1

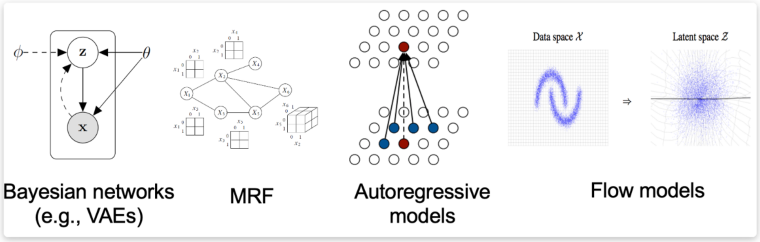
DDM

quote from Yang Song

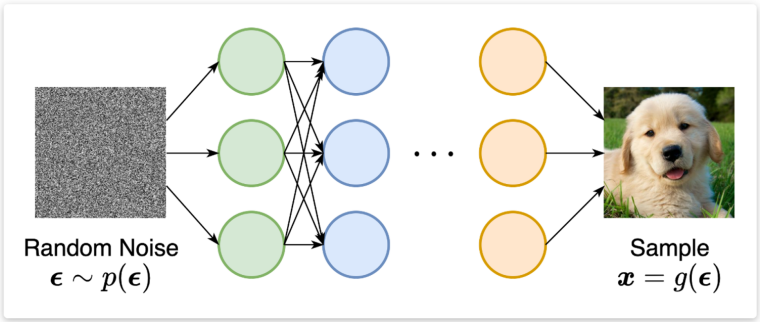
Existing generative modeling techniques can largely be grouped into two categories based on how they represent probability distributions.

1. **likelihood-based models**, which directly learn the distribution's probability density (or mass) function via (approximate) maximum likelihood. Typical likelihood-based models include autoregressive models [1, 2, 3], normalizing flow models [4, 5], energy-based models (EBMs) [6, 7], and variational auto-encoders (VAEs) [8, 9].

2. **implicit generative models** [10], where the probability distribution is implicitly represented by a model of its sampling process. The most prominent example is generative adversarial networks (GANs) [11], where new samples from the data distribution are synthesized by transforming a random Gaussian vector with a neural network.



Bayesian networks, Markov random fields (MRF), autoregressive models, and normalizing flow models are all examples of likelihood-based models. All these models represent the probability density or mass function of a distribution.



GAN is an example of implicit models. It implicitly represents a distribution over all objects that can be produced by the generator network.

Likelihood-based models and implicit generative models, however, both have significant limitations. Likelihood-based models either require strong restrictions on the model architecture to ensure a tractable normalizing constant for likelihood computation, or must rely on surrogate objectives to approximate maximum likelihood training. Implicit generative models, on the other hand, often require adversarial training, which is notoriously unstable [12] and can lead to mode collapse [13].

BG

- generated by latent variable

- generally learn low-dim latent representations

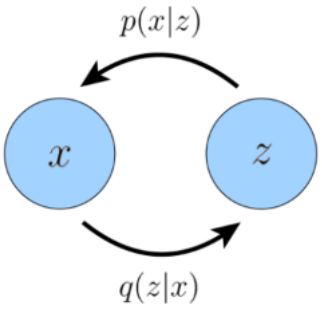
ELBO

1. : model the latent variable and the data, likelihood-based: maximize the likelihood .
2. $$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} \qquad p(\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})}$$
3. evidence lower bound: VLB ELB)

$$\begin{aligned}
\log p(\boldsymbol{x}) &= \log p(\boldsymbol{x}) \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} && \text{(Multiply by } 1 = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}) && (9)\\
&= \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) (\log p(\boldsymbol{x})) d\boldsymbol{z} && \text{(Bring evidence into integral)} && (10)\\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} [\log p(\boldsymbol{x})] && \text{(Definition of Expectation)} && (11)\\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})} \right] && \text{(Apply Equation 2)} && (12)\\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right] && \text{(Multiply by } 1 = \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}) && (13)\\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z}|\boldsymbol{x})} \right] && \text{(Split the Expectation)} && (14)\\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right] + D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}|\boldsymbol{x})) && \text{(Definition of KL Divergence)} && (15)\\
&\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \right] && \text{(KL Divergence always} \geq 0) && (16)
\end{aligned}$$

