Frontiers in Diffusion Model Technologies (1)

XIN GAO

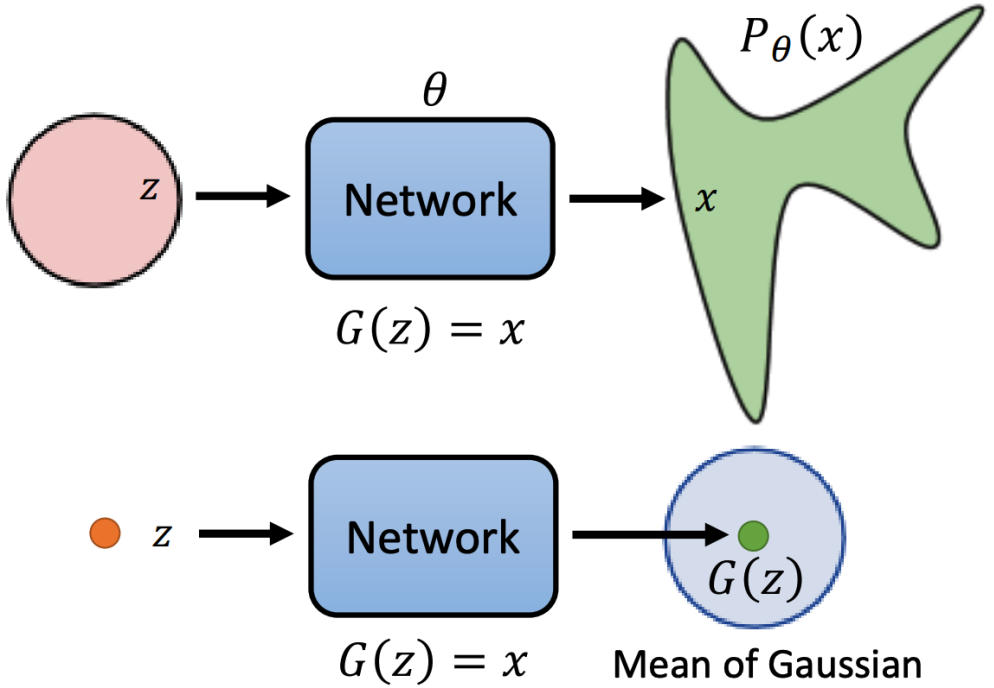
2024.12.8

Content

- **Theoretical foundation** :
  - DDPM
  - DDIM
  - SDE and ODE
  - Conditional Guidance
- **Development Timeline**
- **Stable diffusion**
  - Latent Diffusion
  - VQ-VAE
  - DiT
- **Latest Methodology**: IC-Light (ICLR 2025)

VAE and ELBO

- A **VAE** models the distribution $p_{data}(x)$ of the observed variable $x \in \mathbb{R}^n$ by jointly learning a stochastic latent variable $z \in \mathbb{R}^m$.
- **Generation** is performed by sampling z from the prior $p(z)$, then sampling x according to a probabilistic **decoder** $p_\theta(x|z)$ parametrized by $\theta \in \Theta$.
- How to update θ ? MLE $p_\theta(x) = \int_z p(z)p_\theta(x \mid z)dz$
- Identity:



$$\log p(x) = \int q(z) \log p(x) dz$$

Evidence

$$= \int q(z) \log \frac{p_\theta(x \mid z)p(z)}{p(z \mid x)} \frac{q(z)}{q(z)} dz$$

For arbitrary distribution $q(z)$ of z

$$= \int q(z) \log p_\theta(x \mid z)dz - KL[q(z)||p(z)] + KL[q(z)||p(z \mid x)]$$

Evidence Lower Bound (ELBO)

KL divergence

VAE and ELBO

Do a little math

$$\log p(x) = \int q(z) \log p_\theta(x \mid z) dz - KL[q(z)||p(z)] + KL[q(z)||p(z \mid x)]$$

Evidence — Evidence Lower Bound (ELBO) — KL divergence

- $\log p(x) \geq$ ELBO (KL divergence ≥ 0)

  Maximize ELBO => Increase $\log p(x)$
- What is $q(z)$?

If $q(z) = p(z|x)$, $KL = 0$, $\log p(x) =$ ELBO (EM Algorithm)

Unfortunately, the true posterior $p(z|x)$ is intractable, $p(z|x) = \frac{p(x|z)p(z)}{p(x)}$

$$\begin{aligned} ELBO &= \int q(z|x) \log \frac{p(x,z)}{q(z|x)} dz \\ &= \mathbb{E}_z[\log \frac{p(x,z)}{q(z|x)}] \\ &= \int q(z|x) \log \frac{p(x \mid z)p(z)}{q(z|x)} dz \\ &= \int q(z|x) \log p(x \mid z) dz - KL[q(z|x)||p(z)] \\ &= \mathbb{E}_z[\log p(x \mid z)] - KL[q(z|x)||p(z)] \end{aligned}$$

- We use an encoder network to approximate the posterior

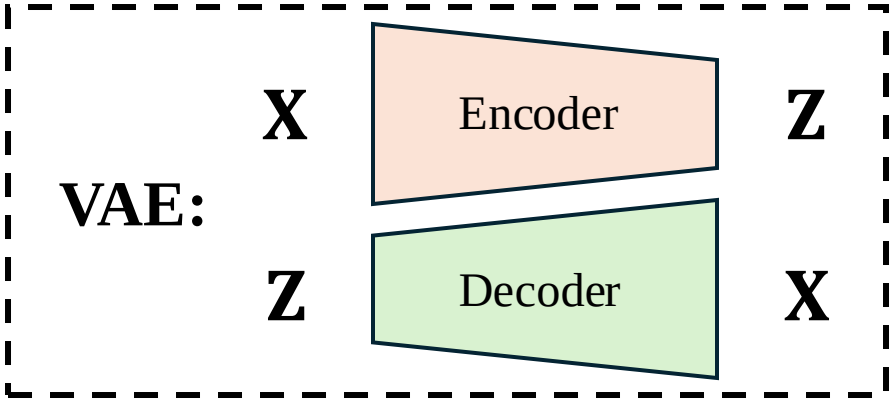
$$q_\phi(z|x) = \mathcal{N}(z; \mu(x), \Sigma(x))$$

- By replacing $q(z)$ with $q(z|x)$, maximizing ELBO not only minimizes KL but also approximates MLE

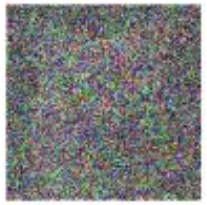
$$\log p(x) = \int q_\phi(z|x) \log p_\theta(x \mid z) dz - KL[q_\phi(z|x)||p(z)] + KL[q_\phi(z|x)||p(z \mid x)]$$

- Objective: $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - KL[q_\phi(z \mid x)||p(z)]$

Diffusion

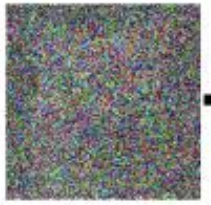
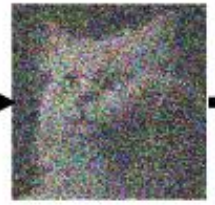


Forward Process

Diffuse X_0  Add noise  Add noise  X_T

Reverse Process

X_t

Denoise X_T  Denoise  Denoise  X_0

Diffusion models create data from noise by inverting the forward paths of data towards noise and have emerged as a powerful generative modeling technique for high-dimensional, perceptual data such as images and videos.

DDPM Denoising Diffusion Probabilistic Model

- Original image x_0
- Step-by-step decomposition, assuming multiple latent variables, $p(x_{1:T}|x_0) := \prod_{t=1}^{T} p(x_t|x_{t-1})$

  Markov chain $x_0 \to x_1 \to x_2 \to ... \to x_T$
- **Forward Process** with decreasing sequence $1 \geq \alpha_1 \geq \alpha_2 \geq ... \geq \alpha_T \geq 0$, $\beta_t := 1 - \alpha_t$

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{(1-\alpha_t)}\varepsilon_t,\ \varepsilon_t \sim \mathcal{N}(0,1),\ t = 1, ..., T$$

  Variable substitution / reparameterization trick $p(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)\mathbf{I})$

  Recursion (Noise $\bar{\varepsilon}_t$, linear combination of Gaussians still results in a Gaussian)

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{(1-\bar{\alpha}_t)}\bar{\varepsilon}_t, \text{ and } p(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t)\mathbf{I}) \qquad \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$$

- When T steps are large enough $\lim_{t \to +\infty} \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s = 0$ $p(x_T) \to \mathcal{N}(0,1)$
- How do we reconstruct the image step by step?

DDPM Denoising Diffusion Probabilistic Model

- **Bayes' Rule**: $p(x_{t-1}|x_t) = \frac{p(x_t|x_{t-1})p(x_{t-1})}{p(x_t)}$

But we do not know $p(x_{t-1}), p(x_t)$

- We know conditional the distribution given x_0

$$p(x_{t-1}|x_t, x_0) = \frac{p(x_t|x_{t-1})p(x_{t-1}|x_0)}{p(x_t|x_0)}$$

$p(x_t|x_{t-1}), p(x_{t-1}|x_0), p(x_t|x_0)$ are all Known Gaussian distributions

We can easily derive that $p(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \mathbf{I}\right)$

- But there's a gap. We can use x_t to predict/estimate x_0, $||x_0 - \mu_\theta(x_t)||^2$

$$p(x_{t-1}|x_t) \approx p(x_{t-1}|x_t, \hat{x}_0), \quad \text{where} \quad \hat{x}_0 = \mu_\theta(x_t)$$

By making a small adjustment, due to $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{(1-\bar{\alpha}_t)}\bar{\epsilon}_t)$

Predict the noise instead $\boldsymbol{\mu}_\theta(x_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{(1-\bar{\alpha}_t)}\boldsymbol{\epsilon}_\theta(x_t, t))$

➡ **Loss:** $\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(x_t, t)||^2 = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)||^2$

DDPM Denoising Diffusion Probabilistic Model

$$\log p(\boldsymbol{x}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] \tag{47}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \tag{48}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \tag{49}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)}\right] \tag{50}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)}\right] \tag{51}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}\right] \tag{52}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)\cancel{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}}{\cancel{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}}\right] \tag{53}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{\cancel{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}} + \log \frac{\cancel{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}\right] \tag{54}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \sum_{t=2}^{T} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}\right] \tag{55}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}\right] \tag{56}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}\right] \tag{57}$$

$$= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p(\boldsymbol{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))\right]}_{\text{denoising matching term}} \tag{58}$$

- ELBO
- Perspective from Latent Variable Model (like VAE)
- 知道就行，这种就是从隐变量模型出发，推导 ELBO 得到 loss，最终 loss带入分布后化简得到相同的结果。但中间有一步推导比较 trick，不如前两页的好理解
- 其实 Diffusion 就是一个中间隐变量是层级建模的VAE (Hierarchical VAE) + 将 encode 过程确定为了扩散过程 instead of learnable encoder

Training

$\bar{\alpha}_1, \bar{\alpha}_2, \ldots \bar{\alpha}_T$

x_0

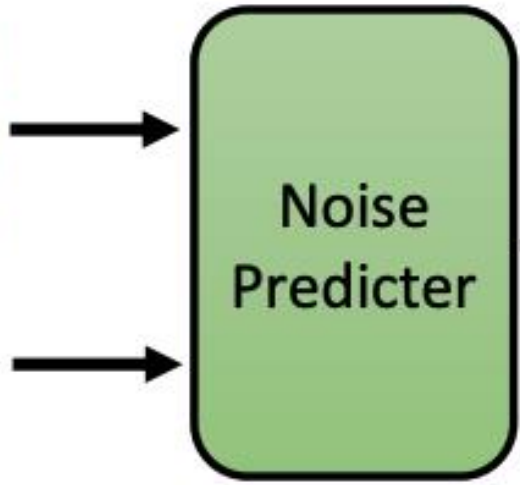
ε

Sample t

$\sqrt{\bar{\alpha}_t}$  $+ \sqrt{1 - \bar{\alpha}_t}$  $=$ 

x_0 ε

t



?????

ε

```python
# construct DDPM noise schedule
b_t = (beta2 - beta1) * torch.linspace(0, 1, timesteps + 1, device=device) + beta1
a_t = 1 - b_t
ab_t = torch.cumsum(a_t.log(), dim=0).exp()
ab_t[0] = 1
```

https://github.com/Ryota-Kawamura/How-Diffusion-Models-Work/tree/main

```python
# helper function: perturbs an image to a specified noise level
def perturb_input(x, t, noise):
    return ab_t.sqrt()[t, None, None, None] * x + (1 - ab_t[t, None, None, None]).sqrt() * noise
```

```python
# set into train mode
nn_model.train()

for ep in range(n_epoch):
    print(f'epoch {ep}')

    # linearly decay learning rate
    optim.param_groups[0]['lr'] = lrate*(1-ep/n_epoch)

    pbar = tqdm(dataloader, mininterval=2 )
    for x, _ in pbar:   # x: images
        optim.zero_grad()
        x = x.to(device)

        # perturb data
        noise = torch.randn_like(x)
        t = torch.randint(1, timesteps + 1, (x.shape[0],)).to(device)
        x_pert = perturb_input(x, t, noise)

        # use network to recover noise
        pred_noise = nn_model(x_pert, t / timesteps)

        # loss is mean squared error between the predicted and true noise
        loss = F.mse_loss(pred_noise, noise, reduction='sum') / x.shape[0]
        print(f'loss: {loss.item():.4f}', end='\r')
        loss.backward()

        optim.step()
```

x_0: clean image

ε: noise

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ ⋯ sample clean image
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ⋯ sample a noise
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

Noisy image

Target Noise

Noise predictor

$\bar{\alpha}_1, \bar{\alpha}_2, \dots \bar{\alpha}_T$ → smaller

DATASET

INPUT

Noise Amount	Noisy Image
3	
14	
7	
42	
2	
21	

OUTPUT / LABEL

Noise sample






MODEL

Noise Predictor

(UNet)

Inference

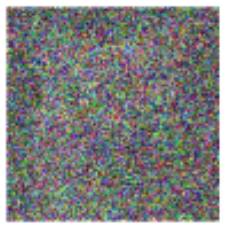
Algorithm 2 Sampling

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

sample a noise?!

$\bar{\alpha}_1, \bar{\alpha}_2, \ldots \bar{\alpha}_T$

$\alpha_1, \alpha_2, \ldots \alpha_T$



x_T

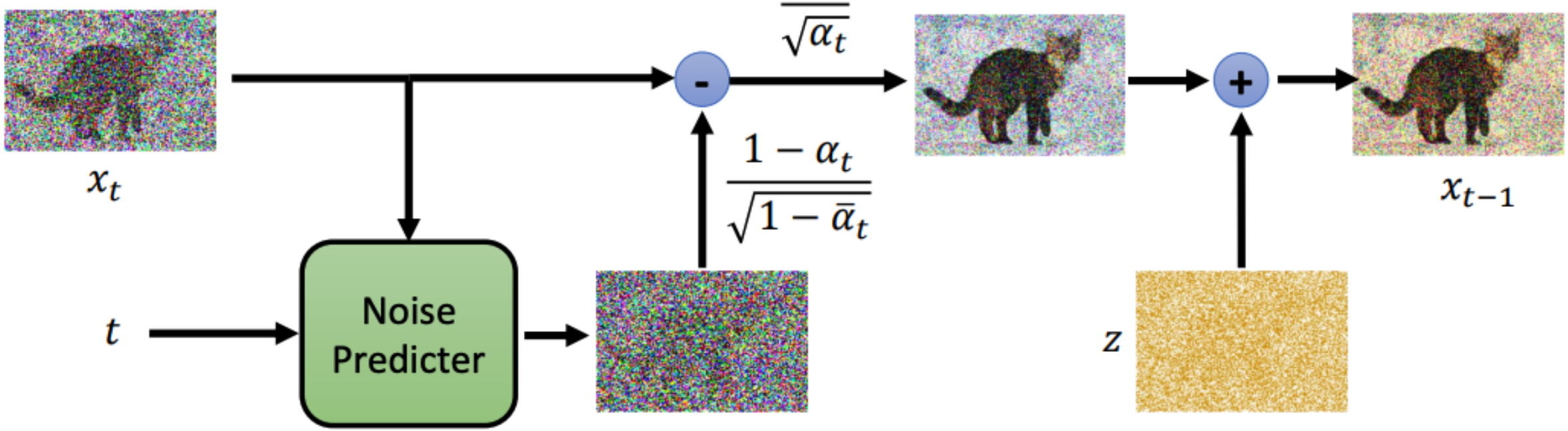
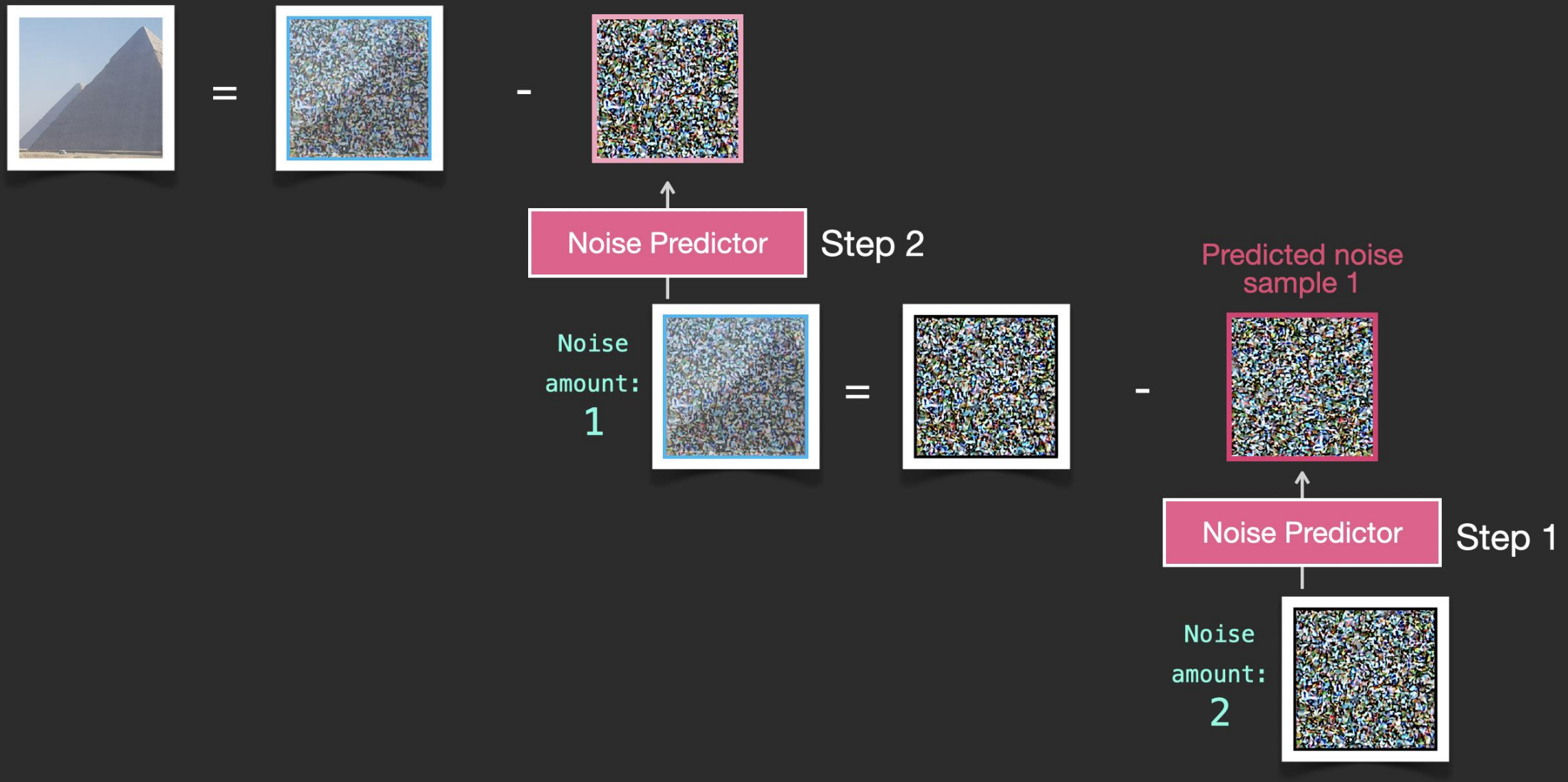


