longrightarrow \boldsymbol{f}^*(X) = \mathbb{E}[X|Y]$$

$$\mathcal{L}_\sigma = \mathbb{E}_{p(\mathbf{x}_0, \mathbf{x}_\sigma)} \|\mathbf{D}_\theta(\mathbf{x}_\sigma; \sigma) - \mathbf{x}_0\|^2$$

Denoising score matching loss

$$\mathcal{L}_\sigma = \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \|\mathbf{D}_\theta(\mathbf{x}_0 + \sigma \mathbf{z}; \sigma) - \mathbf{x}_0\|^2$$

Minimum MSE denoiser is the conditional mean

$$\mathbf{D}^*(\mathbf{x}_\sigma; \sigma) = \mathbb{E}[\mathbf{x}_0 | \mathbf{x}_\sigma]$$

Translate the denoiser to score with Tweedie's

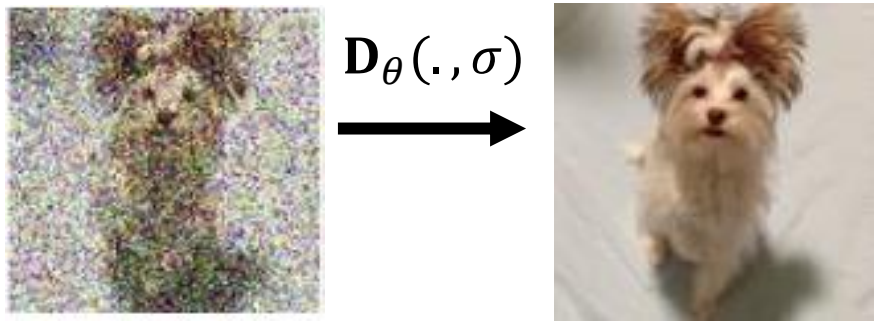
$$\mathbf{s}(\mathbf{x}_\sigma, \sigma) = \frac{\mathbf{D}^*(\mathbf{x}_\sigma; \sigma) - \mathbf{x}_\sigma}{\sigma^2}$$

Summary: Denoising score matching

Learning to denoise $\Leftrightarrow$ Smooth the density $p(\mathbf{x};\sigma)$ and learning its gradient, i.e. **score function** $\nabla \log p(\mathbf{x};\sigma)$

$$\nabla \log p(\mathbf{x};\sigma) = \frac{\mathbf{D}^*(\mathbf{x},\sigma) - \mathbf{x}}{\sigma^2}$$

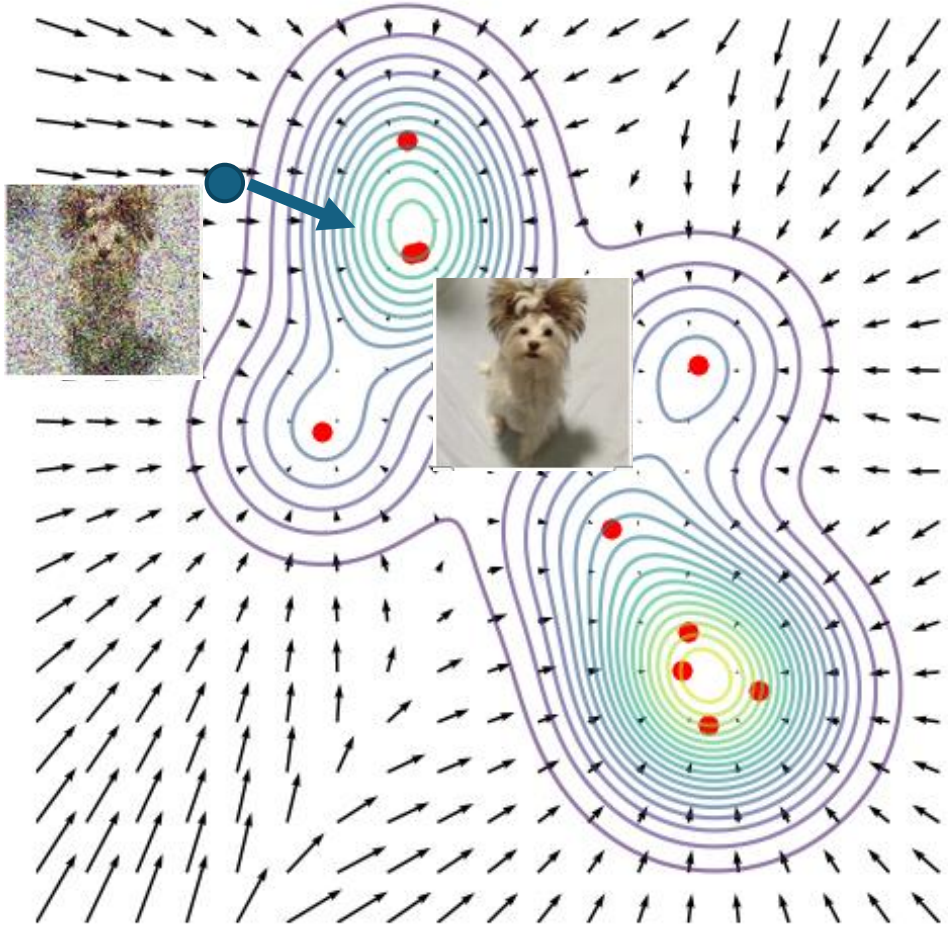
Tweedie's formula



Noisy sample $\mathbf{x}_0 + \sigma\mathbf{z}$

Estimated Clean Sample $\mathbf{D}_\theta(\mathbf{x_0} + \sigma\mathbf{z}, \sigma)$

$$\arg\min \mathcal{L}_{DSM,\sigma} \Rightarrow \mathbf{D}^*$$



Robbins, 1956, An empirical Bayes approach to statistics.
Vincent, 2011, *Neural Computation*

Sampling Diffusion models

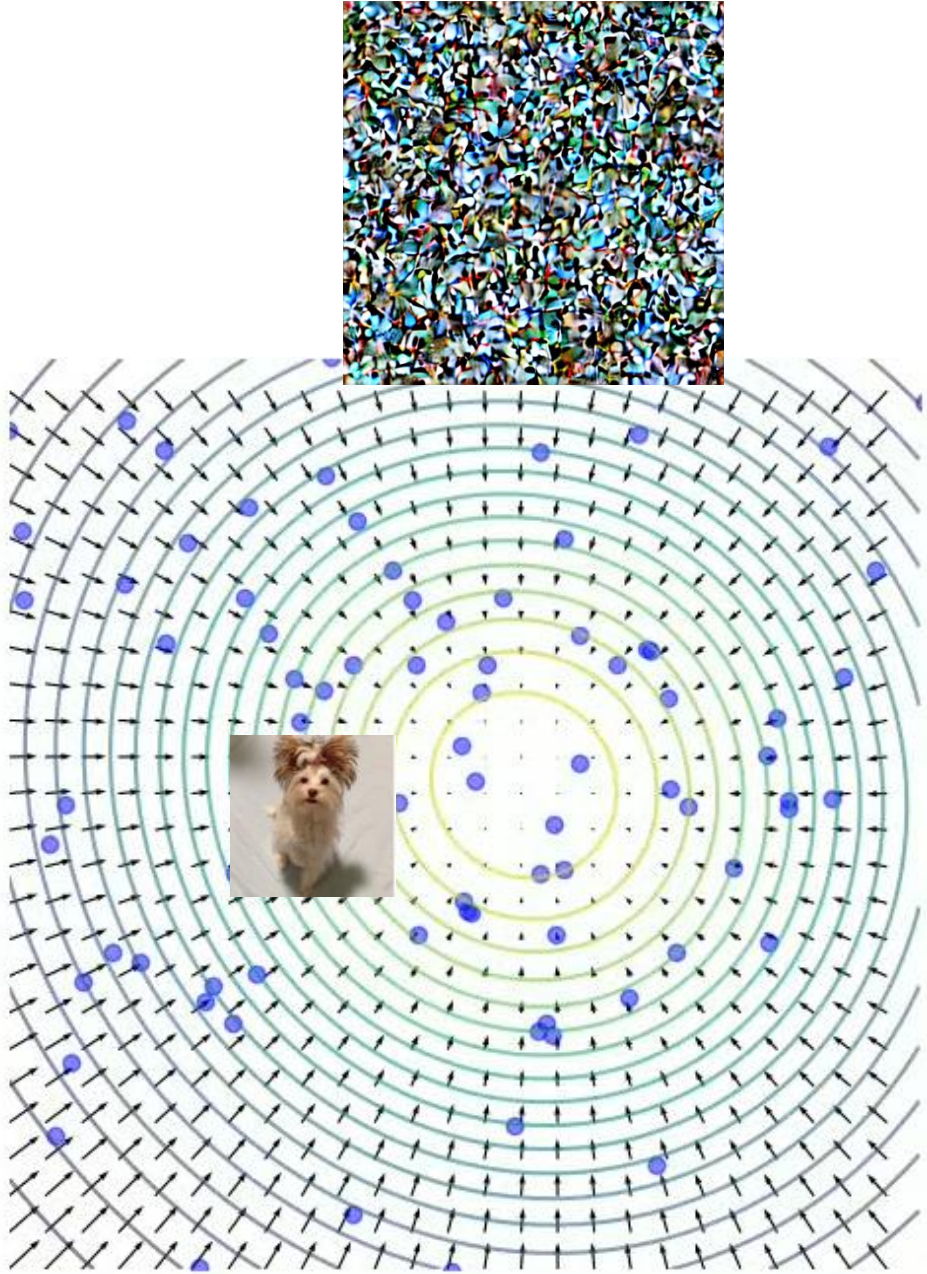
- Deterministic sampler: Probability-Flow Ordinary Differential Equation (PF-ODE)

Gradient of noised data distribution

$$\frac{d\mathbf{x}}{d\sigma} = -\sigma \nabla \log p(\mathbf{x}; \sigma)$$

$$\approx -\frac{\mathbf{D}_{\boldsymbol{\theta}}(\mathbf{x}, \sigma) - \mathbf{x}}{\sigma}$$

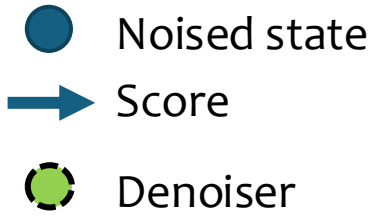
Neural network trained to denoise (DSM)



Initialize $\mathbf{x}_{\sigma_T} \sim \mathcal{N}(0, \sigma_T^2 I), \sigma_T \gg 1$
Integrate from $\sigma_T \to \sigma_0 \approx 0$

