# EDICT: Exact Diffusion Inversion via Coupled Transformations

CVPR 2023

Bram Wallace, Akash Gokul, Nikhil Naik

Salesforce Research

# Outline

- Authorship
- Background
- Architecture
- Experiments

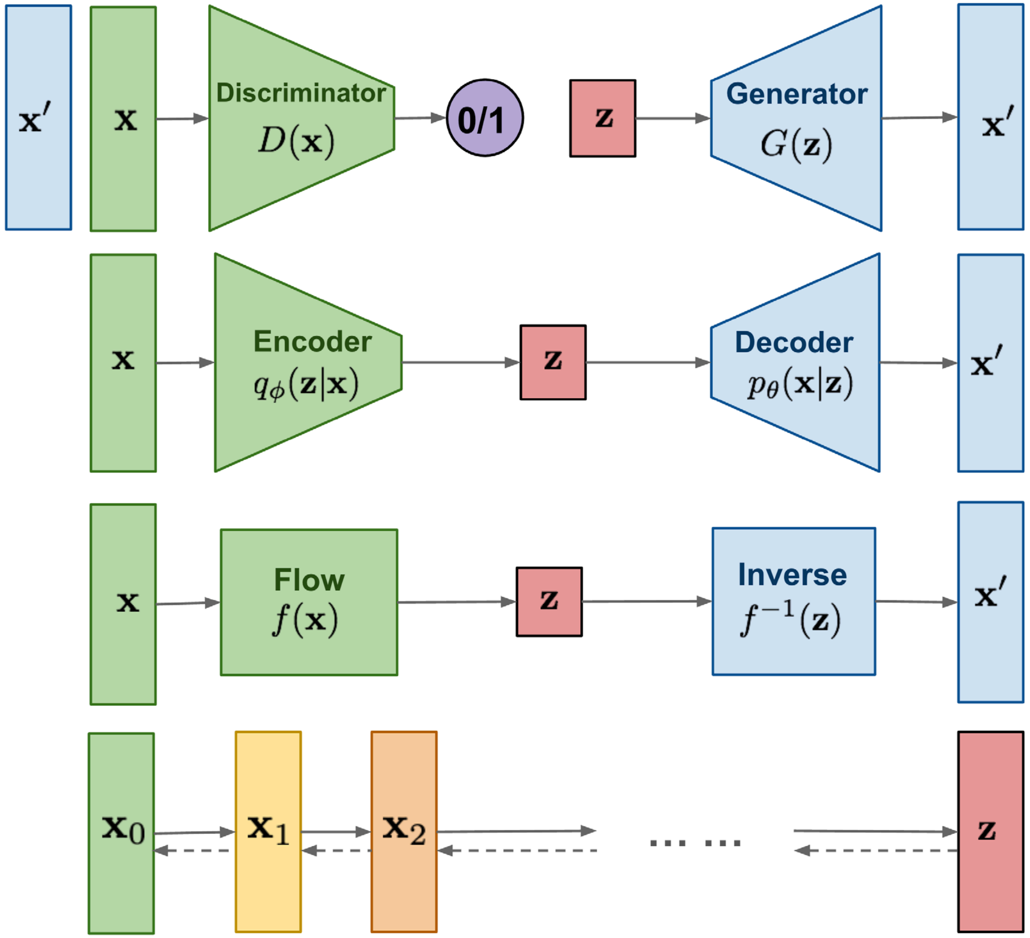
# Generative Models

**GAN:** Adversarial training

**VAE:** maximize variational lower bound

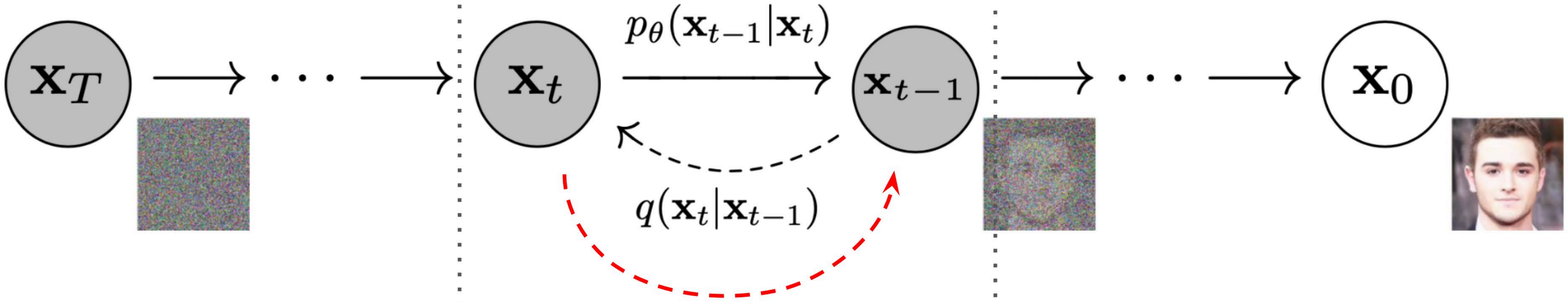
**Flow-based models:** Invertible transform of distributions