Image Generation by Reverse Diffusion (Denoising)



Stochasticity

Think again about the stochasticity

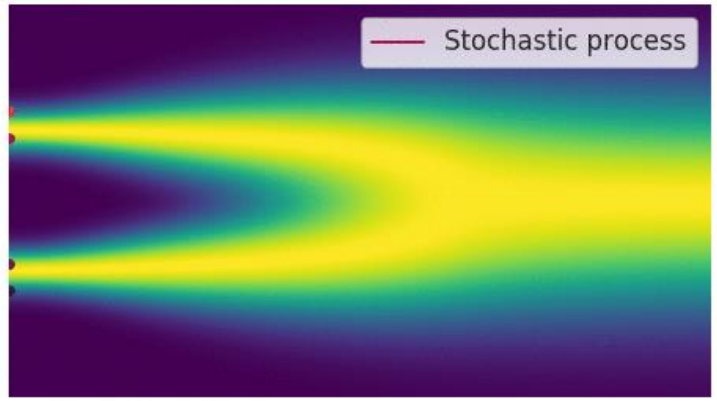
$$p(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathbf{I}\right)$$

$$\hat{x}_0 = \boldsymbol{\mu}_\theta(x_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{(1-\bar{\alpha}_t)}\boldsymbol{\epsilon}_\theta(x_t, t))$$

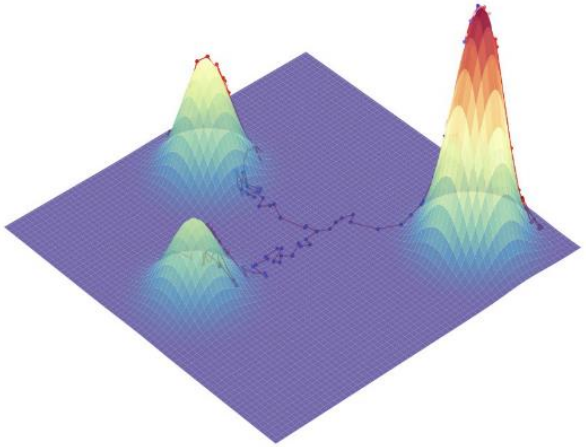
$$q(x_{t-1}|x_t) \approx \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \sigma_t\mathbf{I})$$

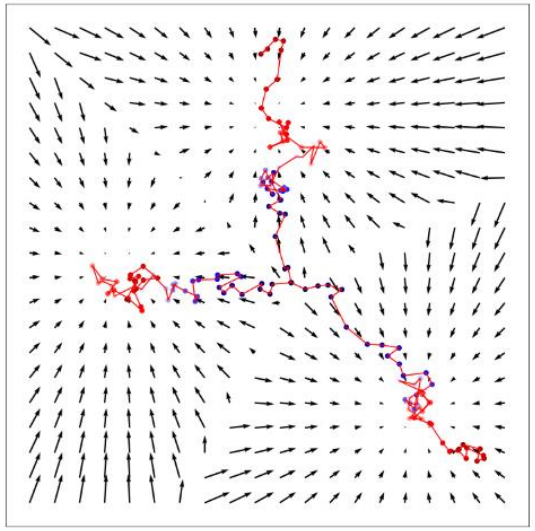
$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(x_t, t)\right) + \sigma_t\boldsymbol{\epsilon}$$

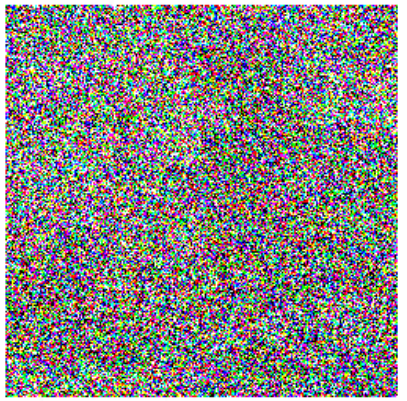
Sampling from a distribution !

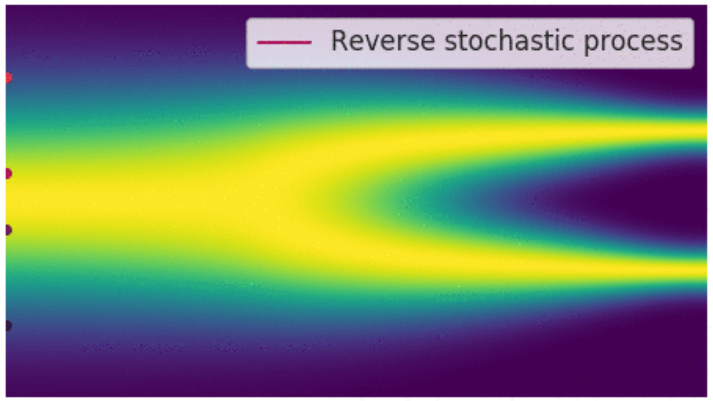


Perturbing data to noise with a continuous-time stochastic process.









Generate data from noise by reversing the perturbation procedure.

DDIM Denoising Diffusion Implicit Model

$$x_0 \underbrace{\xrightarrow{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\ \text{推导}} p(\boldsymbol{x}_t|\boldsymbol{x}_0)}_{\text{Forward/Diffuse}} x_T \underbrace{\xrightarrow{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ \text{近似}} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}_{\text{Reverse/Denoise}} x_0$$

- **Training**: The loss only relies on $p(x_t|x_0)$

$$\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(x_t,t)||^2 = \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}||\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t)||^2$$

- **Sampling**: Each step sampling only relies on $p(x_{t-1}|x_t)$

Maybe we do not need to set $p(x_t|x_{t-1})$ and assume Markov chain process ?

$$p(x_{t-1}|x_t,x_0) = \frac{p(x_t|x_{t-1})p(x_{t-1}|x_0)}{p(x_t|x_0)} \quad (*)$$

$$\int p(x_{t-1}|x_t,x_0)p(x_t|x_0)dx_t = p(x_{t-1}|x_0) \quad (**)$$

- Actually we have more distributions $p(x_{t-1}|x_t,x_0)$ to satisfy Eq. $(**)$

Undetermined Coefficients $\kappa_t,\lambda_t,\sigma_t$, $p(x_{t-1}|x_t,x_0)=\mathcal{N}(x_{t-1};\kappa_t x_t+\lambda_t x_0,\sigma_t^2\mathbf{I})$

DDIM Denoising Diffusion Implicit Model

$$x_0 \quad \underbrace{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \xrightarrow{\text{推导}} p(\boldsymbol{x}_t|\boldsymbol{x}_0)}_{\text{Forward/Diffuse}} \quad x_T \quad \underbrace{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \xrightarrow{\text{近似}} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}_{\text{Reverse/Denoise}} \quad x_0$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\bar{\varepsilon}_t, \text{ and } p(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\int \boxed{p(x_{t-1}|x_t,x_0)} p(x_t|x_0)dx_t = p(x_{t-1}|x_0) \quad (**) \qquad p(x_{t-1}|x_t,x_0) = \mathcal{N}(x_{t-1}; \kappa_t x_t + \lambda_t x_0, \sigma_t^2\mathbf{I})$$

Solution: $\kappa_t = \dfrac{\sqrt{\bar{\beta}_{t-1}-\sigma_t^2}}{\sqrt{\bar{\beta}_t}}, \quad \lambda_t = \sqrt{\bar{\alpha}_{t-1}} - \dfrac{\sqrt{\bar{\alpha}_t}\sqrt{\bar{\beta}_{t-1}-\sigma_t^2}}{\sqrt{\bar{\beta}_t}}, \quad \sigma_t$

α, β 相关的参数都是预先设定好的超参数，是已知的

- **DDPM**: $\sigma_t^2 = \dfrac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$
- **DDIM**: $\sigma_t^2 = 0$ Implicit 隐式的概率模型，确定性采样过程，不带随机性
- **Larger covariance**: $\sigma_t^2 = \beta_t$

$$p(x_{t-1}|x_t,x_0) = \frac{p(x_t|x_{t-1})p(x_{t-1}|x_0)}{p(x_t|x_0)}$$

Remark: 在给定 $p(x_{t-1}|x_t,x_0)$ 后，我们还可以反推出 $p(x_t|x_{t-1})$，即知道每一步是怎么扩散到噪声的

DDIM Denoising Diffusion Implicit Model

- **Accelerated Generation Process**

$$x_0 \quad \underbrace{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \xrightarrow{\text{推导}} p(\boldsymbol{x}_t|\boldsymbol{x}_0)}_{\text{Forward/Diffuse}} \quad x_T \quad \underbrace{p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \xrightarrow{\text{近似}} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}_{\text{Reverse/Denoise}} \quad x_0$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\bar{\varepsilon}_t, \text{ and } p(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Given σ_t : $\kappa_t = \frac{\sqrt{\bar{\beta}_{t-1}-\sigma_t^2}}{\sqrt{\bar{\beta}_t}}$, $\lambda_t = \sqrt{\bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_t}\sqrt{\bar{\beta}_{t-1}-\sigma_t^2}}{\sqrt{\bar{\beta}_t}}$ $\qquad p(x_{t-1}|x_t,x_0) = \mathcal{N}(x_{t-1}; \kappa_t x_t + \lambda_t x_0, \sigma_t^2\mathbf{I})$

Suppose that an increasing subsequence of $[1, \ldots T]$: $[\tau_1, \ldots, \tau_S]$

$$x_0 \quad \underbrace{p(\boldsymbol{x}_{\tau_t}|\boldsymbol{x}_0)}_{\text{Forward/Diffuse}} \xrightarrow{\text{推导}} x_T \quad \underbrace{p(\boldsymbol{x}_{\tau_{t-1}}|\boldsymbol{x}_{\tau_t},\boldsymbol{x}_0) \xrightarrow{\text{近似}} p(\boldsymbol{x}_{\tau_{t-1}}|\boldsymbol{x}_{\tau_t})}_{\text{Reverse/Denoise}} \quad x_0$$

It is allowed to skip steps! Original 1000 steps, 10 steps per jump => 100 steps, 20 steps per jump => 50 steps

SDE

- Forward process in DDPM: $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{(1-\alpha_t)}\varepsilon_t,\ \varepsilon_t \sim \mathcal{N}(0,1),\ t = 1, ..., T$

  连续化 一般化 $x_{t+\Delta t} - x_t = \boldsymbol{f}_t(x_t)\Delta t + g_t\sqrt{\Delta t}\boldsymbol{\varepsilon},\quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}),\quad \Delta t \to 0$

=> SDE: $dx = f_t(x)dt + g_t dw$ ⟹ w: Wiener process or 布朗运动，是一个随机过程，具有独立增量和连续轨迹，增量 $dw \sim \mathcal{N}(0, dt)$

  - Drift coefficient $f_t(x)dt$: 系统的确定性变化
  - Diffusion coefficient $g_t dw$:由随机扰动引起的不确定变化

- 概率分布形式 $p(x_{t+\Delta t}|x_t) = \mathcal{N}\left(x_{t+\Delta t}; x_t + \boldsymbol{f}_t(x_t)\Delta t, g_t^2 \Delta t \mathbf{I}\right) \propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \boldsymbol{f}_t(x_t)\Delta t\|^2}{2g_t^2 \Delta t}\right)$
- 逆向过程推导

$$p(x_t|x_{t+\Delta t}) = \frac{p(x_{t+\Delta t}|x_t)p(x_t)}{p(x_{t+\Delta t})} = p(x_{t+\Delta t}|x_t)\exp(\log p(x_t) - \log p(x_{t+\Delta t}))$$

$$\propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \boldsymbol{f}_t(x_t)\Delta t\|^2}{2g_t^2 \Delta t} + \log p(x_t) - \log p(x_{t+\Delta t})\right)$$

Δt 足够小，Taylor expansion: $\log p(x_{t+\Delta t}) \approx \log p(x_t) + (\boldsymbol{x}_{t+\Delta t} - x_t) \cdot \nabla_{x_t} \log p(x_t) + \Delta t \frac{\partial}{\partial t} \log p(x_t)$

SDE

DDPM

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{(1-\alpha_t)}\varepsilon_t$$

正向 SDE $\boxed{dx = f_t(x)dt + g_t dw}$

$$p(x_t|x_{t+\Delta t}) \propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \boldsymbol{f}_t(x_t)\Delta t\|^2}{2g_t^2\Delta t} + \log p(x_t) - \log p(x_{t+\Delta t})\right)$$

$$\log p(x_{t+\Delta t}) \approx \log p(x_t) + (\boldsymbol{x}_{t+\Delta t} - x_t)\cdot\nabla_{x_t}\log p(x_t) + \Delta t\frac{\partial}{\partial t}\log p(x_t)$$

$$p(x_t|x_{t+\Delta t}) \propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \left[\boldsymbol{f}_t(x_t) - g_t^2\nabla_{x_t}\log p(x_t)\right]\Delta t\|^2}{2g_t^2\Delta t} + \mathcal{O}(\Delta t)\right), \quad \Delta t \to 0, \mathcal{O}(\Delta t) \to 0$$

$$p(x_t|x_{t+\Delta t}) \propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \left[f_t(x_t) - g_t^2\nabla_{x_t}\log p(x_t)\right]\Delta t\|^2}{2g_t^2\Delta t}\right)$$

$$\approx \exp\left(-\frac{\|\boldsymbol{x}_t - \boxed{x_{t+\Delta t} + \left[\boldsymbol{f}_{t+\Delta t}(x_{t+\Delta t}) - g_{t+\Delta t}^2\nabla_{x_{t+\Delta t}}\log p(x_{t+\Delta t})\right]\Delta t}\|^2}{2\boxed{g_{t+\Delta t}^2\Delta t}}\right)$$

$$x_t - x_{t+\Delta t} = -\left[\boldsymbol{f}_{t+\Delta t}(x_{t+\Delta t}) - g_{t+\Delta t}^2\nabla_{x_{t+\Delta t}}\log p(x_{t+\Delta t})\right]\Delta t + g_{t+\Delta t}\sqrt{\Delta t}\boldsymbol{\varepsilon}$$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(x_t, t)\right) + \sigma_t\boldsymbol{\varepsilon}$$

逆向 SDE $\Delta t \to 0,$ $\boxed{dx = [\boldsymbol{f}_t(x) - g_t^2\nabla_x \log p_t(x)]dt + g_t d\boldsymbol{w}}$

Loss: $\mathbb{E}_{x_0, x_t\sim p(x_t|x_0)\tilde{p}(x_0)}\left[\|\boldsymbol{s}_\theta(x_t, t) - \nabla_{x_t}\log p(x_t|x_0)\|^2\right]$

$$\mathbb{E}_{x_0,\boldsymbol{\varepsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}||\boldsymbol{\varepsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}, t)||^2\right]$$

Loss 的推导本次省略

SDE and ODE 大一统

$dx = f_t(x)dt + g_t dw$ (#) ⟹ Fokker-Planck 方程 描述边际分布的 PDE $\frac{\partial}{\partial t}p_t(x) = -\nabla_x \cdot [f_t(x)p_t(x)] + \frac{1}{2}g_t^2 \nabla_x \cdot \nabla_x p_t(x)$

对 FP 方程做等式变换，注意以下式子对 $\forall \sigma_t$ 都成立：

$$\begin{aligned}\frac{\partial}{\partial t}p_t(x) &= -\nabla_x \cdot \left[f_t(x)p_t(x) - \frac{1}{2}(g_t^2 - \sigma_t^2)\nabla_x p_t(x)\right] + \frac{1}{2}\sigma_t^2 \nabla_x \cdot \nabla_x p_t(x) \\ &= -\nabla_x \cdot \left[\left(f_t(x) - \frac{1}{2}(g_t^2 - \sigma_t^2)\nabla_x \log p_t(x)\right)p_t(x)\right] + \frac{1}{2}\sigma_t^2 \nabla_x \cdot \nabla_x p_t(x)\end{aligned}$$

我们发现这个 FP 方程也是以下 SDE 的 FP 方程：

$$dx = \left(f_t(x) - \frac{1}{2}(g_t^2 - \sigma_t^2)\nabla_x \log p_t(x)\right)dt + \sigma_t dw \quad (\#\#)$$

也就是说式 (#) 和式 (##) 对应的marginal distribution $p_t(x)$ 完全相同
即存在不同方差的前向过程，产生的 marginal distribution 完全相同

同样的，我们可以写出 (##) 的反向SDE：

$$dx = \left(f_t(x) - \frac{1}{2}(g_t^2 + \sigma_t^2)\nabla_x \log p_t(x)\right)dt + \sigma_t dw$$

SDE and ODE 大一统

$$dx = \left(\boldsymbol{f}_t(x) - \frac{1}{2}(g_t^2 - \sigma_t^2)\nabla_x \log p_t(x) \right) dt + \sigma_t d\boldsymbol{w} \quad (\#\#)$$

$$dx = \left(\boldsymbol{f}_t(x) - \frac{1}{2}(g_t^2 + \sigma_t^2)\nabla_x \log p_t(x) \right) dt + \sigma_t d\boldsymbol{w}$$

What if $\sigma_t = 0$?