Karras, T., Aittala, M., Aila, T., & Laine, S. (2022), *NeurIPS*. EDM

Score and denoiser



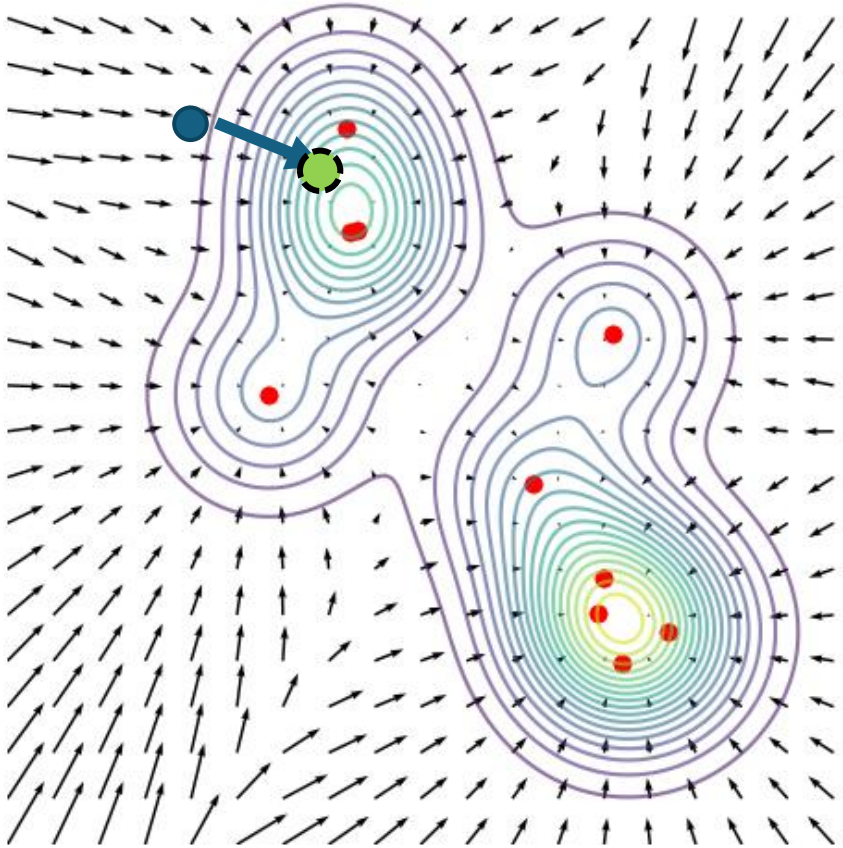
- Score

$$\mathbf{s}(\mathbf{x}, \sigma) \coloneqq \nabla \log p(\mathbf{x}; \sigma)$$

- Denoiser

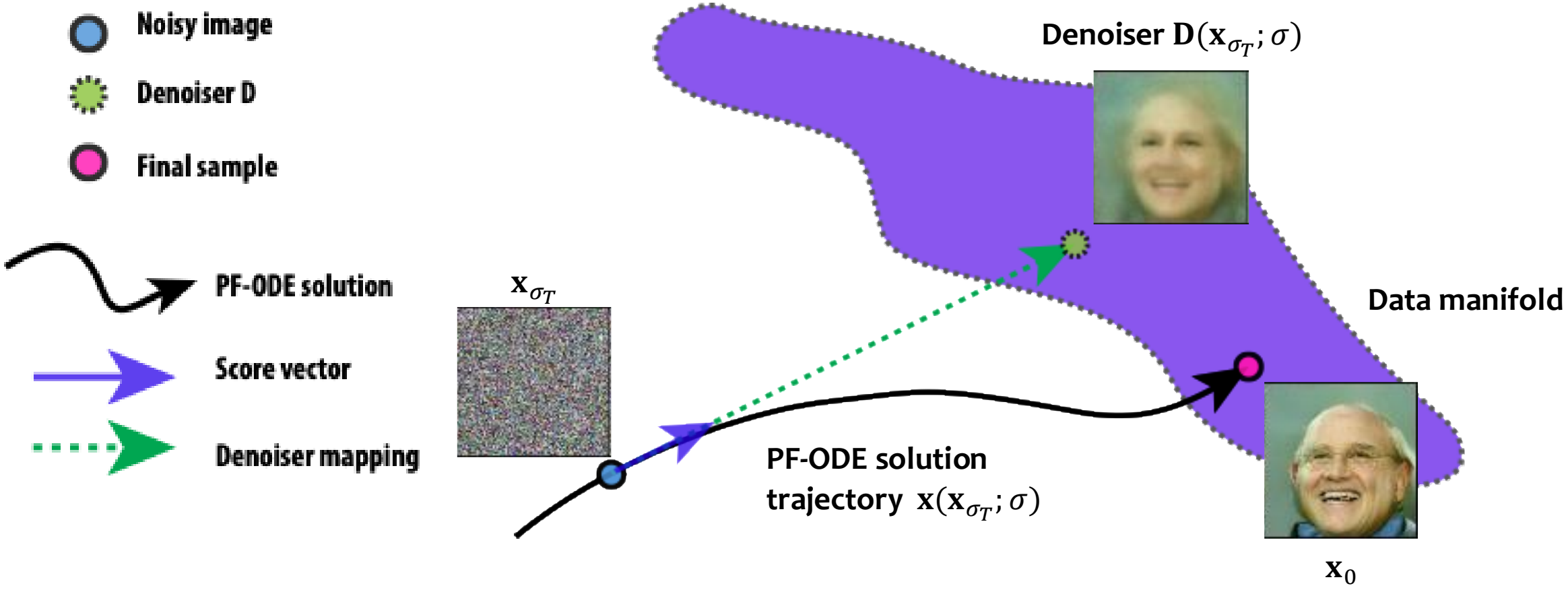
$$\mathbf{D}^*(\mathbf{x}, \sigma) \coloneqq \mathbb{E}[\mathbf{x}_0 | \mathbf{x}] = \mathbf{x} + \sigma^2 \nabla \log p(\mathbf{x}; \sigma)$$

Bayes optimal estimate of clean sample



Robbins, (1992) "An empirical Bayes approach to statistics."
Jonathan, Jain, Abbeel, (2020) NeurIPS, DDPM

Relation between Denoiser and Generated Sample



Denoiser as one-step look ahead of final sample.

Summary: Mathematical Foundation

- Noising process links data distribution and pure noise.
- One can reverse the noising process with score using PF-ODE or a family of SDE with equivalent density.
- One can learn the score function by minimizing denoising objective (DSM).
- Learning a distribution $\Leftrightarrow$ Learning to denoise.

Analytical intuition via linear cases

Linear Score ~ Gaussian statistics of data

Gaussian

$$p_{gauss}(\mathbf{x}) = \mathcal{N}(\mu, \Sigma)$$

Energy function is Quadratic

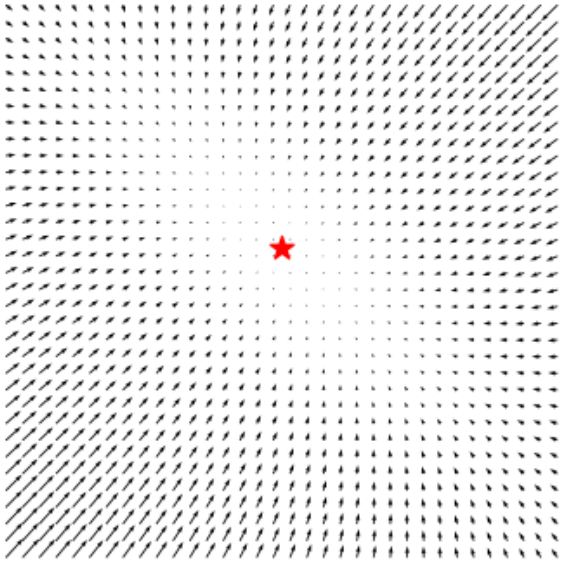
$$E_{gauss}(\mathbf{x}) \propto \frac{1}{2}(\mu - \mathbf{x})^{\top}\Sigma^{-1}(\mu - \mathbf{x})$$

pdf

Score function is Linear

$$\mathbf{s}_{gauss}(\mathbf{x}) = \Sigma^{-1}(\mu - \mathbf{x})$$

score



Gaussian score recovers Wiener filter

At noise scale σ, $p_{gauss}(\mathbf{x};\sigma) = \mathcal{N}(\mu, \Sigma + \sigma^2 I)$

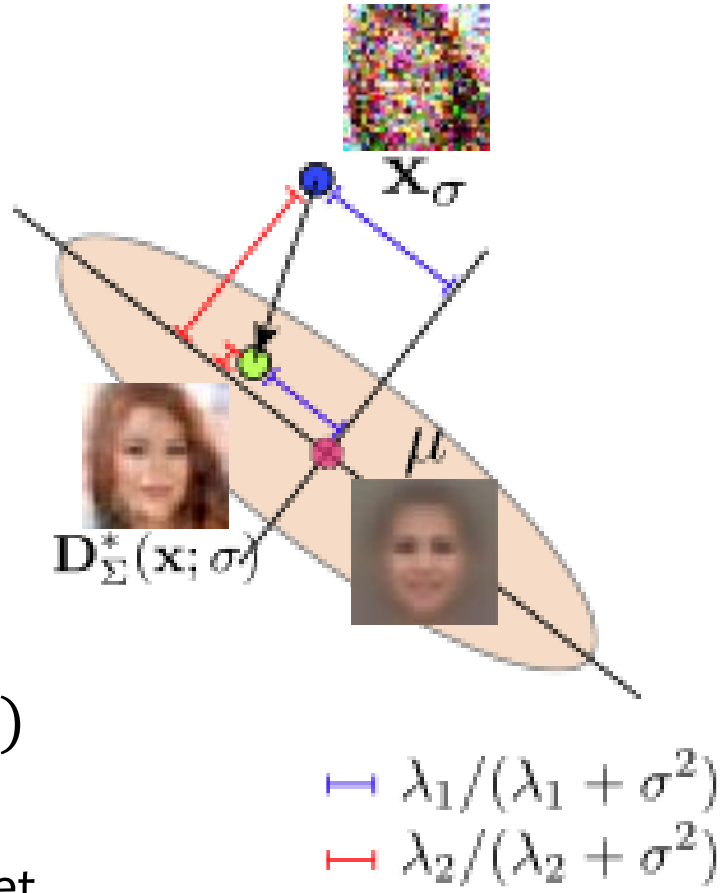
Score function $\mathbf{s}_{gauss}(\mathbf{x};\sigma) = (\Sigma + \sigma^2 I)^{-1}(\mu - \mathbf{x})$

Denoiser function $\mathbf{D}_{gauss}(\mathbf{x};\sigma) = \mu + \Sigma(\Sigma + \sigma^2 I)^{-1}(\mathbf{x} - \mu)$

Wiener filter (1967)

$$= \mu + \sum_k \frac{\lambda_k}{\lambda_k + \sigma^2} \mathbf{u}_k \mathbf{u}_k^\top (\mathbf{x} - \mu)$$

$\mathbf{u}_k$ PC of dataset
λ_k variance of PC



Theory

Gaussian score ⟺ Optimal under linear constraint

For arbitrary dataset,

Linear function approximator

$$\mathbf{D}_\theta(\mathbf{x};\sigma) = W_\sigma \mathbf{x} + b_\sigma$$

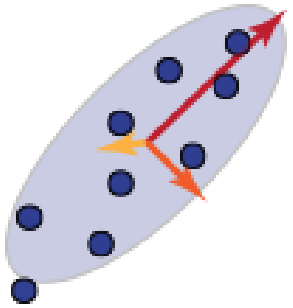
Optimize denoise score matching (DSM) loss

$$W_\sigma^*, b_\sigma^* = \arg\min_{W,b} \mathcal{L}_{DSM,\sigma}$$

$$\mathbf{D}^*(\mathbf{x};\sigma) = \boldsymbol{\mu} + \Sigma(\sigma^2 I + \Sigma)^{-1}(\mathbf{x} - \boldsymbol{\mu})$$

Wiener filter (1967)

From the blurry eye of linear "network", any data looks Gaussian.



Wang, Pehlevan, (2025), *NeurIPS spotlight*
Li, Dai, Qu, (2024). *NeurIPS*

Theory

Gaussian score admits closed-form solution to probability flow ODE

Score of Gaussian $\mathbf{s}_{gauss}$

$$\frac{d\mathbf{x}}{d\sigma} = -\frac{1}{\sigma}\sum_k \frac{\sigma^2}{\lambda_k + \sigma^2}\mathbf{u}_k\mathbf{u}_k^T(\mu - \mathbf{x})$$

Linear time-varying ODE

Dynamics matrices commute!

Sampling trajectory solution

$$\mathbf{x}(\sigma_t) = \mu + \sum_{\boldsymbol{k}} \sqrt{\frac{\sigma_t^2 + \lambda_k}{\sigma_T^2 + \lambda_k}}\mathbf{u}_k\mathbf{u}_k^T\left(\mathbf{x}_{\sigma_T} - \mu\right)$$

$\mathbf{u}_k$ PC of dataset
λ_k variance of PC

Wang, Vastola, TMLR, 2024
Pierret, Galerne, 2024

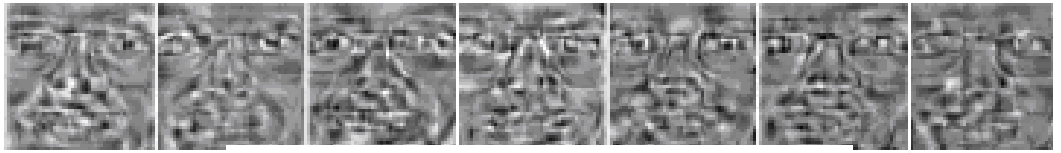
Statistical structure of natural image

$$\mathbf{x} \sim p_{face}$$

Distribution mean
represents average sample

Principal components
(PC) represent variations
of different levels / spatial
frequency

Higher variance PC



Lower variance PC

Burton, Moorhead (1987);
Field (1987, 1994);
Tolhurst et al. (1992)
Torralba, Oliva (2003)

Turk, Pentland (1991) Eigenface

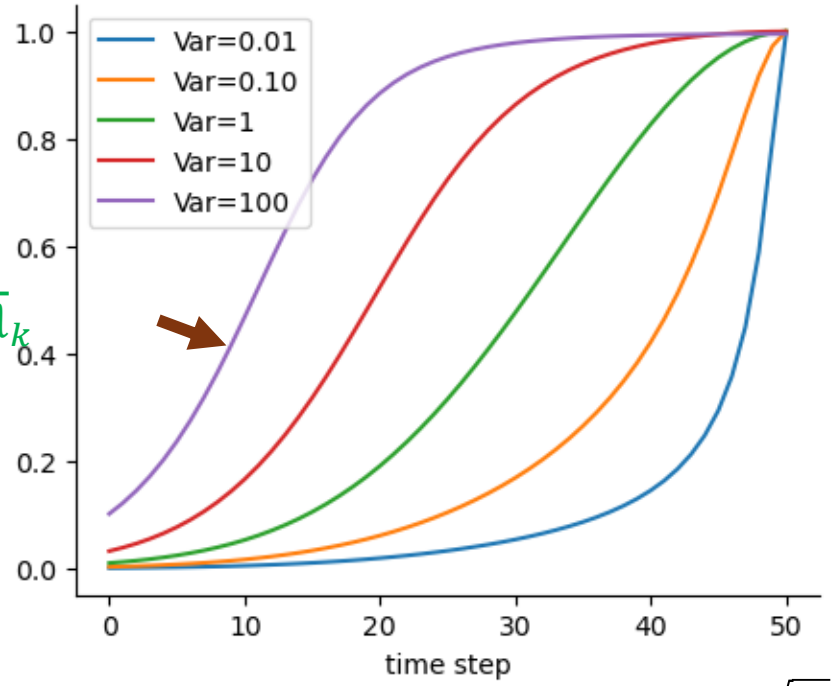
Gaussian theory predicts the spectral order of diffusion generation

$$\xi(t,\lambda) := \frac{\lambda}{\sqrt{(\sigma_t^2+\lambda)(\sigma_T^2+\lambda)}}$$

Dynamics of denoiser:

$$\mathbf{D}_{\text{gauss}}(\mathbf{x}_{\sigma_t};\sigma_t) = \underbrace{\mu}_{\text{Data mean}} + \sum_{k=1}^{r} \underbrace{\xi(t,\lambda_k)\mathbf{u}_k\mathbf{u}_k^T(\mathbf{x}_{\sigma_T}-\mu)}_{\text{Scaling of PC features}}$$

$\xi(t,\lambda)\sigma_T\ /\sqrt{\lambda_k}$



Normalized by $\sqrt{\lambda_k}/\sigma_T$

Starting from "Average" sample

Specifying low frequency info (hairstyle, face orientation)

Adding high frequency finer details.

