Diffusion Models and Their Applications

廖家緯 Jiawei

Ph.D. Candidate in Computer Science

National Taiwan University

jwliao1209 jwliao1209

NTUAI CLUB
台大人工智慧應用社

Slido

Feel free to ask any questions

https://app.sli.do/event/hbjsRgfD9KdhFAcm8eqb9s

My AI Journey

BS in Math

MS in Applied Math

PhD Candidate in CSIE

AAAI 2025

Website

2020

2022

2023

2024

2025

Research Intern

DA Intern

Research Assistant

AI Research Intern

SWE Intern

ML Research Intern

Outline

- Training and Sampling
- Controllability and Applications
- Aesthetic QR Code Generation
- Evasion Attack

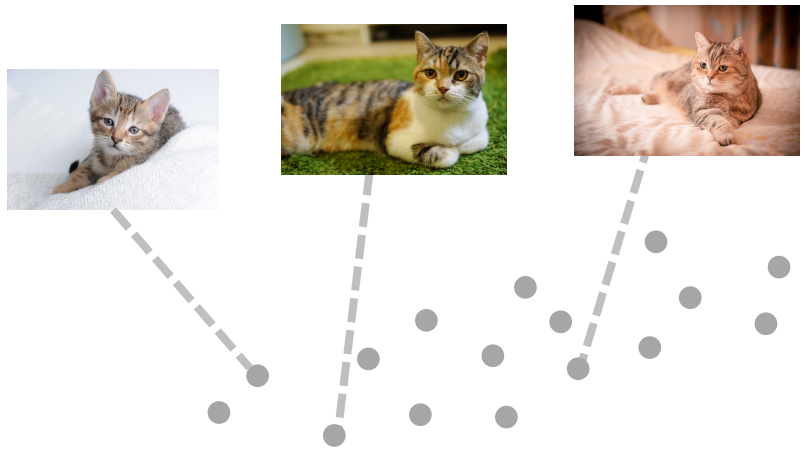
What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional hyper-surface



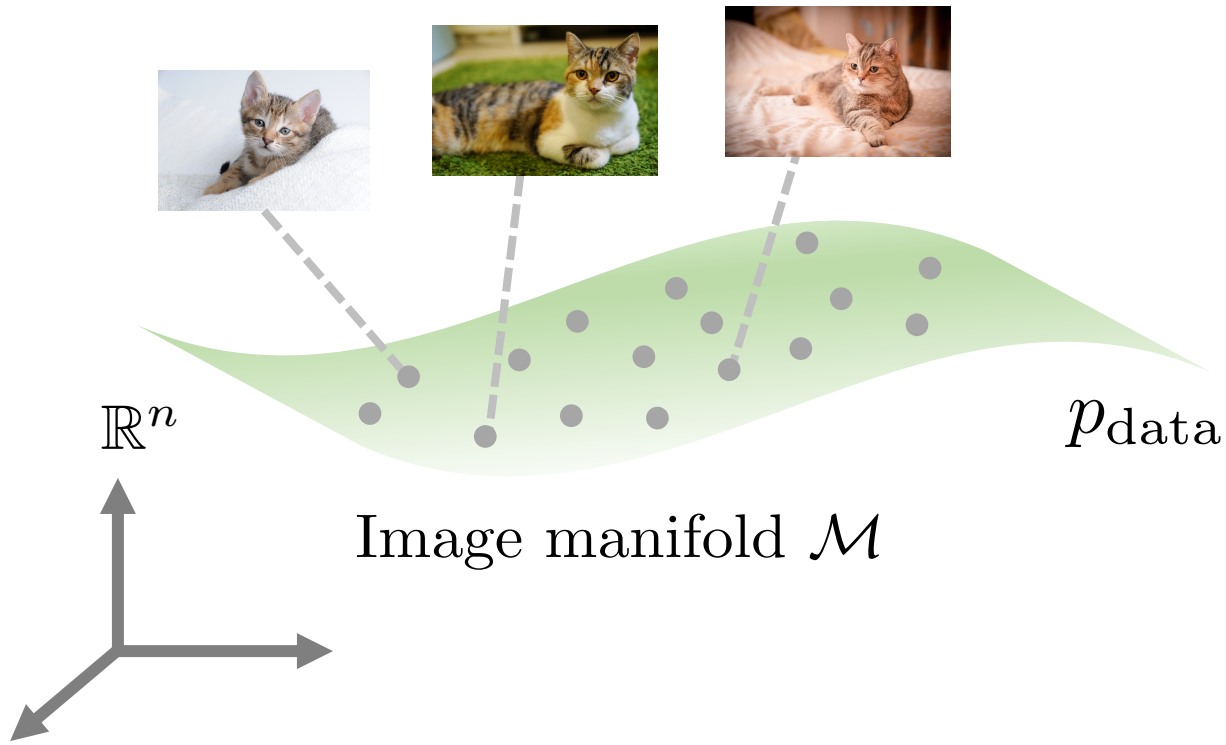
What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional hyper-surface



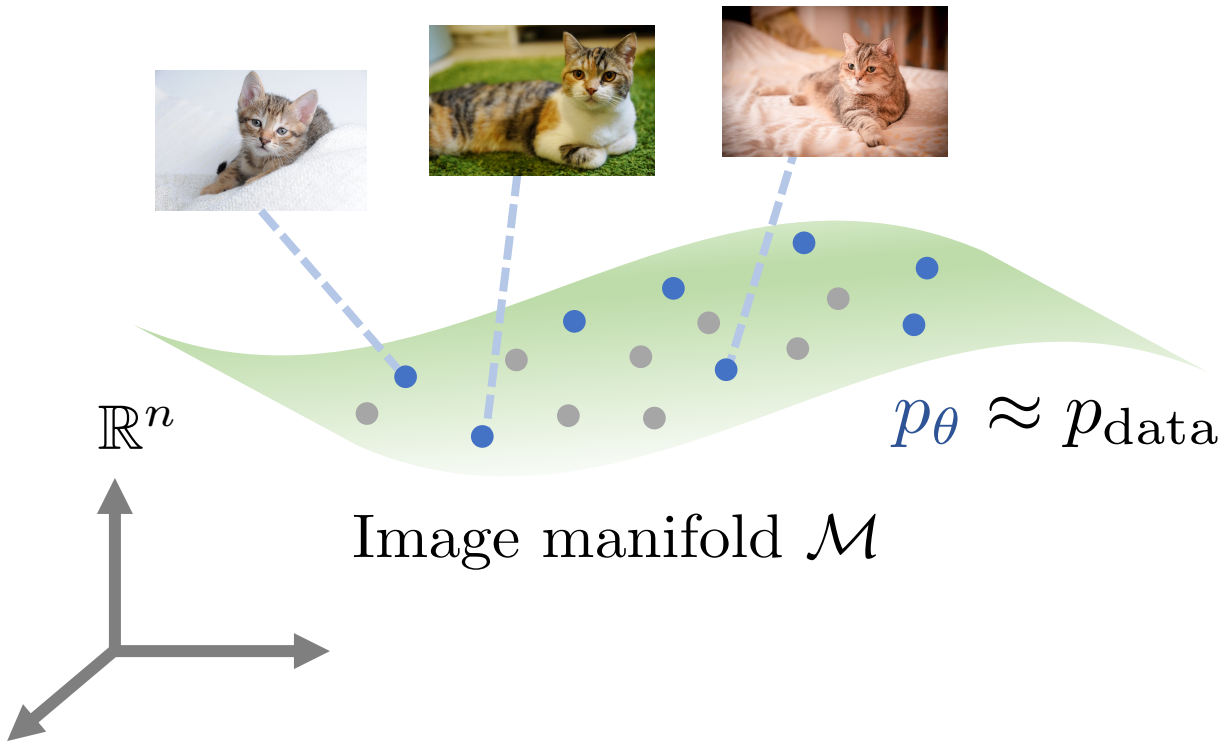
What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional hyper-surface



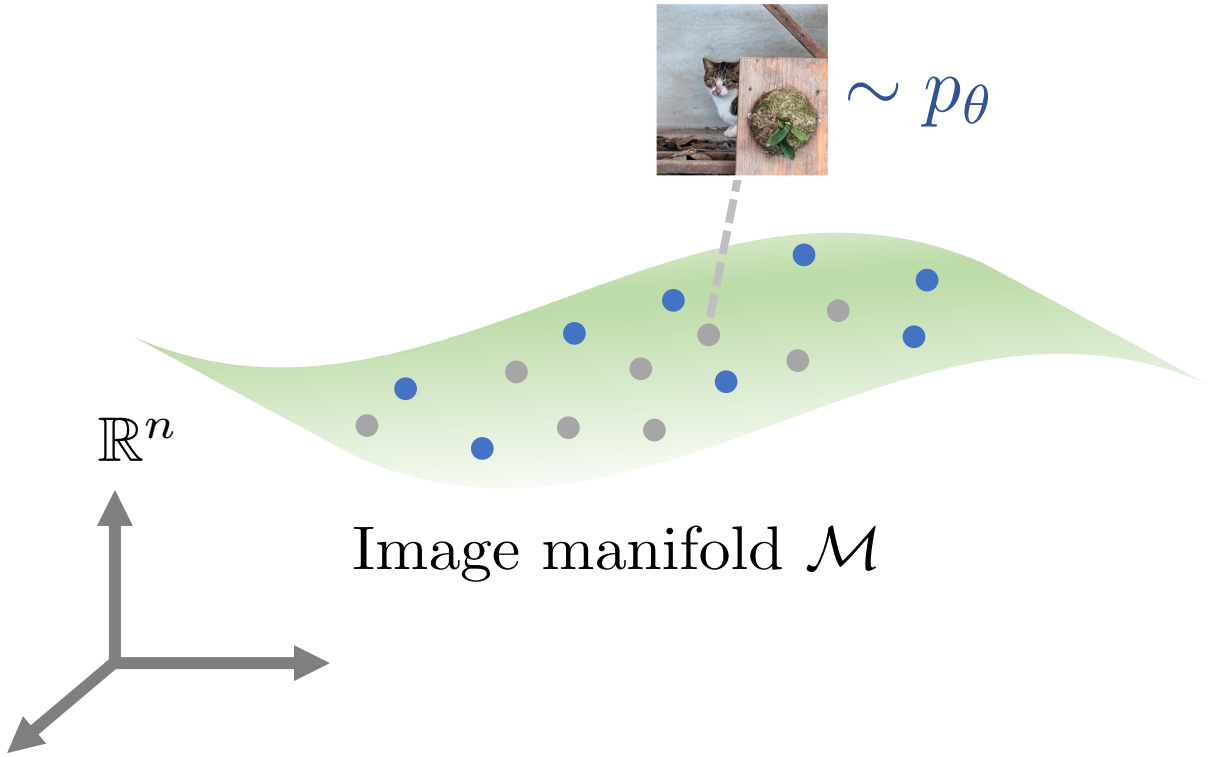
What is Generative Model Learning?

Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional hyper-surface

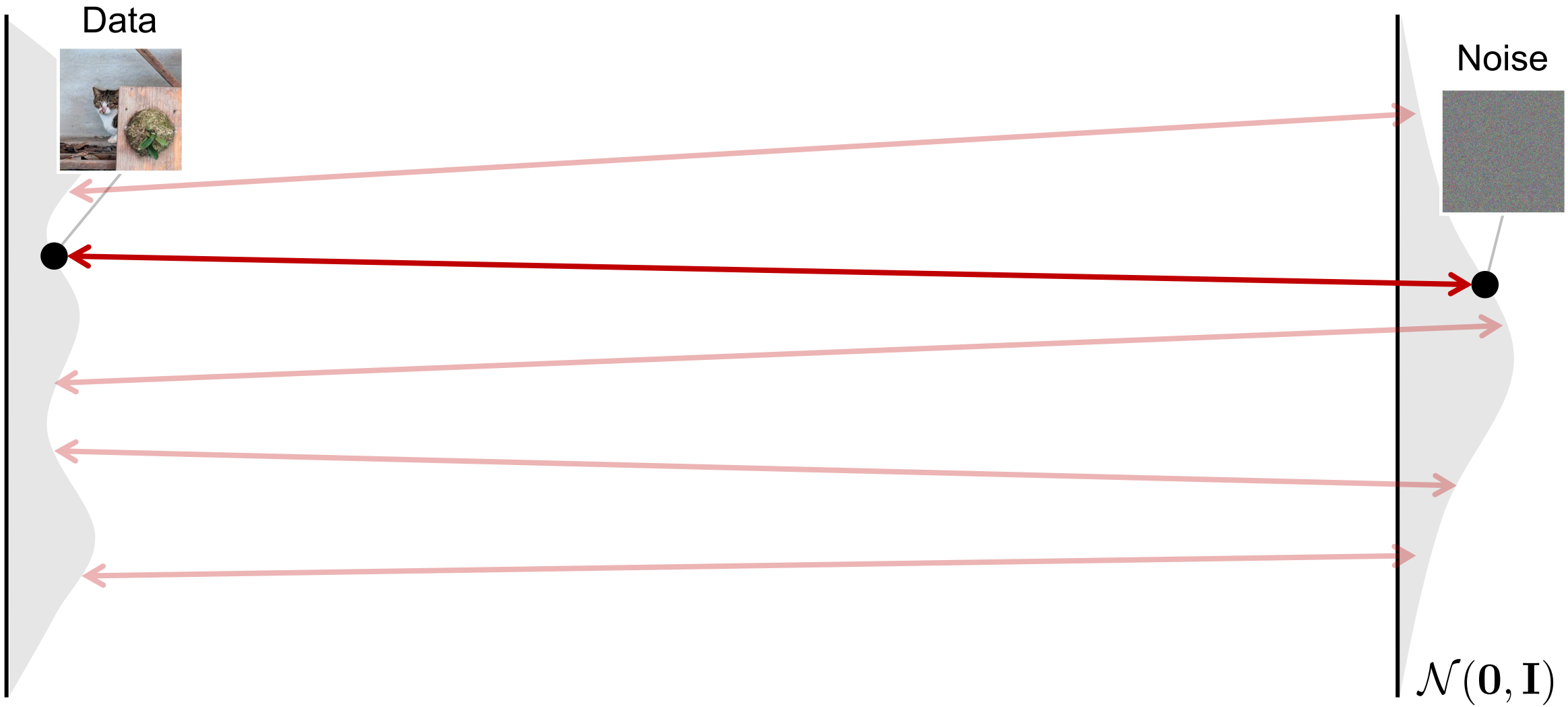


What is Generative Model Learning?

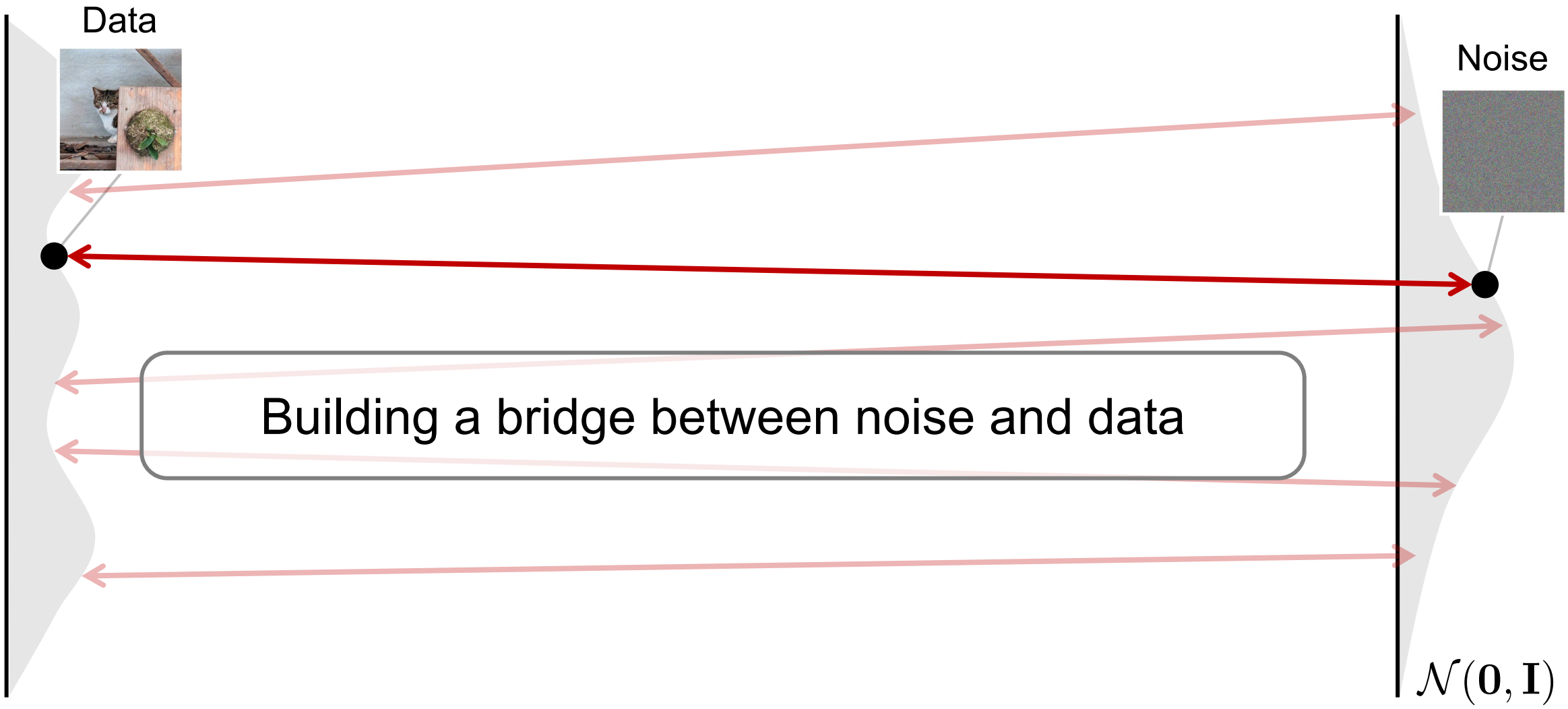
Data Manifold Assumption
Natural high-dimensional data concentrate close to a non-linear low-dimensional hyper-surface



The Goal of Diffusion Models



The Goal of Diffusion Models



What is Diffusion Model?

Forward Process: add noise step by step, from data to pure noise

What is Diffusion Model?

Forward Process: add noise step by step, from data to pure noise

Reverse Process: generate data from pure noise by denoising

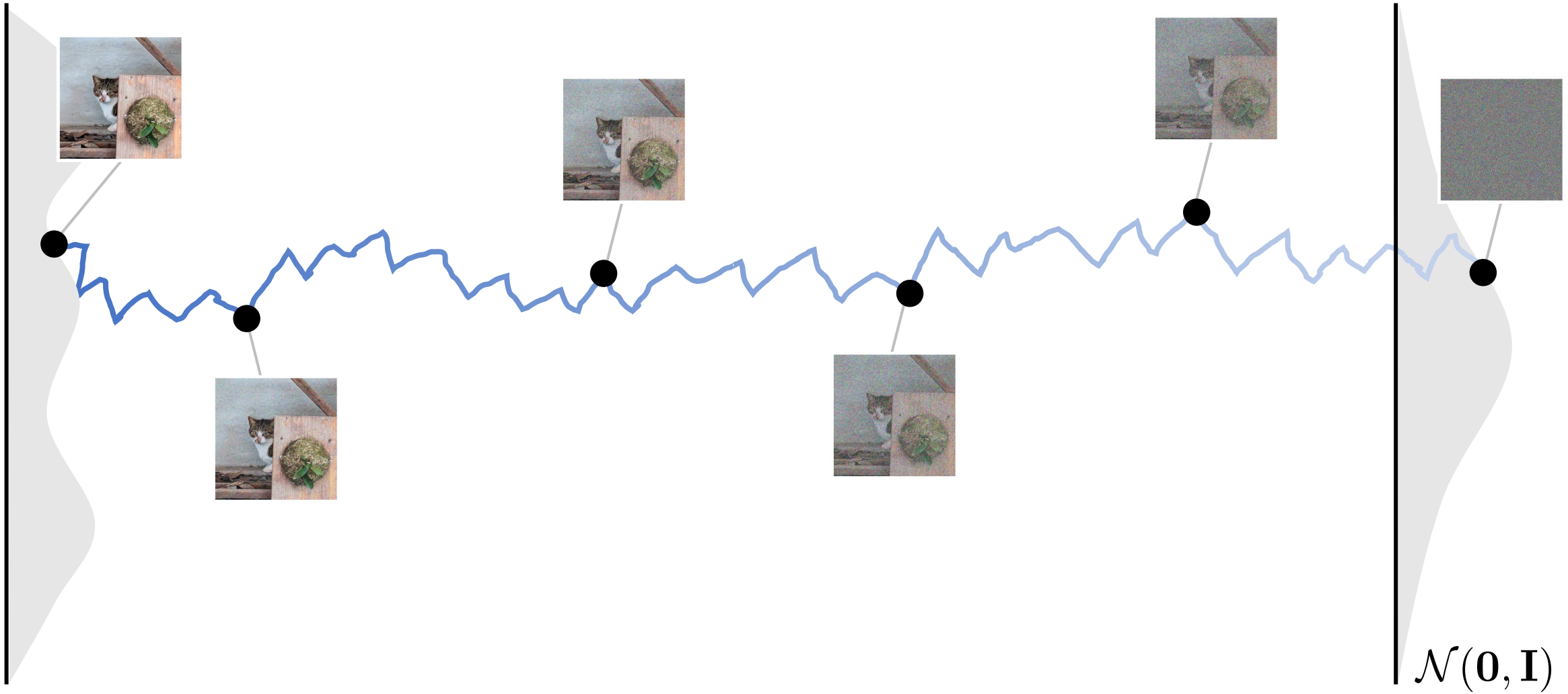
What is Diffusion Model?

