

# 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  $p_{\text{data}}(x)$  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  $\theta$ ? MLE  $p_\theta(x) = \int_z p(z)p_\theta(x|z)dz$
- Identity:

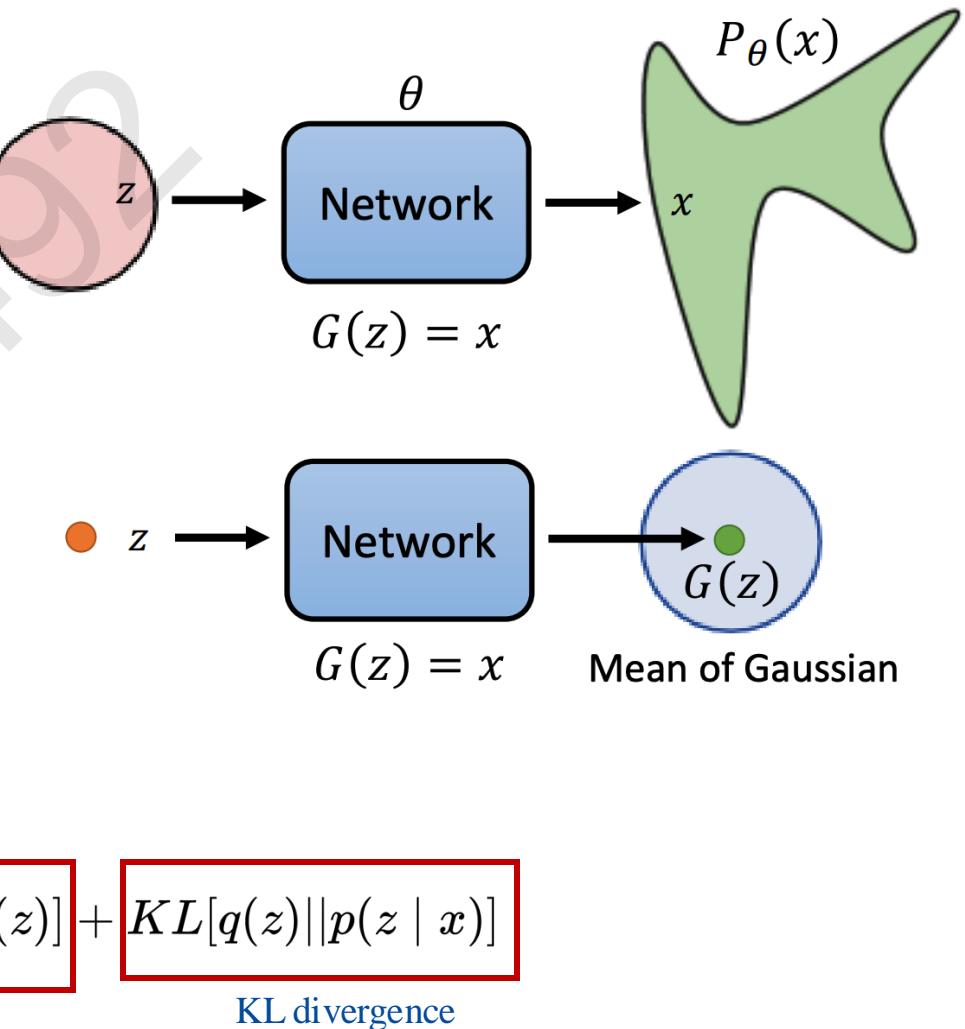
$$\log p(x) = \int q(z) \log p(x|z) dz$$

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

For arbitrary distribution  $q(z)$  of  $z$

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

Evidence Lower Bound (ELBO)



# VAE and ELBO

Do a little math

$$\log p(x) = \underbrace{\int q(z) \log p_\theta(x | z) dz}_{\text{Evidence}} - KL[q(z) || p(z)] + \underbrace{KL[q(z) || p(z | x)]}_{\text{KL divergence}}$$

- $\log p(x) \geq \text{ELBO}$  (KL divergence  $\geq 0$ )

Maximize ELBO  $\Rightarrow$  Increase  $\log p(x)$

- What is  $q(z)$  ?

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

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

- 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

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

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

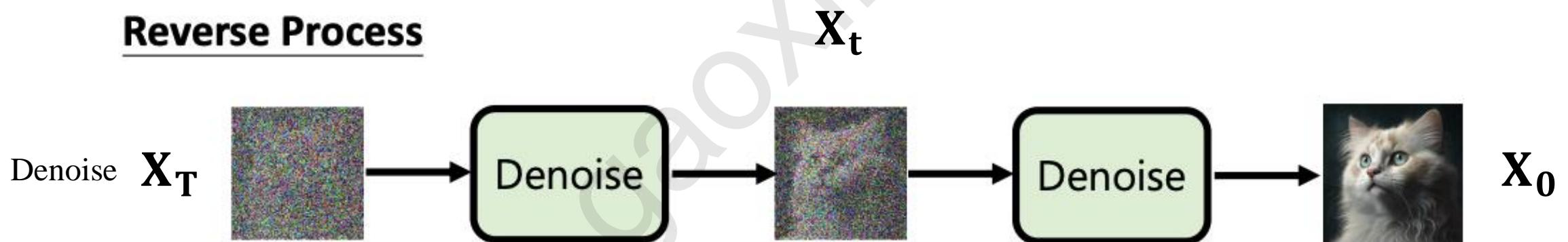
- Objective:  $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x | z)] - KL[q_\phi(z | x) || p(z)]$

# Diffusion

## Forward Process



## Reverse Process



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

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

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)

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

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

# DDPM Denoising Diffusion Probabilistic Model

- **Bayes' Rule:**  $p(x_{t-1}|x_t) = \frac{p(x_t|x_{t-1})p(x_{t-1})}{p(x_t)}$
- We know conditional the distribution given  $x_0$

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

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

$p(x_t|x_{t-1}), p(x_{t-1}|x_0), p(x_t|x_0)$  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  $\mu_\theta(x_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{(1 - \bar{\alpha}_t)}\epsilon_\theta(x_t, t))$

→ **Loss:**  $\frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(x_t, t)\|^2 = \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2$

# DDPM Denoising Diffusion Probabilistic Model

$$\begin{aligned}
 \log p(\mathbf{x}) &\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] & (47) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] & (48) \\
 &\equiv \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^T p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] & (49) \\
 &\equiv \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)\prod_{t=2}^T p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_1|\mathbf{x}_0)\prod_{t=2}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right] & (50) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_{\theta}(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \right] & (51) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} \right] & (52) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}} \right] & (53) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \log \prod_{t=2}^T \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] & (54) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \sum_{t=2}^T \log \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] & (55) \\
 &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] & (56) \\
 &= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)] + \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} \right] + \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{x}_0)} \left[ \log \frac{p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \right] & (57) \\
 &= \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} [\log p_{\theta}(\mathbf{x}_0|\mathbf{x}_1)]}_{\text{reconstruction term}} - \underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{\text{prior matching term}} - \sum_{t=2}^T \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} [D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_{\theta}(\mathbf{x}_{t-1}|\mathbf{x}_t))]}_{\text{denoising matching term}} & (58)
 \end{aligned}$$

- ELBO

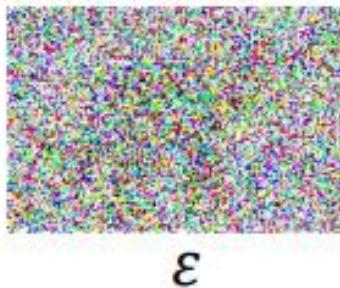
- Perspective from Latent Variable Model (like VAE)

- 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

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

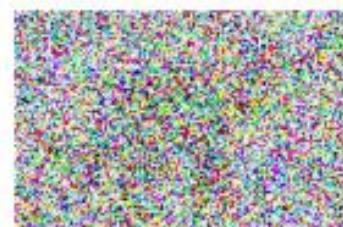
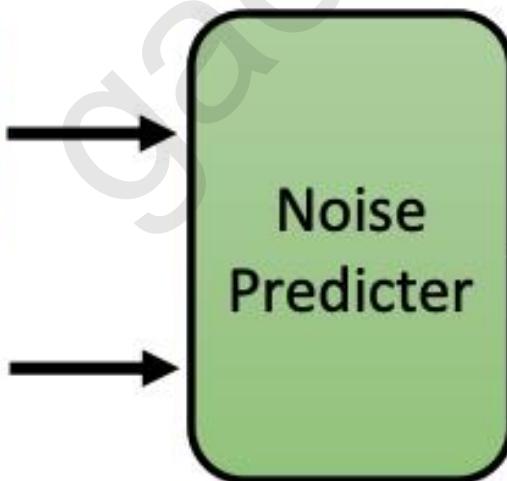


Sample  $t$

$$\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon = \text{Sample } t$$



$t$



$\varepsilon$

```
# 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
```

```
# 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
```

```
# 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)

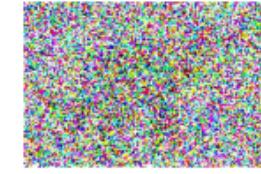
        # loss is mean squared error between the predicted and true noise
        loss = F.mse_loss(pred_noise, noise, reduction='sum') / x.shape[0]
        print(f'loss: {loss.item():.4f}', end='\r')
        loss.backward()

        optim.step()
```

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



$x_0$ : clean image



$\epsilon$ : noise

## Algorithm 1 Training

- 1: **repeat**
  - 2:  $x_0 \sim q(x_0)$   $\leftarrow$  sample clean image
  - 3:  $t \sim \text{Uniform}(\{1, \dots, T\})$
  - 4:  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$   $\leftarrow$  sample a noise
  - 5: Take gradient descent step on  

$$\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2$$
  - 6: **until** converged
- Noisy image
- Target Noise      Noise predictor
- $\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_T$   
smaller

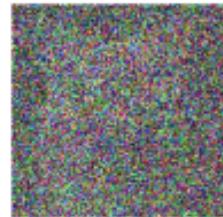
# DATASET

INPUT		OUTPUT / LABEL
Noise Amount	Noisy Image	Noise sample
3		
14		
7		
42		
2		
21		

# MODEL

Noise Predictor  
(UNet)

