भारतीय प्रौद्योगिकी संस्थान दिल्ली
Indian Institute of Technology Delhi

COV877
Special Module on Visual Computing

Generative AI for Visual Content Creation: Image, Video, and 3D

Image Generation and Editing

Instructor:

Dr. Lokender Tiwari

Research Scientist

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

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

GLIDE

- Diffusion models can generate good quality images
- How to generate **something specific?**
- **Answer: Guided Diffusion Model**

Class Conditional
Alaskan Malamute

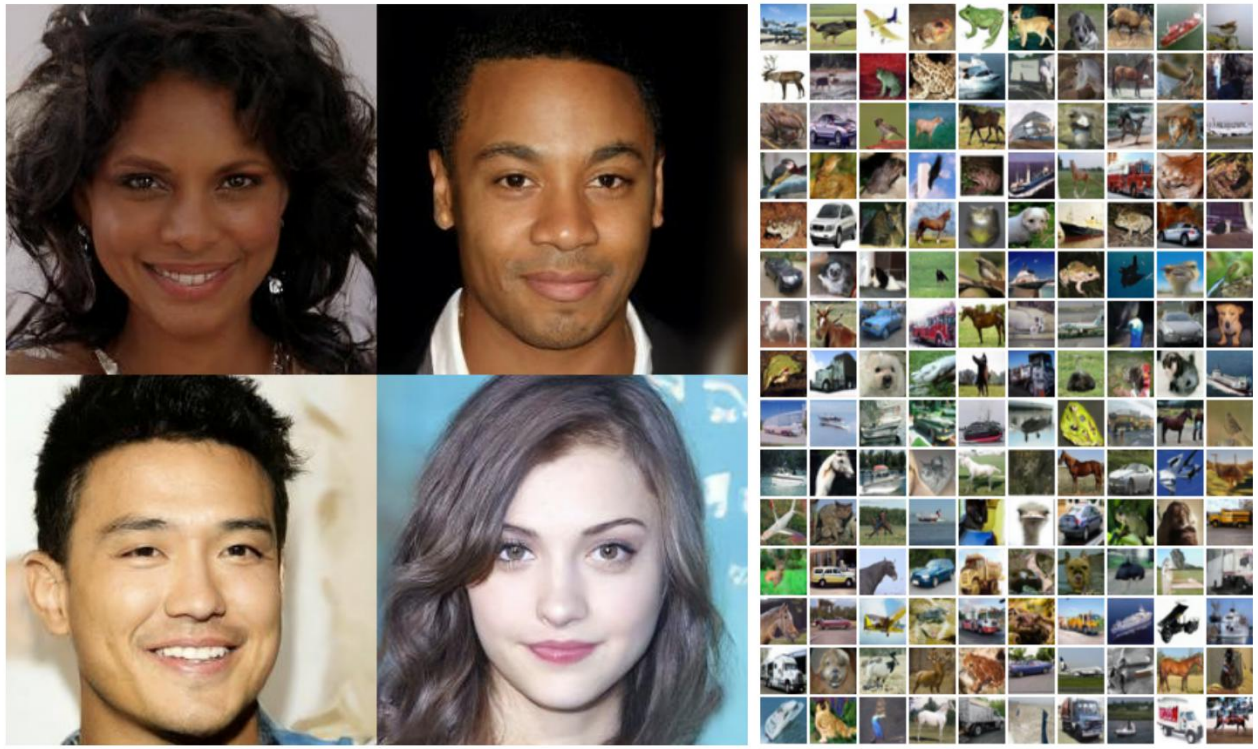


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

How about generate an image of **Alaskan Malamute doing something**
How to generate such image ?

Answer : Text-Conditional Diffusion Model

GLIDE

- What are contributions of GLIDE?
  - A Text-conditional image synthesis
  - Compared two different guidance strategies: *CLIP guidance* and *classifier-free guidance*

  - Shown CFG based generated images are preferred by human evaluators for both photorealism and caption similarity
  - GLIDE can be fine-tuned to perform *image inpainting*, enabling powerful *text-driven image editing*

Ways to condition diffusion models

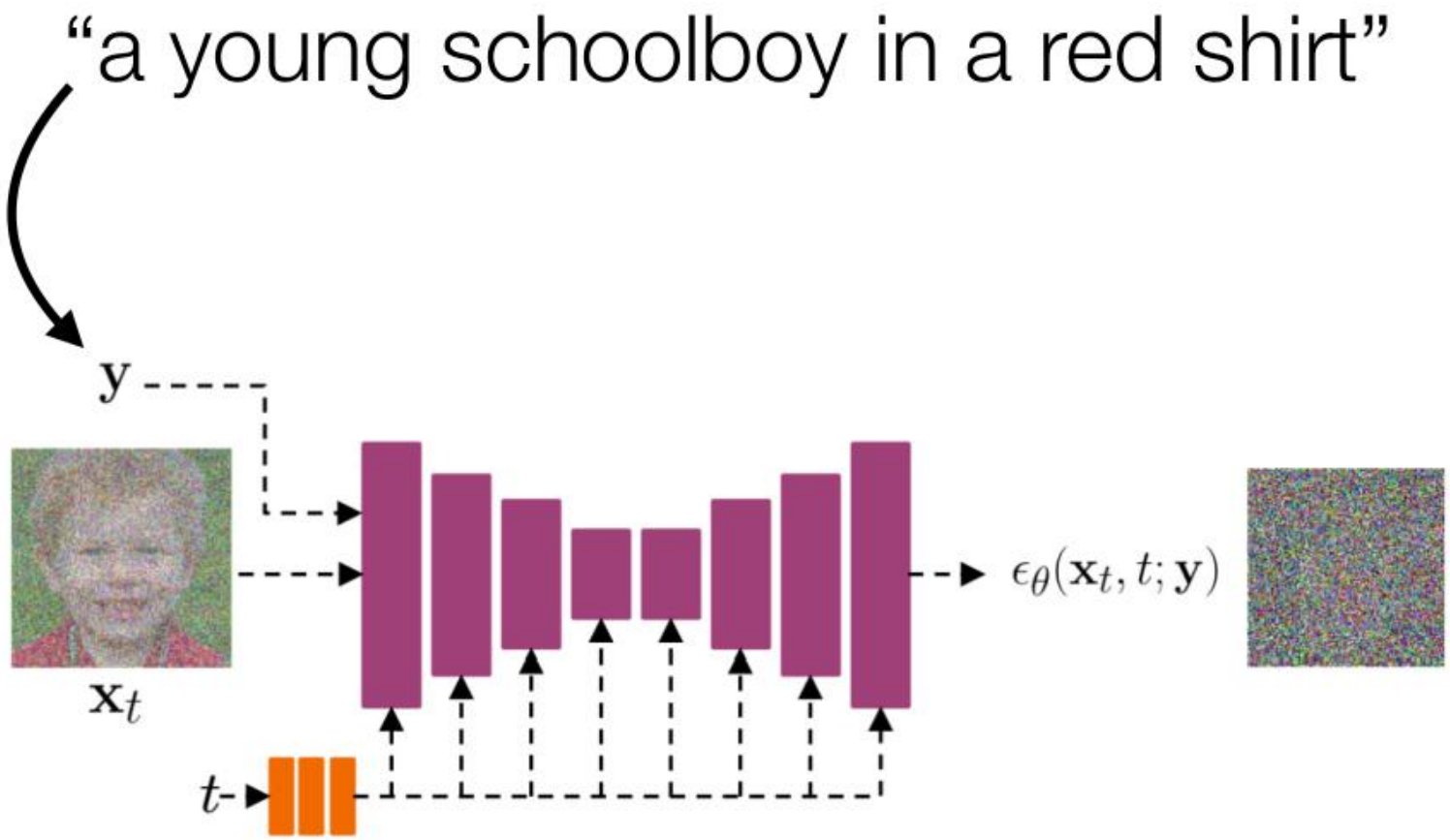
- Explicit
- Classifier Guidance
- Classifier-Free Guidance

Conditioning Diffusion Models - Explicit

“a young schoolboy in a red shirt”

