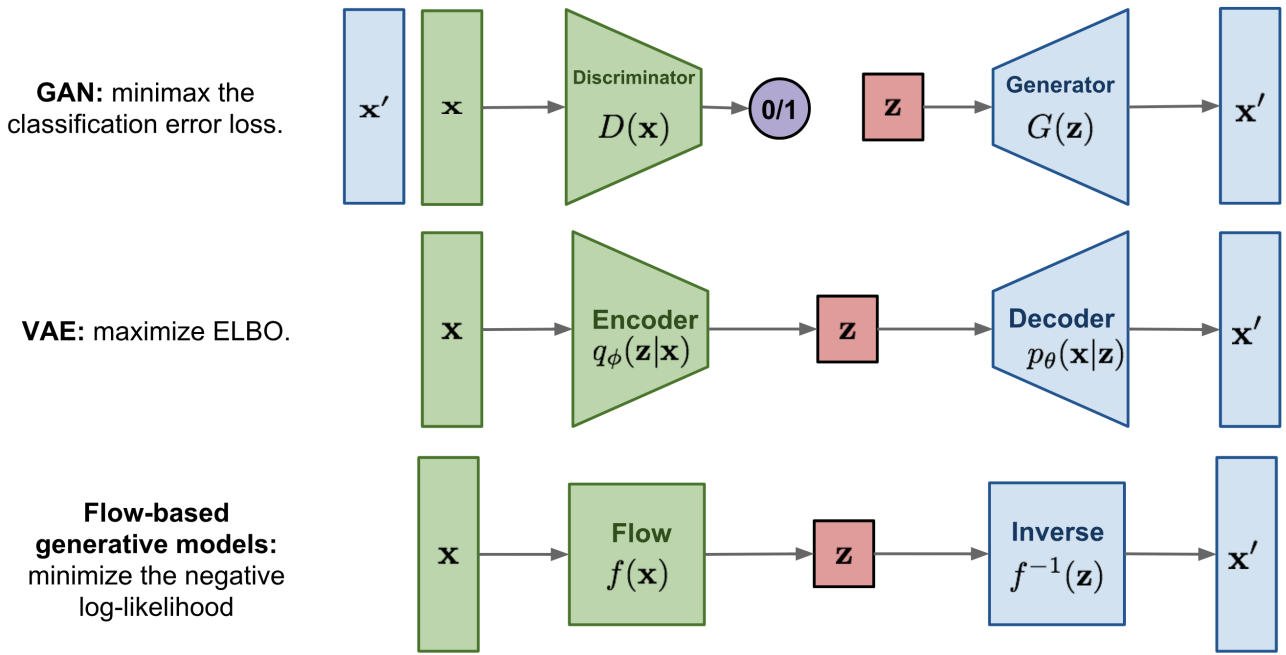
Diffusion Probabilistic Models

2023/03/31

Generative Models

$\mathcal{X} \leftarrow f(\mathcal{Z})$



Overview of different types of generative models. [1]

Generative Models

Explicit/Implicit Density Estimation

Explicit

e.g. VAE, Flow-based Models
Estimates the true pdf or cdf over the sample space, **Stable**