Forward Process: add noise step by step, from data to pure noise

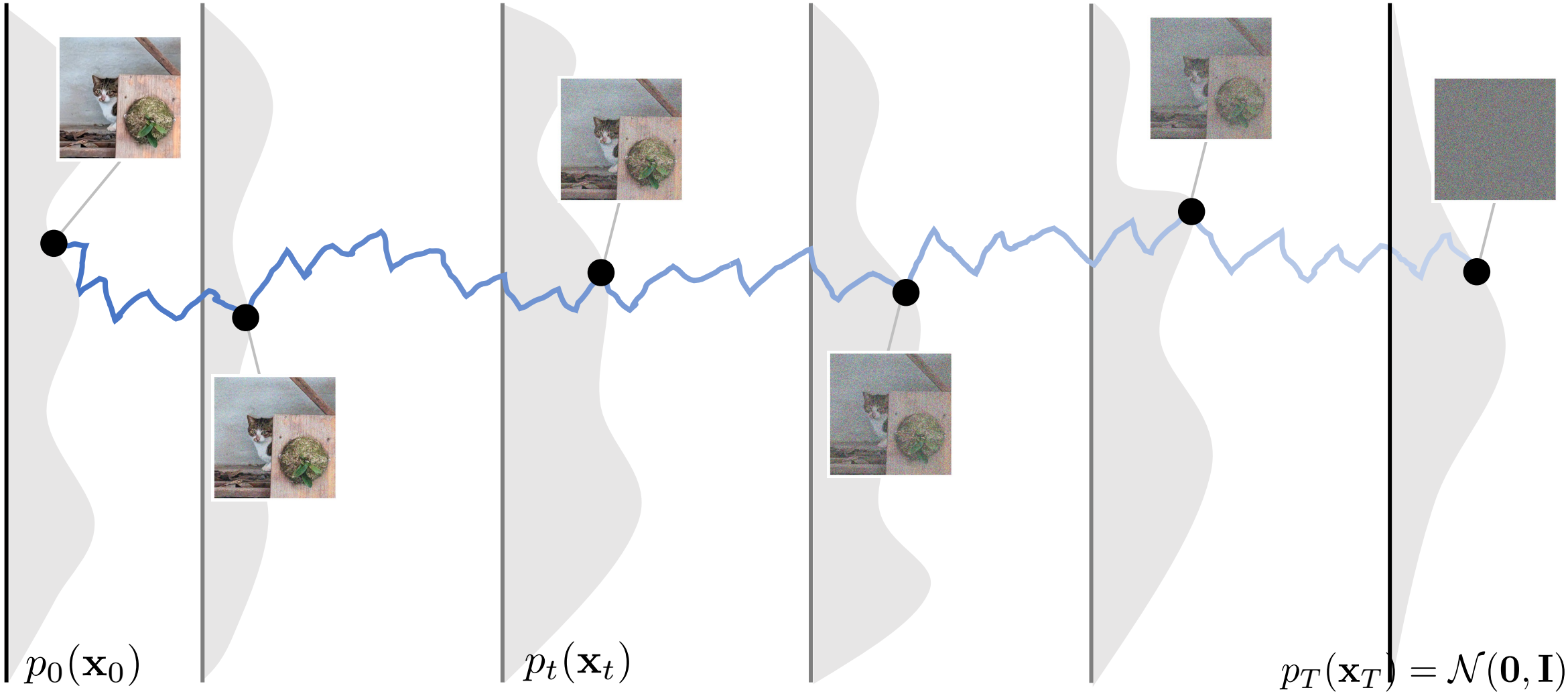
Reverse Process: generate data from pure noise by denoising

Creating noise from data is easy; creating data from noise is generative modeling – *Yang Song*

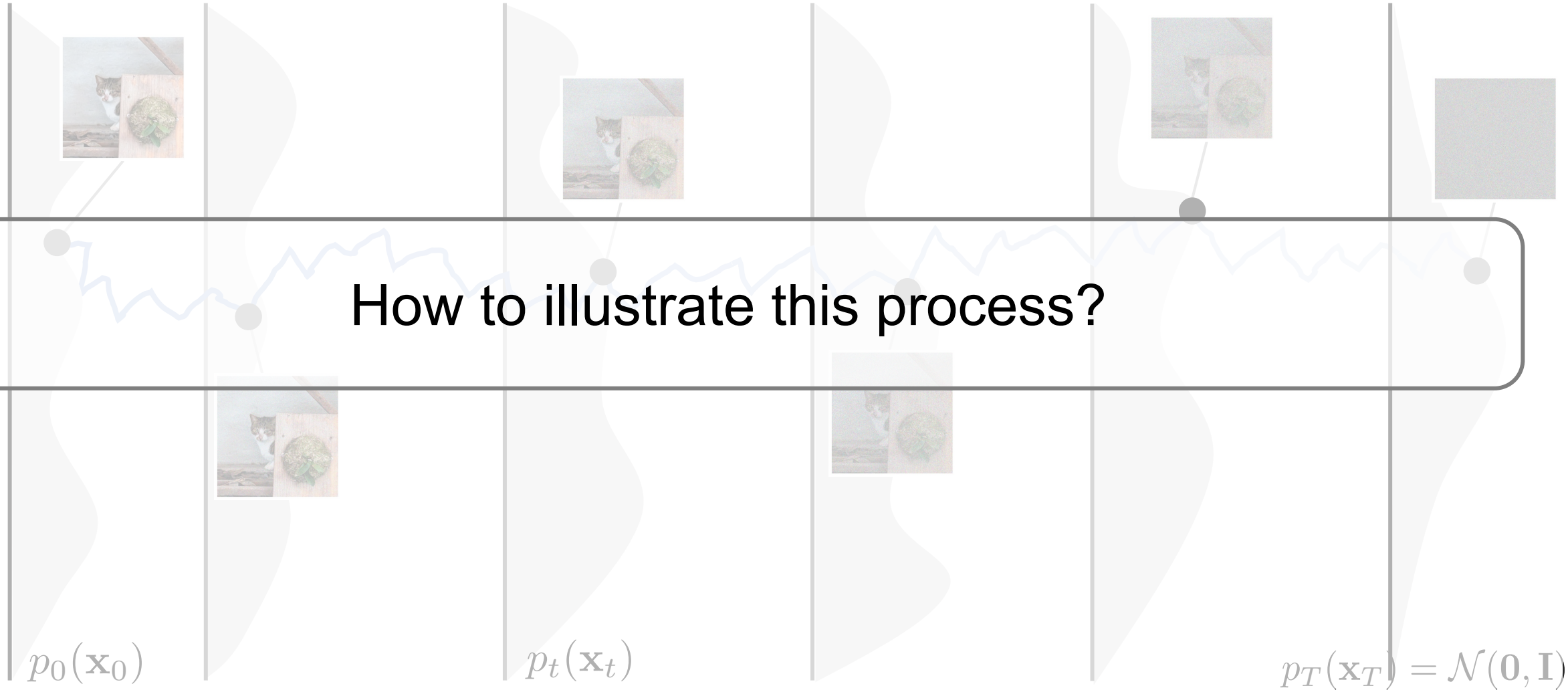
Diffusion Models



Diffusion Models

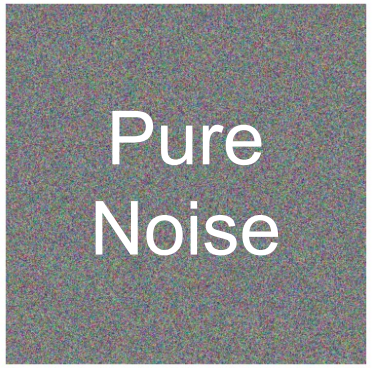


Diffusion Models

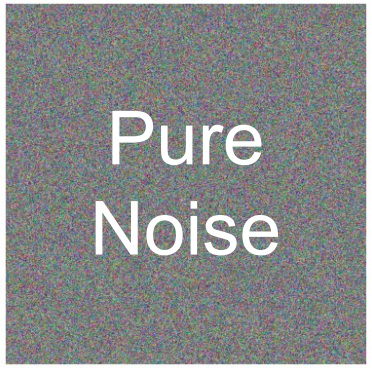


Mathematical Perspective of Adding Noise to Data

$$\sqrt{\alpha_1}\ \mathbf{x}_0 + \sqrt{1-\alpha_1}\ \mathbf{n} = \mathbf{x}_1$$



Mathematical Perspective of Adding Noise to Data

$\sqrt{\alpha_1}$  $+\sqrt{1-\alpha_1}$ 

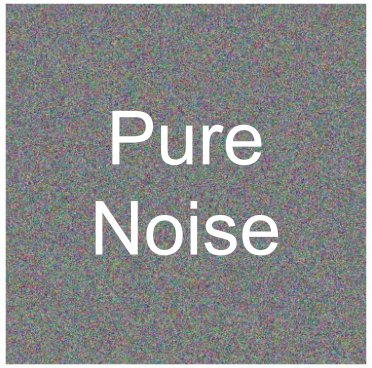
$=$ 

$\mathbf{x}_0$ $\mathbf{n}$ $\mathbf{x}_1$

- **Distribution Form:**

$$p(\mathbf{x}_1|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \sqrt{\alpha_1}\mathbf{x}_0, (1-\alpha_1)\mathbf{I})$$

Mathematical Perspective of Adding Noise to Data

$\sqrt{\alpha_1}$  $+\sqrt{1-\alpha_1}$ 

$=$ 

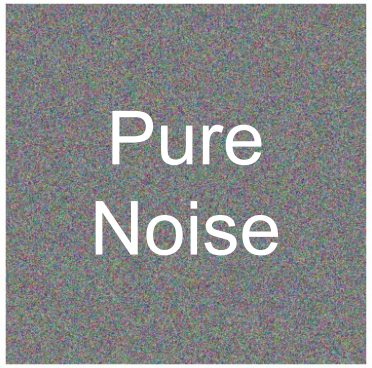
$\mathbf{x}_0$ $\mathbf{n}$ $\mathbf{x}_1$

- **Distribution Form:**

$$p(\mathbf{x}_1|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \sqrt{\alpha_1}\mathbf{x}_0, (1-\alpha_1)\mathbf{I})$$

Mean Covariance

Mathematical Perspective of Adding Noise to Data

$$\sqrt{\alpha_1}$$  $$+\sqrt{1-\alpha_1}$$ 

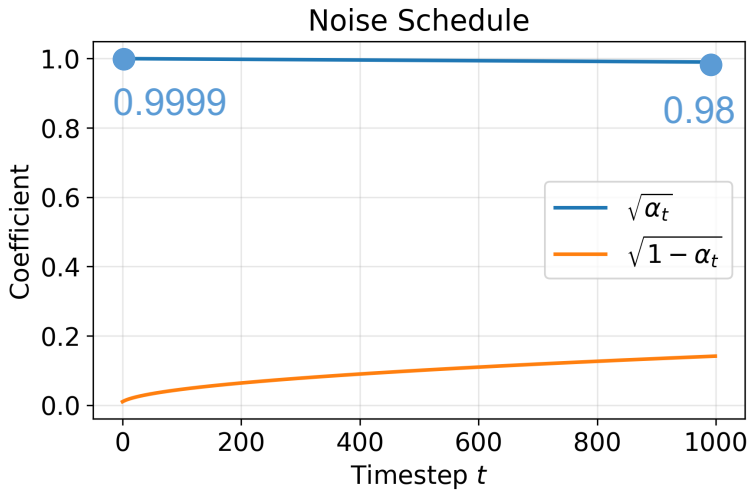
$$=$$ 

$$\mathbf{x}_0 \qquad \mathbf{n} \qquad \mathbf{x}_1$$

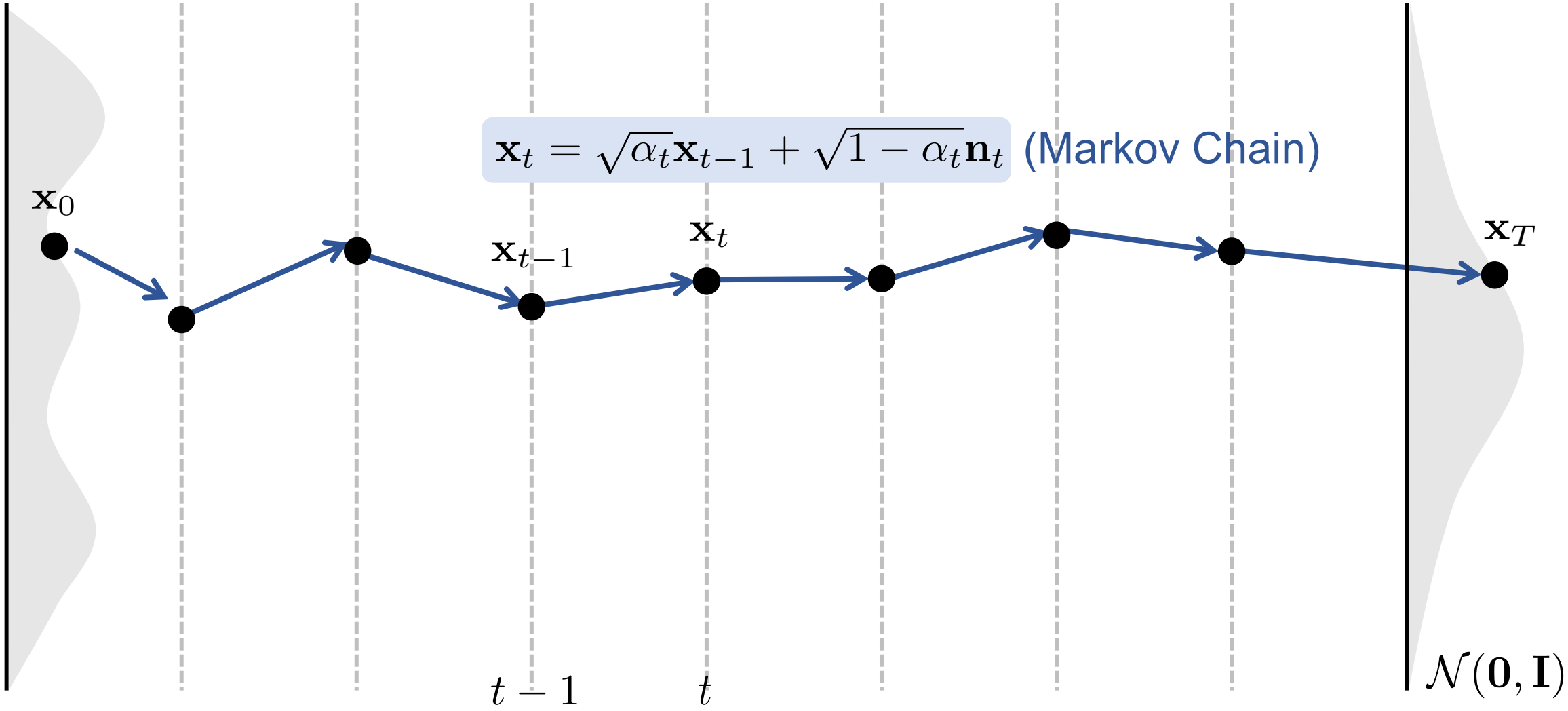
- **Distribution Form:**

$$p(\mathbf{x}_1|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \sqrt{\alpha_1}\mathbf{x}_0, (1-\alpha_1)\mathbf{I})$$

