Diffusion Models

David I. Inouye

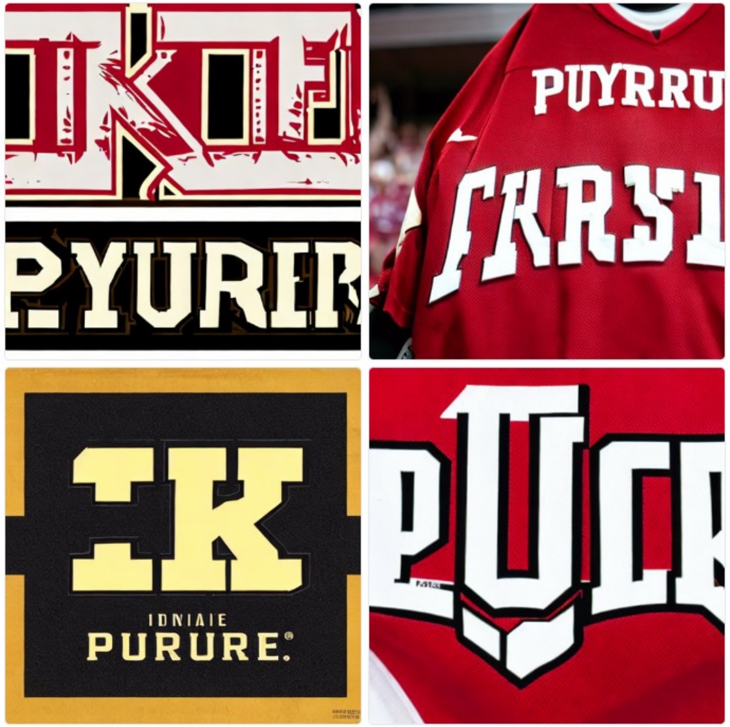
PURDUE UNIVERSITY

Elmore Family School of Electrical
and Computer Engineering

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

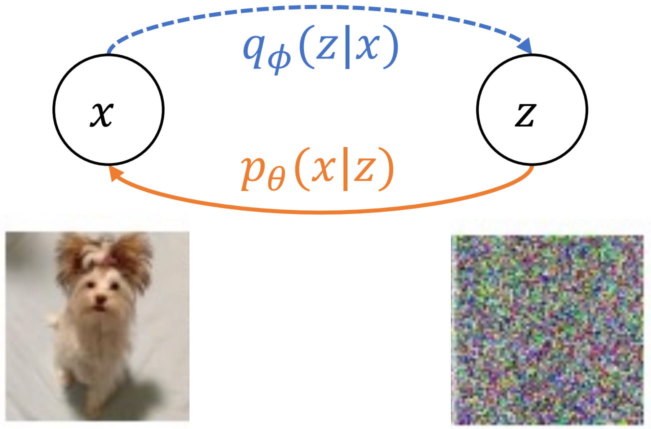
Overview

- 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

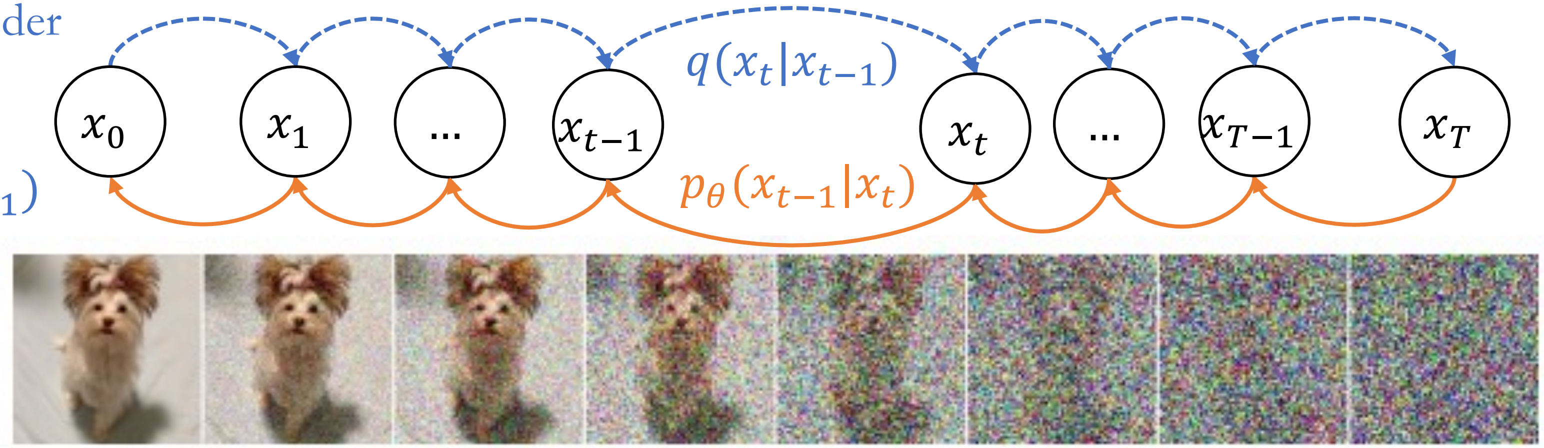
- Diffusion models can be viewed as hierarchical VAEs
  - Forward process = hierarchical **encoder**
  - Reverse process = hierarchical **decoder**
- Several critical differences from VAE
  - Involves **multiple latent representations** rather than one
  - Hierarchical encoder is **fixed** (i.e., no trainable parameters)
  - **Parameters θ are shared** between decoder steps

Simple VAE (Same-Dimension)



Hierarchical Encoder

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



Hierarchical Decoder

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

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

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
$$q(x_t|x_{t-1}) \overset{\text{def}}{=} \mathcal{N}\left(x_t; \mu = w_\mu(t)x_{t-1}, \Sigma = w_\sigma(t)I\right)$$
  - 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
$$q(x_t|x_{t-1}) \overset{\text{def}}{=} \mathcal{N}(x_t; \mu = x_{t-1}, \Sigma = I)$$
  - 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
  - $q(x_{1:t}|x_0)$
  - $= \prod_{t'=1}^{t} q(x_{t'}|x_{t'-1})$
  - $= q(x_1|x_0)q(x_2|x_1)q(x_3|x_2) \dots$
  - $= \mathcal{N}(x_1|x_0, I)\mathcal{N}(x_2|x_1, I)\mathcal{N}(x_3|x_2, I) \dots$

