COMP 760 Week 4: Generative Models Primer II

By Joey Bose and Prakash Panangaden

Admin stuff week 4

Starting from Week 5: Weekly Paper Review

- You must perform a review of 1 **CORE** paper unless you are presenting that week.
- Paper Reviews should be maximum 1 page in length and can be done in Latex or Google Docs.
- These reviews will be unmarked but still mandatory. You should submit these on McGill MyCourses: https://mycourses2.mcgill.ca/d2l/loginh/
- Please use these reviews as a basis for asking good questions during in class student presentations.

How to Review a Paper: A crash course

- The main goal of reviewing is to determine if a submission will bring sufficient value and novelty to a subset of the community.
- Reading papers is time consuming. This is why its critical to also read related work which should help bridge gaps in your knowledge.
- It is important to consider each paper holistically along multiple axis: 1) Is the presentation clear? 2.) Are the theoretical results sound? 3.) Are the experiments compelling?

How to Review a Paper: A crash course

While Reading Consider the Following:

- Objective of the work: What is the goal of the paper? Is it a new framework? Theoretical result? Or a novel application?
- Try to look for strong points: Is the paper clear? Rigorous? Reproducible?
- Are there weaknesses: Is the paper lacking on any front? Maybe another experiment? More detail on the theoretical setting? Unfair assumptions?
- If you were the author, what would you add to the paper to make it stronger? What are areas of improvement?

How to Review a Paper: A crash course

The Review Should Contain the Following Sections

- Summary of the work
- Strengths
- Weaknesses
- Questions to the authors

How to Review a Paper: A crash course

Resources to help you

- CVPR Reviewing Tutorial:
- ICLR 2022 ORAL paper:
- Eric Jang's criteria for reviewing blog post

Mila

Paper Presentation Guidelines:

High Level

- 30 min (40 for two people) presentation + 5 min questions.
- Email slides 1 day before so they can be hosted on the course website.
- It’s important that you read the related work of the paper as well as the appendix.

Paper Presentation Guidelines:

Rough Rubric

- Motivate and define the problem in the paper: 20%
- Given sufficient context and Related Work: 20%
- Present the main result in a figure our break down the theorems in a easy to understand format: 20%
- Outline the experiments and give an overview of all the results: 20%
- Give your own perspectives on the paper: 10%
- Finish under time and answer questions: 10%

Mila

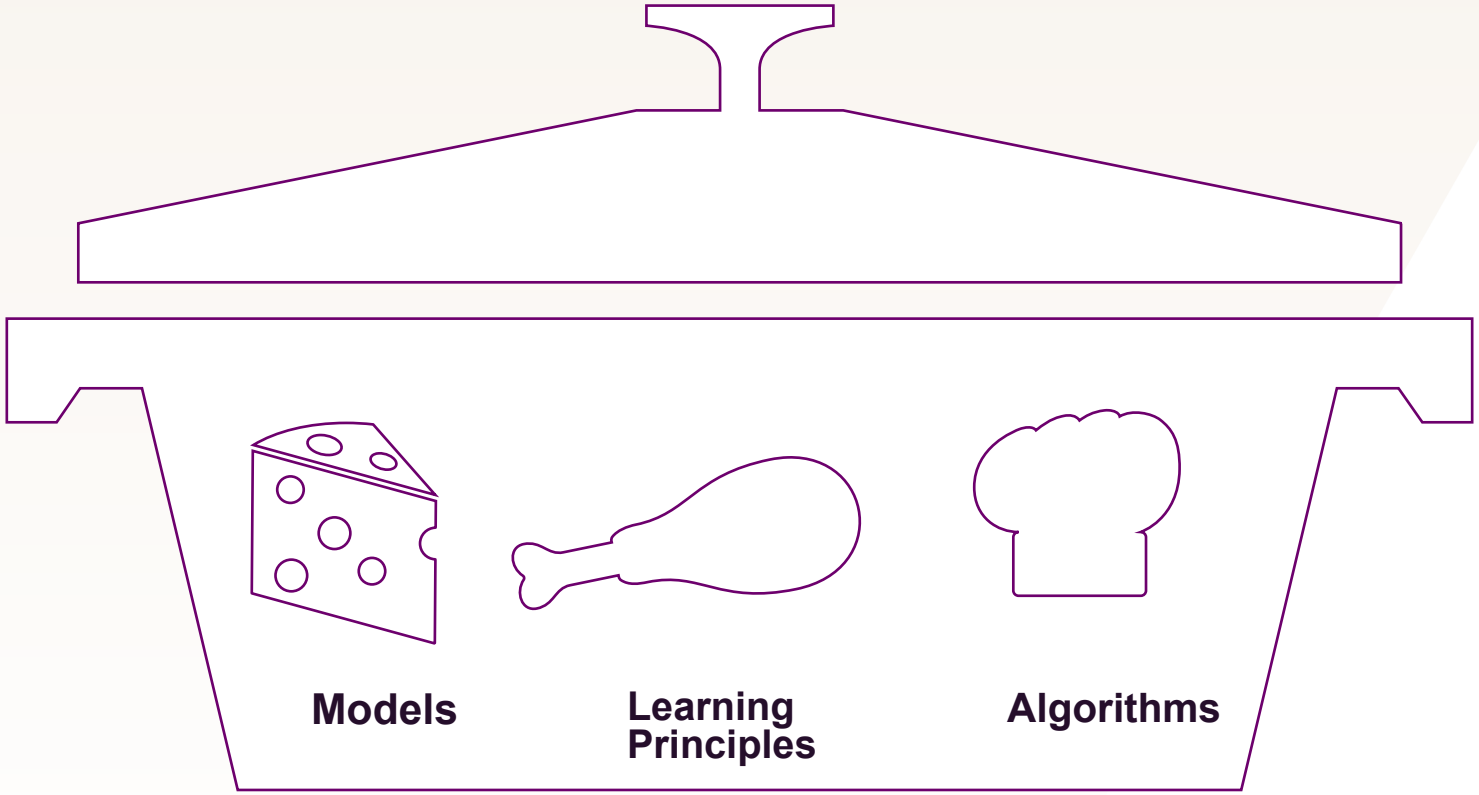
Paper Presentation Guidelines:

General Presentation Guidelines

- Avoid using Beamer if possible.
- Present 1 idea per slide and try to make it light notationally or explain the equations well visually.
- The goal of the presentation is not to convey how much math you understand but to transmit as much information to the audience in palatable format.
- You can use past ICLR 2022 keynotes as examples of good presentations link here

Recap of Week 3

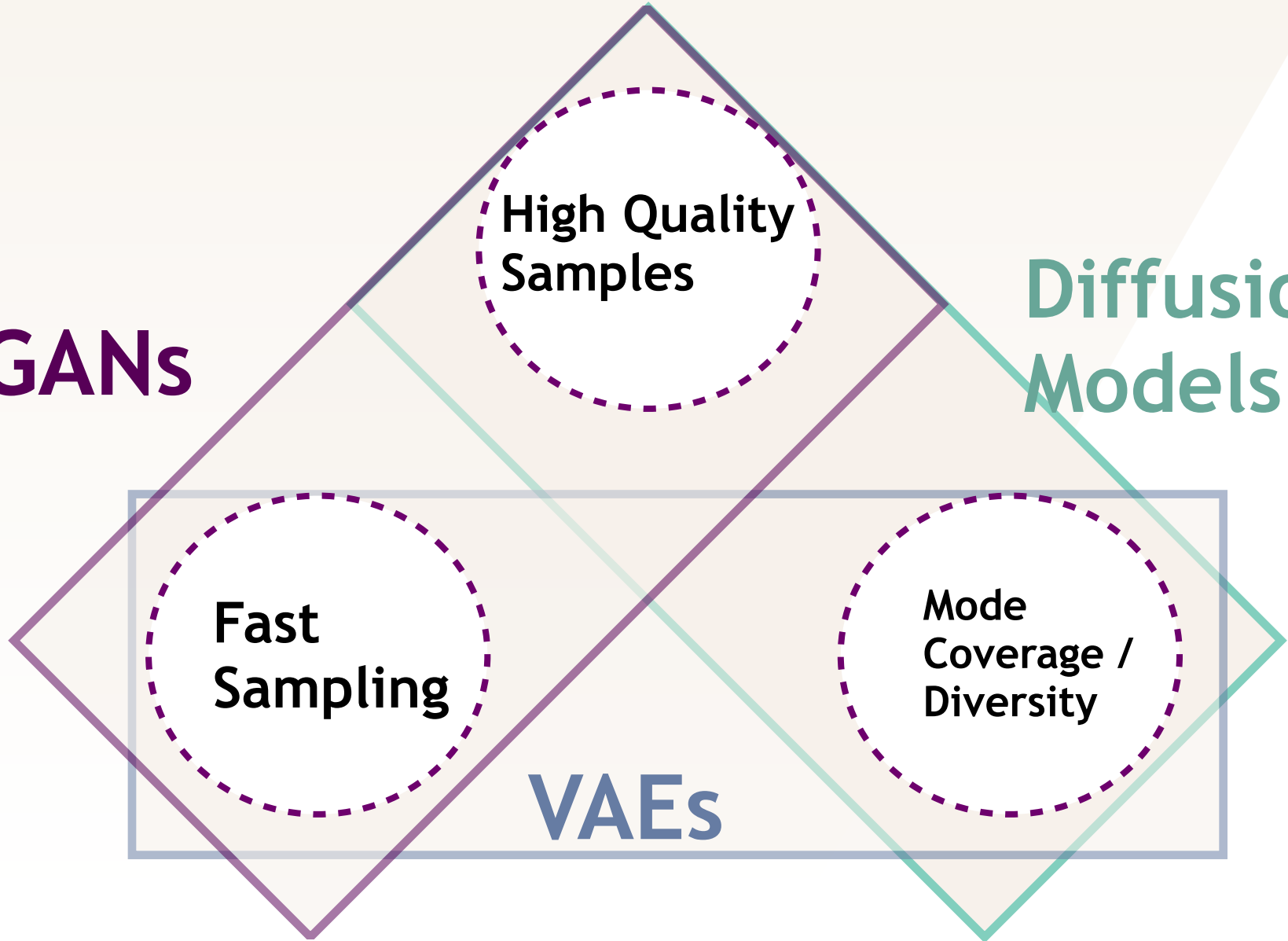
Ingredients Of A Generative Model



Generative Model Soup

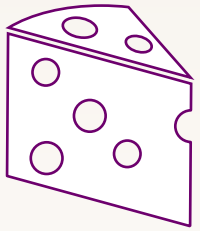
Mila

Tradeoffs in a Generative Model



Mila

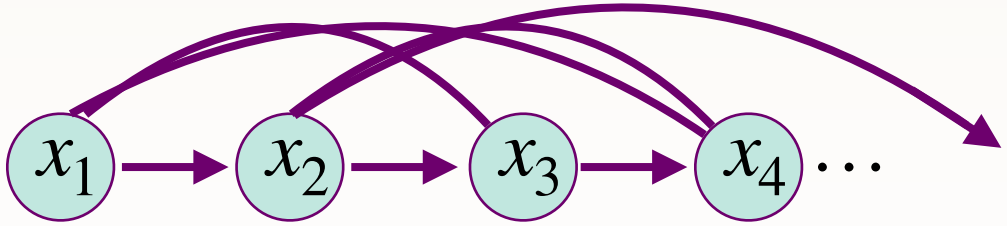
Families of Generative Models



Models

- Fully Observed Models

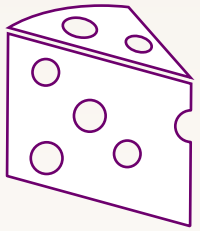
Models that observe all data components directly without introducing any assumptions about unobserved variables.



$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_1, \ldots, x_{i-1})$$

Mila

Families of Generative Models



Likelihood Based Generative models

- Strong Constraints on Model Class to enable Likelihood computations.
- Might require surrogate objectives for approximate Maximum Likelihood Training

Implicit Generative Models

- Requires tricky Adversarial Optimization.
- Known to drop modes of the data distribution

Variational Inference

$$\log p(\mathscr{D}) \geq \mathbb{E}_{q(_i(z)}[\log p(x_i \mid z)] + \mathbb{E}_{q_i(z)}\left[\log \frac{p(z)}{q_i(z)}\right]$$

$$\log p(\mathscr{D}) \geq \mathbb{E}_{q_i(z)}[\log p(x_i \mid z)] - D_{KL}(q_i(z) \mid\mid p(z))$$