Probability flow ODE

$$dx = \left(\boldsymbol{f}_t(x) - \frac{1}{2} g_t^2 \nabla_x \log p_t(x) \right) dt$$

Deterministic transform

- Deterministic representation
- ODE Accelerated Solver Algorithm

Remark: The forward process and reverse process of ODE are exactly the same

Score Function

- **Connecting gradient with the predicted noise:** $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx \mathcal{N}(\boldsymbol{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \sigma_t \mathbf{I})$

  In this case, we apply it to predict the true posterior mean of $\boldsymbol{x}_t$ given its samples. From Equation 70, we know that:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I})$$

  Then, by Tweedie's Formula, we have:

$$\mathbb{E}\left[\boldsymbol{\mu}_{x_t}|\boldsymbol{x}_t\right] = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}_t}\log p(\boldsymbol{x}_t) \tag{131}$$

$$\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla\log p(\boldsymbol{x}_t) \tag{132}$$

$$\therefore \boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla\log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} \tag{133}$$

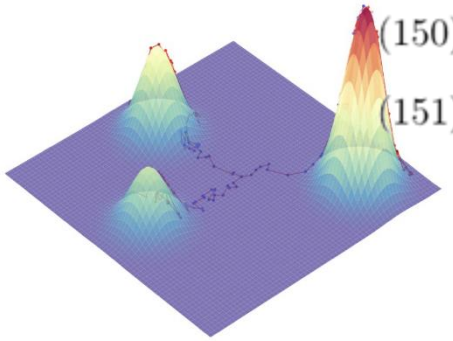
$$\boldsymbol{x}_0 = \frac{\boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla\log p(\boldsymbol{x}_t)}{\sqrt{\bar{\alpha}_t}} = \frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t}{\sqrt{\bar{\alpha}_t}} \tag{149}$$

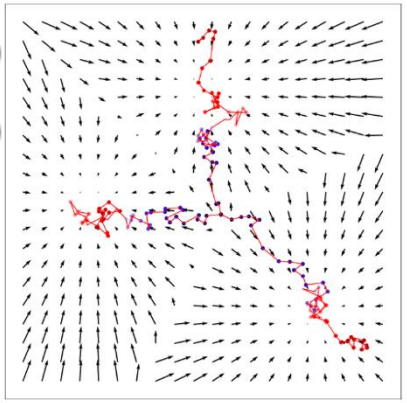
$$\therefore (1-\bar{\alpha}_t)\nabla\log p(\boldsymbol{x}_t) = -\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t \tag{150}$$

$$\nabla\log p(\boldsymbol{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t \tag{151}$$

- What is **Tweedie's Formula** ?

- **Conclusion**: $\boldsymbol{\epsilon}_t = -\sqrt{1-\bar{\alpha}_t}\nabla\log p(\boldsymbol{x}_t)$





Score Function Tweedie’s Formula 补充

Tweedie's Formula 说明：**后验均值（posterior mean）可以通过观测值加上噪声方差乘以观测值的对数概率密度的梯度来计算。**

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1)$$

$$p(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$p(x_t|x_0) = \frac{1}{(2\pi(1-\bar{\alpha}_t))^{d/2}}\exp(-\frac{||x_t - \sqrt{\bar{\alpha}_t}x_0||^2}{2(1-\bar{\alpha}_t)})$$

对原始图像 x_0 的后验和后验均值

$$p(x_0|x_t) = \frac{p(x_t|x_0)p(x_0)}{p(x_t)}$$

$$\mathbb{E}[x_0|x_t] = \int x_0 p(x_0|x_t)dx_0 = \frac{1}{p(x_t)}\int x_0 p(x_t|x_0)p(x_0)dx_0$$

$$\mathbb{E}[x_0|x_t] = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t + (1-\bar{\alpha}_t)\nabla \log p(x_t))$$

$$\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon_\theta(x_t, t))$$

做一点小的推导：

$$p(x_t) = \int p(x_t|x_0)p(x_0)dx_0$$

$$\nabla p(x_t) = \int \nabla p(x_t|x_0)p(x_0)dx_0$$

$$\nabla p(x_t|x_0) = -\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{1-\bar{\alpha}_t}p(x_t|x_0)dx_0$$

$$\nabla p(x_t) = \int -\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{1-\bar{\alpha}_t}p(x_t|x_0)p(x_0)dx_0$$

$$\nabla \log p(x_t) = \frac{1}{p(x_t)}\nabla p(x_t)$$

$$\nabla \log p(x_t) = \frac{1}{p(x_t)}\int -\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{1-\bar{\alpha}_t}p(x_t|x_0)p(x_0)dx_0$$

Guidance Two ways to inject condition

- **Way 1: Classifier-Guidance: Use an unconditional generative model $p_\theta(x_{t-1}|x_t)$ (已经训练好的) + Classifier $p_\phi(y|x_t)$**

  Injecting Condition y in the reverse process $dx = \left(\boldsymbol{f}_t(x) - \frac{1}{2}(g_t^2 + \sigma_t^2)\nabla_x \log p_t(x) \right) dt + \sigma_t d\boldsymbol{w}$

$$
\begin{aligned}
\nabla_{x_t} \log p\left(x_t \mid y\right) &= \nabla \log \left(\frac{p\left(x_t\right) p_\phi\left(y \mid x_t\right)}{p(y)} \right) \\
&= \nabla \log p\left(x_t\right) + \nabla \log p_\phi\left(y \mid x_t\right) - \nabla \log p(y) \\
&= \underbrace{\nabla \log p\left(x_t\right)}_{\text{unconditional score}} + \underbrace{\nabla \log p_\phi\left(y \mid x_t\right)}_{\text{classifier gradient}}
\end{aligned}
$$

$$
\boldsymbol{\epsilon}_t = -\sqrt{1-\bar{\alpha}_t}\nabla \log p(x_t) \quad \Rightarrow \quad \hat{\boldsymbol{\epsilon}}(x_t, t) := \boldsymbol{\epsilon}_\theta(x_t, t) - \gamma\sqrt{1-\bar{\alpha}_t}\nabla_{x_t} \log p_\phi(y|x_t)
$$

- 注意，只改变采样过程，相当于对梯度做一个可控的偏移

Guidance Two ways to inject condition

- **Way 2: Classifier-Free Guidance (CFG)**

直接改变训练过程 $p_\theta(x_{t-1}|x_t, y),\ y = label\ or\ \emptyset$

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training

1: **repeat**
2: $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$ ▷ Sample data with conditioning from the dataset
3: $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond} ▷ Randomly discard conditioning to train unconditionally
4: $\lambda \sim p(\lambda)$ ▷ Sample log SNR value
5: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$ ▷ Corrupt data to the sampled log SNR value
7: Take gradient step on $\nabla_\theta \left\| \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon} \right\|^2$ ▷ Optimization of denoising model
8: **until** converged

$\boldsymbol{\epsilon}_\theta\left(x_t, t, y\right)$ or $\boldsymbol{\epsilon}_\theta\left(x_t, t, \emptyset\right)$

Sampling

$$\begin{aligned}\hat{\boldsymbol{\epsilon}}(x_t, t, y) &:= \boldsymbol{\epsilon}_\theta(x_t, t, y) - \gamma\sqrt{1-\bar{\alpha}_t}\nabla_{x_t}\log p_\phi(y|x_t) \\ &= \boldsymbol{\epsilon}_\theta(x_t, t, y) - \gamma\sqrt{1-\bar{\alpha}_t}(\nabla_{x_t}\log p_\phi(x_t|y) - \nabla_{x_t}\log p_\phi(x_t)) \\ &= \boldsymbol{\epsilon}_\theta(x_t, t, y) + \gamma(\boldsymbol{\epsilon}_\theta(x_t, t, y) - \boldsymbol{\epsilon}_\theta(x_t, t, \emptyset)) \\ &= (1+\gamma)\boldsymbol{\epsilon}_\theta(x_t, t, y) - \gamma\boldsymbol{\epsilon}_\theta(x_t, t, \emptyset)\end{aligned}$$

采样时通过有条件和无条件两种形式做一个线性外推，用引导系数调节控制程度

Timeline (1)

https://arxiv.org/pdf/1503.03585 https://arxiv.org/pdf/1907.05600V1 https://arxiv.org/pdf/2006.11239V1 https://arxiv.org/pdf/2011.13456v1

Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein JASCHA@STANFORD.EDU
Stanford University
Eric A. Weiss EAWEISS@BERKELEY.EDU
University of California, Berkeley
Niru Maheswaranathan NIRUM@STANFORD.EDU
Stanford University
Surya Ganguli SGANGULI@STANFORD.EDU
Stanford University

Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However, these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function $\phi(\mathbf{x})$ yielding the flexible distribution $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$, where Z is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff–for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective[1].

1.1. Diffusion probabilistic models

We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

[1]Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

Physics Foundation

2015.11

Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song Stanford University yangsong@cs.stanford.edu
Stefano Ermon Stanford University ermon@cs.stanford.edu

Abstract

We introduce a new generative model where samples are produced via Langevin dynamics using gradients of the data distribution estimated with score matching. Because gradients might be ill-defined when the data resides on low-dimensional manifolds, we perturb the data with different levels of Gaussian noise and jointly estimate the corresponding scores, *i.e.*, the vector fields of gradients of the perturbed data distribution for all noise levels. For sampling, we propose an annealed Langevin dynamics where we use gradients corresponding to gradually decreasing noise levels as the sampling process gets closer to the data manifold. Our framework allows flexible model architectures, requires no sampling during training or the use of adversarial methods, and provides a learning objective that can be used for principled model comparisons. Our models produce samples comparable to GANs on MNIST, CelebA and CIFAR-10 datasets, achieving a new state-of-the-art inception score of 8.91 on CIFAR-10. Additionally, we demonstrate that our models learn effective representations via image inpainting experiments.

1 Introduction

Generative models have many applications in machine learning. To list a few, they have been used to generate high-fidelity images [22, 4], synthesize realistic speech and music fragments [47], improve the performance of semi-supervised learning [24, 8], detect adversarial examples and other anomalous data [44], imitation learning [19], and explore promising states in reinforcement learning [35]. Recent progress is mainly driven by two approaches: likelihood-based methods [14, 25, 9, 49] and generative adversarial networks (GAN [13]). The former uses log-likelihood (or a suitable surrogate) as the training objective, while the latter uses adversarial training to minimize f-divergences [34] or integral probability metrics [2, 45] between model and data distributions.

Although likelihood-based models and GANs have achieved great success, they have some intrinsic limitations. For example, likelihood-based models either have to use specialized architectures to build a normalized probability model (*e.g.*, autoregressive models, flow models), or use surrogate losses (*e.g.*, the evidence lower bound used in variational auto-encoders [25], contrastive divergence in energy-based models [18]) for training. GANs avoid some of the limitations of likelihood-based models, but their training can be unstable due to the adversarial training procedure. In addition, the GAN objective is not suitable for evaluating and comparing different GAN models. While other objectives exist for generative modeling, such as noise contrastive estimation [16] and minimum probability flow [29], these methods typically only work well for low-dimensional data.

In this paper, we explore a new principle for generative modeling based on estimating the *(Stein) score* [29] of the data density, which is the gradient of the log-density function with respect to the input dimensions. This is a vector field pointing in the direction where the log data density grows the most. We use a neural network trained with score matching [21] to learn this vector field from data. We then produce samples using Langevin dynamics, which approximately works by gradually

Preprint. Under review.

SGM

2019.07

Denoising Diffusion Probabilistic Models

Jonathan Ho UC Berkeley jonathanho@berkeley.edu
Ajay Jain UC Berkeley ajayj@berkeley.edu
Pieter Abbeel UC Berkeley pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at `https://github.com/hojonathanho/diffusion`.

1 Introduction

Deep generative models of all kinds have recently exhibited high quality samples in a wide variety of data modalities. Generative adversarial networks (GANs), autoregressive models, flows, and variational autoencoders (VAEs) have synthesized striking image and audio samples [12, 25, 3, 55, 35, 23, 10, 30, 41, 54, 24, 31, 42], and there have been remarkable advances in energy-based modeling and score matching that have produced images comparable to those of GANs [11, 52].

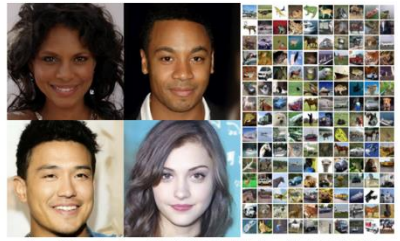


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Preprint. Under review.

DDPM

2020.06

Preprint. Work in progress.

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song* Stanford University yangsong@cs.stanford.edu
Jascha Sohl-Dickstein Google Brain jaschasd@google.com
Diederik P. Kingma Google Brain durk@google.com
Abhishek Kumar Google Brain abhishk@google.com
Stefano Ermon Stanford University ermon@cs.stanford.edu
Ben Poole Google Brain pooleb@google.com

ABSTRACT

Creating noise from data is easy; creating data from noise is generative modeling. We present a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known prior distribution by slowly injecting noise, and a corresponding reverse-time SDE that transforms the prior distribution back into the data distribution by slowly removing the noise. Crucially, the reverse-time SDE depends only on the time-dependent gradient field (a.k.a., score) of the perturbed data distribution. By leveraging advances in score-based generative modeling, we can accurately estimate these scores with neural networks, and use numerical SDE solvers to generate samples. We show that this framework encapsulates previous approaches in diffusion probabilistic modeling and score-based generative modeling, and allows for new sampling procedures. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE. We also derive an equivalent neural ODE that samples from the same distribution as the SDE, which enables exact likelihood computation, and improved sampling efficiency. In addition, our framework enables conditional generation with an unconditional model, as we demonstrate with experiments on class-conditional generation, image inpainting, and colorization. Combined with multiple architectural improvements, we achieve record-breaking performance for unconditional image generation on CIFAR-10 with an Inception score of 9.89 and FID of 2.20, a competitive likelihood of 3.10 bits/dim, and demonstrate high fidelity generation of 1024 × 1024 images for the first time from a score-based generative model.

1 INTRODUCTION

Two successful classes of probabilistic generative models involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption in order to form a generative model of the data. *Score matching with Langevin dynamics* (SMLD) (Song & Ermon, 2019) estimates the *score* (*i.e.*, the gradient of the log probability density) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decreasing noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a sequence of probabilistic models to reverse each step of the noise corruption, using knowledge of the functional form of the reverse distributions to make training tractable. For continuous state spaces, the DDPM training objective implicitly computes scores at each noise scale. We therefore refer to these two model classes together as *score-based generative models*.

Score-based generative models, and related techniques (Bordes et al., 2017; Goyal et al., 2017), have proven effective at generation of images (Song & Ermon, 2019; 2020; Ho et al., 2020), audio (Chen et al., 2020; Kong et al., 2020), graphs (Niu et al., 2020), and shapes (Cai et al., 2020). However, the

*Work done during an internship at Google Brain.

SDE

2020.11

Timeline (2)

https://arxiv.org/pdf/2010.02502v1

DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlinm,ermon}@cs.stanford.edu

ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation without adversarial training, yet they require simulating a Markov chain for many steps to produce a sample. To accelerate sampling, we present denoising diffusion implicit models (DDIMs), a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPMs. In DDPMs, the generative process is defined as the reverse of a Markovian diffusion process. We construct a class of non-Markovian diffusion processes that lead to the same training objective, but whose reverse process can be much faster to sample from. We empirically demonstrate that DDIMs can produce high quality samples 10× to 50× faster in terms of wall-clock time compared to DDPMs, allow us to trade off computation for sample quality, and can perform semantically meaningful image interpolation directly in the latent space. Our implementation is available at this link.

1 INTRODUCTION

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Karras et al., 2020; van den Oord et al., 2016a). In terms of image generation, generative adversarial networks (GANs, Goodfellow et al. (2014)) currently exhibits higher sample quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013), autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2016). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Arjovsky et al., 2017; Gulrajani et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhao et al., 2018).

Recent works on iterative generative models (Bengio et al., 2014), such as denoising diffusion probabilistic models (DDPM, Ho et al. (2020)) and noise conditional score networks (NCSN, Song & Ermon (2019)) have demonstrated the ability to produce samples comparable to that of GANs, without having to perform adversarial training. To achieve this, many denoising autoencoding models are trained to denoise samples corrupted by various levels of Gaussian noise. Samples are then produced by a Markov chain which, starting from white noise, progressively denoises it into an image. This generative Markov Chain process is either based on Langevin dynamics (Song & Ermon, 2019) or obtained by reversing a forward *diffusion process* that progressively turns an image into noise (Sohl-Dickstein et al., 2015).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because that the generative process (from noise to data) approximates the reverse of the forward *diffusion process* (from data to noise), which could have thousands of steps; iterating over all the steps is required to produce a single sample, which is much slower compared to GANs, which only needs one pass through a network. For example, it takes around 20 hours to sample 50k images of size 32 × 32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU. This becomes more problematic for larger images as sampling 50k images of size 256 × 256 could take nearly 1000 hours on the same GPU.