- 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 $q(x_t|x_{t-1})$

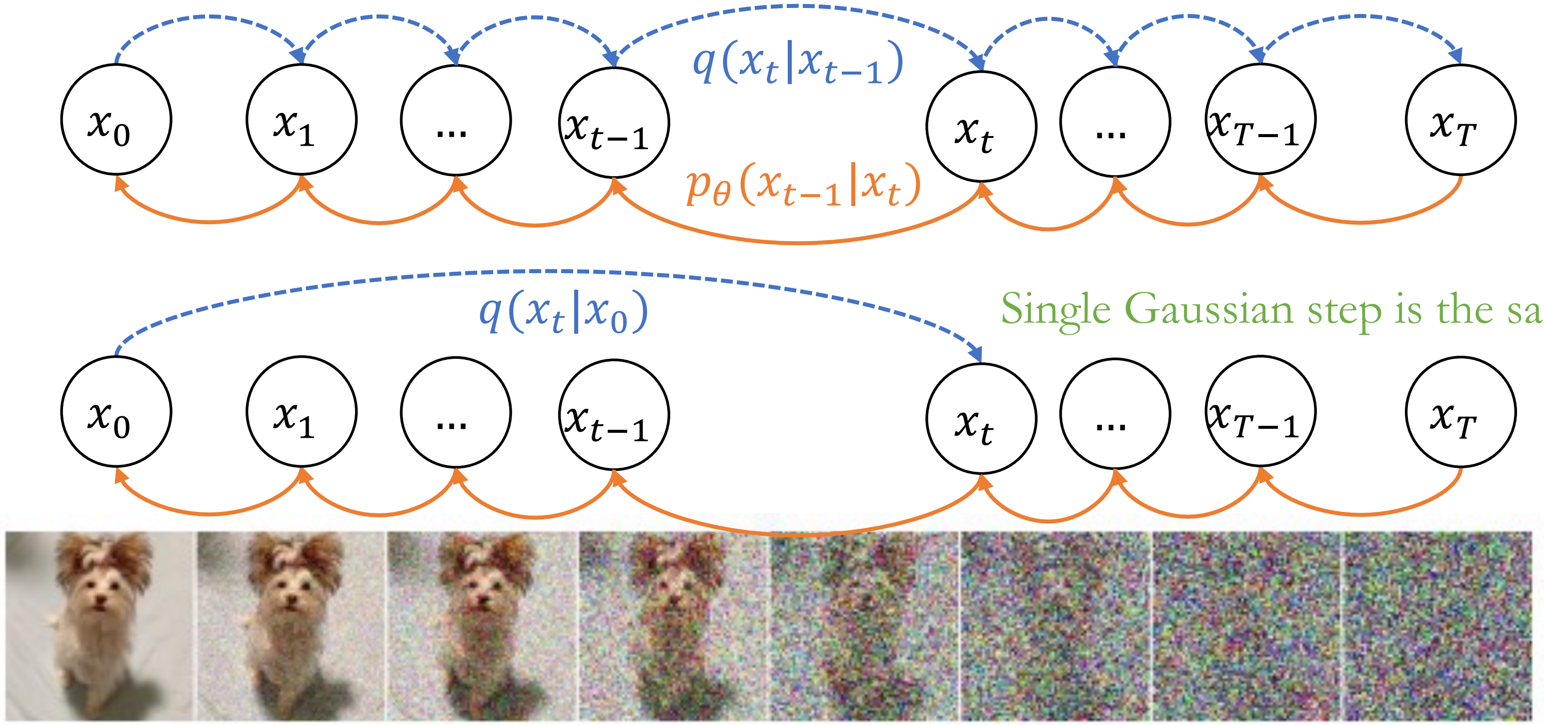
$x_t = x_{t-1} + \epsilon_{t-1}$ where $\epsilon_{t-1} \sim \mathcal{N}(0, I)$

  - $x_t = x_{t-1} + \epsilon_{t-1}$
  - $= x_{t-2} + \epsilon_{t-2} + \epsilon_{t-1}$
  - $= x_{t-3} + \epsilon_{t-3} + \epsilon_{t-2} + \epsilon_{t-1}$
  - $= \cdots = x_0 + \sum_{t'=0}^{t-1} \epsilon_{t'}$
- Fact: Adding Gaussian RVs is another Gaussian RV distributed so that
  - $x_t = x_0 + \sum_{t'=0}^{t-1} \epsilon_{t'} = x_0 + \tilde{\epsilon}_t$
  - 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?

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



Single Gaussian step is the same as many small ones

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})
  - $= \frac{\mathcal{N}(x_t; \mu=x_{t-1}, \Sigma=I)\mathcal{N}(x_{t-1}; \mu=x_0, \Sigma=(t-1)\cdot I)}{\mathcal{N}(x_t; \mu=x_0, \Sigma=t\cdot I)}$
  - $= \mathcal{N}\left(x_{t-1}; \mu = \left(1 - \frac{1}{t}\right) x_t + \frac{1}{t} x_0, \Sigma = \left(1 - \frac{1}{t}\right) I\right)$

