Denoising Diffusion Probabilistic Models (DDPM)

Outline

- Overview
- Probabilistic Modeling
- Building upon DDPM
  - Classifier Guidance and Classifier-Free Guidance
  - Latent Diffusion

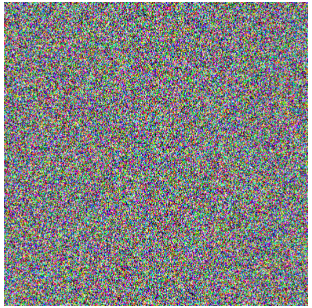
Overview

- DDPM is an image-generation model
  - *Generating* images
  - *Sampling* images from a simple prior
  - *Modeling* the distribution of the data X of interest
  - *Computing* the probability of data p(x), x is an image
- In contrast to:
  - Discriminative models that models p(y | x)

Training: Diffusion to Get the Input

1. Sample an image x from the training set
2. Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
3. Sample a variance $\beta \in (0, 1)$
4. Get the input $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

image: x



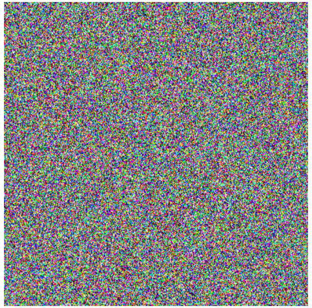
noise: ϵ

input: $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

Training: Diffusion to Get the Input

1. Sample an image x from the training set
2. Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
3. Sample a variance $\beta \in (0, 1)$
4. Get the input $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

image: x



noise: ϵ

input: $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

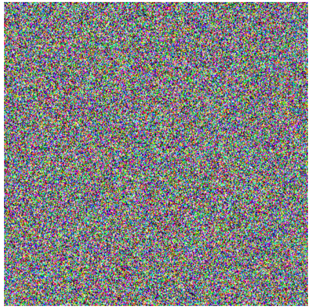
Training: Diffusion to Get the Input

1. Sample an image x from the training set
2. Sample noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
3. Sample a variance $\beta \in (0, 1)$
4. Get the input $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

merge two Gaussians $\mathcal{N}_1(0, \sigma_1^2)$, $\mathcal{N}_2(0, \sigma_2^2)$, the new distribution is $\mathcal{N}(0, \sigma_1^2 + \sigma_2^2)$

← keep the variance unchanged

image: x



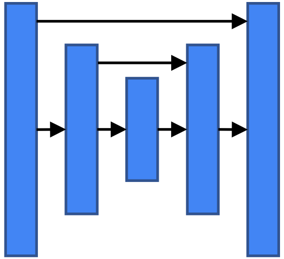
noise: ϵ

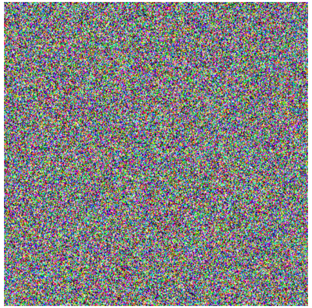
input: $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

Training: Denoising to Get the Output

1. Feed the input into the network
2. Network: U-Net f_θ
3. Predict the noise $f_\theta(x_t, t)$
4. Loss is MSE

input: $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$





output: $\tilde{\epsilon}$

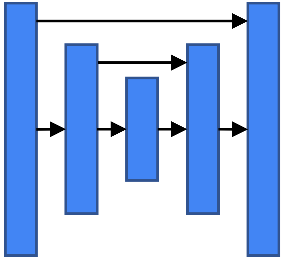
output: $\tilde{x} = \frac{x_t - \sqrt{\beta}\tilde{\epsilon}}{\sqrt{1-\beta}}$

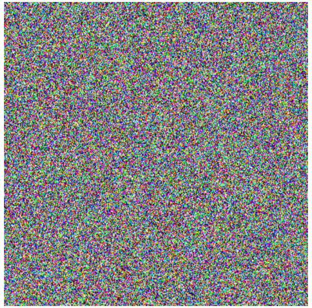
Training: Denoising to Get the Output

1. Feed the input into the network
2. Network: U-Net f_θ
3. Predict the noise $f_\theta(x_t, t)$
4. Loss is MSE

recall #1
Denoising Autoencoder

input: $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$

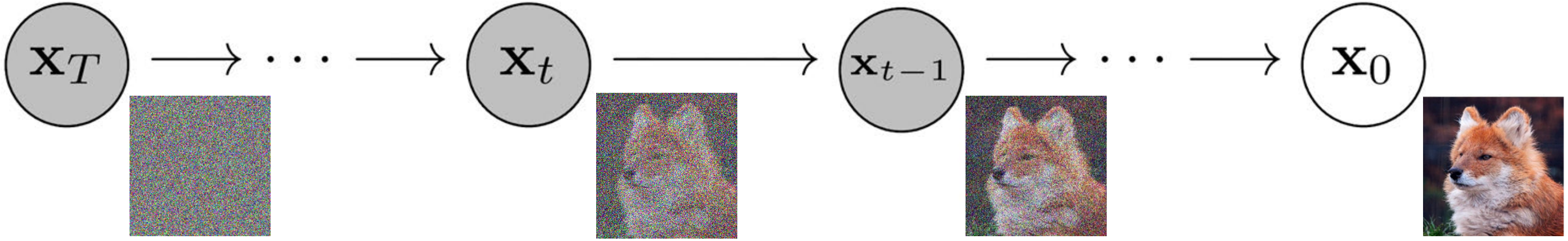




output: $\tilde{\epsilon}$

output: $\tilde{x} = \frac{x_t - \sqrt{\beta}\tilde{\epsilon}}{\sqrt{1-\beta}}$

Inference: Iterative Denoising

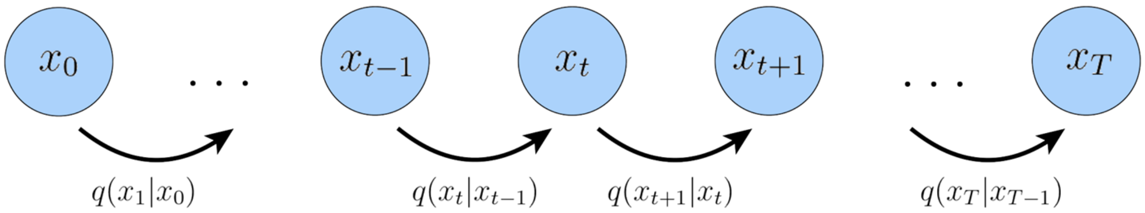


$x_T \sim \mathcal{N}(0, \mathbf{I})$

Outline

- Overview
- Probabilistic Modeling
- Building upon DDPM
  - Classifier Guidance and Classifier-Free Guidance
  - Latent Diffusion

Forward Pass (Diffusion Pass)