Reconstruction Error **Regularizer**

Mila

VAE - Generative Model

- Sample a Reparametrized Latent z

Latent sample

$$z \sim q_\phi(z \mid x)$$

→ **Generative Model** $p_\psi(x \mid z)$ →

Reconstructed datum

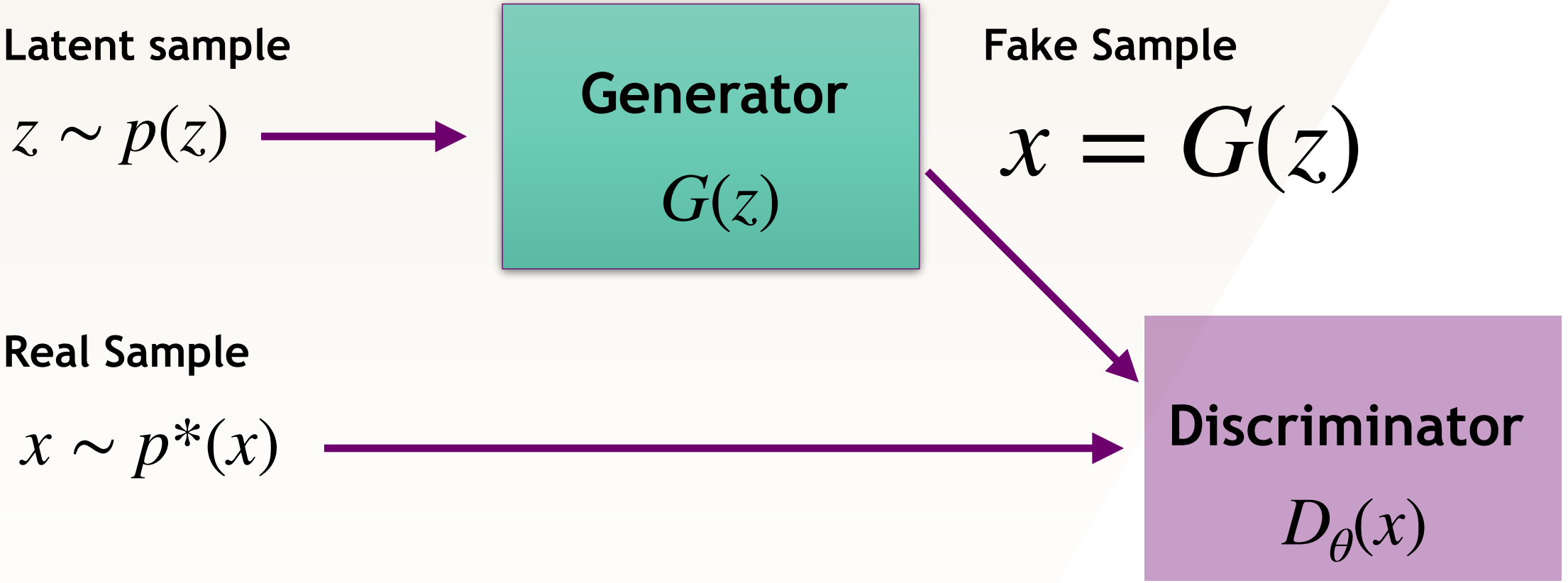
$$\tilde{x}$$

$$\log p(\mathscr{D}) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\psi(x \mid z)] - D_{KL}(q_\phi(z \mid x) \,||\, p(z))$$

Reconstruction Error — Regularizer

Mila

Generative Adversarial Networks



$$\min_G \max_D \mathbb{E}_{p^*(x)}[\log(D_\theta(x))] + \mathbb{E}_{q_\phi(\tilde{x})}[\log(1 - D_\theta(\tilde{x}))]$$

Mila

Density Estimation with Normalizing Flows

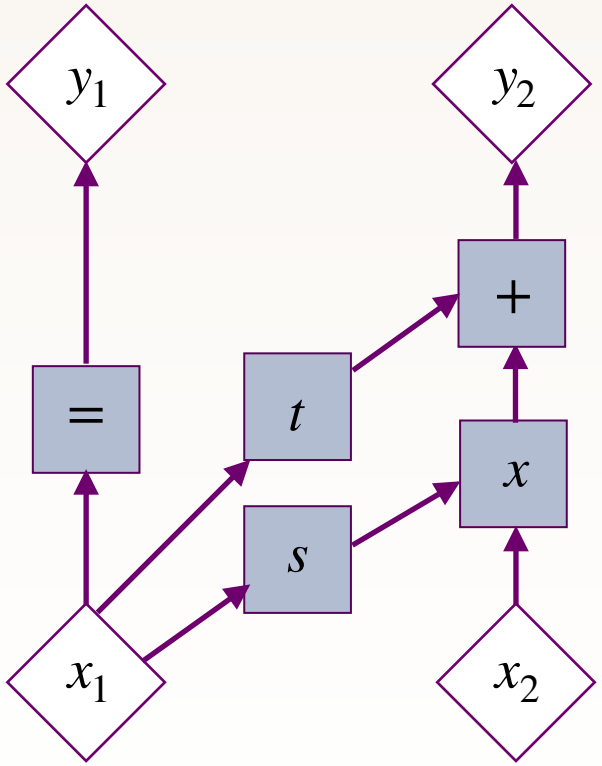
- Each function f_i must be invertible.
- We must be able to efficiently sample from the final distribution, $z_k = f_k \circ f_{k-1} \circ \dots \circ f_1(z_0)$
- We must be able to efficiently compute the associated change in volume

Change in Volume

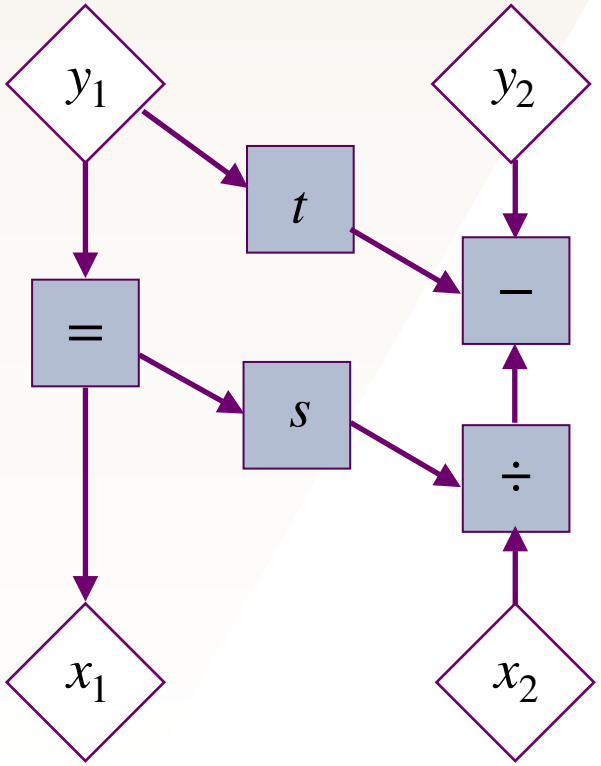
$$\log p(z_k) = \log p(z_0) - \sum_{j=1}^{k} \log \det \left| \frac{\partial f_j}{\partial z_{j-1}} \right|$$

Affine Coupling Flows

Forward Transform

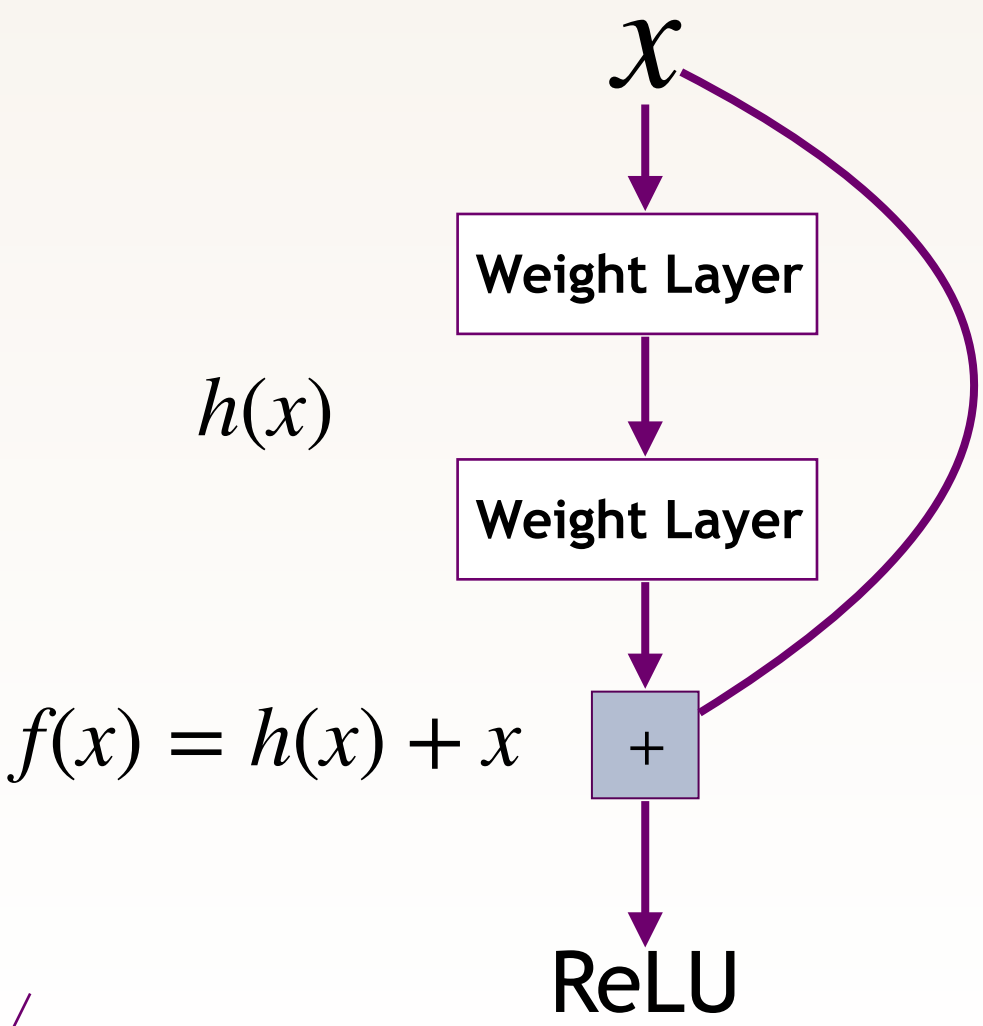


Inverse Transform



Mila

Residual Flows



Forward Dynamics

$$x_{t+1} \longleftarrow x_t + \alpha h(x_t)$$

This looks an Euler Discretization of an ODE.

Reverse Dynamics

$$x_t \longleftarrow x_{t+1} - \alpha h(x_t)$$

When can we invert a ResNet?

Mila

Neural ODE

Forward Dynamics

$$x_{t+1} \longleftarrow x_t + \alpha h(x_t)$$

This looks an Euler Discretization of an ODE

Reverse Dynamics

$$x_t \longleftarrow x_{t+1} - \alpha h(x_t)$$

When can we invert a ResNet?

What if we take the discretization to be infinitesimal?

$$z_{t_0} = z \qquad z_{t_1} = z$$

$$\frac{dz_t}{dt} = h_t(t, z_t)$$

This is an ODE!

Continuous Normalizing Flows

Forward Dynamics

$$x = z_{t_1} = z_0 + \int_{t=t_0}^{t_1} h(z_t, t)dt$$

What if we take the discretization to be infinitesimal?

$$z_{t_0} = z \qquad z_{t_1} = z$$

$$\frac{dz_t}{dt} = h_t(t, z_t)$$

This is an ODE!

Reverse Dynamics

$$z_0 = x + \int_{t=t_1}^{t_0} h_\phi(z_t, t)dt = x - \int_{t=t_0}^{t_1} h_\phi(z_t, t)dt$$

Residual Flows

Forward Dynamics

$$x = z_{t_1} = z_0 + \int_{t=t_0}^{t_1} h(z_t, t)dt$$

Reverse Dynamics

$$z_0 = x + \int_{t=t_1}^{t_0} h_\phi(z_t, t)dt = x - \int_{t=t_0}^{t_1} h_\phi(z_t, t)dt$$

Instantaneous Change of Variable

$$\frac{d \log p(z_t)}{dt} = -\operatorname{Tr}\left(J_{h(t,\cdot)}(z_t)\right)$$

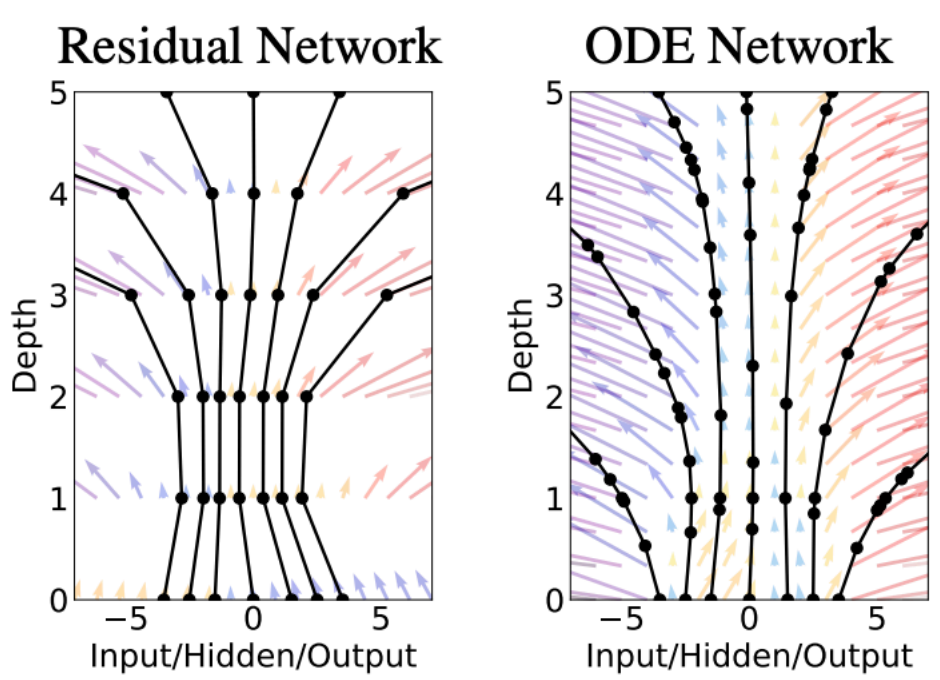


Fig credit: https://arxiv.org/abs/1806.07366

Continuous Normalizing Flows

Instantaneous Change of Variable

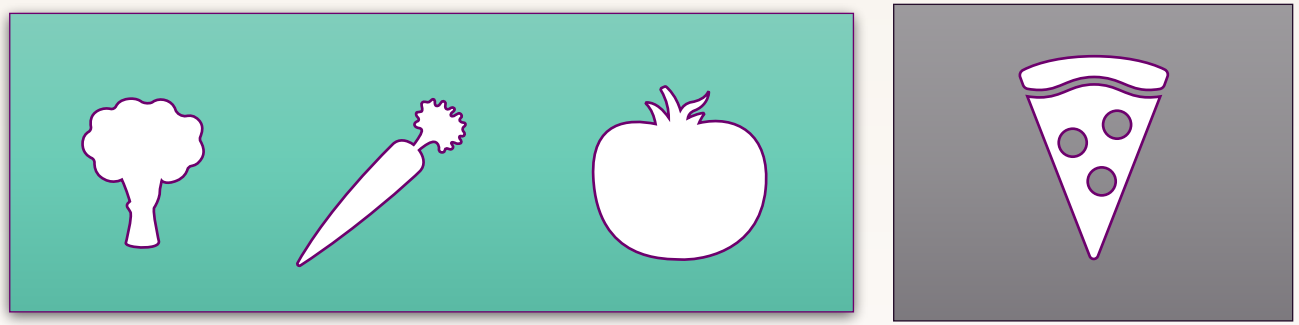
$$\frac{d \log p(z_t)}{dt} = -\operatorname{Tr}\left(J_{h(t,\cdot)}(z_t)\right)$$