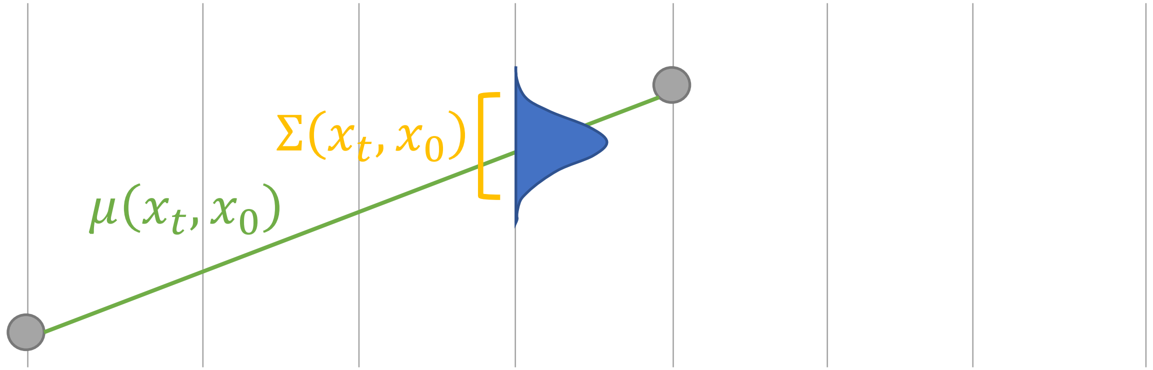
Derivation uses the fact each can be expressed as the exponential of a quadratic function, i.e., a Gaussian. These quadratic functions can be combined to form a single quadratic in terms of x_{t-1} and then used to derive the mean and variance in terms of t, x_t and x_0.

Model: The reverse transition **conditioned on** x_0 is known in closed form ($q(x_{t-1}|x_t, x_0)$)

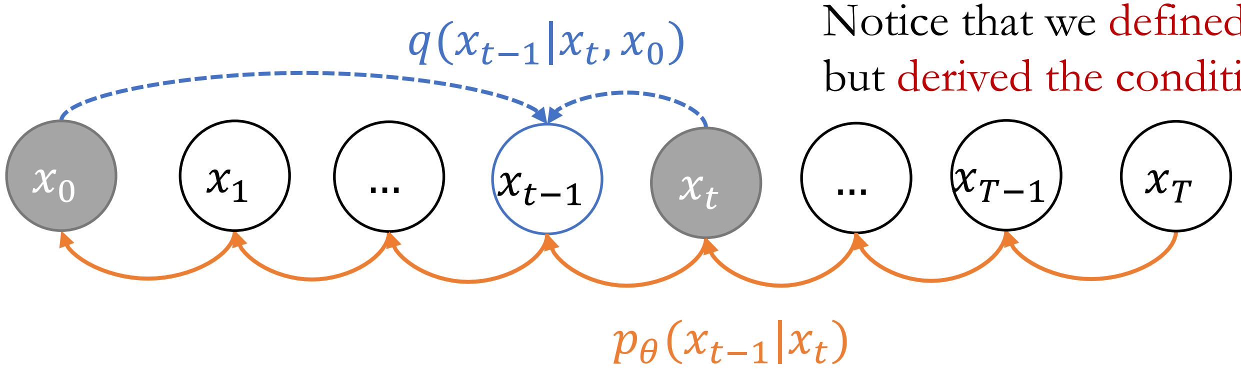
- What does this mean intuitively?