[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Explicit

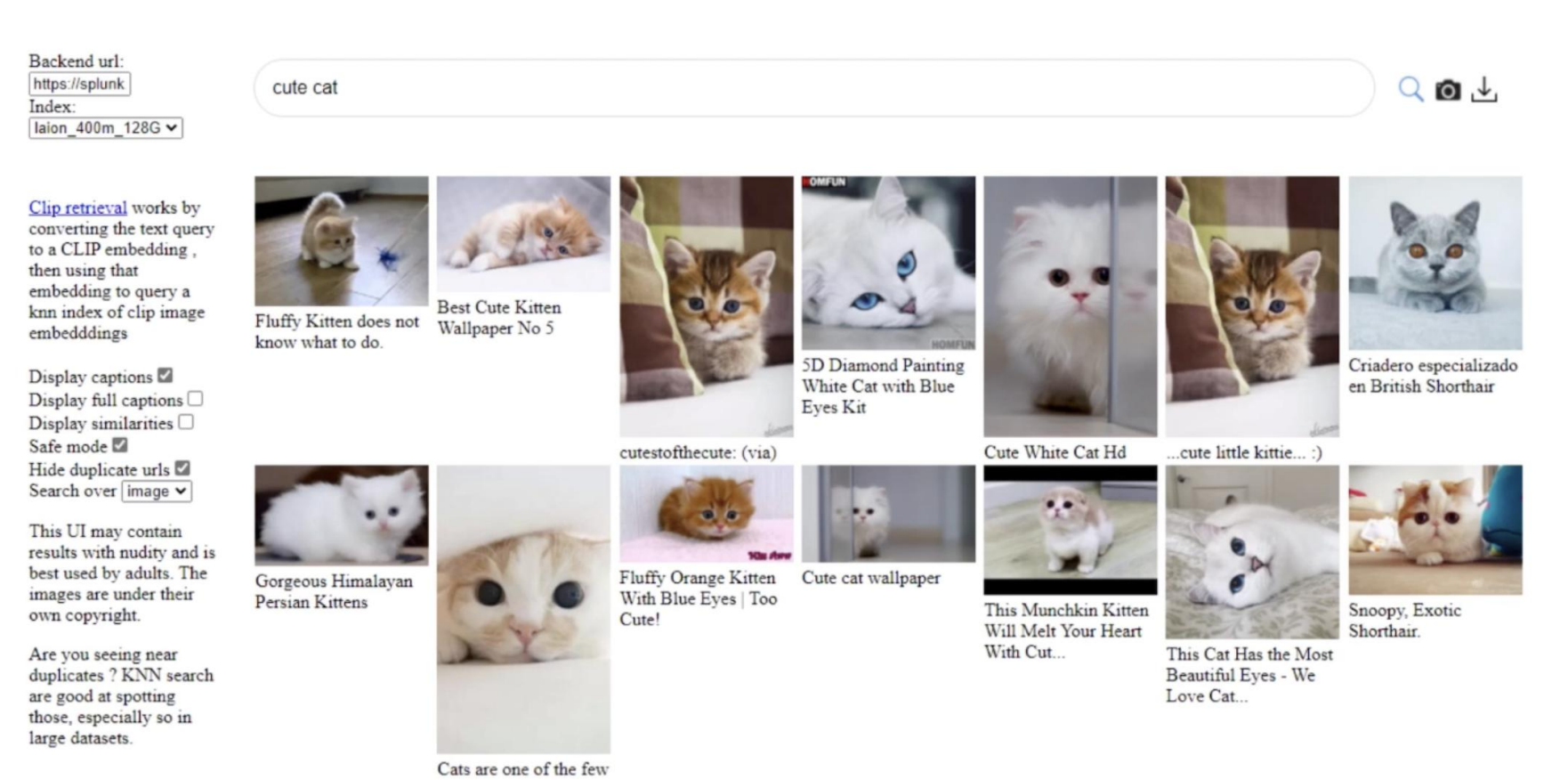


Question : How to train this ?

[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Explicit

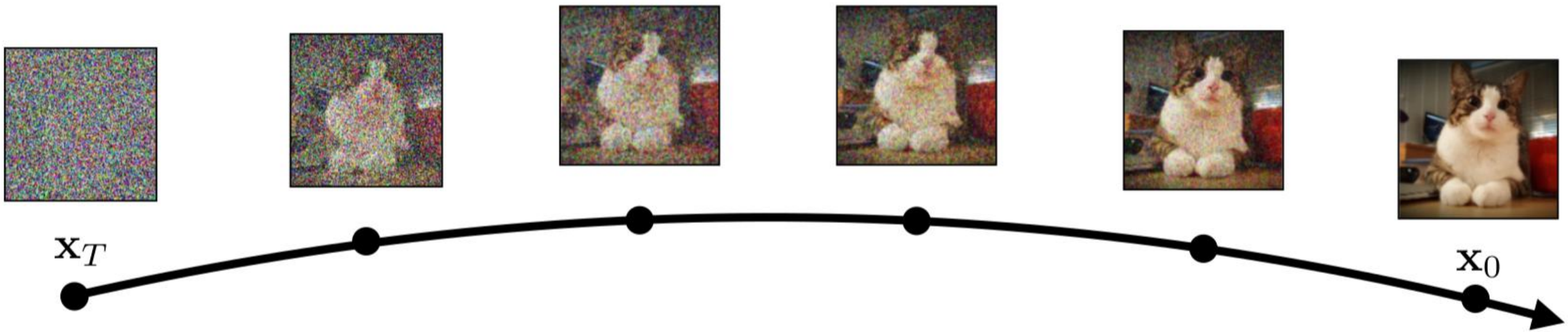
Train it using a large Image-Text dataset e.g., LAION 5B



[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Classifier Guidance

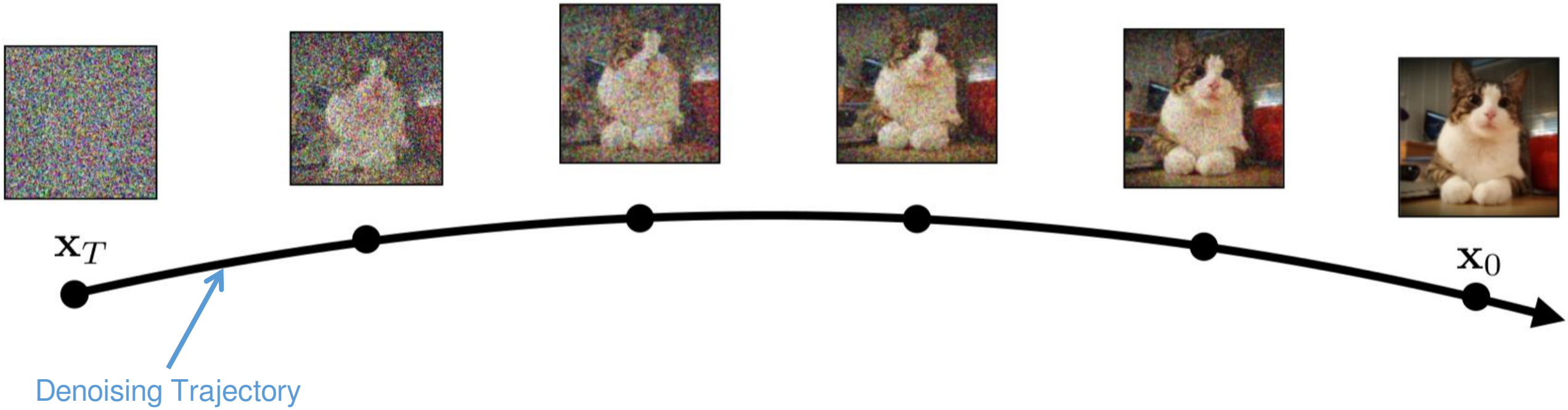
Diffusion is an iterative denoising process that goes from noise to real image in a step by step manner



[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Classifier Guidance

Diffusion is an iterative denoising process that goes from noise to real image in a step by step manner



Main Idea : Perturb the denoising trajectory such that it end up generating the desired image

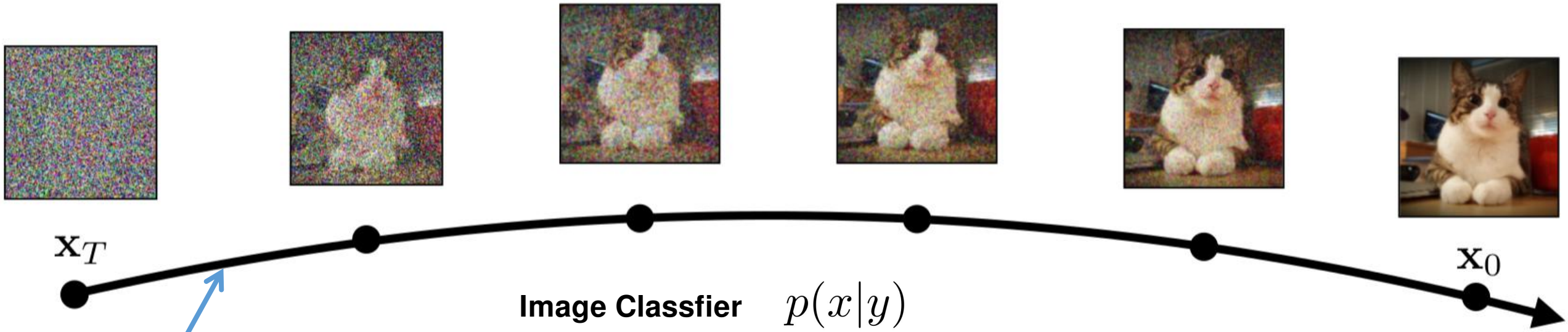
Question : How to perturb the trajectory?

