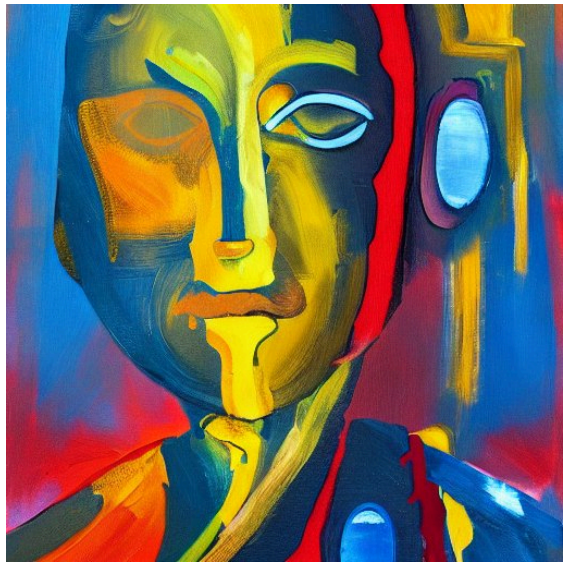
Diffusion Models

David I. Inouye

Diffusion Models Have Become State-Of-The-Art for Generative Modeling

- See demo: https://huggingface.co/spaces/stabilityai/stable-diffusion



Abstract painting of an artificial intelligent agent

The text “Purdue” on an indiana university jersey

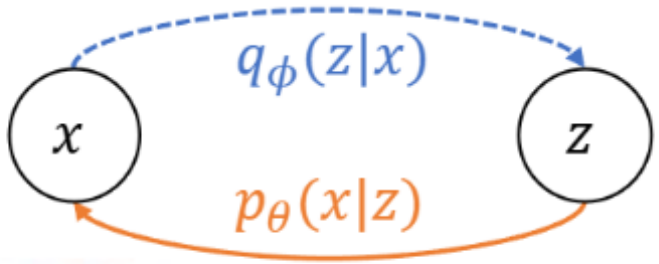
Overview

- **Model**
  - Diffusion models as hierarchical VAEs with fixed encoders
- **Training**
  - Perspective 1: Reweighted joint ELBO
  - Perspective 2: Multiple VAE ELBOs with shared parameters
  - Perspective 3: Multiple denoising AEs with shared parameters
- **Sampling**
  - VAE-based Markov sampling (DDPM)
  - Implicit (deterministic) sampling (DDIM)

Model: Diffusion Models Define Forward and Reverse Diffusion Processes

- Diffusion models can be viewed as hierarchical VAEs
  - **Forward process** = hierarchical encoder
  - **Reverse process** = hierarchical decoder
- Several critical differences from VAE
  - Involves **multiple latent representations** rather than one
  - Hierarchical encoder is **fixed**
  - **Parameters θ are shared** between decoder steps

Simple VAE (Same-Dimension)



Hierarchical Encoder

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

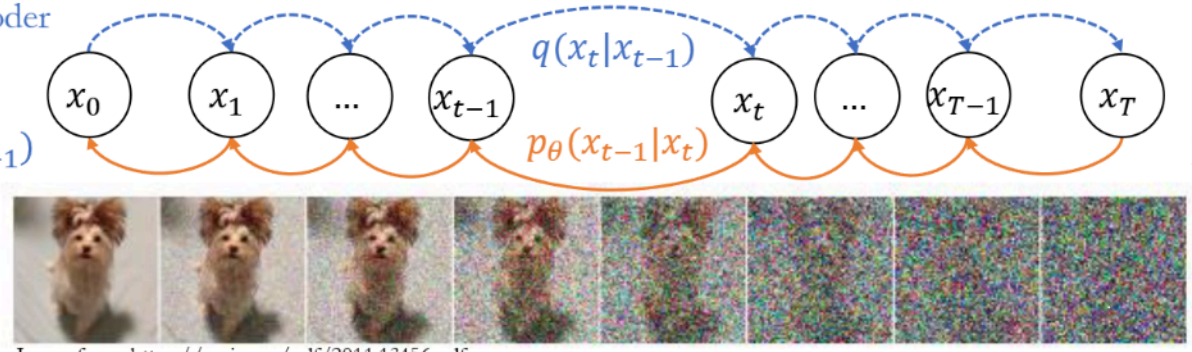


Image from: https://arxiv.org/pdf/2011.13456.pdf

Hierarchical Decoder

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

Model: The Forward Process Is Defined by a **fixed** Markov Transition Distribution $q(x_t|x_{t-1})$