To close this efficiency gap between DDPMs and GANs, we present denoising diffusion implicit models (DDIMs). DDIMs are implicit probabilistic models (Mohamed & Lakshminarayanan, 2016) and are closely related to DDPMs, in the sense that they are trained with the same objective function. In Section 3, we generalize the forward *diffusion process* used by DDPMs, which is Markovian,

DDIM

2020.10

https://arxiv.org/pdf/2102.09672

Improved Denoising Diffusion Probabilistic Models

Alex Nichol * 1 Prafulla Dhariwal * 1

Abstract

Denoising diffusion probabilistic models (DDPM) are a class of generative models which have recently been shown to produce excellent samples. We show that with a few simple modifications, DDPMs can also achieve competitive log-likelihoods while maintaining high sample quality. Additionally, we find that learning variances of the reverse diffusion process allows sampling with an order of magnitude fewer forward passes with a negligible difference in sample quality, which is important for the practical deployment of these models. We additionally use precision and recall to compare how well DDPMs and GANs cover the target distribution. Finally, we show that the sample quality and likelihood of these models scale smoothly with model capacity and training compute, making them easily scalable. We release our code at https://github.com/openai/improved-diffusion.

1. Introduction

Sohl-Dickstein et al. (2015) introduced diffusion probabilistic models, a class of generative models which match a data distribution by learning to reverse a gradual, multi-step noising process. More recently, Ho et al. (2020) showed an equivalence between denoising diffusion probabilistic models (DDPM) and score based generative models (Song & Ermon, 2019; 2020), which learn a gradient of the log-density of the data distribution using denoising score matching (Hyvärinen, 2005). It has recently been shown that this class of models can produce high-quality images (Ho et al., 2020; Song & Ermon, 2020; Jolicoeur-Martineau et al., 2020) and audio (Chen et al., 2020b; Kong et al., 2020), but it has yet to be shown that DDPMs can achieve log-likelihoods competitive with other likelihood-based models such as autoregressive models (van den Oord et al., 2016c) and VAEs (Kingma & Welling, 2013). This raises various questions, such as whether DDPMs are capable of capturing all the modes of a distribution. Furthermore, while Ho et al. (2020) showed extremely good results on the CIFAR-10 (Krizhevsky, 2009) and LSUN (Yu et al., 2015) datasets, it is unclear how well DDPMs scale to datasets with higher diversity such as ImageNet. Finally, while Chen et al. (2020b) found that DDPMs can efficiently generate audio using a small number of sampling steps, it has yet to be shown that the same is true for images.

In this paper, we show that DDPMs can achieve log-likelihoods competitive with other likelihood-based models, even on high-diversity datasets like ImageNet. To more tightly optimise the variational lower-bound (VLB), we learn the reverse process variances using a simple reparameterization and a hybrid learning objective that combines the VLB with the simplified objective from Ho et al. (2020).

We find surprisingly that, with our hybrid objective, our models obtain better log-likelihoods than those obtained by optimizing the log-likelihood directly, and discover that the latter objective has much more gradient noise during training. We show that a simple importance sampling technique reduces this noise and allows us to achieve better log-likelihoods than with the hybrid objective.

After incorporating learned variances into our model, we surprisingly discovered that we could sample in fewer steps from our models with very little change in sample quality. While DDPM (Ho et al., 2020) requires hundreds of forward passes to produce good samples, we can achieve good samples with as few as 50 forward passes, thus speeding up sampling for use in practical applications. In parallel to our work, Song et al. (2020a) develops a different approach to fast sampling, and we compare against their approach, DDIM, in our experiments.

While likelihood is a good metric to compare against other likelihood-based models, we also wanted to compare the distribution coverage of these models with GANs. We use the improved precision and recall metrics (Kynkäänniemi et al., 2019) and discover that diffusion models achieve much higher recall for similar FID, suggesting that they do indeed cover a much larger portion of the target distribution.

Finally, since we expect machine learning models to consume more computational resources in the future, we evaluate the performance of these models as we increase model size and training compute. Similar to (Henighan et al.,

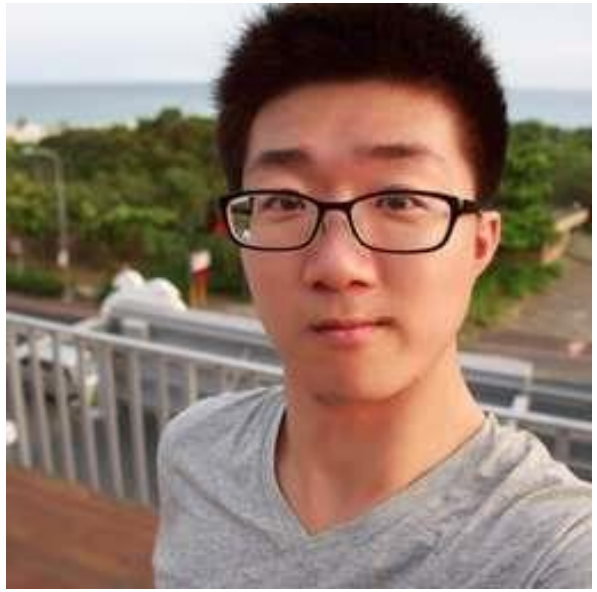
*Equal contribution [1]OpenAI, San Francisco, USA. Correspondence to: <alex@openai.com>, <prafulla@openai.com>.

Improved-diffusion

2021.02

Yang Song

Jiaming Song



- 2012-2016 Tsinghua University
- 2016 PhD at Stanford University, supervised by Stefano Ermon
- OpenAI NVIDIA

Timeline (3)

https://arxiv.org/pdf/2105.05233v1

Diffusion Models Beat GANs on Image Synthesis

Prafulla Dhariwal* OpenAI prafulla@openai.com

Alex Nichol* OpenAI alex@openai.com

Abstract

We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for sample quality using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128 × 128, 4.59 on ImageNet 256 × 256, and 7.72 on ImageNet 512 × 512, and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.85 on ImageNet 512 × 512. We release our code at https://github.com/openai/guided-diffusion.

1 Introduction

Figure 1: Selected samples from our best ImageNet 512 × 512 model (FID 3.85)

Over the past few years, generative models have gained the ability to generate human-like natural language [6], infinite high-quality synthetic images [5, 22, 44] and highly diverse human speech and music [57, 12]. These models can be used in a variety of ways, such as generating images from text prompts [63, 43] or learning useful feature representations [13, 7]. While these models are already

*Equal contribution

Classifier-Guidance

2021.05

https://arxiv.org/pdf/2207.12598

CLASSIFIER-FREE DIFFUSION GUIDANCE

Jonathan Ho & Tim Salimans
Google Research, Brain team
{jonathanho,salimans}@google.com

ABSTRACT

Classifier guidance is a recently introduced method to trade off mode coverage and sample fidelity in conditional diffusion models post training, in the same spirit as low temperature sampling or truncation in other types of generative models. Classifier guidance combines the score estimate of a diffusion model with the gradient of an image classifier and thereby requires training an image classifier separate from the diffusion model. It also raises the question of whether guidance can be performed without a classifier. We show that guidance can be indeed performed by a pure generative model without such a classifier: in what we call classifier-free guidance, we jointly train a conditional and an unconditional diffusion model, and we combine the resulting conditional and unconditional score estimates to attain a trade-off between sample quality and diversity similar to that obtained using classifier guidance.

1 INTRODUCTION

Diffusion models have recently emerged as an expressive and flexible family of generative models, delivering competitive sample quality and likelihood scores on image and audio synthesis tasks (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021b; Kingma et al., 2021; Song et al., 2021a). These models have delivered audio synthesis performance rivaling the quality of autoregressive models with substantially fewer inference steps (Chen et al., 2021; Kong et al., 2021), and they have delivered ImageNet generation results outperforming BigGAN-deep (Brock et al., 2019) and VQ-VAE-2 (Razavi et al., 2019) in terms of FID score and classification accuracy score (Ho et al., 2021; Dhariwal & Nichol, 2021).

Dhariwal & Nichol (2021) proposed *classifier guidance*, a technique to boost the sample quality of a diffusion model using an extra trained classifier. Prior to classifier guidance, it was not known how to generate "low temperature" samples from a diffusion model similar to those produced by truncated BigGAN (Brock et al., 2019) or low temperature Glow (Kingma & Dhariwal, 2018): naive attempts, such as scaling the model score vectors or decreasing the amount of Gaussian noise added during diffusion sampling, are ineffective (Dhariwal & Nichol, 2021). Classifier guidance instead mixes a diffusion model's score estimate with the input gradient of the log probability of a

Figure 1: Classifier-free guidance on the malamute class for a 64x64 ImageNet diffusion model. Left to right: increasing amounts of classifier-free guidance, starting from non-guided samples on the left.

A short version of this paper appeared in the NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications: https://openreview.net/pdf?id=qw8AKxfYbI

1

CFG

2021.12

https://arxiv.org/pdf/2112.10741v1

GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models

Alex Nichol* Prafulla Dhariwal* Aditya Ramesh* Pranav Shyam Pamela Mishkin Bob McGrew Ilya Sutskever Mark Chen

Abstract

Diffusion models have recently been shown to generate high-quality synthetic images, especially when paired with a guidance technique to trade off diversity for fidelity. We explore diffusion models for the problem of text-conditional image synthesis and compare two different guidance strategies: CLIP guidance and classifier-free guidance. We find that the latter is preferred by human evaluators for both photorealism and caption similarity, and often produces photorealistic samples. Samples from a 3.5 billion parameter text-conditional diffusion model using classifier-free guidance are favored by human evaluators to those from DALL-E, even when the latter uses expensive CLIP reranking. Additionally, we find that our models can be fine-tuned to perform image inpainting, enabling powerful text-driven image editing. We train a smaller model on a filtered dataset and release the code and weights at https://github.com/openai/glide-text2im.

1. Introduction

Images, such as illustrations, paintings, and photographs, can often be easily described using text, but can require specialized skills and hours of labor to create. Therefore, a tool capable of generating realistic images from natural language can empower humans to create rich and diverse visual content with unprecedented ease. The ability to edit images using natural language further allows for iterative refinement and fine-grained control, both of which are critical for real world applications.

Recent text-conditional image models are capable of synthesizing images from free-form text prompts, and can compose unrelated objects in semantically plausible ways (Xu et al., 2017; Zhu et al., 2019; Tao et al., 2020; Ramesh et al., 2021; Zhang et al., 2021). However, they are not yet able to generate photorealistic images that capture all aspects of their corresponding text prompts.

On the other hand, unconditional image models can synthesize photorealistic images (Brock et al., 2018; Karras et al., 2019a;b; Razavi et al., 2019), sometimes with enough fidelity that humans can't distinguish them from real images (Zhou et al., 2019). Within this line of research, diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2020b) have emerged as a promising family of generative models, achieving state-of-the-art sample quality on a number of image generation benchmarks (Ho et al., 2020; Dhariwal & Nichol, 2021; Ho et al., 2021).

To achieve photorealism in the class-conditional setting, Dhariwal & Nichol (2021) augmented diffusion models with *classifier guidance*, a technique which allows diffusion models to condition on a classifier's labels. The classifier is first trained on noisy images, and during the diffusion sampling process, gradients from the classifier are used to guide the sample towards the label. Ho & Salimans (2021) achieved similar results without a separately trained classifier through the use of *classifier-free guidance*, a form of guidance that interpolates between predictions from a diffusion model with and without labels.

Motivated by the ability of guided diffusion models to generate photorealistic samples and the ability of text-to-image models to handle free-form prompts, we apply guided diffusion to the problem of text-conditional image synthesis. First, we train a 3.5 billion parameter diffusion model that uses a text encoder to condition on natural language descriptions. Next, we compare two techniques for guiding diffusion models towards text prompts: CLIP guidance and classifier-free guidance. Using human and automated evaluations, we find that classifier-free guidance yields higher-quality images.

We find that samples from our model generated with classifier-free guidance are both photorealistic and reflect a wide breadth of world knowledge. When evaluated by human judges, our samples are preferred to those from DALL-E (Ramesh et al., 2021) 87% of the time when evaluated for photorealism, and 69% of the time when evaluated for caption similarity.

*Equal contribution. Correspondence to alex@openai.com, prafulla@openai.com, aramesh@openai.com

GLIDE

2021.12

Timeline (4)

https://arxiv.org/pdf/2103.00020

https://arxiv.org/pdf/2204.06125

https://arxiv.org/pdf/2205.11487

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford *[1] Jong Wook Kim *[1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1] Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

Abstract

State-of-the-art computer vision systems are trained to predict a fixed set of predetermined object categories. This restricted form of supervision limits their generality and usability since additional labeled data is needed to specify any other visual concept. Learning directly from raw text about images is a promising alternative which leverages a much broader source of supervision. We demonstrate that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study the performance of this approach by benchmarking on over 30 different existing computer vision datasets, spanning tasks such as OCR, action recognition in videos, geo-localization, and many types of fine-grained object classification. The model transfers non-trivially to most tasks and is often competitive with a fully supervised baseline without the need for any dataset specific training. For instance, we match the accuracy of the original ResNet-50 on ImageNet zero-shot without needing to use any of the 1.28 million training examples it was trained on. We release our code and pre-trained model weights at https://github.com/OpenAI/CLIP.

1. Introduction and Motivating Work

Pre-training methods which learn directly from raw text have revolutionized NLP over the last few years (Dai & Le, 2015; Peters et al., 2018; Howard & Ruder, 2018; Radford et al., 2018; Devlin et al., 2018; Raffel et al., 2019).

Task-agnostic objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in compute, model capacity, and data, steadily improving capabilities. The development of "text-to-text" as a standardized input-output interface (McCann et al., 2018; Radford et al., 2019; Raffel et al., 2019) has enabled task-agnostic architectures to zero-shot transfer to downstream datasets removing the need for specialized output heads or dataset specific customization. Flagship systems like GPT-3 (Brown et al., 2020) are now competitive across many tasks with bespoke models while requiring little to no dataset specific training data.

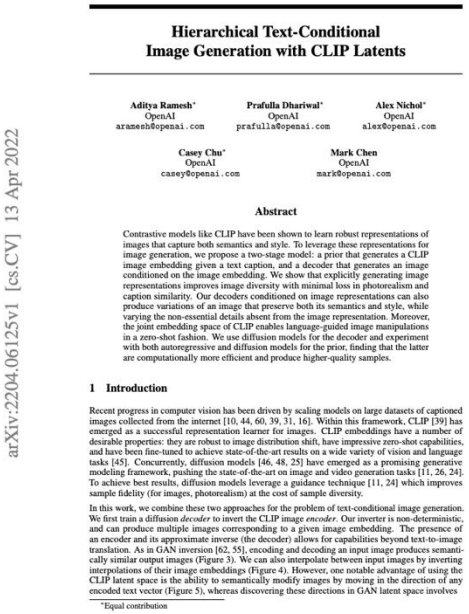
These results suggest that the aggregate supervision accessible to modern pre-training methods within web-scale collections of text surpasses that of high-quality crowd-labeled NLP datasets. However, in other fields such as computer vision it is still standard practice to pre-train models on crowd-labeled datasets such as ImageNet (Deng et al., 2009). Could scalable pre-training methods which learn directly from web text result in a similar breakthrough in computer vision? Prior work is encouraging.

Over 20 years ago Mori et al. (1999) explored improving content based image retrieval by training a model to predict the nouns and adjectives in text documents paired with images. Quattoni et al. (2007) demonstrated it was possible to learn more data efficient image representations via manifold learning in the weight space of classifiers trained to predict words in captions associated with images. Srivastava & Salakhutdinov (2012) explored deep representation learning by training multimodal Deep Boltzmann Machines on top of low-level image and text tag features. Joulin et al. (2016) modernized this line of work and demonstrated that CNNs trained to predict words in image captions learn useful image representations. They converted the title, description, and hashtag metadata of images in the YFCC100M dataset (Thomee et al., 2016) into a bag-of-words multi-label classification task and showed that pre-training AlexNet (Krizhevsky et al., 2012) to predict these labels learned representations which preformed similarly to ImageNet-based pre-training on transfer tasks. Li et al. (2017) then extended this approach to predicting phrase n-grams in addition to individual words and demonstrated the ability of their system to zero-shot transfer to other image

*Equal contribution [1]OpenAI, San Francisco, CA 94110, USA. Correspondence to: <{alec, jongwook}@openai.com>.

CLIP

2022.03



Hierarchical Text-Conditional Image Generation with CLIP Latents

Aditya Ramesh* OpenAI aramesh@openai.com

Prafulla Dhariwal* OpenAI prafulla@openai.com

Alex Nichol* OpenAI alex@openai.com

Casey Chu* OpenAI casey@openai.com

Mark Chen OpenAI mark@openai.com

Abstract

Contrastive models like CLIP have been shown to learn robust representations of images that capture both semantics and style. To leverage these representations for image generation, we propose a two-stage model: a prior that generates a CLIP image embedding given a text caption, and a decoder that generates an image conditioned on the image embedding. We show that explicitly generating image representations improves image diversity with minimal loss in photorealism and caption similarity. Our decoders conditioned on image representations can also produce variations of an image that preserve both its semantics and style, while varying the non-essential details absent from the image representation. Moreover, the joint embedding space of CLIP enables language-guided image manipulations in a zero-shot fashion. We use diffusion models for the decoder and experiment with both autoregressive and diffusion models for the prior, finding that the latter are computationally more efficient and produce higher-quality samples.

1 Introduction

Recent progress in computer vision has been driven by scaling models on large datasets of captioned images collected from the internet [10, 44, 60, 39, 31, 16]. Within this framework, CLIP [39] has emerged as a successful representation learner for images. CLIP embeddings have a number of desirable properties: they are robust to image distribution shift, have impressive zero-shot capabilities, and have been fine-tuned to achieve state-of-the-art results on a wide variety of vision and language tasks [45]. Concurrently, diffusion models [46, 48, 25] have emerged as a promising generative modeling framework, pushing the state-of-the-art on image and video generation tasks [11, 26, 24]. To achieve best results, diffusion models leverage a guidance technique [11, 24] which improves sample fidelity (for images, photorealism) at the cost of sample diversity.