Denoiser $\mathbf{D}(\mathbf{x}_t;\sigma_t)$

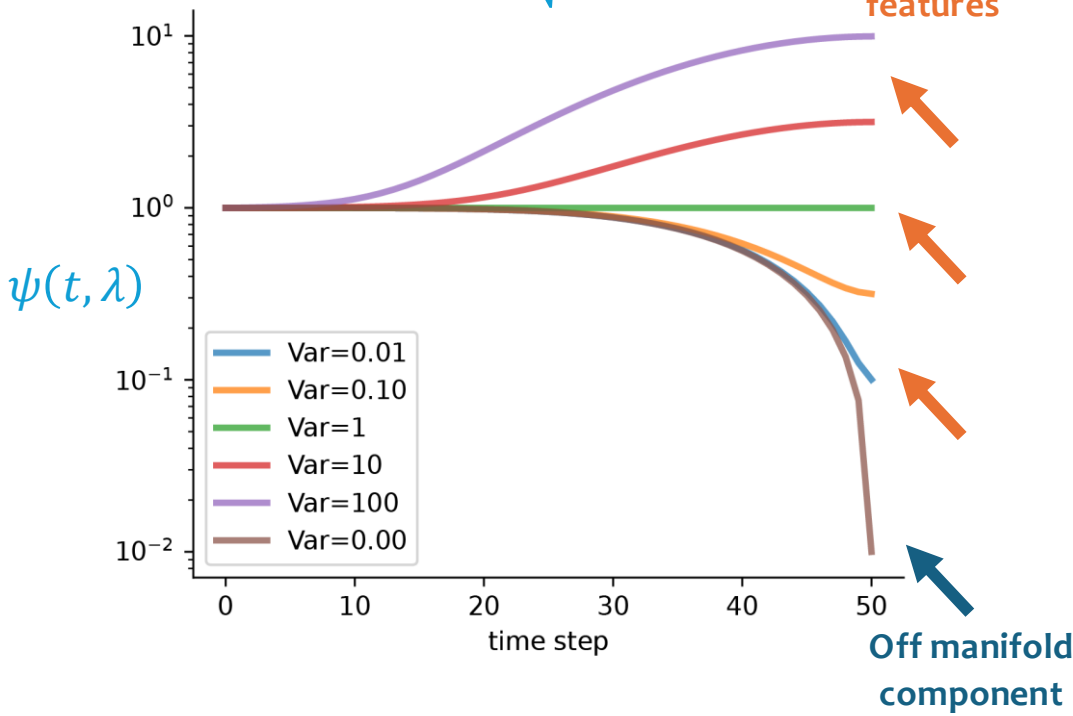
Gaussian analytical solution of $\mathbf{x}_t$ predicts evolution of noisy states

Dynamics of state (VP/ DDIM):

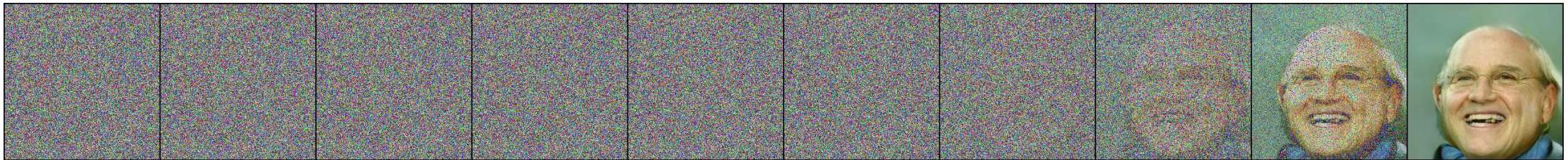
$$\mathbf{x}_t = \underbrace{\alpha_t \boldsymbol{\mu}}_{\text{Scaling mean}} + \underbrace{\psi(t,0)\mathbf{x}_T^{\perp}}_{\text{Off-manifold component}} + \underbrace{\sum_{k=1}^{r} \psi(t,\lambda_k) c_k(T)\mathbf{u}_k}_{\text{On-manifold feature components}}$$

$$\mathbf{x}_T^{\perp} \coloneqq (I - U^T U)(\mathbf{x}_T - \alpha_T \mu) \qquad c_k(T) \coloneqq \mathbf{u}_k^T(\mathbf{x}_T - \alpha_T \mu)$$

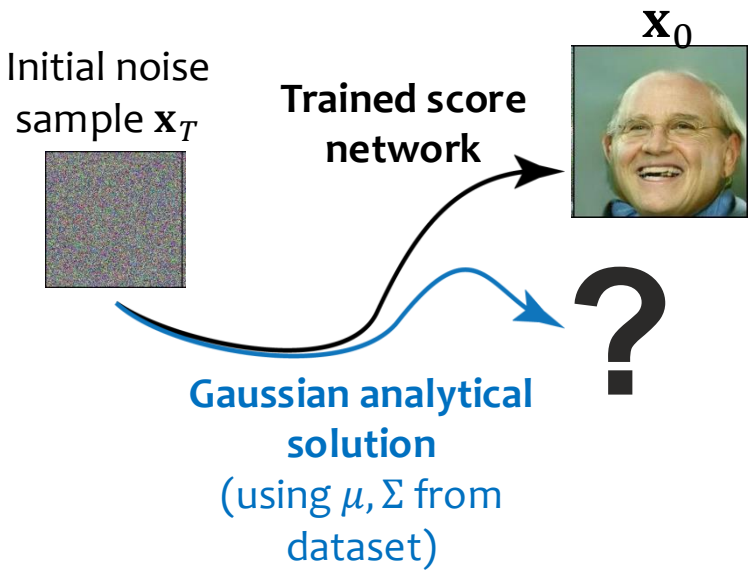
$$\psi(t,\lambda) \coloneqq \sqrt{\frac{\sigma_t^2 + \lambda\alpha_t^2}{\sigma_T^2 + \lambda\alpha_T^2}}$$

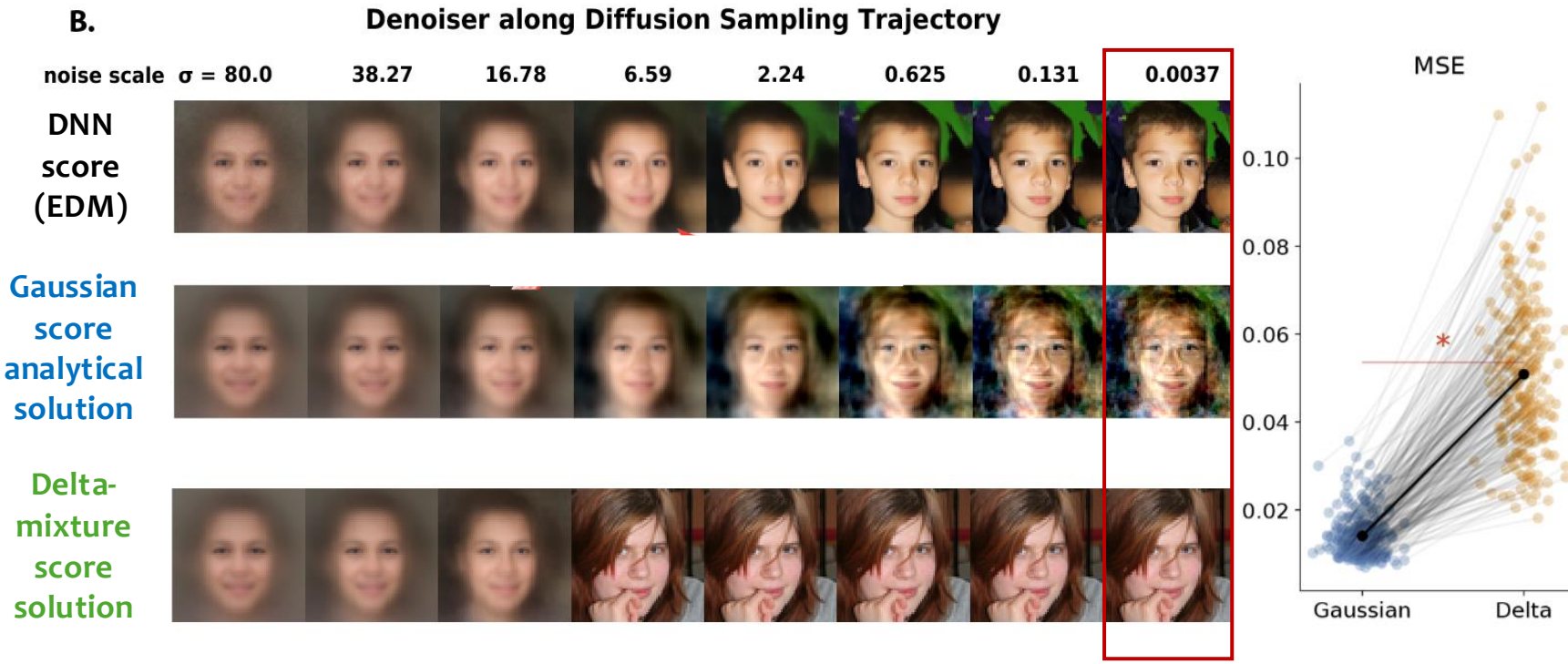


State $\mathbf{x}_t$



Gaussian linear score quantitatively approximates the high noise phase of pre-trained diffusion model and the final sample!



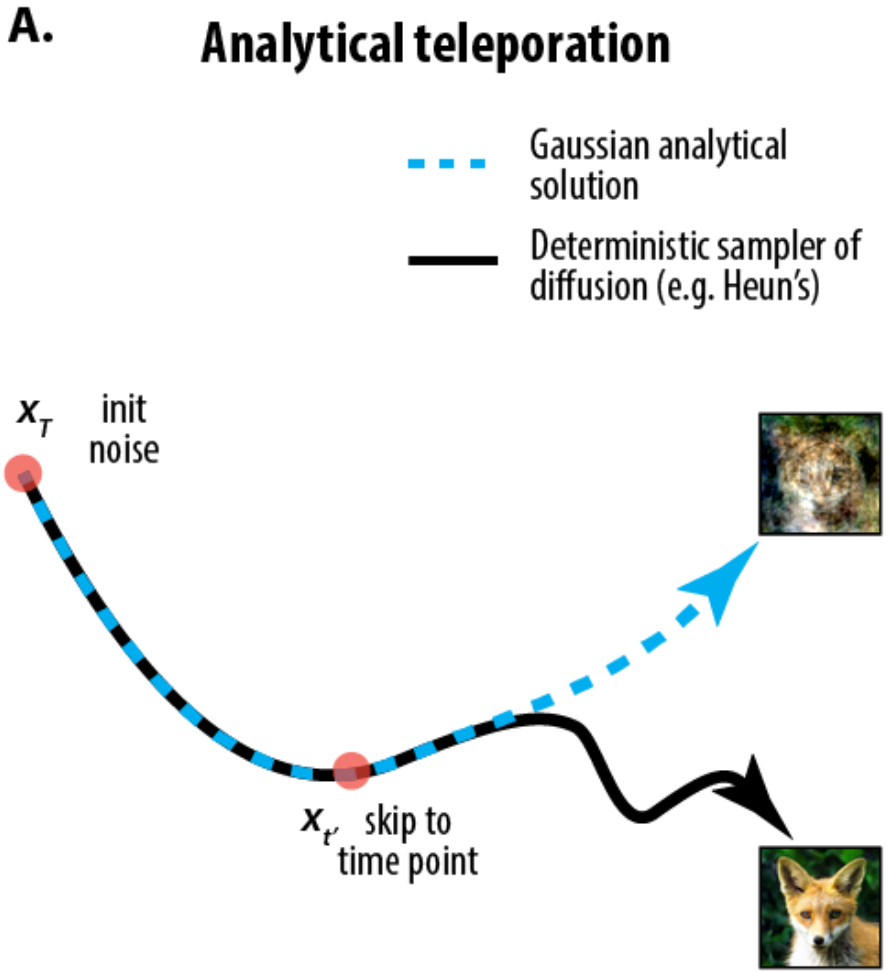


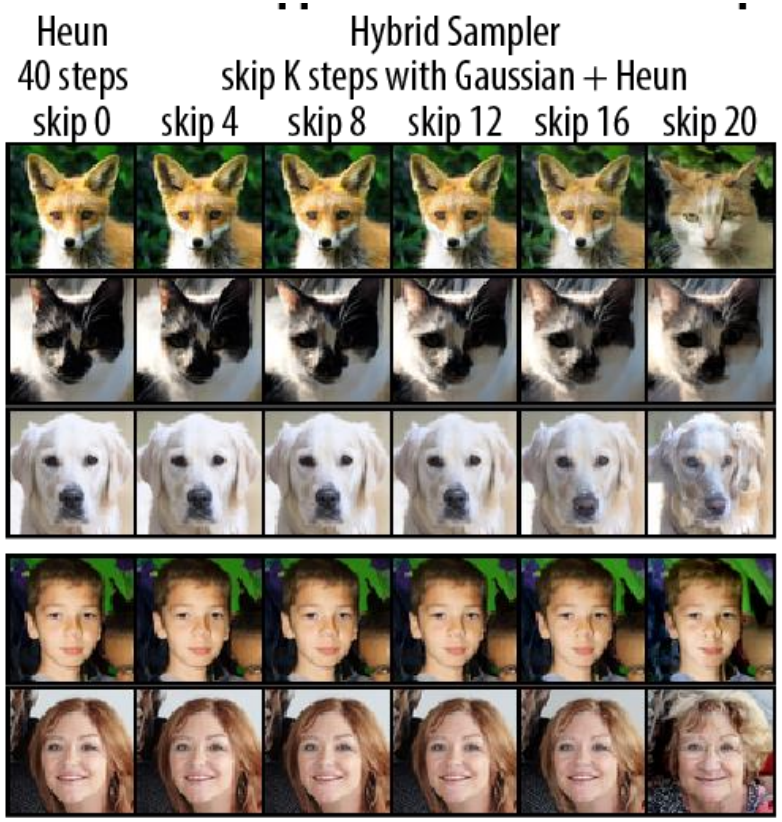
$$\mathbf{D}_{\text{gauss}}(\mathbf{x}_{\sigma_t}; \sigma_t) = \mu + \sum_{k=1}^{r} \xi(t, \lambda_k) \mathbf{u}_k \mathbf{u}_k^T (\mathbf{x}_{\sigma_T} - \mu)$$