Change of Variable

$$\log p(x) = \log p(z_0) - \int_{t=t_0}^{t_1} \operatorname{Tr}\left(J_{h_\phi(t,\cdot)}(z_t)\right)$$

Score-Based Generative Models

General Setup



Data + Noise

$p(\mathscr{D})$

We want to model this

$$x \in \mathbb{R}^n \qquad \mathscr{D} = \{x_i\} \qquad i \in \{1, \ldots, N\}$$

Mila

Motivation for Score-Based Models

$$x \in \mathbb{R}^n \quad \mathcal{D} = \{x_i\} \quad i \in \{1, \ldots, N\}$$

$E_\theta(x) \in \mathbb{R}$ **Energy Function, provides a scalar estimate of the energy of a a certain configuration.**

Boltzmann Distribution

$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{Z_\theta}$$

This is a valid probability distribution if Z_θ is a normalizing constant such that $\int p(x)dx = 1$

Motivation for Score-Based Models

Score Function

$$\mathbf{s}_\theta(x) \approx \nabla_x \log p_{\text{data}}(x) = -\nabla_x E_\theta(x) - \underbrace{\nabla_x Z_\theta}_{=0}$$

- The Energy function E_θ can be unrestricted! No constraints such as invertibility, Lipschitz etc...

- The score function $\mathbf{s}_\theta$ is independent of the normalization constant Z_θ

Motivation for Score-Based Models

How can we learn the score?

$$\mathscr{L} = \mathbb{E}_{x \sim p(x)}[\,||\nabla_x \log p(x) - \mathbf{s}_\theta(x)||_2^2\,]\,.$$

Whats wrong with this current formulation?

Mila

Motivation for Score-Based Models

How can we learn the score?

$$\mathscr{L} = \mathbb{E}_{x \sim p(x)}[\,||\nabla_x \log p(x) - \mathbf{s}_\theta(x)||_2^2\,].$$

Whats wrong with this current formulation?

We don't have access to the real data scores $\nabla_x \log p_{\textbf{data}}(x)$!
Remember we only have a training set which is an empirical distribution not the entire thing!

Mila

Motivation for Score-Based Models

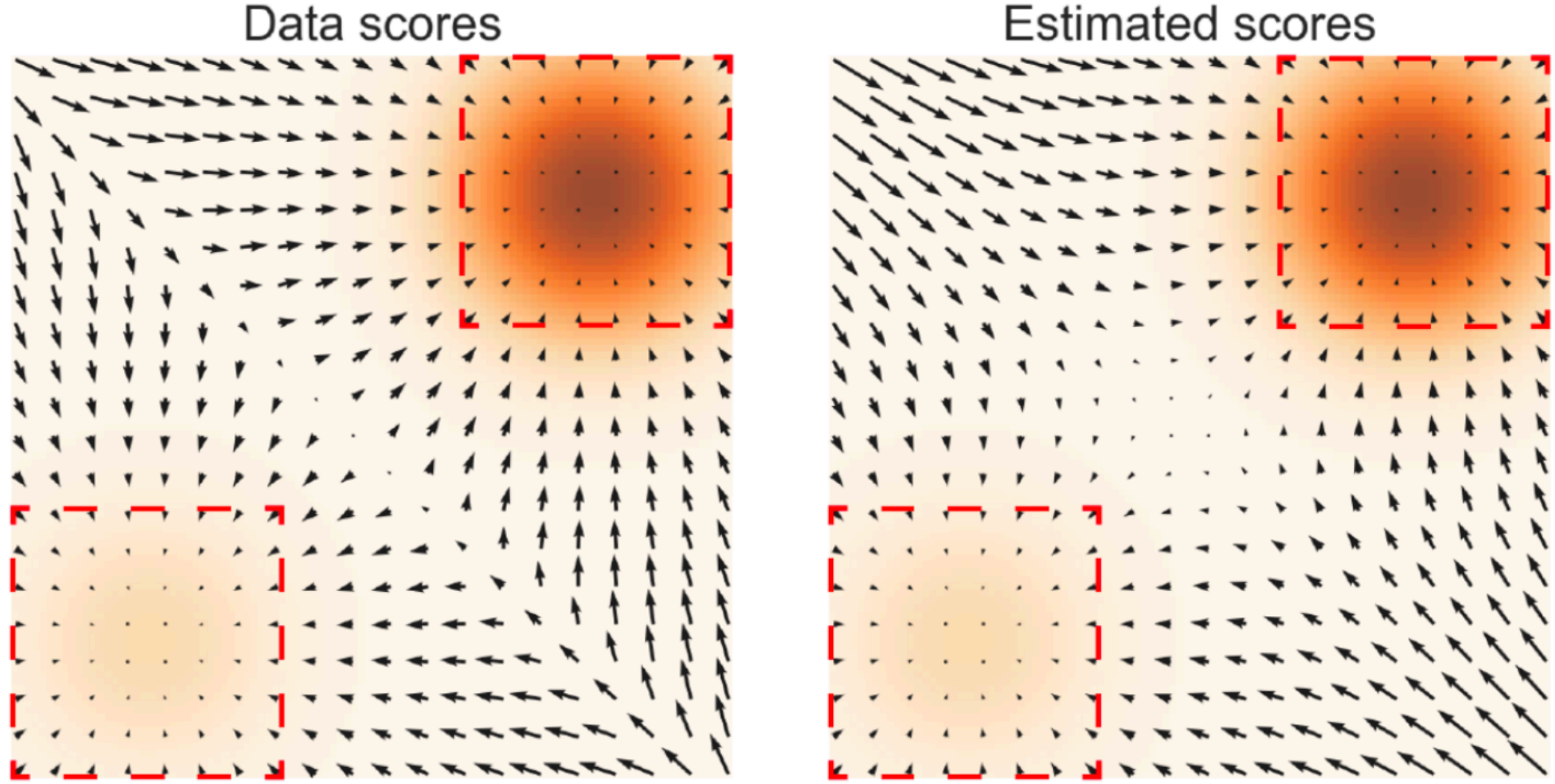


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Motivation for Score-Based Models

Explicit Score Matching

$$\mathscr{L}_{ESM} = \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}) - \mathbf{s}_\theta(\tilde{x})||_2^2\,]$$

- We can use a Parzen window density estimator to fit the empirical distributions.
- q_σ is a Gaussian Kernel.
- $\nabla_{\tilde{x}} q_\sigma(x)$is now computable if $\sigma > 0$ and q_σ is differentiable.

Score-Based Models

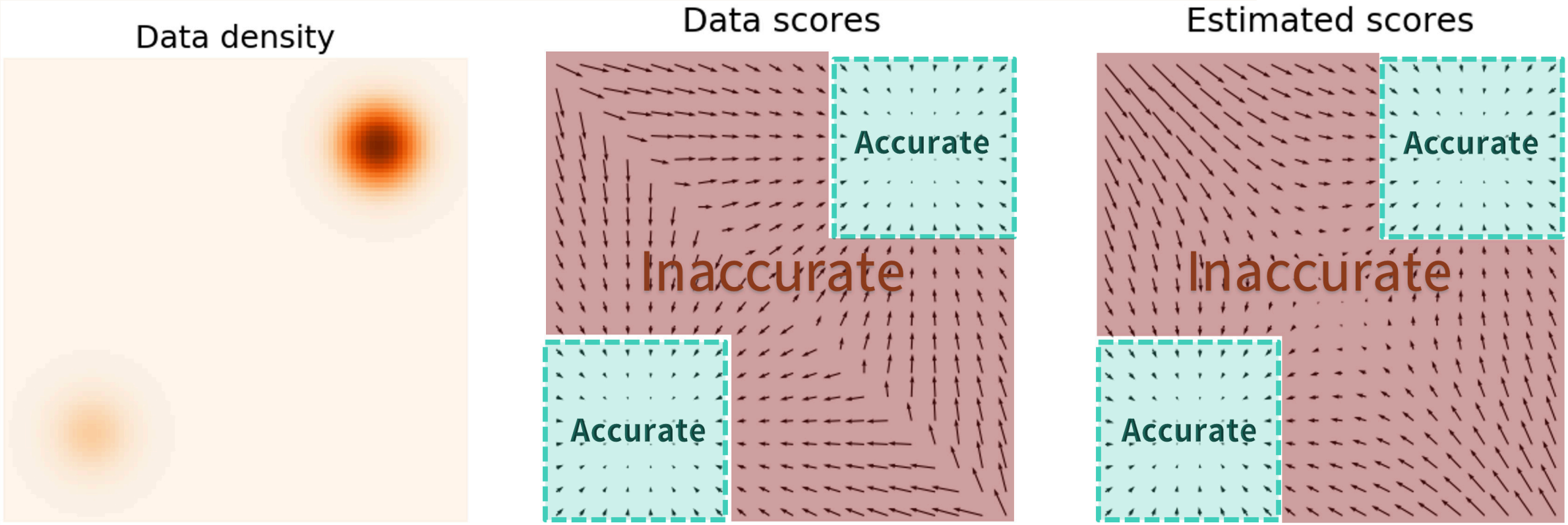


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Motivation for Score-Based Models

Denoising Score Matching

- Perturb clean data with Gaussian noise with a finite number of noise scales L.
- Each noise level $i \in [L]$ defines a corresponding conditional distribution $q_{\sigma_i}(\tilde{x} \mid x)$
- The marginal over the p_{data} is:

$$q_{\sigma_i}(\tilde{x}) = \int q_{\sigma_i}(\tilde{x} \mid x) p_{\text{data}}(x) dx$$

Mila

Score-Based Models

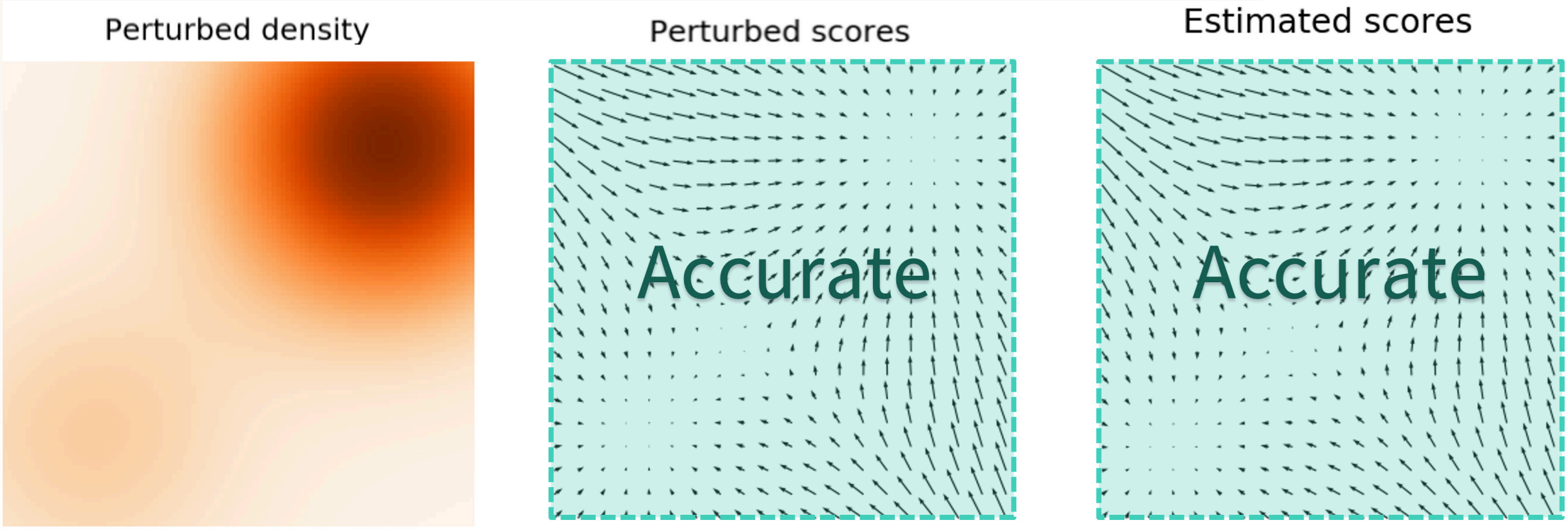


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Motivation for Score-Based Models

Denoising Score Matching Objective

$$\mathscr{L}_{DSM} = \mathbb{E}_{q_\sigma(\tilde{x},x)}[\,||\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x) - \mathbf{s}_\theta(\tilde{x})||_2^2]$$

- Perturb clean data with Gaussian noise with a finite number of noise scales L.
- Each noise level $i \in [L]$ defines a corresponding conditional distribution $q_{\sigma_i}(\tilde{x}\,|\,x)$
- The marginal over the p_{data} is:

Mila

Motivation for Score-Based Models

Denoising Score Matching Objective

$$\mathscr{L}_{DSM} = \mathbb{E}_{q_\sigma(\tilde{x},x)}[\,||\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x) - \mathbf{s}_\theta(\tilde{x})||_2^2]$$