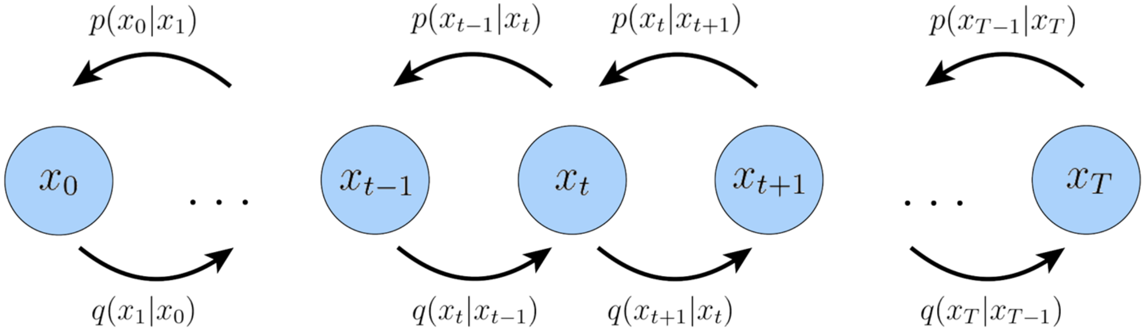
Define a Markov process that gradually adds Gaussian noise to images

$$q(x_t|x_{t-1}) = \mathcal{N}\left(\mu = \sqrt{1-\beta_t}x_{t-1}, \Sigma = \beta_t\mathbf{I}\right)$$

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

- There are T such distributions: $q(x_1|x_0), \ldots, q(x_T|x_{T-1})$
- $\beta_{1:T}$ is a variance schedule
- $\beta_{1:T}$ can be predefined or learned
- β_t are all very small: *"gradually"*
- x_T is close to pure noise, ie., $\mathcal{N}(0, \mathbf{I})$

Reverse Pass (Generation Pass)



Approximate the reverse model $q(x_{t-1}|x_t)$ with a deep net θ

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu = \mu_\theta(x_t, t), \Sigma = \Sigma_\theta(x_t, t)\mathbf{I})$$

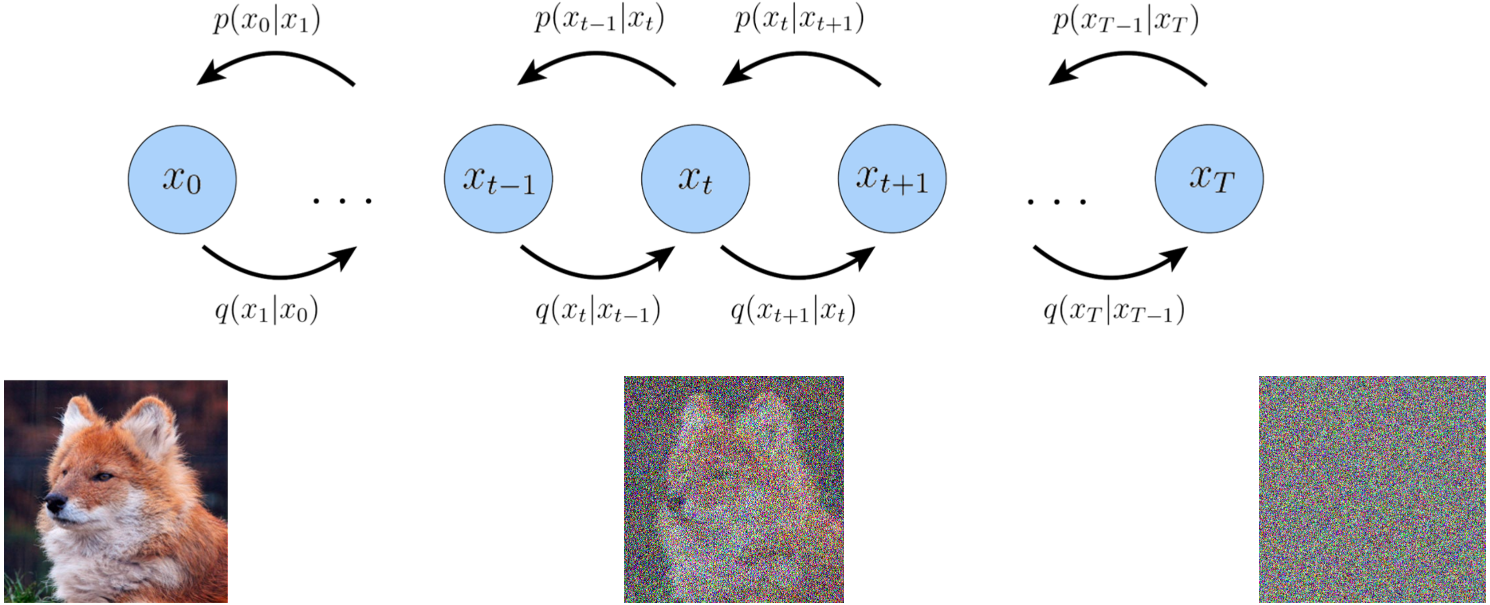
- $q(x_{t-1}|x_t)$ is the real distribution (ground-truth), but intractable
- $p_\theta(x_{t-1}|x_t)$ is our approximation (our model)
- $p_\theta(x_{t-1}|x_t)$ is modeled as Gaussian, for simple optimization
- Model $p_\theta(x_{t-1}|x_t)$ as Gaussian *works* only when β_t are small
  - For the not-working situation, recall how blur the MAE's reconstructions are

Forward and Reverse Pass

recall #2

Autoregressive Models

$$p(x_0) = p(x_T)p(x_{T-1} \mid x_T)p(x_{T-2} \mid x_{T-1}, x_T) \ldots$$

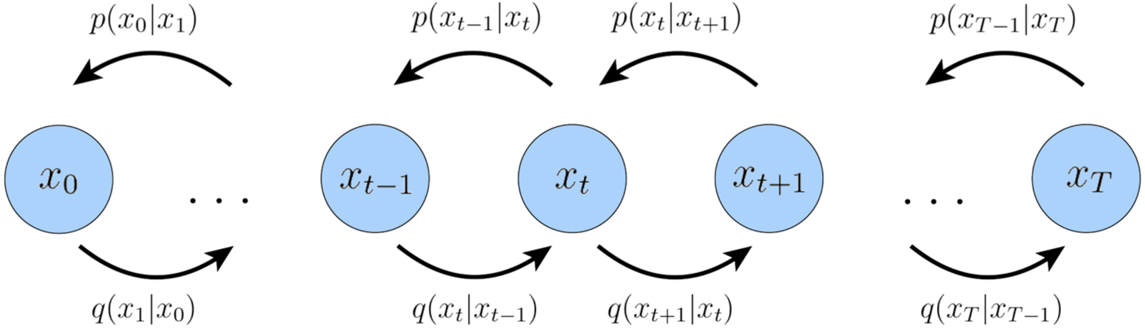


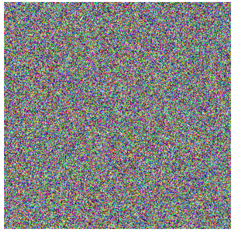
Forward and Reverse Pass

recall #3

(Hierarchical) Variational Autoencoder

x_t as latent variables





Optimization: Variational Lower Bound

$$
\begin{aligned}
-\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{\mathrm{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \| p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T}\sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)}\Big] \\
&= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0)\Big] \\
&= \mathbb{E}_q\Big[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\Big] \\
&= \mathbb{E}_q\Big[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T)\prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\Big] \\
&= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big]
\end{aligned}
$$

image source: blog
original paper: Sohl-Dickstein et al., 2015

$$= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log\Big(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\cdot\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}\Big) + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big]$$

$$= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big]$$

$$= \mathbb{E}_q\Big[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log\frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log\frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}\Big]$$

$$= \mathbb{E}_q\Big[\log\frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^{T}\log\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Big]$$