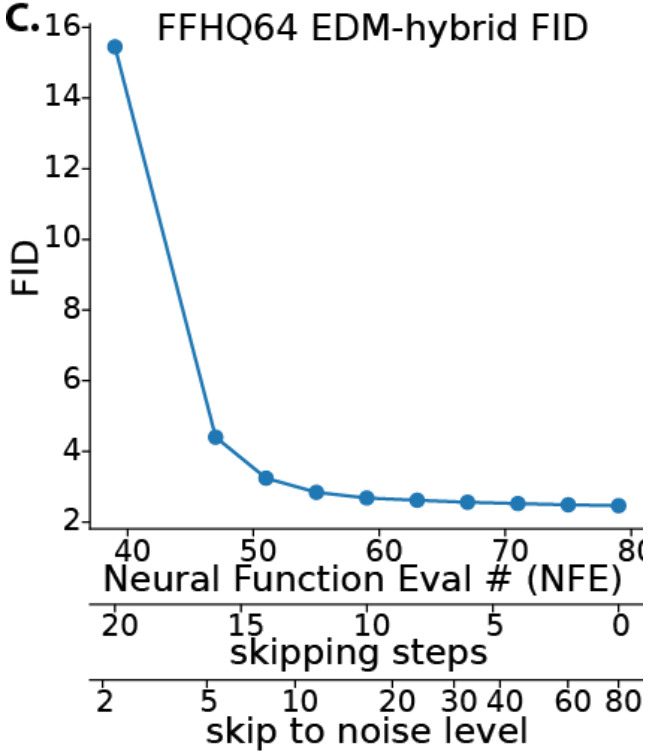
Wang, Vastola, TMLR, 2024

Application

Gaussian solution enables analytical teleportation to accelerate sampling



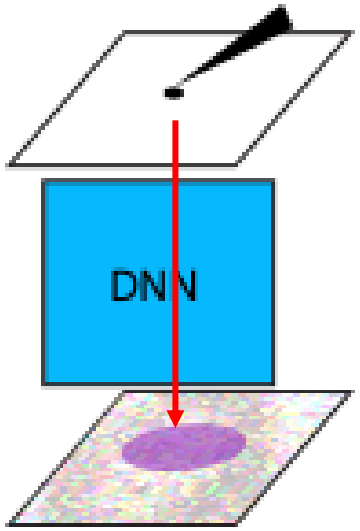




Wang, Vastola, TMLR, 2024

Theory:
Receptive field structure in linear denoiser

Receptive field of output pixel $\nabla_{\mathbf{x}}[\boldsymbol{e}_{i,j}^{\top}\mathbf{D}(\mathbf{x};\sigma)]$



$$\mathbf{D}^*(\mathbf{x};\sigma) = \boldsymbol{\mu} + \Sigma(\sigma^2 I + \Sigma)^{-1}(\mathbf{x} - \boldsymbol{\mu})$$

$$\nabla_{\mathbf{x}}\mathbf{D}^*(\mathbf{x};\sigma) = \Sigma(\sigma^2 I + \Sigma)^{-1}$$

Receptive field. $\mathrm{RF}_{\mathrm{ij}} = \mathbf{e}_{i,j}^{\top}\,\Sigma(\sigma^2 I + \Sigma)^{-1}$

Dependency on noise scale

Dependency on spatial probe point

Low noise limit
$\sigma \to 0$

$$\mathrm{RF}_{\mathrm{ij}} = \mathbf{e}_{i,j}^{\top}\, I = \mathbf{e}_{i,j}$$

Local and equivariant!
Pixel itself

High noise limit
$\sigma \to \infty$

$$\mathrm{RF}_{\mathrm{ij}} = \mathbf{e}_{i,j}^{\top}\,\Sigma/\sigma^2$$

All correlated pixels,
Non-local not-equivariant

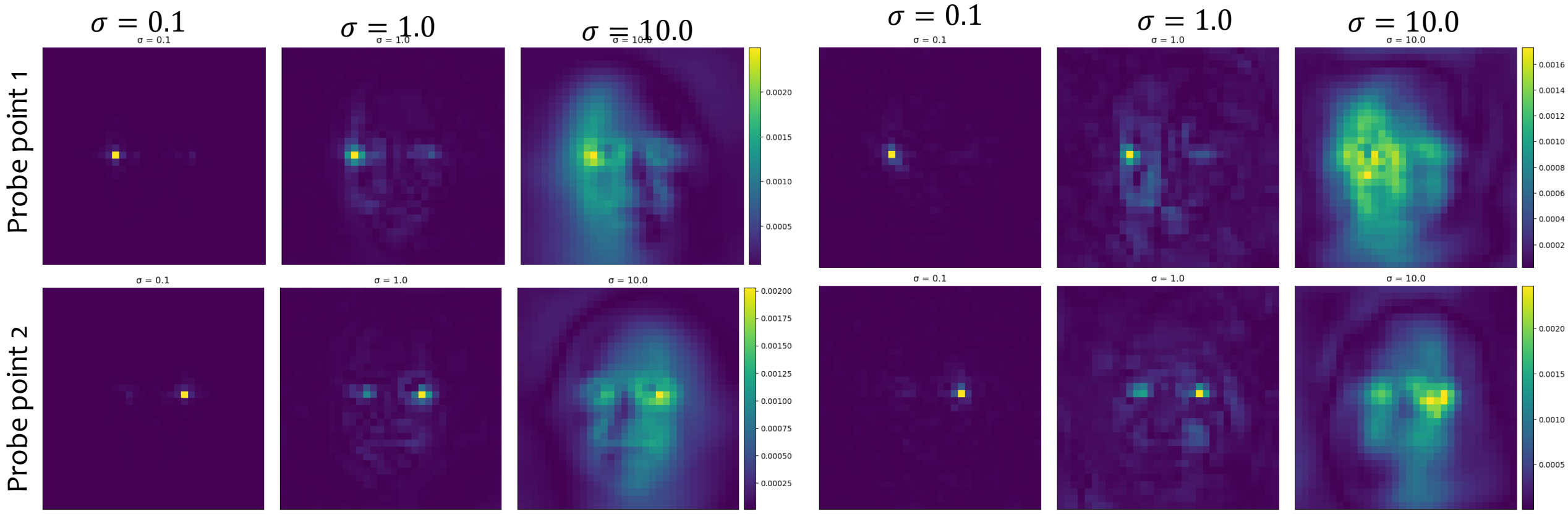
Empirical validation

Linear theory predicts the spatial and noise scale dependency of receptive field

When real data covariance (FFHQ) is used, there ***is non-local, non-equivariant structure*** *in learned receptive field.*

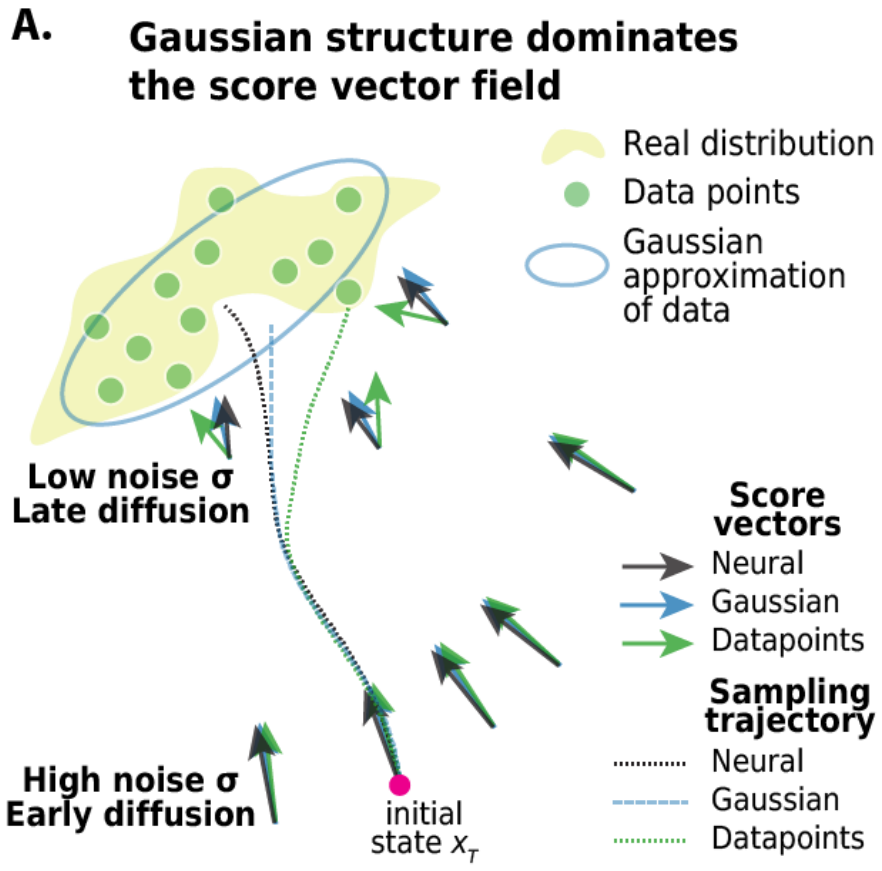
Linear Prediction of Receptive field

Empirical Receptive Field from Trained UNet



Summary of linear case

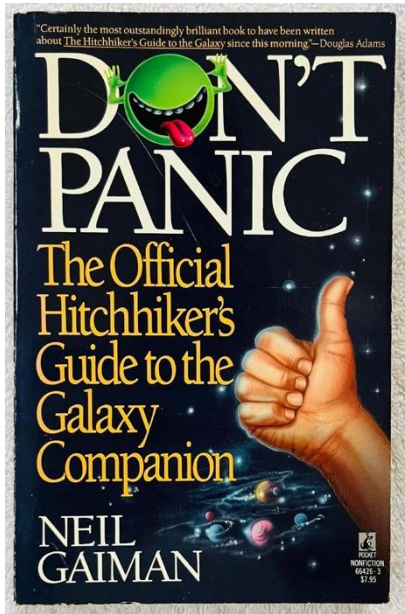
- Gaussian score recovers Wiener filter and admits close-form sampling trajectory.
- Trained diffusion score networks are very close to Gaussian linear score for a wide range of noise scales.
- Gaussian solution predicts the early generation trajectory and final samples, better than delta score solution.



Connections and Equivalences

Every diffusion paper uses different parametrization & notation

But don’t panic!



There are many formulations of diffusion

- But many formulations are
  - Equivalent per time reparametrization or space scaling.
  - Equivalent per same optimum on the loss.
  - Equivalent per discretization of time and variation in ODE/SDE solver.

Link to Variance-Preserving (VP) ODE

Space and time domains are infinite (variance explodes!)

$$\sigma: 0 \to \infty \qquad \mathbf{x}_\sigma \to \infty$$

- Space-time re-parametrization

$$\mathbf{x}_\sigma = \mathbf{x}_0 + \sigma \boldsymbol{z}$$

$$\frac{d\mathbf{x}}{d\sigma} = -\sigma \nabla \log p(\mathbf{x}; \sigma)$$

↓ Reparametrize noise with time t, $\sigma(t)$

$$\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \boldsymbol{z}$$

$$\frac{d\mathbf{x}}{dt} = -\dot{\sigma}_t \sigma_t \nabla \log p(\mathbf{x}; \sigma_t)$$

↓ Reparametrize space via scaling space

$$\tilde{\mathbf{x}}_t \coloneqq \alpha_t \mathbf{x}_t$$

$$\tilde{\mathbf{x}}_t = \alpha_t \mathbf{x}_0 + \alpha_t \sigma_t \boldsymbol{z}$$

$$\begin{aligned}\frac{d\tilde{\mathbf{x}}_t}{dt} &= \frac{d}{dt}(\alpha_t \mathbf{x}_t) = \dot{\alpha}_t \mathbf{x}_t + \alpha_t \dot{\mathbf{x}}_t \\ &= \dot{\alpha}_t \mathbf{x}_t - \alpha_t \dot{\sigma}_t \sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t; \sigma_t) \\ &= \frac{\dot{\alpha}_t}{\alpha_t} \tilde{\mathbf{x}}_t - \alpha_t^2 \dot{\sigma}_t \sigma_t \nabla_{\tilde{\mathbf{x}}_t} \log p\left(\frac{\tilde{\mathbf{x}}_t}{\alpha_t}; \sigma_t\right)\end{aligned}$$

Link to Variance-Preserving (VP) ODE

$$\tilde{\mathbf{x}}_t = \alpha_t \mathbf{x}_0 + \tilde{\sigma}_t \boldsymbol{z}$$

$$\begin{aligned}
\frac{d\tilde{\mathbf{x}}_t}{dt} &= \frac{d}{dt}(\alpha_t \mathbf{x}_t) = \dot{\alpha}_t \mathbf{x}_t + \alpha_t \dot{\mathbf{x}}_t \\
&= \dot{\alpha}_t \mathbf{x}_t - \alpha_t \dot{\sigma}_t \sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t; \sigma_t) \\
&= \frac{\dot{\alpha}_t}{\alpha_t} \tilde{\mathbf{x}}_t - \alpha_t^2 \dot{\sigma}_t \sigma_t \nabla_{\tilde{\mathbf{x}}_t} \log p\left(\frac{\tilde{\mathbf{x}}_t}{\alpha_t}; \sigma_t\right) \\
&= \frac{\dot{\alpha}_t}{\alpha_t} \tilde{\mathbf{x}}_t - \frac{d}{dt}\left(\frac{\tilde{\sigma}_t}{\alpha_t}\right) \alpha_t \tilde{\sigma}_t \nabla_{\tilde{\mathbf{x}}_t} \log p\left(\frac{\tilde{\mathbf{x}}_t}{\alpha_t}; \frac{\tilde{\sigma}_t}{\alpha_t}\right) \\
&= \frac{\dot{\alpha}_t}{\alpha_t} \tilde{\mathbf{x}}_t - \left(\dot{\tilde{\sigma}}_t \tilde{\sigma}_t - \frac{\tilde{\sigma}_t^2 \dot{\alpha}_t}{\alpha_t}\right) \nabla_{\tilde{\mathbf{x}}_t} \log p\left(\frac{\tilde{\mathbf{x}}_t}{\alpha_t}; \frac{\tilde{\sigma}_t}{\alpha_t}\right)
\end{aligned}$$

Redefine the noise scale
$\tilde{\sigma}_t := \alpha_t \sigma_t$

Popular choice : Variance preserving
$\alpha_t^2 + \tilde{\sigma}_t^2 = 1$

Link to other forms of diffusion-loss: $\mathbf{x_0}$, $\boldsymbol{\epsilon}$, $\mathbf{v}$

Recall lemma, optimum of MSE loss are conditional expectation.