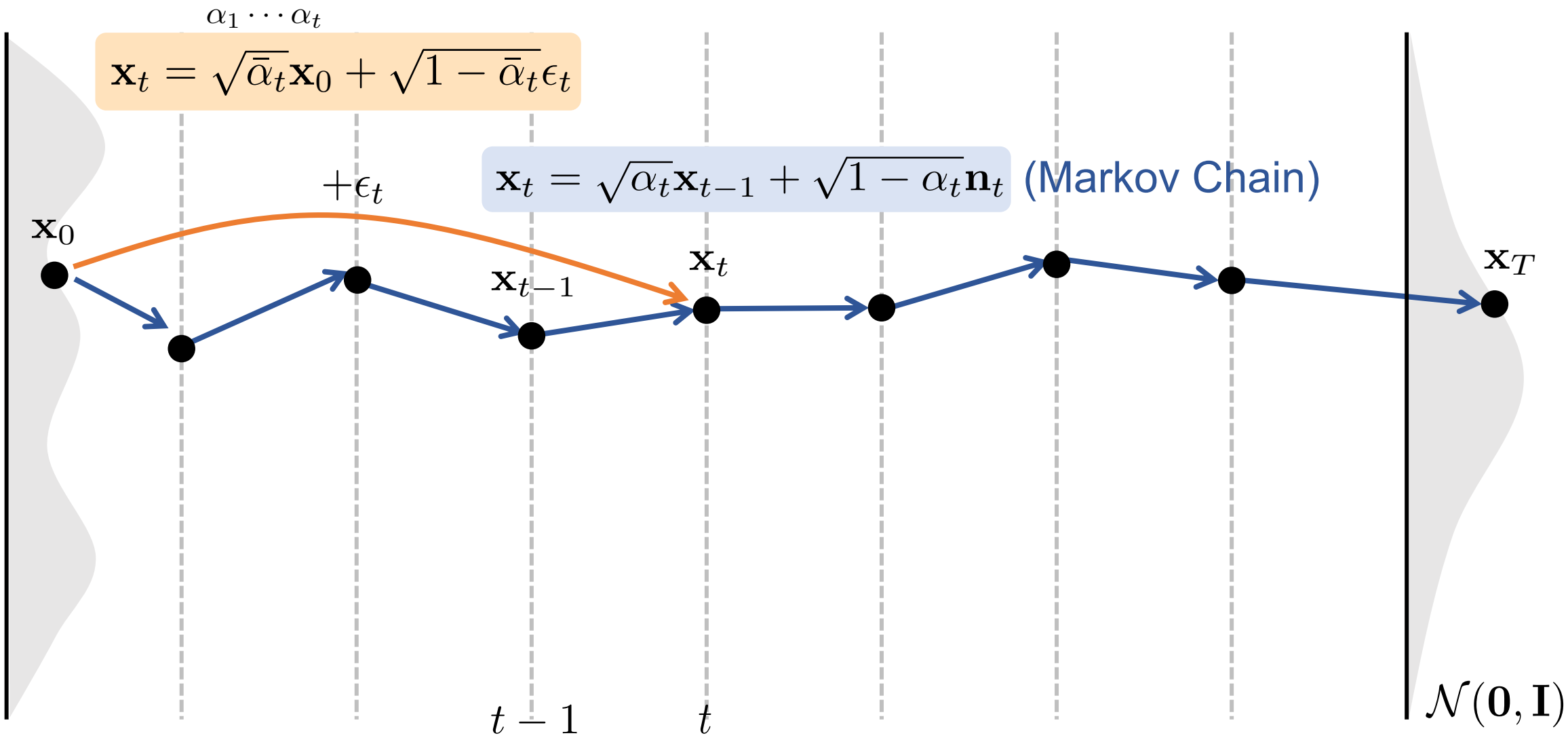
Mean Covariance



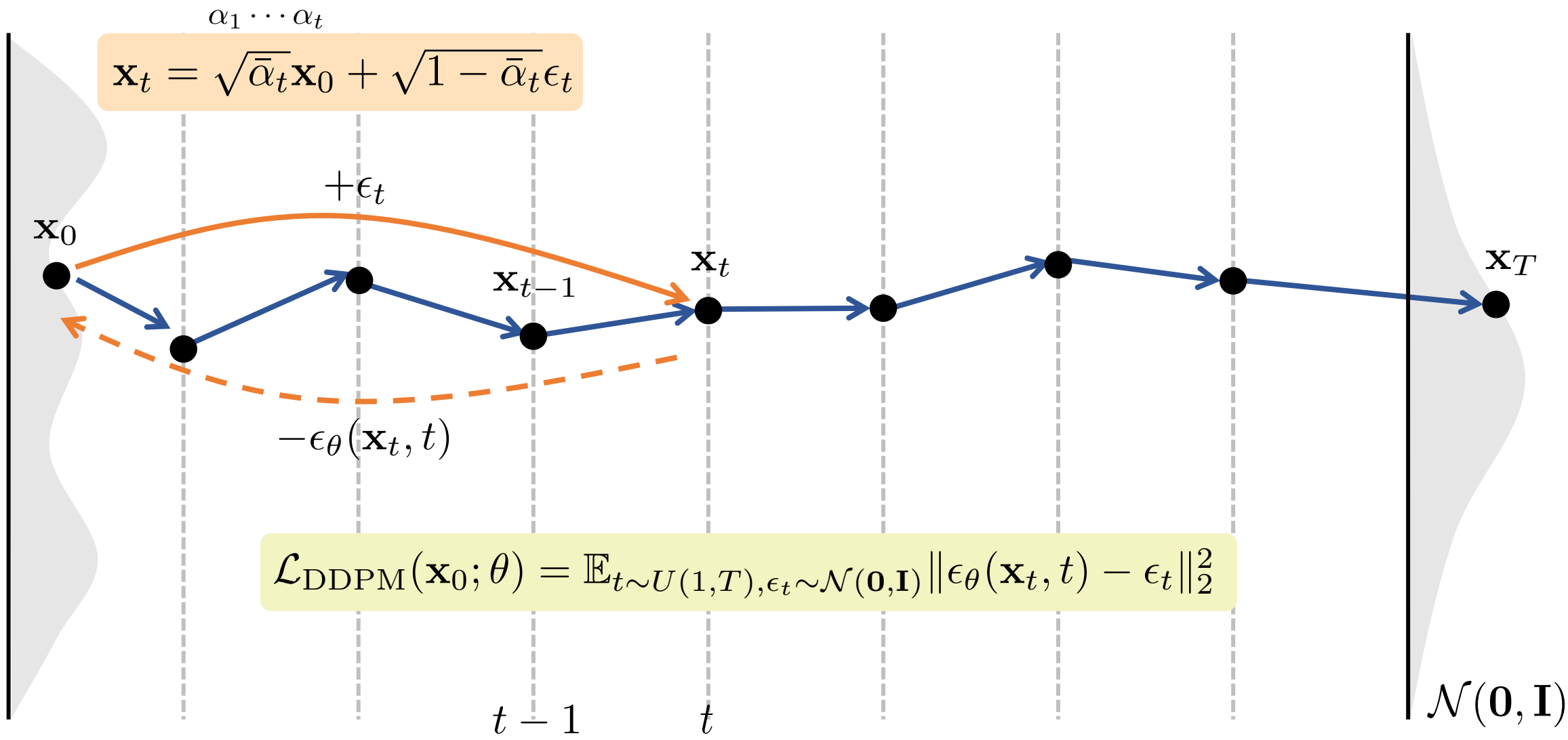
Denoising Diffusion Probabilistic Models (DDPM) [Ho+ NeurIPS'20]



Denoising Diffusion Probabilistic Models (DDPM) [Ho+ NeurIPS'20]



How Do We Train a DDPM?



How Does DDPM Generate Samples?

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1})}{p(\mathbf{x}_t)}$$

How Does DDPM Generate Samples?

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)p(\mathbf{x}_{t-1}|\mathbf{x}_0)}{p(\mathbf{x}_t|\mathbf{x}_0)}$$

How Does DDPM Generate Samples?

$$\mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$\mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})$$

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\, p(\mathbf{x}_{t-1}|\mathbf{x}_0)}{p(\mathbf{x}_t|\mathbf{x}_0)}$$

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

How Does DDPM Generate Samples?

$$\mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

$$\mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})$$

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\, p(\mathbf{x}_{t-1}|\mathbf{x}_0)}{p(\mathbf{x}_t|\mathbf{x}_0)}$$

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

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_t(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2\mathbf{I})$$

$$\mu_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{x}_0 \quad \text{and} \quad \sigma_t^2 = \frac{(1-\bar{\alpha}_{t-1})(1-\alpha_t)}{1-\bar{\alpha}_t}$$

How Does DDPM Generate Samples?

$\mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I})$

$\mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})$

$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \dfrac{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)p(\mathbf{x}_{t-1}|\mathbf{x}_0)}{p(\mathbf{x}_t|\mathbf{x}_0)}$

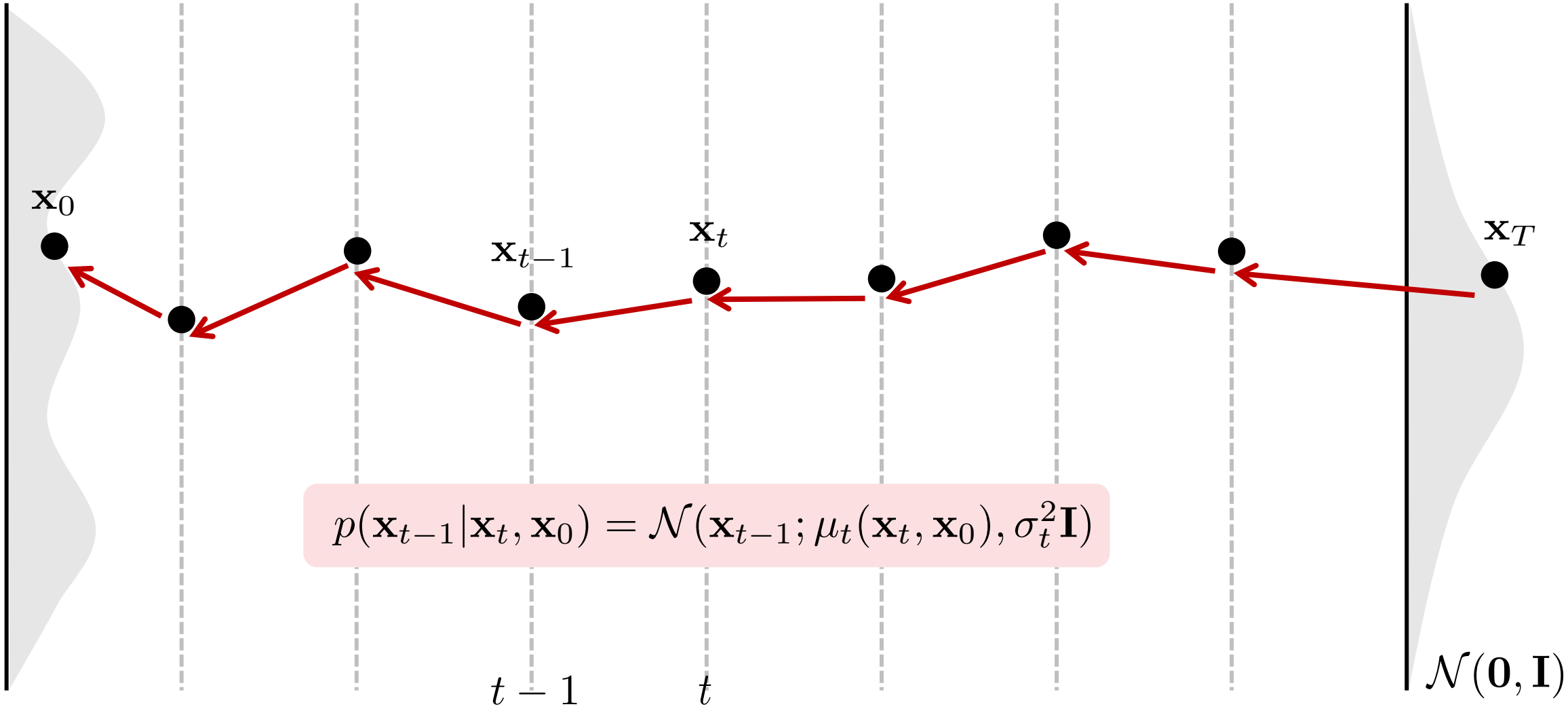
Is x_0 available during sampling?

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

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_t(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2\mathbf{I})$$

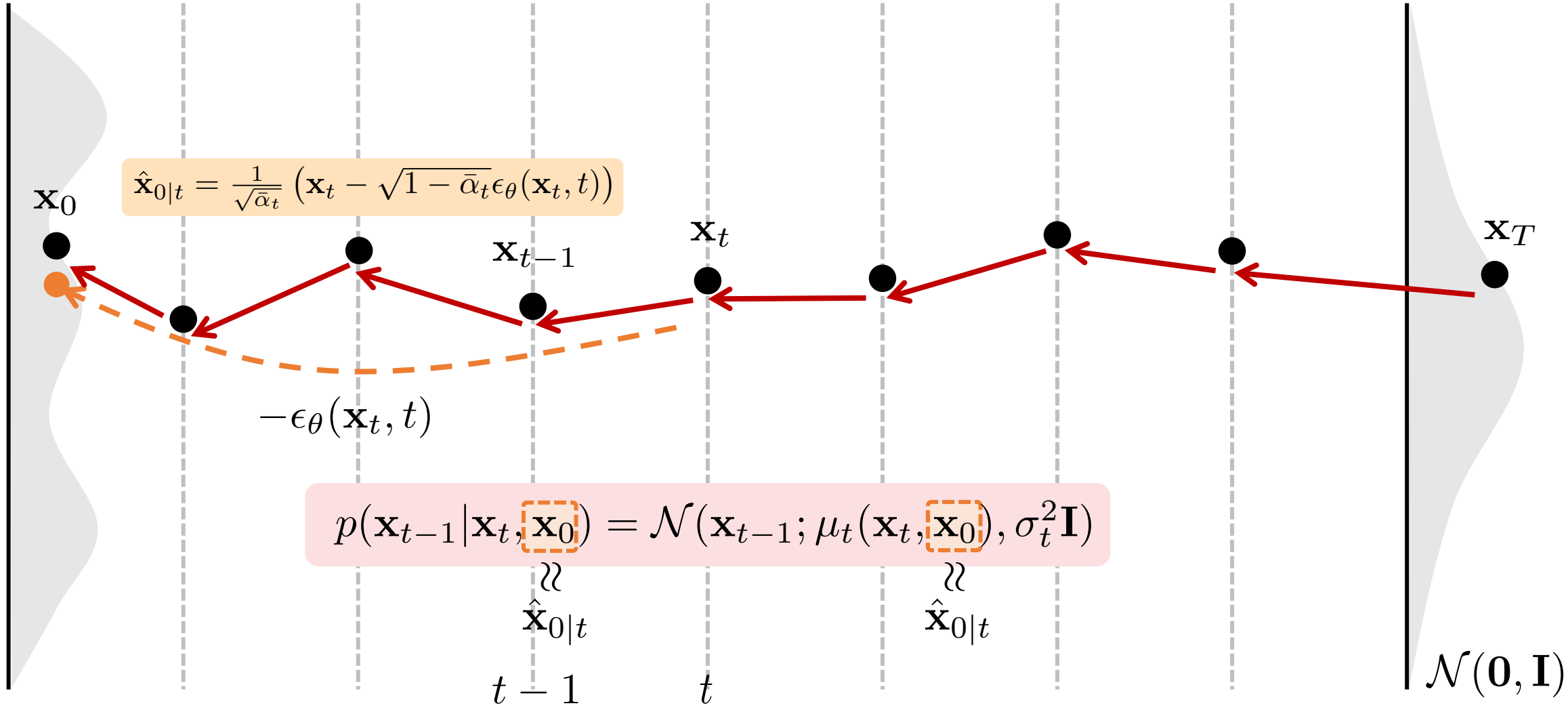
$$\mu_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t}\mathbf{x}_0 \quad \text{and} \quad \sigma_t^2 = \frac{(1-\bar{\alpha}_{t-1})(1-\alpha_t)}{1-\bar{\alpha}_t}$$

How Does DDPM Generate Samples?

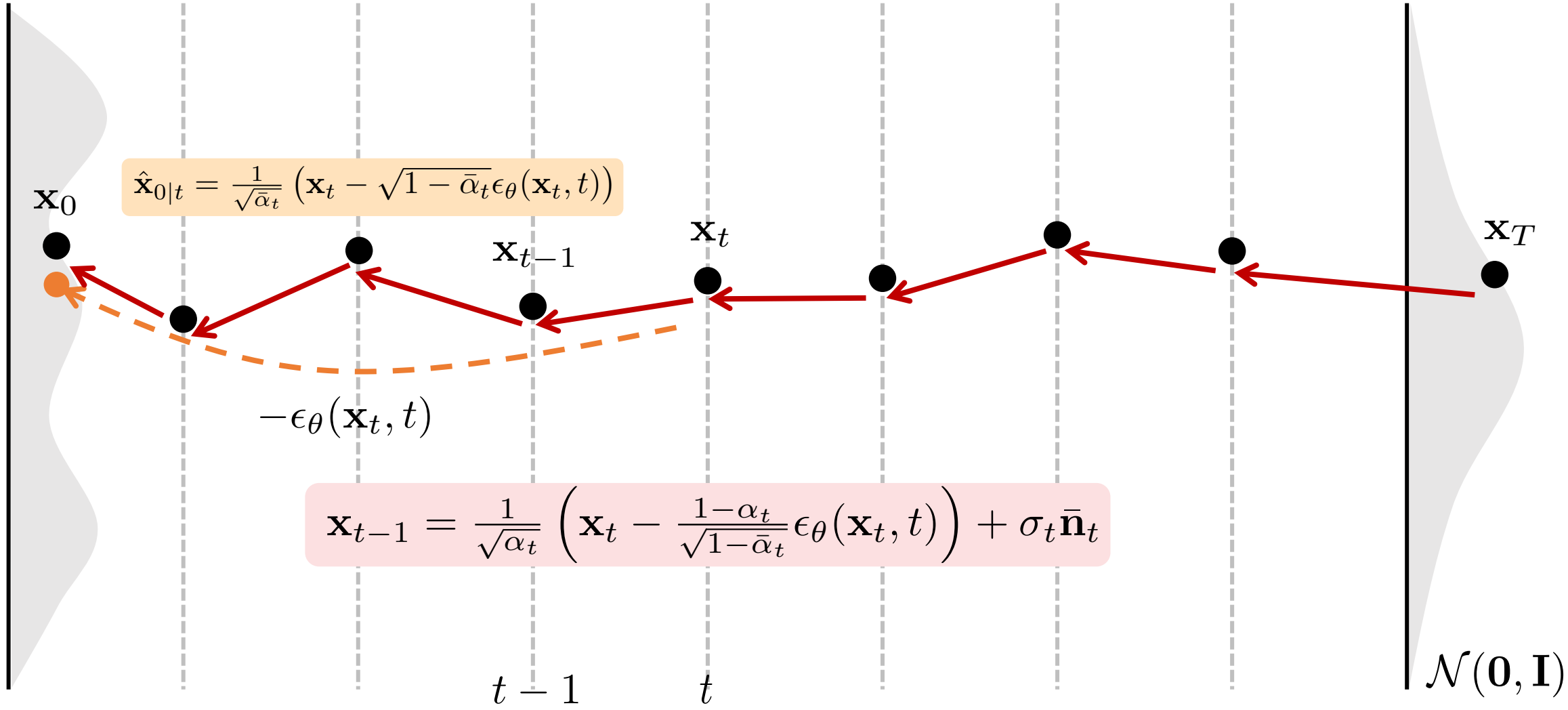


$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_t(\mathbf{x}_t, \mathbf{x}_0), \sigma_t^2 \mathbf{I})$$

How Does DDPM Generate Samples?



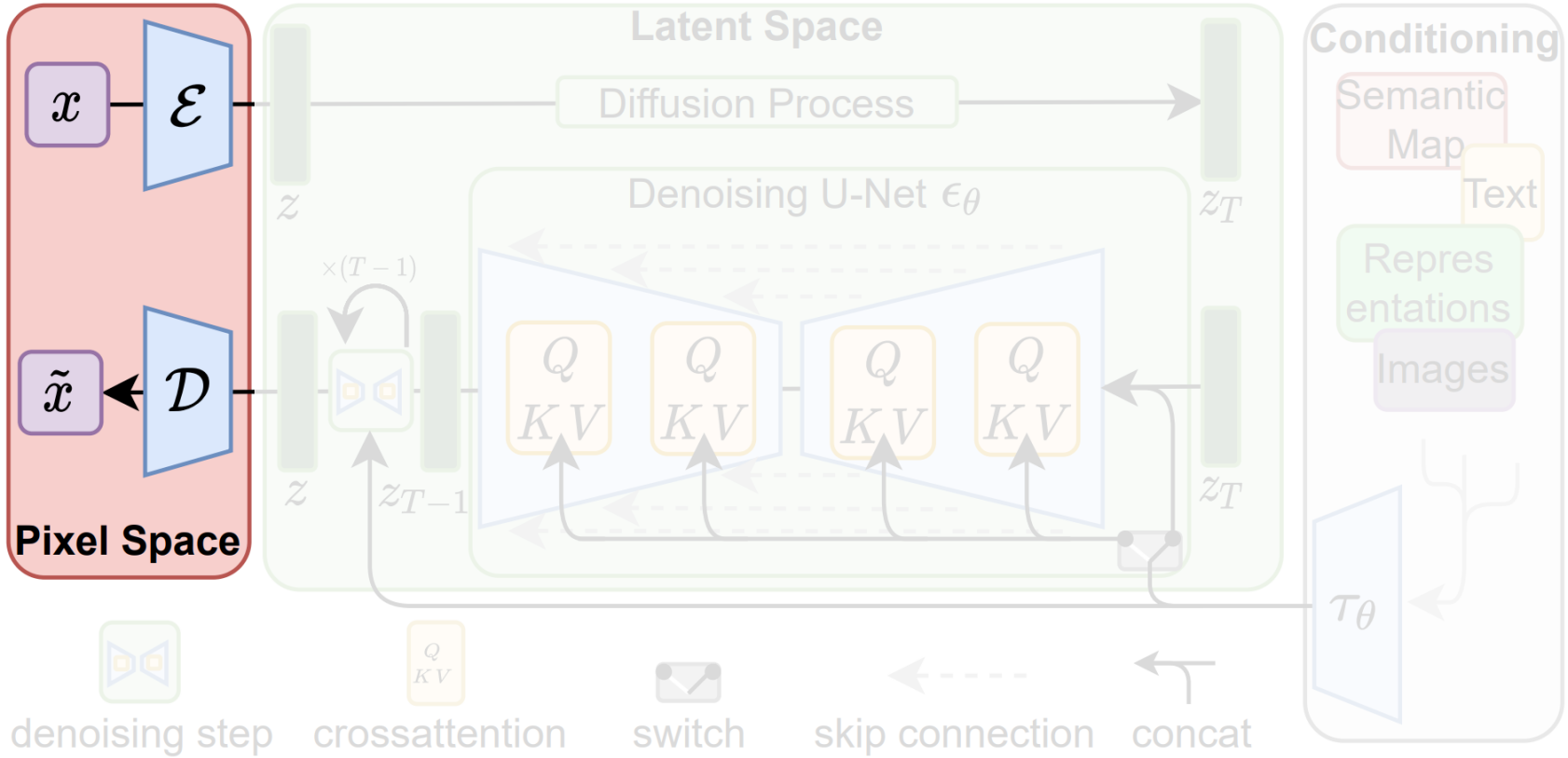
How Does DDPM Generate Samples?



How to reduce the computational cost of diffusion models?