# Inference

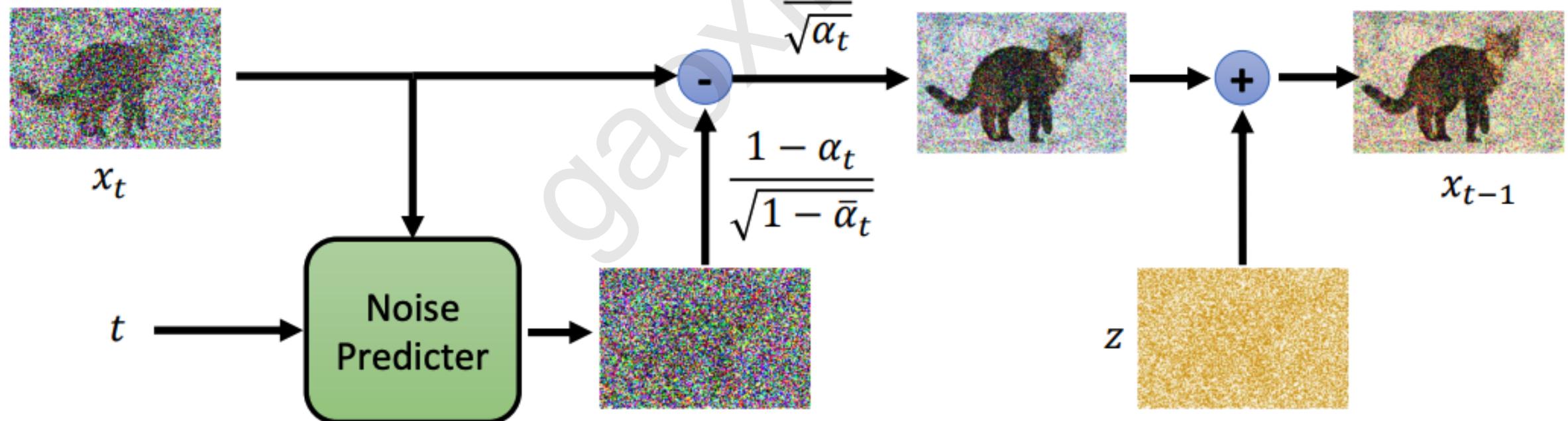


$x_T$

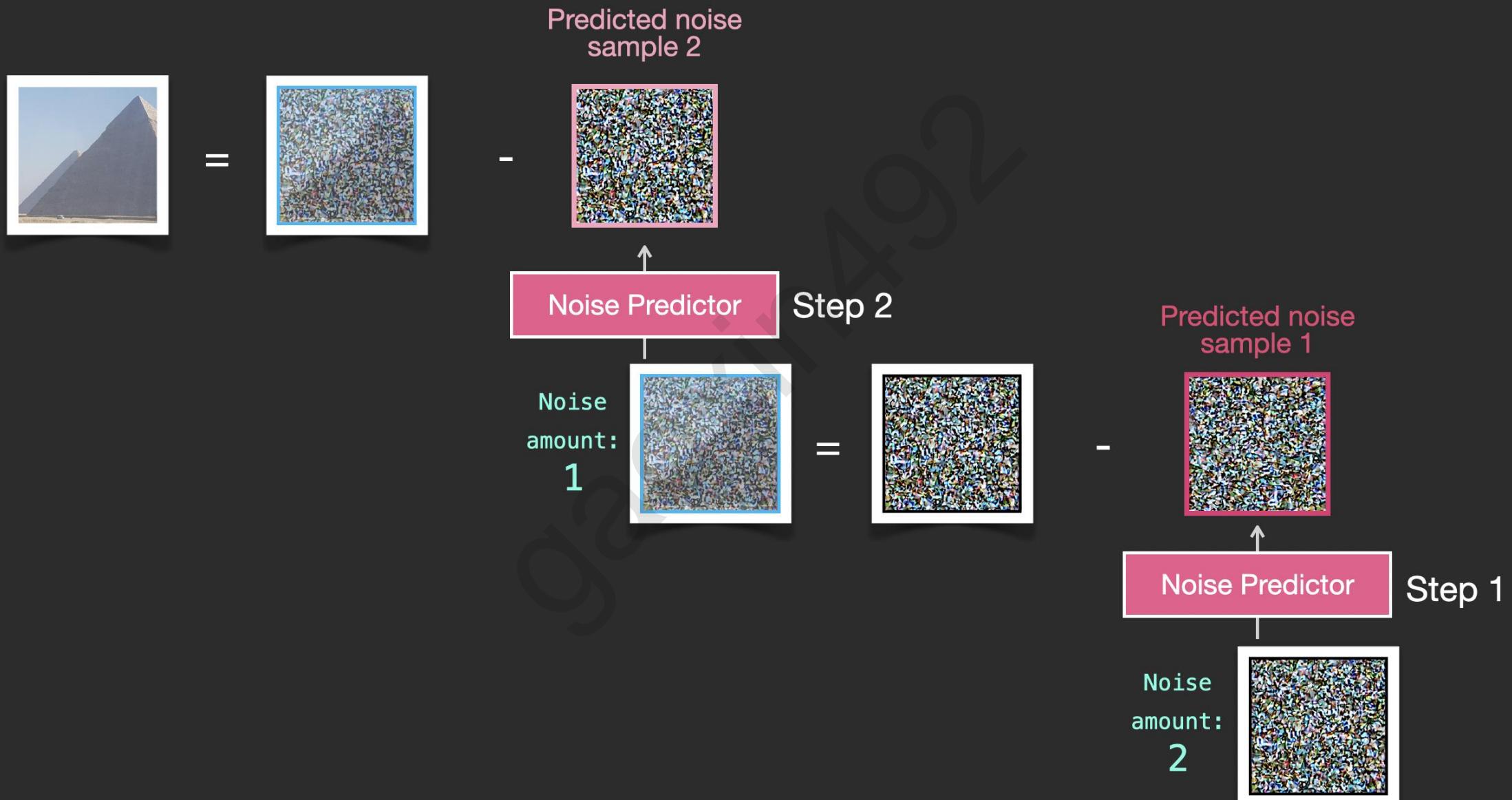
## Algorithm 2 Sampling

```
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$            sample a noise!?
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  $\mathbf{x}_0$ 
```

$\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_T$   
 $\alpha_1, \alpha_2, \dots, \alpha_T$



# Image Generation by Reverse Diffusion (Denoising)



# Stochasticity

Think again about the stochasticity

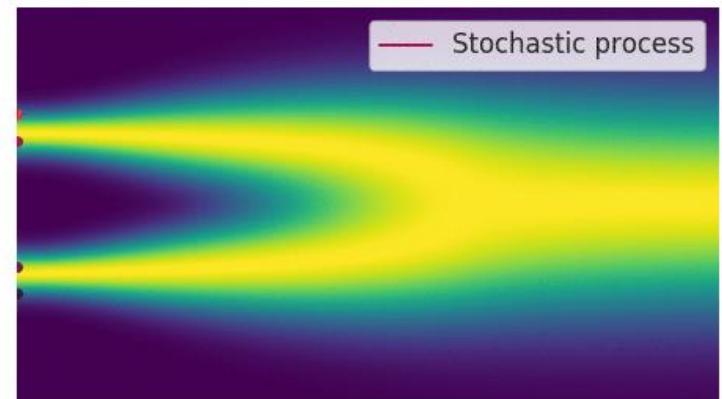
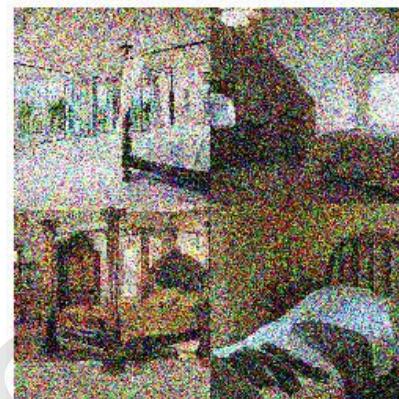
$$p(x_{t-1}|x_t, x_0) = \mathcal{N} \left( x_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \textcolor{red}{x_0} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \mathbf{I} \right)$$

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

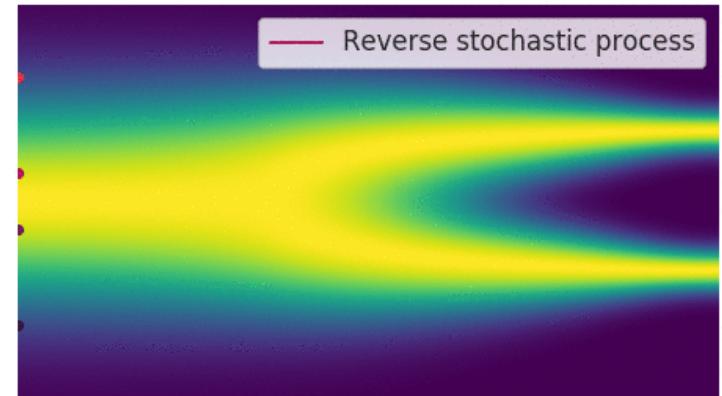
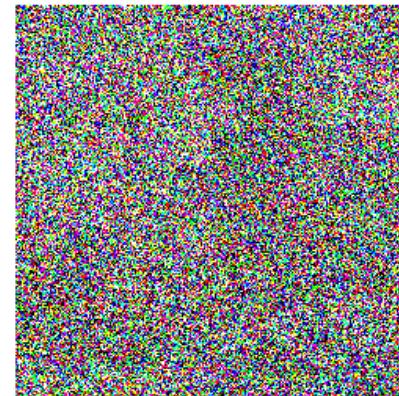
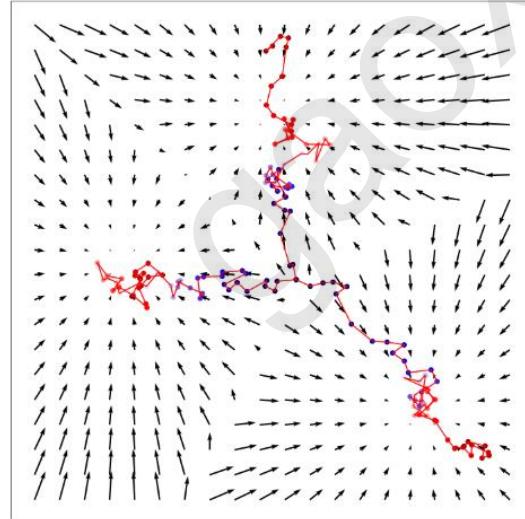
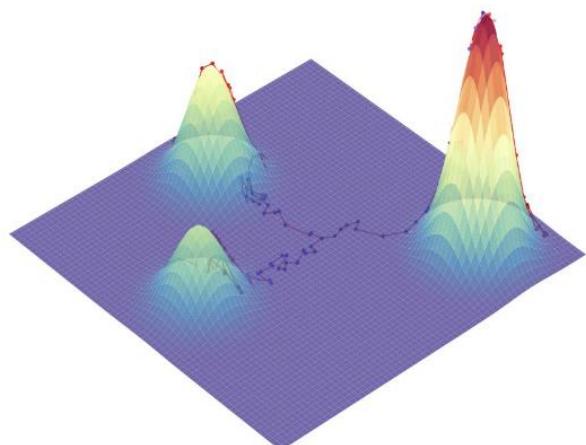
$$\rightarrow q(x_{t-1}|x_t) \approx \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{\alpha_t}} x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_\theta(x_t, t), \sigma_t \mathbf{I})$$

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

**Sampling from  
a distribution !**

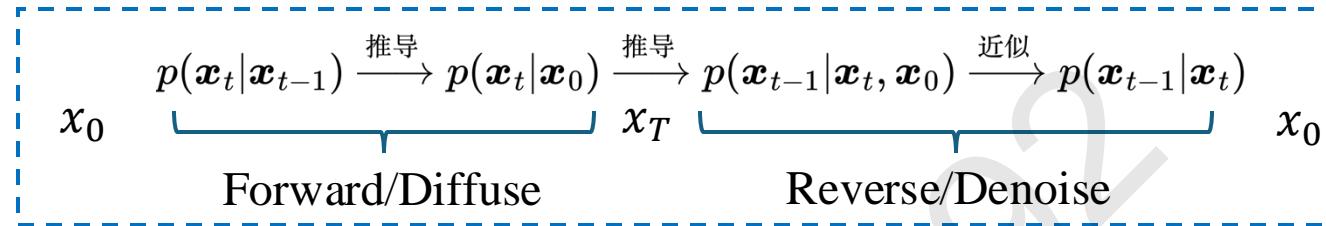


Perturbing data to noise with a continuous-time stochastic process.



Generate data from noise by reversing the perturbation procedure.

# DDIM Denoising Diffusion Implicit Model



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

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

- **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 ?

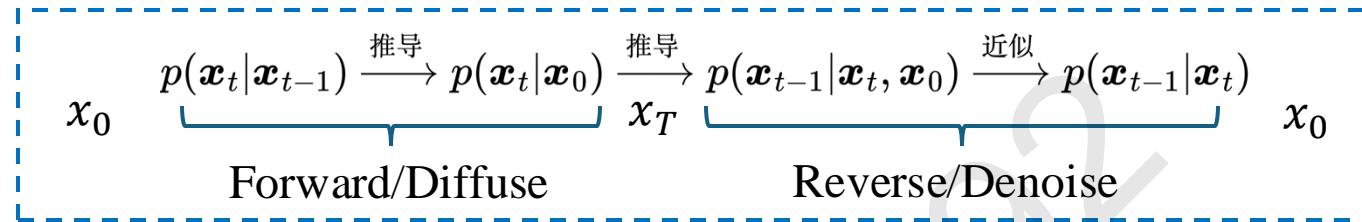
$$p(x_{t-1}|x_t, x_0) = \frac{p(x_t|x_{t-1})p(x_{t-1}|x_0)}{p(x_t|x_0)} \quad (*)$$

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

- 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



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

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

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

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

$\alpha, \beta$  相关的参数都是预先设定好的超参数，是已知的

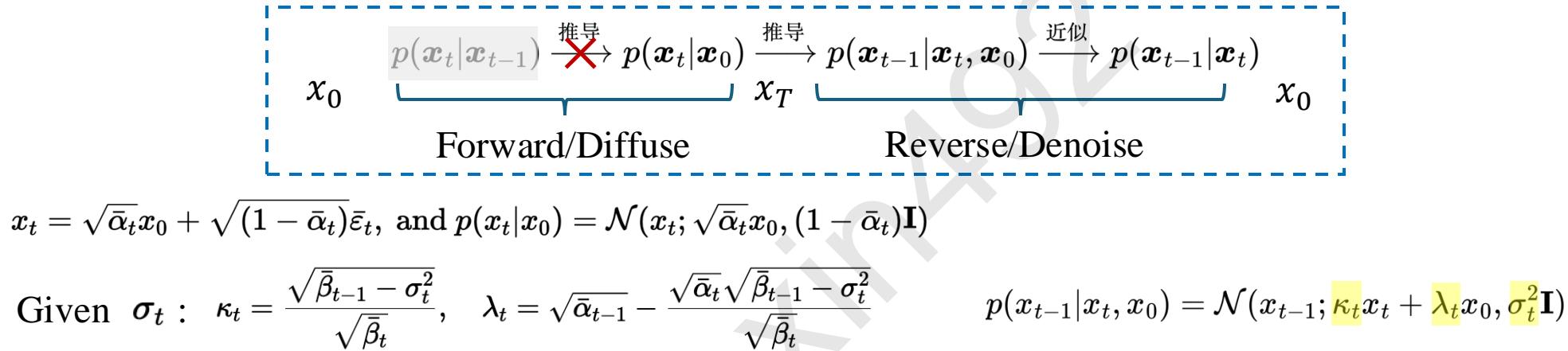
- **DDPM:**  $\sigma_t^2 = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$
- **DDIM:**  $\sigma_t^2 = 0$       Implicit 隐式的概率模型，确定性采样过程，不带随机性
- **Larger covariance:**  $\sigma_t^2 = \beta_t$

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

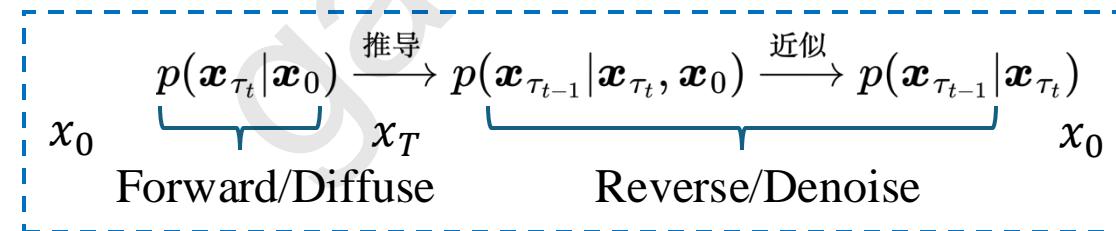
Remark: 在给定  $p(x_{t-1}|x_t, x_0)$  后，我们还可以反推出  $p(x_t|x_{t-1})$ ，即知道每一步是怎么扩散到噪声的

# DDIM Denoising Diffusion Implicit Model

- Accelerated Generation Process



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



**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:  $x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\varepsilon_t$ ,  $\varepsilon_t \sim \mathcal{N}(0, 1)$ ,  $t = 1, \dots, T$   
 连续化一般化  $x_{t+\Delta t} - x_t = \mathbf{f}_t(x_t)\Delta t + g_t\sqrt{\Delta t}\varepsilon$ ,  $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$ ,  $\Delta t \rightarrow 0$
- => 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  $p(x_{t+\Delta t}|x_t) = \mathcal{N}(x_{t+\Delta t}; x_t + \mathbf{f}_t(x_t)\Delta t, g_t^2 \Delta t \mathbf{I}) \propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \mathbf{f}_t(x_t)\Delta t\|^2}{2g_t^2 \Delta t}\right)$
- 逆向过程推导

$$\begin{aligned} p(x_t|x_{t+\Delta t}) &= \frac{p(x_{t+\Delta t}|x_t)p(x_t)}{p(x_{t+\Delta t})} = p(x_{t+\Delta t}|x_t) \exp(\log p(x_t) - \log p(x_{t+\Delta t})) \\ &\propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - \mathbf{f}_t(x_t)\Delta t\|^2}{2g_t^2 \Delta t} + \log p(x_t) - \log p(x_{t+\Delta t})\right) \end{aligned}$$

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

# SDE

DDPM

正向 SDE  $dx = f_t(x)dt + g_t dw$

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

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

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

$$\begin{aligned} p(x_t|x_{t+\Delta t}) &\propto \exp\left(-\frac{\|x_{t+\Delta t} - x_t - [f_t(x_t) - g_t^2 \nabla_{x_t} \log p(x_t)]\Delta t\|^2}{2g_t^2\Delta t}\right) \\ &\approx \exp\left(-\frac{\|\mathbf{x}_t - \mathbf{x}_{t+\Delta t} + [\mathbf{f}_{t+\Delta t}(x_{t+\Delta t}) - g_{t+\Delta t}^2 \nabla_{x_{t+\Delta t}} \log p(x_{t+\Delta t})]\Delta t\|^2}{2g_{t+\Delta t}^2\Delta t}\right) \end{aligned}$$

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

逆向 SDE  $\Delta t \rightarrow 0$ ,  $dx = [\mathbf{f}_t(x) - g_t^2 \nabla_x \log p_t(x)]dt + g_t dw$

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

Loss 的推导本次省略

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

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

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

# SDE and ODE 大一统

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

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

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

我们发现这个 FP 方程也是以下 SDE 的 FP 方程：

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

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：

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

# SDE and ODE 大一统

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

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

What if  $\sigma_t = 0$  ?