**Diffusion models:** Gradually add Gaussian noise and then reverse



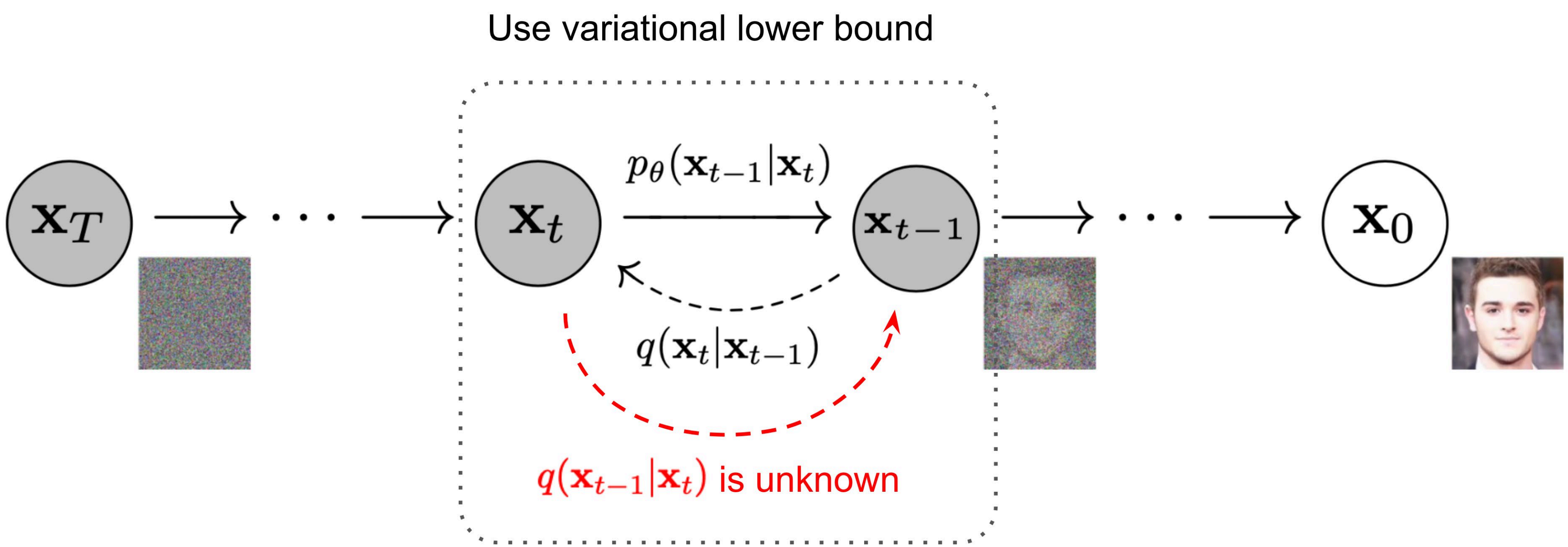
# Denoising Diffusion Models

Use variational lower bound




$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown

# Denoising Diffusion Models

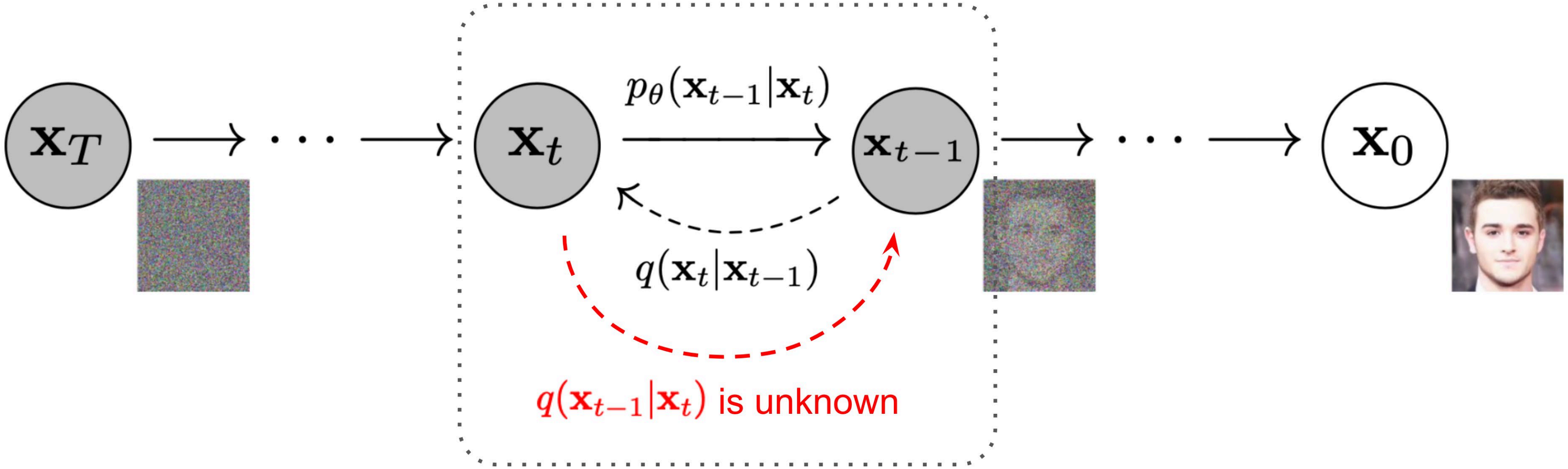



$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

# Denoising Diffusion Models

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Use variational lower bound



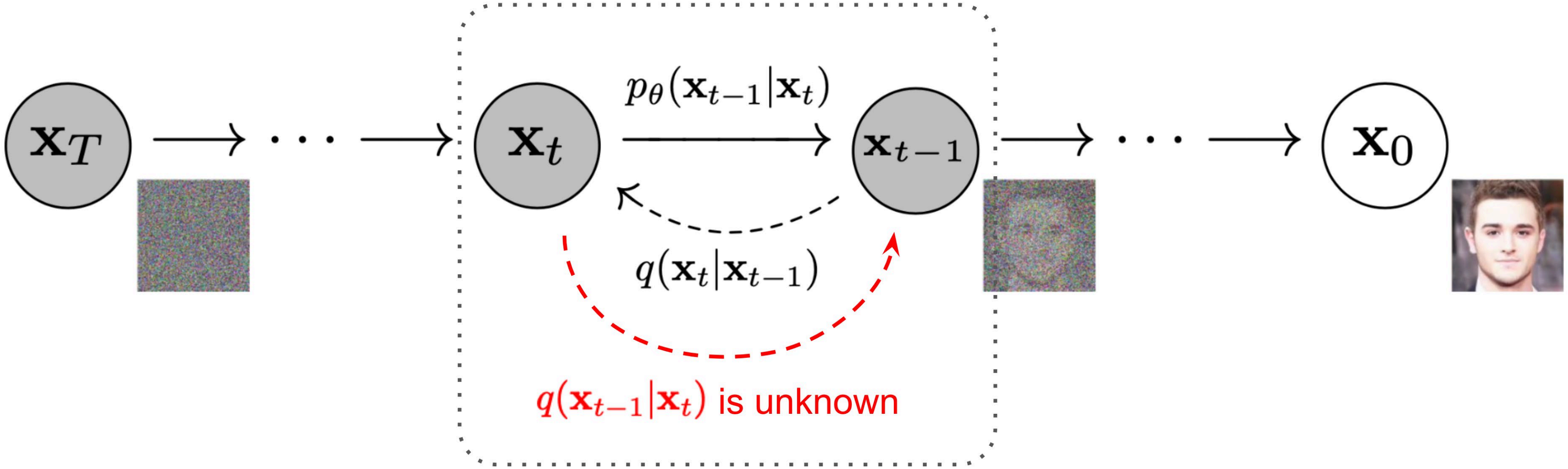

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

# Denoising Diffusion Models

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

if $\beta_t$ is small enough

Use variational lower bound




$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

# Denoising Diffusion Probabilistic Models

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

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

# Denoising Diffusion Probabilistic Models

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

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_{t-1};\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0),\tilde{\beta}_t\mathbf{I}),$$