Classifier Guidance is one approach

[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Classifier Guidance

How to get the perturbation?



Denoising Trajectory

Image Classfier $p(x|y)$

Gradient of the classifier w.r.t image $\nabla_x \log p(x|y)$

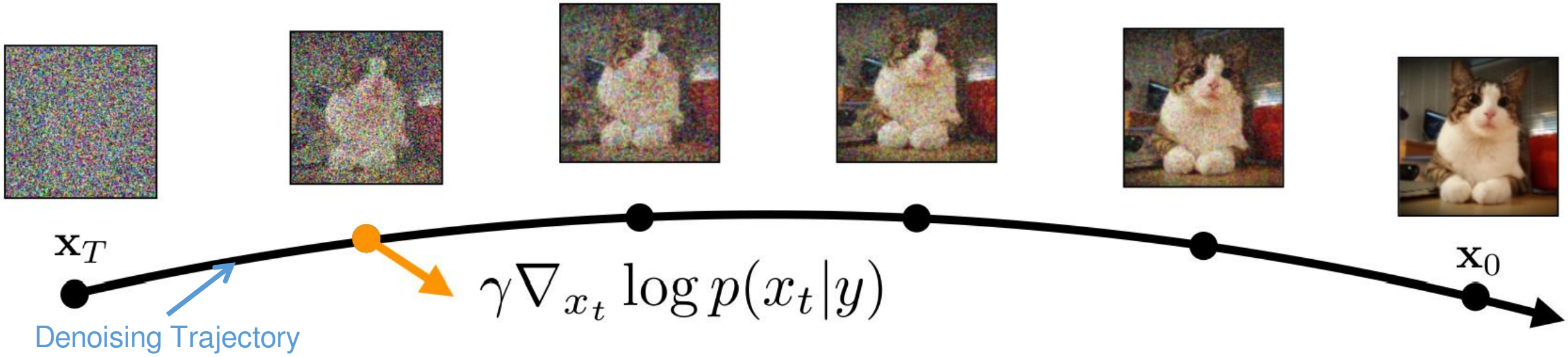
use gradient of the classifier

Objective of the classifier guidance : Perturb x such that it apperance matches to y

[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Classifier Guidance

How to get the perturbation?



$$\tilde{\epsilon}_t(x_t, t, y) = \epsilon_\theta(x_t, t) - \gamma \nabla_{x_t} \log p(x_t|y)$$

Objective of the classifier guidance : Perturb x such that it apperance matches to y

Issue : Classifier needs to be trained on noisy images

use gradient of the classifier

[1] CVPR 2023 Diffusion Models Tutorial

Conditioning Diffusion Models - Classifier-Free Guidance (CFG)

Problems with Classifier Guidance

- Fine-tune or retrain classifier on the noisy data samples

Solution (CFG)

- Let diffusion model itself provide guidance for perturbation

What does this mean ?

$$\epsilon_\theta(x_t, t, \varnothing)$$

Train diffusion model jointly **with** and **without** explicit conditioning

$$\epsilon_\theta(x_t, t, y)$$

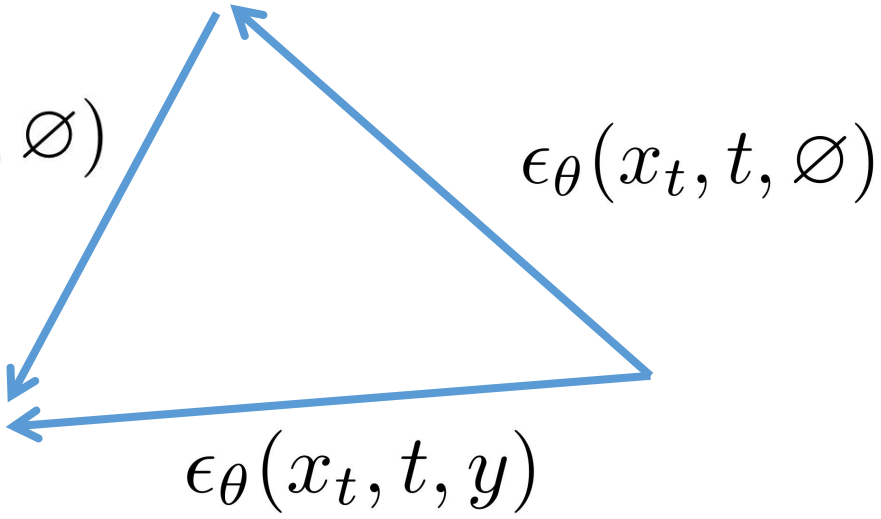
Conditioning Diffusion Models - Classifier-Free Guidance (CFG)

Direction vector from unconditional model to conditional model

$$\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing)$$

What does this mean ?

Use this for perturbation guidance



$$\tilde{\epsilon}(x_t, t, y) = \epsilon_\theta(x_t, t, \varnothing) + \gamma(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing))$$

Guidance Scale

Conditioning Diffusion Models - GLIDE Results

"A stained glass window of a panda eating bamboo"