## Probability flow ODE

$$dx = \left( \mathbf{f}_t(x) - \frac{1}{2}g_t^2\nabla_x \log p_t(x) \right) dt \quad \text{Deterministic transform}$$

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

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

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

Then, by **Tweedie's Formula**, we have:

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

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

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

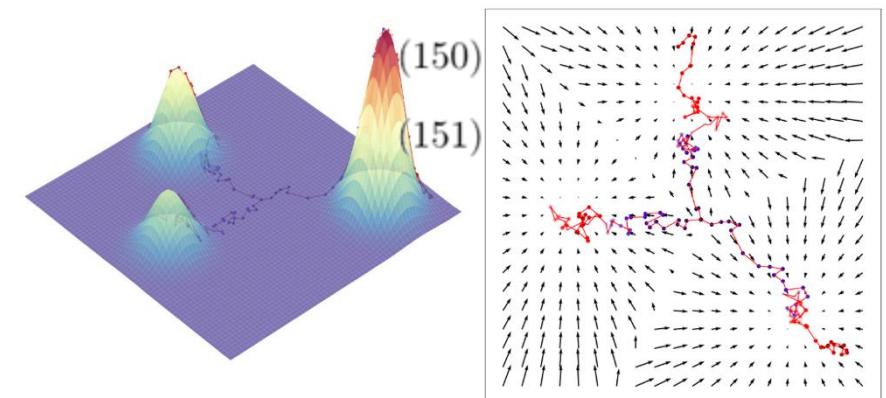
$$\mathbf{x}_0 = \frac{\mathbf{x}_t + (1-\bar{\alpha}_t)\nabla \log p(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}} = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t}{\sqrt{\bar{\alpha}_t}} \quad (149)$$

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

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

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

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



# 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。

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

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

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

对原始图像  $x_0$  的后验和后验均值

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

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

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

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

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

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

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

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

- 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  $p_\theta(x_{t-1}|x_t, y)$ ,  $y = label \text{ or } \emptyset$

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

**Require:**  $p_{\text{uncond}}$ : probability of unconditional training

- ```

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

```

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

$\varepsilon_\theta(x_t, t, y)$  or  $\varepsilon_\theta(x_t, t, \emptyset)$

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)

arXiv:1503.03585v8 [cs.LG] 18 Nov 2015

[https://arxiv.org/pdf/1503.03585](https://arxiv.org/pdf/1503.03585.pdf)

## Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein  
Stanford University  
[jascha@stanford.edu](mailto:jascha@stanford.edu)  
Eric A. Weiss  
University of California, Berkeley  
[eaeweiss@berkeley.edu](mailto:eaeweiss@berkeley.edu)  
Niru Maheswaranathan  
Stanford University  
[nirum@stanford.edu](mailto:nirum@stanford.edu)  
Surya Ganguli  
Stanford University  
[sganguli@stanford.edu](mailto:sganguli@stanford.edu)

### Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions, such that learning, sampling, inference, and evolution are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea is to use by hand, and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a generative model 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 of deep generative models with hundreds of layers of neurons, 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 can be analytically evaluated and easily fit to data (e.g., a Gaussian or Laplace). However,

Proceedings of the 32<sup>nd</sup> International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

these models are unable to apply discrete structure in rich data sets. On the other hand, models that are *flexible* can be modeled to fit complex arbitrary data. For example, we can define models in terms of a smooth (continuous) function  $\phi(x)$  yielding the flexible distribution  $p(x) = \frac{\partial \phi}{\partial Z}$ , where  $Z$  is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from these models typically requires very expensive Monte Carlo processes.

A variety of analytical approximations exist which ameliorate this, 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 (Hinton et al., 2002), Helmholtz free energy and probability flow (Sohl-Dickstein et al., 2011a), minimum KL contractive (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvonen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Koller et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective<sup>1</sup>.

### 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.

<sup>1</sup>Non-parametric methods can be seen as transitioning smoothly between the two extremes. For example, 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.

arXiv:1907.05600v1 [cs.LG] 12 Jul 2019

<https://arxiv.org/pdf/1907.05600.pdf>

## Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song  
Stanford University  
[yangsong@cs.stanford.edu](mailto:yangsong@cs.stanford.edu)  
Stefano Ermon  
Stanford University  
[ermon@cs.stanford.edu](mailto: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 estimate the noise levels of the data manifold to estimate the corresponding scores, i.e., the 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 temperature gets closer to the data manifold. Our framework works with any model architecture, requires no prior knowledge of the data distribution 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-HQ, LSUN, and CIFAR10, and 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 audio [47], improve the performance of semi-supervised learning [24, 8], detect adversarial examples and outliers [19], and variational autoencoders (VAEs) have synthesized striking image and audio samples [12, 25, 3, 55, 35, 23, 10, 30, 51, 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].

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 surrogate losses (e.g., the evidence lower bound) used in variational autoencoders to condition discrete 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, and the latter uses a loss function that minimizes  $J$ -divergences [34] or integral probability metrics [2, 45] between model and data distributions.

In this paper, we explore a new principle for generative modeling based on estimating the *Sisini 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.

<https://arxiv.org/pdf/2006.11239.pdf>

## Denoising Diffusion Probabilistic Models

Jonathan Ho  
UC Berkeley  
[jonathanho@berkeley.edu](mailto:jonathanho@berkeley.edu)  
Alay Falin  
UC Berkeley  
[a.falin@berkeley.edu](mailto:a.falin@berkeley.edu)  
Peter Abbeel  
UC Berkeley  
[pabbeel@cs.berkeley.edu](mailto: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 learn a prior that is expressive enough to act 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/jonathanho/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, 51, 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].

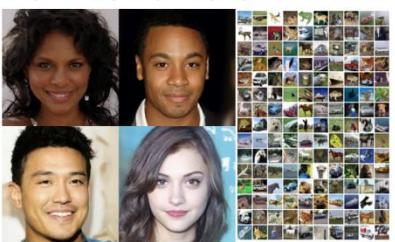


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Preprint. Under review.

<https://arxiv.org/pdf/2011.13456.pdf>

## SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song<sup>\*</sup>  
Stanford University  
[yangsong@cs.stanford.edu](mailto:yangsong@cs.stanford.edu)  
Jascha Sohl-Dickstein  
Google Brain  
[jascha@google.com](mailto:jascha@google.com)  
Abhishek Kumar  
Google Brain  
[abhishek@google.com](mailto:abhishek@google.com)  
Stefano Ermon  
Stanford University  
[ermon@cs.stanford.edu](mailto:ermon@cs.stanford.edu)  
Ben Poole  
Google Brain  
[pooleb@google.com](mailto:pooleb@google.com)

### ABSTRACT

Creating noise-free data is easy; creating data from noise is generative modeling. We propose a stochastic differential equation (SDE) that smoothly transforms a complex data distribution to a known noise distribution (slowly increasing 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 (a.k.a., score) of the prior 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 can quickly produce appealing in-distribution samples, and score-based generative models, and allow for new sampling procedures. In particular, we introduce a predictor-corrector framework to correct errors in the evolution of the discretized reverse-time SDE, as well as an auxiliary 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-conditioned generative image modeling, and colorization. Considering the potential improvements we achieve, we achieve state-of-the-art 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-density) at each noise scale, and then uses Langevin dynamics to sample from a sequence of decaying noise scales during generation. *Denoising diffusion probabilistic modeling* (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) trains a family of prior SDEs to make training tractable. In both cases, we use knowledge of the functional form of the reverse distribution 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 (Borodzki 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

<sup>\*</sup>Work done during an internship at Google Brain.

1

Physics Foundation  
2015.11  
SGM  
2019.07  
DDPM  
2020.06  
SDE  
2020.11

# Timeline (2)

[https://arxiv.org/pdf/2010.02502v1](https://arxiv.org/pdf/2010.02502v1.pdf)

## DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon  
Stanford University  
[{tsong,chenlim,ermon}@cs.stanford.edu](mailto:{tsong,chenlim,ermon}@cs.stanford.edu)

### ABSTRACT

Denoising diffusion probabilistic models (DDPMs) have achieved high quality image generation after 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 DDIMs, we propose a new sampling procedure based on a Markov chain process. We construct a class of non-Markovian diffusions 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<sup>4</sup> to 10<sup>5</sup> times faster than DDPMs, while maintaining quality, allowing 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](https://github.com/jmsong/ddim).

### 1 INTRODUCTION

Deep generative models have demonstrated the ability to produce high quality samples in many domains (Kingma & Welling, 2013; Oord et al., 2016). In terms of image generation, generative adversarial networks (GANs; Goodfellow et al., 2014) currently achieve higher quality than likelihood-based methods such as variational autoencoders (Kingma & Welling, 2013) and autoregressive models (van den Oord et al., 2016b) and normalizing flows (Rezende & Mohamed, 2015; Dinh et al., 2014). However, GANs require very specific choices in optimization and architectures in order to stabilize training (Karras et al., 2017; Karras et al., 2018; Brock et al., 2018), and could fail to cover modes of the data distribution (Zhuo 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 neural conditional score networks (NCSN; Song & Ermon, 2020) have shown that producing samples from these models does not require 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 passed through a Markov Chain process to produce a final denoised 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 (Ho et al., 2020).

A critical drawback of these models is that they require many iterations to produce a high quality sample. For DDPMs, this is because 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 other generative models (e.g., VAEs). For example, it takes about 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 from a DDPM would take about 10 days.

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,

1

DDIM

2020.10

[https://arxiv.org/pdf/2102.09672](https://arxiv.org/pdf/2102.09672.pdf)

## Improved Denoising Diffusion Probabilistic Models

Alex Nichol \*† Prafulla Dhariwal \*†

### 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 and generate high quality sample. Additionally, we find that the reverse process 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 DDPMs and GANs over a 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

Sohn-Dickstein et al. (2015) introduced diffusion probabilistic models, a class of generative models which match a data distribution by learning to reverse a forward, multi-step diffusion 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 learns a gradient of the log-density of data distribution using denoising score matching (Hyvärinen, 2005). It has recently been shown that this class of models can produce images (Ho et al., 2020), audio (Chen et al., 2020; Jaiswal-Nagarajan et al., 2020) and video (Chen et al., 2020; 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.

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 only a few forward passes, thus speeding up sampling for use in practical applications. In this 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 (Kynänen et al., 2019) and discover that diffusion models achieve 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.,

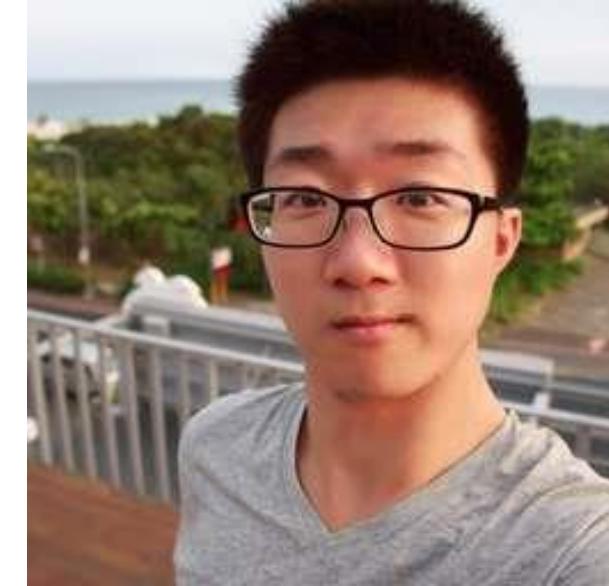
2020), we find that DDIMs are significantly faster than DDPMs, while maintaining quality.

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.pdf>

The figure consists of a 4x4 grid of 16 images. The images include a penguin, a red squirrel, a dog, a fish, a white wolf, a snow leopard, a butterfly, a baboon, a panda, a red panda, a tennis ball, a husky, a classic car, a green frog, a blue bird, and a cake with red hearts.

arXiv:2105.03233v1 [cs.LG] 11 May 2021

# 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 recently introduced method to trade off mode and sample fidelity in generating images from a diffusion model, in the same way as low temperature sampling or annealing in the training 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 separate from the diffusion model. It also asks the question whether one can have classifier-free guidance. We show that this is possible and can be performed by a pure generative model without such a classifier: in fact classifier-free guidance, we jointly train a conditioned and an unconditioned diffusion model to obtain a diffusion model that performs well on classification estimates to attain a trade-off between sample quality and diversity similar to obtained using classifier guidance.