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

$$L_{\text{VLB}} = L_T + L_{T-1} + \cdots + L_0$$

$$\text{where } L_T = D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_T))$$

$$L_t = D_{\text{KL}}(q(\mathbf{x}_t|\mathbf{x}_{t+1},\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})) \text{ for } 1 \le t \le T-1$$

$$L_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

image source: blog
original paper: Sohl-Dickstein et al., 2015

- $$L_t = D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \,||\, p_\theta(x_{t-1}|x_t))$$

- $q(x_{t-1}|\, x_t, x_0)$
    - $q(x_{t-1}|\, x_t)$ is intractable
    - $q(x_{t-1}|\, x_t, x_0)$, however, is tractable

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

$$\tilde{\beta}_t = 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = 1/(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

$$
\begin{aligned}
\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) \\
&= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0
\end{aligned}
$$

$$
\begin{aligned}
\tilde{\boldsymbol{\mu}}_t &= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t) \\
&= \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\Big)
\end{aligned}
$$

image source: blog
original paper: Sohl-Dickstein et al., 2015

- $$L_t = D_{\mathrm{KL}}\big(q(x_{t-1}|x_t, x_0) \,||\, p_\theta(x_{t-1}|x_t)\big)$$

- $q(x_{t-1}|\, x_t, x_0)$
  - $q(x_{t-1}|\, x_t)$ is intractable
  - $q(x_{t-1}|\, x_t, x_0)$, however, is tractable, and a Gaussian

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \quad \tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\Big)$$

image source: blog
original paper: Sohl-Dickstein et al., 2015

- $L_t = D_{\mathrm{KL}}\left(q(x_{t-1}|x_t, x_0)\,||\,p_\theta(x_{t-1}|x_t)\right)$

- $q(x_{t-1}|\,x_t, x_0)$
  - $q(x_{t-1}|\,x_t)$ is intractable
  - $q(x_{t-1}|\,x_t, x_0)$, however, is tractable, and a Gaussian

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I}), \quad \tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$$

- $p_\theta(x_{t-1}|x_t)$
  - $p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_{t-1} = \mu_\theta(x_t, t), \Sigma_{t-1} = \Sigma_\theta(x_t, t)\mathbf{I})$, Gaussian
  - Reparametrize $\mu_\theta(x_t, t)$ to a similar format:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right)$$

image source: blog
original paper: Sohl-Dickstein et al., 2015

- $$L_t = D_{\mathrm{KL}}\big(q(x_{t-1}|x_t, x_0) \,||\, p_\theta(x_{t-1}|x_t)\big)$$

  - $q(x_{t-1}|\, x_t, x_0)$, Gaussians
  - $p_\theta(x_{t-1}|x_t)$, Gaussians
  - KL between two Gaussians has a closed form

$$L_t = \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2\Big]$$

$$= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\Big]$$

  - Assume backward process variances $\boldsymbol{\Sigma}_\theta$ are constants $\sigma_t^2 \boldsymbol{I}$
  - Simplify the term by ignoring the weighting term

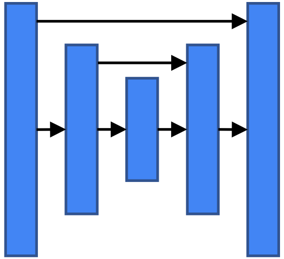
$$L_t^{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\Big[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\Big]$$

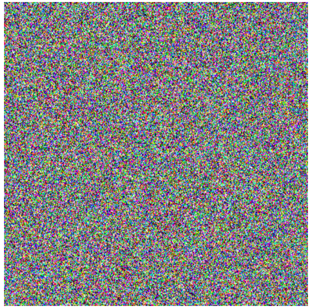
image source: blog
original paper: Ho et al. (2020)

Training: Denoising to Get the Output

1. Feed the input into the network
2. Network: U-Net f_θ
3. Predict the noise $f_\theta(x_t, t)$
4. Loss is MSE

input: $x_t = \sqrt{1-\beta}x + \sqrt{\beta}\epsilon$





output: $\tilde{\epsilon}$

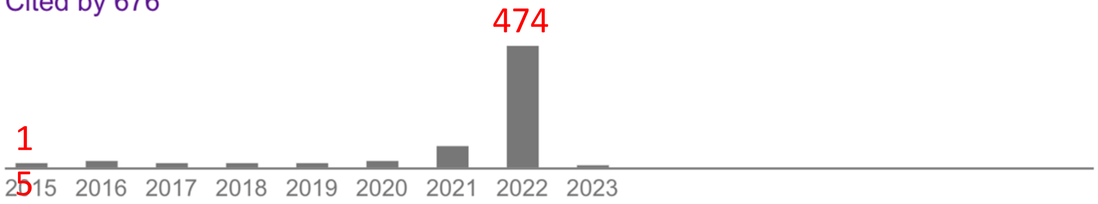
output: $\tilde{x} = \frac{x_t - \sqrt{\beta}\tilde{\epsilon}}{\sqrt{1-\beta}}$

Two theories, One approach

- Variational Lower Bound
  - Sohl-Dickstein et al, ICML 2015 – “Diffusion”
  - Ho et al, NeurIPS 2020
- Denoising Score Matching
  - Song and Ermon, NeurIPS 2019

Deep unsupervised learning using nonequilibrium thermodynamics

Authors	Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, Surya Ganguli
Publication date	2015/3/12
Journal	International Conference on Machine Learning
Description	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.
Total citations	Cited by 676



Two theories, One approach

- Variational Lower Bound
  - Sohl-Dickstein et al, ICML 2015 – “Diffusion”
  - Ho et al, NeurIPS 2020 - DDPM

- Denoising Score Matching
  - Song and Ermon, NeurIPS 2019

Two theories, One approach

- Variational Lower Bound
  - Sohl-Dickstein et al, ICML 2015 – “Diffusion”
  - Ho et al, NeurIPS 2020 - DDPM
- Denoising Score Matching
  - Song and Ermon, NeurIPS 2019

<u>Google</u> (Jonathan Ho)
DDPM: Ho et al, NeurIPS 2020
CFG: Ho et al, 2022
Imagen: Saharia et al, 2022
Imagen video: Ho et al, 2022

<u>OpenAI</u> (Nichol and Dhariwal):
Nichol and Dhariwal, ICML 2021
Dhariwal and Nichol, NeurIPS 2021
GLIDE: Nichol et al, NeurIPS 2020
DALL-E 2: Ramesh et al, 2022

Two theories, One approach

- Variational Lower Bound
  - Sohl-Dickstein et al, ICML 2015 – "Diffusion"
  - Ho et al, NeurIPS 2020 - DDPM

- Denoising Score Matching
  - Song and Ermon, NeurIPS 2019 – "NCSN", Langevin dynamics

Why is DDPM Taking Over?