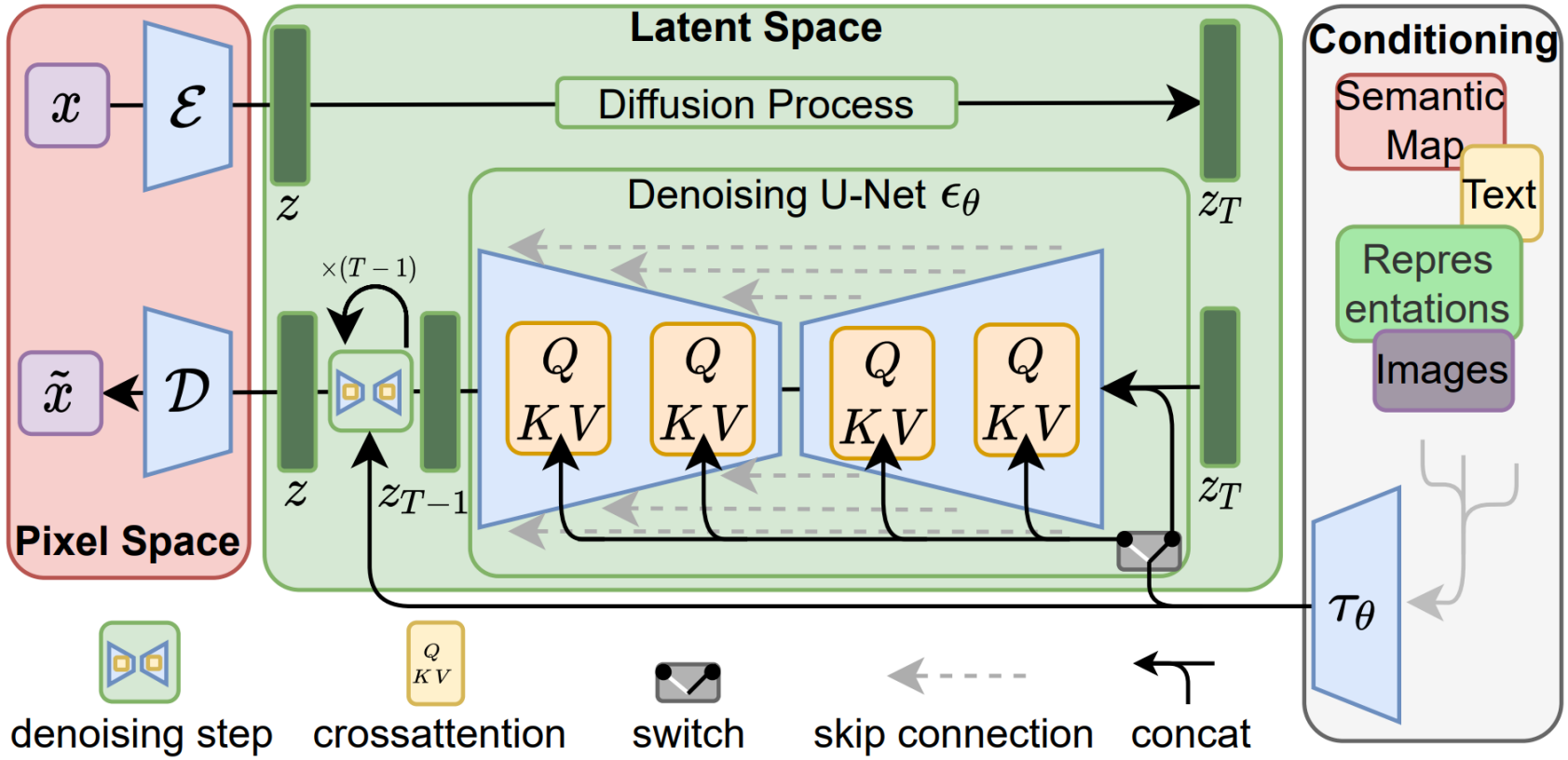
Latent Diffusion Model (LDM) [Rombach+ CVPR'22]

Use the pretrained-VAE to compress the image to latent reducing computation time



Latent Diffusion Model (LDM) [Rombach+ CVPR'22]

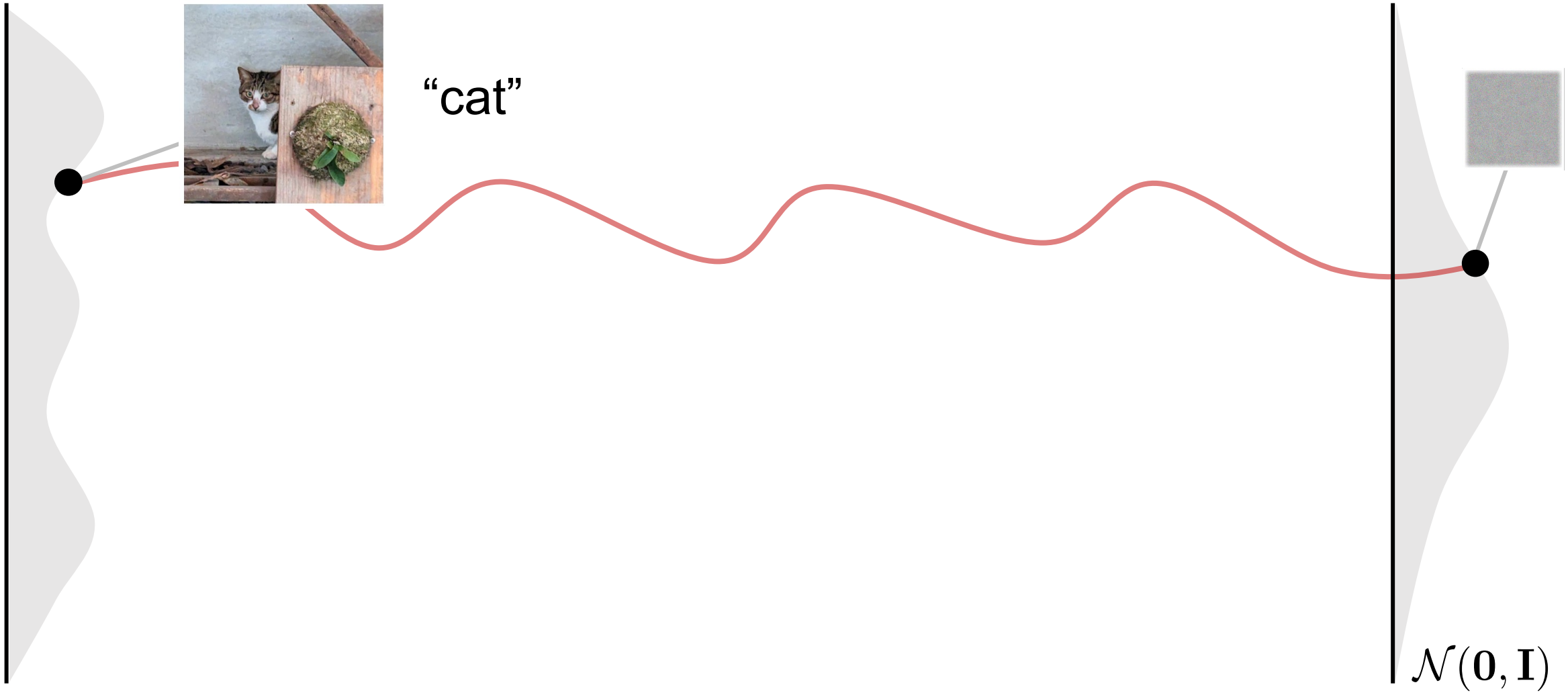
Use the pretrained-VAE to compress the image to latent reducing computation time



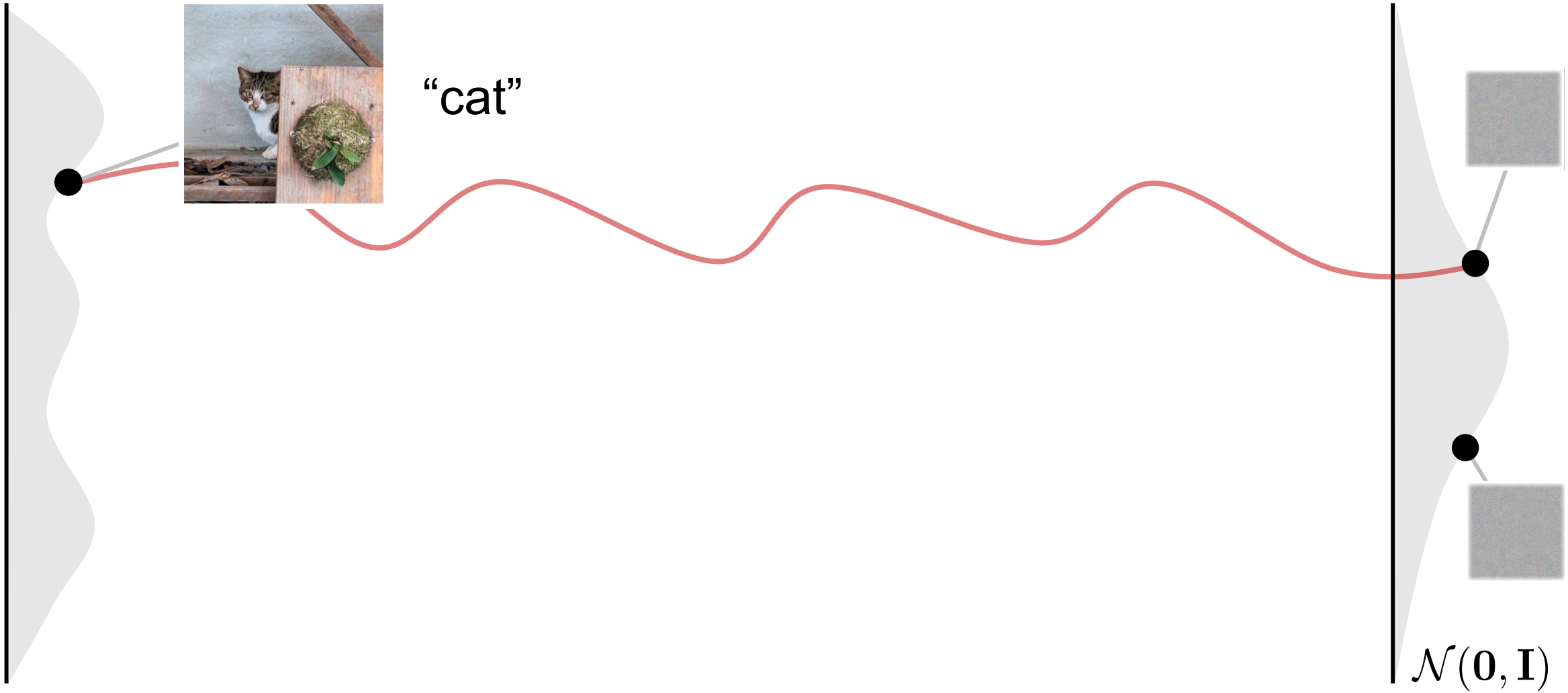
Can we control the Diffusion Models?

Conditional Diffusion Models

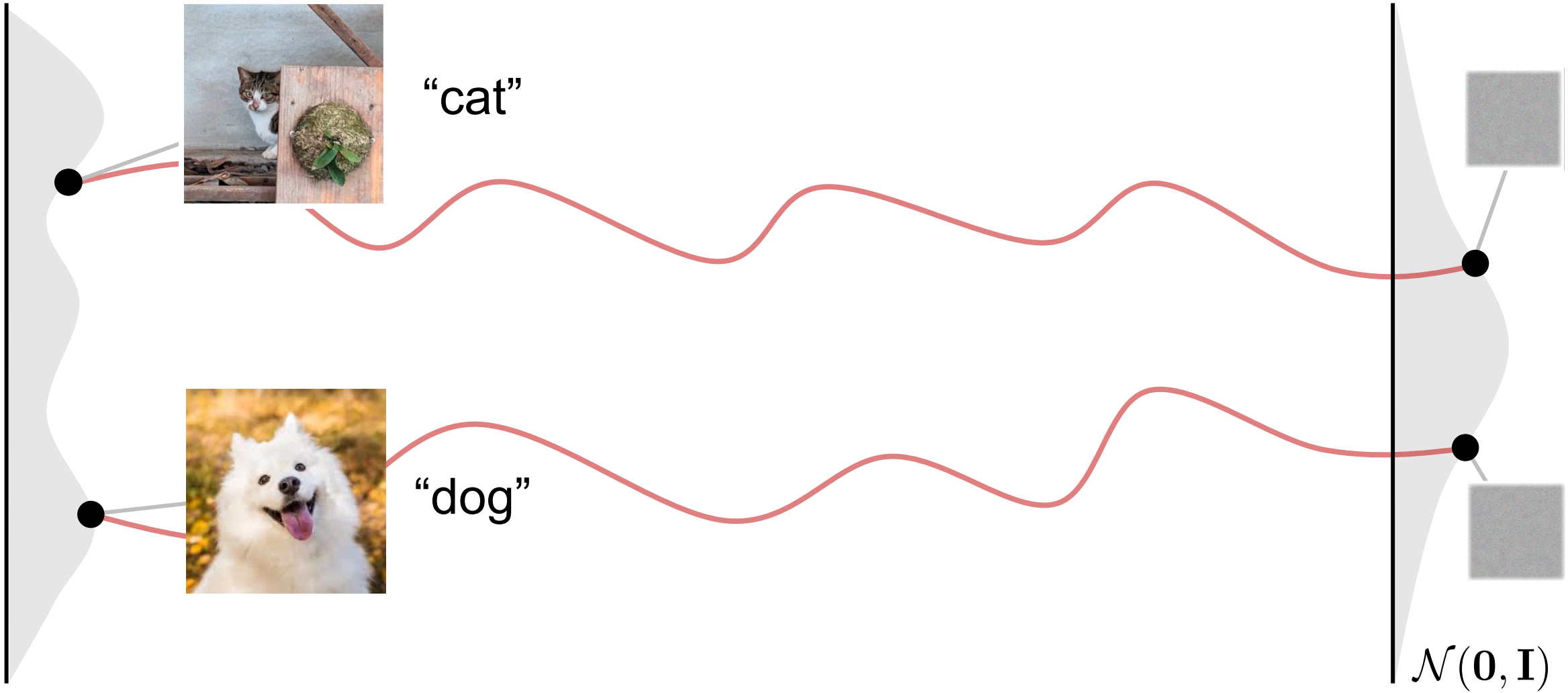
Conditional Diffusion Models



Conditional Diffusion Models



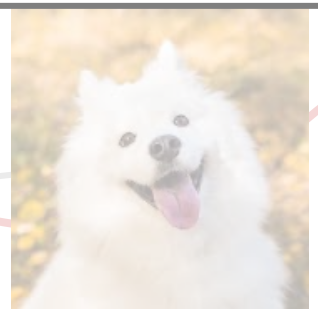
Conditional Diffusion Models



Conditional Diffusion Models

“cat”

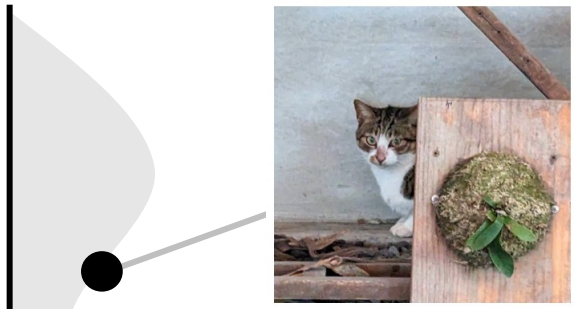
How to control the Diffusion Models?



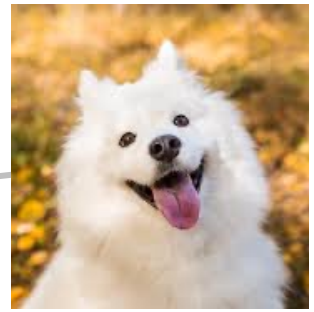
“dog”

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

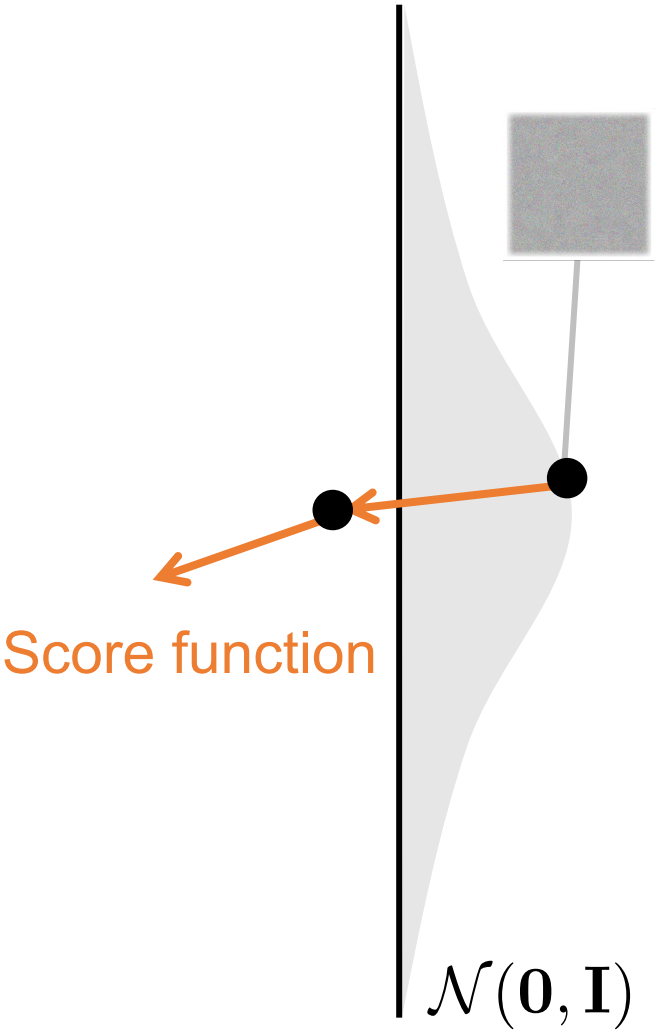
Conditional Diffusion Models



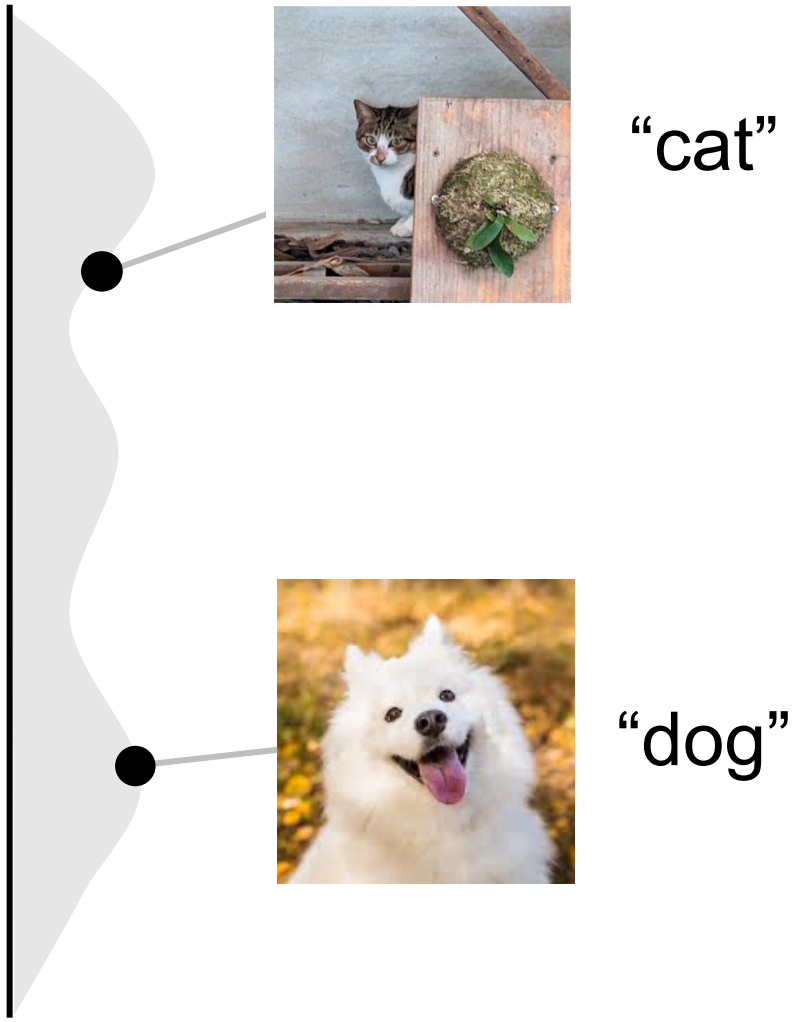
“cat”