## 1 INTRODUCTION

Diffusion models have recently emerged as an expressive and flexible family of generative models for learning competitive sample quality and likelihood scores on image and audio synthesis (Drost et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2012; Kingma & Song et al., 2021a). These models have derived synthesis performance metrics (Ho et al., 2020; Kingma & Song et al., 2021b), which are often used to evaluate generative steps (Ho et al., 2021) and they have delivered ImageNet generation results媲美 pretraining BigBiGAN (Ho et al., 2019) and VQ-VAE-2 (Razavi et al., 2019) in terms of FID score and class consistency (Ho et al., 2021).

Dhariwal & Neelakantan (2021) proposed *classifier guidance*, a technique to boost the performance of a diffusion model using an extra trained classifier. Prior to classifier guidance, how to generate “low temperature” samples from a diffusion model similar to the truncated BigBiGAN (Brook et al., 2019) or low temperature Glow (Kingma & Metz, 2018) models has been studied. The methods, which either directly add noise or add during diffusion sampling, are ineffective (Dhariwal & Neelakantan, 2021). Consequently, instead mixes a diffusion model’s score estimate with the input gradient of the loss



arXiv:2207.12598v1 [cs.LG] 26 Jul 2022

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

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

Alex Nichol<sup>\*</sup> Prafulla Dhariwal<sup>\*</sup> Aditya Ramesh<sup>\*</sup> Pranav Shyam<sup>\*</sup> Pamela Mishkin<sup>\*</sup> Bob McGrew<sup>\*</sup> Ilya Sutskever<sup>\*</sup> Mark Chen<sup>\*</sup>

## 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-conditioned image synthesis, and find that they can outperform state-of-the-art 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-conditioned diffusion model using classifier-free guidance are favored over 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>.

### Introduction

Images, such as illustrations, paintings, and photographs, have often been easily described by text, but require specialized skills and hours of labor to create. Therefore, capable of generating realistic images from natural language prompts, humans are turning to AI to generate 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-conditioned image models are capable of synthesizing images from free-form text prompts, and can generate 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 test prompts.

On the other hand, unconditional image models can synthesize photorealistic images (Brock et al., 2018; Karras et al., 2019a; 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 (Ho et al., 2020; Radford & Ermon, 2020) have emerged as a promising avenue of generative modeling, achieving state-of-the-art sample quality on a number of image generation benchmarks (Ho et al., 2020; Dhariwal et al., 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 trained to output a probability distribution over the image sampling process, gradients from the classifier are used to guide the sample towards the label. Ho et al. (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 create images that are both photorealistic and reflect user instructions, we apply guided diffusion models to the problem of text-conditioned 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, fulia@openai.com, aramesh@openai.com

# Timeline (4)

arXiv:2103.00020v1 [cs.CV] 26 Feb 2021

[https://arxiv.org/pdf/2103.00020](https://arxiv.org/pdf/2103.00020.pdf)

## Learning Transferable Visual Models From Natural Language Supervision

Alec Radford<sup>\*†</sup> Jong Wook Kim<sup>\*†</sup> Chris Hallacy<sup>\*</sup> Aditya Ramesh<sup>\*</sup> Gabriel Goh<sup>\*</sup> Sandhini Agarwal<sup>\*</sup> Girish Sastry<sup>\*</sup> Amanda Askell<sup>\*</sup> Pamela Mishkin<sup>\*</sup> Jack Clark<sup>\*</sup> Gretchen Krueger<sup>\*</sup> Ilya Sutskever<sup>\*</sup>

**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 the model's ability to learn new concepts without 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 most promising task of predicting what common objects with language 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 on this dataset, it is used to learn to predict novel concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study the robustness of this approach by benchmarking on over 30 different evaluation benchmarks, spanning tasks such as OCR, action recognition in videos, geo-localization, and many types of fine-grained object classification.

The model achieves state-of-the-art to test 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 vision examples with only 1.28 million training examples it was trained on. To release our code and pre-trained model weights at <https://github.com/OpenAI/CLIP>.

### 1. Introduction and Motivating Work

Pre-training methods which directly train raw text have revolutionized NLP over the last few years (Bai & Le, 2013; Peters et al., 2018; Bradbury & Radford, 2018; Radford et al., 2018; Devlin et al., 2018; Raffel et al., 2019).

<sup>\*</sup>Equal contribution. <sup>†</sup>OpenAI, San Francisco, CA 94110, USA.  
Correspondence to: <alec.jungwook>@openai.com>

arXiv:2204.06125v1 [cs.CV] 13 Apr 2022

[https://arxiv.org/pdf/2204.06125](https://arxiv.org/pdf/2204.06125.pdf)

## Hierarchical Text-Conditional Image Generation with CLIP Latents

Aditya Ramesh<sup>\*</sup> OpenAI aranesh@openai.com Prafulla Dhariwal<sup>\*</sup> OpenAI prafulla@openai.com Alex Nichol<sup>\*</sup> OpenAI alex@openai.com  
Casey Chu<sup>\*</sup> OpenAI casey@openai.com Mark Chen<sup>\*</sup> OpenAI mark@openai.com

### Abstract

Contrastive objectives such as autoregressive and masked language modeling have scaled across many orders of magnitude in both size and quality, and now enable highly-impressive capabilities. The development of CLIP<sup>†</sup> 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 heads or data-specific pre-training. Figure 1 shows how models 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 contrastive learning models is web-scale collections of images and text, and not the high-quality curated 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 raw text be a similar breakthrough in computer vision? Our work is encouraging.

Over 2 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 a visual representation of sentences via multi-task 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 Models (DBMs) to map low-level image and text features. Justin et al. (2016) extended 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., 2013) 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 performed similarly to ImageNet on downstream 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

arXiv:2205.11487v1 [cs.CV] 23 May 2022

[https://arxiv.org/pdf/2205.11487](https://arxiv.org/pdf/2205.11487.pdf)

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

Chitwan Saharia<sup>\*</sup>, William Chan<sup>\*</sup>, Saurabh Saxena<sup>\*</sup>, Lalita Li<sup>\*</sup>, Jay Wang<sup>\*</sup>, Emily Denton, Seyyed Kaveyvar Seyyed Ghaniipour, Burcu Karagol Ayan, S. Sarra Manzoor, Daniel Green Lopez, Tim Salimans, Jonathan Ho<sup>\*</sup>, David J Fleet<sup>\*</sup>, Mohammad Norouzi<sup>\*</sup>, {saharia, williamchan, masee@ziiidu.com, {xrb9, lala}@fb.com, {wang, jwang, joonathanho, 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-scale text-to-image diffusion models in understanding text and images on the street-level and fine-grained levels of language and imagery. One key discovery is that generic large language models (e.g. T5), pretrained on text-only corpora, are surprisingly large at encoding text for image synthesis: increasing the size of the language model in Imagen leads both to fidelity and image synthesis quality. We also show that scaling the size of the image diffusion model, the primary expense of CLIP encoder latency, leads to improved image quality 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]. This has emerged as a successful representation learner for images. CLIP embeddings have a number of desirable properties: they are learned on a large dataset, are non-discriminative, and have fine-grained and have 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, particularly for image and video generation tasks [11, 26, 24]. These text-to-image diffusion models leverage a prior model (e.g. (1), (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-conditioned 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 text prompt. The presence of an encoder and an inverter (the decoder) provides the capabilities beyond text-to-image translation. As in GAN inversion [62, 55], encoding and decoding an input image produces semantically similar 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 as the direction of any encoded text vector (Figure 5), whereas discovering these directions in GAN latent space involves

<sup>\*</sup>Equal contribution.  
<sup>†</sup>Care contribution.

## Google v.s. OpenAI



Transformer

T5

GPT  
CLIP

DALLE  
ChatGPT

CLIP

2022.03

DALLE 2

2022.04

Imagen

2022.05

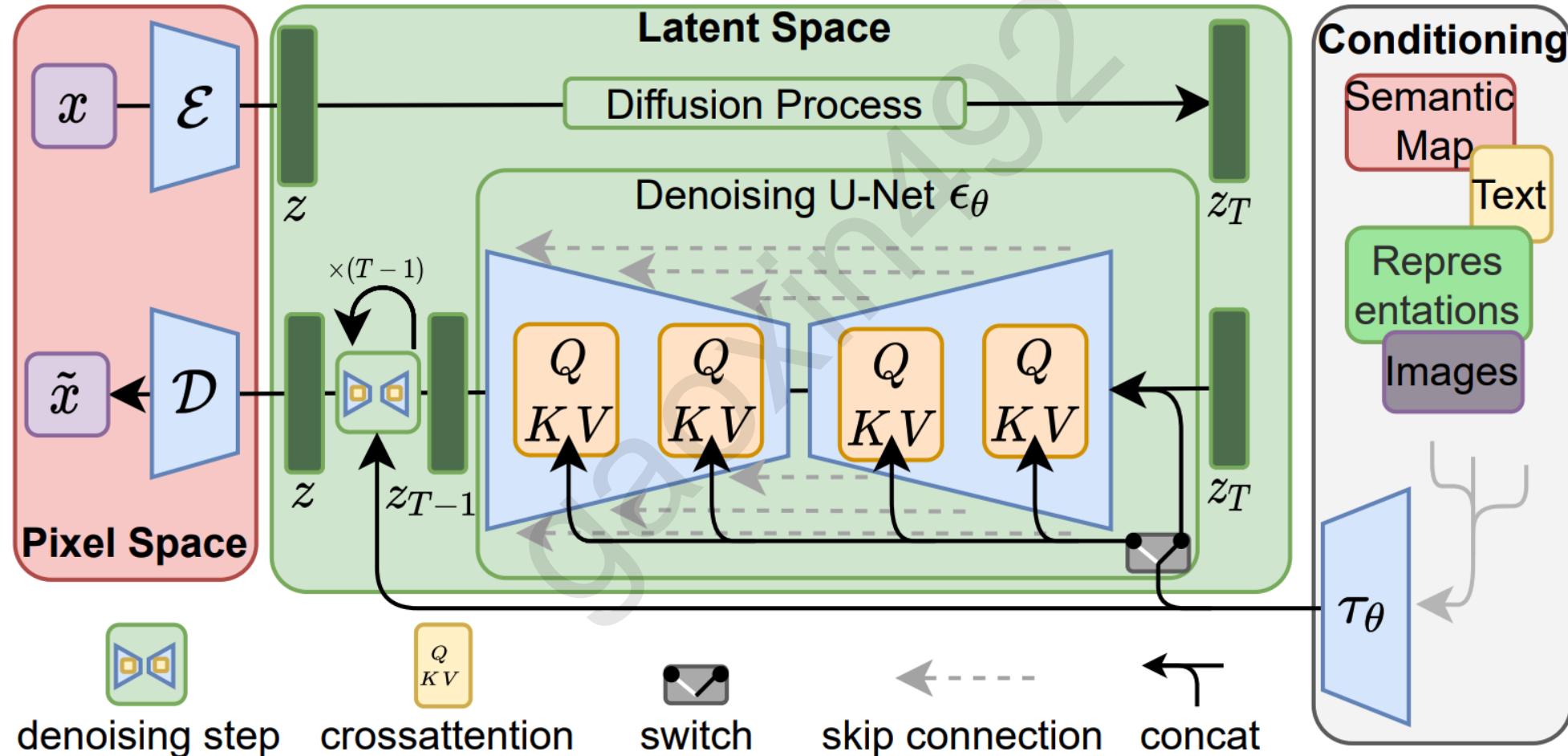


# Latent Diffusion Model

CVPR '22



Patrick Esser   Robin Rombach



# VQ-VAE

- Vector Quantized Variational AutoEncoder

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

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

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

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

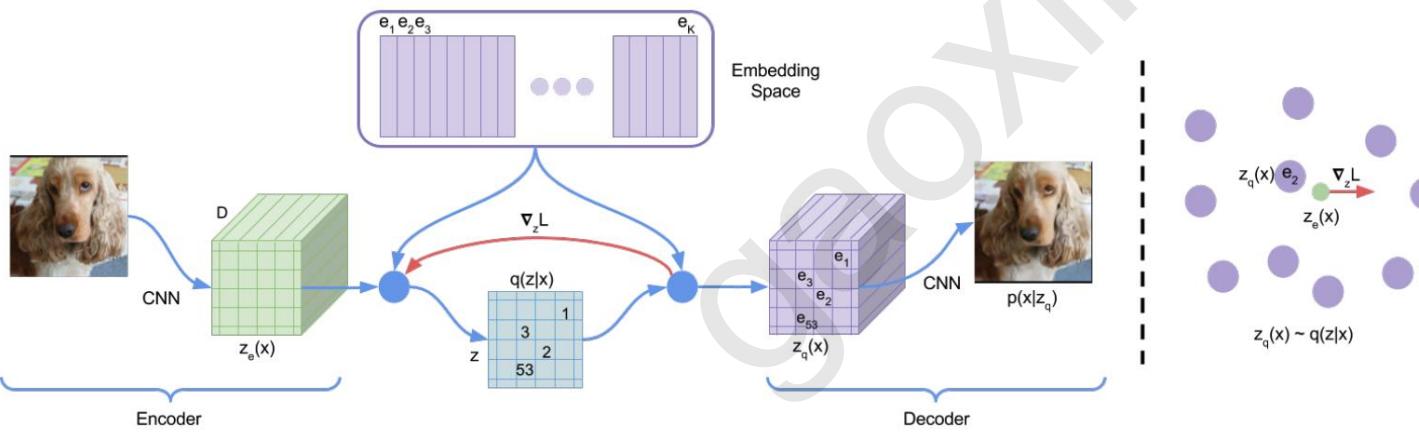


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder  $z(x)$  is mapped to the nearest point  $e_2$ . The gradient  $\nabla_z L$  (in red) will push the encoder to change its output, which could alter the configuration in the next forward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

$$L = x - \text{decoder}(z_e + (z_q - z_e).\text{detach}())$$

<https://arxiv.org/pdf/1711.00937.pdf>

---

## Neural Discrete Representation Learning

---

Aaron van den Oord  
DeepMind  
avondoord@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 does not work on raw discrete, raw than continuous latents; the prior is learned 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 a pixel domain, however neither of these degrades the particular application the feature are used in. One way 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].