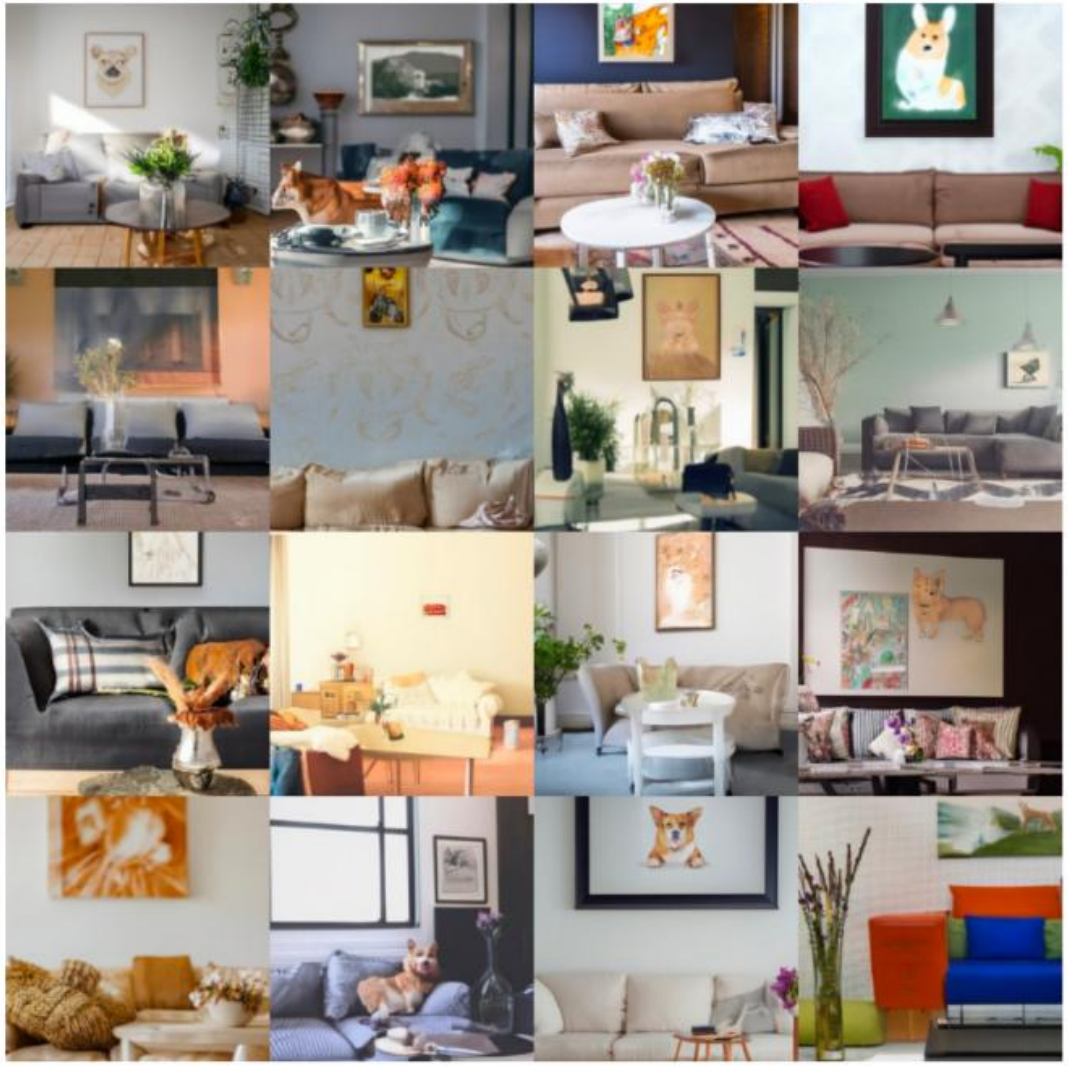
Same as explicit conditioning

$\gamma = 1$

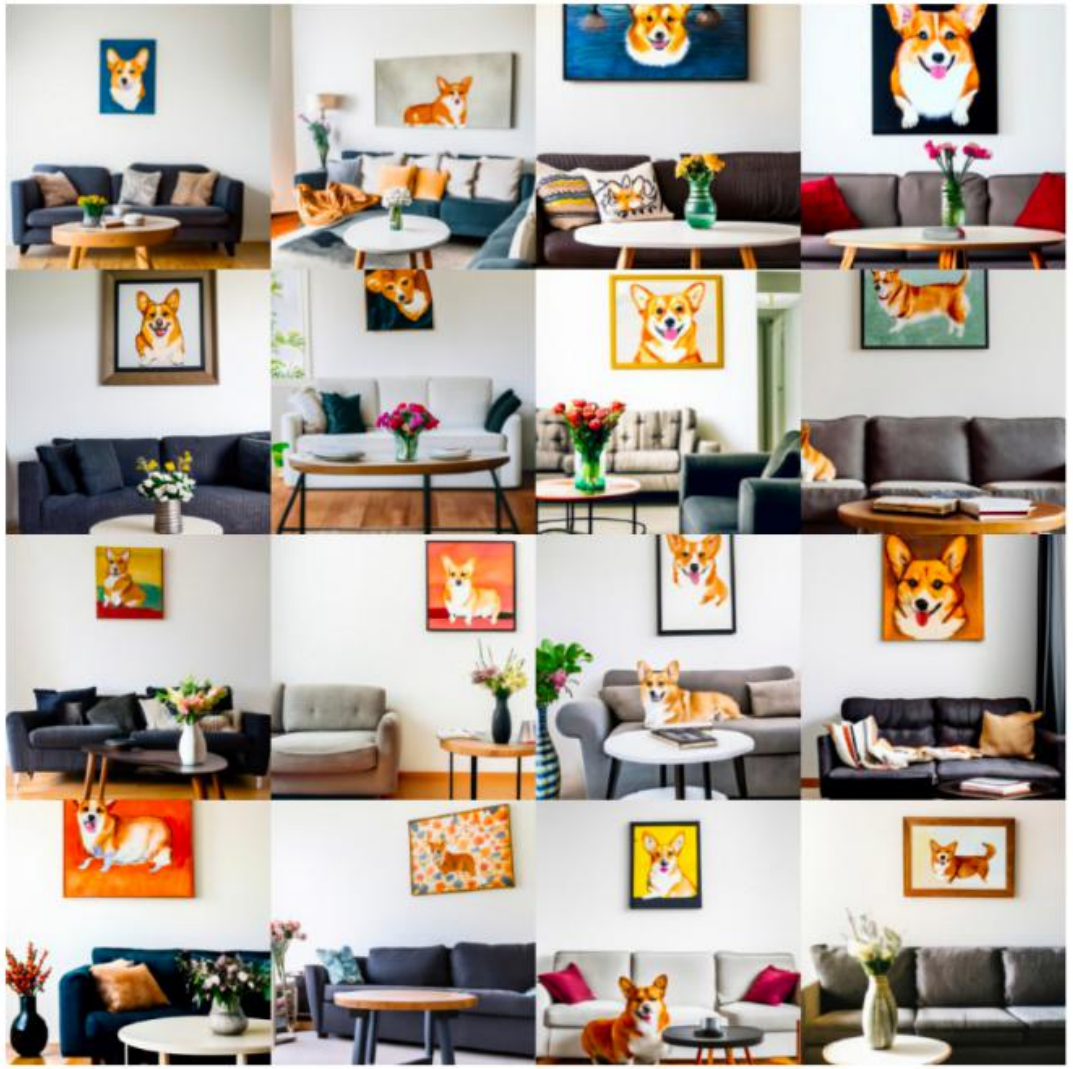
$\gamma = 3$

Conditioning Diffusion Models - GLIDE Results

"A cozy living room with a painting of a corgi on the wall above a couch and a round coffee table in front of a couch and a vase of fowers on a coffee table."