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

$$L_{t-1}-C=\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}),\frac{1}{\sqrt{\bar{\alpha}_t}}(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon})-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon})\right)-\boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}),t)\right\|^2\right] \tag{9}$$

$$=\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon})-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}\right)-\boldsymbol{\mu}_\theta(\mathbf{x}_t(\mathbf{x}_0,\boldsymbol{\epsilon}),t)\right\|^2\right] \tag{10}$$

北京大学 PEKING UNIVERSITY

# Denoising Diffusion Probabilistic Models

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t)\right\|^2\right]$$

**Algorithm 1** Training

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

**Algorithm 2** Sampling

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

北京大学 PEKING UNIVERSITY

# Denoising Diffusion Probabilistic Models

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}}\underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t)\right\|^2\right]$$

**Algorithm 1** Training

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

**Algorithm 2** Sampling

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

# Denoising Diffusion Implicit Models

Improve DDPM because:

1. T = 1000 is too big. Accelerate!
2. Retraining is disturbing. Same objective!
3. Results are stochastic. Deterministic sampling!

# Denoising Diffusion Implicit Models

Improve DDPM with:

1. Non-Markov chain
2. Same marginals, different joints
3. Modeling the uncertainty with $\sigma_t$

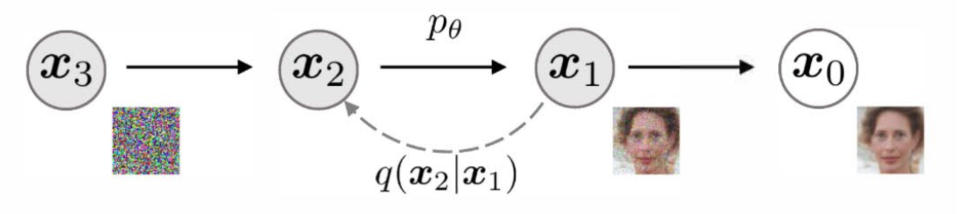
# Denoising Diffusion Implicit Models

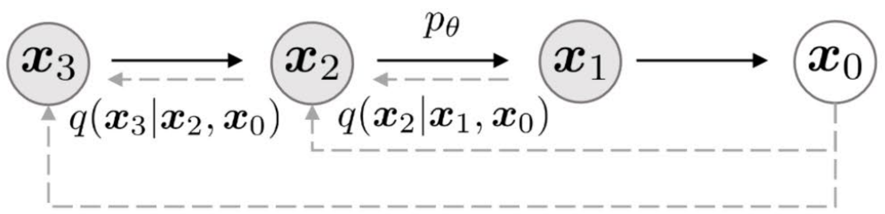
Review objectives of DDPM:

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

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0):=\prod_{t=1}^{T}q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$

$$q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0):=q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0)\prod_{t=2}^{T}q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$$



# Denoising Diffusion Implicit Models

Review the special property of DDPM:

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

⬇

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\boldsymbol{I}\right)$$

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\boldsymbol{I});$$

# Denoising Diffusion Implicit Models

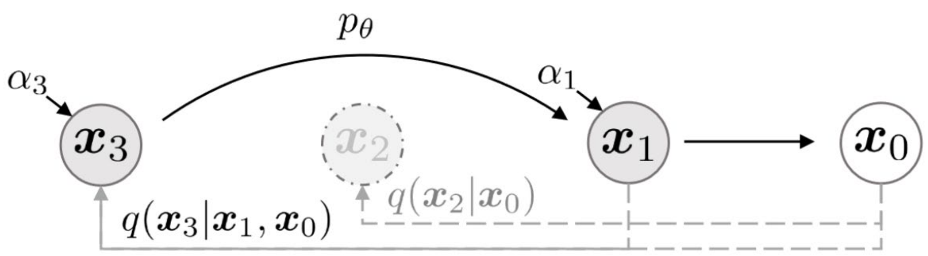
Objective:

$$J_\sigma(\epsilon_\theta) := \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q_\sigma(\boldsymbol{x}_{0:T})}[\log q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) - \log p_\theta(\boldsymbol{x}_{0:T})] \tag{11}$$

$$= \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q_\sigma(\boldsymbol{x}_{0:T})}\left[\log q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) + \sum_{t=2}^{T} \log q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) - \sum_{t=1}^{T} \log p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) - \log p_\theta(\boldsymbol{x}_T)\right]$$

# Denoising Diffusion Implicit Models

Sampling:




$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left( \frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{`` predicted } \boldsymbol{x}_0\text{''}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{``direction pointing to } \boldsymbol{x}_t\text{''}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

# Denoising Diffusion Implicit Models

What does $\sigma_t$ represent:

$$
\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_t + \sigma_t\boldsymbol{\epsilon} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\boldsymbol{\epsilon} \\
q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I}\right)
\end{aligned}
$$