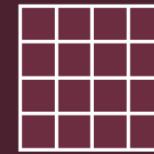
In our work, we introduce a new family of generative models successfully combining the variational autoencoder (VAE) framework with discrete latent representations through a novel parameterisation of the posterior distribution of (discrete) latents given an observation. Our model, which relies on vector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

## Original image



Image  
Encoder

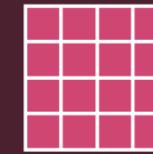
Generate training examples with different amounts of noise added to their compressed/latent version



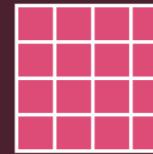
Compressed  
image (latent)



Latent + noise  
sample 1 at  
noise amount 1



Latent + noise  
sample 2 at  
noise amount 2



Latent + noise  
sample 3 at  
noise amount 3

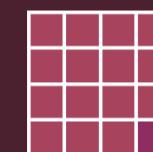
## Image Generation by Reverse Diffusion (Denoising)



Image  
Decoder



Processed  
Image  
Information



UNet  
Step  
**50**



UNet  
Step  
**2**



UNet  
Step  
**1**



Complete  
noise

## Generated image

## Image Information Creator

# Condition

## 1. Cross Attention in UNet

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

To pre-process  $y$  from various modalities (such as language prompts) we introduce a domain specific encoder  $\tau_\theta$  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

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

Here,  $\varphi_i(z_t) \in \mathbb{R}^{N \times d_\epsilon}$  denotes a (flattened) intermediate representation of the UNet implementing  $\epsilon_\theta$  and  $W_V^{(i)} \in \mathbb{R}^{d \times d_\epsilon}$ ,  $W_Q^{(i)} \in \mathbb{R}^{d \times d_\tau}$  &  $W_K^{(i)} \in \mathbb{R}^{d \times d_\tau}$  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

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

where both  $\tau_\theta$  and  $\epsilon_\theta$  are jointly optimized via Eq. 3. This conditioning mechanism is flexible as  $\tau_\theta$  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.pdf>

### Scalable Diffusion Models with Transformers

William Peebles\*  
UC Berkeley

Saining Xie  
New York University

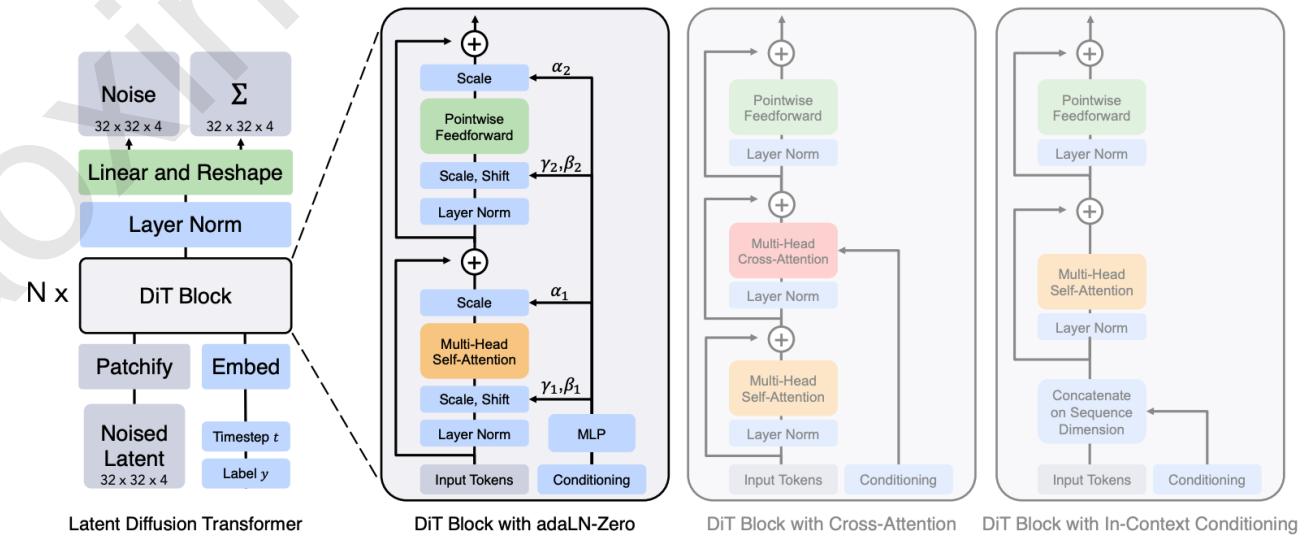
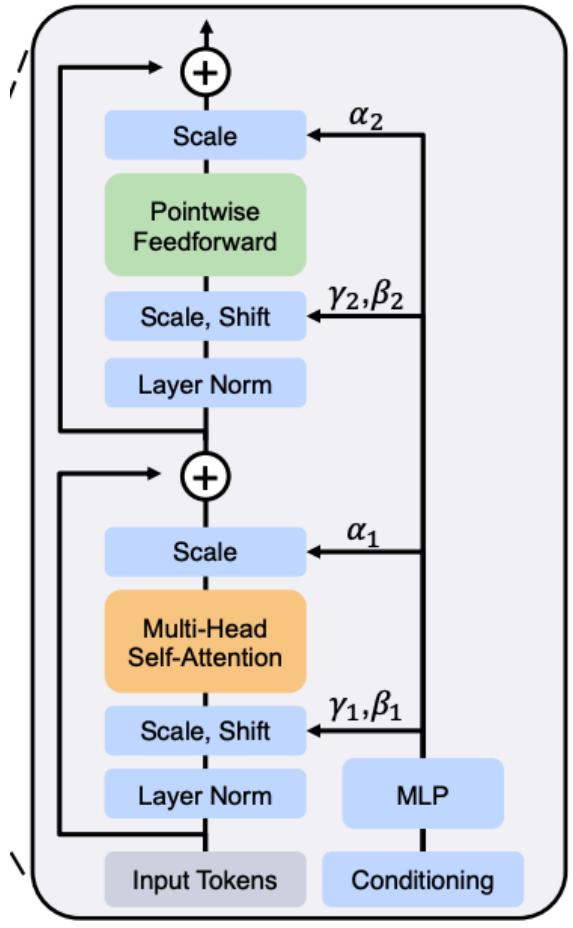


Figure 3. **The Diffusion Transformer (DiT) architecture.** *Left:* We train conditional latent DiT models. The input latent is decomposed into patches and processed by several DiT blocks. *Right:* Details of our DiT blocks. We experiment with variants of standard transformer blocks that incorporate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

# Condition



- Adaptive Layer Normalization

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

- AdaLN-Zero initialize  $\alpha = 0$

$$\alpha(y) \odot f(x) + x$$

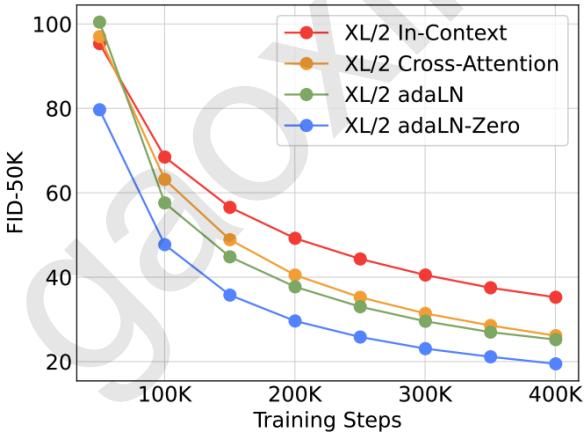
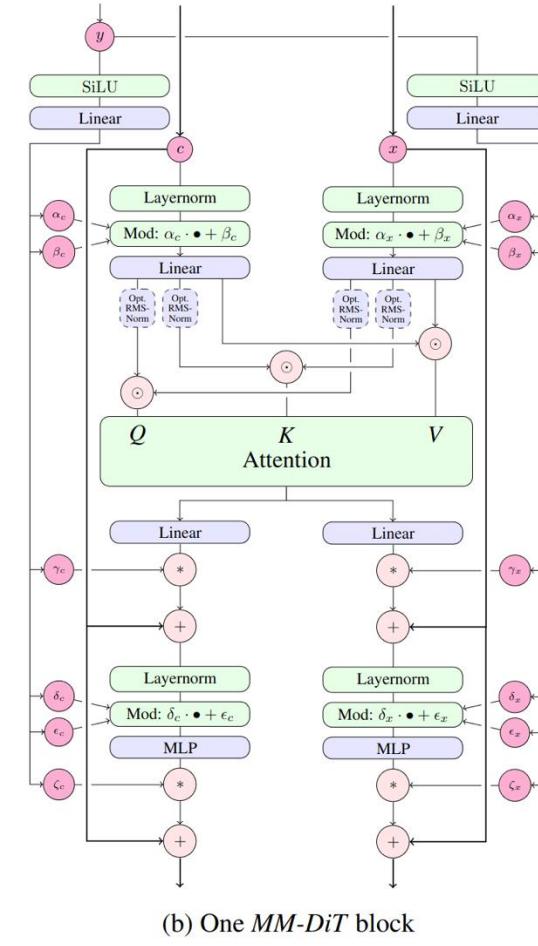


Figure 5. Comparing different conditioning strategies. adaLN-Zero outperforms cross-attention and in-context conditioning at all stages of training.