- The forward process starts at the data distribution, i.e., $q(x_0) = p_{data}(x)$
- Define forward process via Markov transition:

$$q(x_t|x_{t-1}) \triangleq \mathcal{N}(x_t; \mu = w_\mu(t)x_{t-1}, \Sigma = w_\sigma(t)I)$$

  - where $w_\mu(t)$ and $w_\sigma(t)$ can be functions that vary across time t.
  - For simplicity, we will use $w_\mu(t) = 1$ and $w_\sigma(t) = 1$ so that above simplifies:

$$q(x_t|x_{t-1}) \triangleq \mathcal{N}(x_t; \mu = x_{t-1}, \Sigma = I)$$

  - Notice there are **no trainable parameters**.

Model: The Forward Process Can Be **Collapsed** Into a Single Step, I.E., $q(x_t|x_0)$ Is Known in **Closed-Form**

Distribution-based derivation

- The joint distribution is Gaussian because each of the components are conditionally Gaussian:
  - $q(x_{1:t}|x_0)$
  - $= \prod_{t'=1}^{t} q(x_{t'}|x_{t'-1})$
  - $= q(x_1|x_0)q(x_2|x_1)q(x_3|x_2)...$
  - $= \mathcal{N}(x_1|x_0)\mathcal{N}(x_2|x_1)\mathcal{N}(x_3|x_2)...$
- The marginal of a Gaussian is also Gaussian, i.e.,

$$q(x_t|x_0) = \mathcal{N}(x_t; \mu = x_0, \Sigma = t \cdot I)$$

Random variable derivation

- By the definition of $q(x_t|x_{t-1})$:

  $x_t = x_{t-1} + \epsilon_{t-1}$ where $\epsilon_{t-1} \sim \mathcal{N}(0, I)$
  - $x_t = x_{t-1} + \epsilon_{t-1}$
  - $= x_{t-2} + \epsilon_{t-2} + \epsilon_{t-1}$
  - $= x_{t-3} + \epsilon_{t-3} + \epsilon_{t-2} + \epsilon_{t-1}$
  - $= \cdots = x_0 + \sum_{t'=0}^{t-1} \epsilon_{t'}$
- Fact: Adding Gaussian RVs is another Gaussian RV distributed so that:
  - $x_t = x_0 + \sum_{t'=0}^{t-1} \epsilon_{t'}$
  - Where $\tilde{\epsilon}_t \sim \mathcal{N}(0, t \cdot I)$
  - Thus, $x_t \sim \mathcal{N}(x_0, t \cdot I)$

Model: The Forward Process Can Be **Collapsed** Into a Single Step, I.E., $q(x_t|x_0)$ Is Known in **Closed-Form**

- What does this mean intuitively? $q(x_t|x_0) = \mathcal{N}(x_t; \mu = x_0, \Sigma = T \cdot I) \iff x_t \sim \mathcal{N}(x_0, T \cdot I)$