$\gamma = 1$

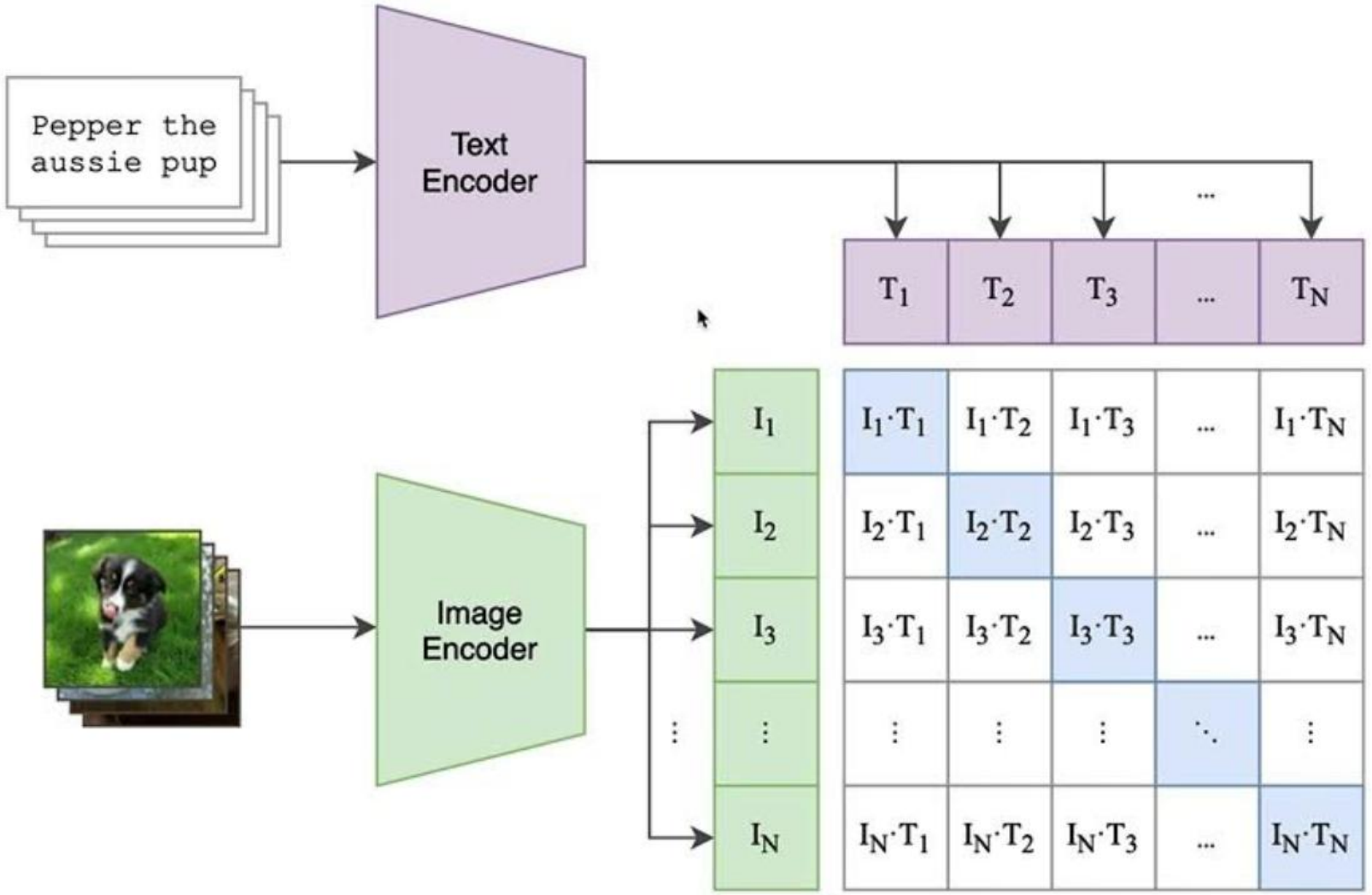


$\gamma = 3$

Conditioning Diffusion Models - CLIP Based Guidance

$$\nabla_{x_t}(f(x_t).g(y))$$

Image Encoder Text Encoder



Use this gradient to guide the diffusion process

High-Resolution Image Synthesis with Latent Diffusion Models

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

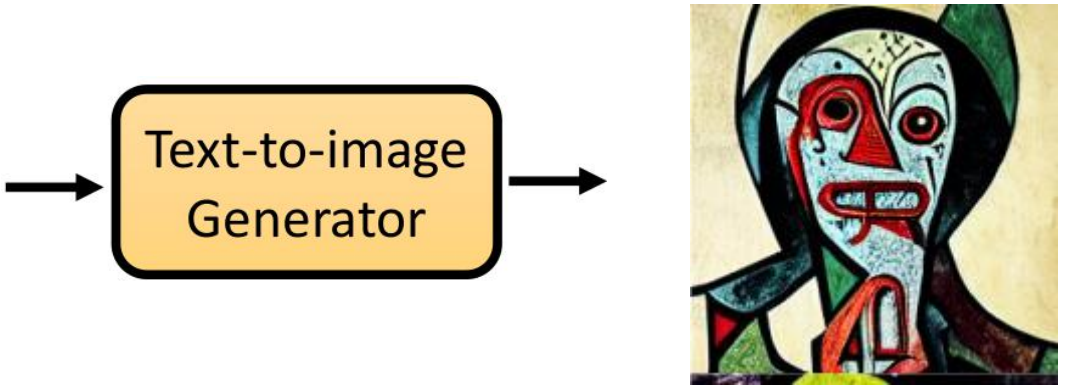
[1]Ludwig Maximilian University of Munich & IWR, Heidelberg University, Germany [R]Runway ML