# Denoising Diffusion Implicit Models

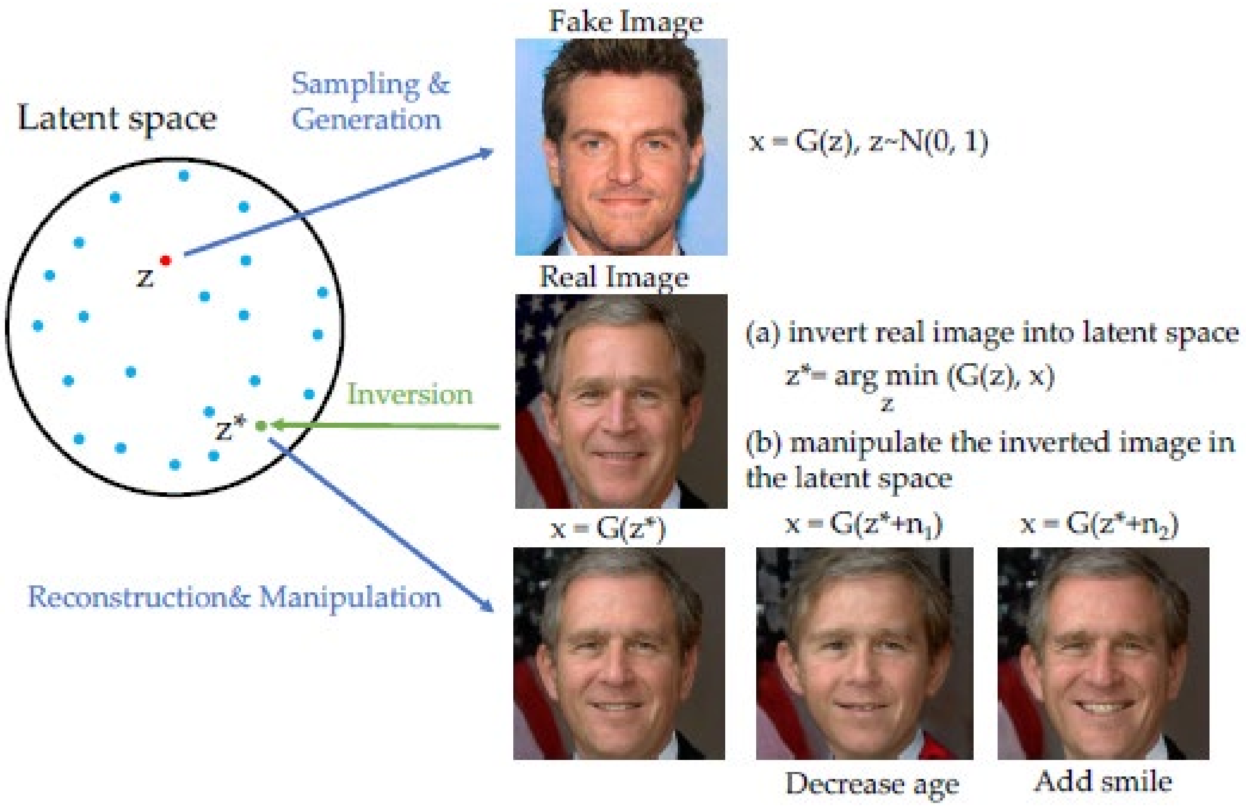
What does $\sigma_t$ represent:

$$
\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_t + \sigma_t\boldsymbol{\epsilon} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\boldsymbol{\epsilon} \\
q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I}\right)
\end{aligned}
$$

$$\sigma_t = 0 \Rightarrow \epsilon_{t-1} = \epsilon_t$$

$$\sigma_t = \sqrt{1-\bar{\alpha}_{t-1}} \Rightarrow \epsilon_{t-1}, \epsilon_t \; i.i.d$$

# GAN Inversion



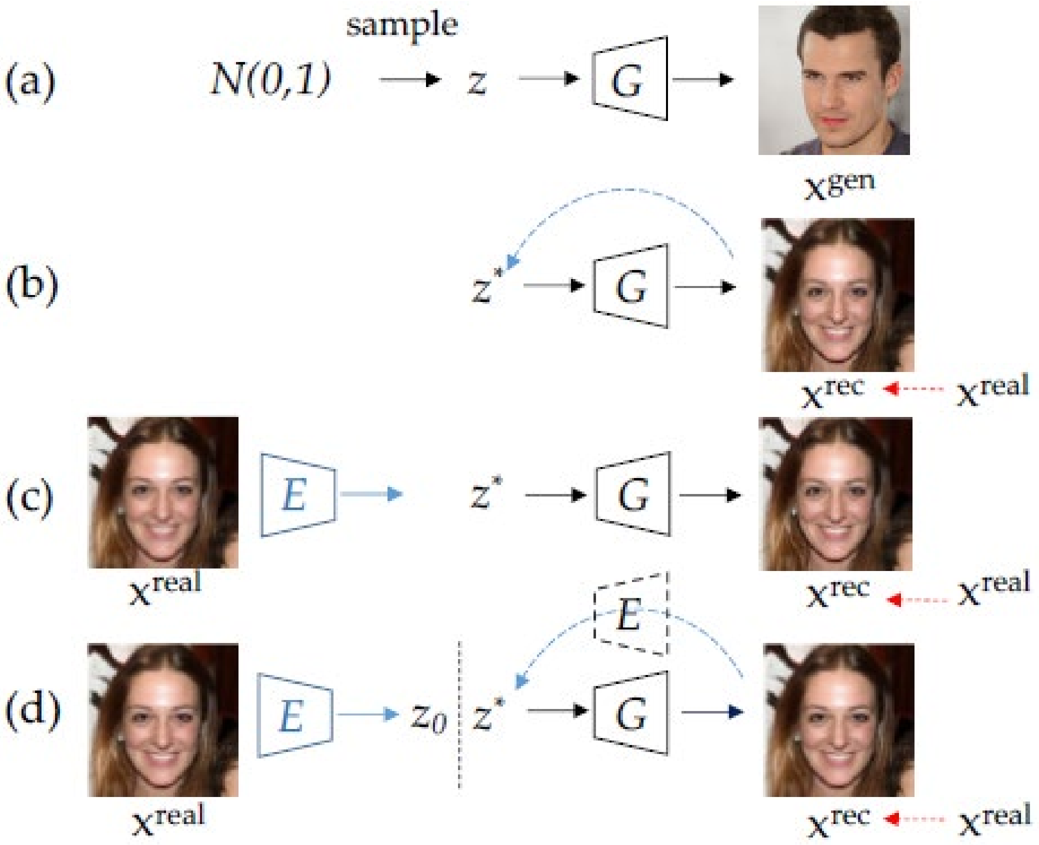

Xia, Weihao, et al. "Gan inversion: A survey." *IEEE TPAMI,* 45.3 (2022): 3121-3138.