## 3. MM-DiT in Stable Diffusion v3



<https://arxiv.org/abs/2403.03206>



# 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
- **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...

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

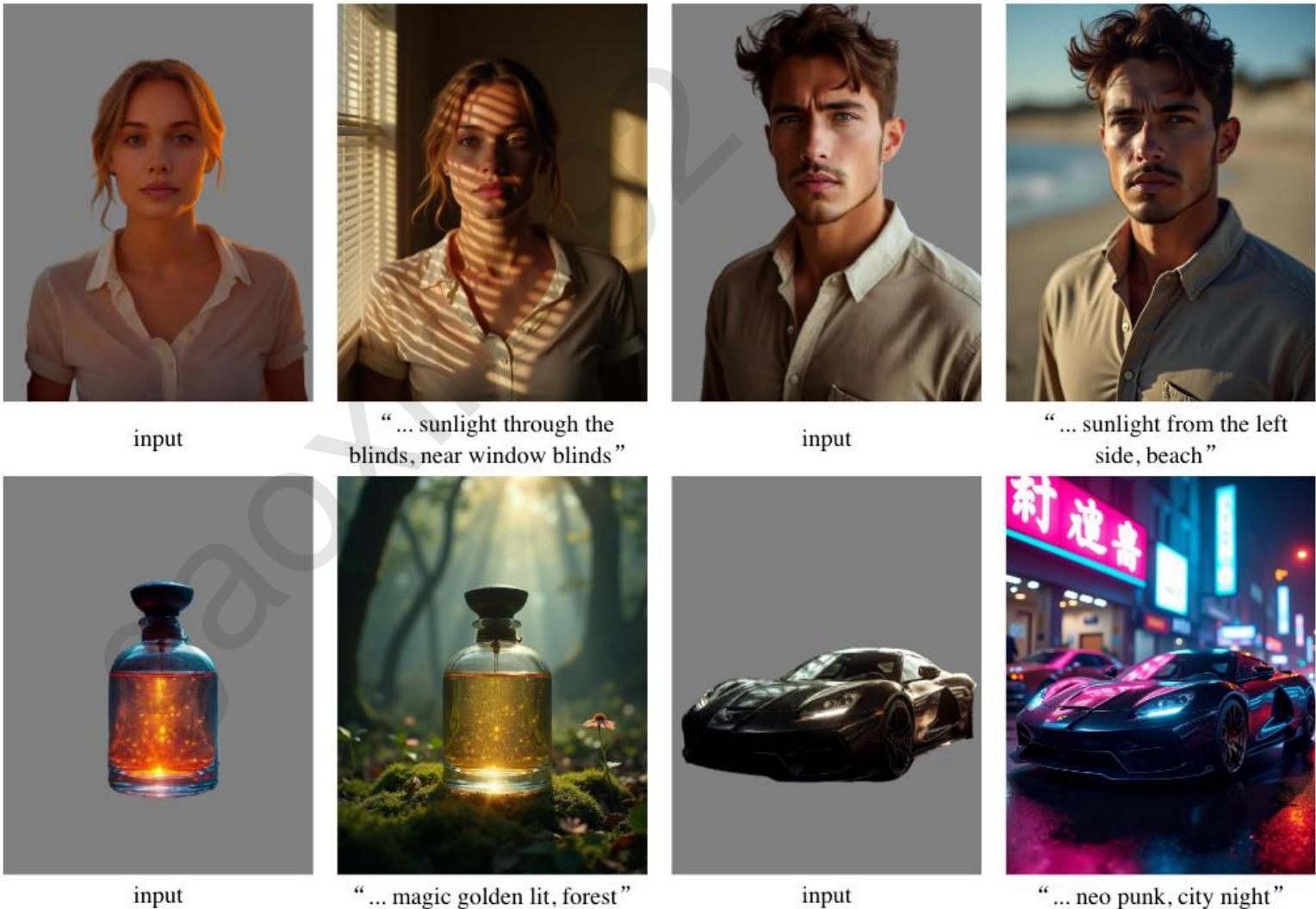
展开 ▾

☆ 保存 引用 被引用次数: 3036 相关文章 所有 6 个版本 ≫

# Illumination harmonization and editing

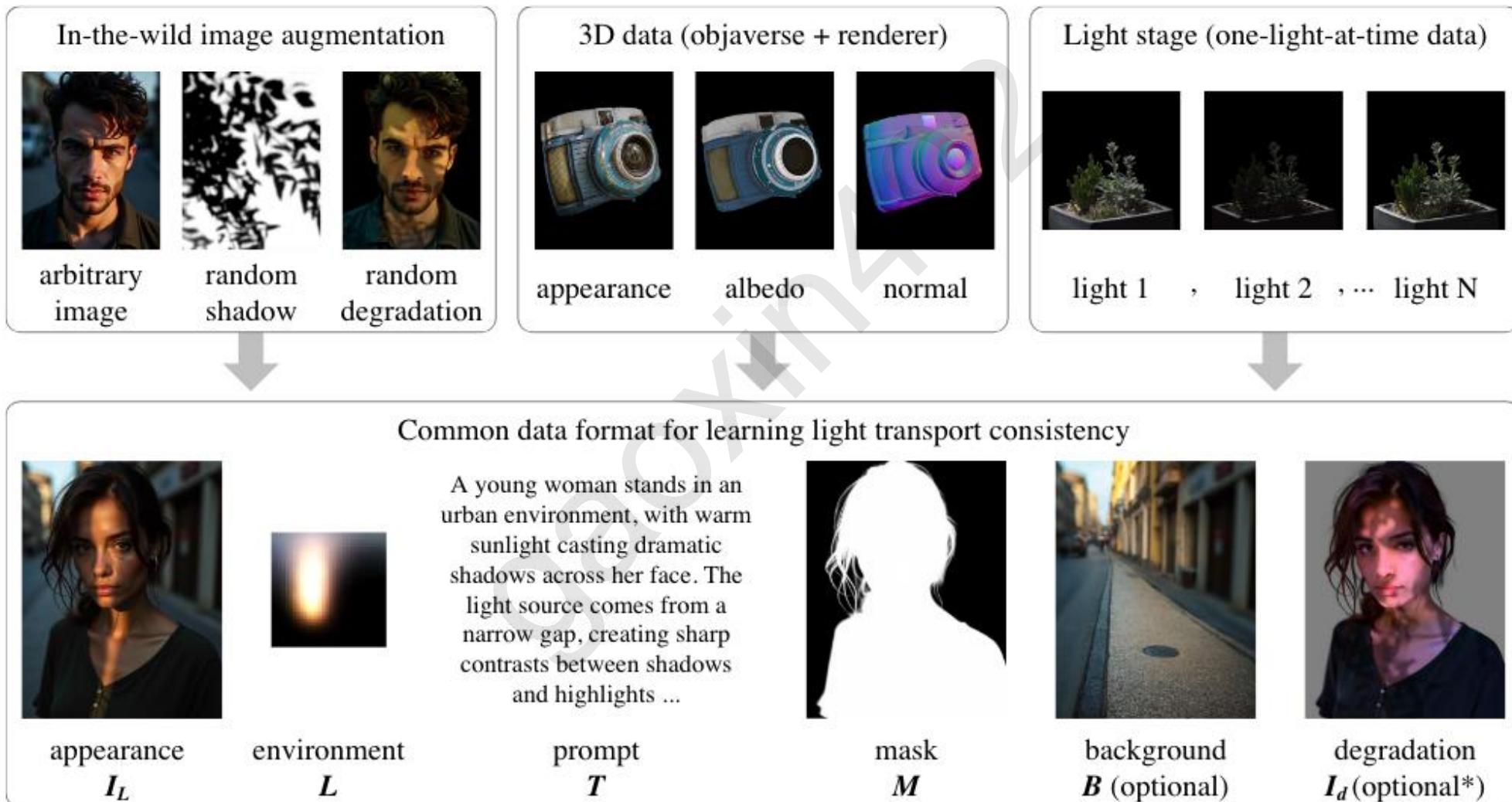
- **Typical Use Case:**  
Users give an object image and illumination description, and our method generates corresponding object appearances and backgrounds.

- **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.

$$\mathcal{L}_{\text{vanilla}} = \|\epsilon - \delta(\varepsilon(\mathbf{I}_L)_t, t, \mathbf{L}, \varepsilon(\mathbf{I}_d))\|_2^2$$

- (a) In computational photography, light transport theory demonstrates that, considering arbitrary appearance  $\mathbf{I}_L^*$  and the correlated environment illumination  $\mathbf{L}$ , a matrix  $\mathbf{T}$  always exists so that

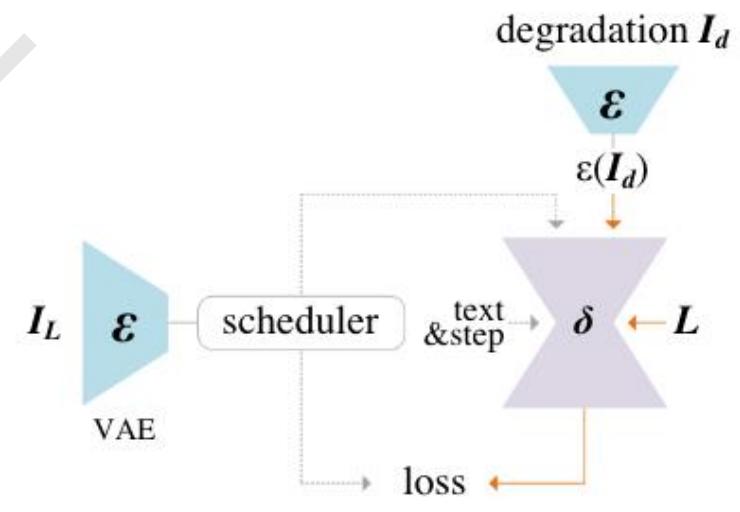
$$\mathbf{I}_L^* = \mathbf{T}\mathbf{L}$$

Because of this linearity, light transport explains appearance merging that

$$\mathbf{I}_{\mathbf{L}_1 + \mathbf{L}_2}^* = \mathbf{T}(\mathbf{L}_1 + \mathbf{L}_2) = \mathbf{I}_{\mathbf{L}_1}^* + \mathbf{I}_{\mathbf{L}_2}^*$$

where  $\mathbf{L}_1, \mathbf{L}_2$  are two arbitrary environment illumination maps.

This intuitively shows that the mixture of an object's appearances under separate illuminations (e.g.,  $\mathbf{L}_1, \mathbf{L}_2$ ) is equivalent to the appearance under merged illumination (e.g.,  $\mathbf{I}_{\mathbf{L}_1 + \mathbf{L}_2}^*$ ).



(a) Vanilla image-conditioned diffusion

# 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?

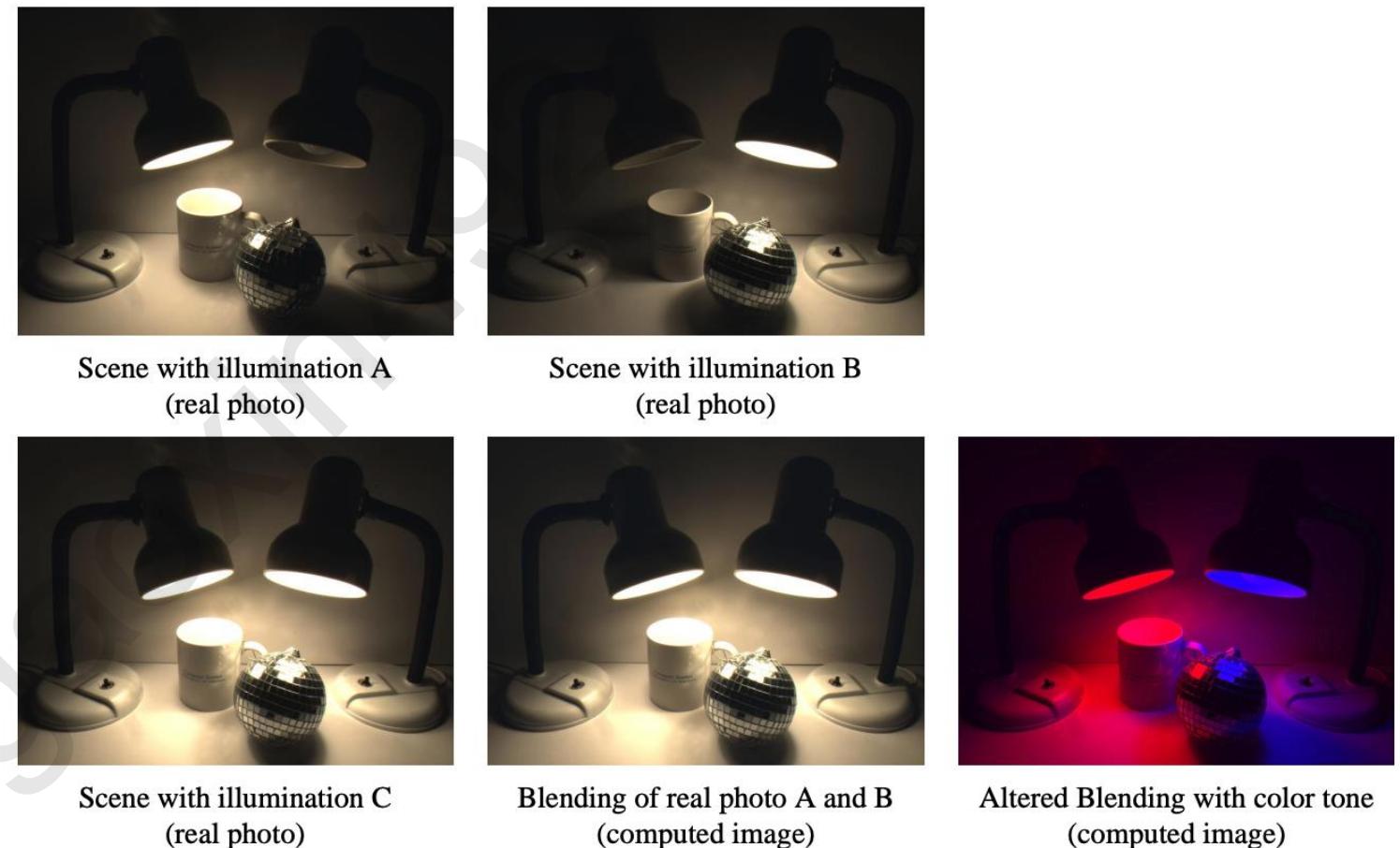


Figure 1: Examples for “the linear blending of an object’s appearances under different illumination conditions is consistent with its appearance under mixed illumination”. Images from OToole (2016).

# Impose Consistent Light

$$\mathbf{I}_{\mathbf{L}_1+\mathbf{L}_2}^* = \mathbf{T}(\mathbf{L}_1 + \mathbf{L}_2) = \mathbf{I}_{\mathbf{L}_1}^* + \mathbf{I}_{\mathbf{L}_2}^* \quad * \text{ 表示 images in raw high-dynamic range}$$

## 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”  $\Rightarrow$  “Estimated Clean image = Noisy Image – Predicted Noise”

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

$$\epsilon_{\mathbf{L}_1+\mathbf{L}_2} = \epsilon_{\mathbf{L}_1} + \epsilon_{\mathbf{L}_2} \quad \Rightarrow \quad \|\epsilon_{\mathbf{L}_1+\mathbf{L}_2} - (\epsilon_{\mathbf{L}_1} + \epsilon_{\mathbf{L}_2})\|_2^2$$

## 2. Latent Space: Linear summation relation => MLP mapping $\phi$

$$\mathcal{L}_{\text{consistency}} = \|M \odot (\epsilon_{\mathbf{L}_1+\mathbf{L}_2} - \phi(\epsilon_{\mathbf{L}_1}, \epsilon_{\mathbf{L}_2}))\|_2^2$$

**Intuition:** Assume mapping  $f$ : latent space  $\rightarrow$  image space

$$f(\epsilon_{\mathbf{L}_1+\mathbf{L}_2}) = f(\epsilon_{\mathbf{L}_1}) + f(\epsilon_{\mathbf{L}_2}) \Rightarrow \epsilon_{\mathbf{L}_1+\mathbf{L}_2} = f^{-1}(f(\epsilon_{\mathbf{L}_1}) + f(\epsilon_{\mathbf{L}_2})) \Rightarrow \epsilon_{\mathbf{L}_1+\mathbf{L}_2} = \phi(\epsilon_{\mathbf{L}_1}, \epsilon_{\mathbf{L}_2})$$

## 3. Implementation of environment illumination maps

# Impose Consistent Light

$$\mathcal{L}_{\text{consistency}} = \|\mathbf{M} \odot (\epsilon - \phi(\delta(\varepsilon(\mathbf{I}_{L_1})_t, t, \mathbf{L}_1, \varepsilon(\mathbf{I}_d))), \delta(\varepsilon(\mathbf{I}_{L_2})_t, t, \mathbf{L}_2, \varepsilon(\mathbf{I}_d)))\|_2^2$$

**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  $L = L_1 + L_2$

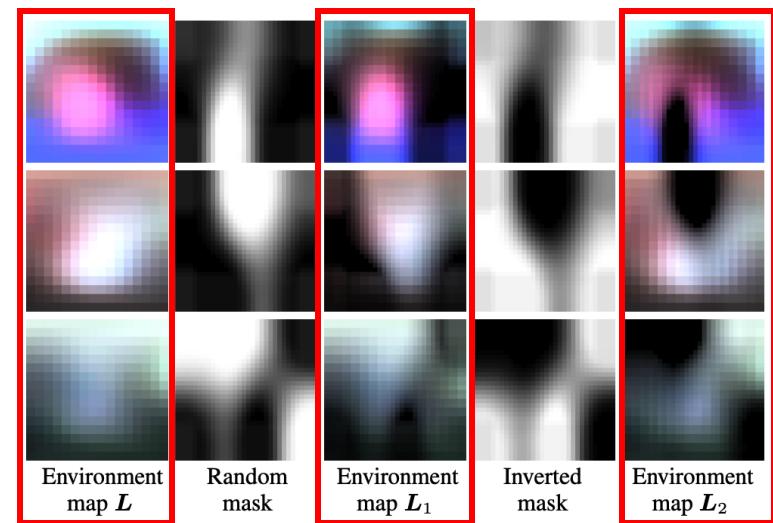
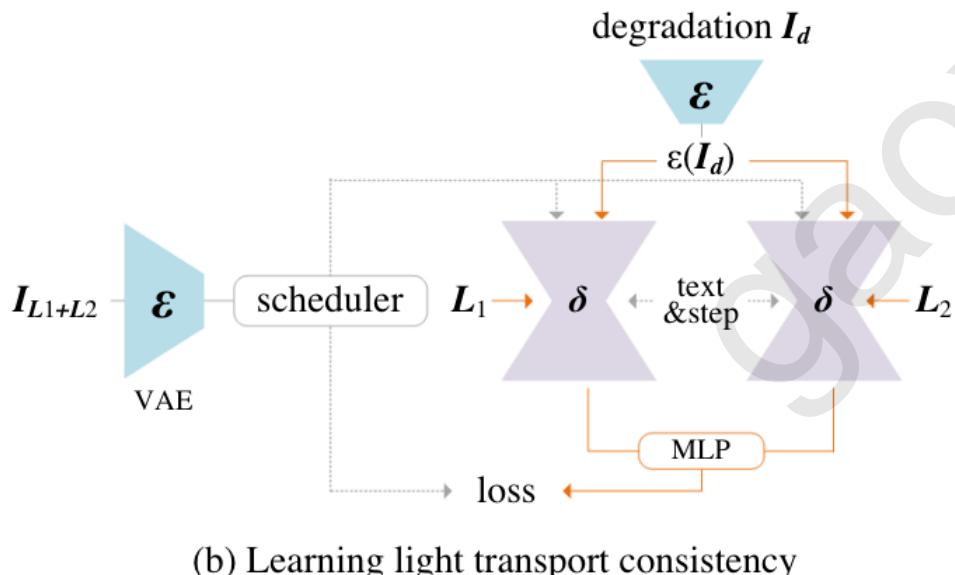