Optima are equivalent up to scaling and skip connection.

$$\mathcal{L}_\sigma = \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \|\mathbf{D}_\theta(\mathbf{x}_0 + \sigma \boldsymbol{z}; \sigma) - \mathbf{x}_0\|^2 \qquad \mathbf{D}_\theta^* = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_\sigma]$$

$\mathbf{x_0}$-prediction $\quad \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \|\mathbf{x}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{z}; t) - \mathbf{x}_0\|^2 \qquad \mathbf{x}_\theta^* = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$

$\boldsymbol{\epsilon}$-prediction $\quad \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \|\boldsymbol{\epsilon}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{z}; t) - \boldsymbol{z}\|^2$

$$\begin{aligned} \boldsymbol{\epsilon}_\theta^* &= \mathbb{E}[\boldsymbol{z}|\mathbf{x}_t] \\ &= \frac{\mathbf{x}_t - \alpha_t \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]}{\sigma_t} \end{aligned}$$

$\mathbf{v}$-prediction $\quad \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \|\mathbf{v}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{z}; t) - (\alpha_t \boldsymbol{z} - \sigma_t \mathbf{x}_0)\|^2$

$$\begin{aligned} \mathbf{v}_\theta^* &= \mathbb{E}[\alpha_t \boldsymbol{z} - \sigma_t \mathbf{x}_0|\mathbf{x}_t] \\ &= \frac{\alpha_t \mathbf{x}_t - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]}{\sigma_t} \end{aligned}$$

$$\propto \frac{d}{dt}(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{z})$$

As $\alpha_t^2 + \sigma_t^2 = 1$

Link to Flow Matching / Rectified flow

- Flow matching / Rectified flow objective

$$\mathbb{E}_{\substack{\mathbf{z}\sim\mathcal{N}(0,I_d)\\ \mathbf{x_0}\sim p_0}} \left\|\mathbf{u}_\theta\big((1-t)\mathbf{x}_0 + t\mathbf{z}; t\big) - (\mathbf{z}-\mathbf{x}_0)\right\|^2$$

$$\mathbf{u}_\theta^* = \mathbb{E}[\mathbf{z}-\mathbf{x}_0|\mathbf{x}_t] = \frac{\mathbf{x}_t - \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]}{t}$$

- Sampling equation

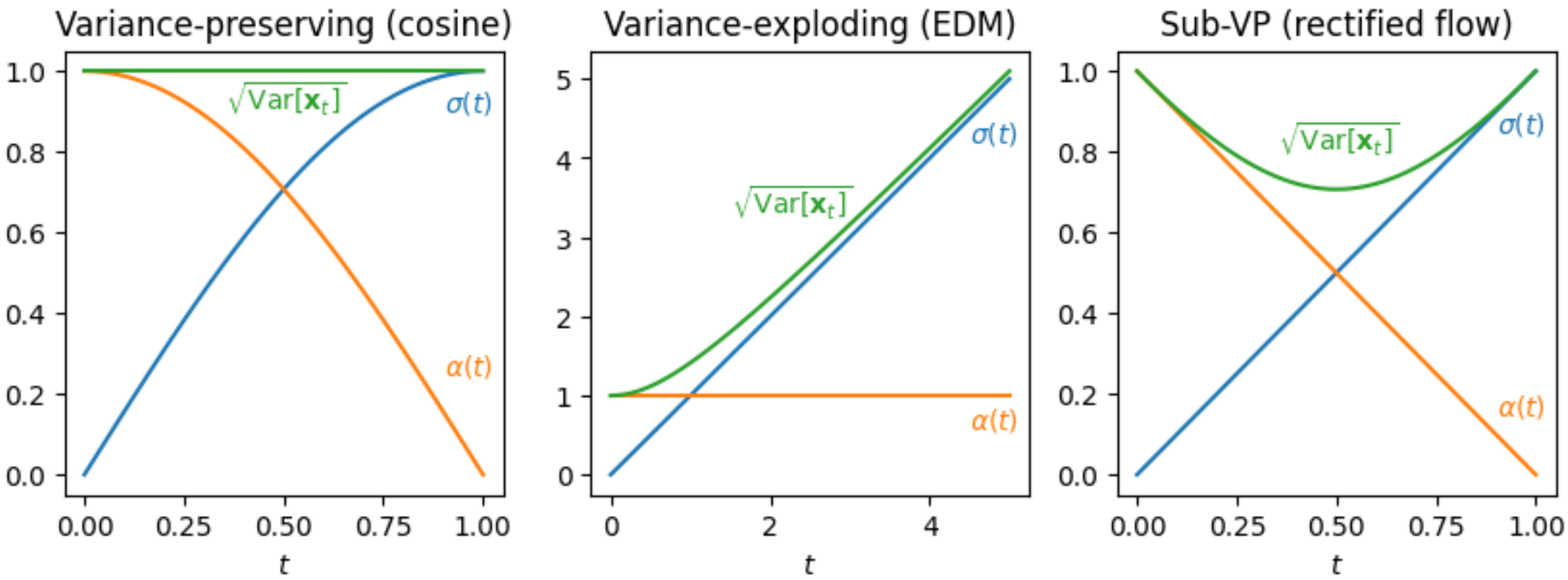
$$\frac{d\mathbf{x}}{dt} = \mathbf{u}_\theta(\mathbf{x}; t) \qquad t: 1 \to 0$$

- Equivalent to diffusion v-prediction loss with specific parametrization
  - $\alpha_t = 1-t, \sigma_t = t$

$$\mathbb{E}_{\substack{\mathbf{x}_0\sim p_0\\ z\sim\mathcal{N}(0,I)}} \|\mathbf{v}_\theta(\underbrace{\alpha_t}_{1-t}\mathbf{x}_0 + \underbrace{\sigma_t}_{t}\boldsymbol{z}; t) - (\underbrace{\dot{\alpha}_t}_{-1}\mathbf{x}_0 + \underbrace{\dot{\sigma}_t}_{1}\boldsymbol{z})\|^2$$

https://diffusionflow.github.io/

Some common noise and signal scale parametrization



https://sander.ai/2024/06/14/noise-schedules.html#adaptive

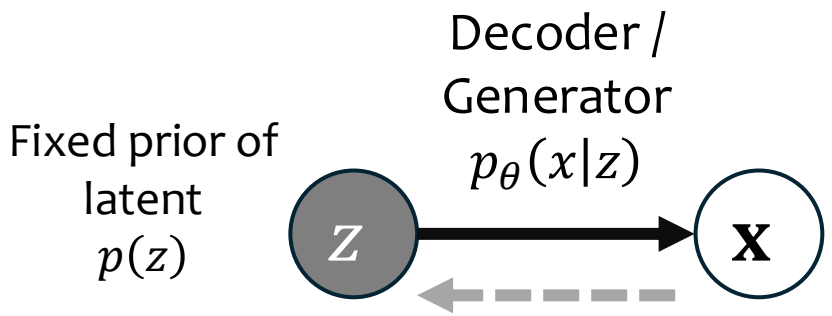
Take home message

- Regardless of the loss, the (optimal) learned network are all affine functions of the score $\mathbf{s}$ / denoiser $\mathbf{D}$.

- Thus, they can be used to drive various form score-based sampling equation (ODE / SDE).

- Training and sampling are disentangled and can be designed separately!

Recap of Variational Auto-Encoder



$$\log p_\theta(x) = \log \int p_\theta(x,z)dz \qquad \textit{Intractable!!!}$$

$$\log p_\theta(x) = \log \frac{p_\theta(x,z)}{p_\theta(z|x)}$$

$$= \mathbb{E}_{q(z)} \log \frac{p_\theta(x,z)\cdot q(z)}{q(z)\cdot p_\theta(z|x)}$$

$$= ELBO + D_{KL}(q(z)||p_\theta(z|x))$$

$$\geq ELBO$$

$$= \mathbb{E}_q[\log p_\theta(x|z)] - D_{KL}(q(z)||p(z))$$

- Generative model with latent variable.
- We want to compute the likelihood of data, but intractable.
- We introduce a variational distribution $q(z)$, to derive ELBO, and maximize ELBO to bound likelihood.
- Parametrize the distribution with a neural network $q_\phi(z|x)$.

Link to DDPM: (*Historical note*) Variational inference perspective

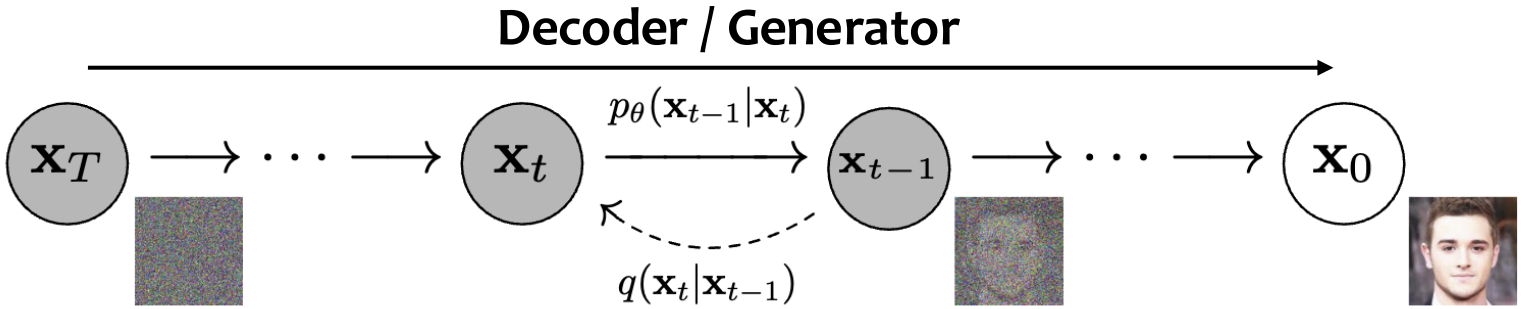


Figure 2: The directed graphical model considered in this work.

Variational distribution (Encoder): a fixed Gaussian noising process.

- Think of Diffusion model as a latent variable model, with a sequence of latents $\mathbf{x}_{T:1}$.
- Denoising (generation) process forms the decoder.
- Noising (forward) process forms the variational distribution, (i.e. encoder without parameter).

Ho et al. 2020 DDPM

Link to DDPM: (*Historical note*) Equivalence to score-based perspective

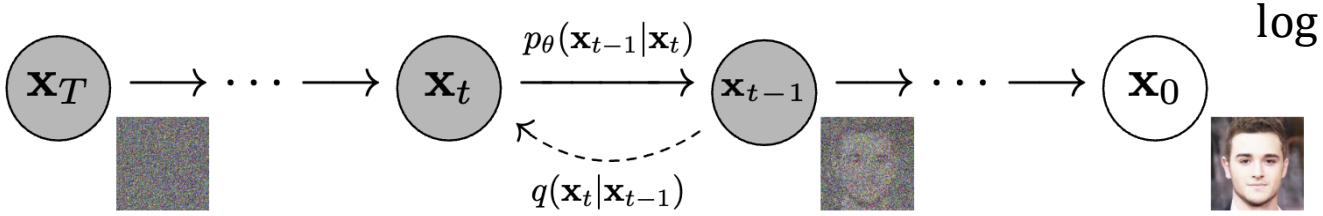


Figure 2: The directed graphical model considered in this work.

$$\log p_\theta(x_0) = \log \frac{p_\theta(x_0, x_{1:T})}{p_\theta(x_{1:T}|x_0)}$$

$$= \mathbb{E}_q \log \frac{p_\theta(x_0, x_{1:T}) \cdot q(x_{1:T}|x_0)}{q(x_{1:T}|x_0) \cdot p_\theta(x_{1:T}|x_0)}$$

$$\geq ELBO$$

$$= \sum_{t=2:T} \mathbb{E}_{q(x_t|x_0)}[D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))] + \cdots$$

- The posterior of latent trajectory given an image $p_\theta(x_{1:T}|x_0)$ is intractable
- The fixed variational distribution (noising process) $q(x_{1:T}|x_0)$ induce an ELBO.
- Their resulting training loss is equivalent to ϵ-prediction loss (with some weighting)

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right] \quad (14)$$

- Their sampling process $p_\theta(x_{t-1}|x_t)$ is a specific instance of VP-SDE with Euler discretization.

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z} \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Ho et al. 2020 DDPM

Connection between VAE and diffusion

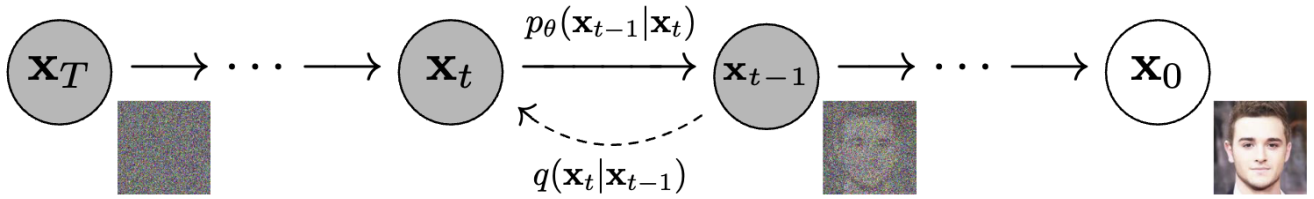
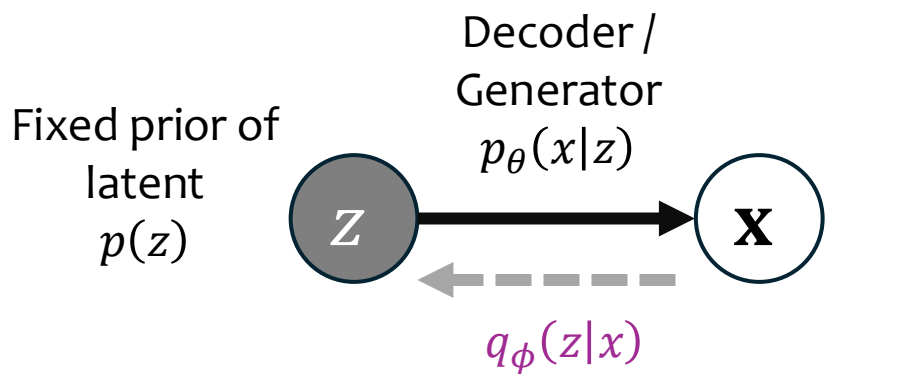


Figure 2: The directed graphical model considered in this work.