In this work, we combine these two approaches for the problem of text-conditional image generation. We first train a diffusion *decoder* to invert the CLIP image *encoder*. Our inverter is non-deterministic, and can produce multiple images corresponding to a given image embedding. The presence of an encoder and its approximate inverse (the decoder) allows for capabilities beyond text-to-image translation. As in GAN inversion [62, 55], encoding and decoding an input image produces semantically similar output images (Figure 3). We can also interpolate between input images by inverting interpolations of their image embeddings (Figure 4). However, one notable advantage of using the CLIP latent space is the ability to semantically modify images by moving in the direction of any encoded text vector (Figure 5), whereas discovering these directions in GAN latent space involves

*Equal contribution

DALLE 2

2022.04

Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

Chitwan Saharia*, William Chan*, Saurabh Saxena†, Lala Li†, Jay Whang†, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho†, David J Fleet†, Mohammad Norouzi*

{saharia,williamchan,mnorouzi}@google.com
{srbs,lala,jwhang,jonathanho,davidfleet}@google.com

Google Research, Brain Team
Toronto, Ontario, Canada

Abstract

We present Imagen, a text-to-image diffusion model with an unprecedented degree of photorealism and a deep level of language understanding. Imagen builds on the power of large transformer language models in understanding text and hinges on the strength of diffusion models in high-fidelity image generation. Our key discovery is that generic large language models (e.g. T5), pretrained on text-only corpora, are surprisingly effective at encoding text for image synthesis: increasing the size of the language model in Imagen boosts both sample fidelity and image-text alignment much more than increasing the size of the image diffusion model. Imagen achieves a new state-of-the-art FID score of 7.27 on the COCO dataset, without ever training on COCO, and human raters find Imagen samples to be on par with the COCO data itself in image-text alignment. To assess text-to-image models in greater depth, we introduce DrawBench, a comprehensive and challenging benchmark for text-to-image models. With DrawBench, we compare Imagen with recent methods including VQ-GAN+CLIP, Latent Diffusion Models, GLIDE and DALL-E 2, and find that human raters prefer Imagen over other models in side-by-side comparisons, both in terms of sample quality and image-text alignment. See imagen.research.google for an overview of the results.

1 Introduction

Multimodal learning has come into prominence recently, with text-to-image synthesis [53, 12, 57] and image-text contrastive learning [49, 31, 74] at the forefront. These models have transformed the research community and captured widespread public attention with creative image generation [22, 54] and editing applications [21, 41, 34]. To pursue this research direction further, we introduce Imagen, a text-to-image diffusion model that combines the power of transformer language models (LMs) [15, 52] with high-fidelity diffusion models [28, 29, 16, 41] to deliver an unprecedented degree of photorealism and a deep level of language understanding in text-to-image synthesis. In contrast to prior work that uses only image-text data for model training [e.g., 53, 41], the key finding behind Imagen is that text embeddings from large LMs [52, 15], pretrained on text-only corpora, are remarkably effective for text-to-image synthesis. See Fig. 1 for select samples.

Imagen comprises a frozen T5-XXL [52] encoder to map input text into a sequence of embeddings and a 64×64 image diffusion model, followed by two super-resolution diffusion models for generating

*Equal contribution.
†Core contribution.

Imagen

2022.05

Google v.s. OpenAI

Transformer

T5

Imagen

Gemini

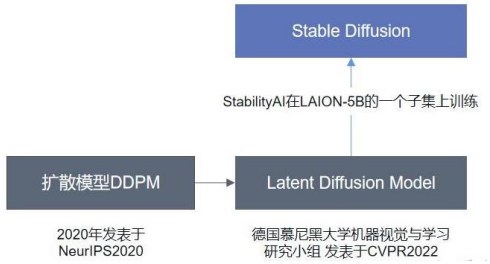
GPT

CLIP

DALLE

ChatGPT

Timeline (5)



https://arxiv.org/pdf/2112.10752v1

https://openreview.net/pdf?id=M3Y74vmsMcY

https://arxiv.org/pdf/2112.10752

https://github.com/CompVis/stable-diffusion

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]

[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML

https://github.com/CompVis/latent-diffusion

arXiv:2112.10752v1 [cs.CV] 20 Dec 2021

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve a new state of the art for image inpainting and highly competitive performance on various tasks, including unconditional image generation, semantic scene synthesis, and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [61,62]. In contrast, the promising results of GANs [3, 28, 30] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [77], which are built from a hierarchy of

*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [20] and PSNR are calculated on ImageNet-val. [11]; see also Tab. 1.

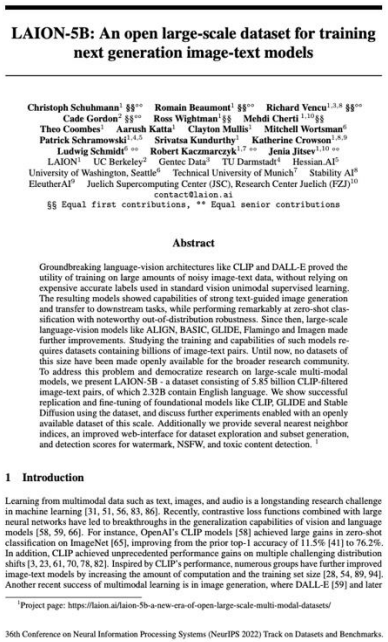
denoising autoencoders, have shown to achieve impressive results in image synthesis [27, 80] and beyond [6, 41, 44, 52], and define the state-of-the-art in class-conditional image synthesis [14, 28] and super-resolution [67]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [80] or stroke-based synthesis [48], in contrast to other types of generative models [17, 42, 61]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [62].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [15, 68]. Although the reweighted variational objective [27] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most

1

LDM v1

2021.12



LAION-5B: An open large-scale dataset for training next generation image-text models

Christoph Schuhmann[1] §§[∘] Romain Beaumont[1] §§[∘] Richard Vencu[1,3,8] §§[∘]
Cade Gordon[2] §§[∘] Ross Wightman[1] §§ Mehdi Cherti[1,10] §§
Theo Coombes[1] Aarush Katta[1] Clayton Mullis[1] Mitchell Wortsman[6]
Patrick Schramowski[1,4,5] Srivatsa Kundurthy[3] Katherine Crowson[1,8,9]
Ludwig Schmidt[6] ∘∘ Robert Kaczmarczyk[1,7] ∘∘ Jenia Jitsev[1,10] ∘∘
LAION[1] UC Berkeley[2] Gentec Data[3] TU Darmstadt[4] Hessian.AI[5]
University of Washington, Seattle[6] Technical University of Munich[7] Stability AI[8]
EleutherAI[9] Juelich Supercomputing Center (JSC), Research Center Juelich (FZJ)[10]
contact@laion.ai
§§ Equal first contributions, ** Equal senior contributions

Abstract

Groundbreaking language-vision architectures like CLIP and DALL-E proved the utility of training on large amounts of noisy image-text data, without relying on expensive accurate labels used in standard vision unimodal supervised learning. The resulting models showed capabilities of strong text-guided image generation and transfer to downstream tasks, while performing remarkably at zero-shot classification with noteworthy out-of-distribution robustness. Since then, large-scale language-vision models like ALIGN, BASIC, GLIDE, Flamingo and Imagen made further improvements. Studying the training and capabilities of such models requires datasets containing billions of image-text pairs. Until now, no datasets of this size have been made openly available for the broader research community. To address this problem and democratize research on large-scale multi-modal models, we present LAION-5B - a dataset consisting of 5.85 billion CLIP-filtered image-text pairs, of which 2.32B contain English language. We show successful replication and fine-tuning of foundational models like CLIP, GLIDE and Stable Diffusion using the dataset, and discuss further experiments enabled with an openly available dataset of this scale. Additionally we provide several nearest neighbor indices, an improved web-interface for dataset exploration and subset generation, and detection scores for watermark, NSFW, and toxic content detection. [1]

1 Introduction

Learning from multimodal data such as text, images, and audio is a longstanding research challenge in machine learning [31, 51, 56, 83, 86]. Recently, contrastive loss functions combined with large neural networks have led to breakthroughs in the generalization capabilities of vision and language models [58, 59, 66]. For instance, OpenAI's CLIP models [58] achieved large gains in zero-shot classification on ImageNet [65], improving from the prior top-1 accuracy of 11.5% [41] to 76.2%. In addition, CLIP achieved unprecedented performance gains on multiple challenging distribution shifts [3, 23, 61, 70, 78, 82]. Inspired by CLIP's performance, numerous groups have further improved image-text models by increasing the amount of computation and the training set size [28, 54, 89, 94]. Another recent success of multimodal learning is in image generation, where DALL-E [59] and later

[1]Project page: https://laion.ai/laion-5b-a-new-era-of-open-large-scale-multi-modal-datasets/

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

LAION-5B

2022.03

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach[1] * Andreas Blattmann[1] * Dominik Lorenz[1] Patrick Esser[2] Björn Ommer[1]
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [2]Runway ML

https://github.com/CompVis/latent-diffusion

Abstract

By decomposing the image formation process into a sequential application of denoising autoencoders, diffusion models (DMs) achieve state-of-the-art synthesis results on image data and beyond. Additionally, their formulation allows for a guiding mechanism to control the image generation process without retraining. However, since these models typically operate directly in pixel space, optimization of powerful DMs often consumes hundreds of GPU days and inference is expensive due to sequential evaluations. To enable DM training on limited computational resources while retaining their quality and flexibility, we apply them in the latent space of powerful pretrained autoencoders. In contrast to previous work, training diffusion models on such a representation allows for the first time to reach a near-optimal point between complexity reduction and detail preservation, greatly boosting visual fidelity. By introducing cross-attention layers into the model architecture, we turn diffusion models into powerful and flexible generators for general conditioning inputs such as text or bounding boxes and high-resolution synthesis becomes possible in a convolutional manner. Our latent diffusion models (LDMs) achieve new state-of-the-art scores for image inpainting and class-conditional image synthesis and highly competitive performance on various tasks, including text-to-image synthesis, unconditional image generation and super-resolution, while significantly reducing computational requirements compared to pixel-based DMs.

1. Introduction

Image synthesis is one of the computer vision fields with the most spectacular recent development, but also among those with the greatest computational demands. Especially high-resolution synthesis of complex, natural scenes is presently dominated by scaling up likelihood-based models, potentially containing billions of parameters in autoregressive (AR) transformers [60, 67]. In contrast, the promising results of GANs [3, 27, 40] have been revealed to be mostly confined to data with comparably limited variability as their adversarial learning procedure does not easily scale to modeling complex, multi-modal distributions. Recently, diffusion models [82], which are built from a hierarchy of denoising autoencoders, have shown to achieve impressive

*The first two authors contributed equally to this work.

Figure 1. Boosting the upper bound on achievable quality with less agressive downsampling. Since diffusion models offer excellent inductive biases for spatial data, we do not need the heavy spatial downsampling of related generative models in latent space, but can still greatly reduce the dimensionality of the data via suitable autoencoding models, see Sec. 3. Images are from the DIV2K [1] validation set, evaluated at 512^2 px. We denote the spatial downsampling factor by f. Reconstruction FIDs [29] and PSNR are calculated on ImageNet-val. [12]; see also Tab. 8.

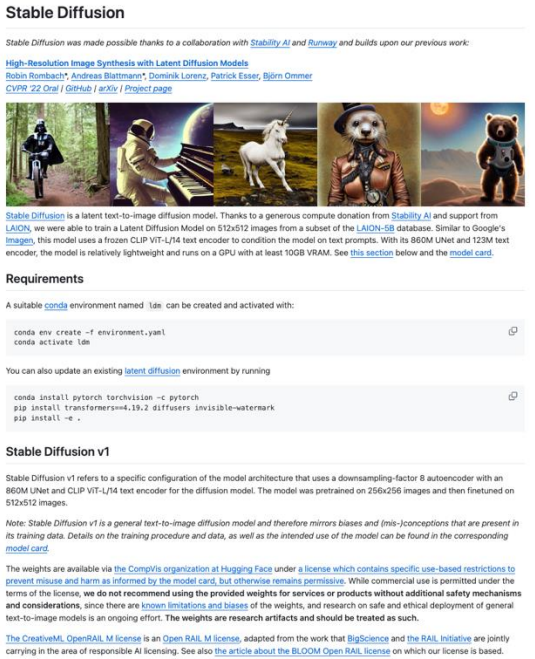
results in image synthesis [30, 85] and beyond [7, 45, 48, 57], and define the state-of-the-art in class-conditional image synthesis [15, 31] and super-resolution [72]. Moreover, even unconditional DMs can readily be applied to tasks such as inpainting and colorization [85] or stroke-based synthesis [53], in contrast to other types of generative models [19, 46, 69]. Being likelihood-based models, they do not exhibit mode-collapse and training instabilities as GANs and, by heavily exploiting parameter sharing, they can model highly complex distributions of natural images without involving billions of parameters as in AR models [67].

Democratizing High-Resolution Image Synthesis DMs belong to the class of likelihood-based models, whose mode-covering behavior makes them prone to spend excessive amounts of capacity (and thus compute resources) on modeling imperceptible details of the data [16, 73]. Although the reweighted variational objective [30] aims to address this by undersampling the initial denoising steps, DMs are still computationally demanding, since training and evaluating such a model requires repeated function evaluations (and gradient computations) in the high-dimensional space of RGB images. As an example, training the most powerful DMs often takes hundreds of GPU days (*e.g.* 150 - 1000 V100 days in [15]) and repeated evaluations on a noisy version of the input space render also inference expensive,

1

LDM v2

2022.04



Stable Diffusion

Stable Diffusion was made possible thanks to a collaboration with Stability AI and Runway and builds upon our previous work:

High-Resolution Image Synthesis with Latent Diffusion Models

Robin Rombach*, Andreas Blattmann*, Dominik Lorenz, Patrick Esser, Björn Ommer

CVPR '22 Oral | GitHub | arXiv | Project page

Stable Diffusion is a latent text-to-image diffusion model. Thanks to a generous compute donation from Stability AI and support from LAION, we were able to train a Latent Diffusion Model on 512x512 images from a subset of the LAION-5B database. Similar to Google's Imagen, this model uses a frozen CLIP ViT-L/14 text encoder to condition the model on text prompts. With its 860M UNet and 123M text encoder, the model is relatively lightweight and runs on a GPU with at least 10GB VRAM. See this section below and the model card.

Requirements

A suitable conda environment named `ldm` can be created and activated with:

```
conda env create -f environment.yaml
conda activate ldm
```

You can also update an existing latent diffusion environment by running

```
conda install pytorch torchvision -c pytorch
pip install transformers==4.19.2 diffusers invisible-watermark
pip install -e .
```

Stable Diffusion v1

Stable Diffusion v1 refers to a specific configuration of the model architecture that uses a downsampling-factor 8 autoencoder with an 860M UNet and CLIP ViT-L/14 text encoder for the diffusion model. The model was pretrained on 256x256 images and then finetuned on 512x512 images.

Note: Stable Diffusion v1 is a general text-to-image diffusion model and therefore mirrors biases and (mis-)conceptions that are present in its training data. Details on the training procedure and data, as well as the intended use of the model can be found in the corresponding model card.

The weights are available via the CompVis organization at Hugging Face under a license which contains specific use-based restrictions to prevent misuse and harm as informed by the model card, but otherwise remains permissive. While commercial use is permitted under the terms of the license, **we do not recommend using the provided weights for services or products without additional safety mechanisms and considerations**, since there are known limitations and biases of the weights, and research on safe and ethical deployment of general text-to-image models is an ongoing effort. **The weights are research artifacts and should be treated as such.**

The CreativeML OpenRAIL M license is an Open RAIL M license, adapted from the work that BigScience and the RAIL Initiative are jointly carrying in the area of responsible AI licensing. See also the article about the BLOOM Open RAIL license on which our license is based.

Stable Diffusion V1

2022.08

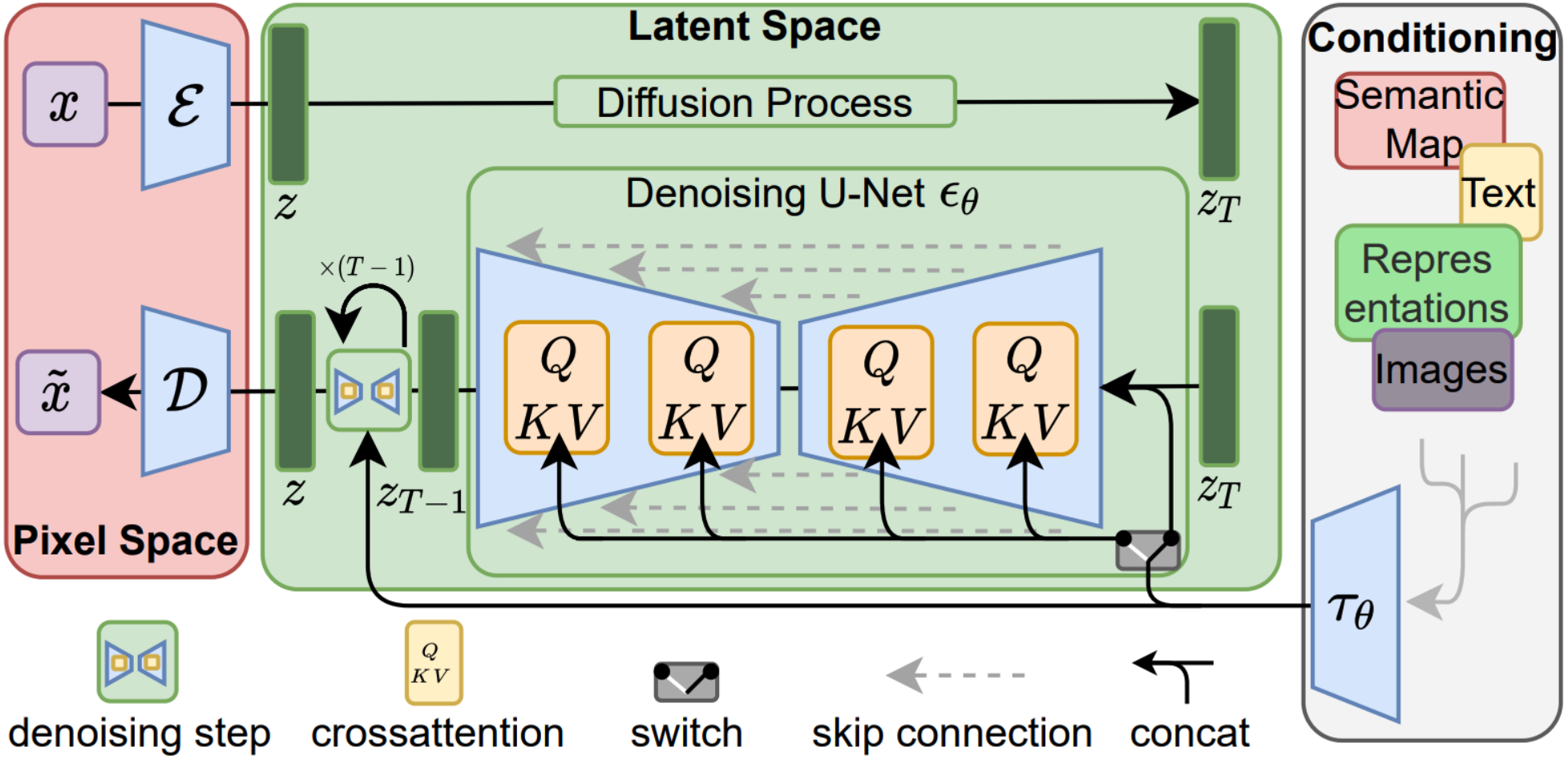
V2 2022.11

Latent Diffusion Model

CVPR ’22

Patrick Esser

Robin Rombach



VQ-VAE

- Vector Quantized Variational AutoEncoder

$$z_q(x) = e_k, \quad \text{where} \quad k = \text{argmin}_j \|z_e(x) - e_j\|_2$$