Figure 4: Examples of decomposed environment maps. We present examples to use random masks to decompose environment map  $L$  into  $L_1$  and  $L_2$ . Note that  $L = L_1 + L_2$ . A typical full environment map is usually of ratio 2:1, with size  $64 \times 32$  when convoluted. We use the front half (facing the image) of the convoluted environment map, which is  $32 \times 32$ . 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 halves 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  $\odot$  Foreground Mask),  
Illumination maps + Text Prompt

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

| Method          | PSNR $\uparrow$ | SSIM $\uparrow$ | LPIPS $\downarrow$ |
|-----------------|-----------------|-----------------|--------------------|
| 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             |

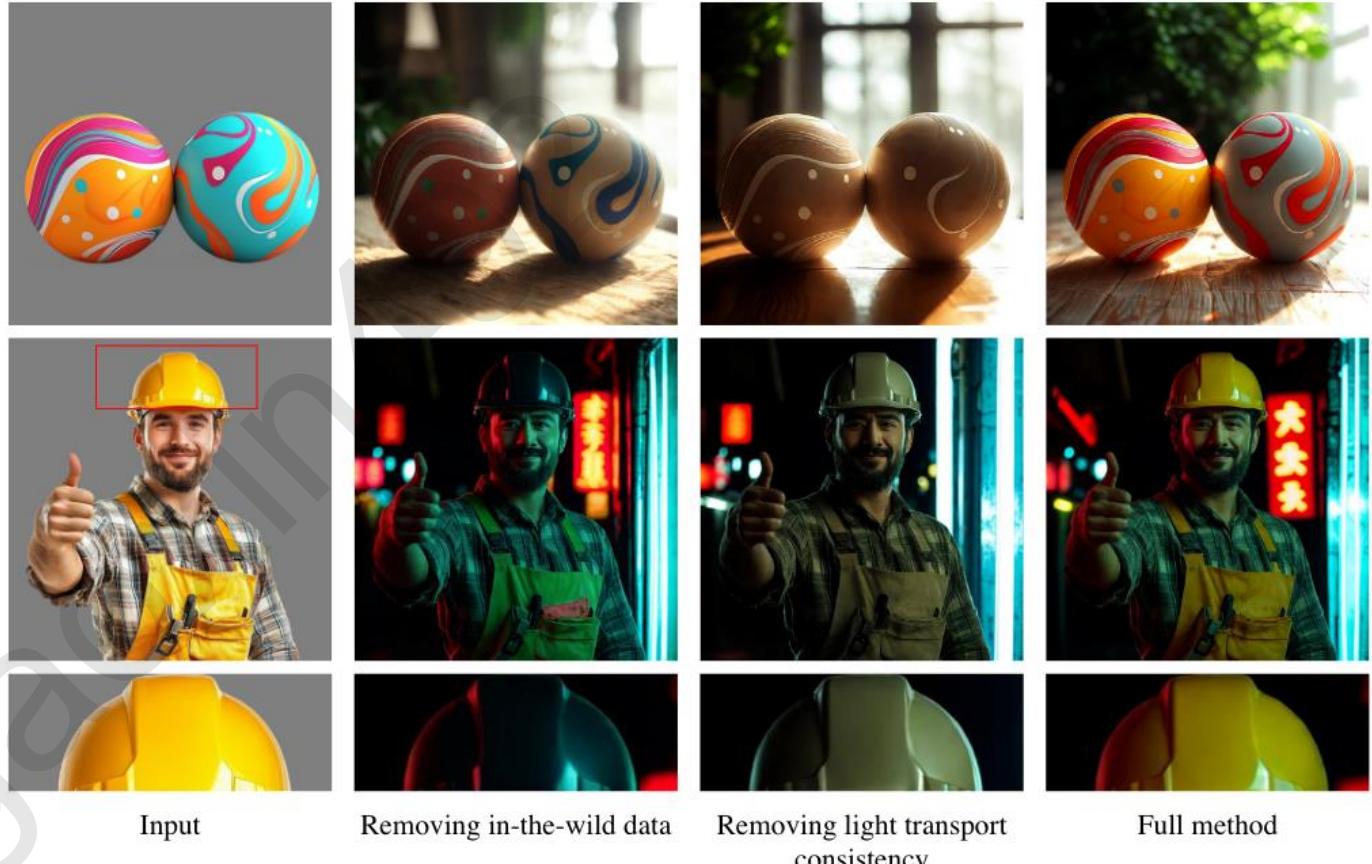
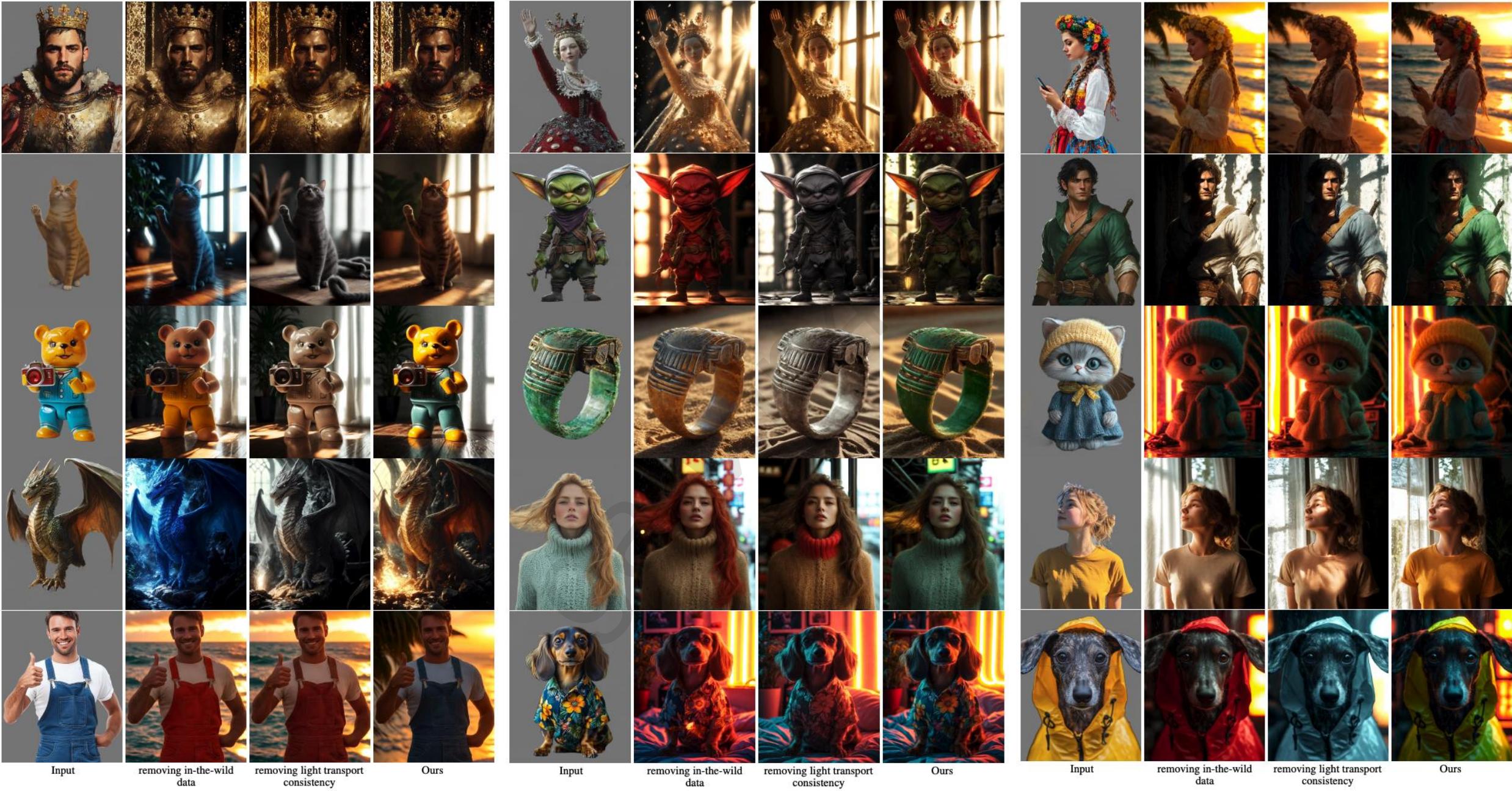


Figure 4: **Ablative Study.** We present results by removing the light transport consistency or in the wild data. More results are in the supplementary material. Results in this figure are from Stable Diffusion 1.5 version of our model. Prompts are “toy in room, studio lighting”, and “a handsome man,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



# Additional Application

- **Background-conditioned Model**

- ① **Training:**

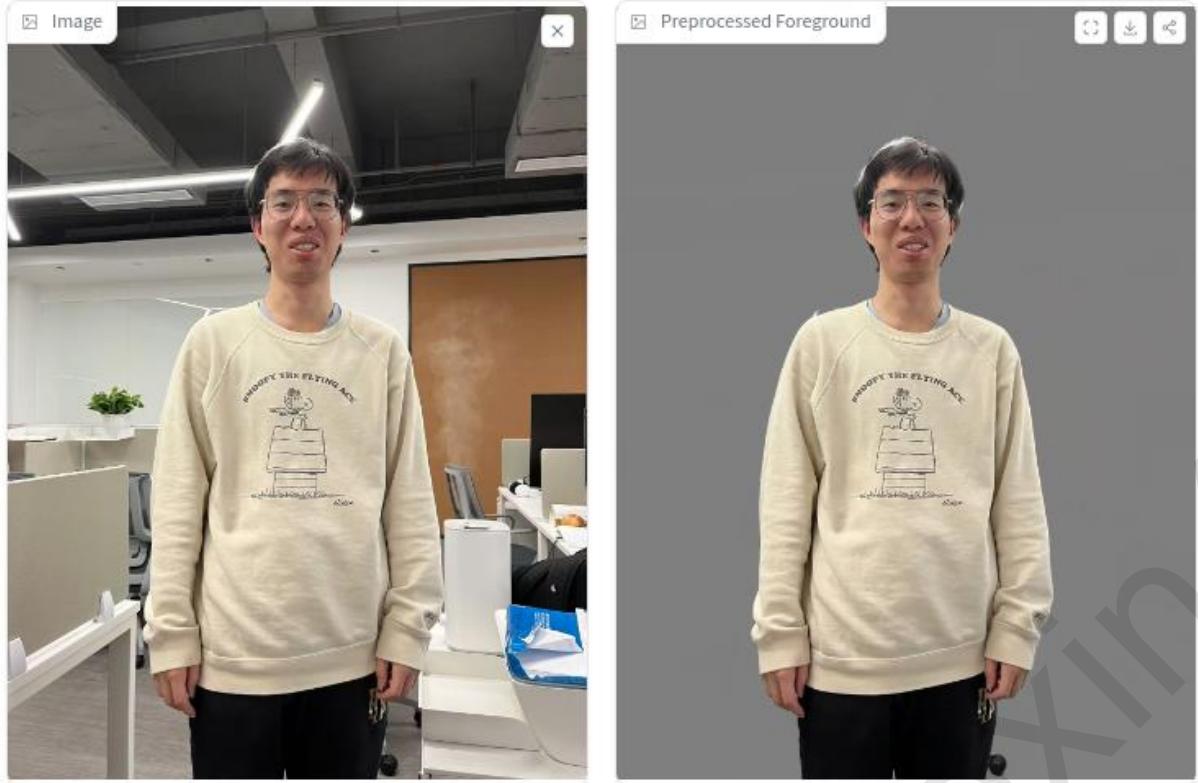
“Besides, to train background-conditioned model, we concatenate  $B$  to  $I_d$  (and fill the extra channel with all zeros if some part of the dataset do not have backgrounds).”

- ② **Inference:**

(Image  $\odot$  Foreground Mask), Background conditions

- Alternative base diffusion models  
SD 1.5, SDXL, Flux
- Normal Estimation (Omitted)





Prompt

vintage photograph of a woman. sunshine from window.