(a) KL 0, : the approximate posterior.p(x):likelihood of observed or generated data KL 0 logp(x) ELBO KL q KL p(x) optimize ELBO KL 0 ELBO p(x)

VAE



1. maximize ELBO

$$\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right] \quad \text{(Chain Rule of Probability)} \quad (17)$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(z)}{q_\phi(z|x)}\right] \quad \text{(Split the Expectation)} \quad (18)$$

$$= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z))}_{\text{prior matching term}} \quad \text{(Definition of KL Divergence)} \quad (19)$$

(a)

i. encoder learn , decoder learn

-

– reconstruction likelihood of decoder: decoder latent

– z variational distribution) q(z|x)) encoder Dirac

2. encoder: multi gaussion, prior standard gaussion:

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x)\mathbf{I}) \quad (20)$$

$$p(z) = \mathcal{N}(z; \mathbf{0}, \mathbf{I}) \quad (21)$$

(a)

(b) reconstruction term monte varlo latents are sampled from . sample -> reparamterization trick

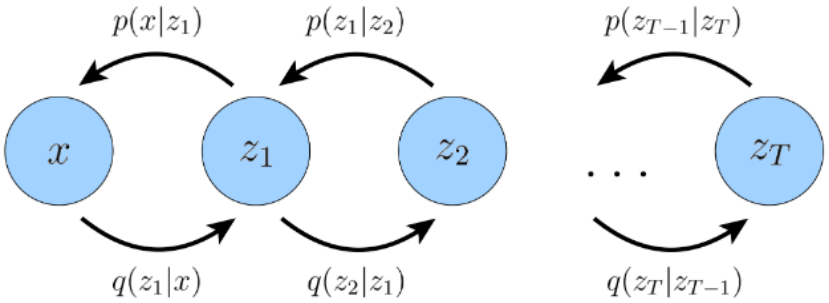
(c) reparamterization trick: r.v. noise variable standard gaussioan sample, gradient descent.(element-wise product)

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon \quad \text{with } \epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \mathbf{I})$$

(d) VAE z dim x-> compact latent vector, latent vector

HVAE

decoding each latent only conditions on previous latent ,



$$p(x, z_{1:T}) = p(z_T)p_\theta(x|z_1)\prod_{t=2}^{T} p_\theta(z_{t-1}|z_t) \quad (23)$$

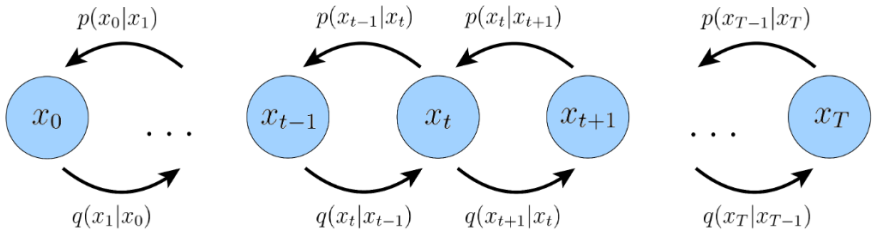
$$q_\phi(z_{1:T}|x) = q_\phi(z_1|x)\prod_{t=2}^{T} q_\phi(z_t|z_{t-1}) \quad (24)$$

Lower-bound:

$$\mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p(x, z_{1:T})}{q_\phi(z_{1:T}|x)}\right] = \mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p(z_T)p_\theta(x|z_1)\prod_{t=2}^{T} p_\theta(z_{t-1}|z_t)}{q_\phi(z_1|x)\prod_{t=2}^{T} q_\phi(z_t|z_{t-1})}\right] \quad (29)$$