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

a.k.a Stable Diffusion

Stable Diffusion

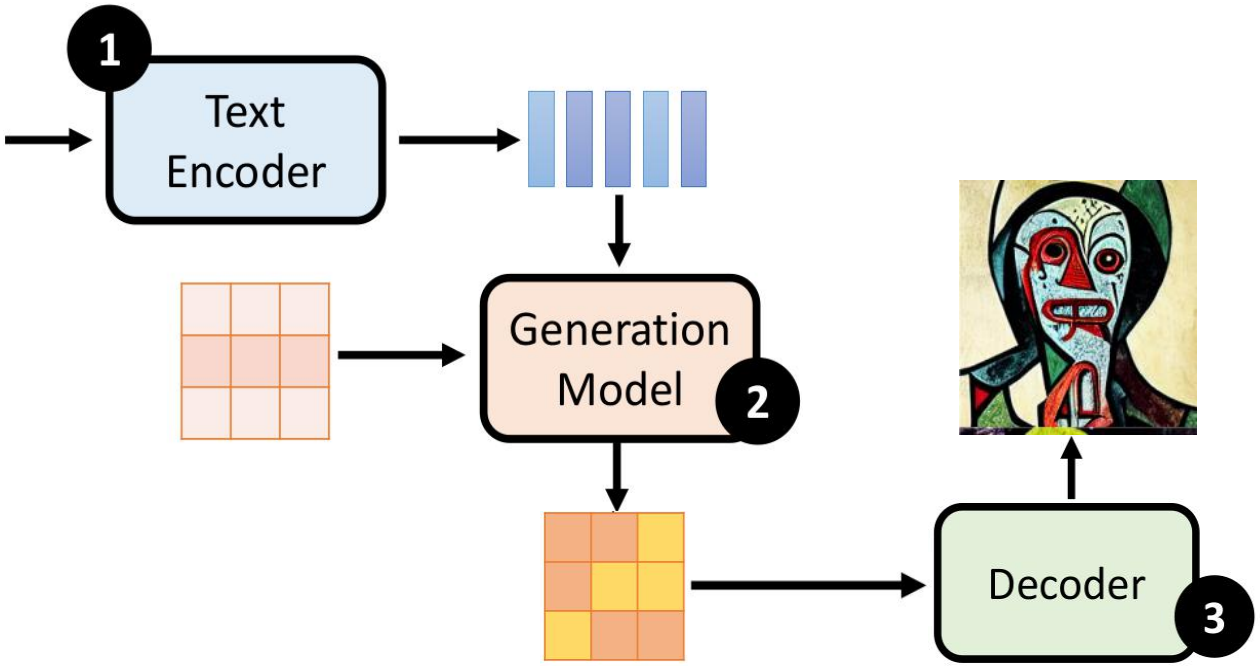
Zombie in Picasso style



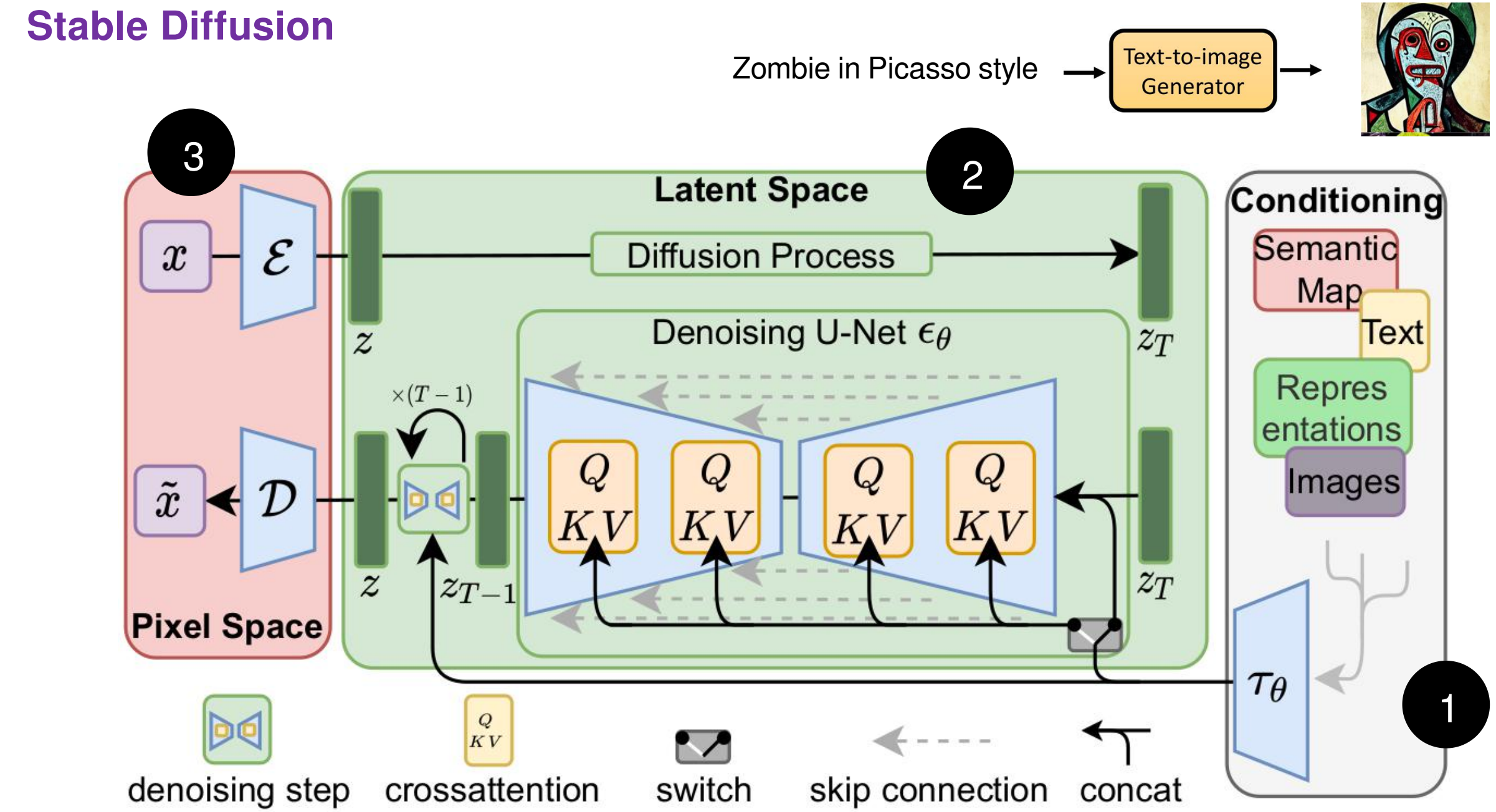
Stable Diffusion

Zombie in Picasso style → Text-to-image Generator →

Zombie in Picasso style



Stable Diffusion

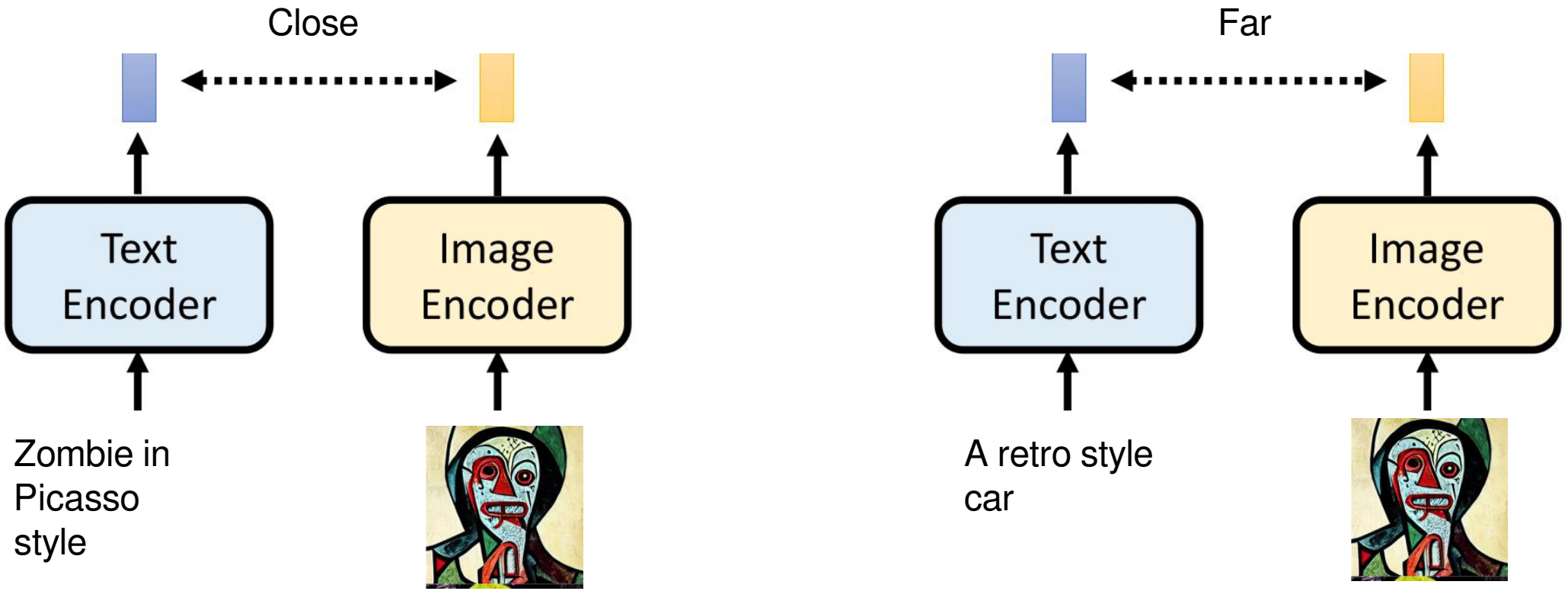


Stable Diffusion

1

Zombie in Picasso style → Text-to-image Generator →

CLIP Objective

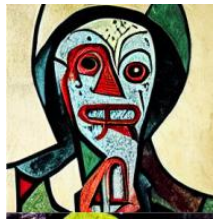


Stable Diffusion

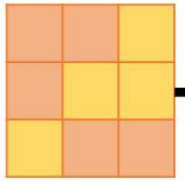
3

Auto-Encoder e.g., VQ-VAE

Zombie in Picasso style → Text-to-image Generator →



Latent Representation

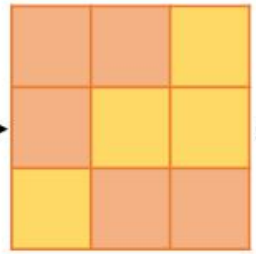


Decoder →

$H \times W \times 3$

Encoder →

$h \times w \times c$



Decoder →

Stable Diffusion

2

Forward Process

$H \times W \times 3$

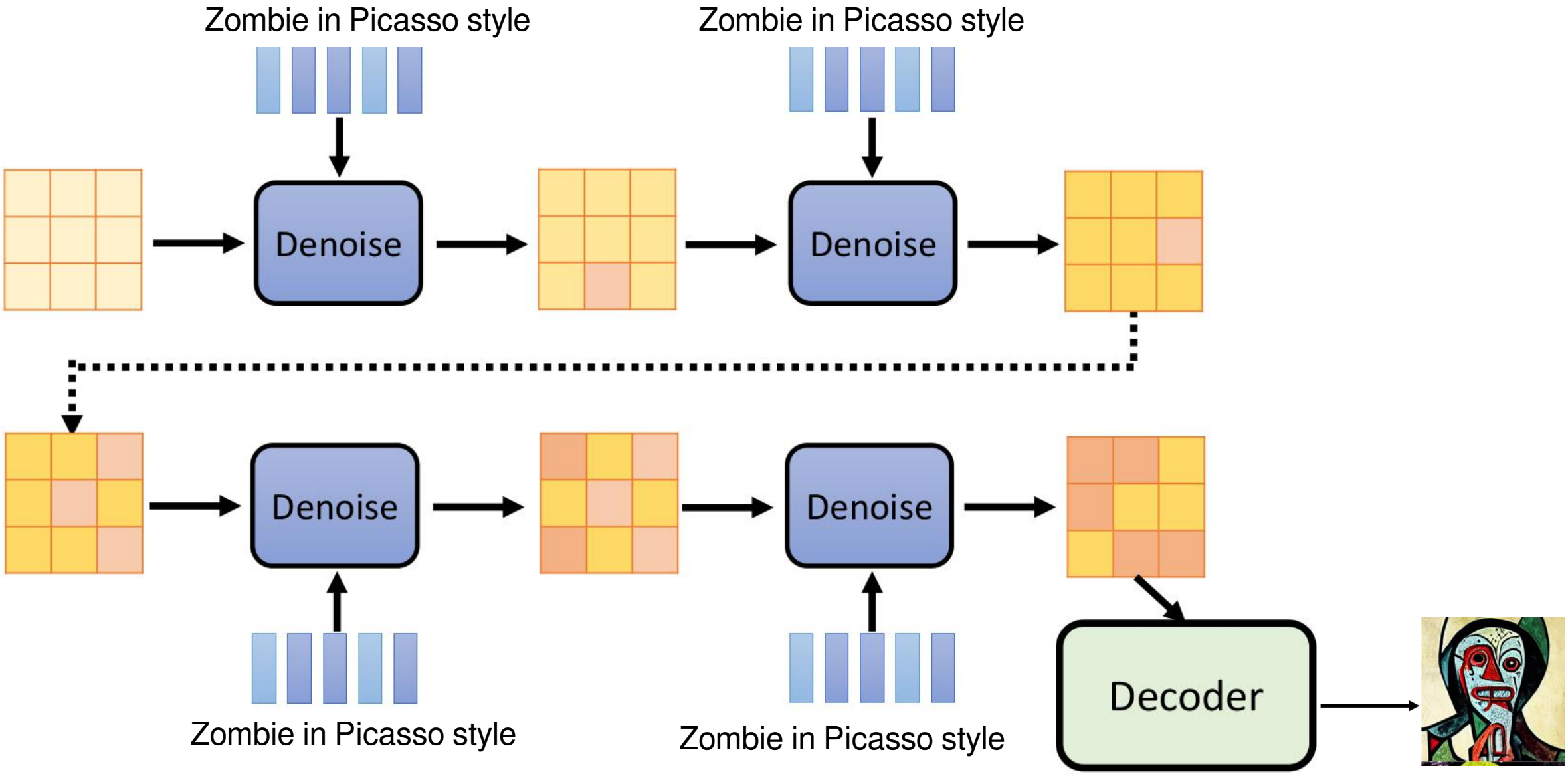
Encoder

noise

Stable Diffusion