## Normal case

Initial Latent

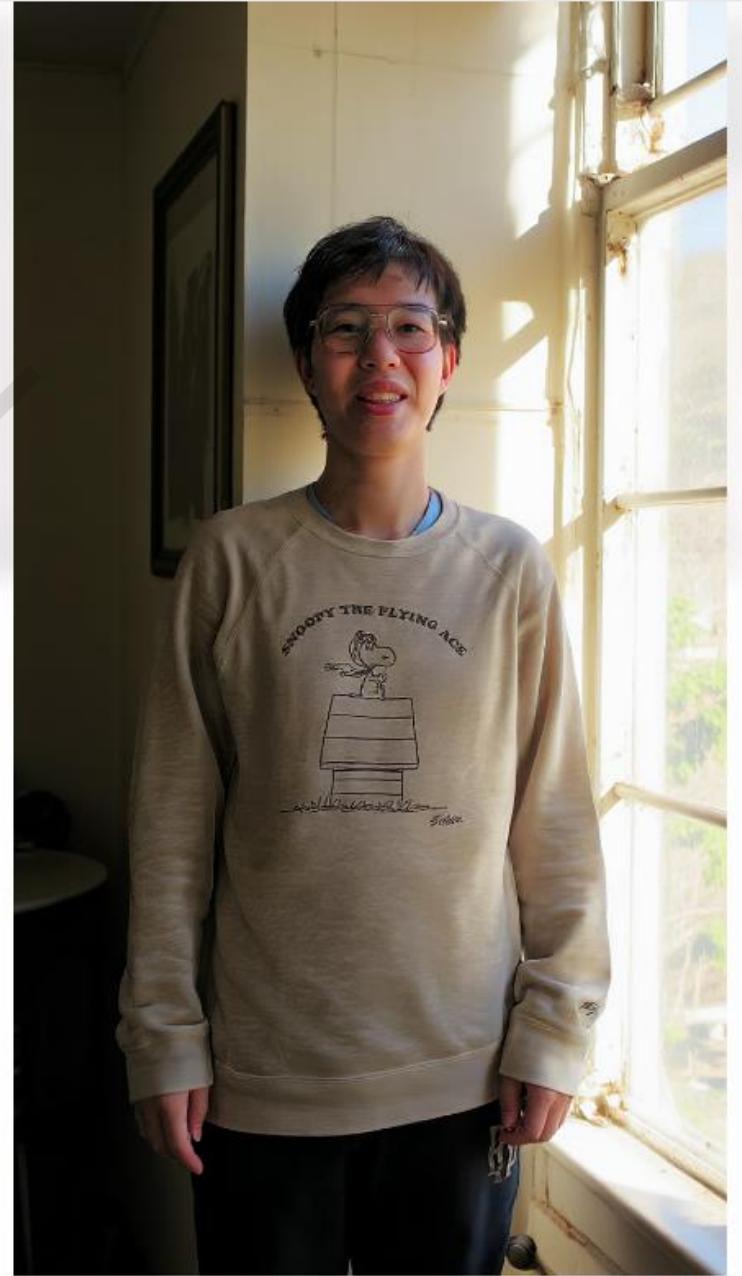
- None
- Left Light
- Right Light
- Top Light
- Bottom Light

Prefix Quick List

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

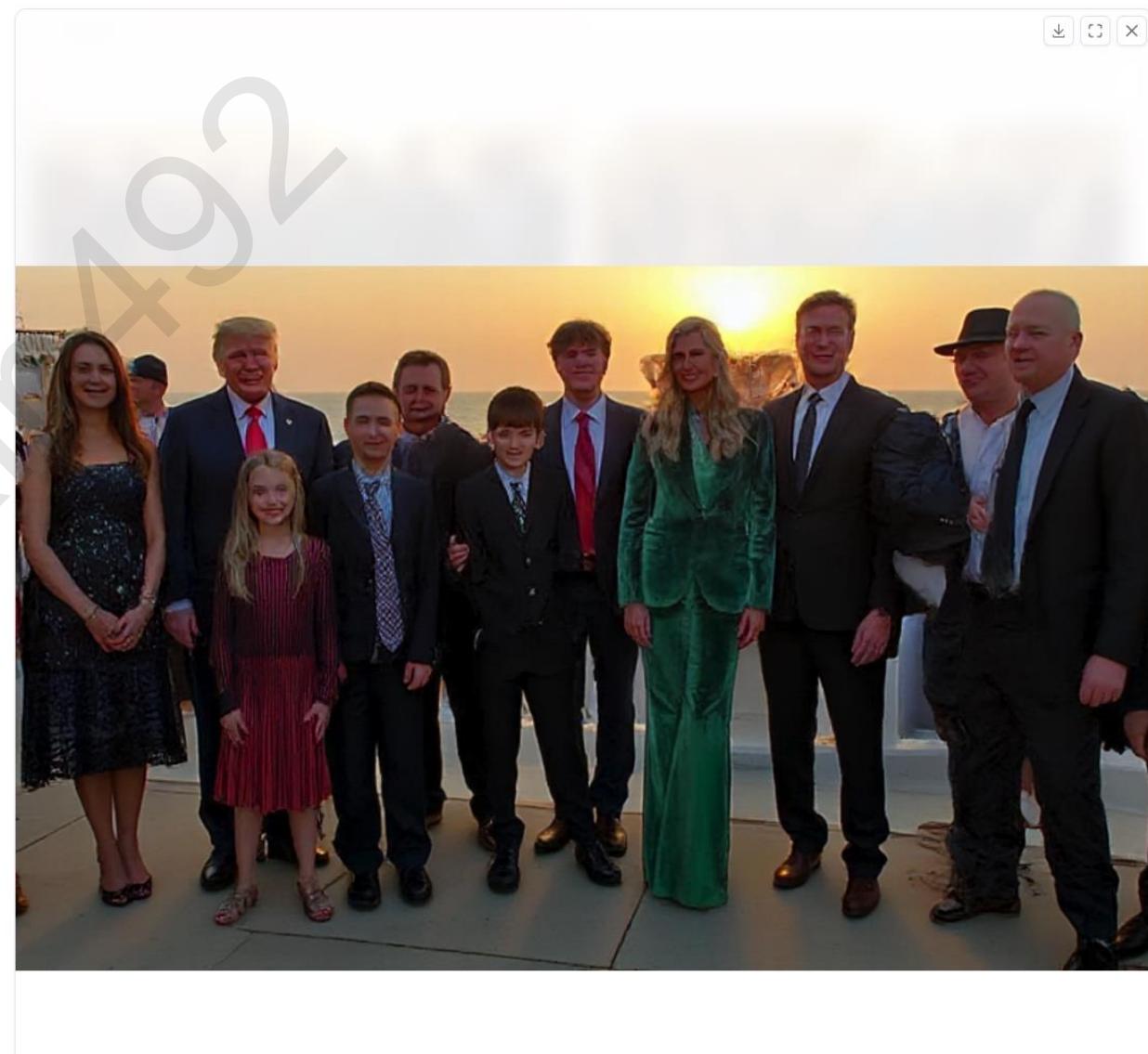
Subject Quick List

- 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/llyasviel/IC-Light/discussions/98>



### Prompt

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

### Initial Latent

None

Left Light

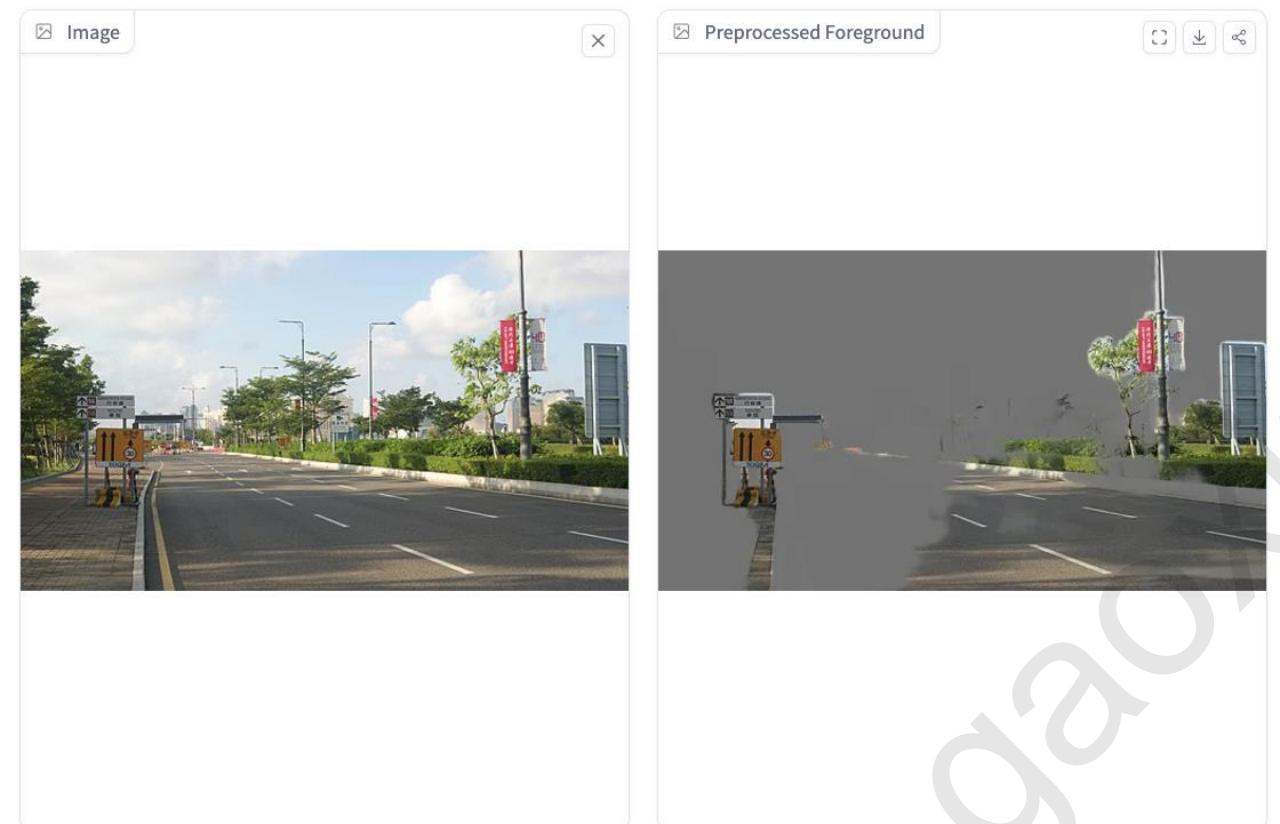
Right Light

Top Light

Bottom Light

## IC-Light V2

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



### Prompt

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

### Initial Latent

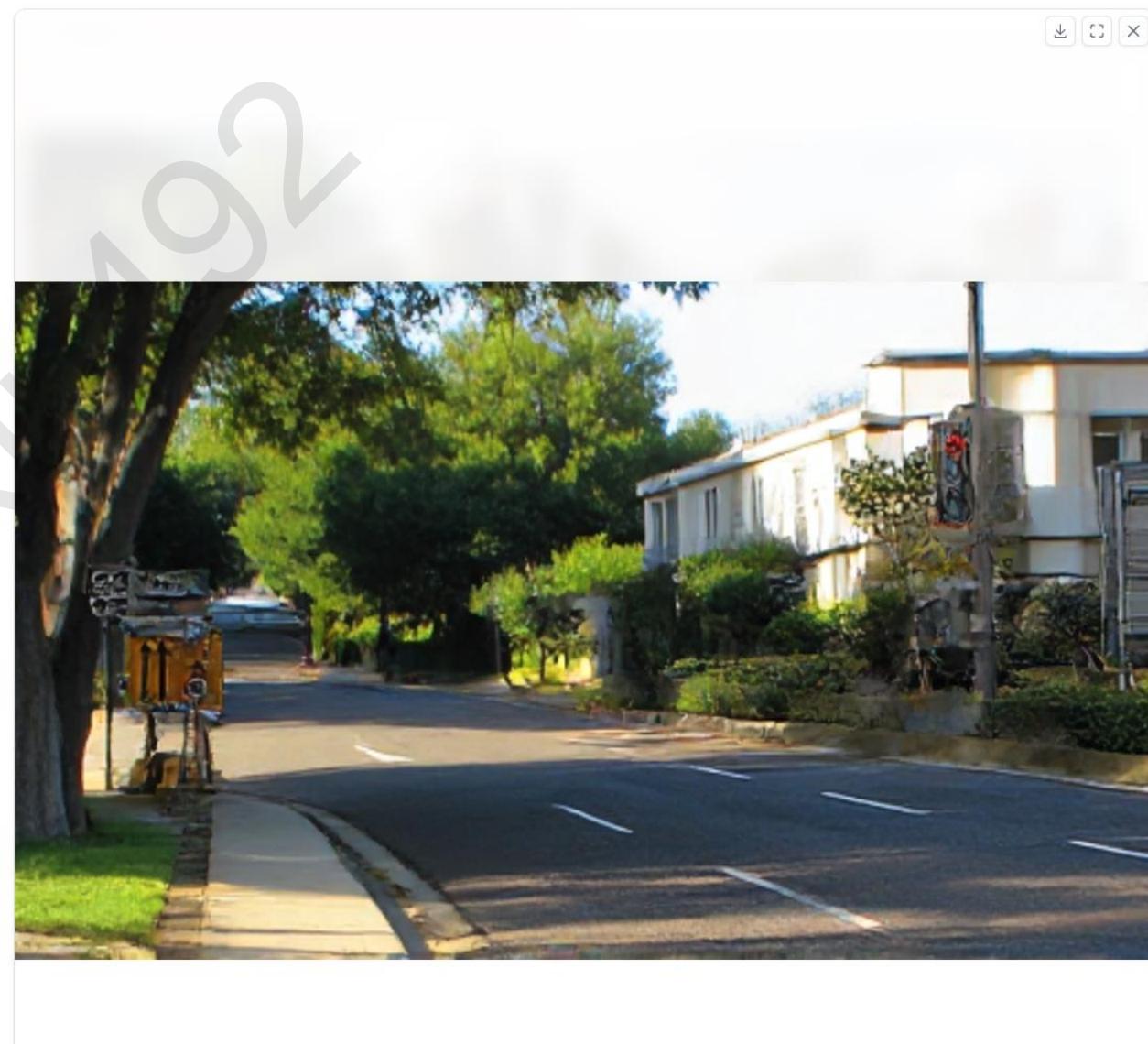
None

Left Light

Right Light

Top Light

Bottom Light



前景比较弱化的 case

# 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://github.com/Fafa-DL/Lhy_Machine_Learning)

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

<https://jalamar.github.io/illustrated-stable-diffusion/>