- sg (stop gradient) $\quad L_{\text{recon}} = ||x - \text{decoder}(z_e(x) + sg(z_q(x) - z_e(x)))||_2^2$

```
L = x - decoder(z_e + (z_q - z_e).detach())
```

$$L_e = ||z_e(x) - z_q(x)||_2^2 \quad \rightarrow \quad L_e = ||sg(z_e(x)) - z_q(x)||_2^2 + \beta||z_e(x) - sg(z_q(x))||_2^2$$

$$\Rightarrow \quad L = L_{\text{recon}} + \alpha L_e$$

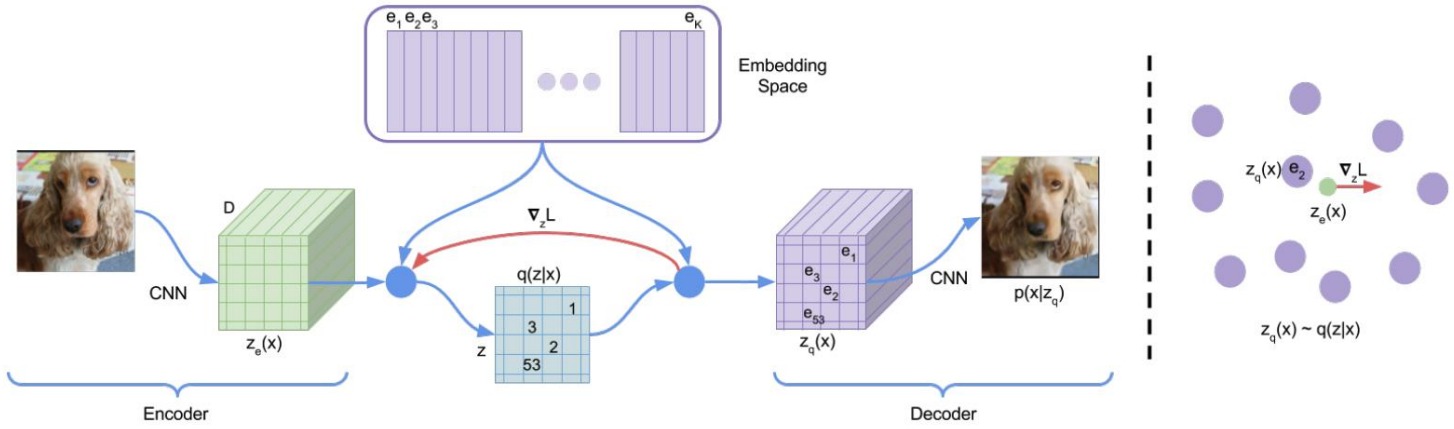


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

https://arxiv.org/pdf/1711.00937

Neural Discrete Representation Learning

Aaron van den Oord
DeepMind
avdnoord@google.com

Oriol Vinyals
DeepMind
vinyals@google.com

Koray Kavukcuoglu
DeepMind
korayk@google.com

Abstract

Learning useful representations without supervision remains a key challenge in machine learning. In this paper, we propose a simple yet powerful generative model that learns such discrete representations. Our model, the Vector Quantised-Variational AutoEncoder (VQ-VAE), differs from VAEs in two key ways: the encoder network outputs discrete, rather than continuous, codes; and the prior is learnt rather than static. In order to learn a discrete latent representation, we incorporate ideas from vector quantisation (VQ). Using the VQ method allows the model to circumvent issues of "posterior collapse" — where the latents are ignored when they are paired with a powerful autoregressive decoder — typically observed in the VAE framework. Pairing these representations with an autoregressive prior, the model can generate high quality images, videos, and speech as well as doing high quality speaker conversion and unsupervised learning of phonemes, providing further evidence of the utility of the learnt representations.

1 Introduction

Recent advances in generative modelling of images [38, 12, 13, 22, 10], audio [37, 26] and videos [20, 11] have yielded impressive samples and applications [24, 18]. At the same time, challenging tasks such as few-shot learning [34], domain adaptation [17], or reinforcement learning [35] heavily rely on learnt representations from raw data, but the usefulness of generic representations trained in an unsupervised fashion is still far from being the dominant approach.

Maximum likelihood and reconstruction error are two common objectives used to train unsupervised models in the pixel domain, however their usefulness depends on the particular application the features are used in. Our goal is to achieve a model that conserves the important features of the data in its latent space while optimising for maximum likelihood. As the work in [7] suggests, the best generative models (as measured by log-likelihood) will be those without latents but a powerful decoder (such as PixelCNN). However, in this paper, we argue for learning discrete and useful latent variables, which we demonstrate on a variety of domains.

Learning representations with continuous features have been the focus of many previous work [16, 39, 6, 9] however we concentrate on discrete representations [27, 33, 8, 28] which are potentially a more natural fit for many of the modalities we are interested in. Language is inherently discrete, similarly speech is typically represented as a sequence of symbols. Images can often be described concisely by language [40]. Furthermore, discrete representations are a natural fit for complex reasoning, planning and predictive learning (e.g., if it rains, I will use an umbrella). While using discrete latent variables in deep learning has proven challenging, powerful autoregressive models have been developed for modelling distributions over discrete variables [37].

In our work, we introduce a new family of generative models succesfully combining the variational autoencoder (VAE) framework with discrete latent representations through a novel parameterisation of the posterior distribution of (discrete) latents given an observation. Our model, which relies on vector quantization (VQ), is simple to train, does not suffer from large variance, and avoids the

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Original image

Image Encoder

Generate training examples with different amounts of noise added to their compressed/latent version

Compressed image (latent)

Latent + noise sample 1 at noise amount **1**

Latent + noise sample 2 at noise amount **2**

Latent + noise sample 3 at noise amount **3**

Image Generation by Reverse Diffusion (Denoising)

Processed Image Information

UNet Step **50**

UNet Step **2**

UNet Step **1**

Complete noise

Image Information Creator

Image Decoder

Generated image

Condition

1. Cross Attention in UNet

https://arxiv.org/pdf/2112.10752v1

To pre-process y from various modalities (such as language prompts) we introduce a domain specific encoder τ_θ that projects y to an intermediate representation $\tau_\theta(y) \in \mathbb{R}^{M \times d_\tau}$, which is then mapped to the intermediate layers of the UNet via a cross-attention layer implementing $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) \cdot V$, with

$$Q = W_Q^{(i)} \cdot \varphi_i(z_t),\ K = W_K^{(i)} \cdot \tau_\theta(y),\ V = W_V^{(i)} \cdot \tau_\theta(y).$$

Here, $\varphi_i(z_t) \in \mathbb{R}^{N \times d_\epsilon^i}$ denotes a (flattened) intermediate representation of the UNet implementing ϵ_θ and $W_V^{(i)} \in \mathbb{R}^{d \times d_\epsilon^i}$, $W_Q^{(i)} \in \mathbb{R}^{d \times d_\tau}$ & $W_K^{(i)} \in \mathbb{R}^{d \times d_\tau}$ are learnable projection matrices [32, 91]. See Fig. 3 for a visual depiction.

Based on image-conditioning pairs, we then learn the conditional LDM via

$$L_{LDM} := \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right], \quad (3)$$

where both τ_θ and ϵ_θ are jointly optimized via Eq. 3. This conditioning mechanism is flexible as τ_θ can be parameterized with domain-specific experts, *e.g.* (unmasked) transformers [91] when y are text prompts (see Sec. 4.3.1)

2. Different conditioning method

https://arxiv.org/pdf/2212.09748

Scalable Diffusion Models with Transformers

William Peebles*
UC Berkeley

Saining Xie
New York University

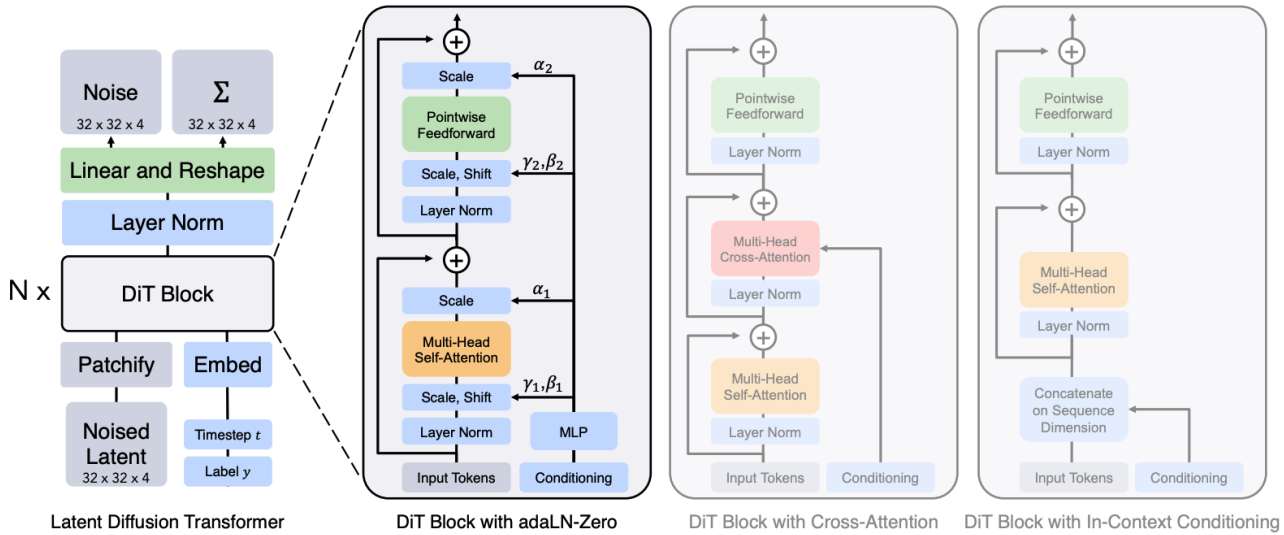
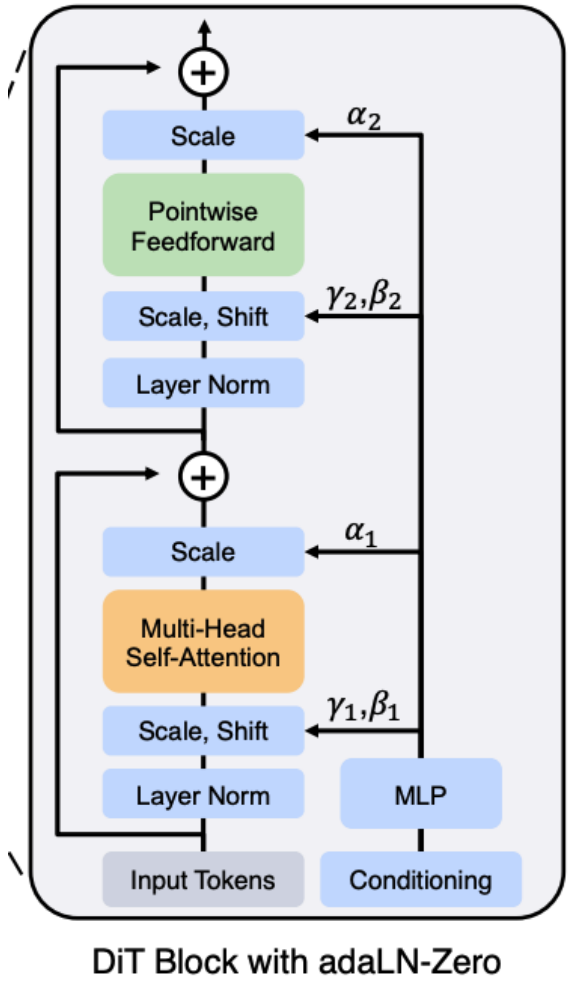


Figure 3. **The Diffusion Transformer (DiT) architecture.** *Left:* We train conditional latent DiT models. The input latent is decomposed into patches and processed by several DiT blocks. *Right:* Details of our DiT blocks. We experiment with variants of standard transformer blocks that incorporate conditioning via adaptive layer norm, cross-attention and extra input tokens. Adaptive layer norm works best.

Condition



DiT Block with adaLN-Zero

- Adaptive Layer Normalization

$$y = \frac{x - \mathbb{E}[x]}{\sqrt{\mathbf{Var}[x] + \epsilon}} \cdot \boldsymbol{\gamma}(t,c) + \boldsymbol{\beta}(t,c)$$

- AdaLN-Zero initialize $\boldsymbol{\alpha} = 0$

$$\boldsymbol{\alpha}(y) \odot f(x) + x$$

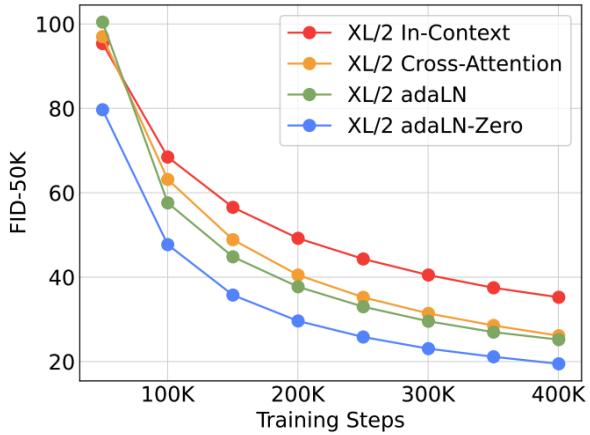
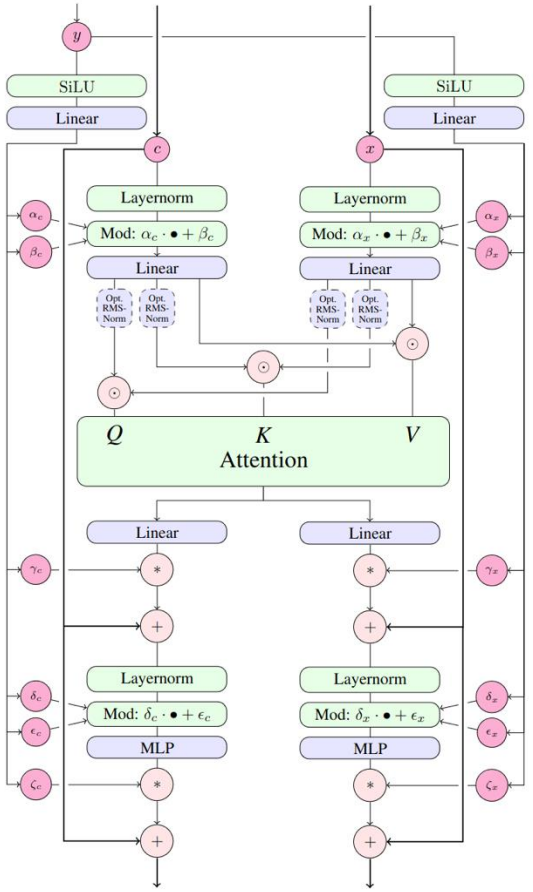


Figure 5. **Comparing different conditioning strategies.** adaLN-Zero outperforms cross-attention and in-context conditioning at all stages of training.