–

Variational Diffusion Models



1. HVAE
   (a) latent dim = data dim latent data
   (b) at t, the latent encoder is defined as a linear Gaussian model
   (c) The Gaussian parameter of latent encoder vary -> at final T, standard Gaussian
2. the distribution of each latent variable in the encoder is a Gaussian centered around its previous hierarchical latent.
   (a) $$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$
   (b) linear Gaussian parameters: hyperparameters or learned
   (c) $\boldsymbol{\mu}_t(\boldsymbol{x}_t) = \sqrt{\alpha_t}\boldsymbol{x}_{t-1},\ \boldsymbol{\Sigma}_t(\boldsymbol{x}_t) = (1-\alpha_t)\mathbf{I},$
   (d) $$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$
   $$p(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \tag{32}$$
   where,
   (e) $$p(\boldsymbol{x}_T) = \mathcal{N}(\boldsymbol{x}_T; \mathbf{0}, \mathbf{I}) \tag{33}$$
   (f) HVAE, encoder , Gaussian. .
3. ELBO
   (a) $$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{q(\boldsymbol{x}_{T-1}|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1}) \parallel p(\boldsymbol{x}_T))\right]}_{\text{prior matching term}} - \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_{t-1},\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \parallel p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}))\right]}_{\text{consistency term}} \tag{45}$$
      i. reconstruction:
      ii. prior matching : learnable parameters, T gaussian, 0
      iii. consistency term: , Monte Carlo r.v. , variance
      $$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}$$
      iv. : given markov
      1. denosing matching. denosing transition step ground-

truth denosing transition step HVAE encoder learning VDM gaussian

1. $$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}$$ -markov,

$$\begin{aligned}
\boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}^*_{t-1} \\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\right. \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t - \alpha} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\sqrt{\alpha_t - \alpha}} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{\alpha_t - \alpha} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_t} \\
&= \dots \\
&= \sqrt{\prod_{i=1}^{t}\alpha_i}\boldsymbol{x}_0 + \sqrt{1-\prod_{i=1}^{t}\alpha} \\
&= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0 \\
&\sim \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I})
\end{aligned}$$

2. $$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$$

: $$\mathcal{N}(\boldsymbol{x}_{t-1}; \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t}}_{\mu_q(\boldsymbol{x}_t, \boldsymbol{x}_0)}, \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}_{\boldsymbol{\Sigma}_q(t)}\mathbf{I})$$

3.

2. denosing transition step ground-truth denosing transition step Gaussian : $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$, mean variance

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\bar{\alpha}_t} \qquad \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\boldsymbol{x}}{}$$

1. neural network

$$\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \\
&= \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q(t)) \parallel \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q(t))) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left[\log\frac{|\boldsymbol{\Sigma}_q(t)|}{|\boldsymbol{\Sigma}_q(t)|} - d + \mathrm{tr}(\boldsymbol{\Sigma}_q(t)^{-1}\boldsymbol{\Sigma}_q(t)) + (\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)^T\boldsymbol{\Sigma}_q(t)^{-1}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left[\log 1 - d + d + (\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)^T\boldsymbol{\Sigma}_q(t)^{-1}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left[(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)^T\boldsymbol{\Sigma}_q(t)^{-1}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2}\left[(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)^T\left(\sigma_q^2(t)\mathbf{I}\right)^{-1}(\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q)\right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)}\left[\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \boldsymbol{\mu}_q\|_2^2\right]
\end{aligned}$$

2.

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left[\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2 \right]$$

3.

4. minimize expectations over all timesteps $\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim U\{2,T\}} \left[\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}($

4. Learning noise parameters

(a) $$\mathrm{SNR}(t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$$ SNR SNR Noise t snr

(b) $$= \frac{1}{2} \left(\frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t} \right) \left[\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2 \right] \quad \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left[\|\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \boldsymbol{x}_0\|_2^2 \right] = \frac{1}{2} (\mathrm{SNR}(t-1)$$

(c) $$\mathrm{SNR}(t) = \exp(-\omega_{\boldsymbol{\eta}}(t))$$ nn model snr,

5. noise

(a) $$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}} \quad = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\boldsymbol{\epsilon}_0$$,

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\|\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2 \right]$$

(b) KL

(c) predict x0 predict noise.(predict noise performance)

6.

(a) Tweedie's Formula : "states that the true mean of an exponential family distribution, given samples drawn from it, can be estimated by the maximum likelihood estimate of the samples (aka empirical mean) plus some correction term involving the score of the estimate."

for a Gaussian variable $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, Tweedie's Formula states that:

$$\mathbb{E}[\boldsymbol{\mu}_z|\boldsymbol{z}] = \boldsymbol{z} + \boldsymbol{\Sigma}_z \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$$

sample

$\boldsymbol{x}_t$ is generated from, $\boldsymbol{\mu}_{x_t} = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0$,

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)$$

$$\mathbb{E}[\boldsymbol{\mu}_{x_t}|\boldsymbol{x}_t] = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) \qquad \therefore \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}}$$

(b)

(c) :

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\nabla \log p(\boldsymbol{x}_t) \qquad \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

i.