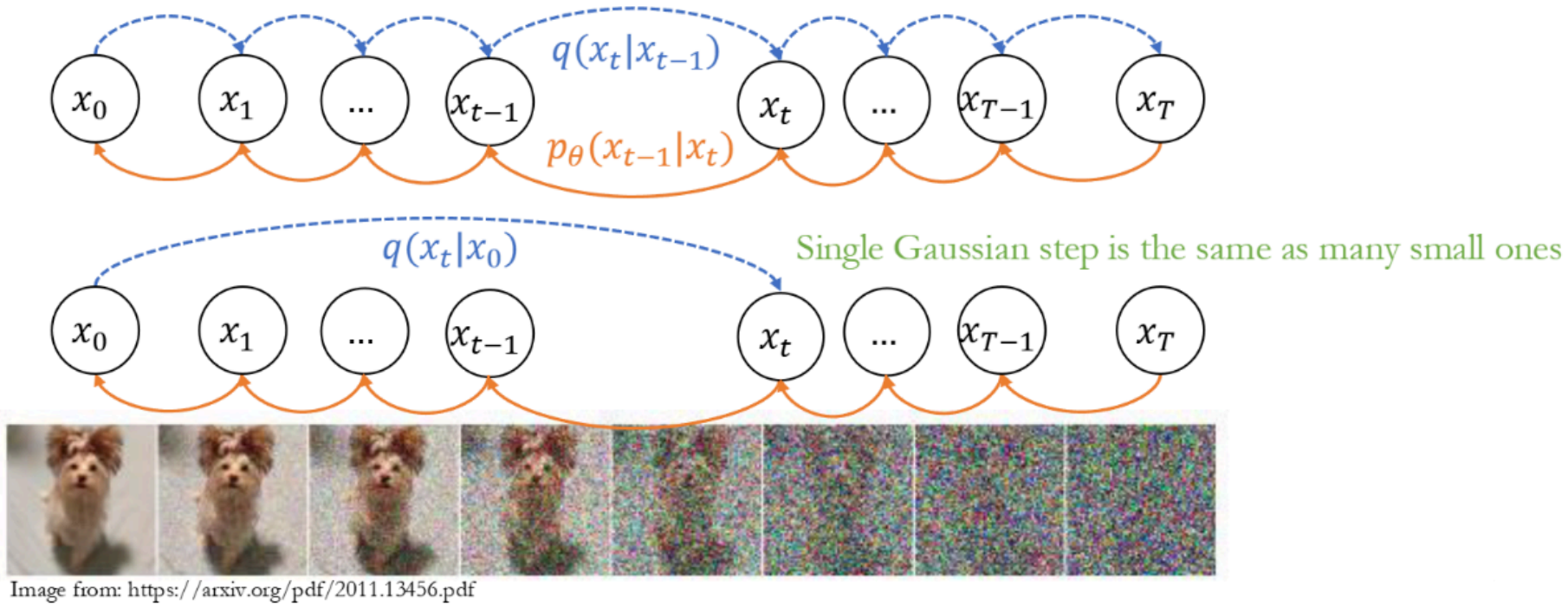


Image from: https://arxiv.org/pdf/2011.13456.pdf

Model: The Reverse Transition **Conditioned on** x_0 Is Known in Closed Form ($q(x_{t-1}|x_t, x_0)$)

- The ideal reverse transition $p^*(x_{t-1}|x_t)$ would be the posterior of q:

$$p^*(x_{t-1}|x_t) = q(x_{t-1}|x_t) = \frac{q(x_t|x_{t-1})q(x_{t-1})}{q(x_t)}$$

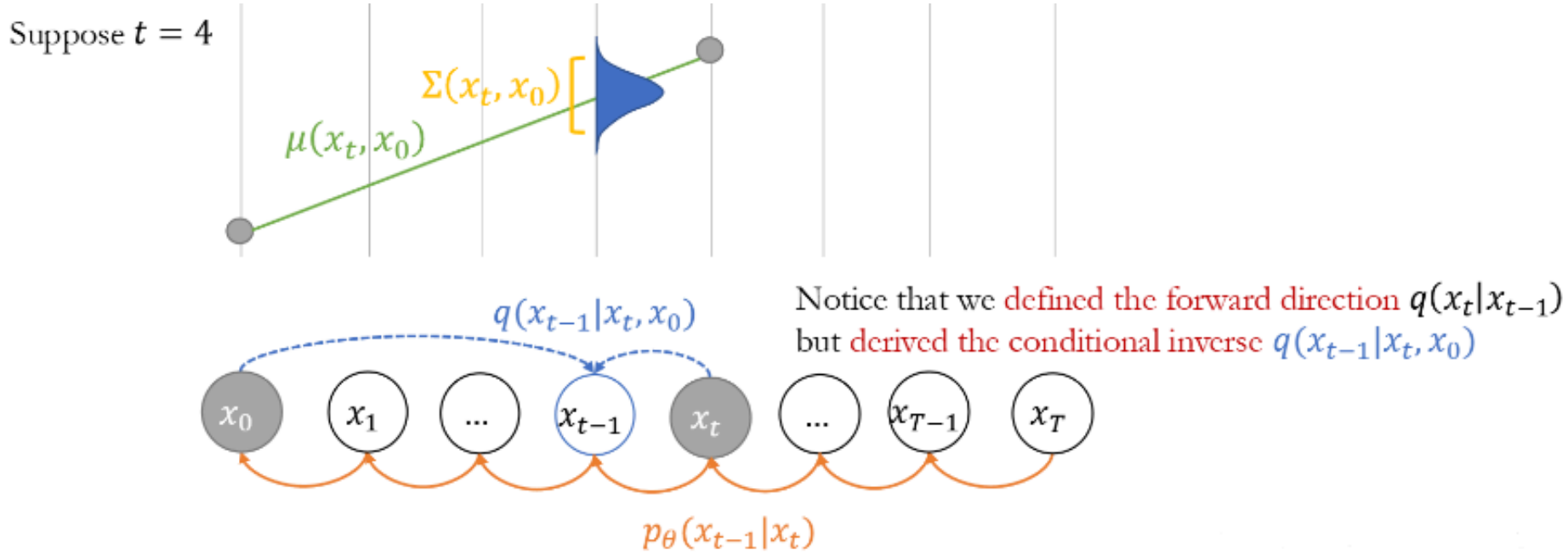
  - However, this is intractable 😕

- However, if conditioned on x_0, the posterior is tractable
  - $q(x_{t-1}|x_t, x_0)$
  - $= \frac{q(x_t|x_{t-1},x_0)q(x_{t-1}|x_0)}{q(x_t|x_0)}$
  - $= \frac{q(x_t|x_{t-1})q(x_{t-1}|x_0)}{q(x_t|x_0)}$ (Markov property of q, i.e., x_t only dependent on x_{t-1})
  - $= \frac{\mathcal{N}(x_t;\mu=x_{t-1},\Sigma=I)\mathcal{N}(x_{t-1};\mu=x_0,\Sigma=(t-1)\cdot I)}{\mathcal{N}(x_t;\mu=x_0,\Sigma=t\cdot I)}$
  - $= \mathcal{N}\left(x_{t-1}; \mu = \left(1-\frac{1}{t}\right)x_t + \frac{1}{t}x_0, \Sigma = \left(1-\frac{1}{t}\right)I\right)$

  Derivation uses the fact each can be expressed as the exponential of a quadratic function, i.e., a Guassian. These quadratic functions can be combined to form a single quadratic in terms of x_{t-1} and then used to derive the mean and variance in terms of t, x_t, and x_0.

Model: The Reverse Transition **Conditioned on** x_0 Is Known in Closed Form ($q(x_{t-1}|x_t, x_0)$)

- What does this mean intuitively? $q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \mu = \left(1 - \frac{1}{t}\right) x_t + \frac{1}{t} x_0, \Sigma = \left(1 - \frac{1}{t}\right) I\right)$
- Notice that we defined the forward direction $q(x_t|x_{t-1})$ but derived the conditional inverse $q(x_{t-1}|x_t, x_0)$.



Model: The Reverse Process Approximates the Posterior Transition of q

- **Prior distribution $p(x_T)$**
  - Theory: As $T \to \infty$, $q(x_T) \to \mathcal{N}(x_T; \mu = \mu_{data}, \Sigma = \Sigma_{data} + T \cdot I)$.
  - Therefore, we choose a Gaussian prior distribution:
    (note that this is with out simplified $w_\mu(t)$ and $w_\sigma(t)$ and is only approximate if T is finite)

$$p(x_T) \triangleq \mathcal{N}(x_T; \mu = \mu_{data}, \Sigma = \Sigma_{data} + T \cdot I) \approx q(x_T)$$

- **Reverse transition distribution $p_\theta(x_{t-1}|x_t)$**
  - Theory: As $T \to \infty$, then $q(x_{t-1}|x_t)$ is known to be Gaussian.
  - Therefore, we choose the approximate posterior to be Gaussian:
    (note with finite timesteps the posterior is not Gaussian)

$$p_\theta(x_{t-1}|x_t) \triangleq \mathcal{N}(x_{t-1}; \mu = \mu_\theta(x_t, t), I) \approx q(x_{t-1}|x_t)$$

Training(1): Reweighted ELBO Simplifies to Predicting Noise From Noisy Input at Each Time t