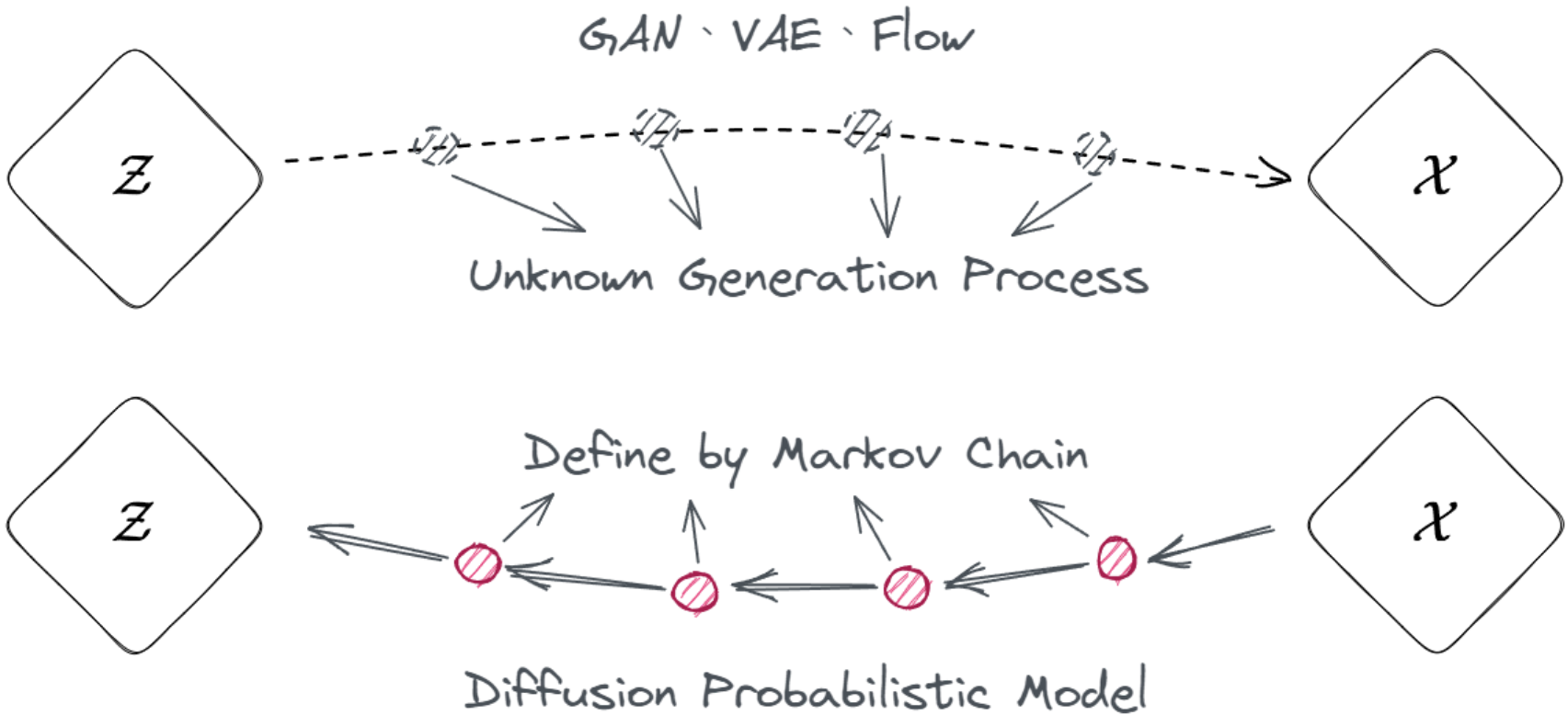
However, there will be problems such as low generation quality, limited architecture, high computing cost, etc.

Implicit

Like GAN, it uses confrontation training to approximate the real distribution and has excellent generation quality.

But it is difficult to train stably and prone to mode collapse.