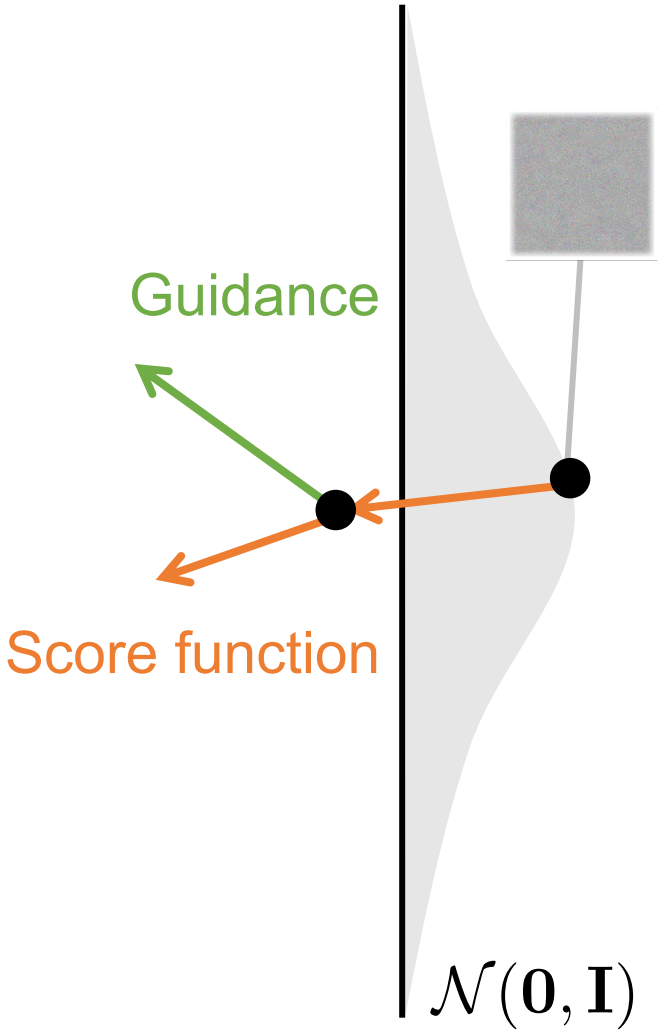


“dog”

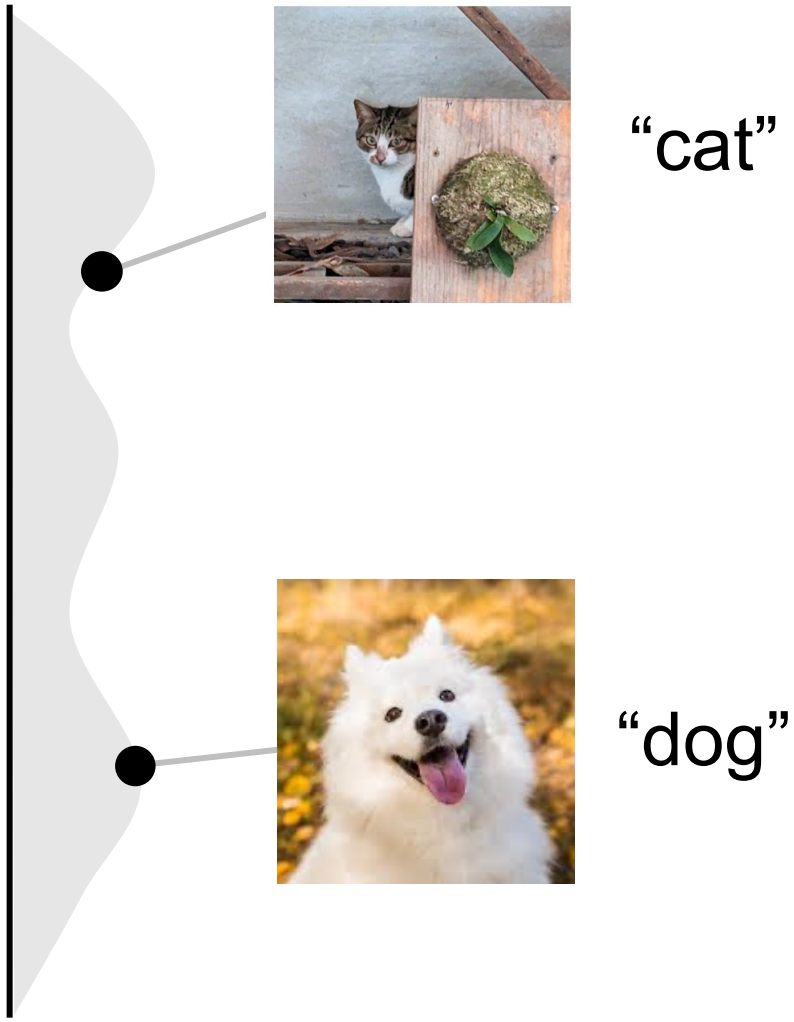


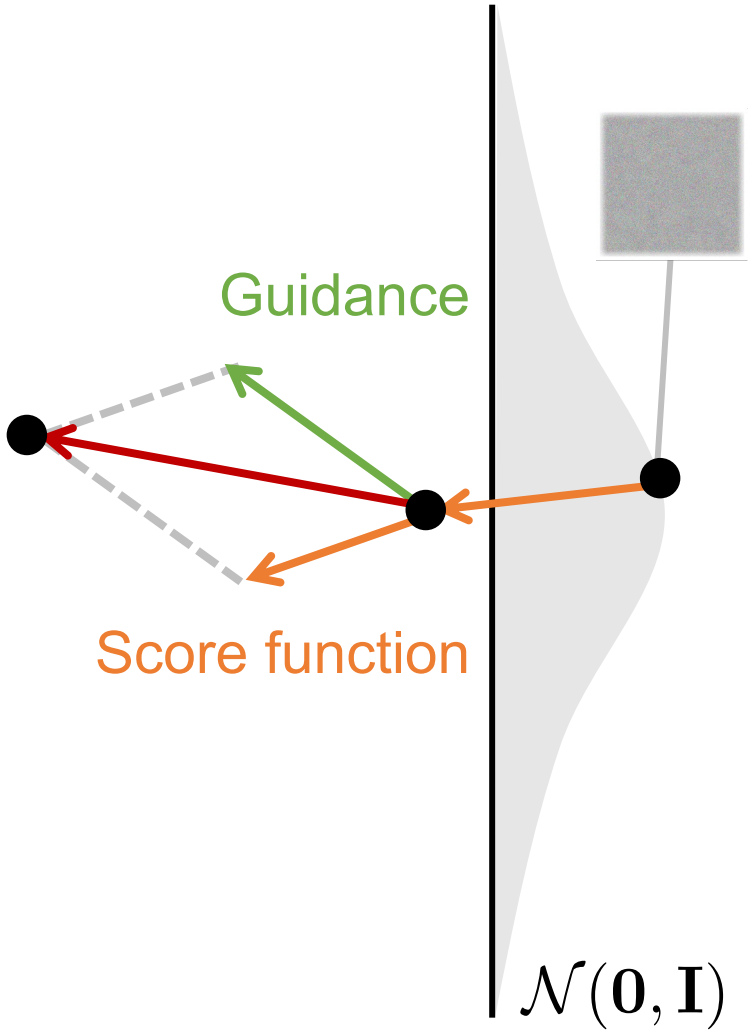
Conditional Diffusion Models



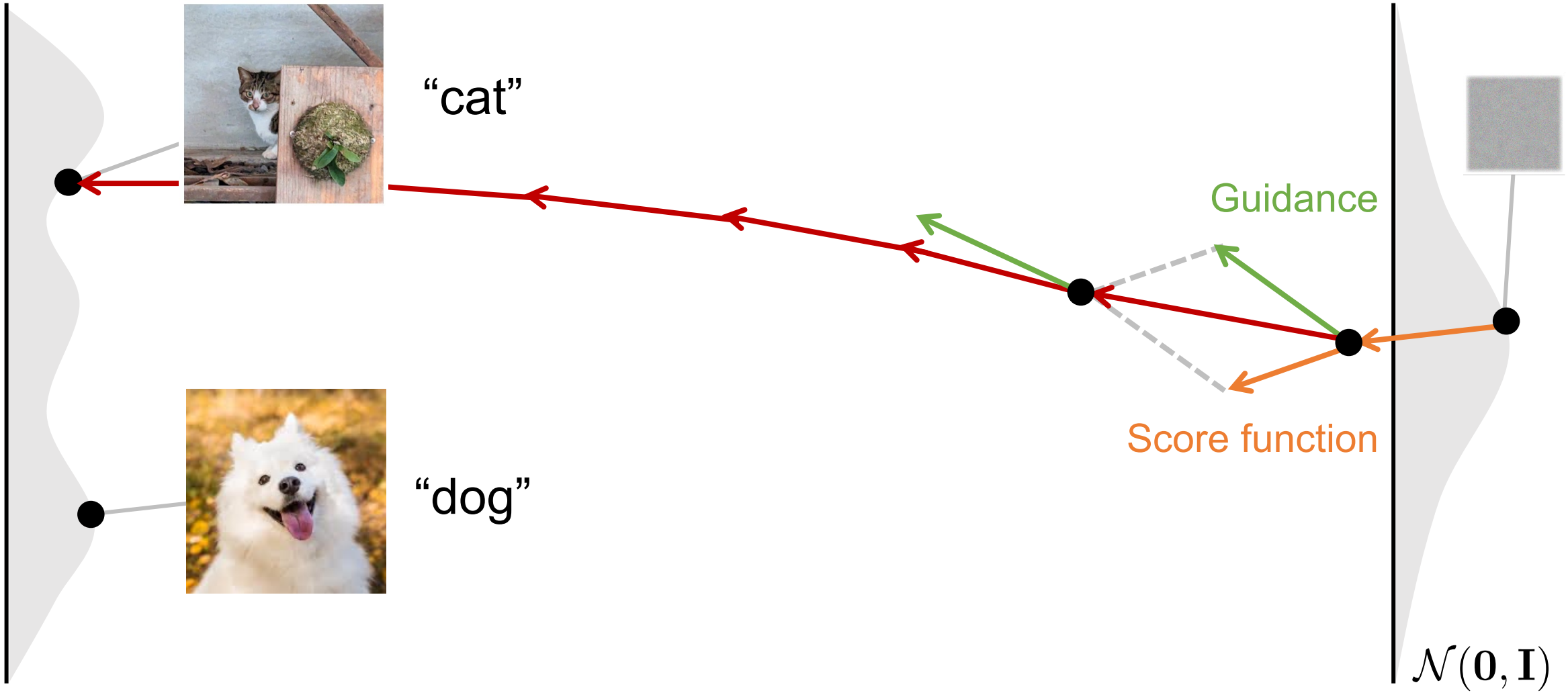


Conditional Diffusion Models





Conditional Diffusion Models



Classifier Guidance (CG) [Dhariwal+ NeurIPS'21]

- **Bayes' Rule:**

$$P\left(\mathbf{x}_t = \text{  } \;\middle|\; \mathbf{y} = \text{“cat”}\right)$$

Classifier Guidance (CG) [Dhariwal+ NeurIPS'21]

- **Bayes' Rule:**

$$P\left(\mathbf{x}_t = \text{  } \,\middle|\, \mathbf{y} = \text{“cat”}\right) = \frac{p(\mathbf{x}_t)p(\mathbf{y}|\mathbf{x}_t)}{p(\mathbf{y})}$$

Classifier Guidance (CG) [Dhariwal+ NeurIPS'21]

- **Bayes' Rule:**

$$P\left(\mathbf{x}_t = \text{  } \,\middle|\, \mathbf{y} = \text{“cat”}\right) = \frac{p(\mathbf{x}_t)p(\mathbf{y}|\mathbf{x}_t)}{p(\mathbf{y})}$$

- **Bayes' Rule for Score Function:**

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{y})$$

Classifier Guidance (CG) [Dhariwal+ NeurIPS'21]

- **Bayes' Rule:**

$$P\left(\mathbf{x}_t = \text{  } \;\middle|\; \mathbf{y} = \text{“cat”}\right) = \frac{p(\mathbf{x}_t)p(\mathbf{y}|\mathbf{x}_t)}{p(\mathbf{y})}$$

- **Bayes' Rule for Score Function:**

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{y})$$

0

Unconditional score

$\propto$

$-\epsilon_\theta(\mathbf{x}_t, t)$

Classifier

(Need to additional training)

Classifier-Free Guidance (CFG) [Ho+ NeurIPSW'21]

From CG, we have

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Reweight the coefficient between unconditional score and classifier score

$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) := \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

Classifier-Free Guidance (CFG) [Ho+ NeurIPSW'21]

From CG, we have

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Reweight the coefficient between unconditional score and classifier score

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) &:= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \\ &= \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) + (1-\gamma) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\end{aligned}$$

Classifier-Free Guidance (CFG) [Ho+ NeurIPSW'21]

From CG, we have

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Reweight the coefficient between unconditional score and classifier score

$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) := \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

$$= \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) + (1-\gamma) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Conditional score

Unconditional score

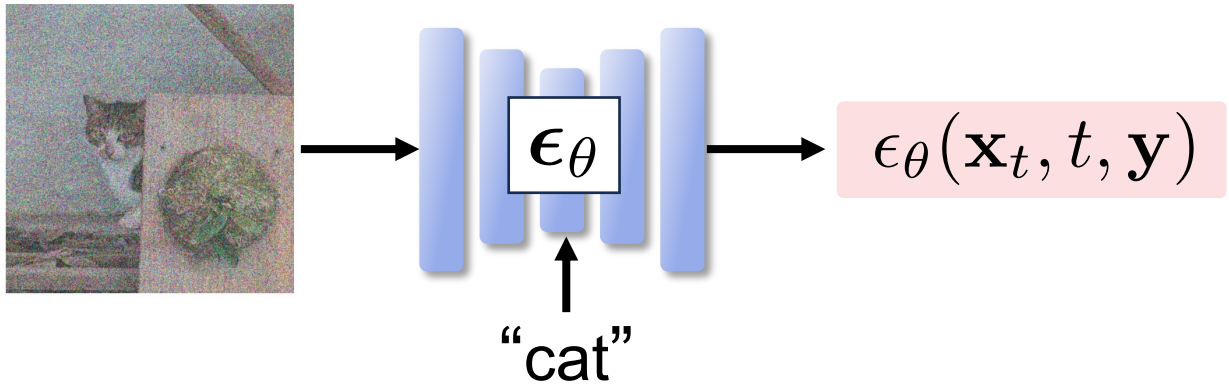
Training CFG

$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) = \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) + (1-\gamma) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Conditional score

Unconditional score

Predict Conditional Score



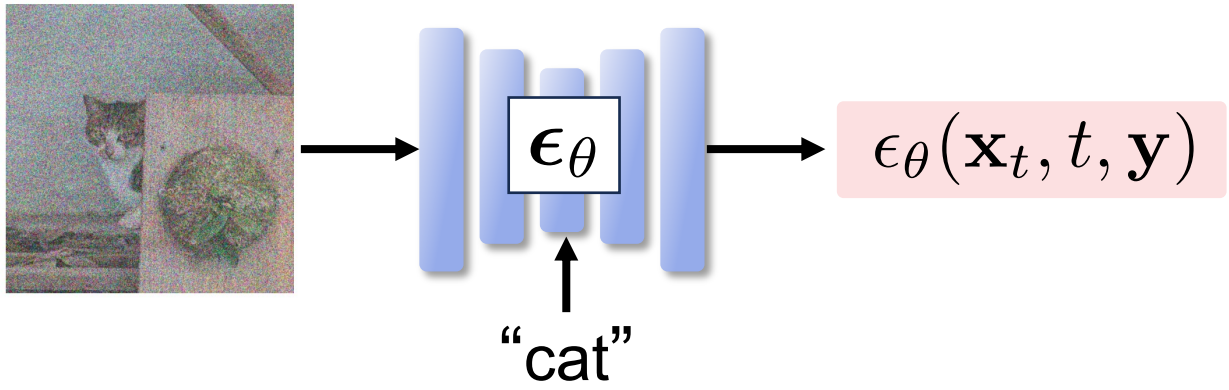
Training CFG

$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) = \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) + (1-\gamma) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

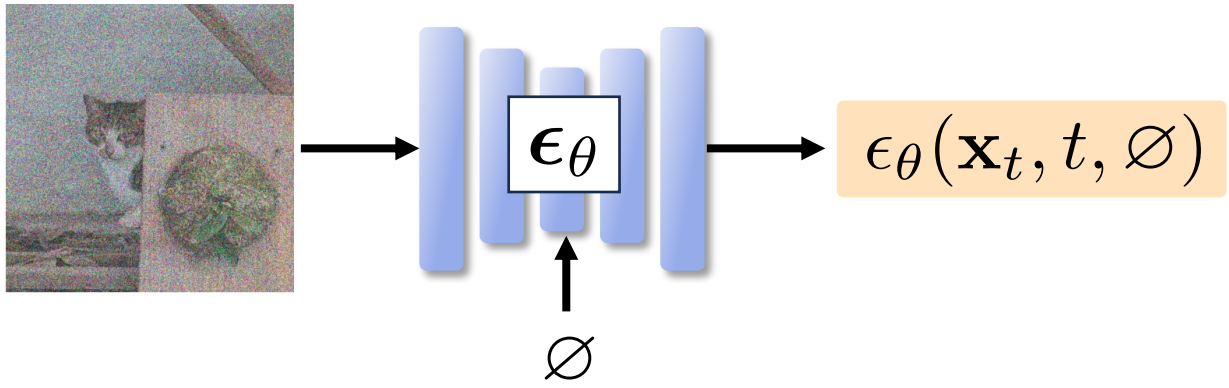
Conditional score

Unconditional score

Predict Conditional Score



Predict Unconditional Score



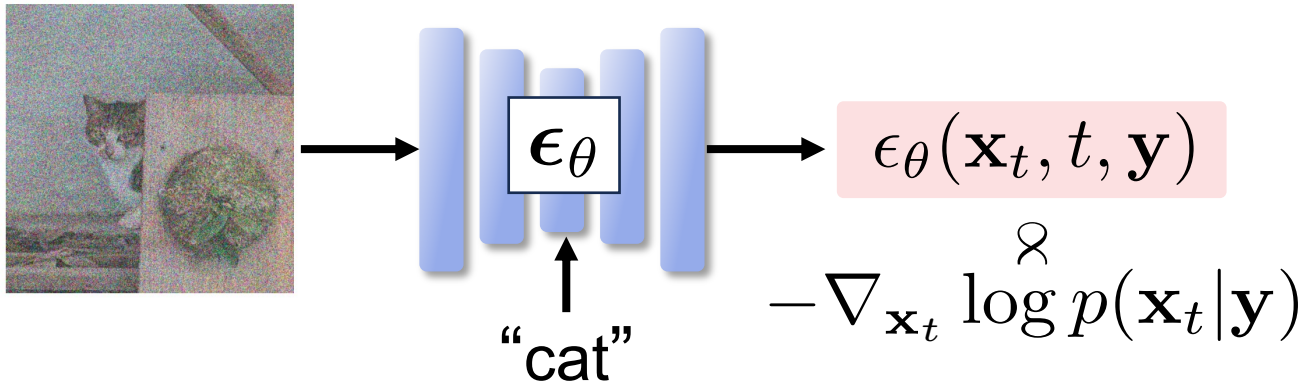
Training CFG

$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) = \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) + (1-\gamma) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

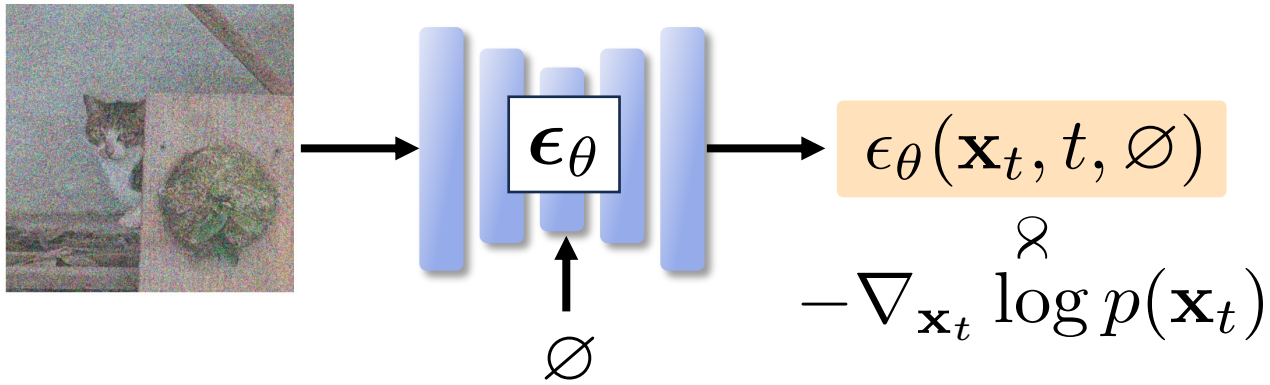
Conditional score

Unconditional score

Predict Conditional Score



Predict Unconditional Score



Training CFG

$$\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{y}) = \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) + (1-\gamma) \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$$

Conditional score

Unconditional score

Predict Conditional Score

ϵ_θ

"cat"

$\epsilon_\theta(\mathbf{x}_t, t, \mathbf{y})$

$-\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$

Can we control the semantic details of an image?