(d) $$\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{\alpha_t} \left[\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \nabla \log$$

is a nn learns to predict the score function

(e) scale score function log p intuitively source noise

$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0}{\sqrt{\bar{\alpha}_t}}$$

$$\therefore (1-\bar{\alpha}_t)\nabla \log p(\boldsymbol{x}_t) = -\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_0$$

$$\nabla \log p(\boldsymbol{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_0$$

i.

DDPM

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** $\mathbf{x}_0$

ControlNet

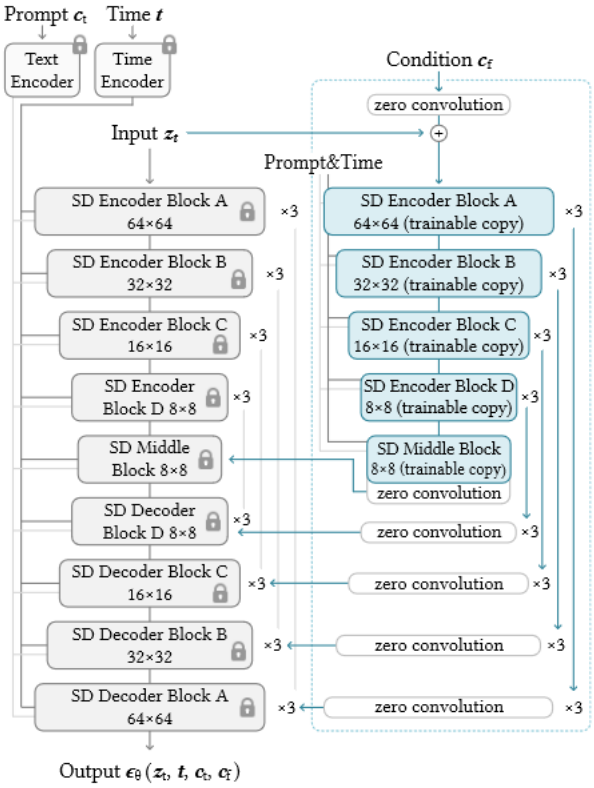
Motivation

1. , edge maps, human pose skeletons, segmentation maps, depth, normals,
2. T2I end-to-end
   (a) overfitting and catastrophic forgetting

Contribution

1.
2. encoder layer trainnable copy
3. zero convolution layer diffusion

Method



(a) Stable Diffusion (b) ControlNet

:feature map : conditionnal vector

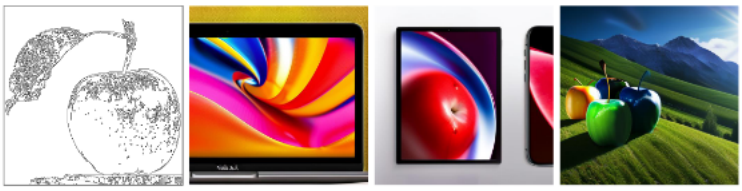
$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}; \Theta).$$

$$\boldsymbol{y}_c = \mathcal{F}(\boldsymbol{x}; \Theta) + \mathcal{Z}(\mathcal{F}(\boldsymbol{x} + \mathcal{Z}(\boldsymbol{c}; \Theta_{z1}); \Theta_c); \Theta_{z2}), \quad (2)$$

1. zero conv 0 , noise
2. trainable copy: 12 encoding blocks,1 moddle block
3. Controlnet decoder 12 skip connection
4. SD 512 x 512 64x64 latent images c embedd

Training

1. empty string 50%text-> ControlNet conditioning image (??CFG)