- Diffusion can be viewed as a *probabilistic graph* model: denoising trajectory as a sequence of latents. Similar to hierarchical VAE.
- Forward noising process is a fixed encoder (variational distribution); Denoising process is a learned decoder.
- Denoising loss (with certain weighting) is equivalent to ELBO.
- The sampling of the Markov chain is equivalent to discretized SDE.

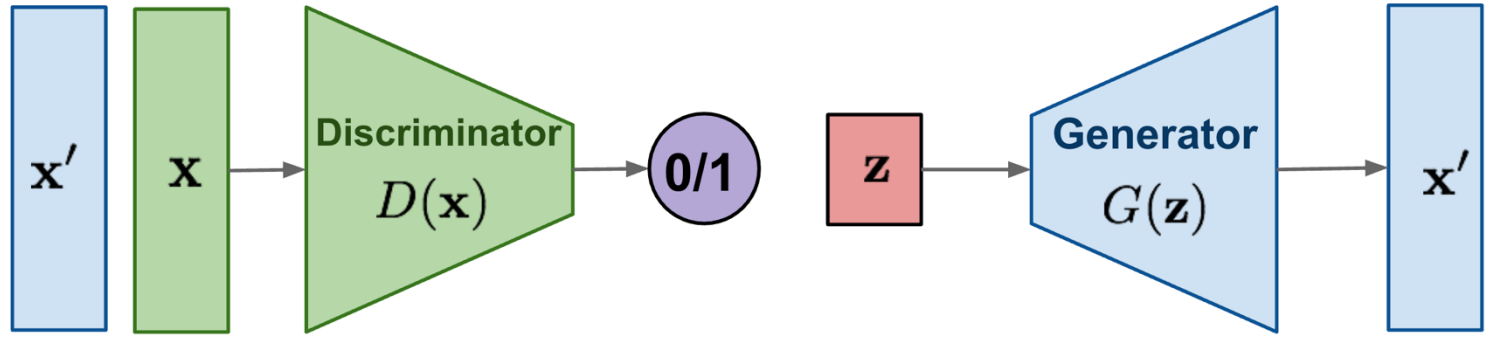
Derivation of ELBO in DDPM

$$
\begin{aligned}
\log p_\theta(x_0) &= \log \frac{p_\theta(x_0, x_{1:T})}{p_\theta(x_{1:T}|x_0)} \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0, x_{1:T})}{q(x_{1:T}|x_0)} \cdot \frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)}\right] \\
&\geq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_0, x_{1:T})}{q(x_{1:T}|x_0)}\right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{\prod_{t=1:T} p_\theta(x_{t-1}|x_t) p(x_T)}{\prod_{t=2:T} q(x_{t-1}|x_0, x_t)}\right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{\prod_{t=1:T} p_\theta(x_{t-1}|x_t) p(x_T)}{\prod_{t=2:T} q(x_{t-1}|x_0, x_t) q(x_T|x_0)}\right] \\
&= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log p_\theta(x_0|x_1) + \sum_{t=2:T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_0, x_t)} + \log \frac{p(x_T)}{q(x_T|x_0)}\right] \\
&= \mathbb{E}_{q(x_1|x_0)}\Big[\log p_\theta(x_0|x_1)\Big] + \sum_{t=2:T} \mathbb{E}_{q(x_{t-1}, x_t|x_0)} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_0, x_t)} + \mathbb{E}_{q(x_T|x_0)} \log \frac{p(x_T)}{q(x_T|x_0)} \\
&= \mathbb{E}_{q(x_1|x_0)}\Big[\log p_\theta(x_0|x_1)\Big] + \sum_{t=2:T} \mathbb{E}_{q(x_t|x_0)} q(x_{t-1}|x_t, x_0) \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_0, x_t)} + \mathbb{E}_{q(x_T|x_0)} \log \frac{p(x_T)}{q(x_T|x_0)} \\
&= \mathbb{E}_{q(x_1|x_0)}\Big[\log p_\theta(x_0|x_1)\Big] - \sum_{t=2:T} \mathbb{E}_{q(x_t|x_0)} D_{KL}\Big(q(x_{t-1}|x_0, x_t)||p_\theta(x_{t-1}|x_t)\Big) - D_{KL}\Big(q(x_T|x_0)||p(x_T)\Big)
\end{aligned}
$$

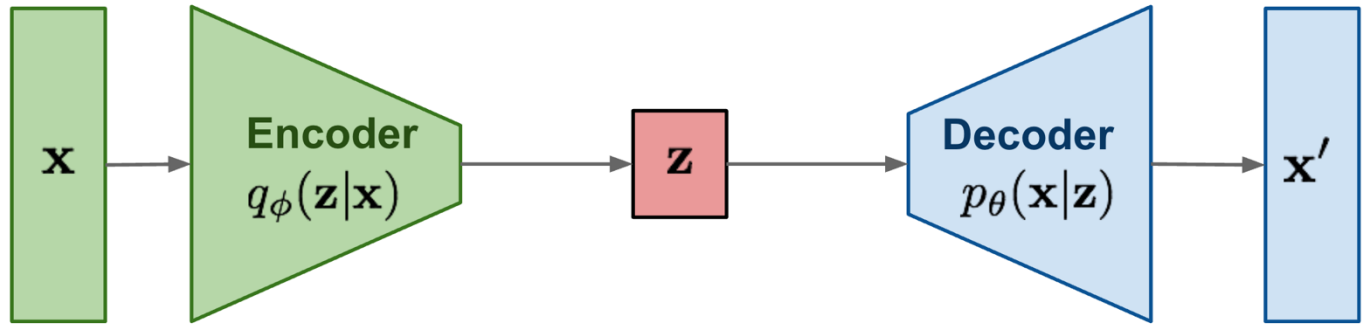
Link to normalizing flow

- Both are using a neural-ODE to transform a distribution of interest to Gaussian.
- Normalizing flow has inverse and explicit Jacobian, thus can maximize log likelihood directly.
- Diffusion bound log likelihood with the ELBO.

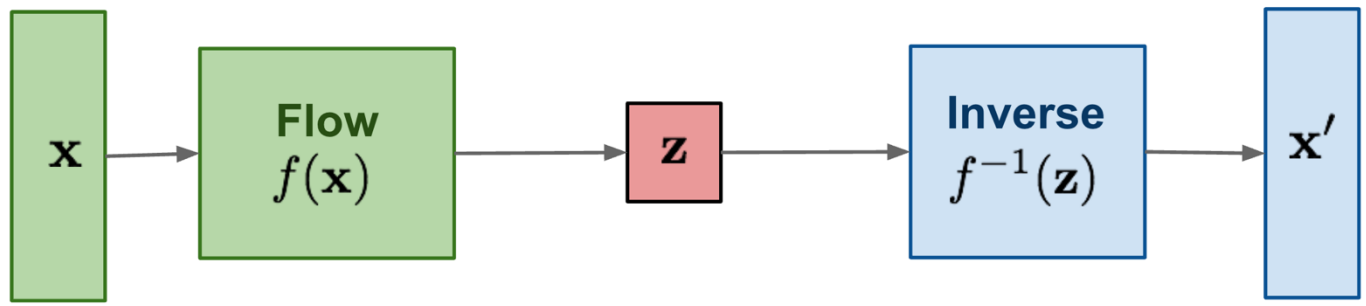
GAN: Adversarial training



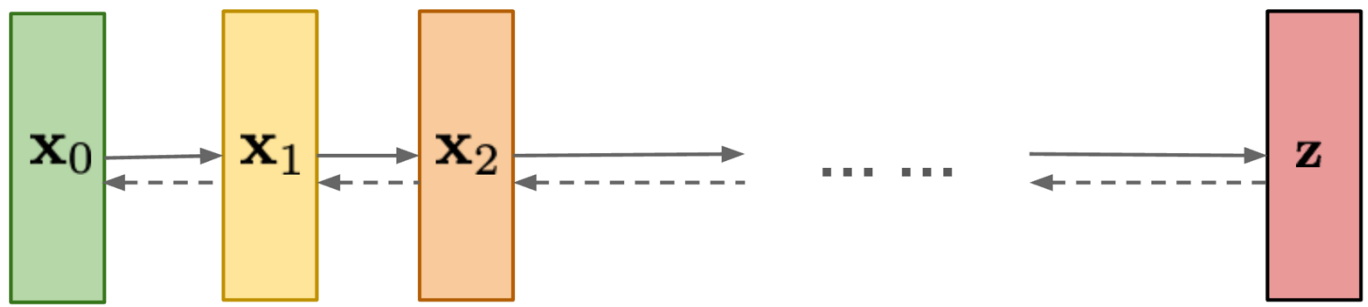
VAE: maximize variational lower bound



Flow-based models: Invertible transform of distributions



Diffusion models: Gradually add Gaussian noise and then reverse



https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Summary

- Link to VP formulation via reparametrization of space and time
- Various diffusion loss are equivalent up to a scaling and identity.
- DDPM brings the variational inference perspective, ELBO interpretation of the DSM loss.
- DDPM is discretized SDE (Maruyama) as a Markov chain.
  - DDIM is discretized ODE (Euler).

Code Demo

Conditioning Diffusion

Why conditioning?

- We want to control the sampling by some property!
- Many problems can be framed as conditional sampling.

- For example:
  - $p(\text{image} \mid \text{caption})$
  - $p(\text{face} \mid \text{identity})$
  - $p(\text{speech} \mid \text{text, voice identity})$
  - $p(\text{next frame} \mid \text{current frame, action})$
  - $p(\text{action trajectory} \mid \text{goal})$

Training Conditioning Diffusion

- Conditional score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x} \mid c, \sigma)$$

- Training by conditional denoising loss

$$\mathcal{L}_\sigma = \mathbb{E}_{\substack{\mathbf{x}_0, c \sim p_0(\mathbf{x}, c) \\ z \sim \mathcal{N}(0, I)}} \|\mathbf{D}_\theta(\mathbf{x}_0 + \sigma \mathbf{z}; c, \sigma) - \mathbf{x}_0\|^2$$

  - Conditioning c are dropped randomly (for CFG)

Sampling: Classifier-Free Guidance (CFG)

- Guided score:

$$\varepsilon_{guided} = \varepsilon_{uncond} + w \cdot (\varepsilon_{cond} - \varepsilon_{uncond})$$
$$\varepsilon(x_t; \emptyset, t) \qquad \varepsilon(x_t; \text{cond}, t)$$

Forward pass twice with and without conditioning.

- Sampling: treat as normal score.
- Training
  - conditional + unconditional (null token)

- Guidance scale w:
  - diversity – prompt alignment trade-off

- Extensions:
  - negative prompts – sth to guide away from.
  - ε_{uncond} can also be a worse version of score (under-trained) or attention perturbed.

https://sander.ai/2022/05/26/guidance.html

One interpretation of CFG

- Using Bayes' rule:

$$p(x|c) = \frac{p(x)p(c|x)}{p(c)}$$

Taking log derivative

$$\nabla_{\mathbf{x}} \log p(x|c) = \nabla_{\mathbf{x}} \log p(x) + \nabla_{\mathbf{x}} \log p(c|x)$$

- Thus

$$\nabla \log p(x|c) - \nabla \log p(x) = \nabla_{\mathbf{x}} \log p(c|x)$$
$$w(\nabla \log p(x|c) - \nabla \log p(x)) = \nabla \log p^w(c|x)$$

- Guided score can be understood as

$$\nabla \log p(x) + w(\nabla \log p(x|c) - \nabla \log p(x)) = \nabla \log\big(p(x) \cdot p^w(c|x)\big)$$

https://sander.ai/2022/05/26/guidance.html

Practical Details:
How to Make It Work in Real Life

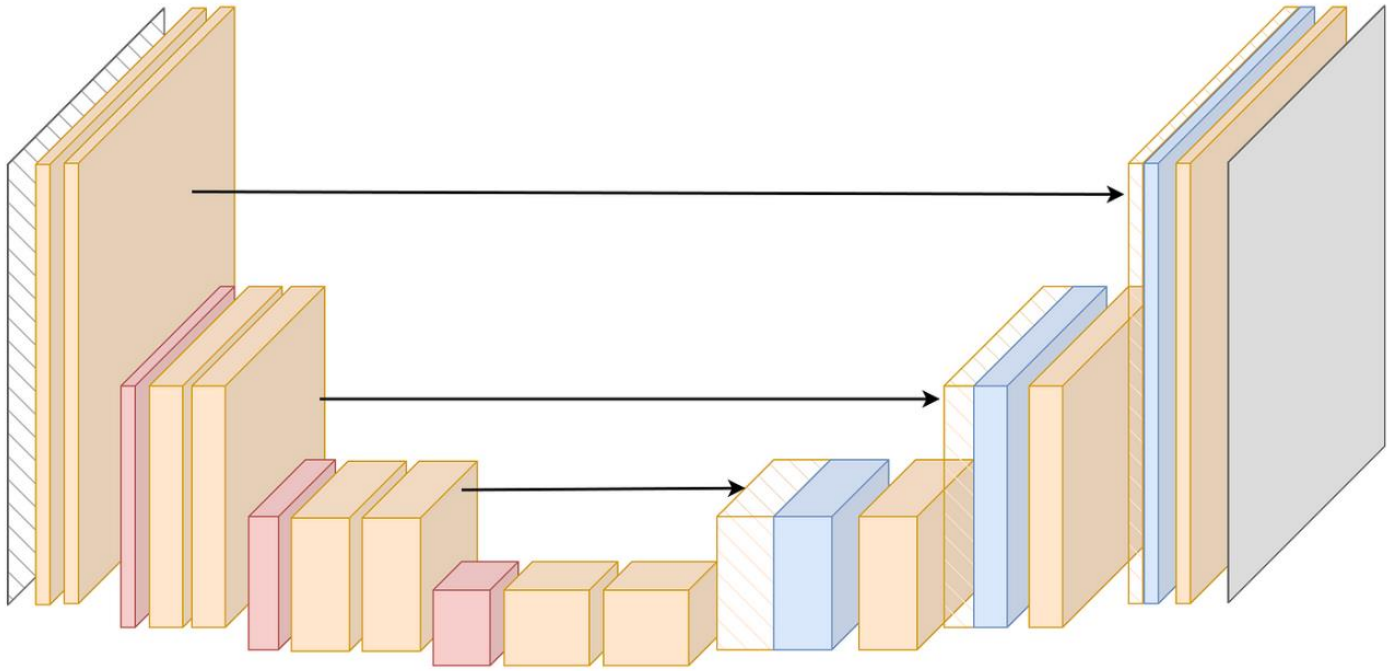
Modelling score / denoiser function

$$\widehat{\mathbf{x}_0} = \mathbf{D}_\theta(\mathbf{x}_\sigma; \sigma)$$

- Desiderata
  - It's a vector field: input $\mathbf{x}$ and output $\widehat{\mathbf{x}_0}$ has the same shape.
  - It's time / noise scale dependent, output change based on the scalar σ.
  - Input, output and function property has certain scaling based on time.
    - E.g. $\sigma \to 0$, $\mathbf{D}_\theta \to \mathrm{Id}$

What network do we use?