# GAN Inversion

Optimization

Encoder

Hybrid



# Outline

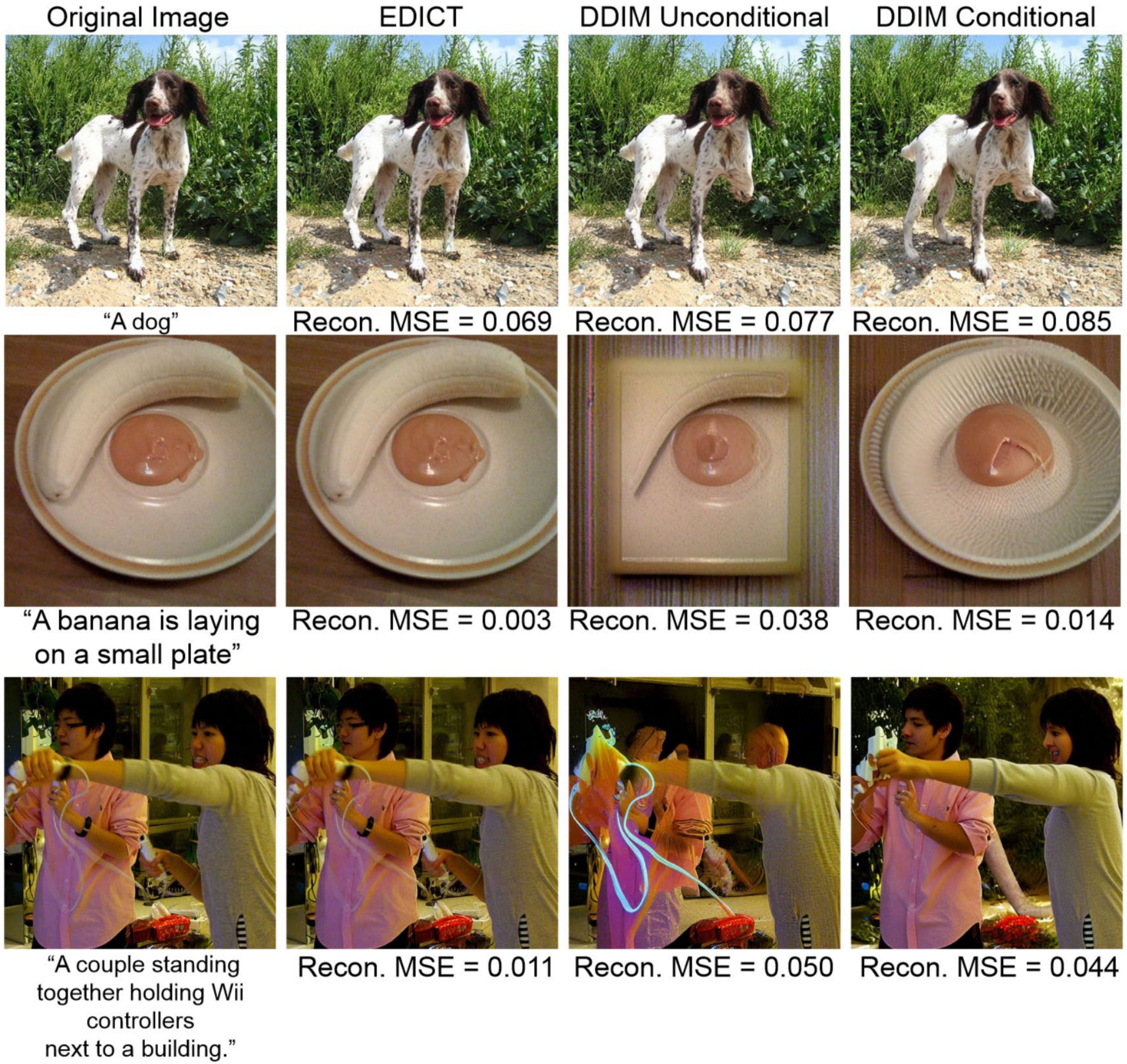
- Authorship
- Background
- Architecture
- Experiments

# Extract Diffusion Inversion

- The generative process in DDIMs is defined in a non-Markovian manner, which results in a deterministic denoising process.
- DDIM can also be used for inversion, deterministically noising an image to obtain the initial noise vector.

- DDIM inversion is unstable in many cases.*

* Amir Hertz, et al. Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626, 2022.



Original Image
EDICT
DDIM Unconditional
DDIM Conditional
"A dog"
Recon. MSE = 0.069
Recon. MSE = 0.077
Recon. MSE = 0.085
"A banana is laying on a small plate"
Recon. MSE = 0.003
Recon. MSE = 0.038
Recon. MSE = 0.014
"A couple standing together holding Wii controllers next to a building."
Recon. MSE = 0.011
Recon. MSE = 0.050
Recon. MSE = 0.044

# Extract Diffusion Inversion

- DDIM inversion relies on the local linear assumption.

$$x_t = \frac{x_{t-1} - b_t \epsilon(x_t, t)}{a_t} \approx \frac{x_{t-1} - b_t \epsilon(x_{t-1}, t)}{a_t}$$

- Recall:

$$\begin{aligned} \mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\boldsymbol{\epsilon}_t + \sigma_t\boldsymbol{\epsilon} \end{aligned}$$

# Affine Coupling Layers

- Widely used in flow-based models.

$$z = [z_a, z_b]$$

$$z'_a = \Psi(z_b) z_a + \psi(z_b)$$

$$z_a = (z'_a - \psi(z_b)) / \Psi(z_b)$$

# EDICT-Invertible

- Observe the similarity of

$$z_a' = \Psi(z_b)z_a + \psi(z_b)$$

and

$$x_{t-1} := a_t x_t + b_t \epsilon(x_t, t)$$

, replace second $x_t$ with $y_t$

$$x_{t-1} = a_t x_t + b_t \epsilon(y_t, t)$$

# EDICT-Invertible

$$x_{t-1} = a_t x_t + b_t \epsilon(y_t, t)$$

is invertible

$$x_t = (x_{t-1} - b_t \cdot \epsilon(y_t, t))/a_t$$

Define the updating function:

$$x_{t-1} = a_t x_t + b_t \cdot \epsilon(y_t, t)$$
$$y_{t-1} = a_t y_t + b_t \cdot \epsilon(x_{t-1}, t)$$

# EDICT-Invertible

$$x_{t-1} = a_t x_t + b_t \cdot \epsilon(y_t, t)$$
$$y_{t-1} = a_t y_t + b_t \cdot \epsilon(x_{t-1}, t)$$

is invertible

$$y_t = (y_{t-1} - b_t \cdot \epsilon(x_{t-1}, t))/a_t$$
$$x_t = (x_{t-1} - b_t \cdot \epsilon(y_t, t))/a_t$$

# EDICT-Invertible

The gap between defined procedure and DDIM depends of the gap of $x_t, x_{t-1}$ and $y_t$.

$$x_{t-1} = a_t x_t + b_t \cdot \epsilon(y_t, t)$$

$$y_{t-1} = a_t y_t + b_t \cdot \epsilon(x_{t-1}, t)$$

should be $x_t$

should be $x_t$

should be $x_t$

# EDICT-Stabilization

Directly using it is unsatisfactory due to error accumulation.