- Idea is to simply optimize the negative ELBO of this VAE: $\min_\theta \mathbb{E}_{q(x_0)}[-ELBO(x_0; p_\theta, q)]$
- The objective can be simplified to reconstruction error across time: $\min_\theta \mathbb{E}_{t\in\{1,\dots,T\},x_0,\tilde{\epsilon}_t}[\frac{1}{2t^2}||x_0 - \mu_\theta(x_t, t)||^2]$
  - The $\frac{1}{2t^2}$ term is from the ELBO derivation, where μ_θ is like the decoder and tries to predict the clean x_0
- The model usually predicts the **noise** instead of the clean image.
  - First we rewrite the decoder as the noisy input minus predicted noise: $\mu_\theta(x_t, t) = x_t - \epsilon_\theta(x_t, t)$
  - Then, we can rewrite the objective: $||x_0 - \mu_\theta(x_t, t)||^2 = ||x_0 - (x_t - \epsilon_\theta(x_t, t))||^2 = ||x_0 - ((x_0 + \epsilon_t) - \epsilon_\theta(x_0 + \epsilon_t, t))||^2 = ||\tilde{\epsilon}_t - \epsilon_\theta(x_0 + \tilde{\epsilon}_t, t)||^2$
- Thus, this objective can be simplified to (full derivation in last slides):

$$\min_\theta \mathbb{E}_{t\in\{1,\dots,T\},x_0,\tilde{\epsilon}_t}[||\tilde{\epsilon}_t - \epsilon_\theta(x_0 + \tilde{\epsilon}_t, t)||_2^2]$$

  - (Where a scaling of $\frac{1}{2t^2}$ from the ELBO is dropped for each term)

Training(2): Multiple VAEs With Fixed Encoder and Shared Parameters

- Reconsidering the following as VAE objectives: $\min_\theta \mathbb{E}_{t\in\{1,\dots,T\},x_0,\tilde{\epsilon}_t}[||\tilde{\epsilon}_t - \epsilon_\theta(x_0+\tilde{\epsilon}_t, t)||_2^2]$
  - Let $z \equiv x_t$ and $x \equiv x_0$, now let's define a VAE for each t:
  - Encoders: $q_t(z|x) = \mathcal{N}(x, tI)$
  - Decoders: $p_{\theta_t}(x|z) = \mathcal{N}(z - t\epsilon_{\theta_t}(z), tI)$
- For any t, the VAE objective is equivalent to minimizing:

$$\min_{\theta_t} \mathbb{E}_{x,\epsilon}[\frac{1}{t^2}||x - (\underbrace{(x+t\epsilon)}_{z} - t\epsilon_{\theta_t}(x+t\epsilon))||_2^2] = \min_{\theta_t} \mathbb{E}_{x,\epsilon}[||\epsilon - \epsilon_{\theta_t}(x+t\epsilon)||_2^2]$$

- These could all be run in parallel

$$\frac{1}{n}\Sigma_t \min_{\theta_t} \mathbb{E}_{x,\epsilon}[||\epsilon - \epsilon_{\theta_t}(x+t\epsilon)||_2^2] = \min_{\theta_1\dots\theta_T} \mathbb{E}_{t\in\{1,\dots,T\},x,\epsilon}[||\epsilon - \epsilon_{\theta_t}(x+t\epsilon)||_2^2]$$

- If parameters θ are shared, i.e., $\epsilon_{\theta_t}(z) \equiv \epsilon_\theta(z,t)$, the objectives are equivalent!

Training(3): Multiple Denoising AEs

$$\min_{\theta} \mathbb{E}_{t\in\{1,\dots,T\},x_0,\tilde{\epsilon}_t}[||\tilde{\epsilon}_t - \epsilon_\theta(x_0 + \tilde{\epsilon}_t, t)||_2^2]$$