UNet: image-to-image mapping

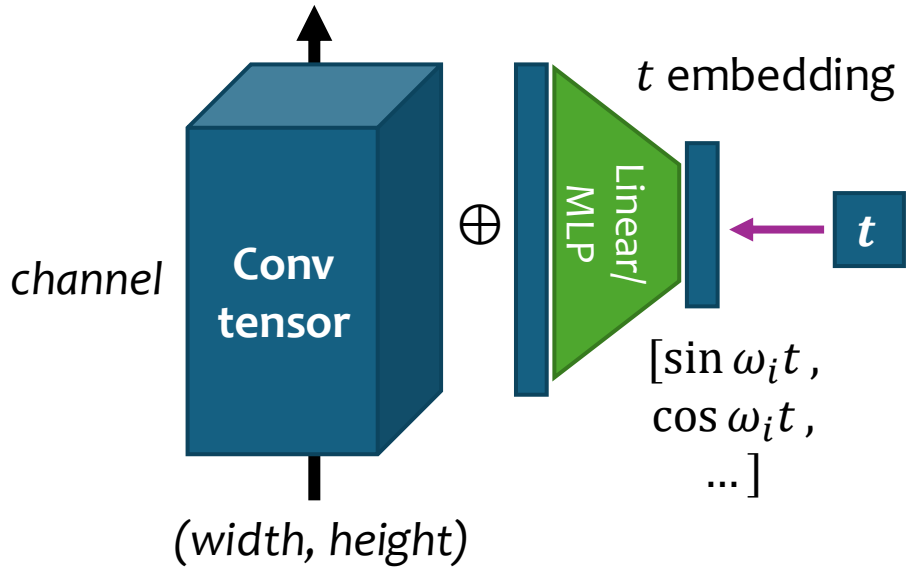


Inductive biases

- Convolution operation
  - Save parameter, spatial invariant
- Down/Up sampling
  - Multiscale / Hierarchy
  - Learn features at multi scale and multi-abstraction levels.
- Skip connection
  - ***No bottleneck***
  - Route features at the same scale directly.
- Adding self attention for non local relation.

Modelling Time/noise scale-dependency

- Add time dependency
  - Assume time dependency is spatially homogeneous.
  - **SongUnet:** Add one scalar per channel
    $silu(norm1(x + shift))$
  - **DhariwalUNet:** Scale and shift each channel
    $silu(norm1(x) \cdot (scale + 1) + shift)$
  - Parametrize scale / shift by MLP / linear of Fourier embedding of time / noise.

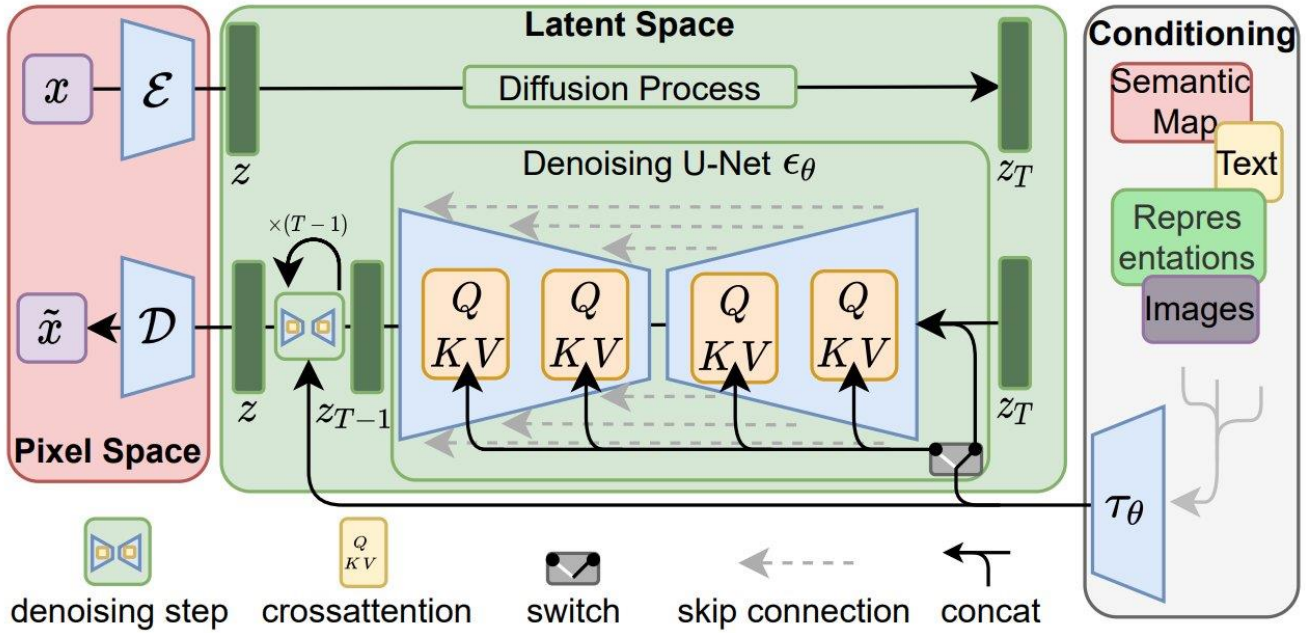


SongUnet, DhariwalUNet Per NVlab/EDM

Latent Diffusion: VAE + diffusion

- Compress input into a latent space via VAE $\mathcal{E}, \mathcal{D}$.
  - The latent is spatial and high capacity (e.g. $(4,64,64)$).

- Modelling latent density $p(z)$ with diffusion.



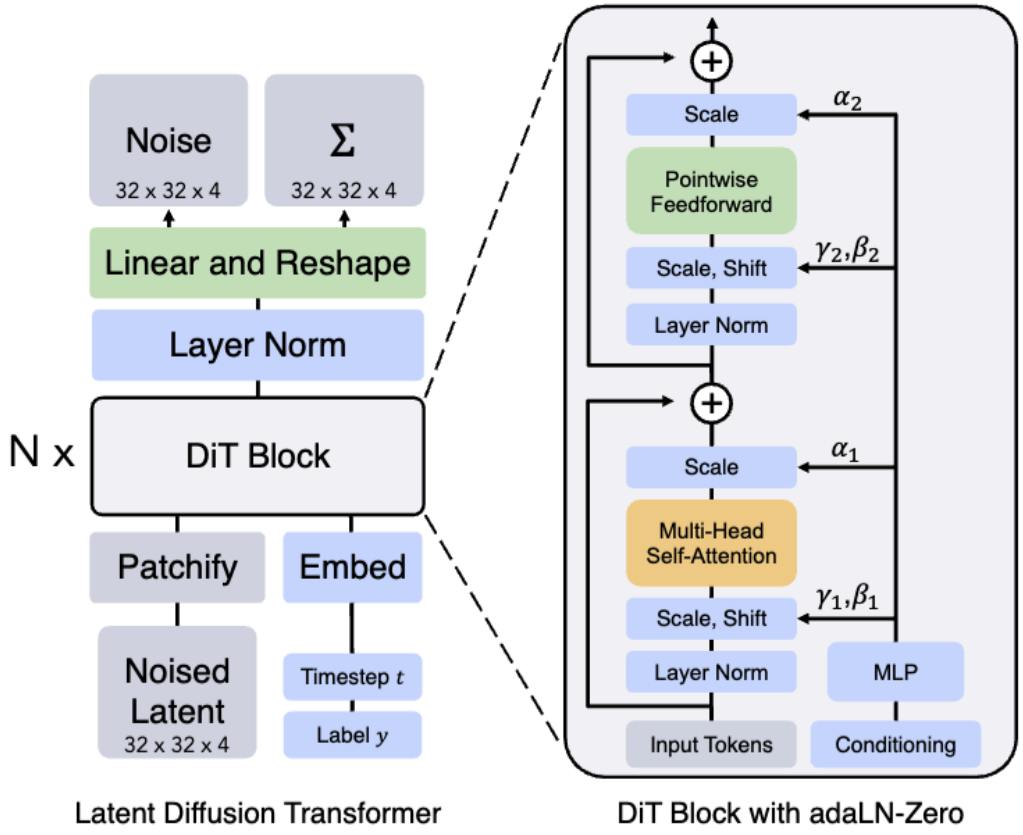
Stable diffusion: 2022

Diffusion Transformer (DiT)

- Use vision transformer to model the VAE latents

- Time conditioning
  - AdaLN-zero:
    Shift, scale each channel at input; gate at output of Attn and MLP.

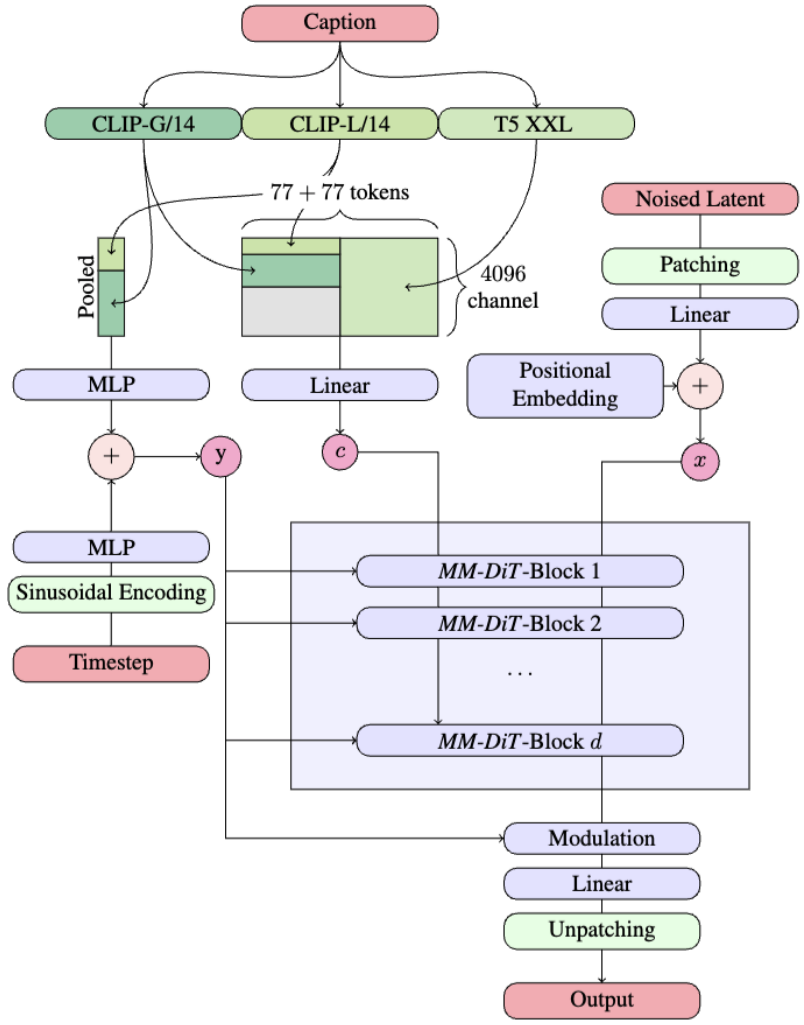
$$
\begin{aligned}
&\mathrm{x} \\
&\leftarrow \mathrm{x} + \mathrm{gate\_mlp} \cdot \mathrm{MLP}(\mathrm{LN}_2(\mathrm{x}) \odot (1 + \mathrm{scale\_mlp}) + \mathrm{shift\_mlp})\mathrm{x} \\
&\leftarrow \mathrm{x} + \mathrm{gate\_msa} \cdot \mathrm{Attn}(\mathrm{LN}_1(\mathrm{x}) \odot (1 + \mathrm{scale\_msa}) + \mathrm{shift\_msa})
\end{aligned}
$$



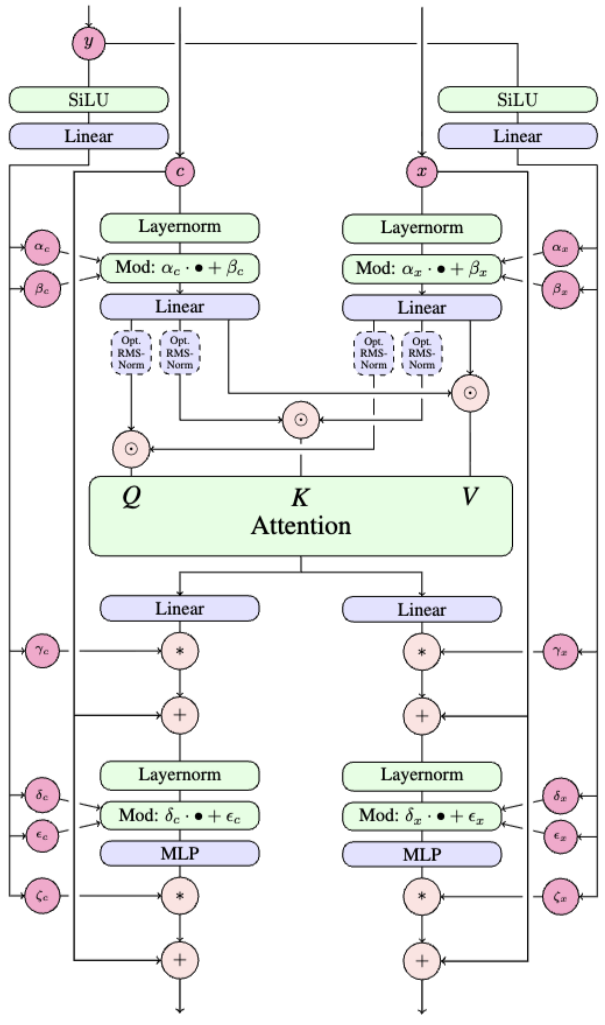
DiT, Peebles, Xie,2023

Multi-modal DiT (MMDiT)

- Text and image tokens transmit information between each other via “joint” attention.
- Time and global semantics modulate via shift, scale, gate.