2. sudden convergence phenomenon

Inference

CFG

$$\text{as } \epsilon_{\text{prd}} = \epsilon_{\text{uc}} + \beta_{\text{cfg}}(\epsilon_{\text{c}} - \epsilon_{\text{uc}})$$

conditioning image

1. uc c: prompt guidance
2. c: guidance
   c, SD 13 block image guidance

controlnet

Ablative study

trainable copy zero conv 1 trainable copy conv 2)zero Gaussian

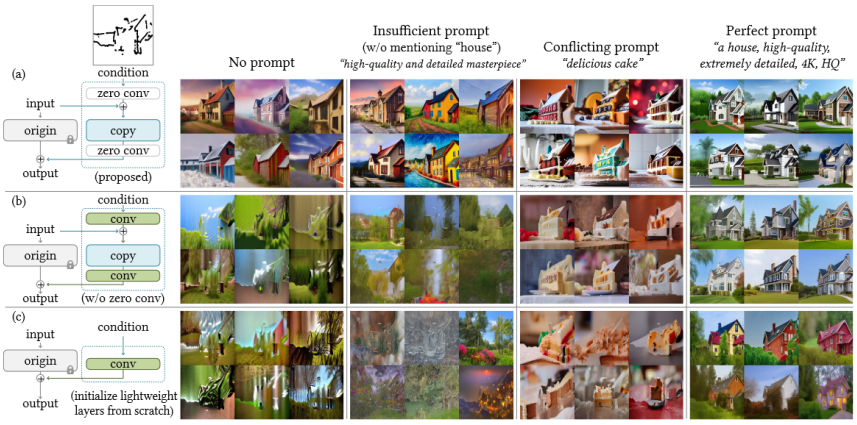


Figure 8: Ablative study of different architectures on a sketch condition and different prompt settings. For each setting, we show a random batch of 6 samples without cherry-picking. Images are at 512 × 512 and best viewed when zoomed in. The green "conv" blocks on the left are standard convolution layers initialized with Gaussian weights.

- user study
- ADE20K : to evaluate the conditioning fidelity.
- FID,CLIP-score,CLIP aesthetic score: distribution distance

ADE20K (GT)	VQGAN [19]	LDM [72]	PITI [89]	ControlNet-lite	ControlNet
0.58 ± 0.10	0.21 ± 0.15	0.31 ± 0.09	0.26 ± 0.16	0.32 ± 0.12	$\mathbf{0.35 \pm 0.14}$

Table 2: Evaluation of semantic segmentation label reconstruction (ADE20K) with Intersection over Union (IoU ↑).

Method	FID ↓	CLIP-score ↑	CLIP-aes. ↑
Stable Diffusion	6.09	0.26	6.32
VQGAN [19](seg.)*	26.28	0.17	5.14
LDM [72](seg.)*	25.35	0.18	5.15
PITI [89](seg.)	19.74	0.20	5.77
ControlNet-lite	17.92	0.26	6.30
ControlNet	15.27	0.26	6.31

DreamBooth

new concept learning(subject-driven)

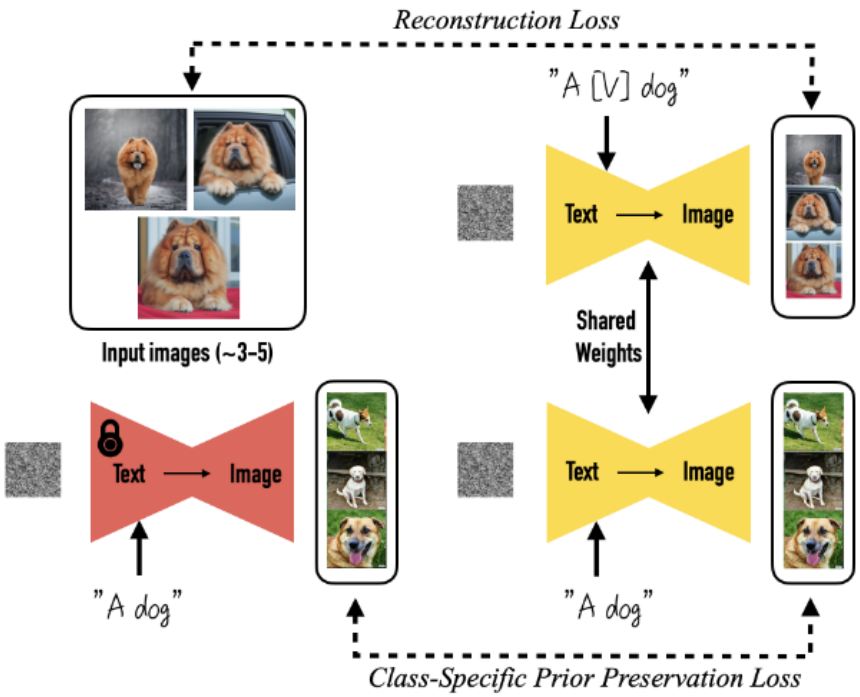
Motivation

- T2I same subject,). global editing, fine-grained control

Contribution

- fine-tune T2I :
  - text prompt identifier
  - lauguage shift: autogenous, class-specific prior preservation loss

Method



T2I

c: conditioning vector

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}\left[w_t \left\| \hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \mathbf{c}) - \mathbf{x} \right\|_2^2\right]$$

Personalization

1. GANs fine-tune overfit mode-collapse:

   large text-to-image diffusion models seem to excel at integrating new information into their domain without forgetting the prior or overfitting to a small set of training images.