- Identity encoders: $f_t(x) = x$
- Decoders: $g_t(z) = z - t\epsilon_{\theta_t}(z)$
- Noise added to input: $n_t(x) = x + t\epsilon$
- For any t, the denoising AE objective with MSE would be:
  - $\min_{\theta_t} \mathbb{E}_{x,\epsilon}[||x - g_t(f_t(x + t\epsilon))||_2^2]$
  - $\equiv \min_{\theta_t} \mathbb{E}_{x,\epsilon}[||x - (x + t\epsilon - t\epsilon_\theta(x + t\epsilon)||_2^2]$
  - $\equiv \min_{\theta_t} \mathbb{E}_{x,\epsilon}[t^2||\epsilon - \epsilon_\theta(x + t\epsilon)||_2^2]$
- Again, the global objective is equivalent if parameters θ are shared.
  - Parameters θ are shared, i.e., $\epsilon_{\theta_t} = \epsilon_\theta(z, t)$
  - All objectives combined where the t-th objective has a weight of $\frac{1}{t^2}$

Sampling(1): DDPM Sampling Simply Samples the Generative Model Sequentially

- Remember: $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}|\mu = x_t - \frac{1}{t}\epsilon_\theta(x_t, t), I)$
- Sample from prior distribution $x_T \sim p(x_T)$
- For $t = T, \dots, 1$ do:
  - $z \sim \mathcal{N}(0, I)$
  - $x_{t-1} = x_t - \frac{1}{t}\epsilon_\theta(x_t, t) + z$
- For the last step, we may also quantize using rounding to get integer value for pixels.

Sampling(2): DDIM Redefines $p_\theta(x_{t-1}|x_t)$ in Terms of $q_\sigma(x_{t-1}|x_t, x_0)$ Where x_0 Is Approximated

- Note that we can approximate x_0 using $\epsilon_\theta(x_t, t)$:

$$x_0 \approx \hat{x}_0 \triangleq f_\theta(x_t, t) = x_t - \sqrt{t}\epsilon_\theta(x_t, t)$$

- The generative model p_θ can now be defined **using** q_σ:

$$p_\theta(x_{t-1}|x_t) \triangleq \begin{cases} \mathcal{N}(f(x_1, 1), \sigma_1^2 I), & \text{if } t = 1 \\ q_\sigma(x_{t-1}|x_t, f_\theta(x_t, t)), & \text{otherwise} \end{cases}$$

- A special case of DDIM allows for **deterministic sampling.**
  - Stochastic training but determinisitic sampling (i.e., non-stochastic)
- DDIM also allows different timesteps in sampling compared to training—thus enabling faster sampling with the **same model** $\epsilon_\theta(x_t, t)$.
- We can use a pretrained version of ϵ_θ and just **sample differently**

Resources

- **Reference work from original creators**
  - Lai, C. H., Song, Y., Kim, D., Mitsufuji, Y., & Ermon, S. (2025). The Principles of Diffusion Models. arXiv preprint arXiv:2510.21890.
- **Diffusion model tutorial from our own Prof. Stanley Chan**
  - Chan, S. (2024). Tutorial on diffusion models for imaging and vision. Foundations and Trends® in Computer Graphics and Vision, 16(4), 322-471. https://arxiv.org/abs/2510.21890

Resources

- **Excellent diffusion models blog post**
  - https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- **Excellent score-based generative models blog post**
  - https://yang-song.net/blog/2021/score/
- **Score-based comprehensive literature**
  - https://scorebasedgenerativemodeling.github.io/

A Few Important Diffusion Model Works

- **Diffusion Models:** Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics.” ICML 2015.
  - Introduced the learning of diffusion models as forward noising and reverse denoising process.
- **Denoising Diffusion Probabilistic Models (DDPM):** Jonathan Ho et al. “Denoising diffusion probabilistic models.” NeurIPS 2020.
  - Made several key design decisions and connected to Noise-Conditioned Score Networks (NSCN).
- **DDPM++:** Alexander Nichol & Dhariwal. “Improved Denoising Diffusion Probabilistic Models.” ICML 2021.
  - Makes several engineering improvements over DDPM including faster sampling and better likelihood.
- **Denoising Diffusion Implicit Model (DDIM):** Jiaming Song et al. “Denoising diffusion implicit models.” ICLR 2021.
  - Proposed a non-Markovian sampling procedure that includes a deterministic variant.