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \mu = \left(1 - \frac{1}{t}\right) x_t + \frac{1}{t} x_0, \Sigma = \left(1 - \frac{1}{t}\right) I\right)$$

Suppose $t = 4$



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 our simplified $w_\mu(t)$ and $w_\sigma(t)$ and is only approximate if T is finite)
    $$p(x_T) \overset{\text{def}}{=} \mathcal{N}(x_T; \mu = \mu_{data}, \Sigma = \Sigma_{data} + T \cdot I) \ \left(\approx q(x_T)\right)$$
- Reverse transition distribution $p_\theta(x_{t-1}|x_t)$
  - Theory: As the number of timesteps approaches infinity, i.e., $T \to \infty$, 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)
    $$p_\theta(x_{t-1}|x_t) \overset{\text{def}}{=} \mathcal{N}(x_{t-1}; \mu = \mu_\theta(x_t), I) \ \left(\approx q(x_{t-1}|x_t)\right)$$

Training(1): Reweighted ELBO simplifies to predicting noise from noisy input at each time t

- The main idea is to simply optimize the negative ELBO of this VAE

$$\min_{\theta} \mathbb{E}_{q(x_0)}[-\mathrm{ELBO}(x_0; p_\theta, q)]$$

- In practice, this objective can be simplified to (derivation in last slides)

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

  - Where a scaling of $\frac{1}{2t^2}$ is dropped for each term

Training(2): Multiple VAEs with fixed encoder and shared parameters

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

- Encoders based on t: $q_t(z|x) = \mathcal{N}(x, tI)$
- Decoders based on t: $p_{\theta_t}(x|z) = \mathcal{N}\left(z - t\epsilon_{\theta_t}(z), tI\right)$

  (prior $p(z)$ is irrelevant for training)
- For any t, the VAE objective would be:
  - $\min_{\theta_t} \mathbb{E}_{x,\epsilon}\left[\frac{1}{t^2}\left\|x - \left((x + t\epsilon) - t\epsilon_{\theta_t}(x + t\epsilon\ )\right)\right\|_2^2\right] \equiv \min_{\theta_t} \mathbb{E}_{x,\epsilon}\left[\left\|\epsilon - \epsilon_{\theta_t}(x)\right\|_2^2\right]$
- These could all be run in parallel
  - $\frac{1}{n}\sum_t \min_{\theta_t} \mathbb{E}_{x,\epsilon}\left[\left\|\epsilon - \epsilon_{\theta_t}(x + t\epsilon)\right\|_2^2\right] = \min_{\theta_t} \mathbb{E}_{t\in\{1,\dots,T\},x,\epsilon}\left[\left\|\epsilon - \epsilon_{\theta_t}(x + t\epsilon)\right\|_2^2\right]$
- 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

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

- 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:
  - $\min_{\theta_t} \mathbb{E}_{x,\epsilon}\left[\left\|x - g_t\big(f_t(x+\epsilon)\big)\right\|_2^2\right]$
  - $\equiv \min_{\theta_t} \mathbb{E}_{x,\epsilon}\,[\|x - (x + t\epsilon - t\epsilon_\theta(x + t\epsilon)\|_2^2]$
  - $\equiv \min_{\theta_t} \mathbb{E}_{x,\epsilon}\,[t^2\|\epsilon - \epsilon_\theta(x + t\epsilon)\|_2^2]$
- Again, global objective equivalent if
  - Parameters θ are shared, i.e., $\epsilon_{\theta_t}(z) \equiv \epsilon_\theta(z, t)$
  - 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: $p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}\middle|\mu = x_t - \frac{1}{t}\epsilon_\theta(x_t, t), I\right)$
- 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 $\epsilon_\theta(x_t, t)$
  - $x_0 \approx \hat{x}_0 \overset{\text{def}}{=} f_\theta(x_t, t) = x_t - t\epsilon_\theta(x_t, t)$
- The generative model p_θ can now be defined **using** q_σ
  - $p_\theta(x_{t-1}|x_t) \overset{\text{def}}{=} \begin{cases} \mathcal{N}(f(x_1, 1), \sigma_1^2 I), & \text{if } t = 1 \\ q_\sigma(x_{t-1}|x_t, f_\theta(x_t, t)), & \text{otherwise} \end{cases}$
- A special case of DDIM allows for **deterministic sampling**
  - Stochastic training but deterministic 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