(a) Overview of all components.



(b) One *MM-DiT* block

Stable Diffusion 3, 2024

Note on modelling Time-dependency

- Network preconditioning
  - Build in certain noise scale dependency to help learning.
  - Input/output scaling for numerical stability

$$D_\theta(\mathbf{x};\sigma) = c_{skip}(\sigma)\mathbf{x} + c_{out}(\sigma)F_\theta(c_{in}(\sigma)\mathbf{x}; c_{noise}(\sigma))$$

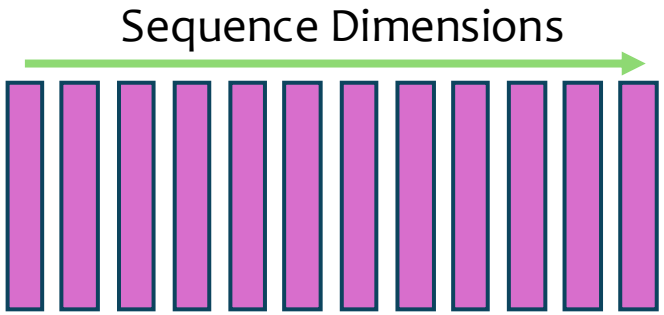
$$\frac{\sigma_{data}^2}{\sigma^2+\sigma_{data}^2} \qquad \frac{\sigma_{data}\cdot\sigma}{\sqrt{\sigma^2+\sigma_{data}^2}} \qquad \frac{1}{\sqrt{\sigma^2+\sigma_{data}^2}} \qquad \frac{1}{4}\ln\sigma$$

Conditioning Mechanisms

- Adaptive Layer Norm (AdaLN) — class/timestep conditioning
- Cross-attention (text-to-image: CLIP / T5 embeddings)
- In-context / concatenation conditioning, self attention.
- ControlNet — structural control via auxiliary encoder

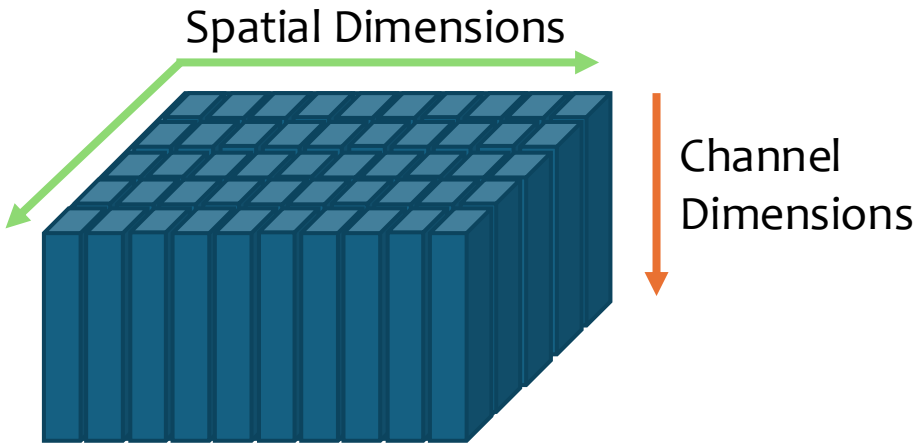
Text2Image as translation

Source language: Words

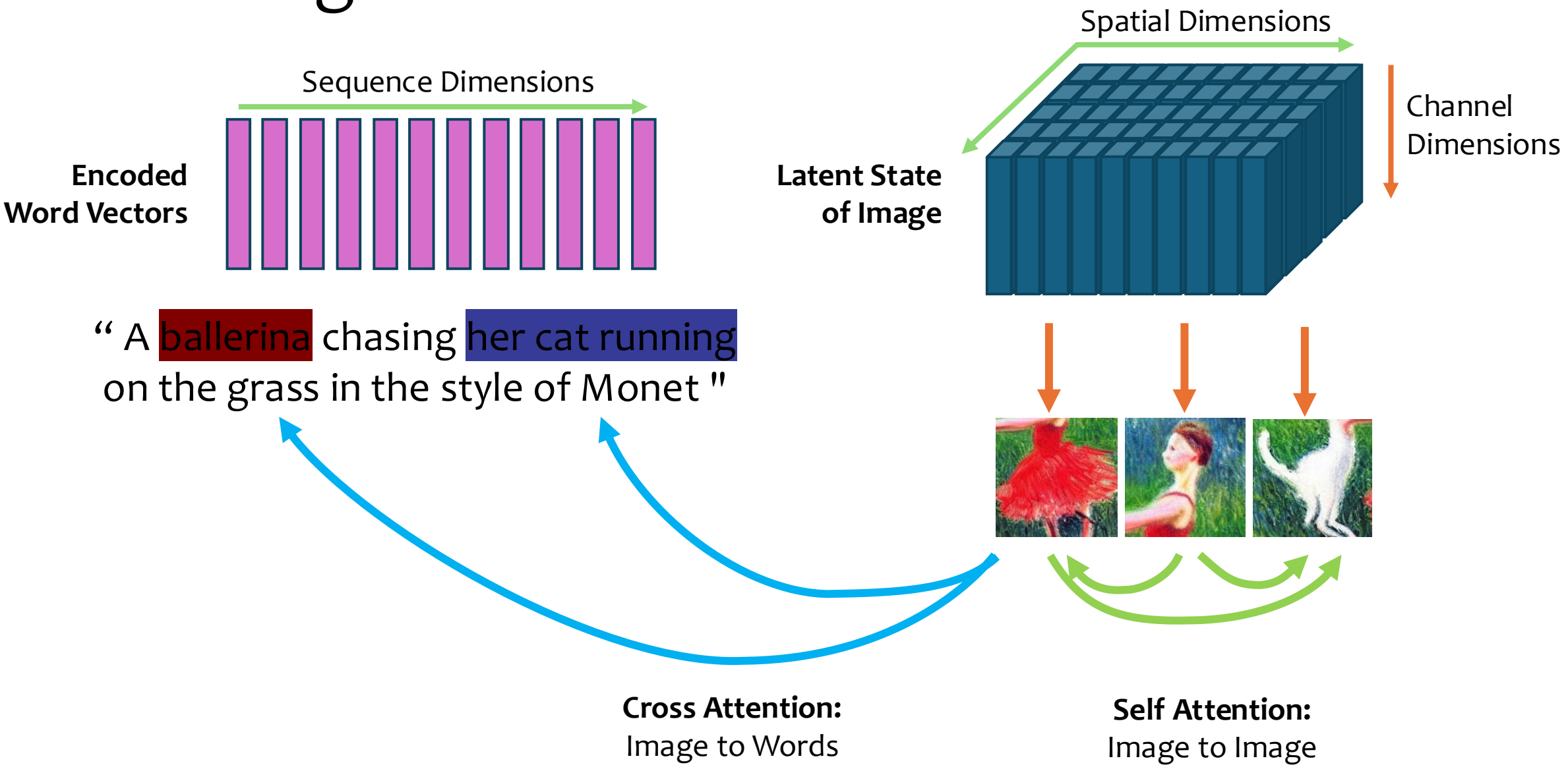


" A ballerina chasing her cat running on the grass in the style of Monet "

Target language: Images

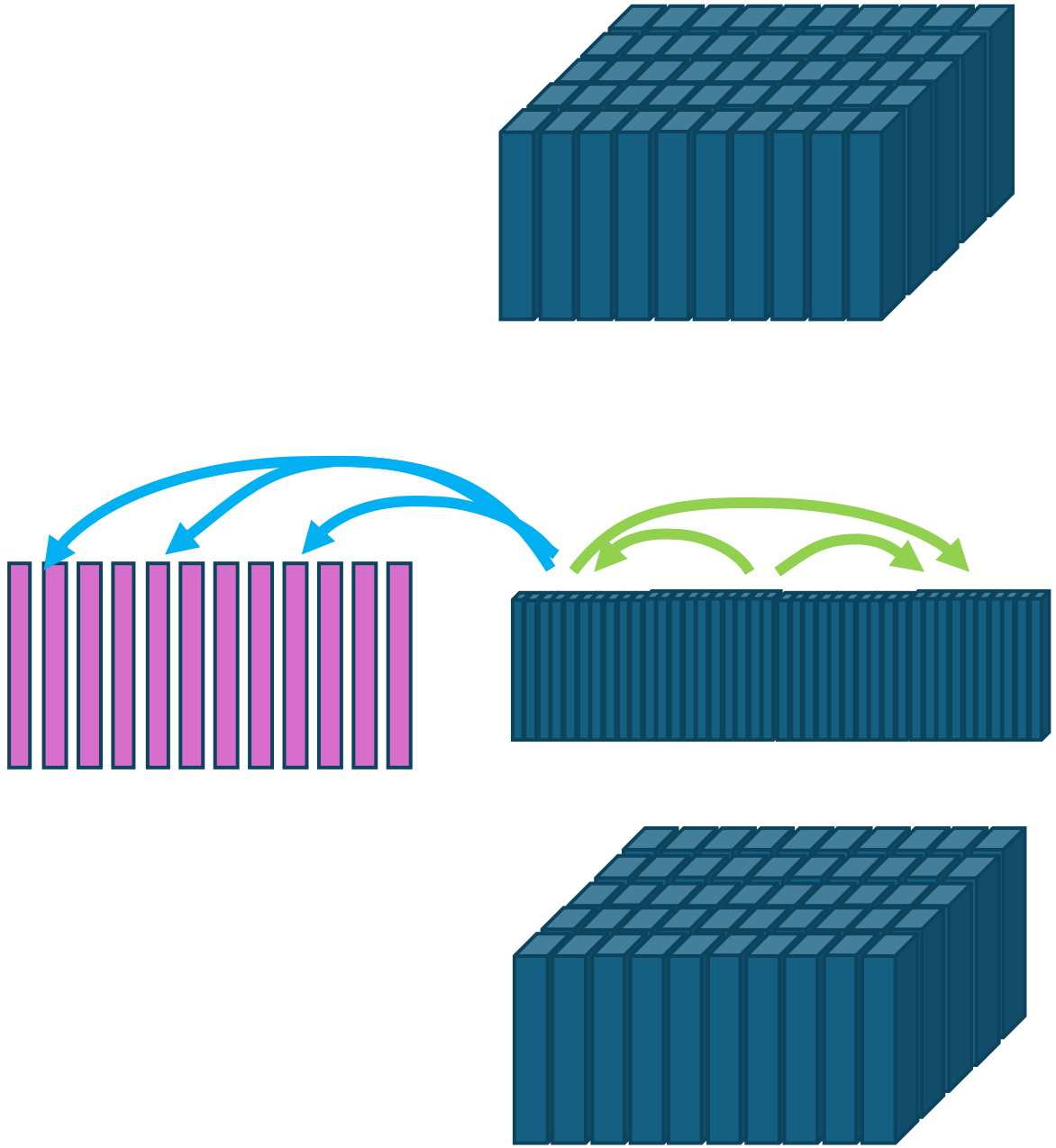


Text2Image as translation



Spatial Transformer

- Rearrange spatial tensor to sequence.
- Cross Attention
- Self Attention
- FFN
- Rearrange back to spatial tensor (same shape)



Summary: Network Architecture

- UNet — encoder-decoder with skip connections
- Latent Diffusion (VAE + UNet)
  - Encode to latent space; diffuse and denoise there
- Diffusion Transformer (DiT)
  - Patch-based ViT backbone — scales with compute
- MMDiT (Multi-Modal DiT)
  - Separate streams for image & text, joined attention

- Takeaway
  - Use Conv to handle high resolution image, use attention at lower resolution latent
  - Channel wise modulation is popular and successful for time or class.

Training Details

- Loss weighting across noise scale

$$\int \lambda(\sigma)\, \mathcal{L}_\sigma d\sigma = \int \lambda(\sigma)\, \mathbb{E}_{\substack{\mathbf{x}_0 \sim p_0 \\ z \sim \mathcal{N}(0,I)}} \|\mathbf{D}_\theta(\mathbf{x}_0 + \sigma \boldsymbol{z}; \sigma) - \mathbf{x}_0\|^2 d\sigma$$

  - Source of weighting
    - Explicit weighting function
    - Density of sampling σ
    - Implicit weighting due to loss type ($\mathbf{x}_0$, v, ϵ-pred) and network parametrization.
  - Upweight noise levels relevant to perceptual quality. Reduce variance.

- Exponential Moving Average (EMA)
  - Smooth the model weights across training trajectory.

Efficient generation

- Efficient sampler design
  - Higher-order, better ODE solver: DDIM, Heun, PLDM, DPM-Solver++,v3
  - Leveraging geometric property of the trajectory to accelerate.

- Distillations
  - ODE solution is a deterministic mapping.
  - Train / finetune a network to predict the sampling outcome
  - Consistency model learns this ODE trajectory.

Reference ; Further reading

- Classic paper
  - Ho (2020), DDPM, arxiv.org/abs/2006.11239
  - Song (2020), Score based model, arxiv.org/abs/2011.13456
  - *Rombach, et al. (2021) Latent diffusion arxiv.org/abs/2112.10752*
  - Karras et al. (2022). EDM, arxiv.org/abs/2206.00364
  - *Peebles, Xie (2022). DiT, arxiv.org/abs/2212.09748*

- *Blog posts*
  - *Discussion on noise scheduling sander.ai/2024/06/14/noise-schedules.html*
  - *Equivalence of flow matching and diffusion diffusionflow.github.io/*
  - *Diffusion for Video lilianweng.github.io/posts/2024-04-12-diffusion-video/*