3. MM-DiT in Stable Diffusion v3



(b) One *MM-DiT* block

https://arxiv.org/abs/2403.03206

Timeline (6) ★★★★★

Finetune

Control

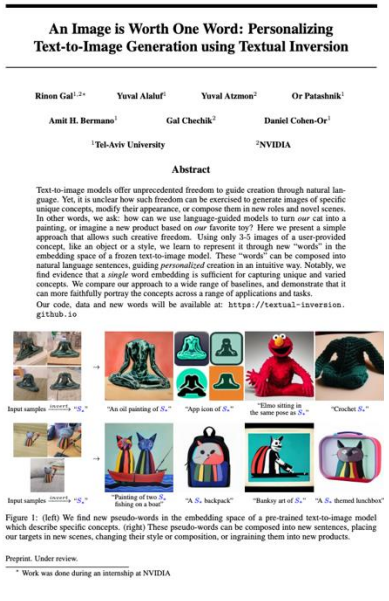
https://arxiv.org/pdf/2208.01618v1

https://arxiv.org/pdf/2208.12242v1

https://arxiv.org/pdf/2301.07093v1

https://arxiv.org/pdf/2302.05543

https://arxiv.org/pdf/2302.08453v1

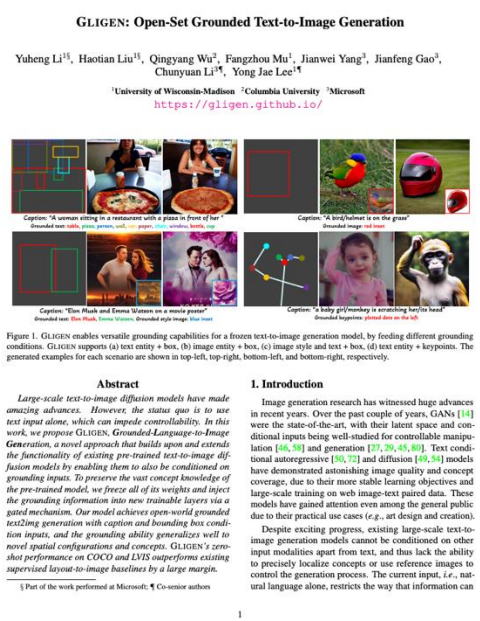


Textual Inversion

2022.08

DreamBooth

2022.08



GLIGEN

2023.01

ControlNet

2023.02

T2I-Adapter

2023.02

IC-Light

《Scaling In-the-Wild Training for Diffusion-Based Illumination Harmonization and Editing by Imposing Consistent Light Transport》

- **Author:** Lvmin Zhang　苏大本科 => Stanford 博
- **Task**: Illumination harmonization and editing
- **Difficulty**: Preserving the underlying image details and maintaining intrinsic properties unchanged.
- **Goal**: Precise illumination manipulation


Adding conditional control to text-to-image diffusion models
L Zhang, A Rao, M Agrawala
Proceedings of the IEEE/CVF International Conference on …, 2023 · openaccess.thecvf.com

Abstract
We present ControlNet, a neural network architecture to add spatial conditioning controls to large, pretrained text-to-image diffusion models. ControlNet locks the production-ready large diffusion models, and reuses their deep and robust encoding layers pretrained with billions of images as a strong backbone to learn a diverse set of conditional controls. The neural architecture is connected with" zero convolutions"(zero-initialized convolution layers) that progressively grow the parameters from zero and ensure that no harmful noise

展开

☆ 保存　99 引用　被引用次数：3036　相关文章　所有 6 个版本　»

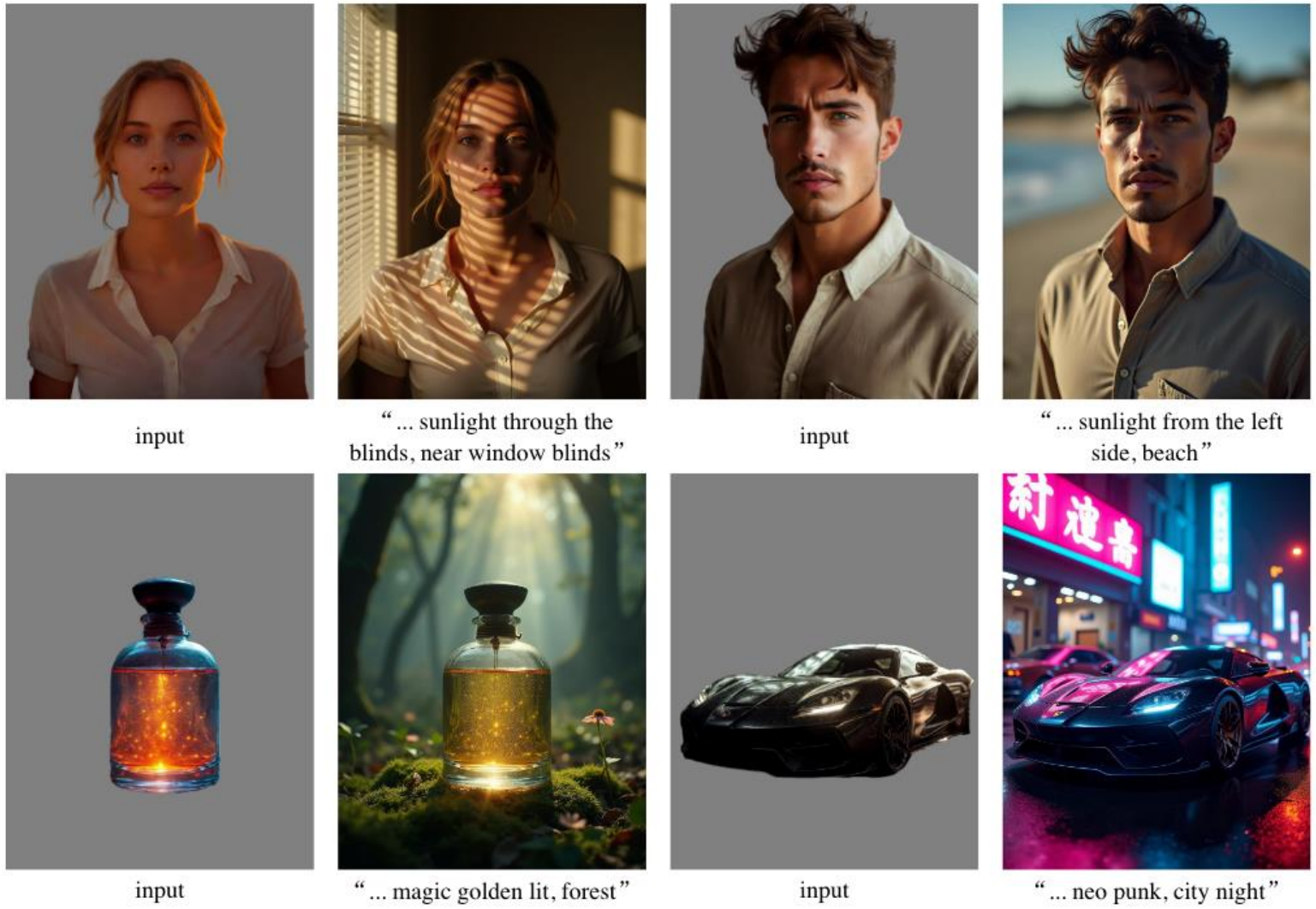
- **Method**: Impose Consistent Light (IC-Light) transport during training (rooted in physical principle)
- **Results**: Stable and scalable illumination learning, scale up the training of diffusion-based illumination editing models to large data quantities, reduces uncertainties and mitigates artifacts…

Illumination harmonization and editing

- **Typical Use Case**: Users give an object image and illumination description, and our method generates corresponding object appearances and backgrounds.
- **Challenge**:

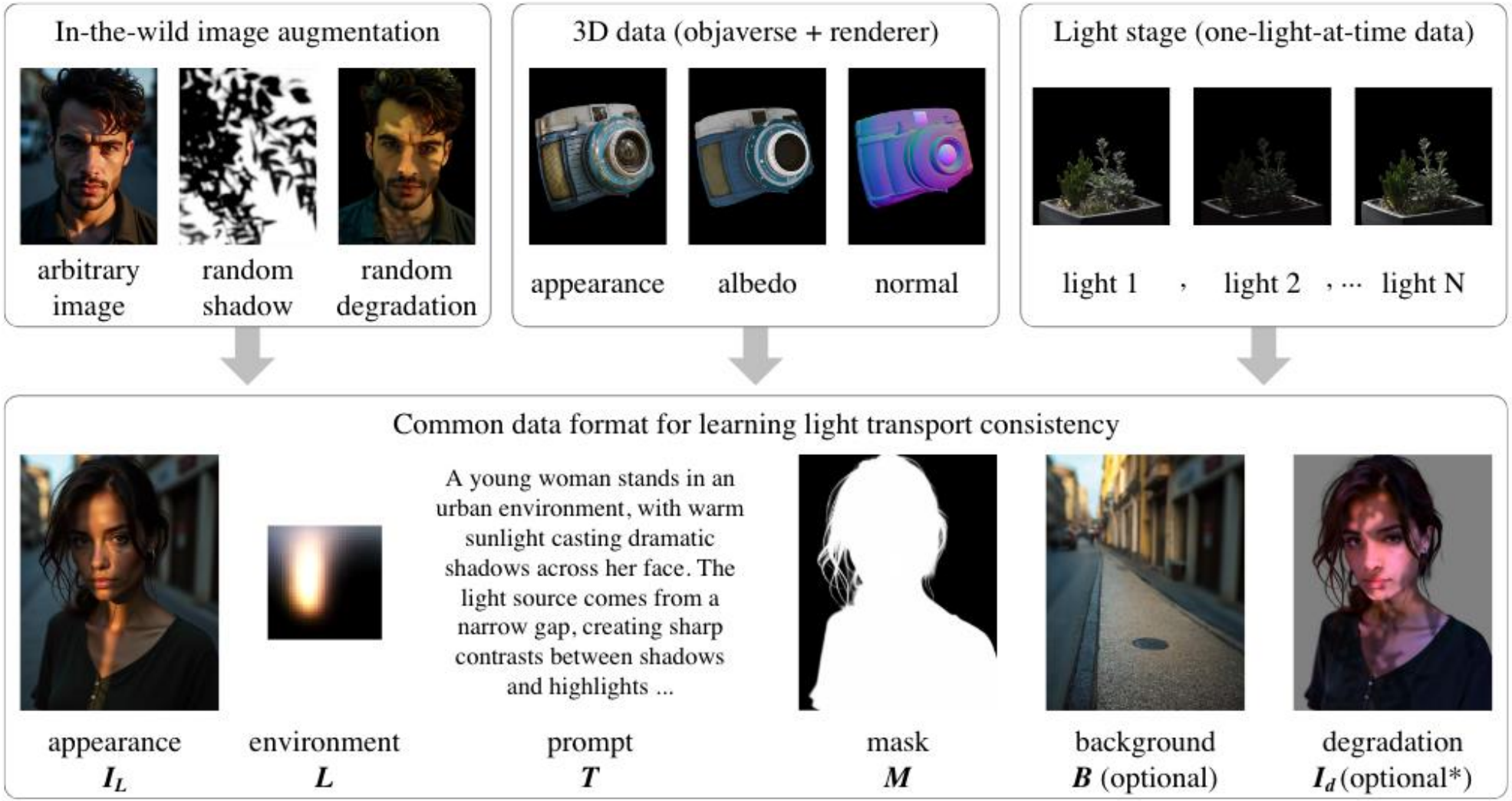
① 加上的东西 Line49

② 本身有的东西 Line86



Dataset formation

用了很多别人训好的功能特异的模型来构造数据集



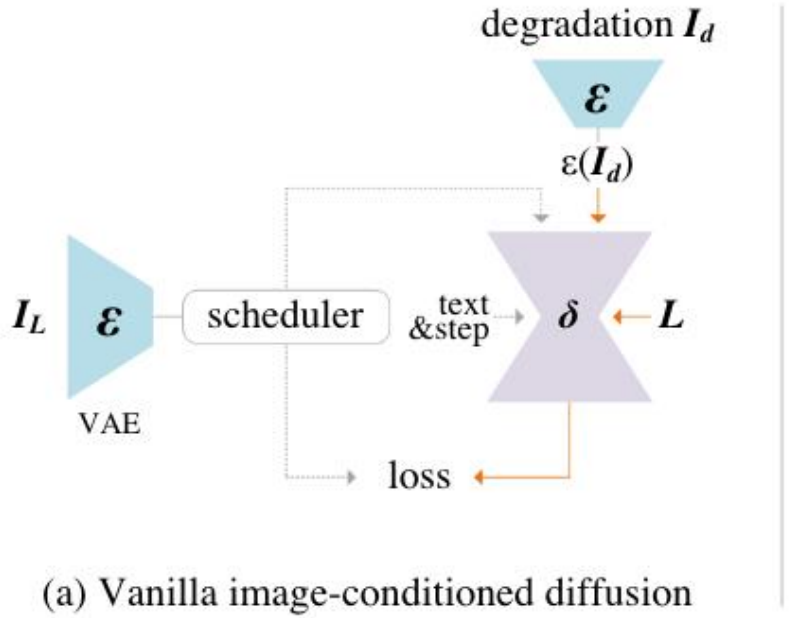
Impose Consistent Light

(a) The vanilla objective will often lead to random model behaviors, e.g., color mismatch, incorrect details, etc.

$$\mathcal{L}_{\text{vanilla}} = \|\epsilon - \boldsymbol{\delta}(\boldsymbol{\varepsilon}(\boldsymbol{I_L})_t, t, \boldsymbol{L}, \boldsymbol{\varepsilon}(\boldsymbol{I_d}))\|_2^2$$

(a) In computational photography, light transport theory demonstrates that, considering arbitrary appearance I_L^* and the correlated environment illumination L, a matrix T always exists so that

$$\boldsymbol{I_L^* = TL}$$



(a) Vanilla image-conditioned diffusion

Because of this linearity, light transport explains appearance merging that

$$\boldsymbol{I^*_{L_1+L_2} = T(L_1 + L_2) = I^*_{L_1} + I^*_{L_2}}$$

where L_1, L_2 are two arbitrary environment illumination maps.

This intuitively shows that the mixture of an object's appearances under separate illuminations (e.g., L_1, L_2) is equivalent to the appearance under merged illumination (e.g., $I^*_{L_1+L_2}$).

Impose Consistent Light

$$I^*_{L_1+L_2} = T(L_1 + L_2) = I^*_{L_1} + I^*_{L_2}$$

This intuitively shows that the mixture of an object's appearances under separate illuminations (e.g., L_1, L_2) is equivalent to the appearance under merged illumination (e.g., $I^*_{L_1+L_2}$).

怎么把这个一致性约束加到 Diffusion 的损失函数里面去?

Scene with illumination A
(real photo)

Scene with illumination B
(real photo)

Scene with illumination C
(real photo)

Blending of real photo A and B
(computed image)

Altered Blending with color tone
(computed image)

Figure 1: Examples for "the linear blending of an object's appearances under different illumination conditions is consistent with its appearance under mixed illumination". Images from OToole (2016).

Impose Consistent Light

$$I^*_{L_1+L_2} = T(L_1 + L_2) = I^*_{L_1} + I^*_{L_2}$$

* 表示 images in raw high-dynamic range

1. Image Space: Image => Predicted Noise 图像的线性关系可以转变为噪声的线性关系

"Clean image + Noise = Noisy Image" $\Longrightarrow$ "Estimated Clean image = Noisy Image – Predicted Noise"

A simple k-diffusion epsilon target at sigma-space step σ_t, estimated noise ϵ_L (conditioned on L), and noisy image I_{σ_t} , the estimated clean appearance $\hat{I_L} = (I_{\sigma_t} - \epsilon_L)/\sigma_t$

$$\epsilon_{L_1+L_2} = \epsilon_{L_1} + \epsilon_{L_2} \quad \Longrightarrow \quad \|\epsilon_{L_1+L_2} - (\epsilon_{L_1} + \epsilon_{L_2})\|_2^2$$

2. Latent Space: Linear summation relation => MLP mapping ϕ

$$\mathcal{L}_{\text{consistency}} = \|M \odot (\epsilon_{L_1+L_2} - \phi(\epsilon_{L_1}, \epsilon_{L_2}))\|_2^2$$

Intuition: Assume mapping f: latent space → image space

$$f(\epsilon_{L_1+L_2}) = f(\epsilon_{L_1}) + f(\epsilon_{L_2}) \Rightarrow \epsilon_{L_1+L_2} = f^{-1}\left(f(\epsilon_{L_1}) + f(\epsilon_{L_2})\right) \Rightarrow \epsilon_{L_1+L_2} = \phi(\epsilon_{L_1}, \epsilon_{L_2})$$

3. Implementation of environment illumination maps

Impose Consistent Light

$$\mathcal{L}_{\text{consistency}} = \|M \odot (\epsilon - \phi(\delta(\varepsilon(I_{L_1})_t, t, L_1, \varepsilon(I_d))), \delta(\varepsilon(I_{L_2})_t, t, L_2, \varepsilon(I_d)))\|_2^2$$

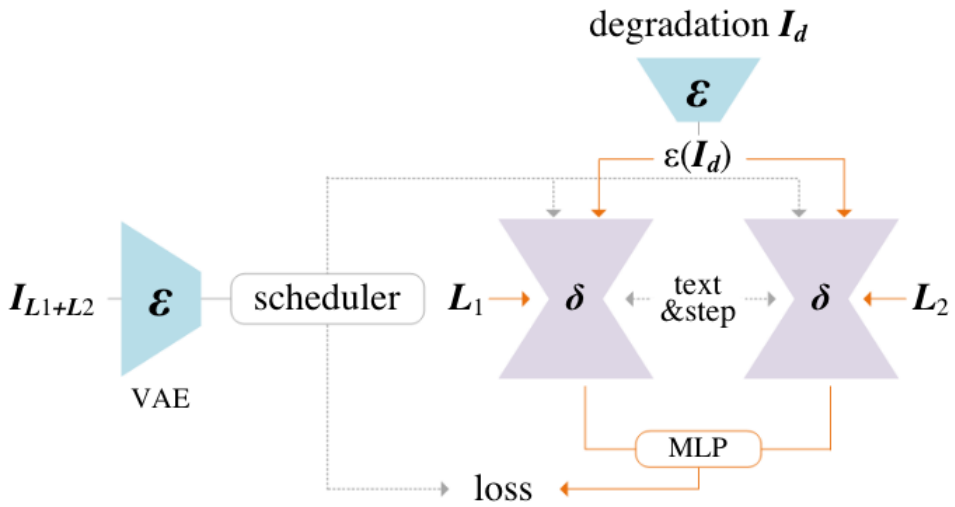
Joint learning objective The final learning objective can be written as

$$\mathcal{L} = \lambda_{\text{vanilla}}\mathcal{L}_{\text{vanilla}} + \lambda_{\text{consistency}}\mathcal{L}_{\text{consistency}} ,$$

where $\mathcal{L}$ is the merged objective, and we use $\lambda_{\text{vanilla}} = 1.0, \lambda_{\text{consistency}} = 0.1$ as default weights.

3. Implementation of environment illumination maps

满足 $L = L_1 + L_2$



(b) Learning light transport consistency

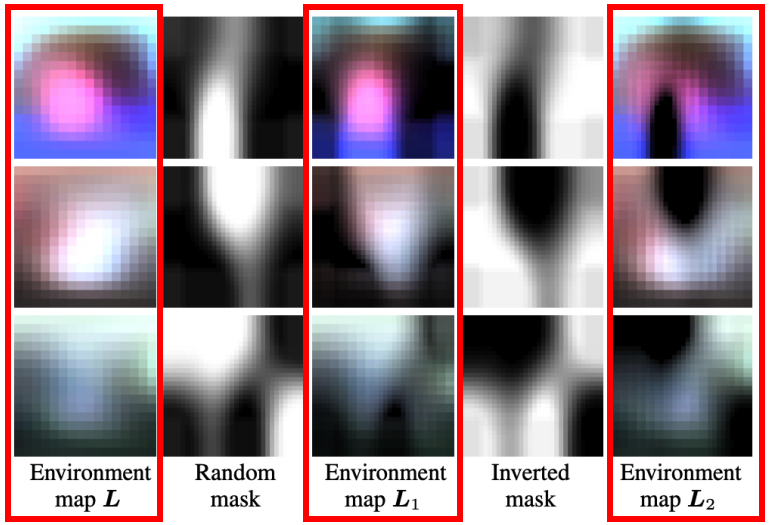


Figure 4: Examples of decomposed environment maps. We present examples to use random masks to decompose environment map L into L_1 and L_2. Note that $L = L_1 + L_2$. A typical full environment map is usually of ratio 2:1, with size 64×32 when convoluted. We use the front half (facing the image) of the convoluted environment map, which is 32×32. Using the front half makes normal-based environment extraction easier (since the image-space normals often do not have any pixels facing to the back half). Besides, the back halfs of environment maps from DiffusionLight Phongthawee et al. (2023) are usually not strictly correlated to image contents and can be excluded.