- 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/ (in particular, notice section Connection to diffusion models and others)
- 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.
  - Sohl-Dickstein et al. [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.
  - Ho et al. [2020] made several key design decisions and connected to Noise-Conditioned Score Networks (NSCN) [Yang & Ermon, 2019]
- *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.
  - Song et al. [2020] proposed a non-Markovian sampling procedure that includes a deterministic variant (note: the training is the same as DDPM)

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 NCSN to continuous time
  - Can convert stochastic diffusion 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

Key Derivations
(time permitting)

Training(1): Minimizing joint negative ELBO across all timesteps

- Remember the negative evidence lower bound (ELBO) from VAEs

$$-\mathrm{ELBO}(x;\, p_g, q_f) = \mathbb{E}_{q_f}\left[-\log \frac{p_g(x,z)}{q_f(z|x\,)}\right] = \underbrace{\mathbb{E}_{q_f}[-\log p_g(x|z)]}_{\text{Computable, see reconstruction error slides}} + \underbrace{\mathrm{KL}\left(q_f(z|x), p_g(z)\right)}_{\text{Computable in closed-form for Gaussian distributions}}$$

- Now let $x \equiv x_0$ and $z \equiv x_{1:T}$ in the above equation
- $-\mathrm{ELBO}(x_0; p_\theta, q) = \mathbb{E}_{q(x_{0:T})}\left[-\log \frac{p_\theta(x_0, x_{1:T})}{q(x_{1:T}|x_0\;)}\right]$
- $= \mathbb{E}_q[-\log p_\theta(x_0|x_{1:T})] + KL\left(q(x_{1:T}|x_0), p_\theta(x_{1:T})\right)$
- $= \underbrace{\mathbb{E}_{q(x_1|x_0)}}_{\text{(Markov property)}}[-\log p_\theta(x_0|x_1)] + KL\left(q(x_{1:T}|x_0), p_\theta(x_{1:T})\right)$

Lemma: Chain rule of KL

- Chain rule of KL
  - $KL\big(q(x), p(x)\big) = \sum_{i=1}^{d} \mathbb{E}_{q(x_{<i})}\big[KL\big(q(x_i|x_{<i}), p(x_i|x_{<i})\big)\big]$
- Inverted chain rule of KL (equivalent)
  - $KL\big(q(x), p(x)\big) = \sum_{i=1}^{d} \mathbb{E}_{q(x_{>i})}\big[KL\big(q(x_i|x_{>i}), p(x_i|x_{>i})\big)\big]$
- Derivation for two dimensions
- $KL\big(q(x_1, x_2), p(x_1, x_2)\big) = \mathbb{E}_{q(x_1,x_2)}\left[\log \frac{q(x_1,x_2)}{p(x_1,x_2)}\right]$
- $= \mathbb{E}_{q(x_1)}\left[\mathbb{E}_{q(x_2|x_1)}\left[\log \frac{q(x_1)q(x_2|x_1)}{p(x_1)p(x_2|x_1)}\right]\right]$
- $= \mathbb{E}_{q(x_1)}\left[\log \frac{q(x_1)}{p(x_1)} + \mathbb{E}_{q(x_2|x_1)}\left[\log \frac{q(x_2|x_1)}{p(x_2|x_1)}\right]\right]$
- $= KL\big(q(x_1), p(x_1)\big) + \mathbb{E}_{q(x_1)}\big[KL\big(q(x_2|x_1), p(x_2|x_1)\big)\big]$

Diffusion ELBO: Simplification using KL chain rule and Markov property

For notational simplicity, let x_{T+1} be a dummy random variable that is independent of all other random variables (the distribution does not matter).