- **Gradient of a log gaussian w.r.t $\tilde{x}$ is** $\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x) = -\frac{1}{\sigma^2}(\tilde{x} - x)$
- **How does the DSM objective relate to the ESM objective?**

Motivation for Score-Based Models

Equivalency of Objectives

$$\mathscr{L}_{ESM} = \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\nabla_{\tilde{x}}\log q_\sigma(\tilde{x}) - \mathbf{s}_\theta(\tilde{x})||_2^2]$$

$$= \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\mathbf{s}_\theta(\tilde{x})||_2^2] - \underbrace{\mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}\left[\langle \nabla_{\tilde{x}}\log q_\sigma(\tilde{x}), \mathbf{s}_\theta(\tilde{x})\rangle\right]}_{\Gamma_\theta}$$

$$+ \underbrace{\mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\nabla_{\tilde{x}}\log q_\sigma(\tilde{x})||_2^2]}_{\text{does not depend on } \theta}.$$

Mila

Motivation for Score-Based Models

Equivalency of Objectives

$$\Gamma_\theta = \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}\left[\left\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}), \mathbf{s}_\theta(\tilde{x})\right\rangle\right]$$

$$= \int q_\sigma(\tilde{x}) \left\langle \frac{\nabla_{\tilde{x}} q_\sigma(\tilde{x})}{q_\sigma(\tilde{x})}, \mathbf{s}_\theta(\tilde{x}) \right\rangle d\tilde{x}$$

Log Gradient Trick

Motivation for Score-Based Models

Equivalency of Objectives

$$\Gamma_\theta = \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}\left[\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}), \mathbf{s}_\theta(\tilde{x})\rangle\right]$$

$$= \int q_\sigma(\tilde{x}) \left\langle \frac{\nabla_{\tilde{x}} q_\sigma(\tilde{x})}{q_\sigma(\tilde{x})}, \mathbf{s}_\theta(\tilde{x}) \right\rangle d\tilde{x} \qquad \textbf{Log Gradient Trick}$$

$$= \int \langle \nabla_{\tilde{x}} q_\sigma(\tilde{x}), \mathbf{s}_\theta(\tilde{x})\rangle d\tilde{x}$$

Mila

Motivation for Score-Based Models

Equivalency of Objectives

$$\Gamma_\theta = \int \langle \nabla_{\tilde{x}} q_\sigma(\tilde{x}), \mathbf{s}_\theta(\tilde{x}) \rangle d\tilde{x}$$

$$= \int \left\langle \nabla_{\tilde{x}} \int q_\sigma(\tilde{x} \mid x) p_{\text{data}}(x) dx, \mathbf{s}_\theta(\tilde{x}) \right\rangle d\tilde{x}$$

$$= \int \left\langle \int \nabla_{\tilde{x}} q_\sigma(\tilde{x} \mid x) p_{\text{data}}(x) dx, \mathbf{s}_\theta(\tilde{x}) \right\rangle d\tilde{x}$$

Swap Gradient and Integral

Motivation for Score-Based Models

Equivalency of Objectives

$$\Gamma_\theta = \int \left\langle \int \nabla_{\tilde{x}} q_\sigma(\tilde{x} \mid x) p_{\text{data}}(x) dx, \mathbf{s}_\theta(\tilde{x}) \right\rangle d\tilde{x}$$

$$= \int \left\langle \int q_\sigma(\tilde{x} \mid x) \nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x) p_{\text{data}}(x) dx, \mathbf{s}_\theta(\tilde{x}) \right\rangle d\tilde{x}$$

Reverse Log-Gradient Trick

$$= \int \int q_\sigma(\tilde{x} \mid x) p_{\text{data}}(x) \left\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x} \mid x), \mathbf{s}_\theta(\tilde{x}) \right\rangle dx d\tilde{x}$$

Mila

Motivation for Score-Based Models

Equivalency of Objectives

$$\Gamma_\theta = \int\int q_\sigma(\tilde{x}\,|\,x) p_{\text{data}}(x) \left\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x), \mathbf{s}_\theta(\tilde{x}) \right\rangle dx d\tilde{x}$$

$$= \int\int q_\sigma(\tilde{x}, x) \left\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x), \mathbf{s}_\theta(\tilde{x}) \right\rangle dx d\tilde{x}$$

$$= \mathbb{E}_{q_\sigma(\tilde{x},x)}[\left\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x), \mathbf{s}_\theta(\tilde{x})] \right\rangle]$$

Mila

Motivation for Score-Based Models

Equivalency of Objectives

$$\mathscr{L}_{ESM} = \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\nabla_{\tilde{x}}\log q_\sigma(\tilde{x}) - \mathbf{s}_\theta(\tilde{x})||_2^2]$$

$$= \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\mathbf{s}_\theta(\tilde{x})||_2^2] - \mathbb{E}_{q_\sigma(\tilde{x},x)}[\left\langle \nabla_{\tilde{x}}\log q_\sigma(\tilde{x}\,|\,x), \mathbf{s}_\theta(\tilde{x})\right]\rangle$$

$$+\underbrace{\mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\nabla_{\tilde{x}}\log q_\sigma(\tilde{x})||_2^2]}_{\text{does not depend on } \theta}.$$

Mila

Motivation for Score-Based Models

Working Backwards and writing the DSM objective

$$\mathscr{L}_{DSM} = \mathbb{E}_{q_\sigma(\tilde{x},x)}[\,||\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x) - \mathbf{s}_\theta(\tilde{x})\,||_2^2]$$

$$= \mathbb{E}_{\tilde{x}\sim q_\sigma(\tilde{x})}[\,||\mathbf{s}_\theta(\tilde{x})\,||_2^2] - \mathbb{E}_{q_\sigma(\tilde{x},x)}[\left\langle \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}\,|\,x), \mathbf{s}_\theta(\tilde{x})\right]\rangle$$

$$+ \underbrace{\mathbb{E}_{q_\sigma(\tilde{x},x)}[\,||\nabla_{\tilde{x}} q_\sigma(\tilde{x}\,|\,x)\,||_2^2]}_{\text{does not depend on } \theta}$$

Ignoring Constants! They are the same!!!

Mila

Sampling in Score-Based Methods

Assume we have a fully trained score model s_θ

- Lets take a fixed step size $\alpha > 0$ and standard noise $\epsilon_t \sim \mathcal{N}(0,I)$
- A sample from our prior distribution $x_0 \sim p(x_0)$

$$x_t = x_{t-1} + \frac{\alpha}{2}\nabla_x \log p_{\text{data}}(x_{t-1}) + \sqrt{\alpha}\epsilon_t$$

Move in the direction of high density

Add noise

Sampling in Score-Based Methods

Assume we have a fully trained score model $\mathbf{s}_\theta$

- The distribution over x_T equals the target $p_{\text{data}}(x)$ when we take $\alpha \to 0$ and $T \to \infty$
- In practice, we run for a finite number of steps $T < \infty$ and do not take $\alpha \to 0$
- Use approximate score given to us by our model $\mathbf{s}_\theta(x) \approx \nabla_x \log p_{\text{data}}(x)$

$$x_t = x_{t-1} + \frac{\alpha}{2}\mathbf{s}_\theta + \sqrt{\alpha}\epsilon_t$$

Mila

Visualizing Langevin Dynamics

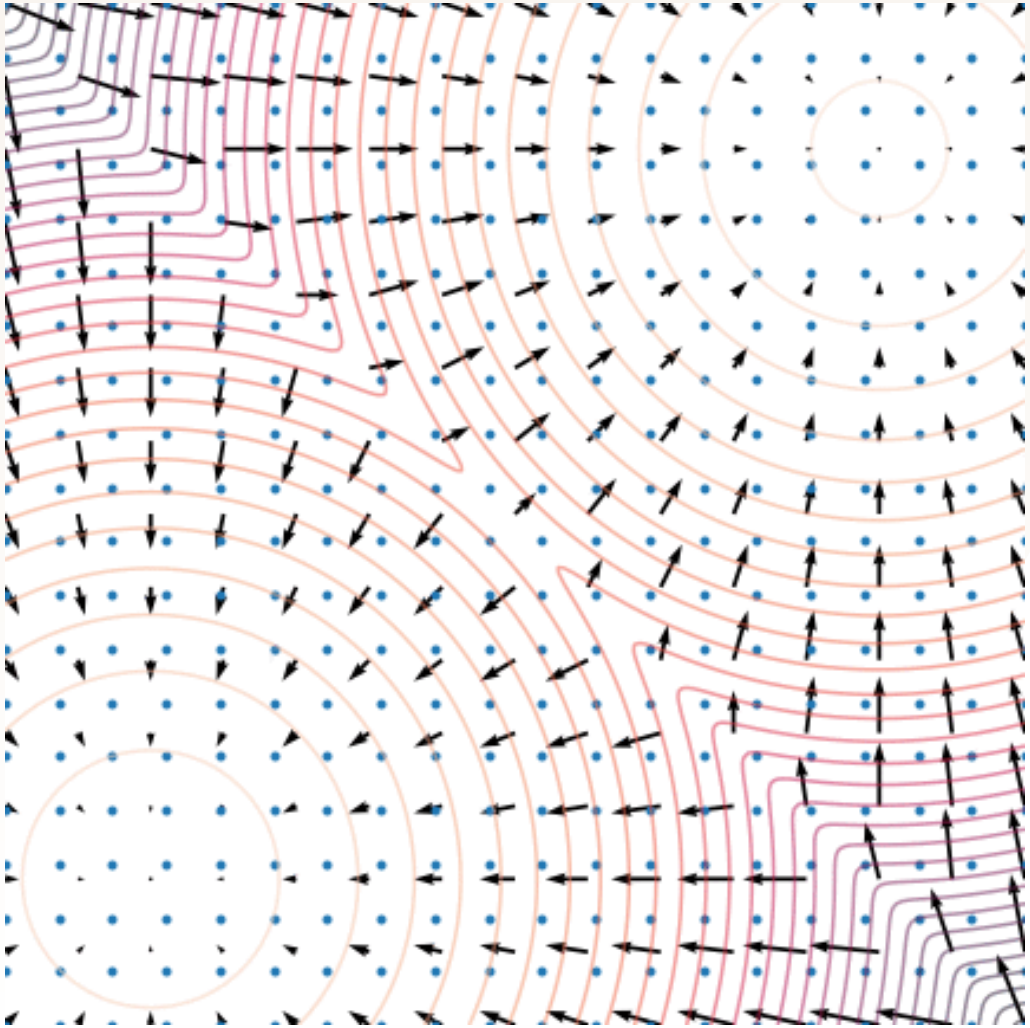


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Noise Conditional Score Matching

DSM objective requires multiple noise scales

- We can incorporate the noise directly into the score network

$$\mathbf{s}_\theta(\tilde{x}, i) \approx \nabla_{\tilde{x}} \log q_{\sigma_i}(\tilde{x})$$

$$\mathscr{L}_{DSM,\sigma_i} = \mathbb{E}_{q_{\sigma_i}(\tilde{x},x)} \left[|| \frac{\tilde{x} - x}{\sigma_i^2} + \mathbf{s}_\theta(\tilde{x}, \sigma_i) ||_2^2 \right]$$

- We can combine this objective across all the noise levels $\sigma_i \in \{\sigma_i\}_{i=1}^L$

$$\mathscr{L} = \frac{1}{L} \sum_{i=1}^{L} \lambda(\sigma_i) \mathscr{L}_{DSM,\sigma_i}, \qquad \lambda(\sigma_i) > 0$$

Mila

Denoising Diffusion Models

Diffusion Models at a Glance

- Diffusion models consists of two processes.
- Forward Diffusion Process gradually corrupts an input.
- Reverse Denoising Process gradually generates a sample.

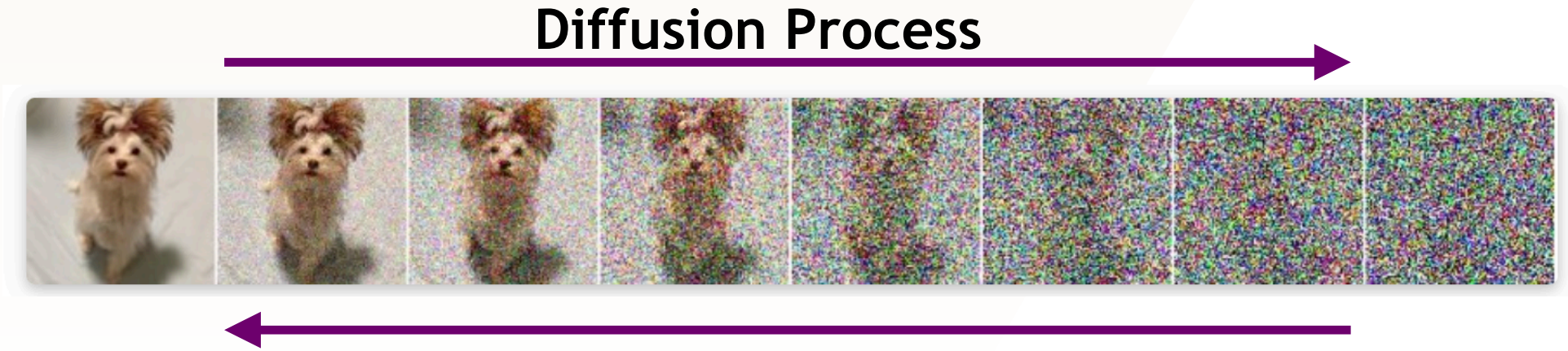


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Diffusion Models at a Glance

Forward Diffusion Process

- Define $q(x_t | x_{t-1})$ as the forward Gaussian centered around x_{t-1}

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I)$$

- β_t is a known variance schedule.
- Mean $\mu_t = \sqrt{1-\beta_t} x_{t-1}$.
- Sampling from this distribution is then simply: $x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon_{t-1}$.