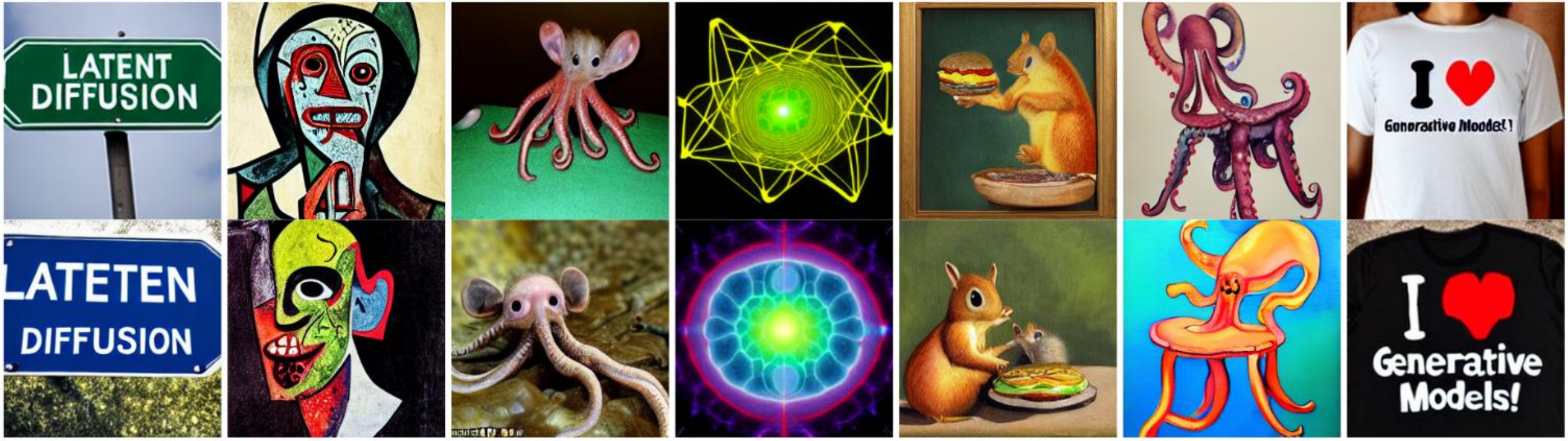
2

Reverse Process

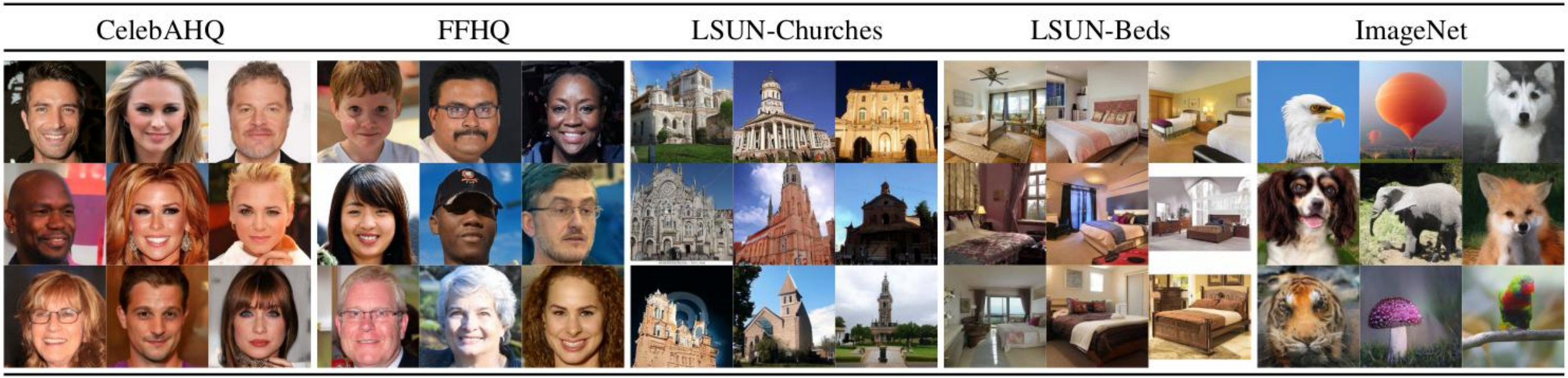


Stable Diffusion - Results

Text-to-Image Synthesis on LAION. 1.45B Model.



Stable Diffusion - Results



Hierarchical Text-Conditional Image Generation with CLIP Latents

Aditya Ramesh[*]
OpenAI
aramesh@openai.com

Prafulla Dhariwal[*]
OpenAI
prafulla@openai.com

Alex Nichol[*]
OpenAI
alex@openai.com

Casey Chu[*]
OpenAI
casey@openai.com

Mark Chen
OpenAI
mark@openai.com

a.k.a DALL-E 2

DALL-E 2

Generate Images From Captions

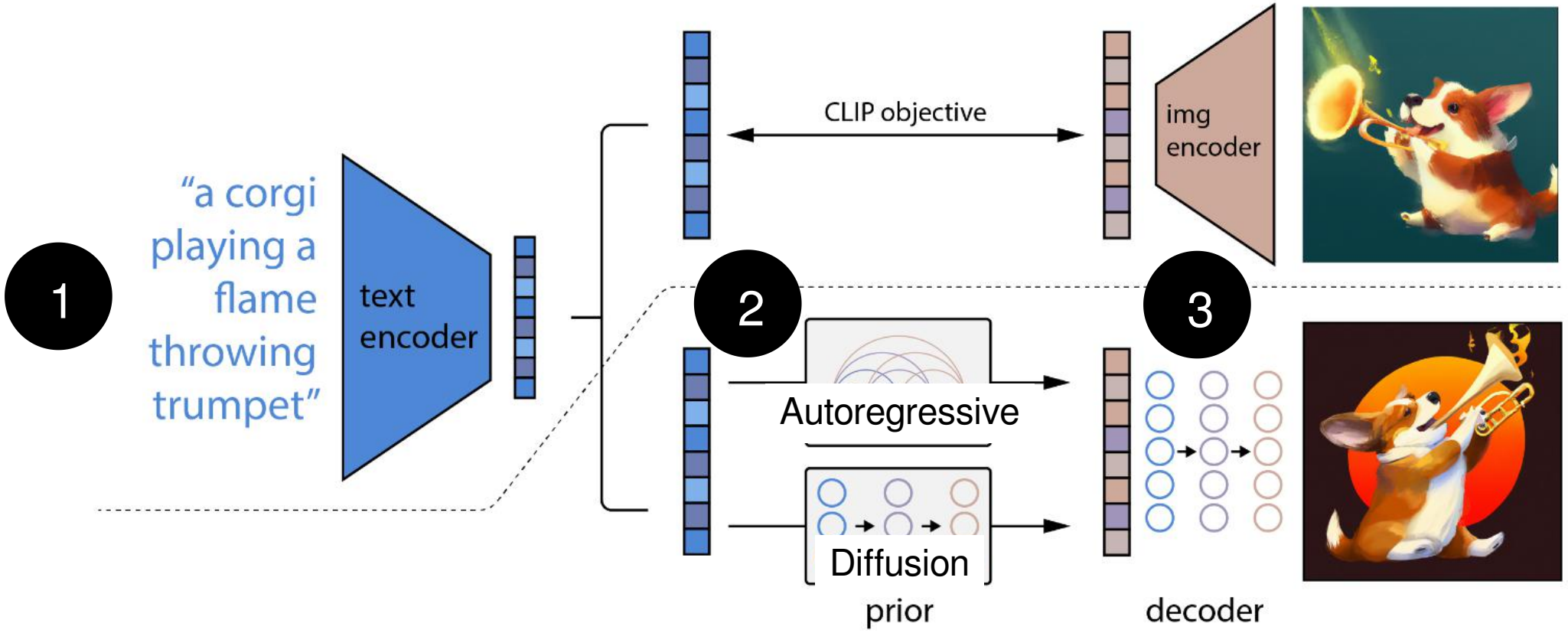


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

DALL-E 2

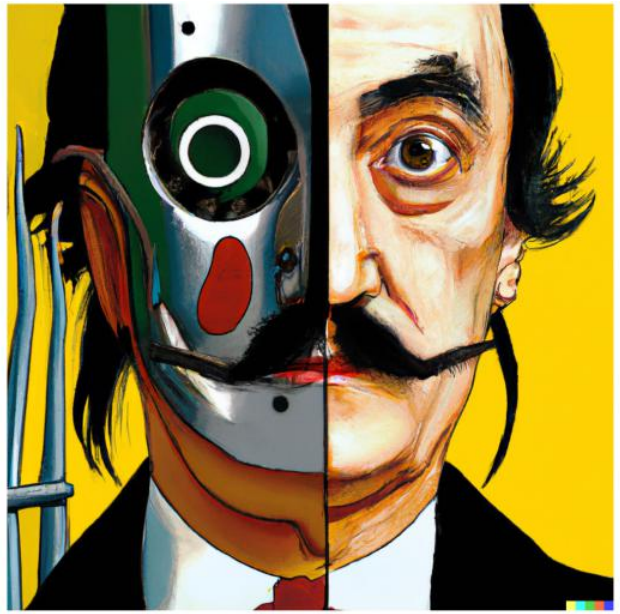
Generate Images From Captions

- A *prior* $P(z_i|y)$ that produces CLIP image embeddings z_i conditioned on captions y.
- A *decoder* $P(x|z_i, y)$ that produces images x conditioned on CLIP image embeddings z_i (and optionally text captions y).