- $KL\left(q(x_{1:T}|x_0), p_\theta(x_{1:T})\right)$
- $= \sum_{t=1}^{T} \mathbb{E}_{q(x_{>t}|x_0)}\left[KL\left(q(x_t|x_{>t}, x_0), p_\theta(x_t|x_{>t})\right)\right]$ (KL chain rule)
- $= \sum_{t=2}^{T+1} \mathbb{E}_{q(x_{\geq t}|x_0)}\left[KL\left(q(x_{t-1}|x_{\geq t}, x_0), p_\theta(x_{t-1}|x_{\geq t})\right)\right]$
- $= \sum_{t=2}^{T+1} \mathbb{E}_{q(x_{\geq t}|x_0)}\left[KL\left(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)\right)\right]$ (Markov properties)
- $= \sum_{t=2}^{T+1} \mathbb{E}_{q(x_t|x_0)}\left[KL\left(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)\right)\right]$
- $= \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[KL\left(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)\right)\right] + KL\left(q(x_T|x_0), p(x_T)\right)$

Proof of Markov property for q and an alternative derivation that is usually used are provided at the end.

Diffusion ELBO: A reconstruction term and many KL terms

- $-\mathrm{ELBO}(x_0; p_\theta, q)$
- $= \mathbb{E}_q[-\log p_\theta(x_0|x_{1:T})] + KL\big(q(x_{1:T}|x_0), p_\theta(x_{1:T})\big)$

  ($x \equiv x_0$ and $z \equiv x_{1:T}$)
- $= \mathbb{E}_{q(x_1|x_0)}[-\log p_\theta(x_0|x_1)] + KL\big(q(x_{1:T}|x_0), p_\theta(x_{1:T})\big)$

  (Markov property)
- $= \mathbb{E}_{q(x_1|x_0)}[-\log p_\theta(x_0|x_1)]$ (L_0 Initial reconstruction term, e.g., dequantization)
  - $+ \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\big[KL\big(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)\big)\big]$ (L_1 to L_{T-1} KL terms)
  - $+KL\big(q(x_T|x_0), p(x_T)\big)$ (L_T “prior” term, constant w.r.t. θ)

The KL terms simplify to MSE between true posterior mean and predicted mean

- KL between two Gaussians
  - $KL\left(\mathcal{N}_1(\mu_0, \sigma_0^2 I), \mathcal{N}_2(\mu_1, \sigma_1^2 I)\right) = \frac{1}{2\sigma_1^2}\|\mu_1 - \mu_0\|_2^2 + \frac{1}{2}\left(\frac{\sigma_0^2}{\sigma_1^2} - d + \log\frac{\sigma_1^2}{\sigma_0^2}\right)$
- $KL\big(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)\big)$
- $= KL\left(\mathcal{N}\left(\mu_q = \left(1 - \frac{1}{t}\right)x_t + \frac{1}{t}x_0, \Sigma = \left(1 - \frac{1}{t}\right)I\right), \mathcal{N}(\mu_\theta(x_t, t), I)\right)$
- $= \frac{1}{2}\left\|\mu_q - \mu_\theta(x_t, t)\right\|_2^2 + C$

The KL term can equivalently be written as predicting the noise

- We can *equivalently* rewrite μ_q in terms of x_t and the noise $\tilde{\epsilon}_t \sim \mathcal{N}(x_0, tI)$
  - $\mu_q = \left(1 - \frac{1}{t}\right) x_t + \frac{1}{t} x_0 = \left(1 - \frac{1}{t}\right) x_t + \frac{1}{t}(x_t - \tilde{\epsilon}_t) = x_t - \frac{1}{t}\tilde{\epsilon}_t$
- We can also re-parameterize $\mu_\theta(x_t, t)$
  - $\mu_\theta(x_t, t) = x_t - \frac{1}{t}\epsilon_\theta(x_t, t)$
- Now this simplifies to predicting Gaussian noise
  - $KL\big(q(x_{t-1}|x_t, x_0), p_\theta(x_{t-1}|x_t)\big) = \frac{1}{2\sigma_1^2}\left\|\mu_q - \mu_\theta(x_t, t)\right\|_2^2 + C$
  - $= \frac{1}{2}\left\|x_t - \frac{1}{t}\tilde{\epsilon}_t - \left(x_t - \frac{1}{t}\epsilon_\theta(x_t, t)\right)\right\|_2^2 + C$
  - $= \frac{1}{2}\left\|-\frac{1}{t}\left(\tilde{\epsilon}_t - \epsilon_\theta(x_t, t)\right)\right\|_2^2 + C$
  - $= \frac{1}{2t^2}\|\tilde{\epsilon}_t - \epsilon_\theta(x_t, t)\|_2^2 + C \qquad \left(\equiv \frac{1}{2}\left\|\mu_q - \mu_\theta(x_t, t)\right\|_2^2 + C\right)$