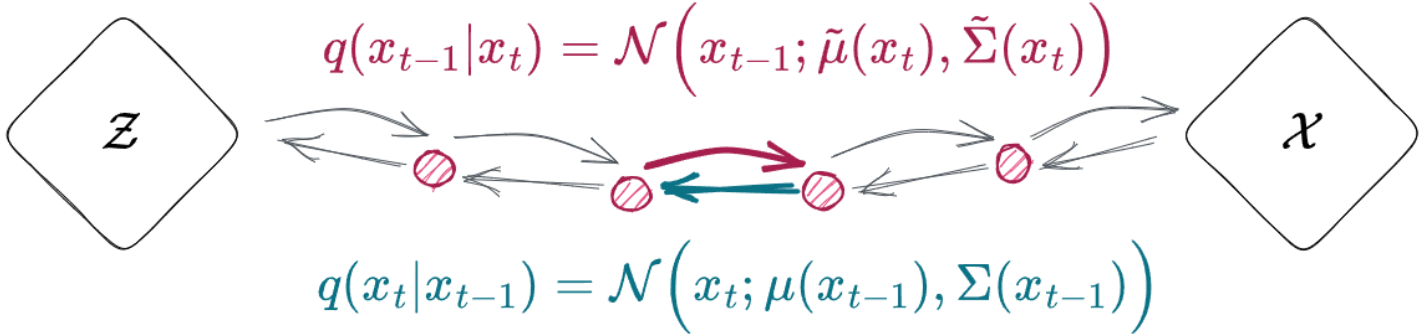
Generation Process



Why Use Markov Chains

Connect two distributions

1. Each state is only related to the previous state.
2. When the diffusion process (forward process) $q(x_t|x_{t-1})$ is **Normal Distribution or Binomial Distribution**, as long as the **variation is small** enough, the reverse process $q(x_{t-1}|x_t)$ will also be **the same type of distribution**.



$$q(x_0) = \int q(x_{0..T})\, dx_{1..T}$$

Variational Lower Bound

Derivation of Objectives

$$L_{\text{CE}} = -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) = -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \right)$$

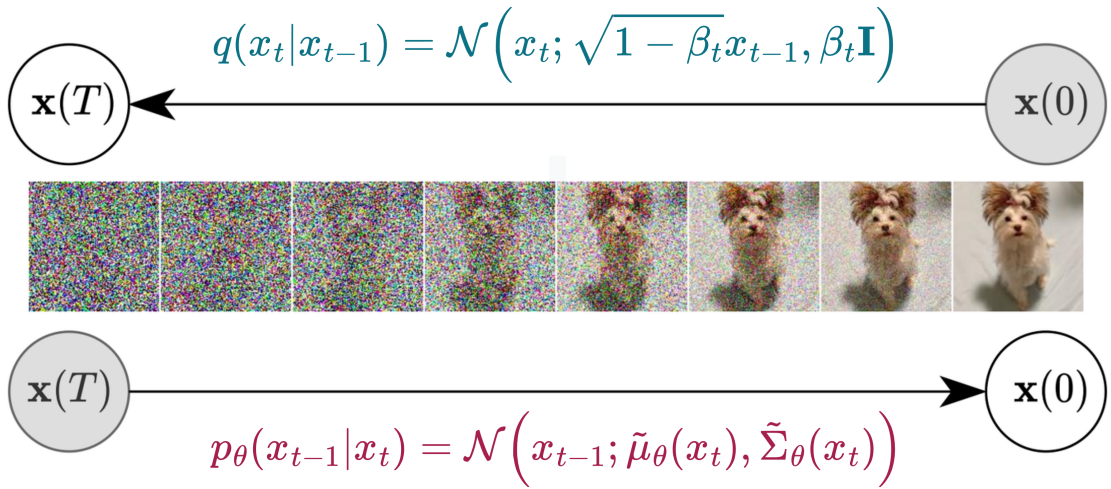
$$\leq \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] = L_{\text{VLB}}$$

$$= \mathbb{E}_q [\underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}]$$

In fact, the goal of learning is not $q(x_{t-1}|x_t)$, but $q(x_{t-1}|x_t, x_0)$ of L_t

By Bayes' rules, it can be rewritten to consist only of forward process

$$q(x_{t-1}|x_t, x_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$



$$
\begin{aligned}
x_t &\sim \mathcal{N}(\sqrt{\alpha_t}x_{t-1}, 1-\alpha_t\mathbf{I}) && ;\text{let } \alpha_i = 1-\beta_i \\
&= \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} && ;\text{where } \epsilon_i \sim \mathcal{N}(0, I) \\
&= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{\alpha_t}\sqrt{1-\alpha_{t-1}}\epsilon_{t-2} + \sqrt{1-\alpha_t}\epsilon_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2} && ;\text{where } \bar{\epsilon}_{t-2} \text{ merges two Gaussians} \\
&= \dots \\
&= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}\mathbf{I}) && ;\bar{\alpha}\_t = \prod_{i=1}^{t}\alpha_i
\end{aligned}
$$