- An image-to-image formulation: U-Net + Transformer
- Stable training: MSE
- Can be analytically evaluated
- Able to fit large-scale, complex dataset
- ……

Outline

- Overview
- Probabilistic Modeling
- Building upon DDPM
    - Classifier Guidance and Classifier-Free Guidance
    - Latent Diffusion

Classifier Guidance

Applied to class-conditional generation $p(x \mid c)$

- Train a classifier $f_\phi(c \mid x_t)$
- During sampling, update from x_t to x_{t-1} with gradients:

$$\tilde{\epsilon}_\theta = \epsilon_\theta - \omega \nabla_x f_\phi(c|x_t)$$

- ω is a hyper-parameter to control the strength of the guidance
- adversarial attack?

Dhariwal and Nichol, “Diffusion Models Beat GANs on Image Synthesis”, NeurIPS 2021

Classifier Guidance

Applied to class-conditional generation $p(x \mid c)$

- Train a classifier $f_\phi(c \mid x_t)$: What if text-conditioned?

  A: CLIP guidance $f_\phi(t \mid x_t)$

  B: Classifier-Free Guidance

Nichol et al, "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models", ICML 2022

Classifier-Free Guidance

$$\tilde{\epsilon}_\theta(x_t \mid t) = \epsilon_\theta(x_t \mid \emptyset) + \backslash\mathrm{omega}(\epsilon_\theta(x_t \mid t) - \epsilon_\theta(x_t \mid \emptyset))$$

- Each step involves two inference
    - $\epsilon_\theta(x_t \mid \emptyset)$, unconditional
    - $\epsilon_\theta(x_t \mid t)$, condition on the text t
    - Prediction is a linear combination of the two above

Ho and Salisman, “Classifier-Free Diffusion Guidance”, NeurIPS Workshop

Latent Diffusion

- Diffusion in the latent space instead of pixel space
- Latent space, or feature space of a prefixed autoencoder
- Diffusion in the feature space
    - Diffusion models $p(x)$
    - x can be data
    - or other structured high-dimensional, continuous distribution
    - eg. features and weights
- Lower training cost and faster inference speed
- THE model behind stable diffusion

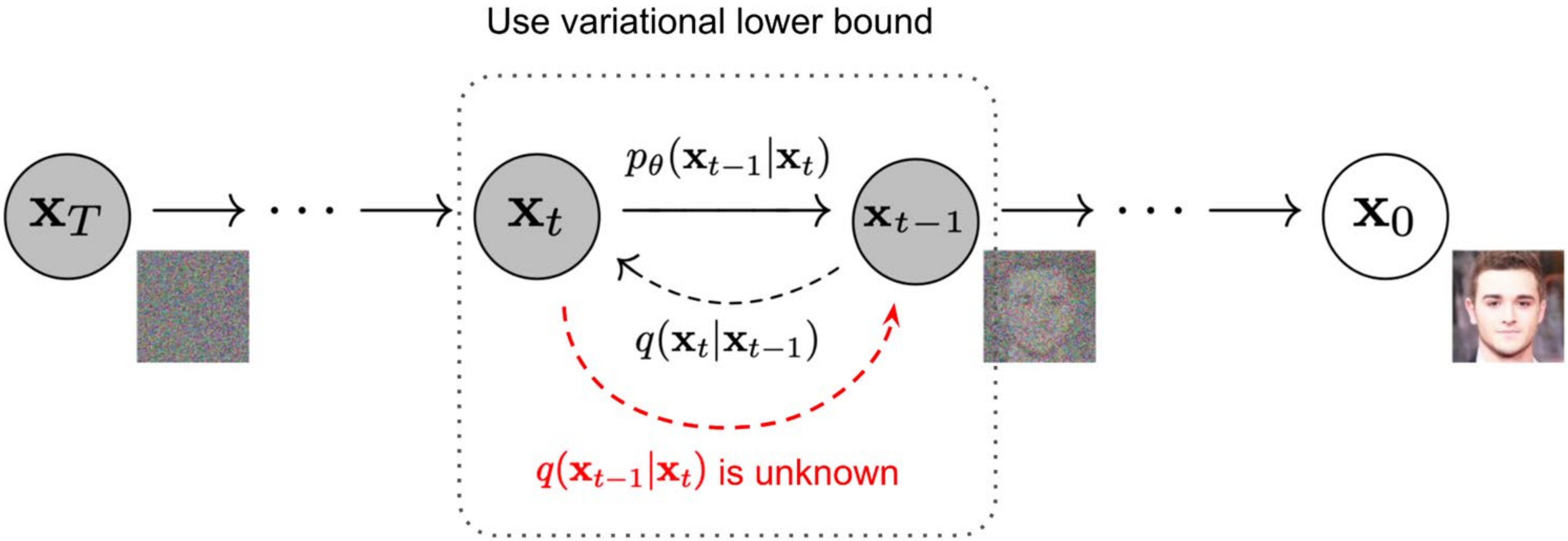
What I didn’t mention

- Faster sampling (DDIM)
- Inversion, interpolation and image editing
- Progress in architecture design
- Text-to-image generation
- Scaling up model, data, resolution
- 2D-to-3D generation
- 3D model generation
- Diffusion model for recognition
- ……

Recent Progresses on Diffusion Models

Diffusion Models for Image Generation

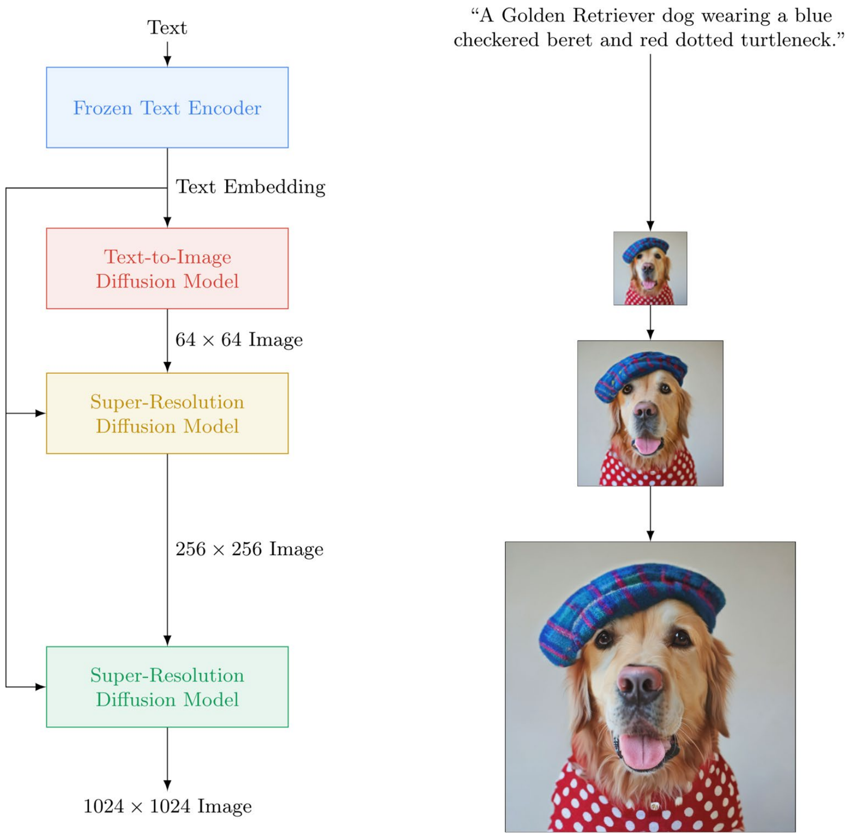
P(X)