Diffusion Models at a Glance

Forward Diffusion Process

- The joint distribution over all timesteps can be written as follows

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

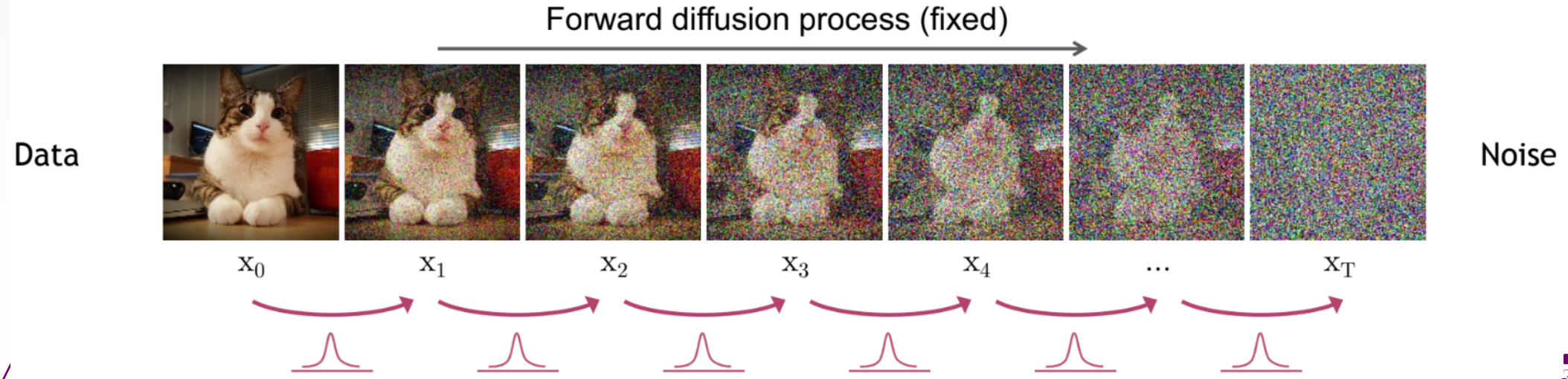


Fig credit: CVPR Diffusion Tutorial Kreis et. Al 2022

Mila

Diffusion Models at a Glance

What if we want to generate a noisy sample at specific timestep?

- We could run the diffusion, progressively adding noise for t steps.
- Or we could exploit that fact that we're using Gaussians and jump directly to t
- Define: $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} (\alpha_i)$

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \epsilon_{t-1}$$

Diffusion Models at a Glance

What if we want to generate a noisy sample at specific timestep?

- **Define:** $\alpha_t = 1 - \beta_t$ **and** $\bar{\alpha}_t = \prod_{i=1}^{t} (\alpha_i)$

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$$

$$= \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon_{t-2}$$ **Unrolling in time**

We used the merging two Gaussian identity $\mathcal{N}(0,\sigma_1^2 I)$ **and** $\mathcal{N}(0,\sigma_2^2 I)$ **as** $\mathcal{N}(0,(\sigma_1^2+\sigma_2^2)I)$ **as follows:**

$$\sqrt{1-\alpha_t + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$$

Diffusion Models at a Glance

What if we want to generate a noisy sample at specific timestep?

- **Define:** $\alpha_t = 1 - \beta_t$ **and** $\bar{\alpha}_t = \prod_{i=1}^{t} (\alpha_i)$

$$
\begin{aligned}
x_t &= \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\epsilon_{t-2} \quad \text{Unrolling in time} \\
&\vdots \\
&= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\alpha_t}\epsilon_0
\end{aligned}
$$

Diffusion Models at a Glance

What if we want to generate a noisy sample at specific timestep?

- A Gaussian at timestep t conditioned on x_0 is given by:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

- The noise schedule is designed such that the terminal conditional is:

$$q(x_T \mid x_0) \approx \mathcal{N}(0, I)$$

Diffusion Models at a Glance

What about the marginal distribution $q(x_t)$?

- We can use our jump gaussian to do this

$$q(x_t) = \int q(x_0, x_t) dx_0 = \int q(x_0) q(x_t \mid x_0) dx_0 \,.$$

Diffusion Models at a Glance

Reverse Denoising Process

- To go backwards we need access to the conditional $q(x_{t-1} \mid x_t)$.

$$q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1}) q(x_{t-1})}{q(x_t)}$$

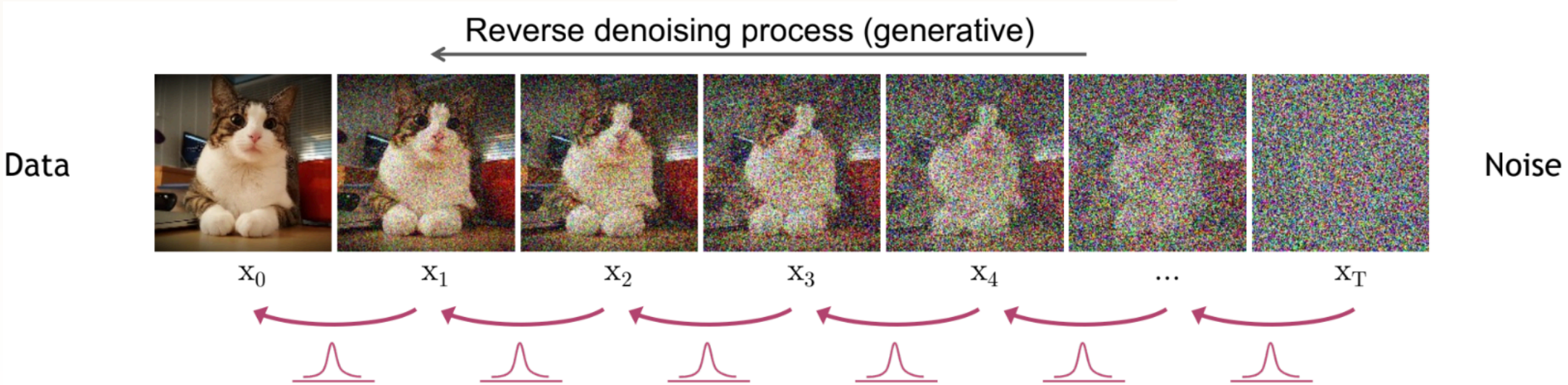


Fig credit: CVPR Diffusion Tutorial Kreis et. Al 2022

Diffusion Models at a Glance

Reverse Denoising Process

- To go backwards we need access to the conditional $q(x_{t-1} \mid x_t)$.

$$q(x_{t-1} \mid x_t) = \frac{q(x_t \mid x_{t-1}) q(x_{t-1})}{q(x_t)}$$

- Recall the marginal at timestep t is:

$$q(x_t) = \int q(x_0, x_t) dx_0 = \int q(x_0) q(x_t \mid x_0) dx_0 \,.$$

- The troublesome term is $q(x_0)$; we only have an empirical distribution.

Diffusion Models at a Glance

Reverse Denoising Process

- If β_t is small enough we know that $q(x_{t-1} \mid x_t) \propto q(x_t \mid x_{t-1}) q(x_{t-1})$ is also a Gaussian distribution.

- We can parametrize $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$ as our generative distribution with learned a mean.

$$p(x_T) = \mathcal{N}(0, I)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$

$$p(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

Mila

Diffusion Models at a Glance

Reverse Denoising Process

- If β_t is small enough we know that $q(x_{t-1} \mid x_t) \propto q(x_t \mid x_{t-1}) q(x_{t-1})$ is also a Gaussian distribution.

- We can parametrize $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$ as our generative distribution with learned a mean.

$$p(x_T) = \mathcal{N}(0, I)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$

$$p(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

Mila

Training Diffusion Models

- **We train by maximizing the log marginal likelihood** $\log p(x_0)$

$$\log p(x_0) = \log \int p(x_{0:T}) dx_{1:T}$$

$$= \log \int p(x_{0:T}) \frac{q(x_{1:T} \mid x_0)}{q(x_{1:T} \mid x_0)} dx_{1:T}$$

Importance Sampling with a chain of conditional Gaussians.

$$= \log \int q(x_{1:T} \mid x_0) p(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1} \mid x_t)}{q(x_t \mid x_{t-1})} dx_{1:T}$$

Mila

Training Diffusion Models

- We can form a variational upper bound like VAEs

$$\log p(x_0) \leq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \left(p(x_T) \prod_{t=1}^{T} \frac{p_\theta(x_{t-1} \,|\, x_t)}{q(x_t \,|\, x_{t-1})} \right) \right] := -\mathscr{L}$$

This will be our starting point to derive the Loss

Training Diffusion Models

- We can form a variational upper bound like VAEs

$$\mathscr{L} = -\,\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log\left(p(x_T)\prod_{t=1}^{T}\frac{p_\theta(x_{t-1}\,|\,x_t)}{q(x_t\,|\,x_{t-1})}\right)\right]$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log p(x_T) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}\,|\,x_t)}{q(x_t\,|\,x_{t-1})} - \log\frac{p_\theta(x_0\,|\,x_1)}{q(x_1\,|\,x_0)}\right]$$

Note the change in time index

Training Diffusion Models

- We can exploit a nice trick by conditioning on x_0

$$\mathscr{L} = \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log p(x_T) - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1}\,|\,x_t)}{q(x_t\,|\,x_{t-1})} - \log \frac{p_\theta(x_0\,|\,x_1)}{q(x_1\,|\,x_0)}\right]$$

$$= \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log p(x_T) - \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1}\,|\,x_t)}{q(x_{t-1}\,|\,x_t, x_0)} \cdot \frac{q(x_{t-1}\,|\,x_0)}{q(x_t\,|\,x_0)} - \log \frac{p_\theta(x_0\,|\,x_1)}{q(x_1\,|\,x_0)}\right]$$

Conditioning

Training Diffusion Models

- We can exploit a nice trick by conditioning on x_0

$$\mathscr{L} = \mathbb{E}_{q(x_{1:T}|x_0)}\left[-\log\frac{p(x_T)}{q(x_T|x_0)} - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - \log p_\theta(x_0|x_1)\right]$$

$$= \mathbb{E}_q\Big[\underbrace{D_{KL}(p(x_T)\,||\,q(x_T|x_0))}_{\mathscr{L}_T}$$

$$- \sum_{t=2}^{T}\underbrace{D_{KL}(p_\theta(x_{t-1}|x_t)\,||\,q(x_{t-1}|x_t,x_0))}_{\mathscr{L}_{t-1}} - \underbrace{\log p_\theta(x_0|x_1)}_{\mathscr{L}_0}\Big]$$

Both these terms are comparisons between 2 Gaussians.

Training Diffusion Models

$$= \mathbb{E}_q \Big[\underbrace{D_{KL}(p(x_T) \,||\, q(x_T|x_0))}_{\mathscr{L}_T} \Big]$$

- Recall $p(x_T) = \mathcal{N}(0, I)$ so this term has no learnable parameters θ.

$$\underbrace{-\log p_\theta(x_0|x_1)}_{\mathscr{L}_0}$$

- Can be treated separately using a discrete decoder $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$

Mila

Training Diffusion Models

- Let's reexamine the main trick we used to compute $\mathscr{L}_{t-1}$.
- Let $q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}, \tilde{\mu}(x_t, x_0), \tilde{\beta}_t)$ be Gaussian conditioned on x_0.

$$\mathcal{Q} := q(x_{t-1} \mid x_t, x_0) = q(x_t \mid x_{t-1}, x_0) \frac{q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)}$$

By Bayes Rule

Training Diffusion Models

- Let's reexamine the main trick we used to compute $\mathscr{L}_{t-1}$.
- Let $q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}, \tilde{\mu}(x_t, x_0), \tilde{\beta}_t)$ be Gaussian conditioned on x_0.

$$\mathcal{Q} \propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t} x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{1-\bar{\alpha}_{t-1}} x_0)^2}{1-\alpha_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t} x_0)^2}{1-\bar{\alpha}_t} \right) \right)$$

Mila

Training Diffusion Models

- Let's reexamine the main trick we used to compute $\mathscr{L}_{t-1}$.
- Let $q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}, \tilde{\mu}(x_t, x_0), \tilde{\beta}_t)$ be Gaussian conditioned on x_0.

$$\mathcal{Q} \propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t} x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{1-\bar{\alpha}_{t-1}} x_0)^2}{1-\alpha_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t} x_0)^2}{1-\bar{\alpha}_t} \right) \right)$$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha_{t-1}}} \right) x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0 \right) x_{t-1} + C(x_t, x_0) \right) \right)$$

Mila

Training Diffusion Models

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right) x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0 \right) x_{t-1} + C(x_t, x_0) \right) \right)$$

$$\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right)$$

$$= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\tilde{\mu}(x_t, x_0) = \left(\frac{\sqrt{\alpha_t}}{\beta_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} \right) \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Factor out $\tilde{\beta}_t$ from above $\mathcal{Q}$

Training Diffusion Models

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right) x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0 \right) x_{t-1} + C(x_t, x_0) \right) \right)$$

$$\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right) \qquad \tilde{\mu}(x_t, x_0) = \left(\frac{\sqrt{\alpha_t}}{\beta_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} \right) \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$$

$$= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \qquad = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0$$

Training Diffusion Models

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right) x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0 \right) x_{t-1} + C(x_t, x_0) \right) \right)$$

$$\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right)$$

$$= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\tilde{\mu}(x_t, x_0) = \left(\frac{\sqrt{\alpha_t}}{\beta_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} \right) \cdot \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0$$

$$= \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right)$$

Mila

Training Diffusion Models

We can now rewrite the per-tilmestep loss $\mathscr{L}_{t-1}$

$$\mathscr{L}_{t-1} = \frac{1}{2}\mathbb{E}_q \left[\frac{1}{\sigma_t^2} || \tilde{\mu}(x_t, x_0) - \mu_\theta(x_t, t) ||^2 \right] + C$$

We basically compare the means of two Gaussians.