Predict Unconditional Score

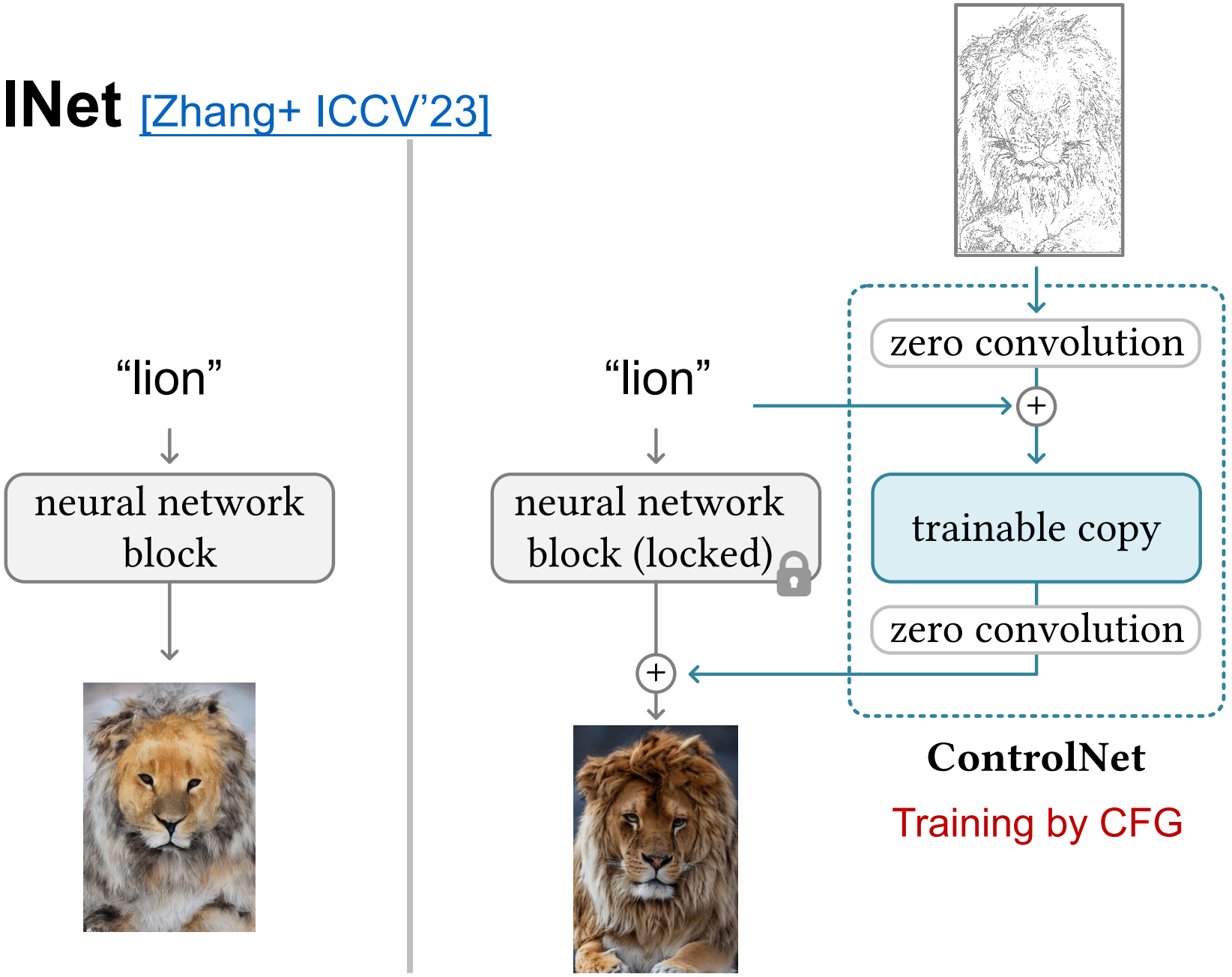
ϵ_θ

$\varnothing$

$\epsilon_\theta(\mathbf{x}_t, t, \varnothing)$

$-\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$

ControlNet [Zhang+ ICCV'23]



Summary

- Diffusion models build a bridge between noise and data, forming a powerful generative modeling framework.
- Conditional methods like CFG and ControlNet significantly improve controllability and application scope.

WACV 2025
TUCSON, ARIZONA • FEB 28 - MAR 4

DiffQRCoder: Diffusion-based Aesthetic QR Code Generation with Scanning Robustness Guided Iterative Refinement


Jia-Wei Liao[1,2]


Winston Wang[2,*]

Tzu-Sian Wang[2,*]


Li-Xuan Peng[2,*]


Ju-Hsuan Weng[1,2]


Cheng-Fu Chou[1]


Jun-Cheng Chen[2]

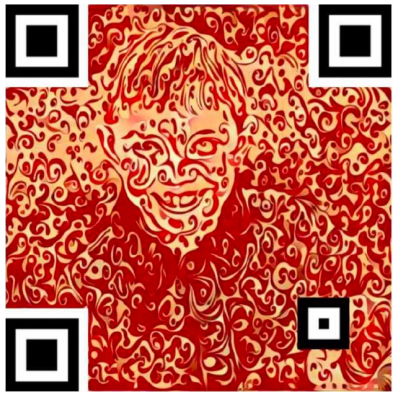
[1] National Taiwan University,
[2] Research Center for Information Technology Innovation, Academia Sinica

Aesthetic QR Code

Traditional Methods

Qart

ArtCoder



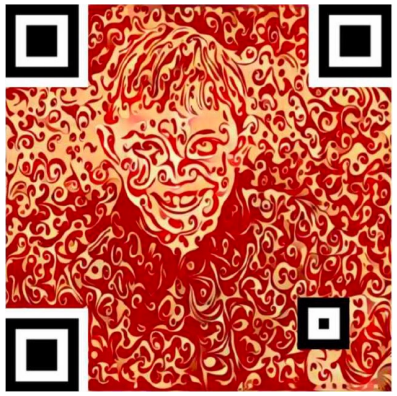
Q-Art Code

Aesthetic QR Code

Traditional Methods

Qart

ArtCoder



Q-Art Code

Generative-based Method

DiffQRCoder (Ours)

Winter wonderland, fresh snowfall, evergreen trees, cozy log cabin, smoke rising from chimney, aurora borealis in night sky.

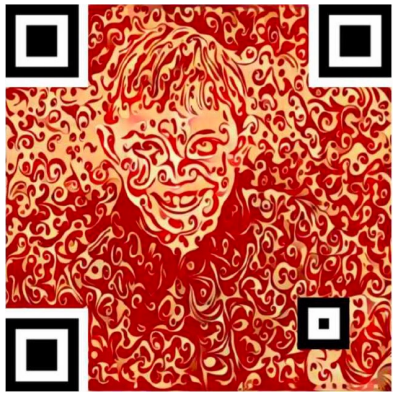
Cherry blossom festival, pink petals floating in the air, traditional lanterns, peaceful river, people in kimonos, sunny day.

Aesthetic QR Code

Traditional Methods

Qart

ArtCoder



Q-Art Code

Generative-based Method

DiffQRCoder (Ours)

Winter wonderland, fresh snowfall, evergreen trees, cozy log cabin, smoke rising from chimney, aurora borealis in night sky.

Cherry blossom festival, pink petals floating in the air, traditional lanterns, peaceful river, people in kimonos, sunny day.

QR Code ✚ Prompt ✚ Diffusion Model ⟶ Next-Generation Aesthetic QR Code

Motivation & Challenge

Most Diffusion-based aesthetic QR code generation struggle to balance scannability and aesthetics.

Green: scannable, **Red:** unscannable

Motivation & Challenge

Most Diffusion-based aesthetic QR code generation struggle to balance scannability and aesthetics.

- QR Code AI Art and QR Diffusion produce better scanning robust QR codes but are visually less appealing.

QR Code AI Art

QR Diffusion

Green: scannable, **Red:** unscannable

Motivation & Challenge

Most Diffusion-based aesthetic QR code generation struggle to balance scannability and aesthetics.

- QR Code AI Art and QR Diffusion produce better scanning robust QR codes but are visually less appealing.
- QRBTF could generate visually appealing QR codes, however, they lack scanning robustness.

QR Code AI Art | QR Diffusion | QRBTF | DiffQRCode (Ours)



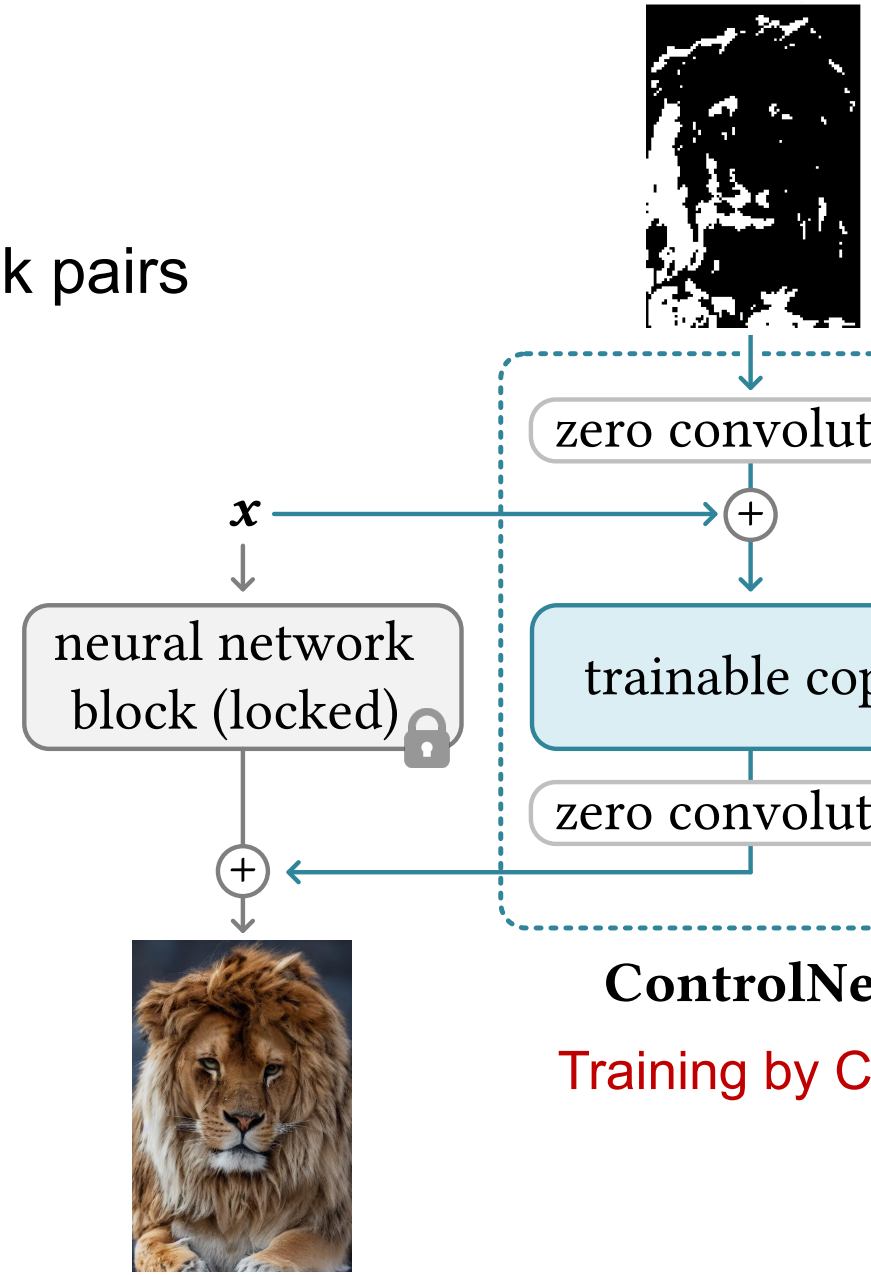

Green: scannable, **Red:** unscannable

Previous Method

Training ControlNet with image–binary mask pairs

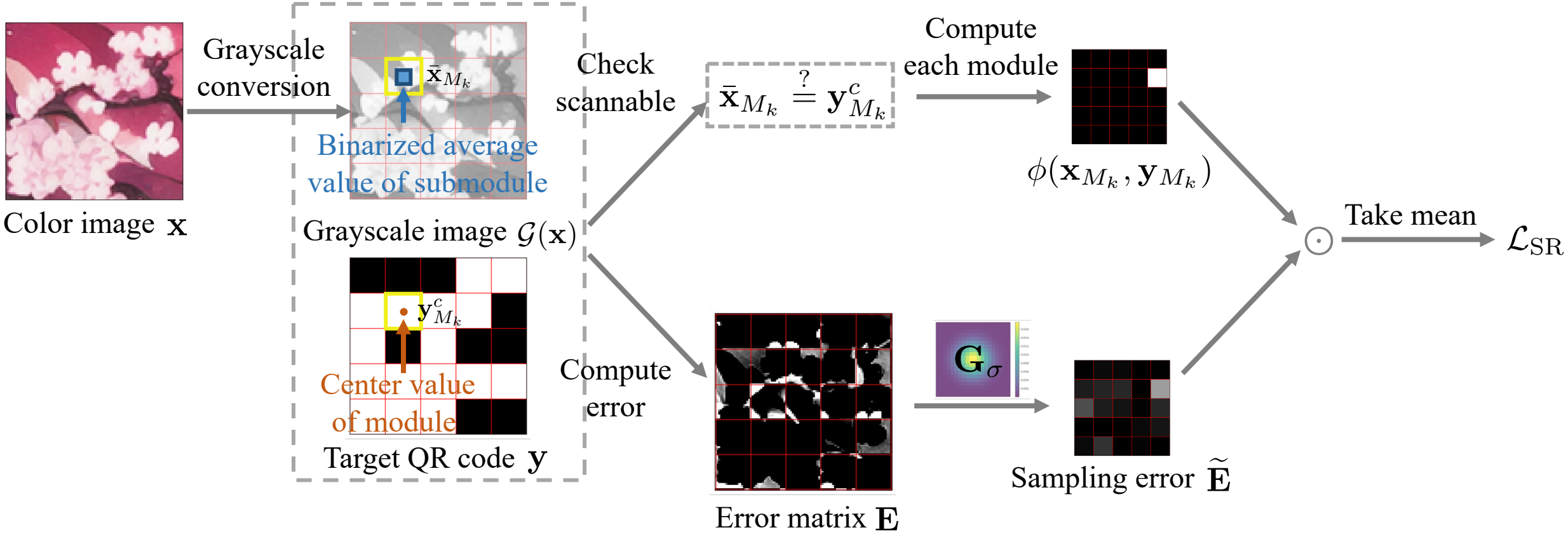
Image

Binarized image



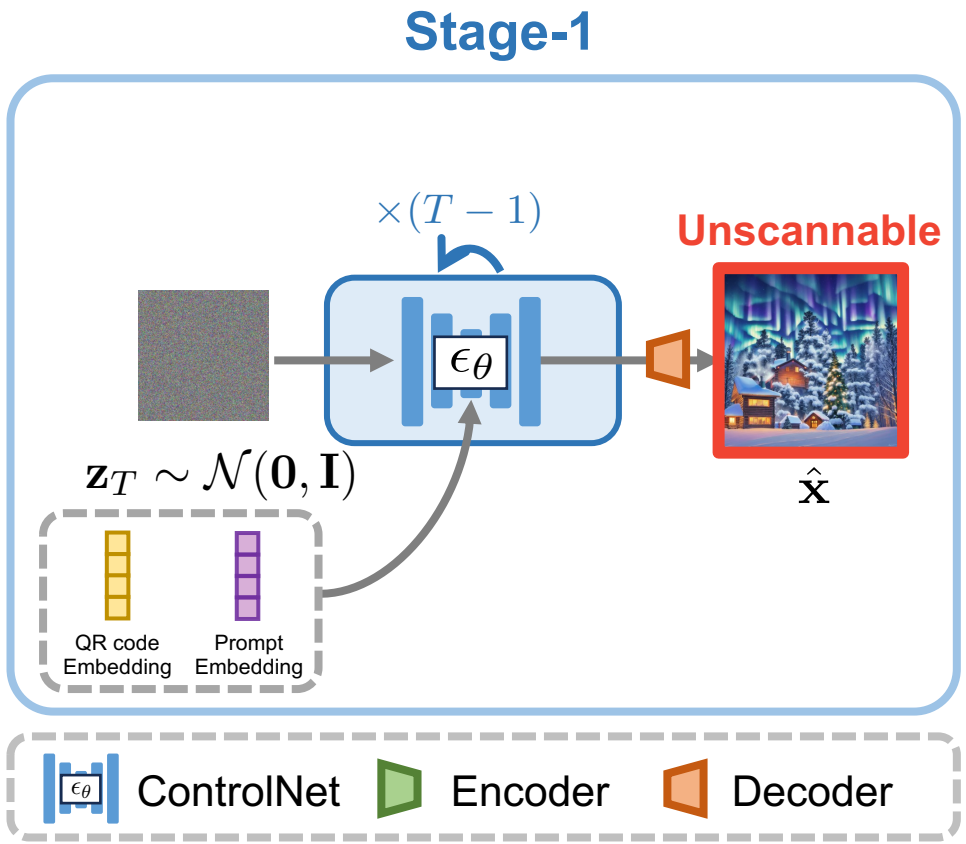
Training by CFG

Scanning Robust Perceptual Guidance (SRPG)

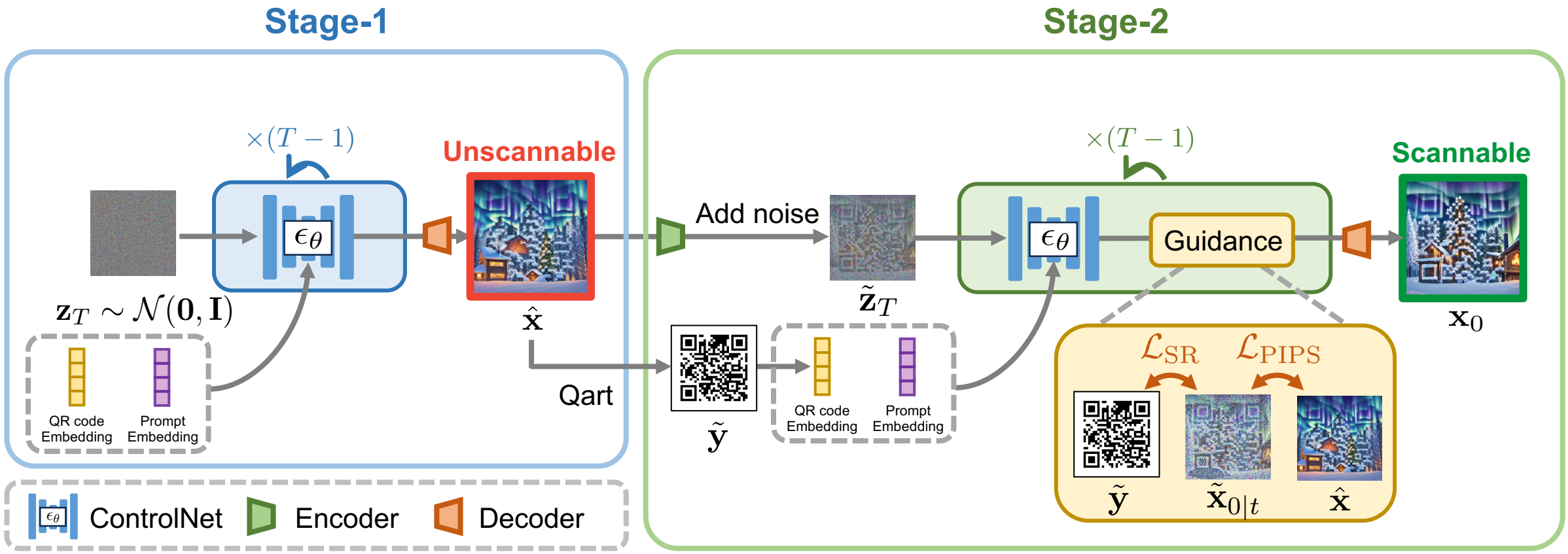


CG: $$\hat{\epsilon}_t = \epsilon_\theta(\tilde{\mathbf{z}}_t, t, \mathbf{e}_p, \mathbf{e}_{\text{code}}) + \sqrt{1-\bar{\alpha}_t}\nabla_{\tilde{\mathbf{z}}_t}[\lambda_1 \mathcal{L}_{\text{SR}}(\tilde{\mathbf{x}}_{0|t}, \tilde{\mathbf{y}}) + \lambda_2 \mathcal{L}_{\text{LPIPS}}(\tilde{\mathbf{x}}_{0|t}, \hat{\mathbf{x}})]$$