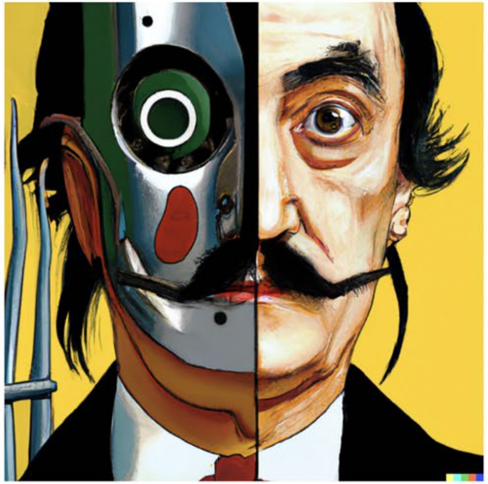


Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models.", NeurIPS 2020

Text-to-Image Diffusion Models

P(X | T)





vibrant portrait painting of Salvador Dalí with a robotic half face

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

DALL-E 2

Compositionality

An astronaut riding a horse in photorealistic style.

Compositionality

An astronaut riding a horse in photorealistic style.

Compositionality

A dog looking curiously in the mirror, seeing a cat.

Compositionality

A majestic oil painting of a raccoon Queen wearing red French royal gown. The painting is hanging on an ornate wall decorated with wallpaper.

However

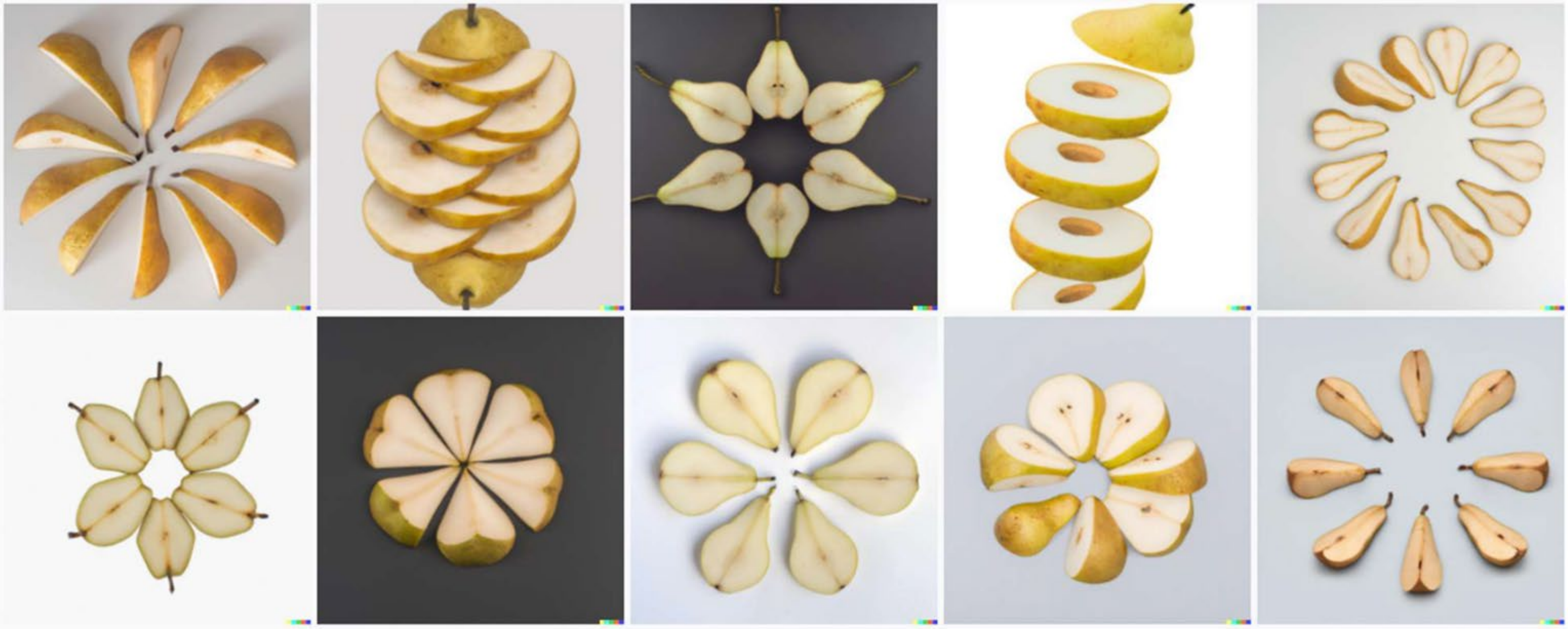
Bad at spatial positions:

A red ball on top of a blue pyramid with the pyramid behind a car that is above a toaster.

However

Bad at counting:

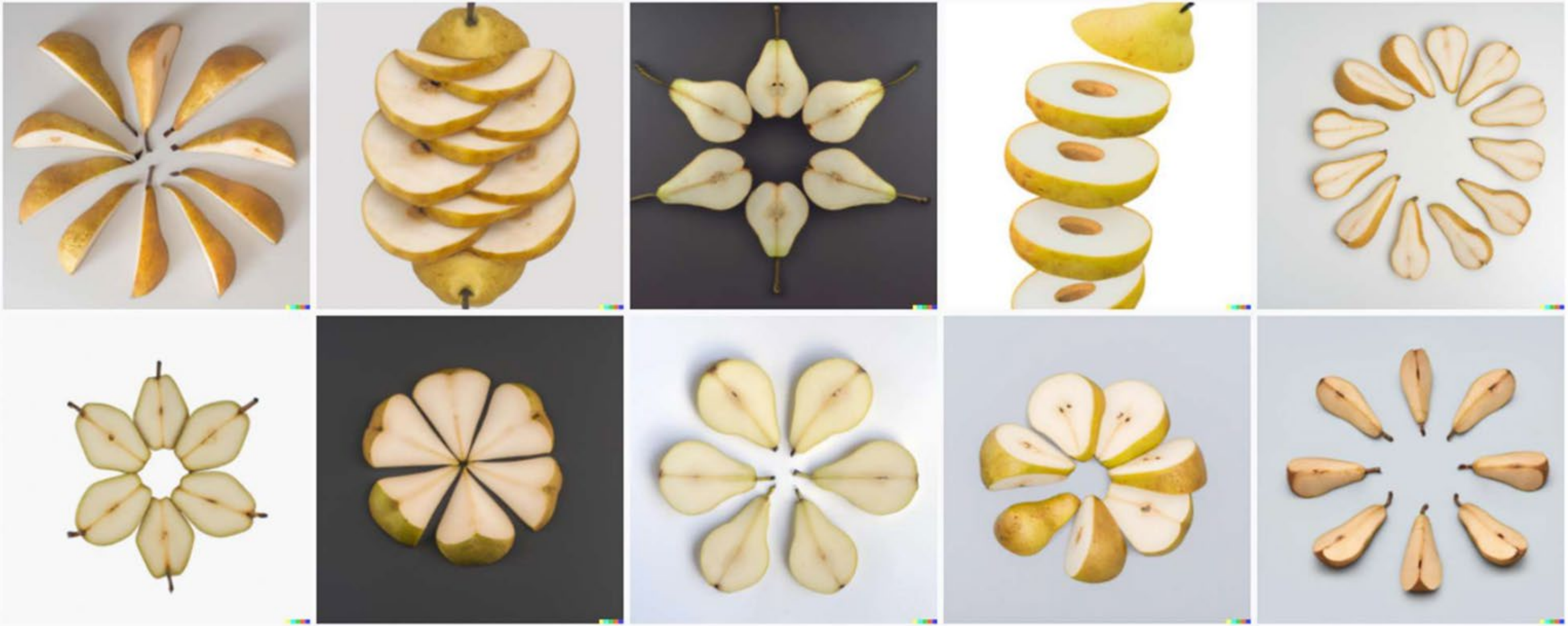
A red ball on top of a blue pyramid with the pyramid behind a car that is above a toaster.