Experiments

- **Metric**:

PSNR：基于像素差异，简单

SSIM：通过结构信息评估图像相似度

LPIPS：基于深度学习的感知评价

- **Inference**: Condition on (Image ⊙ Foreground Mask), Illumination maps + Text Prompt

Table 1: Quantitative tests of ablative architectures and alternative methods.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
SwitchLight	18.45	0.7024	0.3245
DiLightNet	21.78	0.8013	0.1721
w/o LTC	20.32	0.7542	0.1927
w/o aug. data	23.95	0.8723	0.1115
w/o 3d data	22.10	0.8041	0.1298
w/o light stage	23.70	0.8501	0.1077
Ours	23.72	0.8513	0.1025

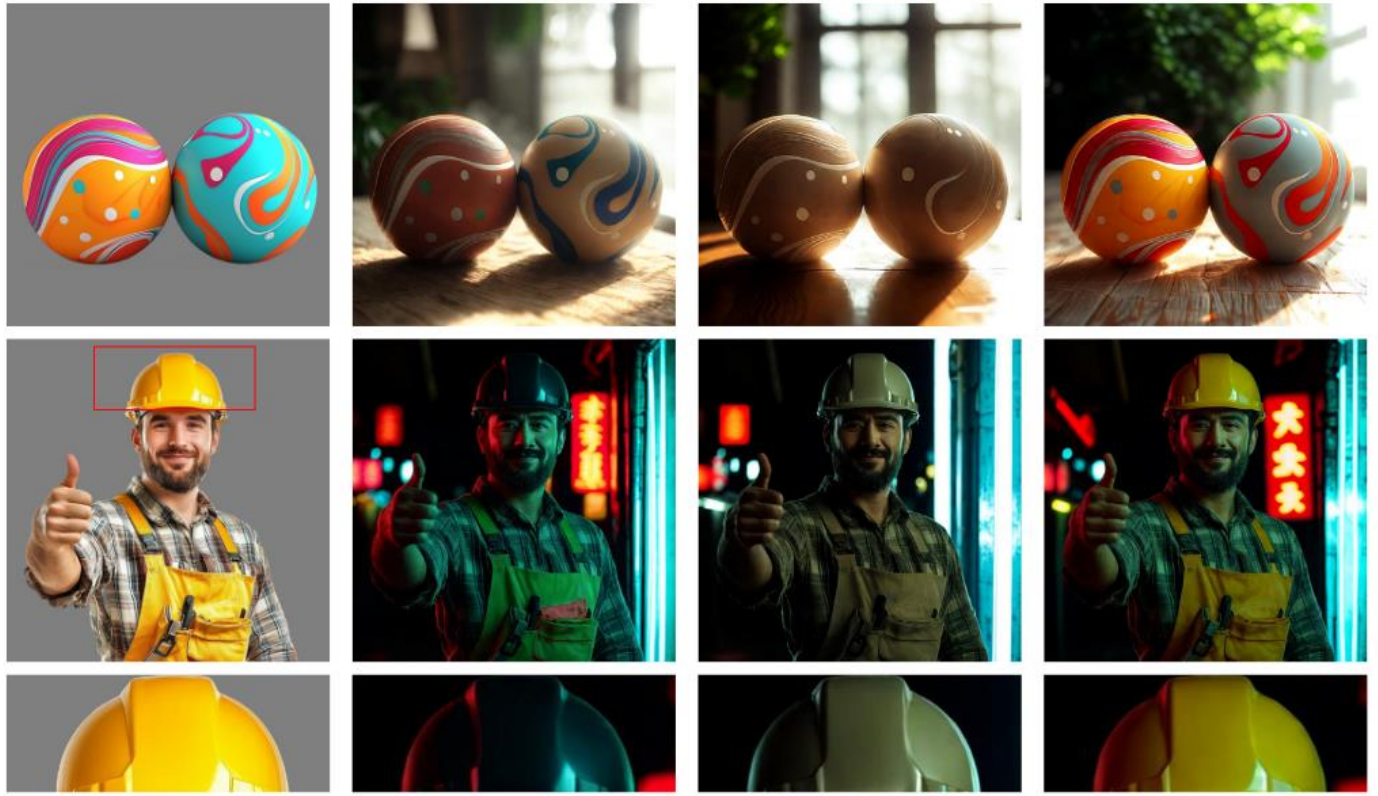
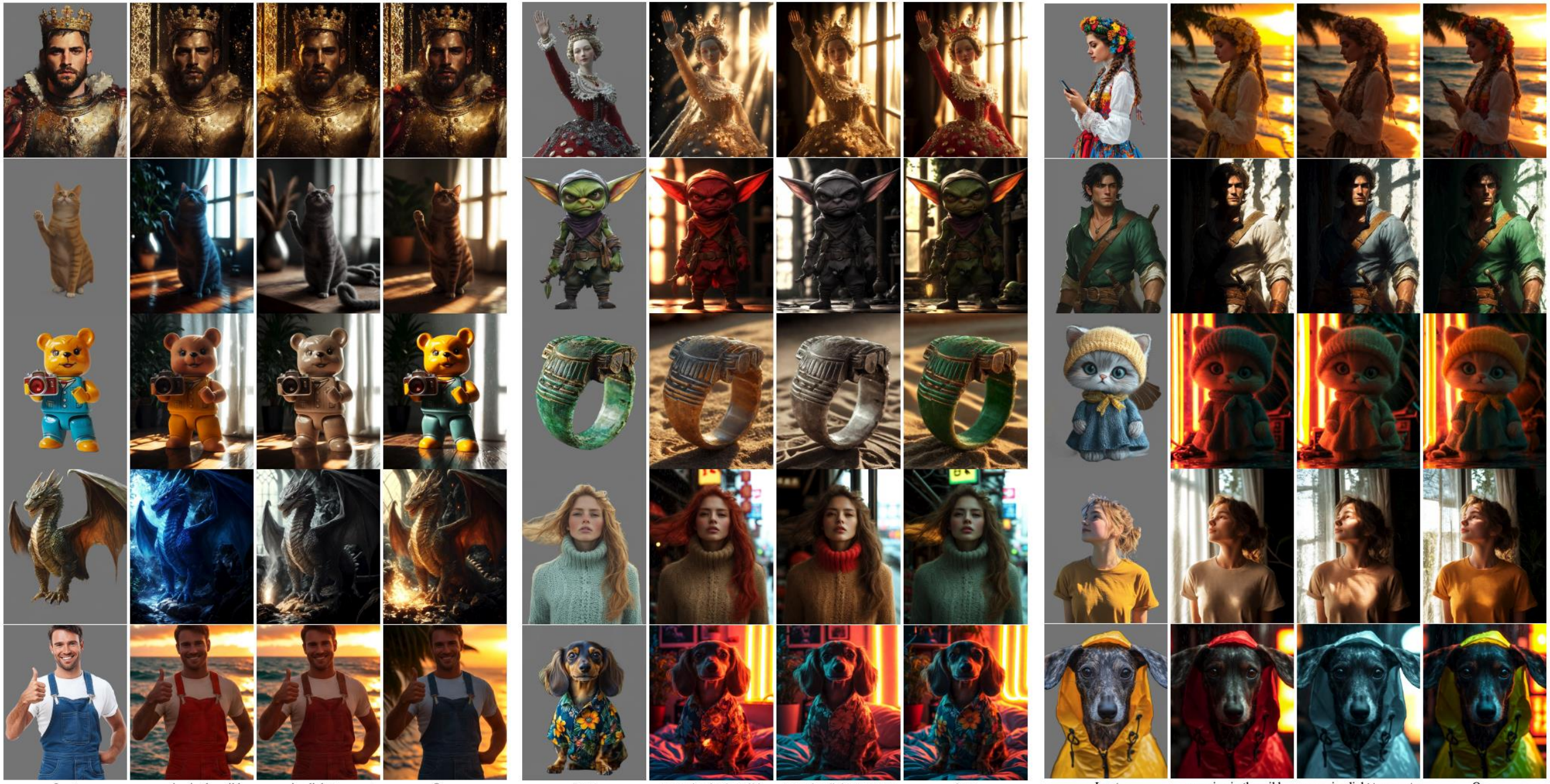


Figure 4: **Ablative Study.** We present results by removing the light transport consistency or in the wild data. More results are in the supplementary material. Results in this figure are from Stable Diffusion 1.5 version of our model. Prompts are "toy in room, studio lighting", and "a handsome man, neon city".


Input
removing in-the-wild data
removing light transport consistency
Ours
Input
removing in-the-wild data
removing light transport consistency
Ours
Input
removing in-the-wild data
removing light transport consistency
Ours

Additional Application

- **Background-conditioned Model**

① **Training**:

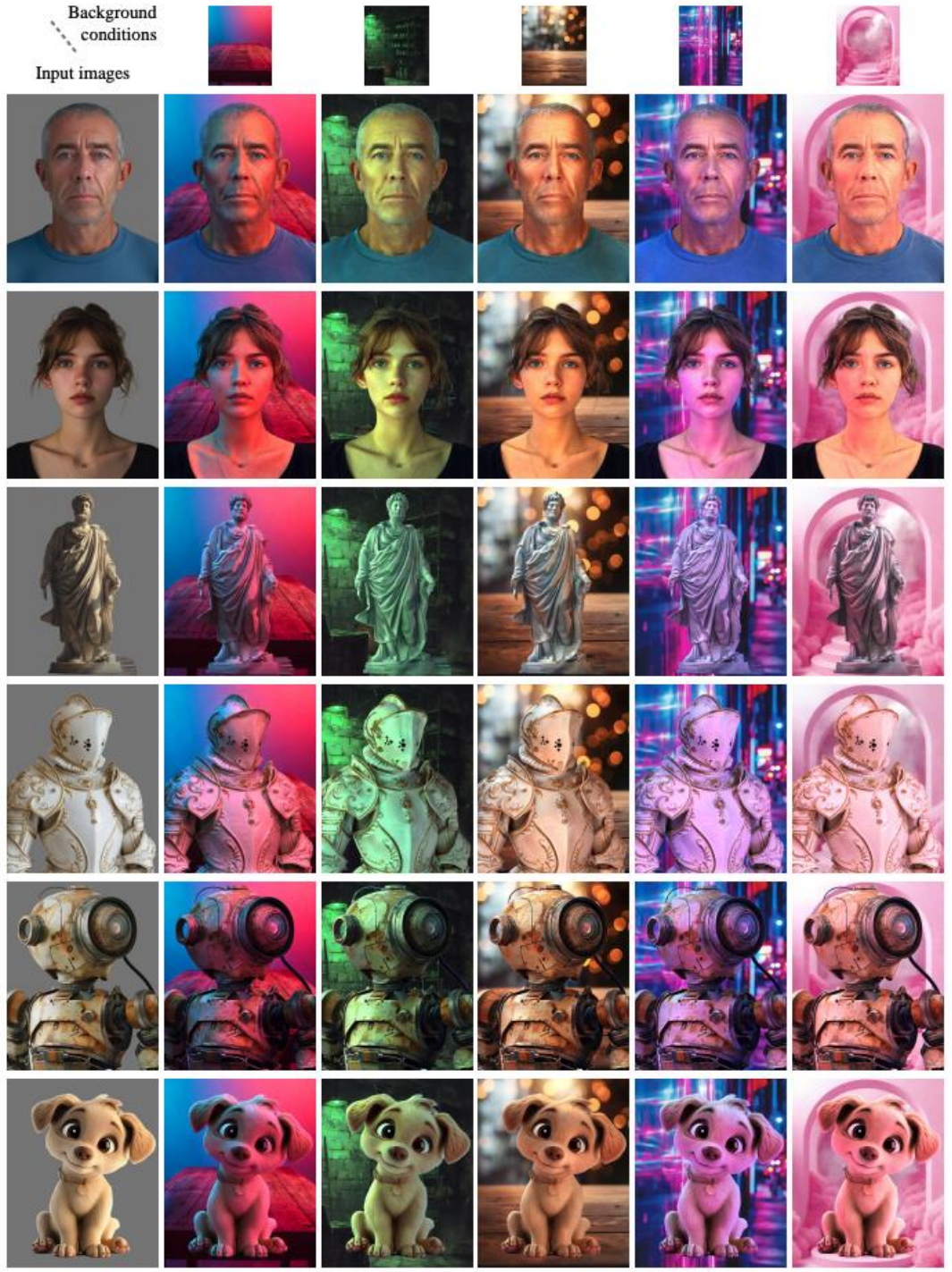
"Besides, to train background-conditioned model, we concatenate B to I_d (and fill the extra channel with all zeros if some part of the dataset do not have backgrounds)."

② **Inference**:

(Image ⊙ Foreground Mask), Background conditions

- Alternative base diffusion models

  SD 1.5, SDXL, Flux
- Normal Estimation (Omitted)



Flux-based IC-Light Model with 16ch VAE and native high resolution. See also https://github.com/lllyasviel/IC-Light/discussions/98

Prompt

vintage photograph of a woman. sunshine from window.

Normal case

Initial Latent

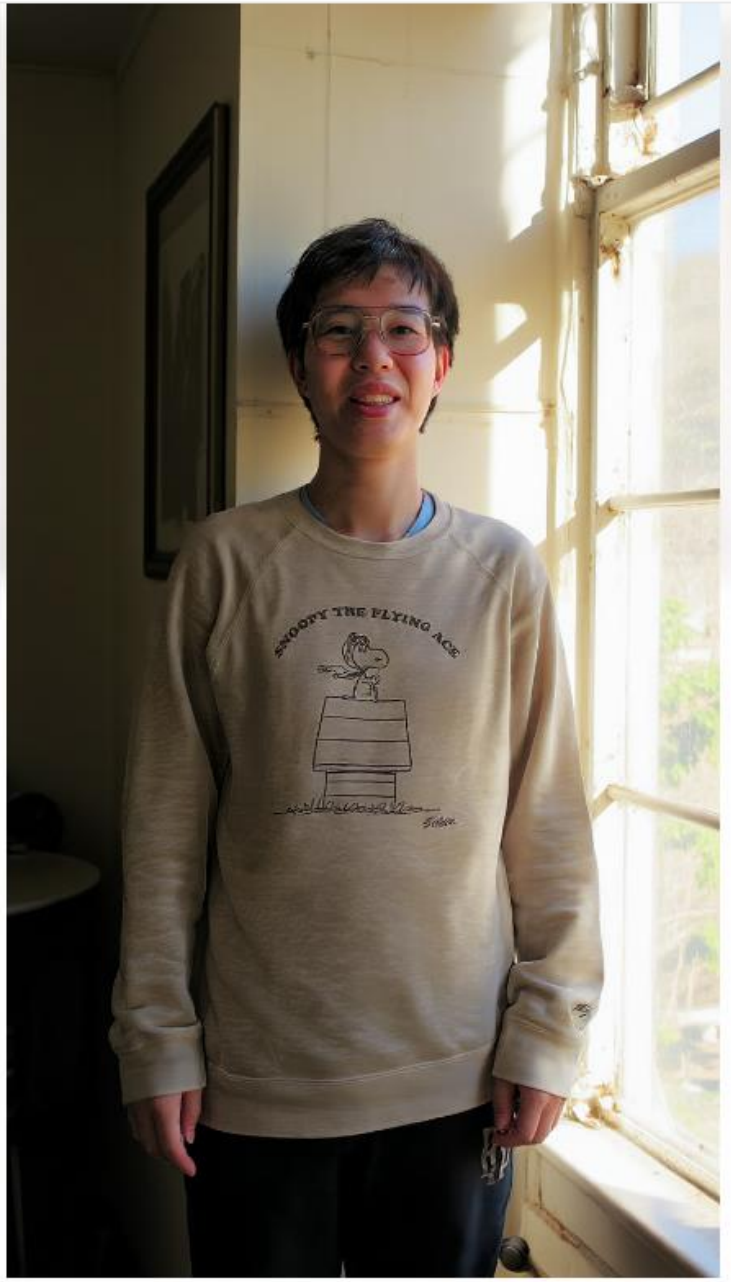
None | Left Light | Right Light | Top Light | Bottom Light

Prefix Quick List

detailed photo of | amateur photo of | flicker 2008 photo of | fantastic artwork of | vintage photograph of | Unreal 5 render of | surrealist painting of | professional advertising design of

Subject Quick List

a man | a woman | a handsome man | a beautiful woman | a monster | a toy | a product



IC-Light V2

Flux-based IC-Light Model with 16ch VAE and native high resolution. See also https://github.com/lllyasviel/IC-Light/discussions/98

前景很多的 case

Image

Preprocessed Foreground

Prompt

detailed photo of Donald Trump and his families, Elon Musk, and many people, sunset over sea

Initial Latent

(•) None ( ) Left Light ( ) Right Light ( ) Top Light ( ) Bottom Light

IC-Light V2

Flux-based IC-Light Model with 16ch VAE and native high resolution. See also https://github.com/lllyasviel/IC-Light/discussions/98

前景比较弱化的 case

Image

Preprocessed Foreground

Prompt

detailed photo of driveways, next to trees, buildings, and traffic light, afternoon light filtering through trees.

Initial Latent

(•) None ( ) Left Light ( ) Right Light ( ) Top Light ( ) Bottom Light

Preview of the later lecture

- 最优扩散方差估计
- SDE and ODE
- Score-based Generative Model
- Pseudo Numerical Methods for Diffusion Models on Manifolds： PNMD/PLMS，对 DDPM 的改进
- 加速采样
- Flow Matching
- Rectified Flow
- 大图生成 upscaling
- 蒸馏 for one-step generation
- Consistency Model

References

See in main text

- Others:

[1] Luo C. Understanding diffusion models: A unified perspective. arXiv 2022[J]. arXiv preprint arXiv:2208.11970.

[2] Yang L, Zhang Z, Song Y, et al. Diffusion models: A comprehensive survey of methods and applications[J]. ACM Computing Surveys, 2023, 56(4): 1-39.

- Other resources you may refer to:

https://github.com/Fafa-DL/Lhy_Machine_Learning

https://huggingface.co/docs/diffusers/index

https://jalammar.github.io/illustrated-stable-diffusion/