Backward Process

Using Bayes' rule

$$q(x_t|x_{t-1}) = \mathcal{N}\Big(x_t; \sqrt{\alpha_t}x_{t-1}, \beta_t\mathbf{I}\Big) \propto \exp\Big(-\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{2\beta_t}\Big)$$

$$q(x_t|x_0) = \mathcal{N}\Big(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}\Big) \propto \exp\Big(-\frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{2(1-\bar{\alpha}_t)}\Big)$$

$$\begin{aligned}
q(x_{t-1}|x_t, x_0) &= q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\frac{\big(x_{t-1} - \tilde{\mu}(x_t, x_0)\big)^2}{\tilde{\Sigma}(x_t, x_0)}\Big)
\end{aligned}$$

Backward Process

Estimate target

$$\tilde{\Sigma}(x_t, x_0) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

$$\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0$$

$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right) \qquad ; x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon)$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t)$$

The Goal is to Estimate x_0 or ϵ from x_t.

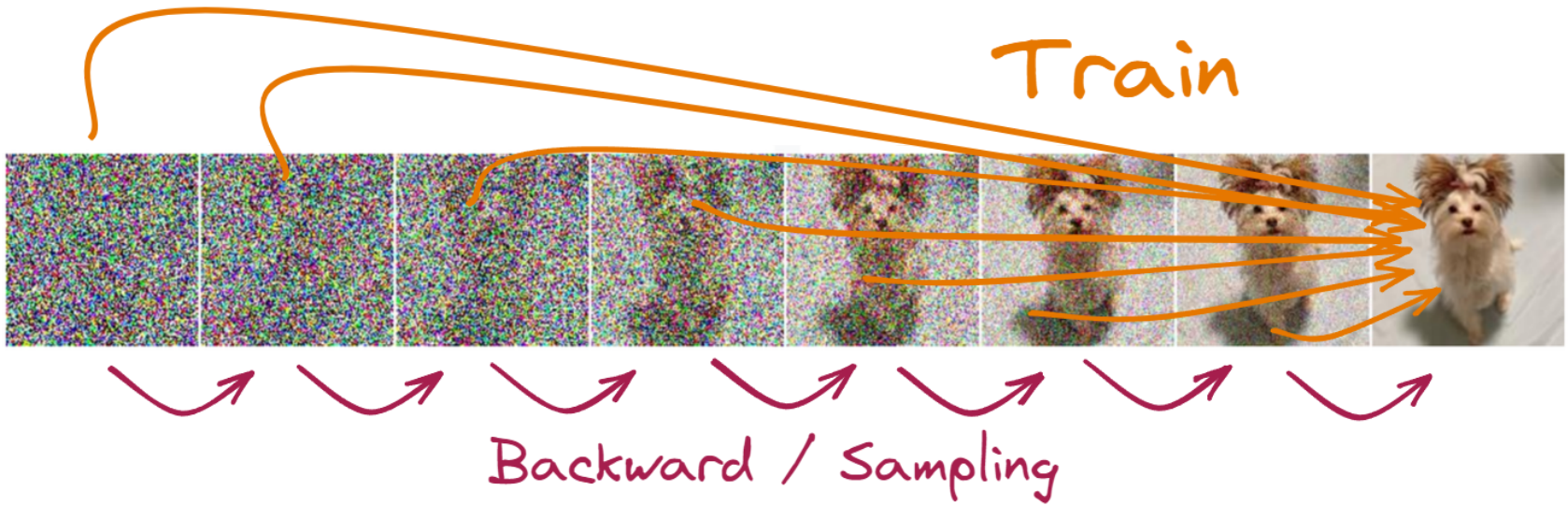
DPM Algorithm

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$



Conditional Generation

Classifier Guided Generation

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t, y) &= \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log q(y|\mathbf{x}_t) \\ &\approx -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log f_\phi(y|\mathbf{x}_t) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} (\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) - \sqrt{1-\bar{\alpha}_t} \nabla_{\mathbf{x}_t} \log f_\phi(y|\mathbf{x}_t))\end{aligned}$$