Two-stage Iterative Refinement Pipeline



Two-stage Iterative Refinement Pipeline



Qualitative Comparisons

Prompt	QR Code AI Art	QR Diffusion	QRBTF	DiffQRCode (Ours)
Winter wonderland, fresh snowfall, evergreen trees, cozy log cabin, smoke rising from chimney, aurora borealis in night sky.				
(a) Encoded message: Thanks reviews!				
Old European town square, cobblestone streets, café terraces, flowering balconies, gothic cathedral, bustling morning.				
(b) Encoded message: I think, therefore I am!				
Forest clearing at night, fireflies, full moon, ancient oak tree, soft grass, mystical ambiance.				
(c) Encoded message: https://www.google.com.tw/				

Quantitative Results (I)

- **SSR:** Utilize qr-verify to assess the scanning success rate
- **CLIP-aes.:** Utilize CLIP aesthetic predictor to quantify the aesthetic
- **CLIP-score:** Utilize CLIP to quantify the text-image alignment
- **Avg-rank:** Perform user subjective aesthetic preference study

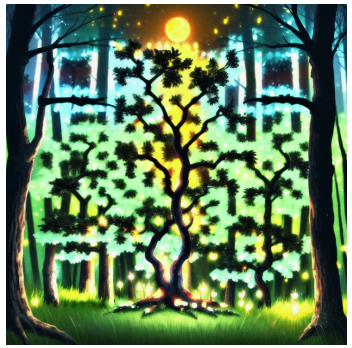
Method	SSR ↑	CLIP-aes. ↑	CLIP-score ↑	Avg-rank ↓
QR Code AI Art [13]	90%	5.7003	0.2341	2.71
QR Diffusion [15]	96%	5.5150	0.2780	3.18
QRBTF [18]	56%	**7.0156**	**0.3033**	**1.86**
DiffQRCoder (Ours)	**99%**	6.8233	0.2992	2.25

Summary

- We can add control to diffusion models via customized deterministic loss function without relying on pre-trained models or adapting additional modules.
- By breaking down the QR code scanning process and underlying mechanisms, we can design a differentiable loss function that serves as a gradient source for diffusion model guidance.
- Furthermore, leveraging VAE for latent optimization ensures improved visual quality while maintaining scannability.

Project Page

Paper



Code

AAAI-25 / IAAI-25 / EAAI-25
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, USA

Pixel Is Not A Barrier: An Effective Evasion Attack for Pixel-Domain Diffusion Models





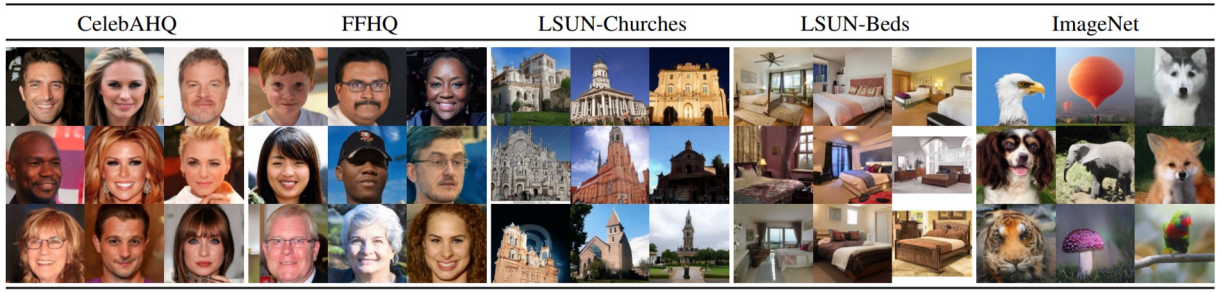

Chun-Yen Shih[1,3,*] Li-Xuan Peng[3,*] **Jia-Wei Liao**[1,3] Ernie Chu[2,3] Cheng-Fu Chou[1] Jun-Cheng Chen[3]

[1] National Taiwan University,
[2] Johns Hopkins University,
[3] Research Center for Information Technology Innovation, Academia Sinica

Background

Diffusion Models allows users to generate photorealistic image with ease.

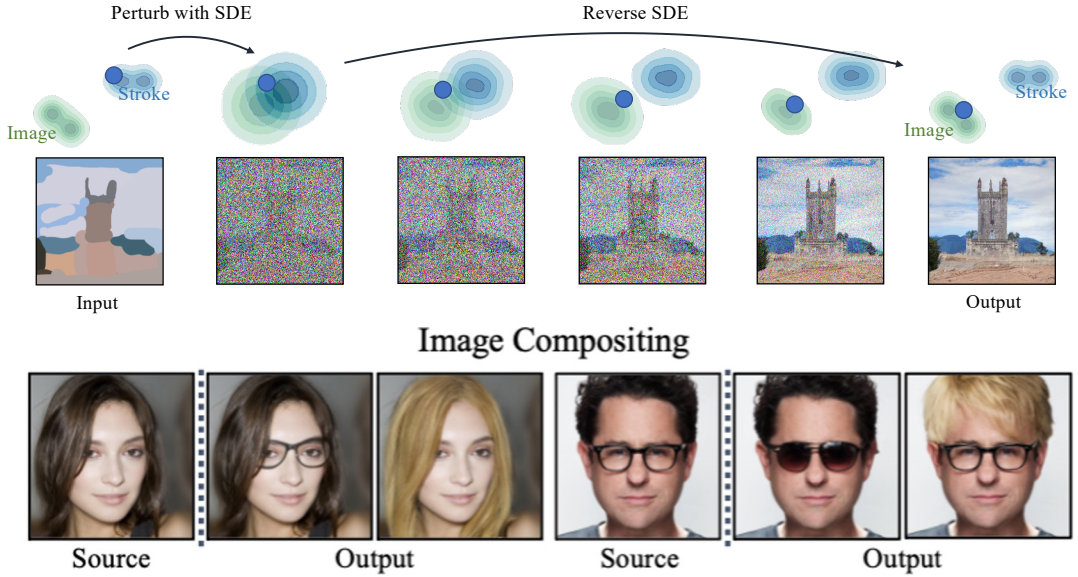
Stable Diffusion



ControlNet

Background

Diffusion Models also allow easily converting image to noisy latent for image translations or editing.



1. Chen-Lin Meng et al. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. ICLR 2022.
2. Gaurav Parmar et al. Zero-shot image-to-image translation. ACM SIGGRAPH 2023.
3. Wen-Liang Zhao et al. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. CVPR 2023

Motivation of Attacking as Protection

How to protect our image against diffusion-based editing?

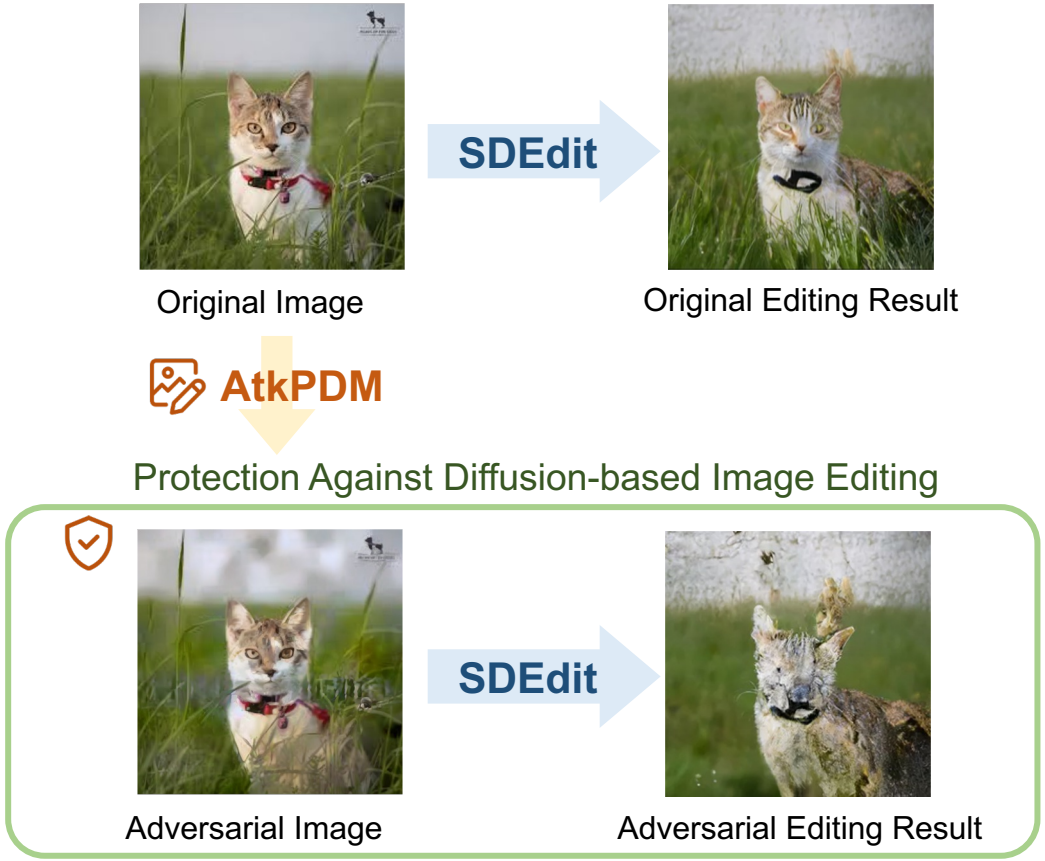
Original Image

SDEdit

Original Editing Result

Motivation of Attacking as Protection

How to protect our image against diffusion-based editing?
We can approach this goal as an adversarial attack to the diffusion models.



Problem Formulation and Methodology

Problem

$$\max_{\mathbf{x}^{\text{adv}} \in \mathcal{M}} d(\text{SDEdit}(\mathbf{x}, t), \text{SDEdit}(\mathbf{x}^{\text{adv}}, t))$$

$$\text{subject to } d'(\mathbf{x}, \mathbf{x}^{\text{adv}}) \leq \delta$$

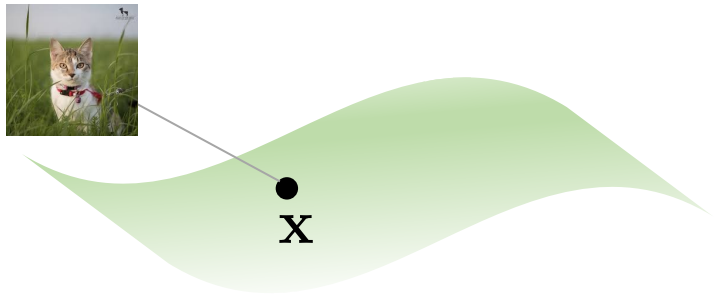
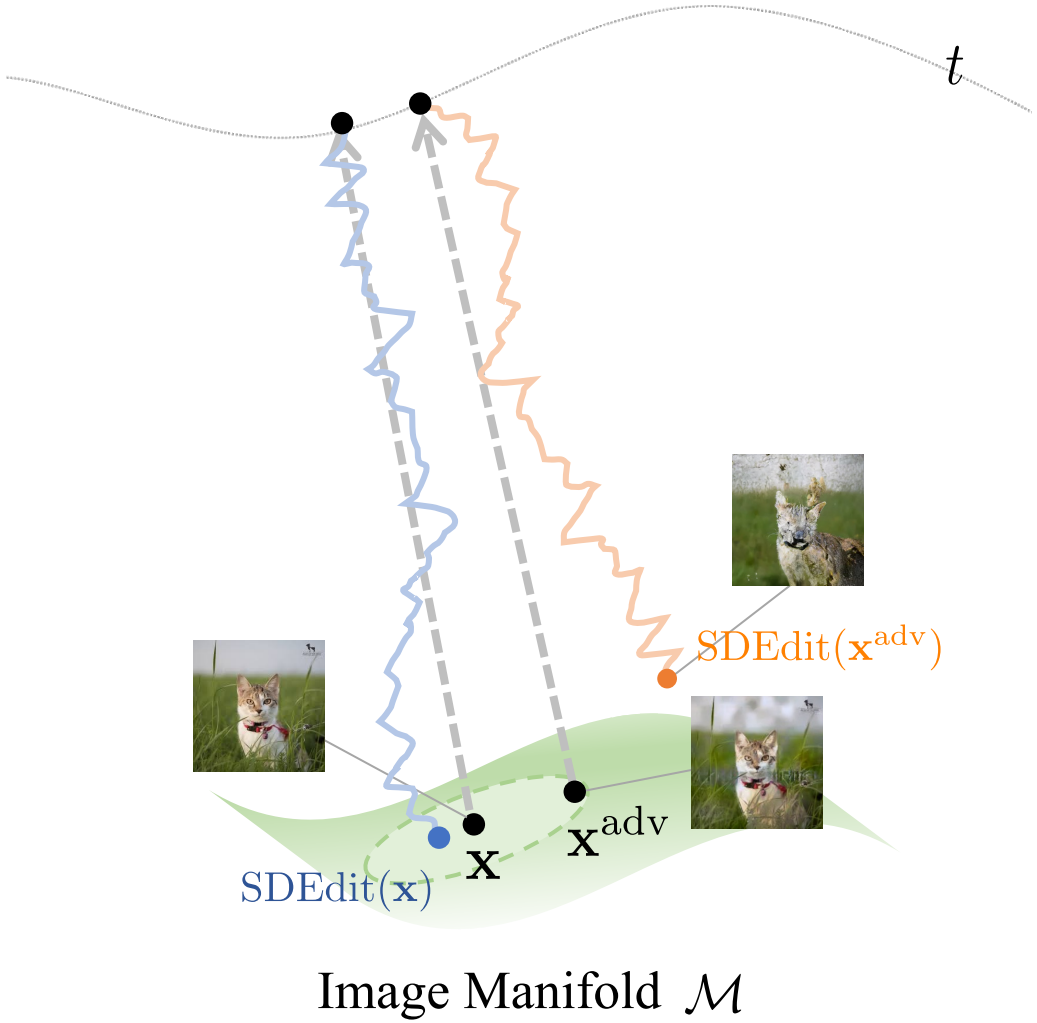


Image Manifold $\mathcal{M}$

Problem Formulation and Methodology

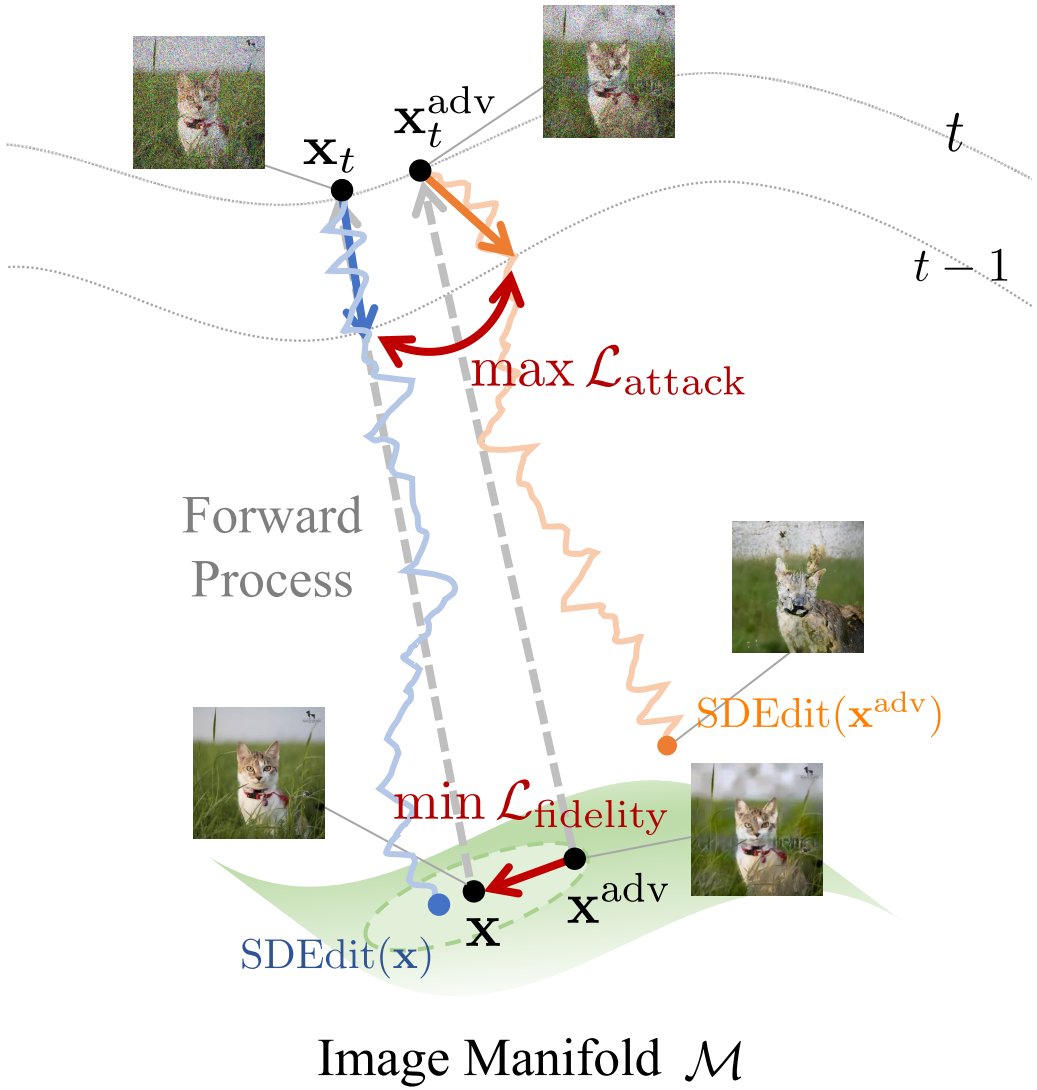


Problem

$$\max_{\mathbf{x}^{\mathrm{adv}} \in \mathcal{M}} d(\mathrm{SDEdit}(\mathbf{x}, t), \mathrm{SDEdit}(\mathbf{x}^{\mathrm{adv}}, t))$$

$$\text{subject to } d'(\mathbf{x}, \mathbf{x}^{\mathrm{adv}}) \leq \delta$$

Problem Formulation and Methodology



Problem

$$\max_{\mathbf{x}^{\text{adv}} \in \mathcal{M}} d(\text{SDEdit}(\mathbf{x}, t), \text{SDEdit}(\mathbf{x}^{\text{adv}}, t))$$

$$\text{subject to } d'(\mathbf{x}, \mathbf{x}^{\text{adv}}) \leq \delta$$

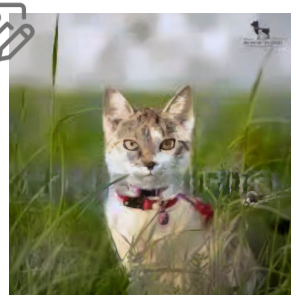
Proposed Losses

$$\max_{\mathbf{x}^{\text{adv}} \in \mathcal{M}} \mathbb{E}_{t, \mathbf{x}_t | \mathbf{x}, \mathbf{x}_t^{\text{adv}} | \mathbf{x}} \mathcal{L}_{\text{attack}}(\mathbf{x}_t, \mathbf{x}_t^{\text{adv}})$$

$$\text{subject to } \mathcal{L}_{\text{fidelity}}(\mathbf{x}, \mathbf{x}^{\text{adv}}) \leq \delta$$