Training Diffusion Models

Does this suggest a parametrization we could exploit?

Instead of mean prediction we could do noise level prediction

$$\mu_\theta = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$

Noise prediction network

Training Diffusion Models

Putting it all together

$$\mathscr{L}_{t-1} = \frac{1}{2}\mathbb{E}_q\left[\frac{1}{\sigma_t^2}\left|\left|\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)\right|\right|^2\right]$$

$$= \frac{1}{2}\mathbb{E}_{x\sim q(x_0),\epsilon_t}\left[\underbrace{\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}}_{\lambda_t}\left|\left|\epsilon_t - \epsilon_\theta(x_t, t)\right|\right|^2\right]$$

Weighting Term

Mila

Training Diffusion Models

Noise prediction network

Different choices for Weighting terms

$$\mathscr{L}_{t-1} = \frac{1}{2}\mathbb{E}_{x\sim q(x_0),\epsilon_t}\left[\underbrace{\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}}_{\lambda_t}||\epsilon_t - \epsilon_\theta(x_t, t)||^2\right]$$

$$\mathscr{L}_{\text{simple}} = \frac{1}{2}\mathbb{E}_{x\sim q(x_0),\epsilon_t}\left[||\epsilon_t - \epsilon_\theta(x_t, t)||^2\right]$$

Generates higher quality samples

Mila

Training Diffusion Models

Training Algorithm

Algorithm 1 Training

while not converged **do**

$x_0 \sim q(x_0)$

$t \sim \text{Uniform}(\{0, \dots, T\})$

$\epsilon \sim \mathcal{N}(0, I)$

$\nabla_\theta \mathcal{L}_t = \nabla_\theta \left[\lambda_t || \epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) ||^2 \right]$

end while

Mila

Training Diffusion Models

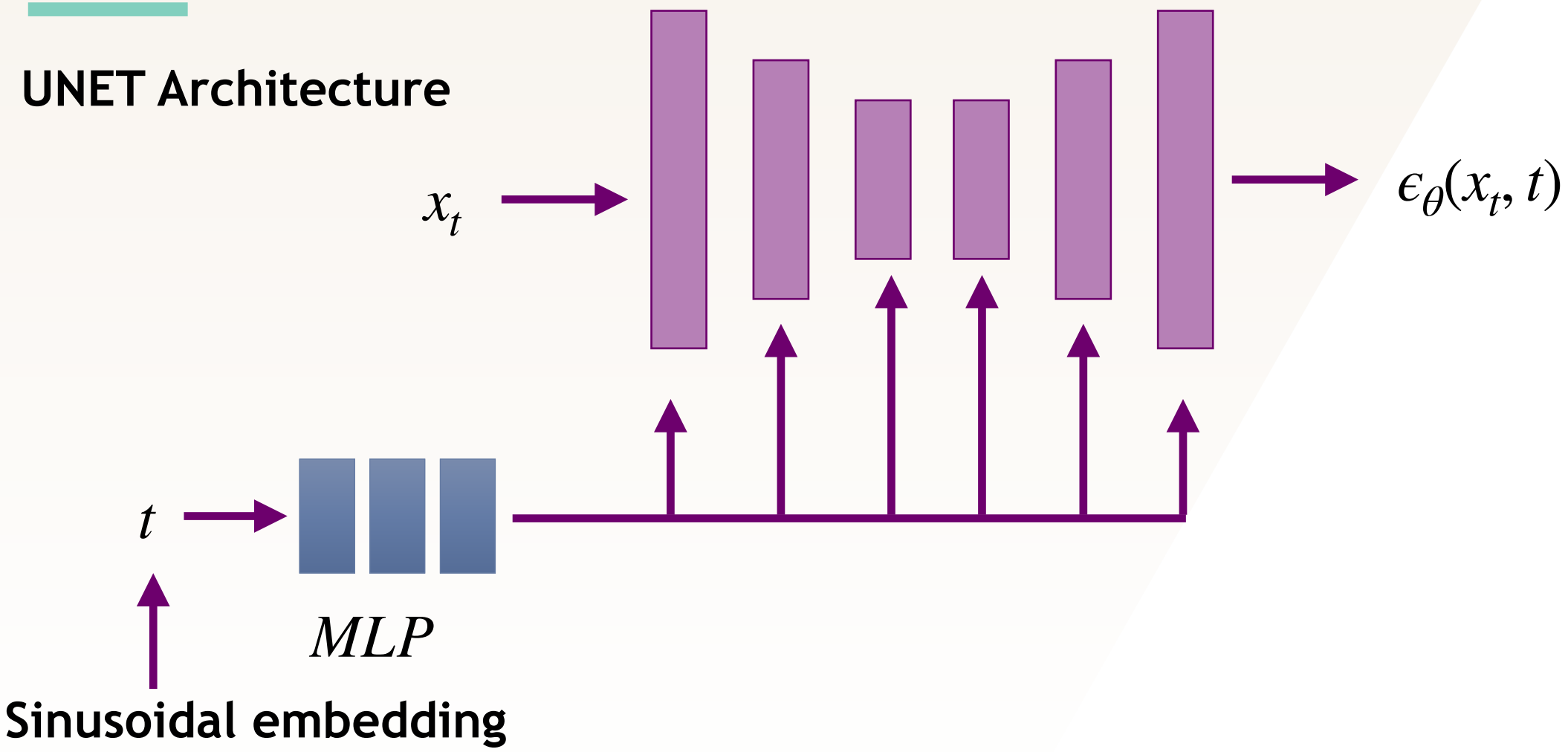
Sampling Algorithm

Algorithm 2 Sampling

$x_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 1$ **do**
  $\epsilon \sim \mathcal{N}(0, I)$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\epsilon(x_t, t)} \right) + \sigma_t \epsilon$
end for

Mila

Implementation Considerations



Understanding the Forward Diffusion Process

- A Gaussian at timestep t conditioned on x_0 is:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \alpha_t} \epsilon_0$$

Fourier transform

$$\mathcal{F}(x_t) = \sqrt{\bar{\alpha}_t} \mathcal{F}(x_0) + \sqrt{1 - \alpha_t} \mathcal{F}(\epsilon_0)$$

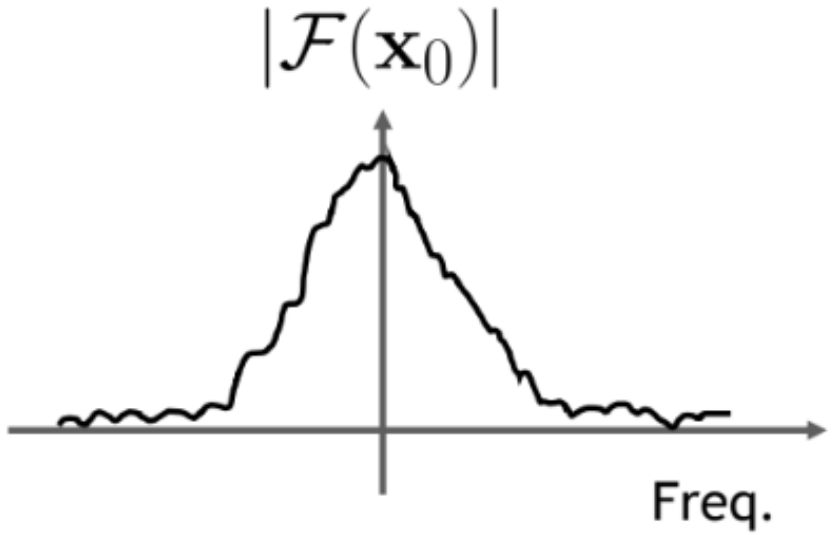


Fig credit: CVPR Diffusion Tutorial Kreis et. Al 2022

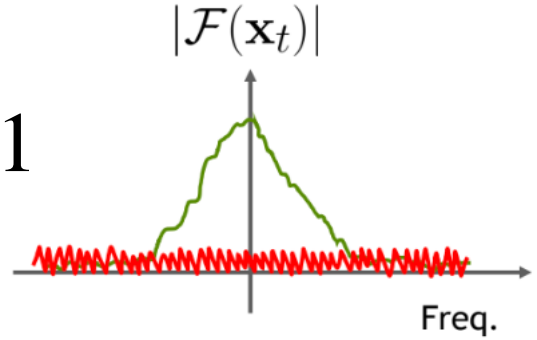
Mila

Understanding the Forward Diffusion Process

- **A Gaussian at timestep t conditioned on x_0 is:**

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

small $t; \bar{\alpha}_t \approx 1$



$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \alpha_t} \epsilon_0$$

Fourier transform

$$\mathcal{F}(x_t) = \sqrt{\bar{\alpha}_t} \mathcal{F}(x_0) + \sqrt{1 - \alpha_t} \mathcal{F}(\epsilon_0)$$

Large $t; \bar{\alpha}_t \approx 0$

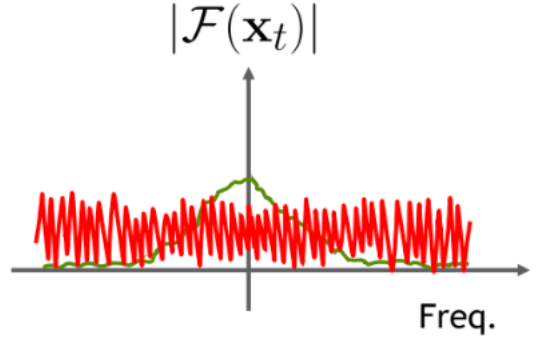


Fig credit: CVPR Diffusion Tutorial Kreis et. Al 2022

Mila

Connection to VAEs

- Diffusion models can be thought of as a hierarchical VAE.
- The Encoder is fixed and not learnable!
- The latents have the same dimensionality as the data.
- The denoising model is shared across all timesteps.
- Training is done with a weighted Variational Bound.

Annotated Google Collab

[Diffusion Google Collab]

Try it Out!

Improving Sampling with Guidance

- Can we improve the sample quality of a pre-trained diffusion model maybe trading off diversity (e.g. low temperature sampling)?
- Method #1 on improving sample quality: Classifier based guidance by training an offline classifier and using it with conditional model.
- Method #2 on improving sample quality: Classifier-free guidance by simultaneously training an unconditional and conditional model.

Classifier-Guidance

- Intuitive idea: low temperature sampling by decreasing the variance or range of noise inputs to the generative model at sampling time.
- Introduce an external classifier: $p_\theta(x_t \mid c)$ which is trained offline and w is a weighting term.

$$\tilde{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t, c) - w\sigma_t \nabla_{x_t} \log p(c \mid x_t)$$

Modify the diffusion score by taking into account the classifiers gradient

Mila

Classifier-Guidance

- Intuitive idea: low temperature sampling by decreasing the variance or range of noise inputs to the generative model at sampling time.
- Introduce an external classifier: $p_\theta(x_t \mid c)$ which is trained offline

$$\tilde{\epsilon}_\theta(x_t, c) = \epsilon_\theta(x_t, c) - w\sigma_t \nabla_{x_t} \log p(c \mid x_t)$$

$$\approx -\sigma_t \nabla_{x_t} [\log p(x_t \mid c) + w \log p_\theta(c \mid x_t)]$$

- We are now sampling from $\tilde{p}_\theta(x_t \mid c) \propto p_\theta(x_t \mid c) p_\theta(c \mid x_t)^w$
- $p_\theta(c \mid x_t)$ assigns high likelihood to the correct label; rewarding the generative model for generating samples that are more realistic to the class.

Mila

Classifier Free-Guidance

- Can we achieve the same results as classifier based guidance without training another extra model?
- Train two diffusion models: unconditional and conditional giving scores $\epsilon_\theta(x_t)$ and $\epsilon_\theta(x_t \mid c)$ respectively.
- Use a single model to parametrize both unconditional and conditional models as follows:

$$\epsilon_\theta(x_t) = \epsilon_\theta(x_t \mid c = \varnothing)$$

Null-Token to signify unconditional model

Mila

Classifier Free-Guidance

New Diffusion Scores

$$\tilde{\epsilon}_\theta(x_t, c) = (1 + w)\epsilon_\theta(x_t, c) - w\epsilon_\theta(x_t)$$

- No classifier gradient present. Cannot be interpreted with the same exact intuition because we are using unconstrained neural networks.
- Intuition: $p(c \mid x_t) \propto p(x_t \mid c)/p(x_t)$. If we had exact scores $\epsilon_\theta(x_t, c)$ and $\epsilon_\theta(x_t)$ then the gradient would look like:

$$\nabla_{x_t} p(c \mid x_t) = -\frac{1}{\sigma_t}[\epsilon_\theta(x_t, c) - \epsilon_\theta(x_t)]$$

- Plugging this back into the classifier guidance equation we saw we get the above score $\tilde{\epsilon}_\theta(x_t, c)$.