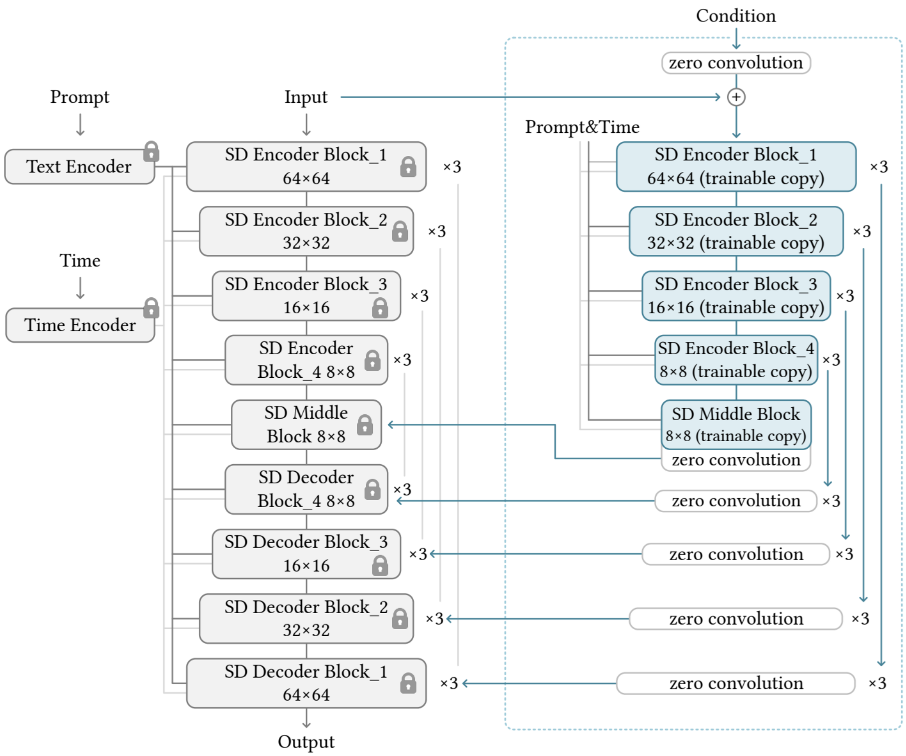
However

more examples in

Marcus, Gary, Ernest Davis, and Scott Aaronson. "A very preliminary analysis of DALL-E 2." arXiv 2204.13807



More Conditions: ControlNet



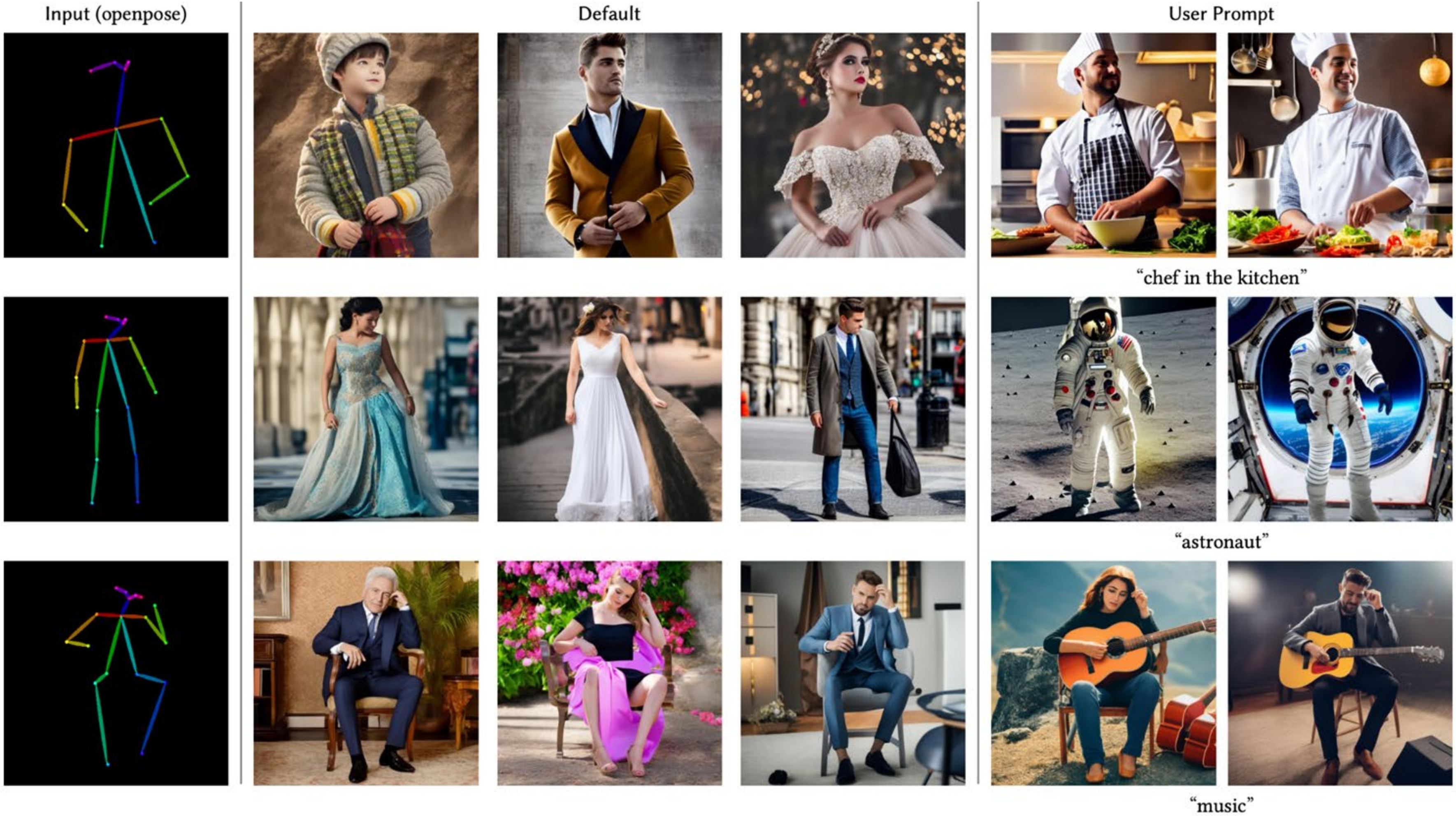
(a) Stable Diffusion (b) ControlNet

Zhang, Lvmin, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models." *arXiv preprint arXiv:2302.05543* (2023).