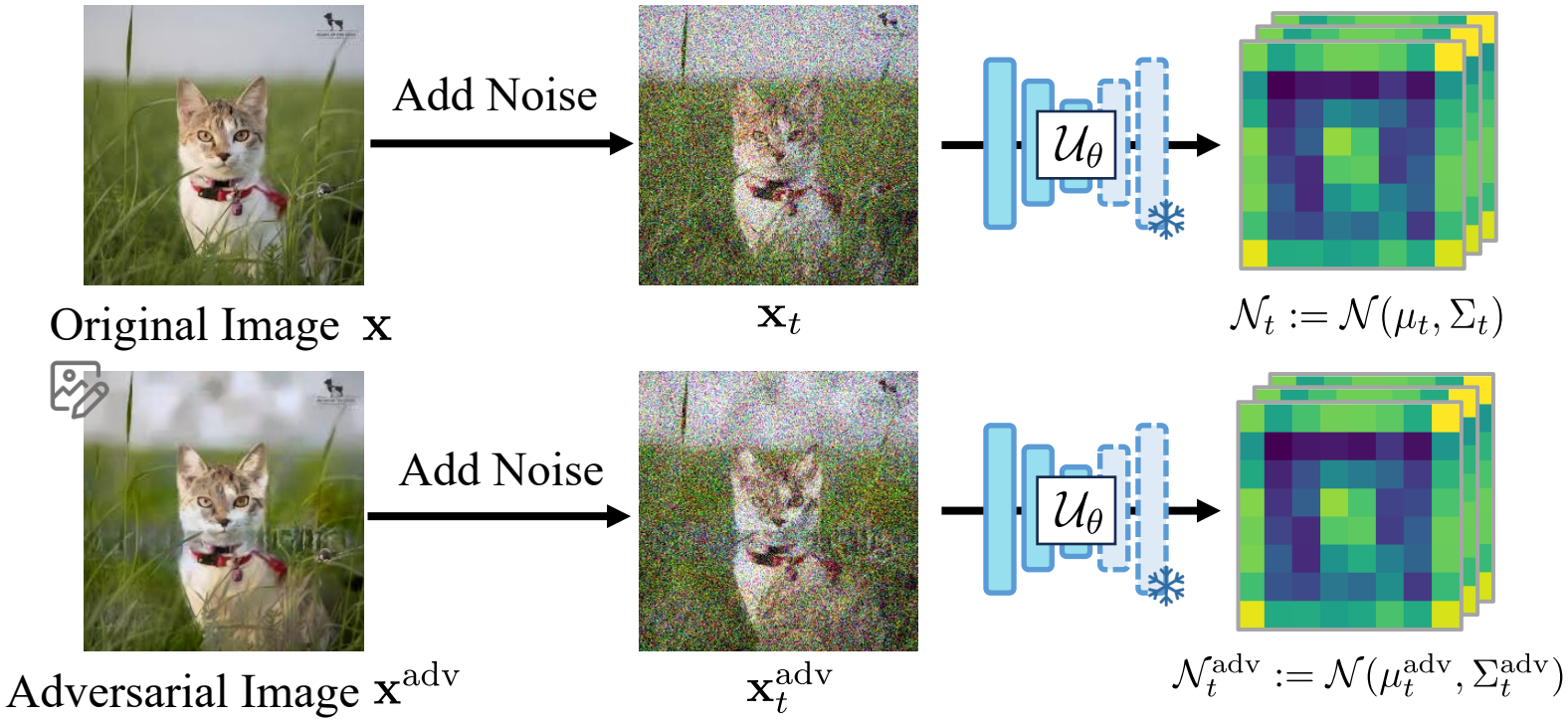
Proposed Method

Original Image $\mathbf{x}$

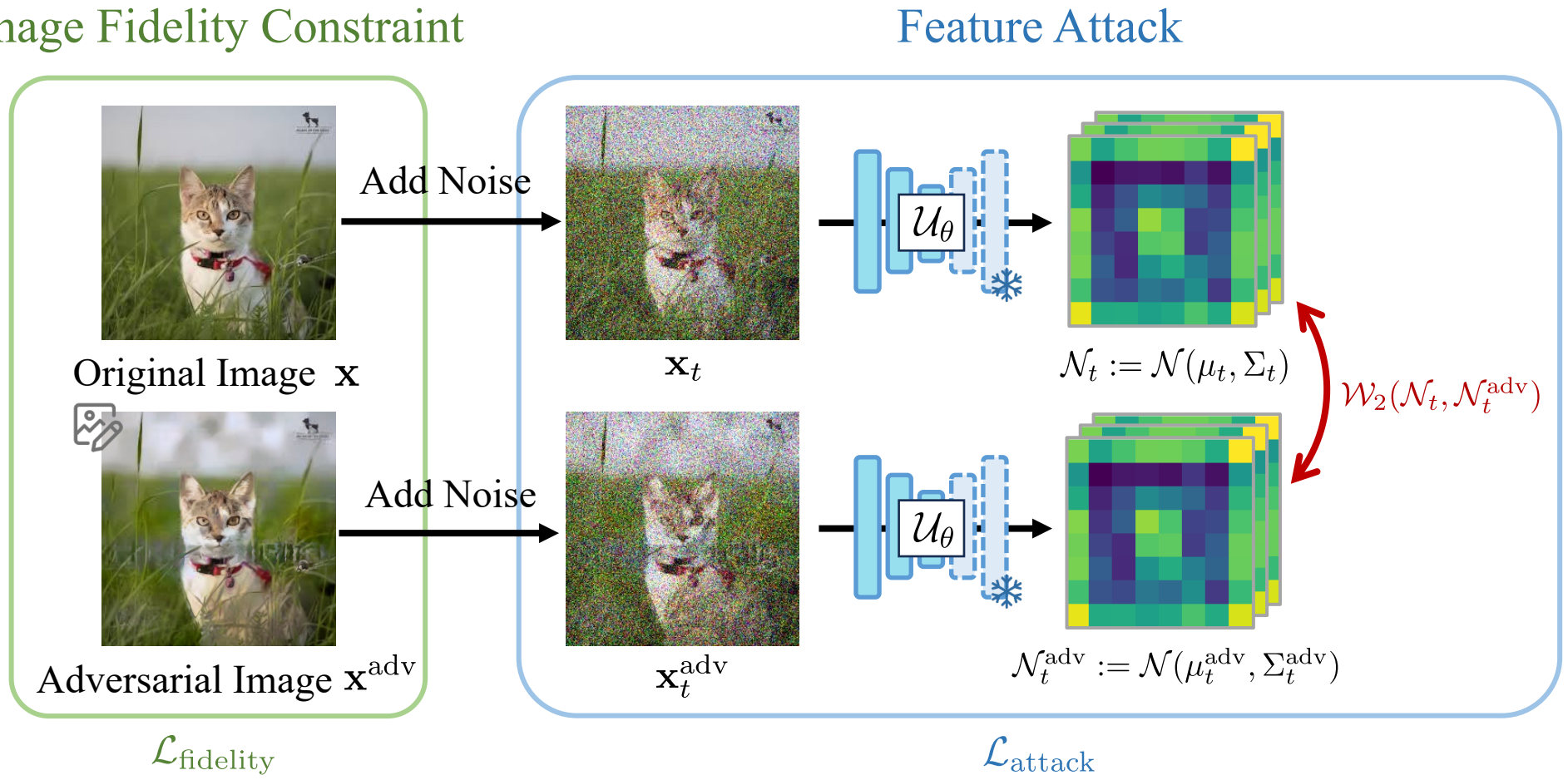


Adversarial Image $\mathbf{x}^{\mathrm{adv}}$

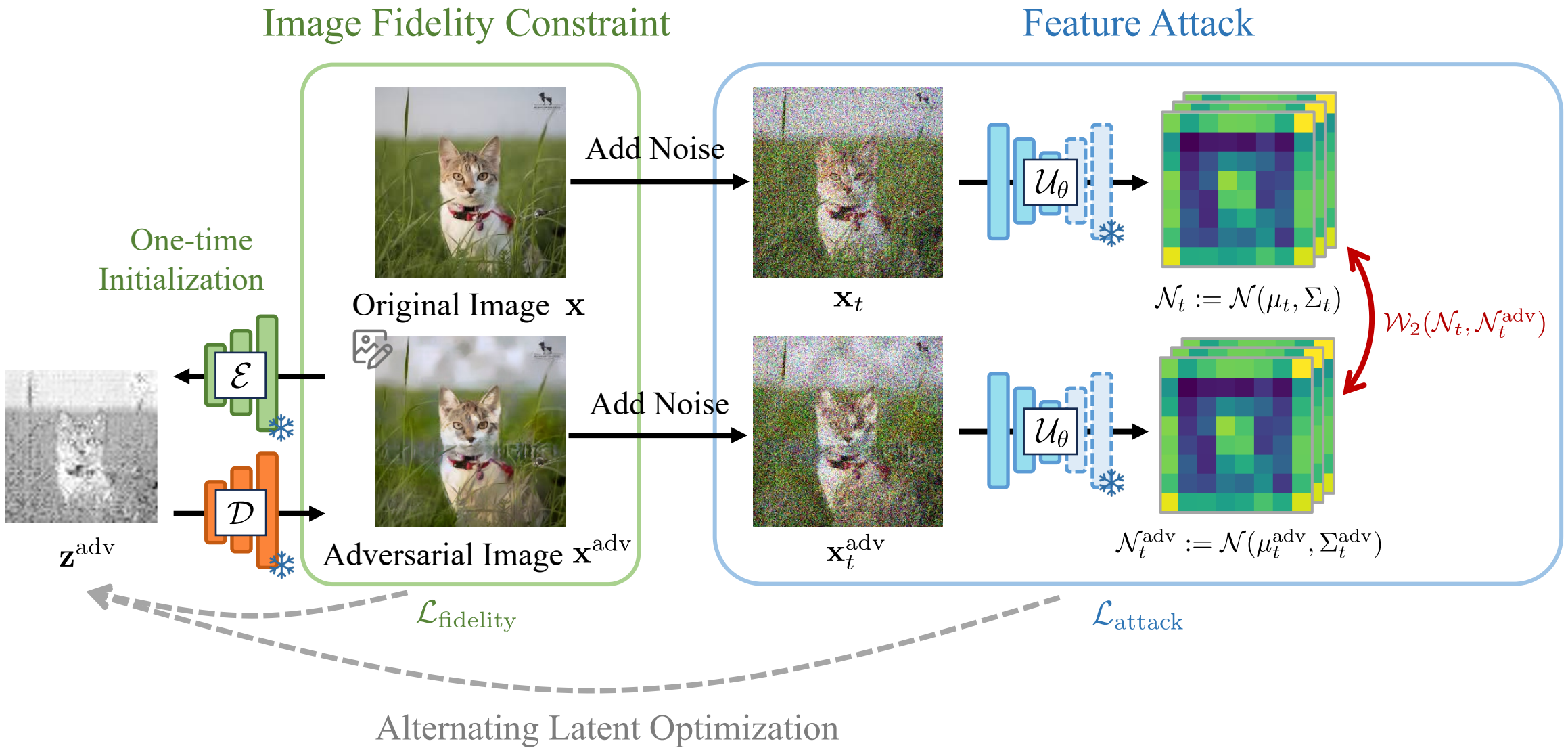
Proposed Method



Proposed Method

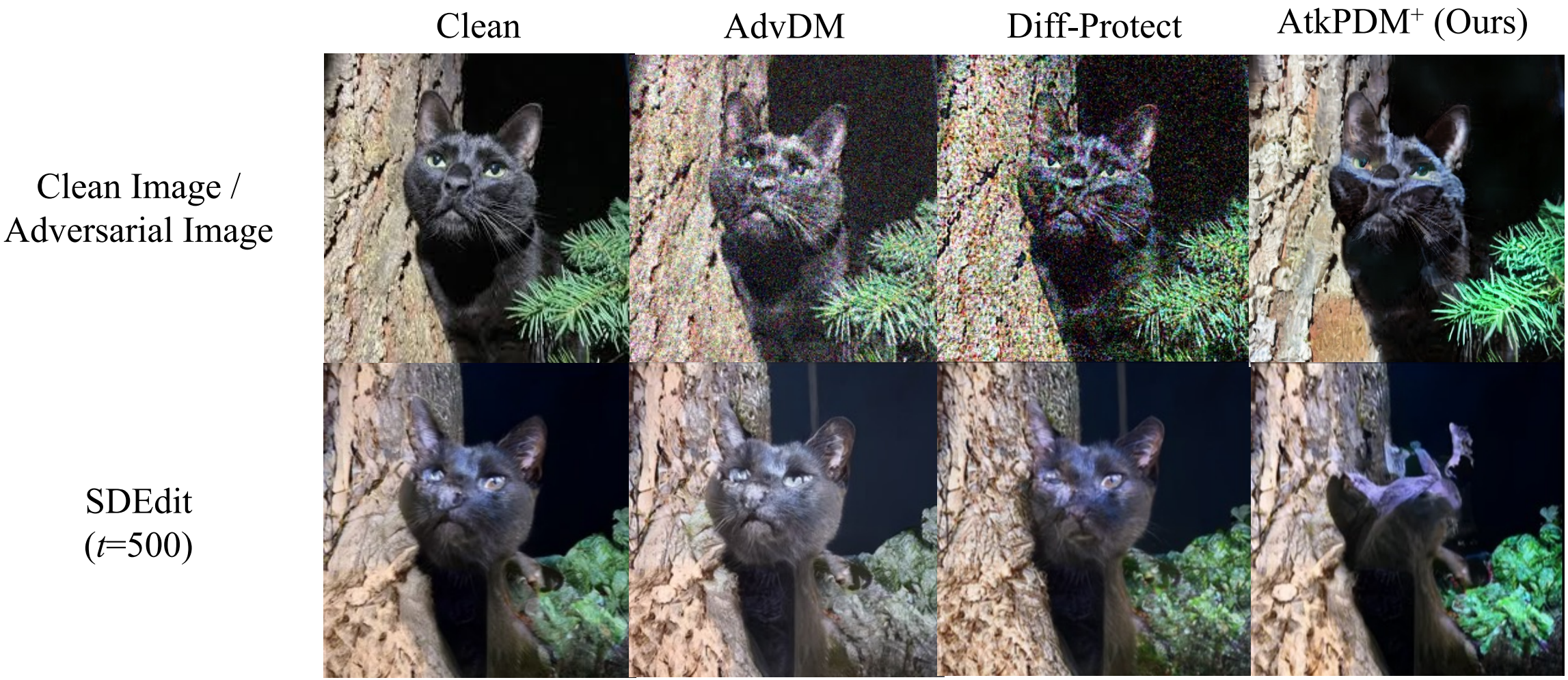


Proposed Method



Qualitative Results (I)

Qualitative Results (II)



Summary

- Although the denoising processes of PDM and LDM seems robust, there still exists vulnerabilities in the feature space inherent in the diffusion models.
- Our study shows the denoising process of the PDMs are robust to pixel-level adversarial perturbation but susceptible to perceptual-level adversarial perturbation.
- We can perform optimization over the latent space of a victim-model-agnostic Variational Autoencoder (VAE) to craft an effective perceptual-level adversarial perturbation against PDM while maintaining the image fidelity.

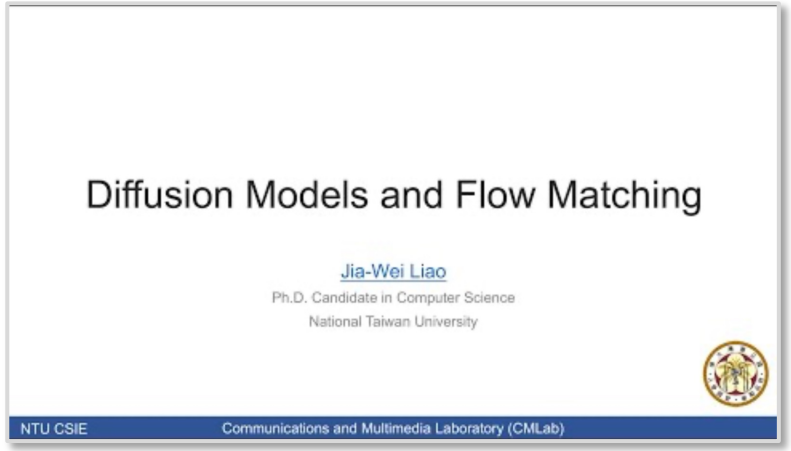
Project Page Paper Code



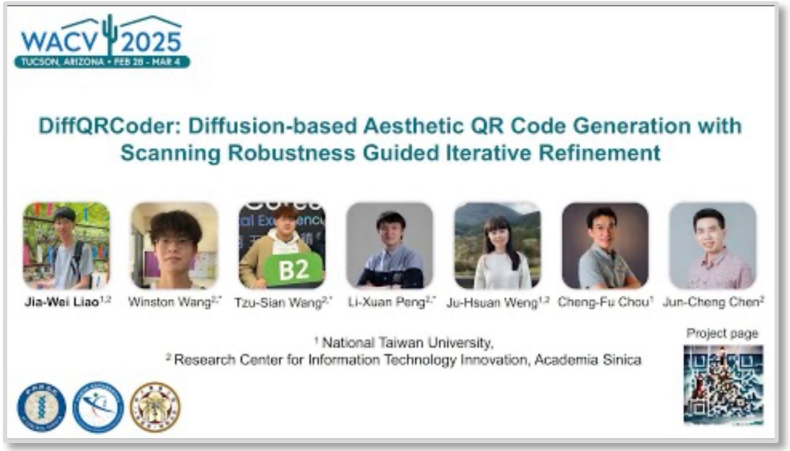
Takeaway

- Diffusion models learn to generate data by reversing a noise-adding process, forming a powerful generative framework.
- Controllability in diffusion models spans training-free (CG), learning-based (CFG), and semantic-level (ControlNet) approaches.
- DiffQRCoder balances aesthetics and scannability by leveraging custom-designed SRPG with CG, generating visually pleasing yet functional QR codes.
- AtkPDM protects images by optimizing feature-space attack loss to break diffusion-based editing.

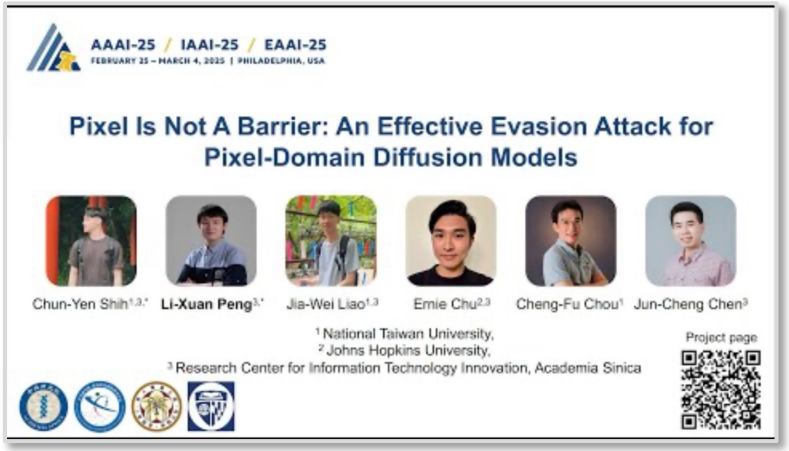
YouTube Channel: JWAI



[Diffusion Models and Flow Matching]



[DiffQRCoder]



[AtkPDM]

Learning Resources for Diffusion Models

- 【漫士科普】人工智慧博士生告訴你 SORA 擴散模型究竟是怎麼產生影片的？
- Hung-Yi Lee YouTube
- Jia-Bin Huang YouTube
- Diffusion and Score-Based Generative Models (Yang Song)
- Evolution of Diffusion Models: From Birth to Enhanced Efficiency and Controllability (Jesse)
- Lil'Log What are Diffusion Models?
- 生成擴散模型漫談 (蘇劍林)

Feedback Form

Please take a moment to fill out the feedback form. Your input helps us improve future sessions.

https://forms.gle/SH5pG8uE7KbzAfHNA

Thank you