Mila

Classifier Free-Guidance Updated Training

Algorithm 2 Conditional sampling with classifier-free guidance

Require: w: guidance strength
Require: $\mathbf{c}$: conditioning information for conditional sampling
Require: $\lambda_1, \ldots, \lambda_T$: increasing log SNR sequence with $\lambda_1 = \lambda_{\min}, \lambda_T = \lambda_{\max}$

1: $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = 1, \ldots, T$ **do**
▷ Form the classifier-free guided score at log SNR λ_t
3: $\tilde{\boldsymbol{\epsilon}}_t = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)$
▷ Sampling step (could be replaced by another sampler, e.g. DDIM)
4: $\tilde{\mathbf{x}}_t = (\mathbf{z}_t - \sigma_{\lambda_t}\tilde{\boldsymbol{\epsilon}}_t)/\alpha_{\lambda_t}$
5: $\mathbf{z}_{t+1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda_{t+1}|\lambda_t}(\mathbf{z}_t, \tilde{\mathbf{x}}_t), (\tilde{\sigma}^2_{\lambda_{t+1}|\lambda_t})^{1-v}(\sigma^2_{\lambda_t|\lambda_{t+1}})^v)$ if $t < T$ else $\mathbf{z}_{t+1} = \tilde{\mathbf{x}}_t$
6: **end for**
7: **return** $\mathbf{z}_{T+1}$

Fig credit: https://arxiv.org/pdf/2207.12598.pdf

Mila

Example Samples with Classifier Free Guidance

Fig credit: https://arxiv.org/pdf/2207.12598.pdf

Mila

Latent Diffusion

Training and sampling in Pixel Space can be very expensive

- Departure to latent space!!!
- First train a powerful auto encoder on Pixel Space.
- Do Diffusion in latent Space and use decoder to generate in pixel space.

$$\mathscr{L}_{\text{simple}} = \frac{1}{2}\mathbb{E}_{\mathscr{E}(x),\epsilon_t}\left[\,||\,\epsilon_t - \epsilon_\theta(z_t, t)\,||^2\right]$$

Mila

Stable Diffusion

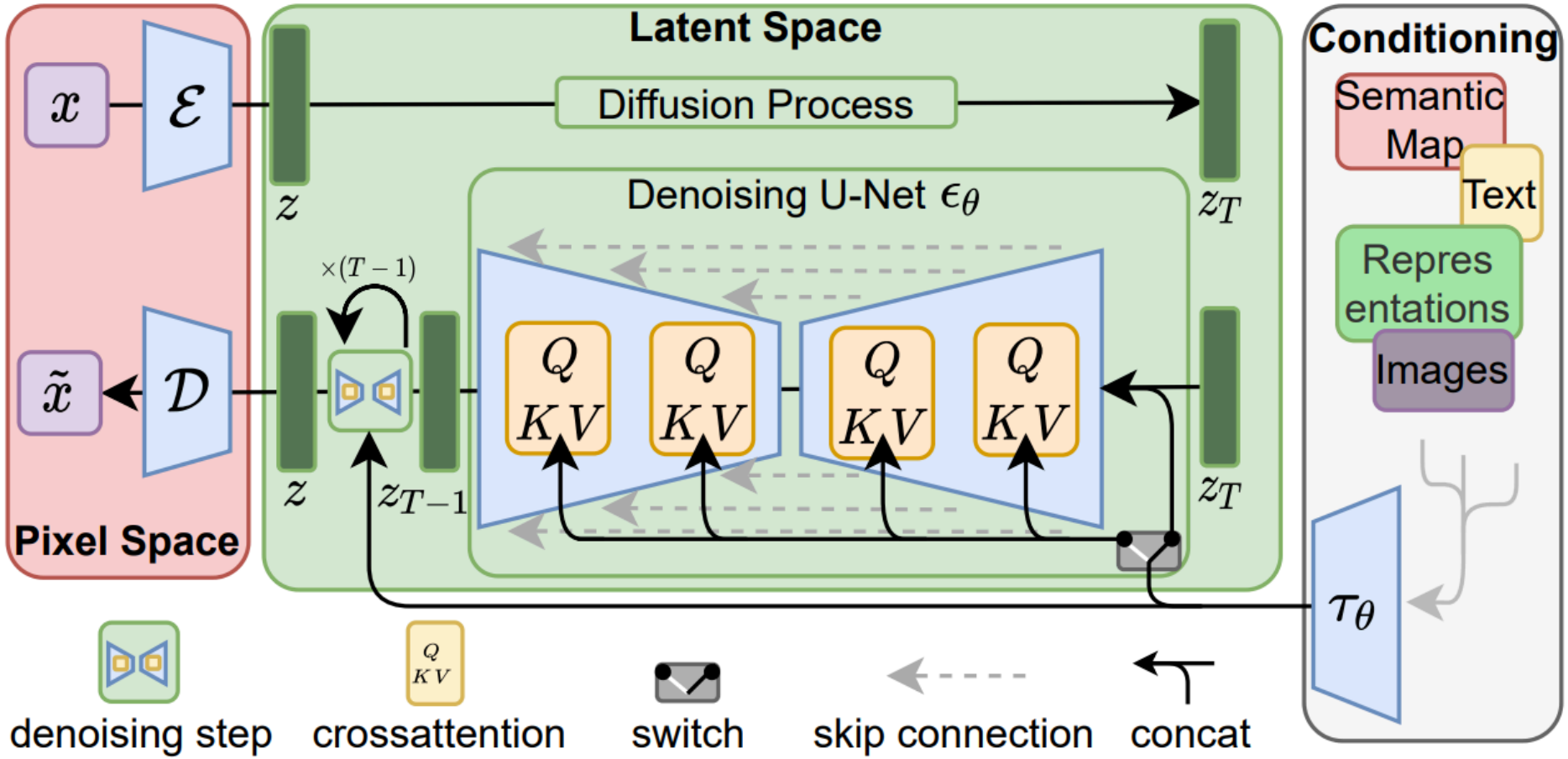


Fig credit: https://arxiv.org/pdf/2112.10752.pdf

Mila

Continuous Time Diffusion Models

Diffusion and Score Based Generative Models are really two sides of the same coin.

Diffusion Models at the continuous time limit

Diffusion Models as Stochastic Differential Equations

- What happens when we take the finite noise scales to its limit $L \to \infty$?
- We get a stochastic differential equation which can generally be written as follows:

$$dx = \mu dt + \sigma dB_t$$

μdt: Drift Coefficient

σdB_t: Diffusion Coefficient

dB_t: Brownian Motion

Diffusion Models at the continuous time limit

ODEs

$$dx = \mu dt$$

$$x_t = x_0 + \int_0^t \mu(x_\tau) d\tau$$

Analytic Solution

$$x_{t+\Delta t} \approx x_t + \mu(x_t)\Delta t$$

Iterative Solution

SDEs

$$dx = \mu dt + \sigma dB_t$$

$$x_{t+\Delta t} \approx x_t + \mu(x_t)\Delta t + \mu(x_t)\sqrt{\Delta t}\mathcal{N}(0,I)$$

Iterative Solution

Diffusion Models at the continuous time limit

Forward Diffusion

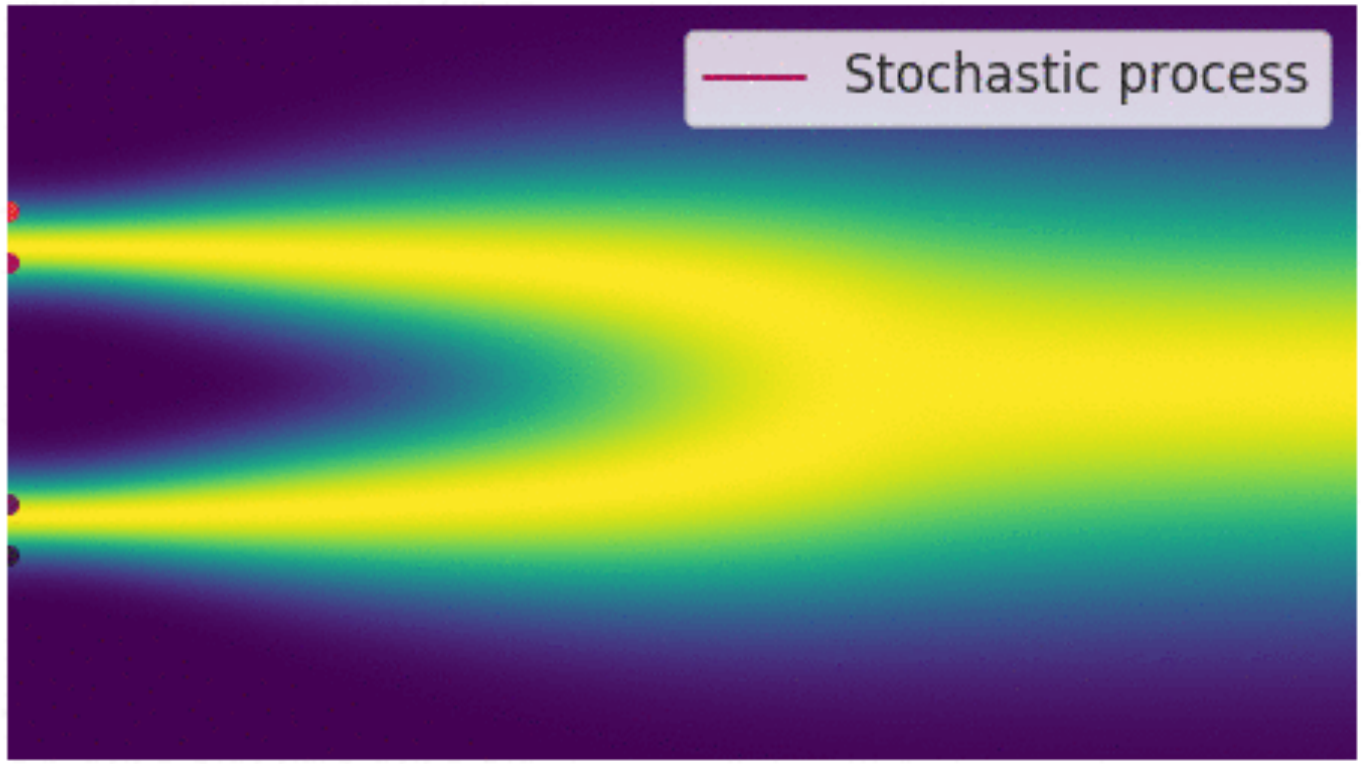


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Diffusion Models at the continuous time limit

Training Objectives

- **Noise Conditional Score Matching in Continuous time.**

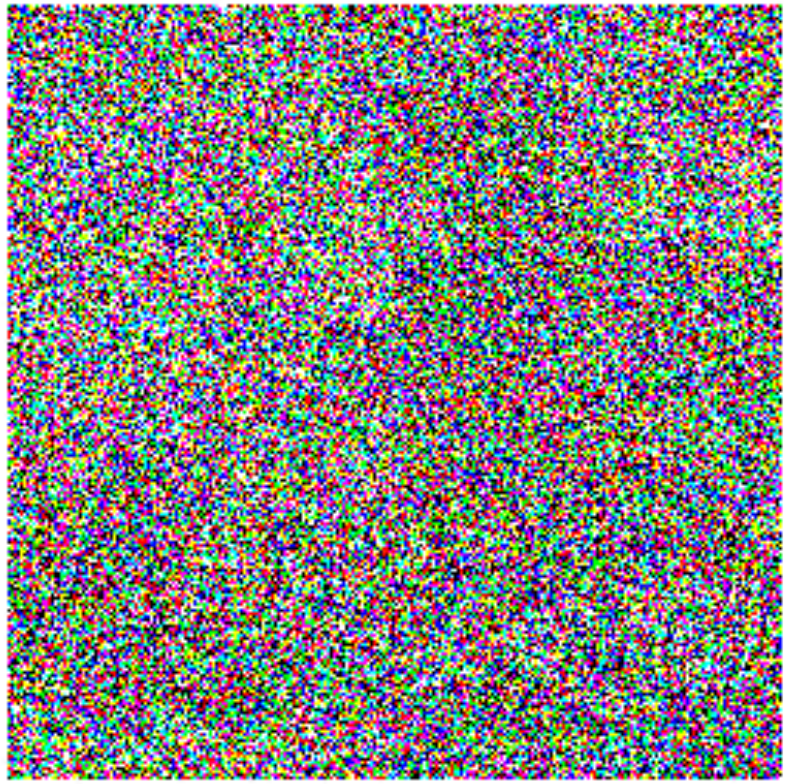
$$\mathscr{L} = \frac{1}{2}\mathbb{E}_{t\sim\text{Unif}(0,T),p(x_t)}\left[\lambda_t||\mathbf{s}_\theta(x_t,t) - \nabla_x \log p(x_t)||_2^2\right]$$

- **Denoising Score Matching in Continuous time.**

$$\mathscr{L}_{DSM} = \frac{1}{2}\mathbb{E}_{t\sim\text{Unif}(0,T),p(x_0)p(\tilde{x}_t|x_0)}\left[\lambda_t||\mathbf{s}_\theta(\tilde{x}_t,t) - \nabla_x \log p(\tilde{x}_t|x_0)||_2^2\right]$$

Diffusion Models at the continuous time limit

Reverse Diffusion



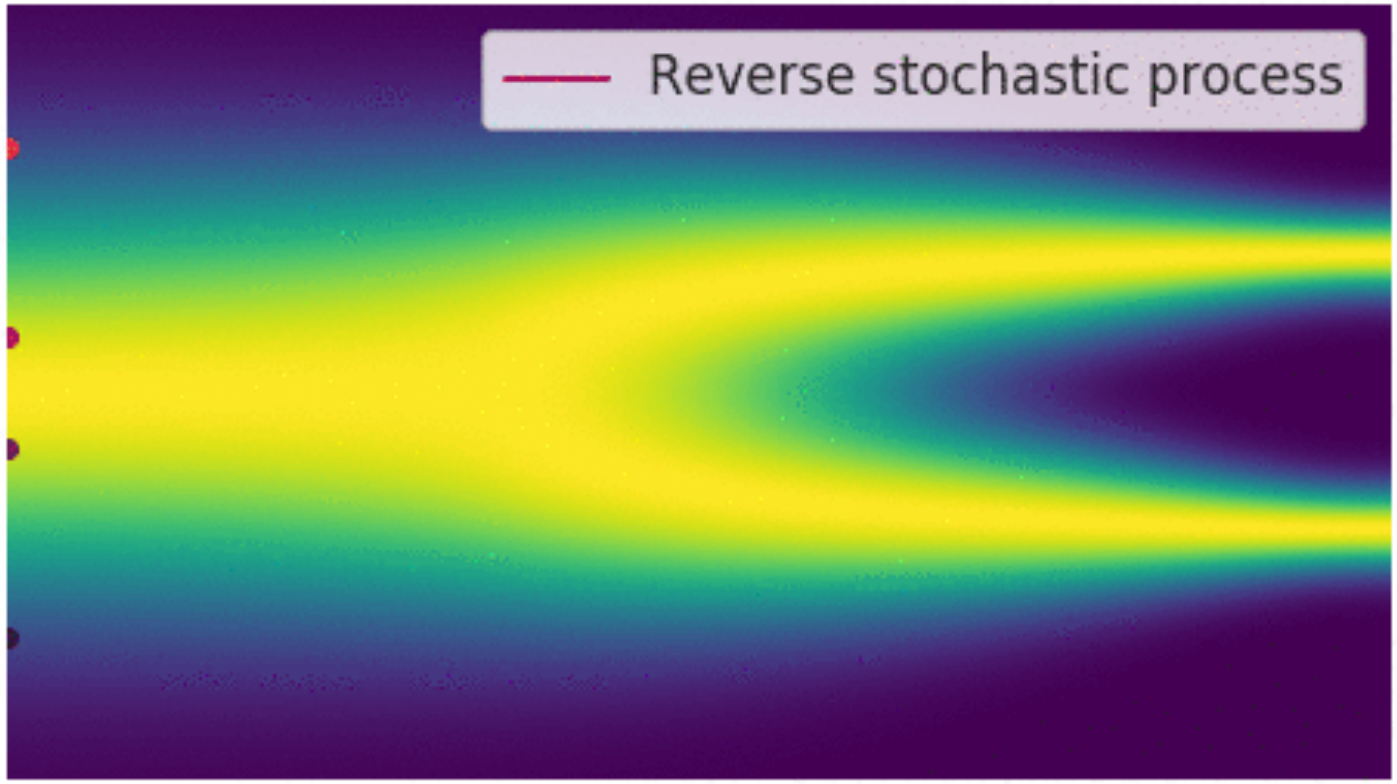


Fig credit: https://yang-song.net/blog/2021/score/

Mila

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

- We will derive a lower bound to this like before.