Edge


Input (HED Edge)
Default
Automatic Prompt
User Prompt
"a painting of a woman"
"... in cyan dress"
"... in red dress"
"a clown with a hat and a clown face"
"a clown with blue hair"
"a bird on a branch of a tree"
"white sparrow"

Pose



Input (openpose)
Default
User Prompt
“chef in the kitchen”
“astronaut”
“music”

Depth: Involving 3D information

Subject-Driven Generation

Given a few of images (3~5) of one object, generate more images of *this object*

We are familiar with class-driven and text-driven generation

different styles

"An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion"

Subject-Driven Generation

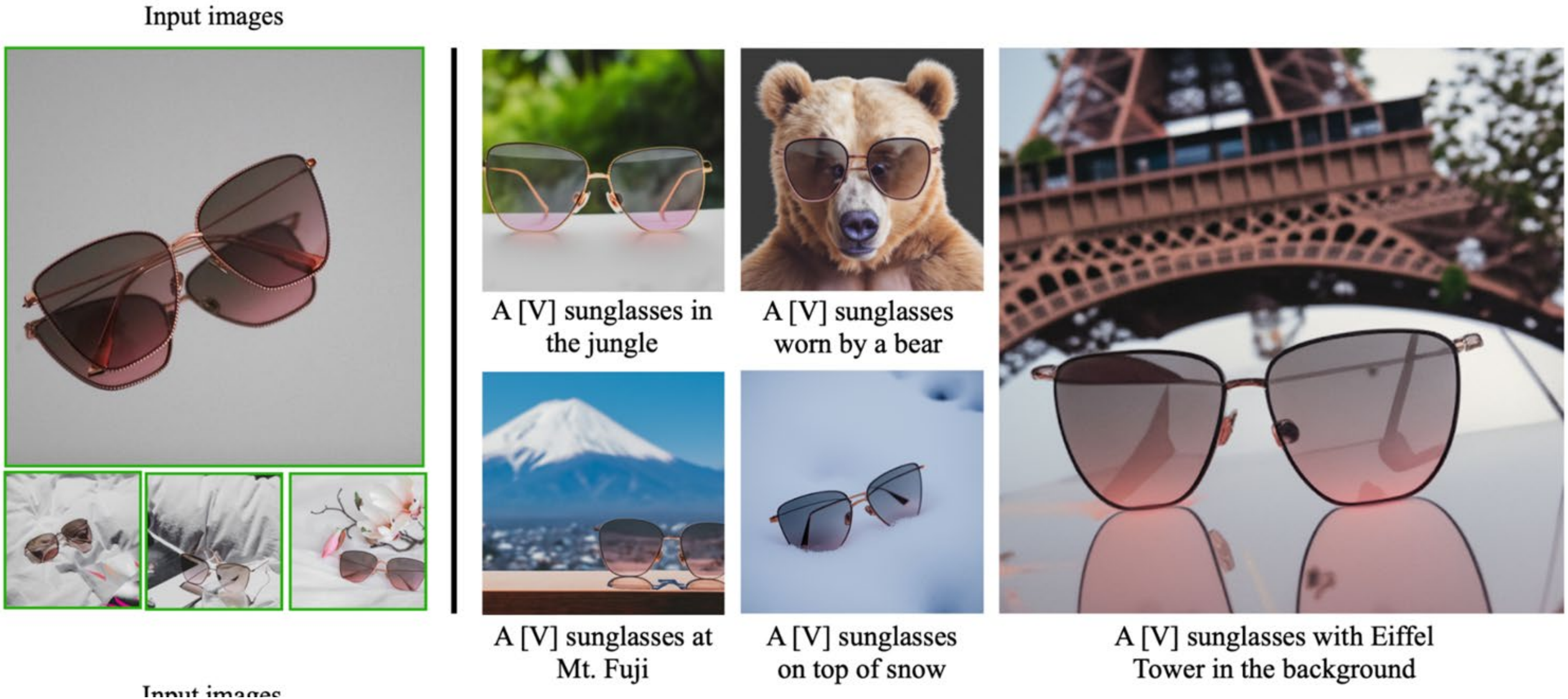
Given a few of images (3~5) of one object, generate more images of *this object*

We are familiar with class-driven and text-driven generation

different scenes and poses

“DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”

Subject-Driven Generation



different scenes and perspective

Image Editing

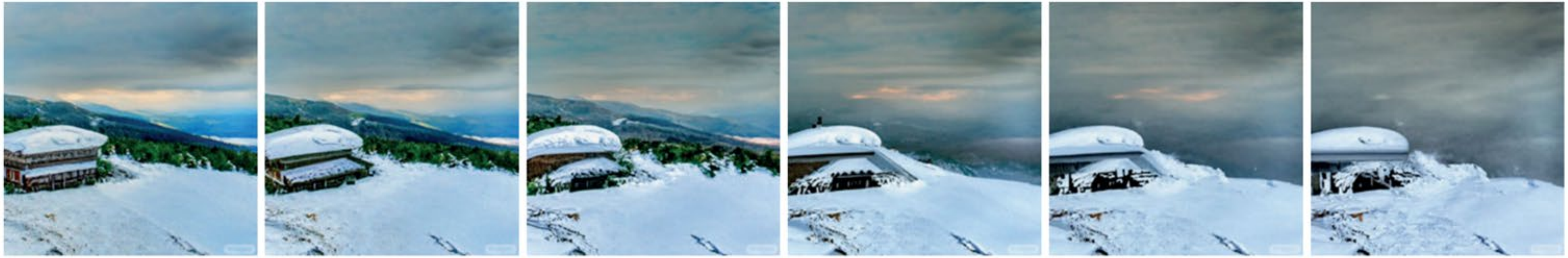
Editing the images with different text inputs, *without* changing the overall structure

"Prompt-to-Prompt Image Editing with Cross Attention Control"

Image Editing

Editing the images with different text inputs, *without* changing the overall structure