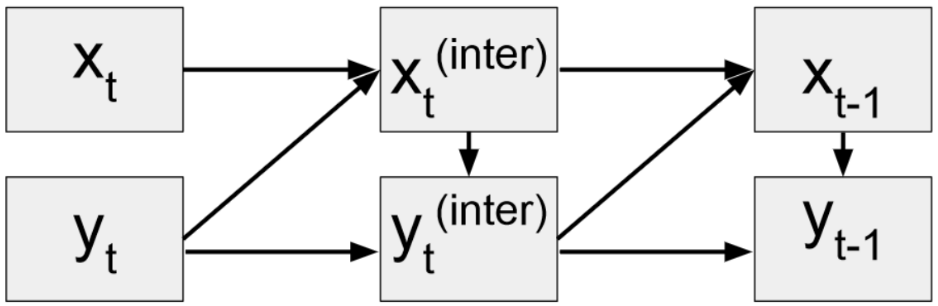
# EDICT-Stabilization

Propose mixing layers:

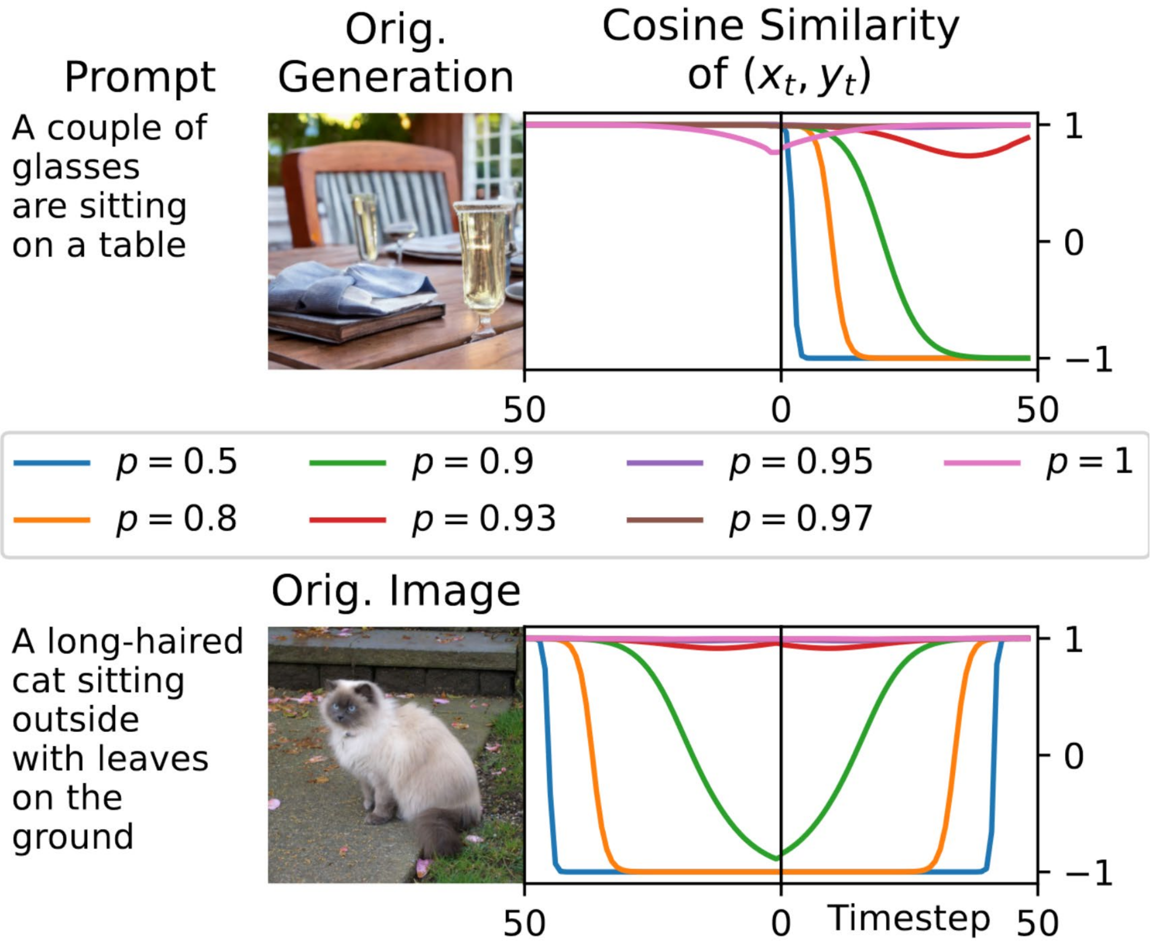
$$x' = px + (1-p)y,\ 0 \leq p \leq 1$$

to stretch them closer.

$$x_t^{inter} = a_t \cdot x_t + b_t \cdot \epsilon(y_t, t)$$
$$y_t^{inter} = a_t \cdot y_t + b_t \cdot \epsilon(x_t^{inter}, t)$$
$$x_{t-1} = p \cdot x_t^{inter} + (1-p) \cdot y_t^{inter}$$
$$y_{t-1} = p \cdot y_t^{inter} + (1-p) \cdot x_{t-1}$$



# EDICT-Stabilization



# EDICT-Application

While EDICT can theoretically operate on either pixel-based or latent-diffusion models, we present the latter case in this work.

Noising w/ condition $C_{base}$, denoising w/ condition $C_{target}$.