- The instantaneous change in density is given by the Fokker-Plank equation.

$$\partial_t p(x_t) = -\sum_j \partial x_t^j [\mu^j(x_t) p(x_t)] + \sum_{i,j} \partial^2 x^i x^j [D_{i,j}(x_t) p(x_t)]$$

x^j is the j-th element of the vector

$$D = \frac{1}{2}\sigma\sigma^T$$

Mila

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

- We will derive a lower bound to this like before.

- The instantaneous change in density is given by the Fokker-Plank equation.

$$\partial_t p(x_t) = -\sum_j \partial x_t^j [\mu^j(x_t) p(x_t)] + \sum_{i,j} \partial^2 x^i x^j [D_{i,j}(x_t) p(x_t)]$$

x^j is the j-th element of the vector

$$D = \frac{1}{2}\sigma\sigma^T$$

Mila

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

- With some algebra magic and grouping terms.

$$\partial_t p(x_t) = -(\nabla \cdot \mu(x_t))p(x_t) - \mu(x_t)^T \nabla p(x_t) + \langle D, H_p(x_t) \rangle_F$$

$\nabla \cdot$ is the Euclidean divergence

$\langle \cdot, \cdot \rangle_F$ Frobenius norm between matrices

Hessian

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

- **This equation admits a probabilistic representation by the *Feynmann-Kac representation:***

$$\partial_t p(x_t) = -(\nabla \cdot \mu(x_t))p(x_t) - \mu(x_t)^T \nabla p(x_t) + \langle D, H_p(x_t) \rangle_F$$

$$p(x_T) = \mathbb{E}\left[p_0(y_T) \exp\left(\int_0^T -\nabla \cdot \mu_{T-s}(y_s) ds \right) \middle| y_0 = x \right]$$

Mila

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

- This equation admits a probabilistic representation by the *Feynmann-Kac representation:*

$$p(x_T) = \mathbb{E}\left[p_0(y_T) \exp\left(\int_0^T -\nabla \cdot \mu_{T-s}(y_s) ds \right) \middle| \, y_0 = x \right]$$

- y_s follows the diffusion SDE (the generative coefficients are evaluated in reversed time i.e. $T - s$).

$$dy = -\mu_{T-s}(y)ds + \sigma_{T-s}\, d\hat{B}_s$$

Variational Framework for Continuous Diffusion

Intuitive Interpretation

- This can be thought of as a mixture of Continuous Normalizing Flows.

$$p(x_T) = \mathbb{E}\left[p_0(y_T) \exp\left(\int_0^T -\nabla \cdot \mu_{T-s}(y_s) ds \right) \middle| y_0 = x \right]$$

- The diffusion coefficient is independent of the spatial variable x
- The sole contributor to the change in density is thus $x \to x + \mu \Delta t$

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

- The current problem is that we would have to integrate out all Brownian motion paths, which is intractable.
- We can view Brownian motion as an infinite dimensional latent variable. This means we are working with an infinite dimensional VAE.

$$\log p(x_T) \geq \mathbb{E}_{\mathbb{Q}} \left[\log \frac{d\mathbb{P}}{d\mathbb{Q}} + \log p_0(y_T) - \int_0^T \nabla \cdot \mu_{T-s}(y_s) ds \,\middle|\, y_0 = x \right]$$

Variational Framework for Continuous Diffusion

We want an expression for $\log p(x_0)$

$$\log p(x_T) \geq \mathbb{E}_{\mathbb{Q}}\left[\log \frac{d\mathbb{P}}{d\mathbb{Q}} + \log p_0(y_T) - \int_0^T \nabla \cdot \mu_{T-s}(y_s) ds \,\middle|\, y_0 = x\right]$$

Change of measure

Still how do we compute this? This is an infinite dimensional object.

Mila

Variational Framework for Continuous Diffusion

Girsanov Theorem

Theorem 3.2.1 (**Girsanov theorem**, Theorem 8.6.3 of Øksendal [2003])
Let $\hat{B}_s$ be an Itô process solving

$$\mathrm{d}\hat{B}_s = -a(\omega, s)\,\mathrm{d}s + \mathrm{d}B_s, \tag{3.71}$$

A bunch of math jargon skipped

$$\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}}(\omega) := \exp\left(\int_0^T a(\omega, s)\cdot \mathrm{d}B_s - \frac{1}{2}\int_0^T ||a(\omega, s)||_2^2\,\mathrm{d}s\right).$$

Variational Framework for Continuous Diffusion

Continuous-Time ELBO

$$\log p(x_T) \geq \mathbb{E}\left[\log p_0(y_T) - \int_0^T \left(\frac{1}{2}||a(y_s, s)||_2^2 + \nabla \cdot \mu_{T-s}(Y_s)\right) ds \,\middle|\, y_0 = x\right]$$

a is the variational degree of freedom

- The relationship between a and a score network is thus $a = \sigma^T \mathbf{s}_\theta$

Probability Flow ODE

Continuous-Time ELBO

- What if we want to compute the exact log density rather than a lower bound?
- A probability flow ODE which has the same marginal $p(x_t)$ can be extracted from an SDE.

$$\frac{dx}{dt} = \mu(x_t) - \frac{1}{2}\sigma\sigma^T \nabla_x \log p_t(x_t)$$

Probability Flow ODE

Continuous-Time ELBO

- What if we want to compute the exact log density rather than a lower bound?
- A probability flow ODE which has the same marginal $p(x_t)$ can be extracted from an SDE.

$$\frac{dx}{dt} = \mu(x_t) - \frac{1}{2}\sigma\sigma^T \nabla_x \log p_t(x_t) \longrightarrow \frac{dx}{dt} = \mu(x_t) - \frac{1}{2}\sigma\sigma^T \mathbf{s}_\theta(x_t, t)$$

- This is exactly a CNF and we can use the instantaneous change of variable formula to get the log density.

Things we didn't talk about

- Generating samples using SDE solvers.
- Accelerating sampling and 2nd order Samplers.
- Variance Reduction using Importance Sampling.

Advanced Topics: Cold Diffusion

Do we really need Gaussian noise?

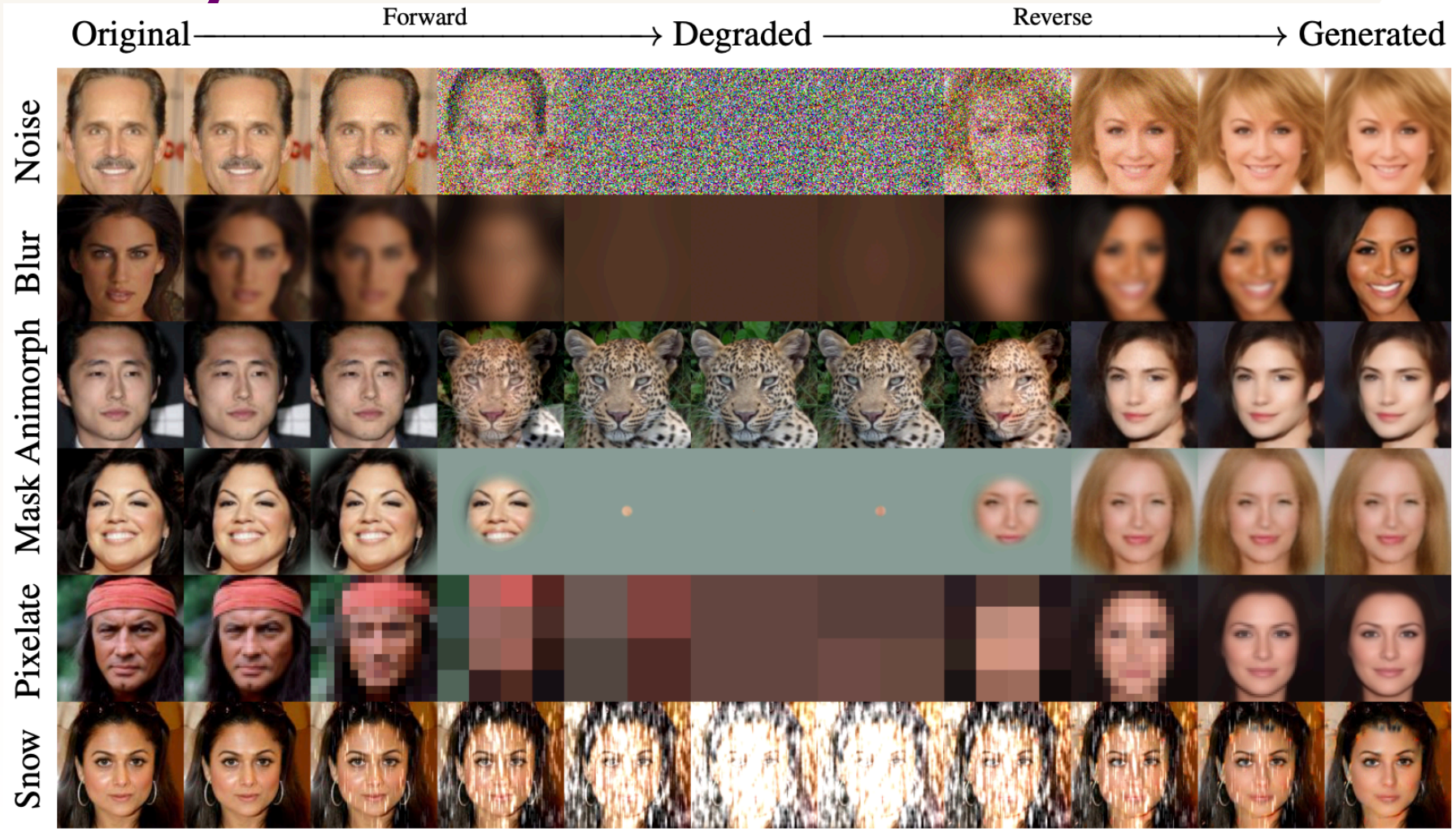


Fig credit: https://arxiv.org/pdf/2208.09392.pdf

Mila

Advanced Topics: Cold Diffusion

Do we really need Gaussian noise?

- $x \in \mathbb{R}^n$ and define a degradation operator D with severity t.

$$x_t = D(x_0, t)$$

- Define a restoration operator R that restores an image.

$$R(x_t, t) \approx x_0$$

$$\min_\theta \mathbb{E}_{x \sim p(x)}[||R_\theta(D(x,t)) - x||]$$

Training Objective

Advanced Topics: Cold Diffusion

Do we really need Gaussian noise?

Algorithm 2 Improved Sampling for Cold Diffusion

Input: A degraded sample x_t
for $s = t, t-1, \ldots, 1$ **do**
  $\hat{x}_0 \leftarrow R(x_s, s)$
  $x_{s-1} = x_s - D(\hat{x}_0, s) + D(\hat{x}_0, s-1)$
end for

Mila

Advanced Topics: Cold Diffusion

What if D and R are not perfect inverses?

$$D(x, s) \approx x + s \cdot e + HOT$$ → Taylor Expansion

$$\hat{x}_0 \leftarrow R(x_s, s)$$
$$x_{s-1} = x_s - D(\hat{x}_0, s) + D(\hat{x}_0, s-1)$$

↓

$$x_{s-1} = x_s - D(R(x_s, s), s) + D(R(x_s, s), s-1)$$

/

Mila

Advanced Topics: Cold Diffusion

What if D and R are not perfect inverses?

$$D(x, s) \approx x + s \cdot e + HOT \longrightarrow \text{Taylor Expansion}$$

$$\begin{aligned} x_{s-1} &= x_s - D(R(x_s, s), s) + D(R(x_s, s), s-1) \\ &= D(x_0, s) - D(R(x_s, s), s) + D(R(x_s, s), s-1) \\ &= x_0 + s \cdot e - R(x_s, s) - s \cdot e + R(x_s, s) + (s-1) \cdot e \\ &= x_0 + (s-1) \cdot e \\ &= D(x_0, s-1) \end{aligned}$$

Mila