Prompt

a [identifier] [class noun]

Identifier

identifier LM diffusion weak prior.

- rare identifier: , f: tockenizer; : decoded text
-

Class-specific Prior Preservation Loss

- Fine-tune Problem
  - finetune lauguage drift
    - (LM finetune diffusion "slowly forgets how to generate subjects of the same class as the target subject")
  - diversity
- loss
  - [class noun]

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}[w_t \|\hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \mathbf{c}) - \mathbf{x}\|_2^2 + \lambda w_{t'} \|\hat{\mathbf{x}}_\theta(\alpha_{t'} \mathbf{x}_{\text{pr}} + \sigma_{t'} \boldsymbol{\epsilon}', \mathbf{c}_{\text{pr}}) - \mathbf{x}_{\text{pr}}\|_2^2], \quad (2)$$

  -
    - prior-preservation term:

Experiments

- subject fidelity:
  - CLIP-I : average pairwise cosine similarity between CLIP embeddings of generated and real images( )
  - DINO:
- prompt fidelity
  + CLIP-T: average pairwise cosine similarity between CLIP embeddings of generated and real images

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑
Real Images	0.774	0.885	N/A
DreamBooth (Imagen)	**0.696**	**0.812**	**0.306**
DreamBooth (Stable Diffusion)	0.668	0.803	0.305
Textual Inversion (Stable Diffusion)	0.569	0.780	0.255

Table 1. Subject fidelity (DINO, CLIP-I) and prompt fidelity (CLIP-T, CLIP-T-L) quantitative metric comparison.

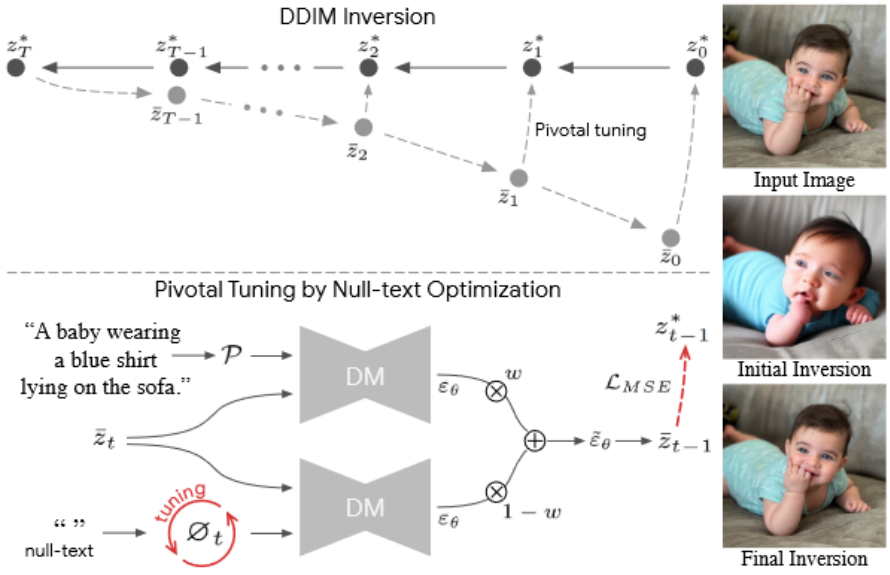
Method	Subject Fidelity ↑	Prompt Fidelity ↑
DreamBooth (Stable Diffusion)	**68%**	**81%**
Textual Inversion (Stable Diffusion)	22%	12%
Undecided	10%	7%

Imagen SD

- Prior preservation loss
- class-prior
-

-
  - context
    - a weak prior for contexts,
- context-appearance entanglement
- overfitting to the real images prompt training set setting
-

Null-text Inversion



Inversion: uncond null-text