# Outline

- Authorship
- Background
- Architecture
- Experiments

# Outline

- Authorship
- Background
- Architecture
- Experiments

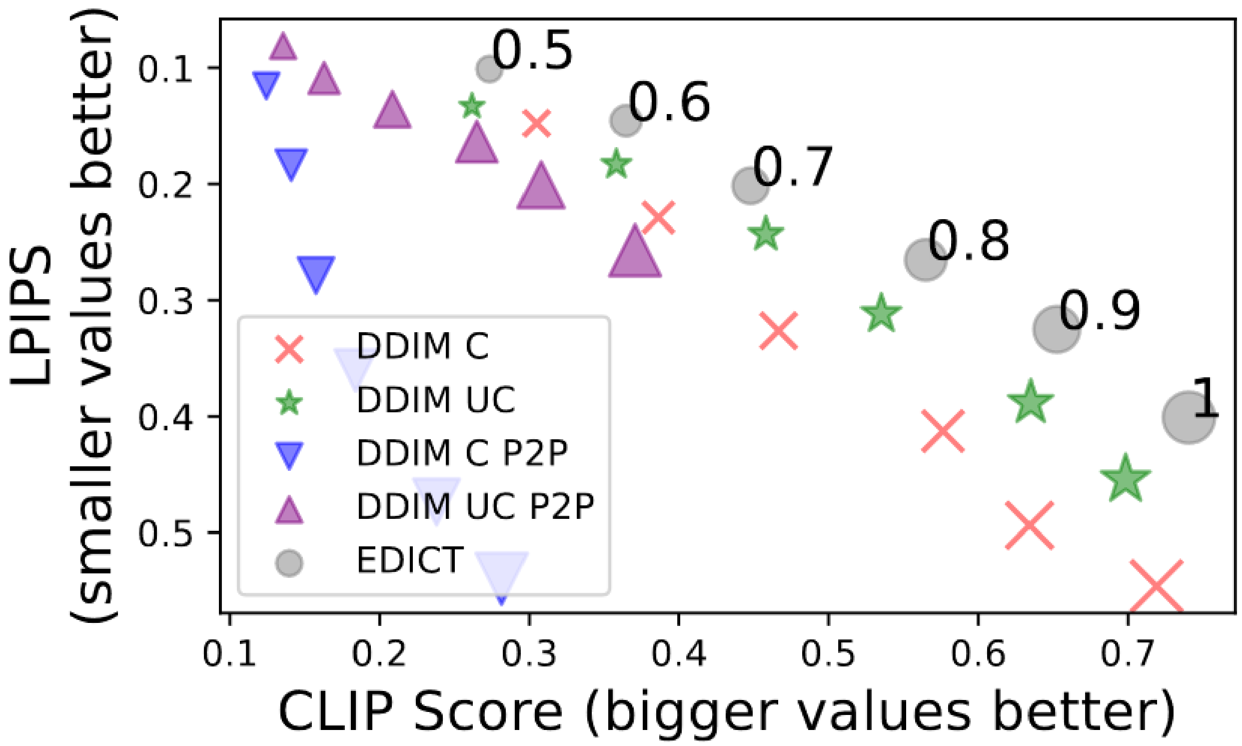
# Reconstruction

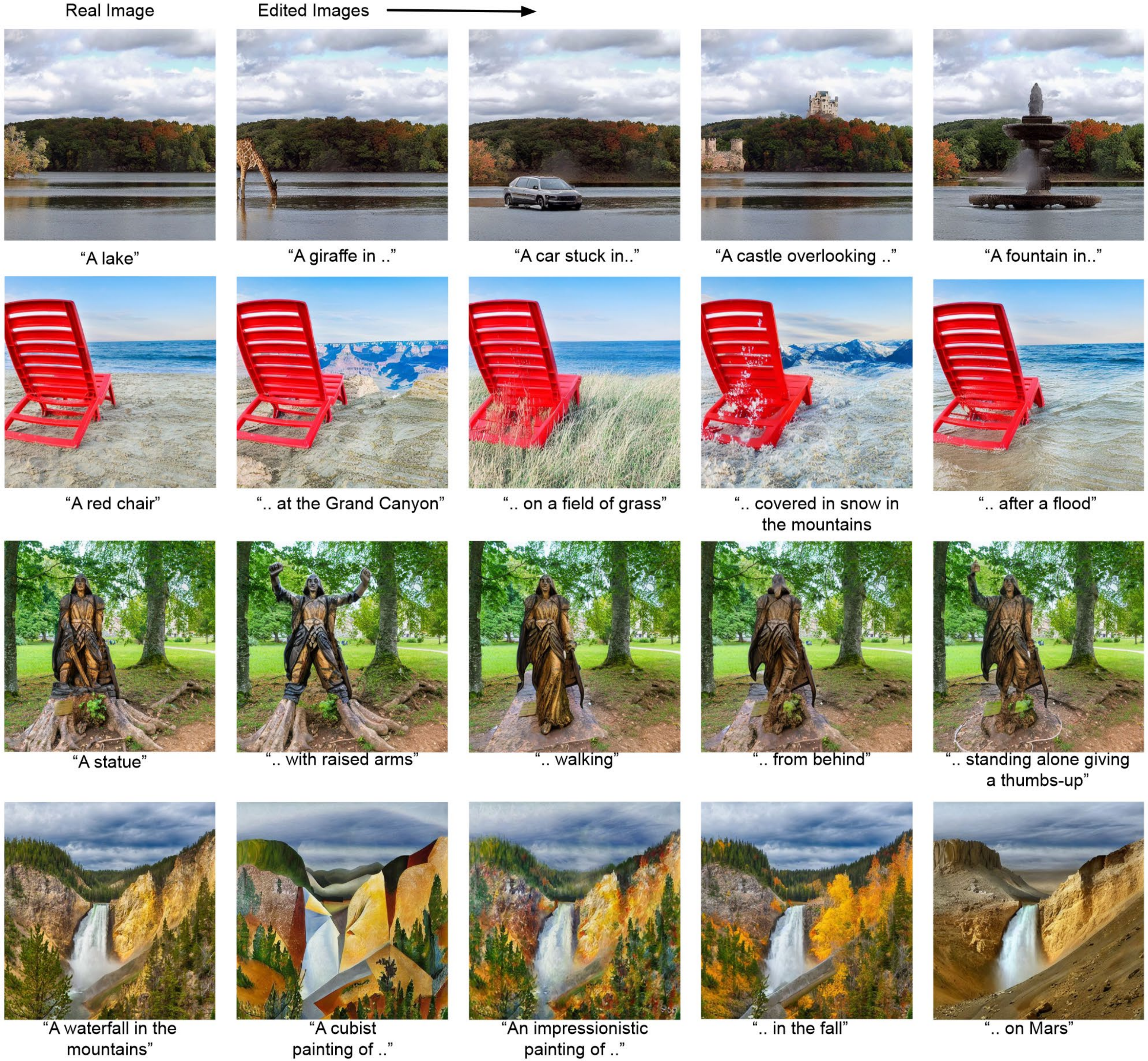
COCO Reconstruction Error (MSE)