Related Score-Based Modeling Key Papers

- **Noise-Conditioned Score Networks (NCSN):** Yang Song et al. "Generative Modeling by Estimating Gradients of the Data Distribution." NeurIPS 2019.
  - Trains many score functions (i.e., $\nabla_x \log p_t(x)$) at multiple noise levels t and uses Langevin sampling for generation.
- **Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations." ICLR 2021.**
  - Unifies diffusion and score-based methods under common framework.
  - Generalizes DDPM and NCSM to continuous time
  - Can convert stochastic diffusiong model to continuous normalizing flow
- **Tero Karras et al. "Elucidating the Design Space of Diffusion-Based Generative Models." NeurIPS 2022.**
  - Unifies the key practical/engineering design decisions for diffusion models.

Extra derivations (Time permitting)

Lemma: Markov property for $q(x_{t-1}|x_{\geq t}, x_0)$

- $q(x_{t-1}|x_{\geq t}, x_0)$
- $= \frac{q(x_{\geq t}|x_{t-1},x_0)q(x_{t-1}|x_0)}{q(x_{\geq t}|x_0)}$
- $= \frac{q(x_{\geq t}|x_{t-1})q(x_{t-1}|x_0)}{q(x_{\geq t}|x_0)}$
- $= \frac{q(x_{t-1}|x_0)\Pi_{t'=t+1}^{T} q(x_{t'}|xt'-1)}{q(x_t|x_0)\Pi_{t'=t+1}^{T} q(x_{t'}|x_{t'-1})}$
- $= \frac{q(x_{t-1}|x_0)q(x_{t'}|x_{t'-1})\Pi_{t'=t+1}^{T} q(x_{t'}|x_{t'-1})}{q(x_t|x_0)\Pi_{t'=t+1}^{T} q(x_{t'}|x_{t'-1})}$
- $= \frac{q(x_{t-1}|x_0)q(x_t|x_{t-1},x_0)}{q(x_t|x_0)}$
- $= q(x_{t-1}|x_t, x_0)$

Alternate simplification of KL term from ELBO

- $KL(q(x_{1:T}|x_0), p_\theta(x_{1:T})) = \mathbb{E}_{q(x_{1:T}|x_0)}[\log(\frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T})})]$
- $=\mathbb{E}_{q(x_{1:T}|x_0)}[\log(\frac{q(x_1|x_0)\Pi_{t=2}^T q(x_t|xt{-}1,x_0)}{p(x_T)\Pi_{t=2}^T p_\theta(x_{t-1}|x_t)})]$
- $=\mathbb{E}_{q(x_{1:T}|x_0)}[\Sigma_{t=2}^T \log(\frac{q(x_t|x_{t-1},x_0)}{p_\theta(x_{t-1}|x_t)}) + \log(\frac{q(x_1|x_0)}{p(x_T)})]$
- $=\mathbb{E}_{q(x_{1:T}|x_0)}[\Sigma_{t=2}^T \log(\frac{q(x_t|x_{t-1},x_0)}{p_\theta(x_{t-1}|x_t)}) * \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)} + \log(\frac{q(x_1|x_0)}{p(x_T)})]$
- $=\mathbb{E}_{q(x_{1:T}|x_0)}[\Sigma_{t=2}^T \log(\frac{q(x_t|x_{t-1},x_0)}{p_\theta(x_{t-1}|x_t)}) + \Sigma_{t=2}^T \log(\frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}) + \log(\frac{q(x_1|x_0)}{p(x_T)})]$
- $=\mathbb{E}_{q(x_{1:T}|x_0)}[\Sigma_{t=2}^T \log(\frac{q(x_t|x_{t-1},x_0)}{p_\theta(x_{t-1}|x_t)}) + \log(\frac{q(x_T|x_0)}{q(x_1|x_0)}) + \log(\frac{q(x_1|x_0)}{p(x_T)})]$
- $=\mathbb{E}_{q(x_{1:T}|x_0)}[\Sigma_{t=2}^T \log(\frac{q(x_t|x_{t-1},x_0)}{p_\theta(x_{t-1}|x_t)}) + \log(\frac{q(x_T|x_0)}{p(x_T)})]$
- $=\Sigma_{t=2}^T \mathbb{E}_{q(x_t|x_0)}[KL(q(x_{t-1}|x_t,x_0), p_\theta(x_{t-1}|x_t))] + KL(q(x_T|x_0), p(x_T))$