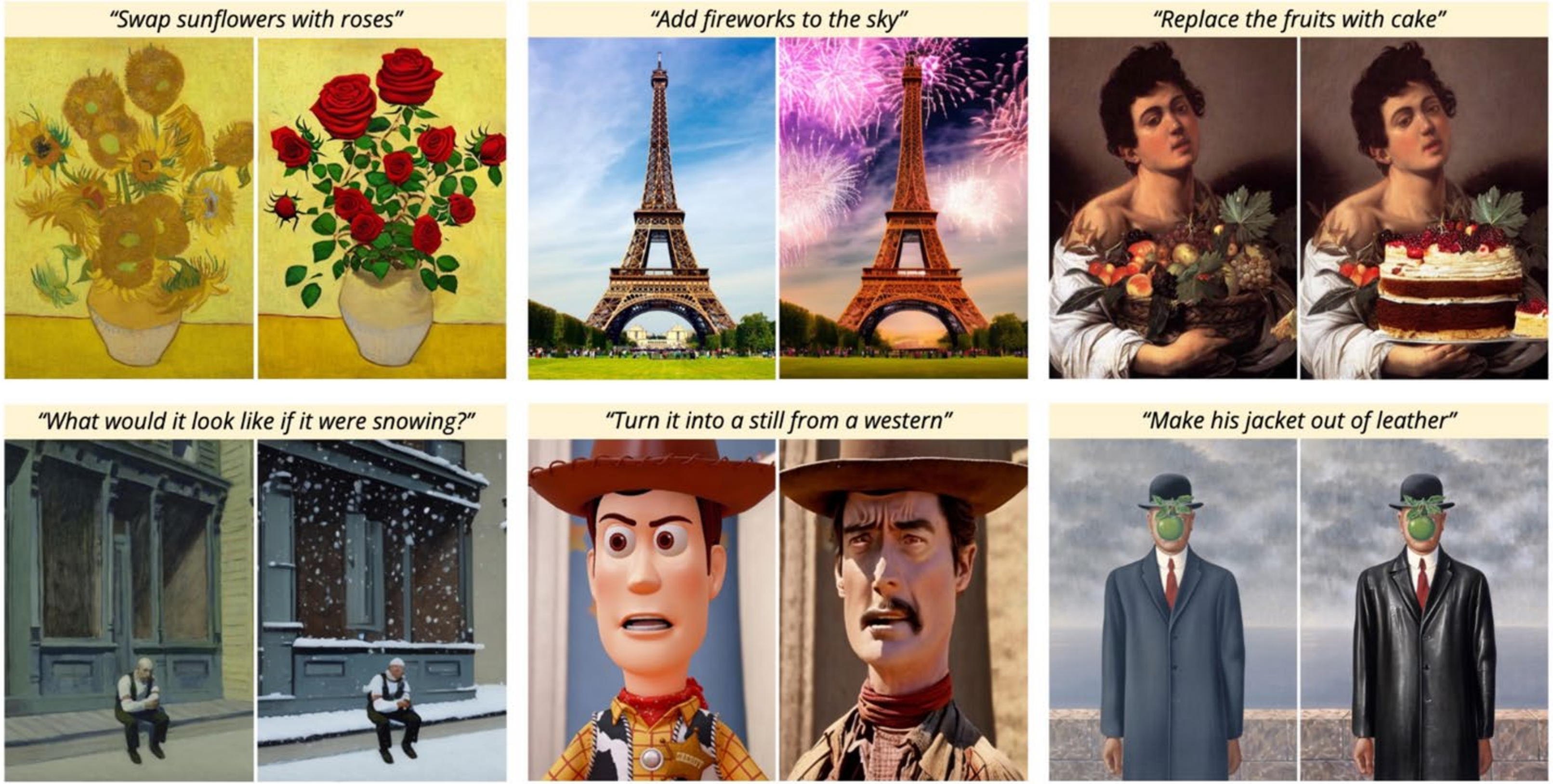

"A car on the side of the street."
source image
"...sport car..."
"...old car..."
"...mat black car..."
"...American car..."
"...crushed car..."
"...limousine car..."
"...convertibae car..."
Local description
Global description
"...the flooded street."
"...in Manhattan."
"...the blossom street."
"...at autumn."
"...at sunset."
"...in the snowy street."
"...in the forset."
"...at evening."
Imagen



“A photo of a house on a snowy(↑) mountain.”

“My fluffy(↑) bunny doll.

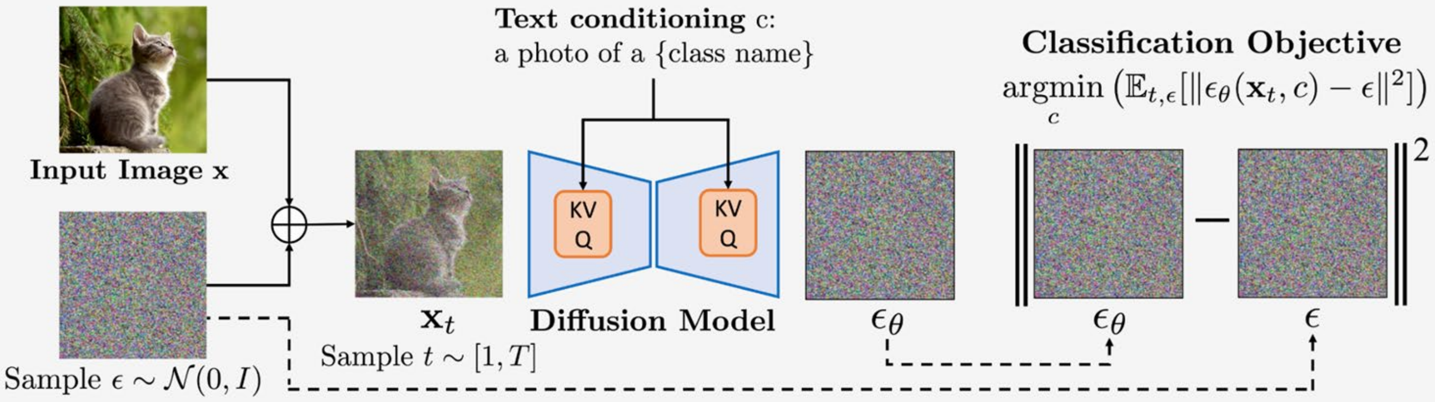
Instruction Based Image Editing



Given an image and a written instruction, our method follows the instruction to edit the image.

Brooks, Tim, Aleksander Holynski, and Alexei A. Efros. "Instructpix2pix: Learning to follow image editing instructions." *arXiv preprint arXiv:2211.09800* (2022).

Diffusion Model as One Classifier



Given an input image $\mathbf{x}$ and text conditioning $\mathbf{c}$, we use a diffusion model to choose the class that best fits this image. Our approach, ***Diffusion Classifier***, is theoretically motivated through the variational view of diffusion models and uses the ELBO to approximate $\log p_\theta(\mathbf{x}|\mathbf{c})$. Diffusion Classifier chooses the conditioning $\mathbf{c}$ that best predicts the noise added to the input image. Diffusion Classifier can be used to extract a *zero-shot classifier from a text-to-image model* (like Stable Diffusion) and a *standard classifier from a class-conditional model* (like DiT) without any additional training.

Li, Alexander C., et al. "Your Diffusion Model is Secretly a Zero-Shot Classifier." arXiv preprint arXiv:2303.16203

Diffusion Model as One Classifier

	Zero-shot?	Food	CIFAR10	FGVC	Pets	Flowers	STL10	ImageNet	ObjectNet
Synthetic SD Data	✓	12.6	35.3	9.4	31.3	22.1	38.0	18.9	5.2
SD Features	✗	73.0	**84.0**	**35.2**	75.9	**70.0**	87.2	56.6	10.2
Diffusion Classifier	✓	**77.9**	76.3	24.3	**85.7**	56.8	**94.2**	**58.4**	**38.3**
CLIP ResNet50	✓	81.1	75.6	19.3	85.4	65.9	94.3	58.2	40.0
OpenCLIP ViT-H/14	✓	92.7	97.3	42.3	94.6	79.9	98.3	76.8	69.2

Zero-shot classification performance on a suite of tasks.

Li, Alexander C., et al. "Your Diffusion Model is Secretly a Zero-Shot Classifier." arXiv preprint arXiv:2303.16203