Training(1): Reweighted ELBO simplifies to predicting noise from noisy input at each time t

- $\min_{\theta} \mathbb{E}_{q(x_0)}[-\mathrm{ELBO}(x_0; p_\theta, q)]$
- $\equiv \min_{\theta} \mathbb{E}_{q(x_0,x_1)}[-\log p_\theta(x_0|x_1)]$ (L_0 in practice is dequantization term)
  - $+\sum_{t=2}^{T} \mathbb{E}_{q(x_0,x_t)}\left[KL\left(q(x_{t-1}|x_t,x_0), p_\theta(x_{t-1}|x_t)\right)\right]$ (L_1 to L_{T-1} KL terms)
  - $+\mathbb{E}_{q(x_0)}\left[KL\left(q(x_T|x_0), p(x_T)\right)\right]$ (L_T “prior” term, constant w.r.t. θ)
- $\equiv \min_{\theta} \mathbb{E}_{q(x_1|x_0)}[-\log p_\theta(x_0|x_1)] + \sum_{t=2}^{T} \mathbb{E}_{t,x_0,\tilde{\epsilon}_t}\left[\frac{1}{2t^2}\left\|\tilde{\epsilon}_t - \epsilon_\theta\left(x_0 + \tilde{\epsilon}_t, t\right)\right\|_2^2\right]$
- In practice, this objective is simplified to

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

  - By combining an approximation of L_0 with L_1 etc.
  - And dropping scaling of $\frac{1}{2t^2}$

Sampling(2): DDIM redefines the forward process in terms of $q(x_{t-1}|x_t, x_0)$ instead of $q(x_t|x_{t-1})$

- DDIM notices that the training objective only depends on $q(x_t|x_0)$ rather than the joint $q(x_{1:T}|x_0)$
  - Thus, there exist many joint distributions $q(x_{1:T}|x_0)$ that have the same marginals $q(x_t|x_0)$ as DDPM
- Instead of defining $q(x_t|x_{t-1})$, DDIM defines
  - $q_\sigma(x_{1:T}|x_0) \stackrel{\text{def}}{=} q_\sigma(x_T|x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0)$
  - $q_\sigma(x_T|x_0) \stackrel{\text{def}}{=} \mathcal{N}(x_0, T \cdot I)$
  - $q_\sigma(x_{t-1}|x_t, x_0) \stackrel{\text{def}}{=} \mathcal{N}(x_{t-1}; \mu = h(x_t, x_0, \sigma_t), \Sigma = \sigma_t I)$ (Not sure the form for our simple example.)
- DDIM derives that $q_\sigma(x_t|x_0) \equiv q(x_t|x_0)$, i.e., it matches the marginals of DDPM, **for any** $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_T]^T$
- Thus, the same training objective can be used!

Extra derivations

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

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

Alternative simpliciation of KL term from ELBO

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

- $\sum_{t=2}^{T} \log \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}$
- $= -\log q(x_1|x_0) + \log q(x_2|x_0) - \log q(x_2|x_0)$
  $+ \log q(x_3|x_0) \cdots + \log q(x_T\,|x_0)$
- $= -\log q(x_1|x_0) + \log q(x_T|x_0)$