| Method | LDM AE | EDICT (UC) | EDICT (C) | DDIM (UC) | DDIM (C) |
|---|---|---|---|---|---|
| 50 Steps | 0.015 | 0.015 | 0.015 | 0.030 | 0.420 |
| 100 Steps | 0.015 | 0.015 | 0.015 | 0.027 | 0.471 |
| 200 Steps | 0.015 | 0.015 | 0.015 | 0.023 | 0.497 |

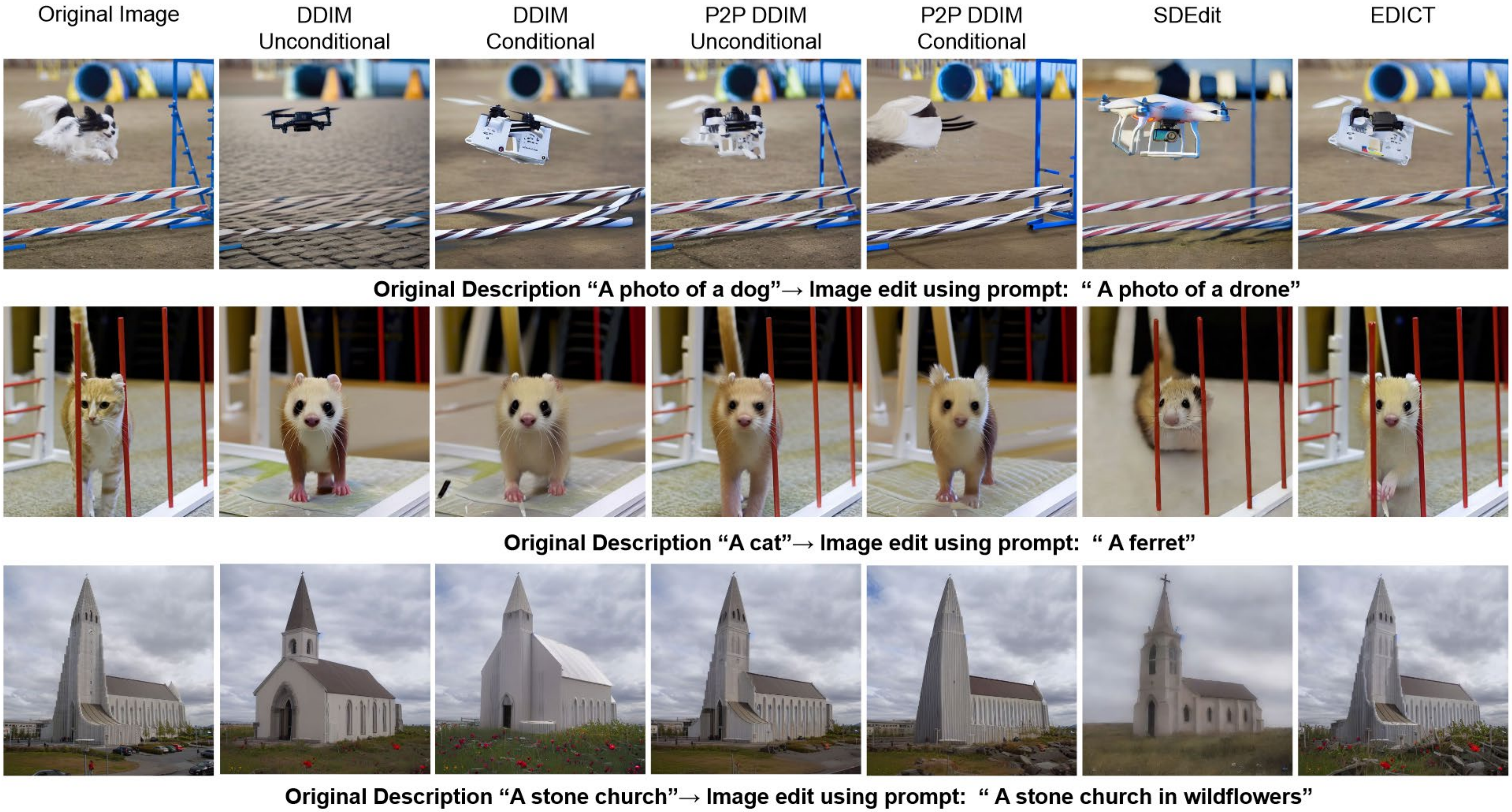
Table 1. Mean-square reconstruction error for COCO-val using the first listed prompt as conditioning with $G = 7$. The latent diffusion model autoencoder (LDM AE) is the lower bound on reconstruction error. Using half precision increases 50-step EDICT (C) MSE by 6%. More step values are in the Supplementary.

# Editing


LPIPS (smaller values better)
CLIP Score (bigger values better)
0.1
0.2
0.3
0.4
0.5
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.5
0.6
0.7
0.8
0.9
1
DDIM C
DDIM UC
DDIM C P2P
DDIM UC P2P
EDICT




北京大學 PEKING UNIVERSITY


Original Image
DDIM Unconditional
DDIM Conditional
P2P DDIM Unconditional
P2P DDIM Conditional
SDEdit
EDICT
Original Description "A photo of a dog"→ Image edit using prompt: " A photo of a drone"
Original Description "A cat"→ Image edit using prompt: " A ferret"
Original Description "A stone church"→ Image edit using prompt: " A stone church in wildflowers"

# Conclusion

- A method to promise the invertibility of diffusion steps.
- Performs well both on conditional and unconditional cases.
- Less changes on irrelative regions with editing.

- No FID scores.
- Insufficient comparison.

Thanks for your listening!