- Bootstrap generation with additional classifiers.
- Do not use parallel data when training DPM.
- General pre-trained classifiers cannot be used directly.

Conditional Generation

Classifier Guided Generation

- Bootstrap generation with additional classifiers.
- Do not use parallel data when training DPM.
- General pre-trained classifiers cannot be used directly.

Conditional	Guidance	Scale	FID	sFID	IS	Precision	Recall
✗	✗		26.21	**6.35**	39.70	0.61	0.63
✗	✓	1.0	33.03	6.99	32.92	0.56	**0.65**
✗	✓	10.0	**12.00**	10.40	**95.41**	**0.76**	0.44
✓	✗		10.94	6.02	100.98	0.69	**0.63**
✓	✓	1.0	**4.59**	**5.25**	186.70	0.82	0.52
✓	✓	10.0	9.11	10.93	**283.92**	**0.88**	0.32

Table 4: Effect of classifier guidance on sample quality. Both conditional and unconditional models were trained for 2M iterations on ImageNet 256×256 with batch size 256.

Conditional Generation

Classifier Free Guidances

$$
\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \Big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \Big) \\
\bar{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t, y) &= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \sqrt{1-\bar{\alpha}_t}\, w \nabla_{\mathbf{x}_t} \log p(y|\mathbf{x}_t) \\
&= \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) + w\big(\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \big) \\
&= (w+1)\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, y) - w\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)
\end{aligned}
$$

- High quality.
- Parallel data required.

Conditional Generation

As Plug-and-Play Priors

Algorithm 1 Inferring a point estimate of $p(\mathbf{x}|\mathbf{y}) \approx \delta(\mathbf{x} - \boldsymbol{\eta})$, under a DDPM prior and constraint.

input pretrained DDPM $\boldsymbol{\epsilon}_\theta$, auxiliary data $\mathbf{y}$, constraint c, time schedule $(t_i)_{i=1}^T$, learning rate λ

1: Initialize $\mathbf{x} \sim \mathcal{N}(\mathbf{0}; \mathbf{I})$.
2: **for** $i = T..1$ **do**
3: Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}; \mathbf{I})$
4: $\mathbf{x}_{t_i} = \sqrt{\bar{\alpha}_{t_i}}\mathbf{x} + \sqrt{1 - \bar{\alpha}_{t_i}}\boldsymbol{\epsilon}$
5: $\mathbf{x} \leftarrow \mathbf{x} - \lambda \nabla_{\mathbf{x}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_{t_i}, t_i)\|_2^2 - \log c(\mathbf{x}, \mathbf{y})]$
6: **end for**

output $\boldsymbol{\eta} = \mathbf{x}$

- Execute conditional generation with an optimized concept.
- Models trained on data without Gaussian noise can also be used to guide generation.
- The gradient needs to be calculated for the entire generation model.

Compare

Advantage

- High flexibility
- Good quality results

  Diffusion Models Beat GANs on Image Synthesis
- Stable training

Defect

- The sampling time is much longer than VAE, GAN, etc.
  - ODE sampling

    e.g. DDIM, DPM-Solver++, UniPC
  - Knowledge Distillation

    e.g. Progressive Distillation, Consistency Models

Pretrained Model

- Stable Diffusion (Image)
- AudioLDM (Audio)
- GENIE (Text)

More Control

ControlNet, DDIB, DPM-Encoder, SDEdit, EdiTTS, etc.