- *Autoregressive (AR)* prior: the CLIP image embedding z_i is converted into a sequence of discrete codes and predicted autoregressively conditioned on the caption y.
- *Diffusion* prior: The continuous vector z_i is directly modelled using a Gaussian diffusion model conditioned on the caption y.

$$L_{\text{prior}} = \mathbb{E}_{t\sim[1,T], z_i^{(t)}\sim q_t}\left[\|f_\theta(z_i^{(t)}, t, y) - z_i\|^2\right]$$

DALL-E 2 - Results



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 - Results

a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

a photo of a landscape in winter → a photo of a landscape in fall

SDEDIT: GUIDED IMAGE SYNTHESIS AND EDITING WITH STOCHASTIC DIFFERENTIAL EQUATIONS

Chenlin Meng[1] **Yutong He**[1] **Yang Song**[1] **Jiaming Song**[1]
Jiajun Wu[1] **Jun-Yan Zhu**[2] **Stefano Ermon**[1]
[1]Stanford University [2]Carnegie Mellon University

SDEdit

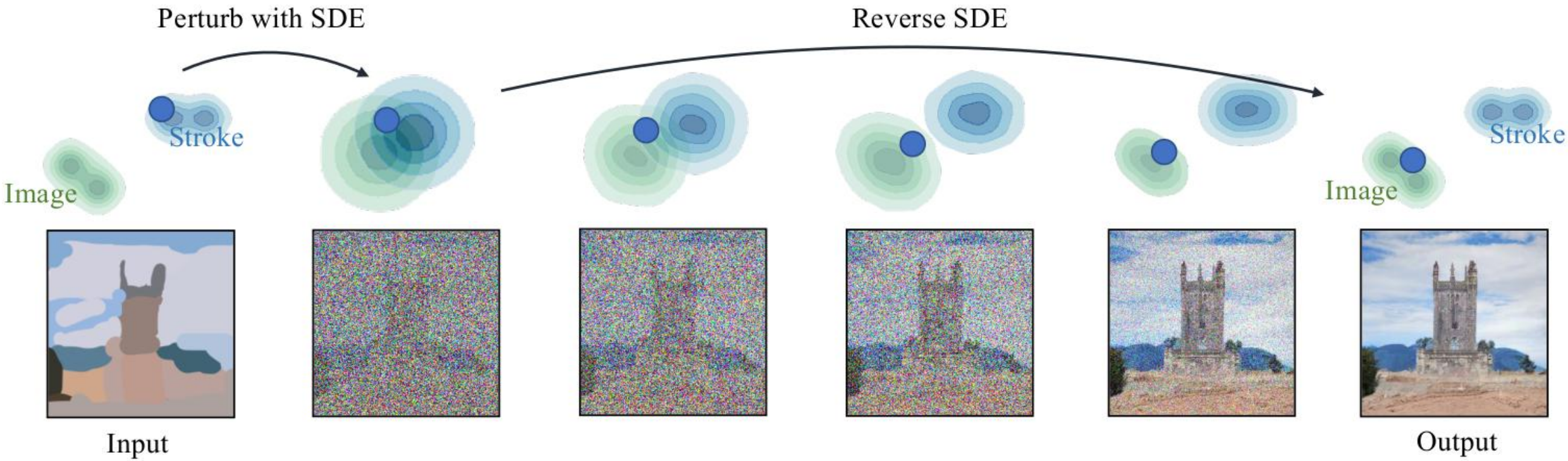


Figure 2: Synthesizing images from strokes with SDEdit. The blue dots illustrate the editing process of our method. The green and blue contour plots represent the distributions of images and stroke paintings, respectively. Given a stroke painting, we first perturb it with Gaussian noise and progressively remove the noise by simulating the reverse SDE. This process gradually projects an unrealistic stroke painting to the manifold of natural images.

SDEdit

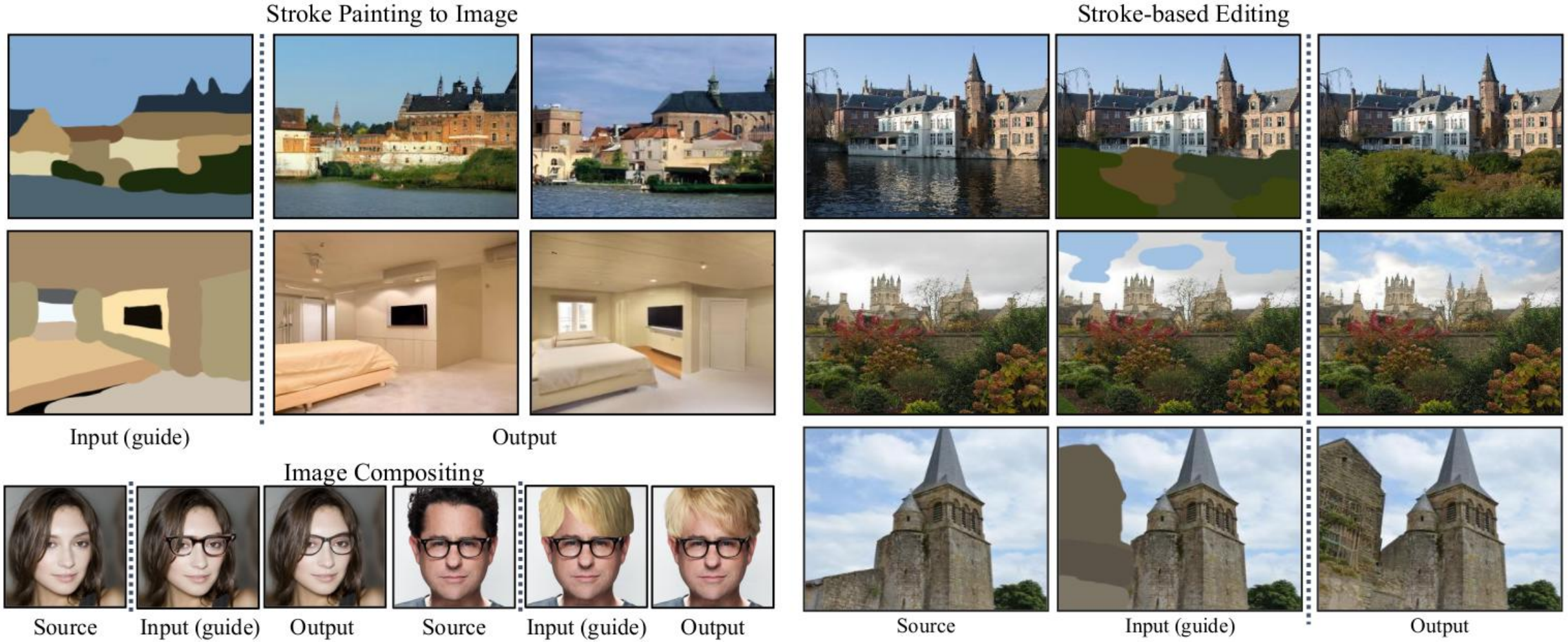


Figure 1: Stochastic Differential Editing (SDEdit) is a **unified** image synthesis and editing framework based on stochastic differential equations. SDEdit allows stroke painting to image, image compositing, and stroke-based editing **without** task-specific model training and loss functions.

SDEdit

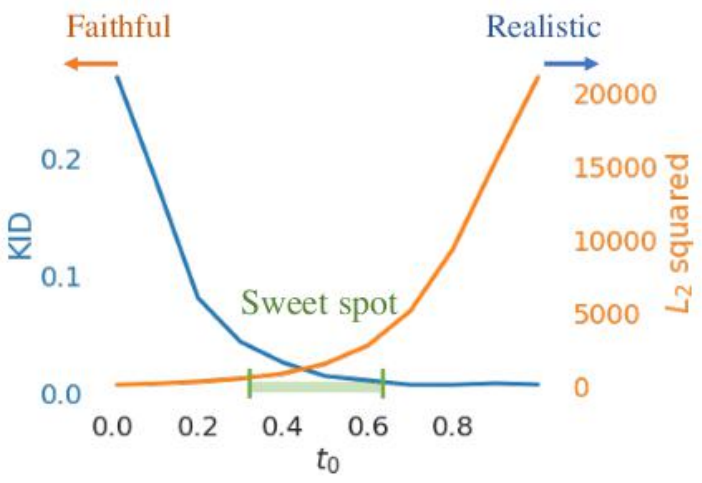
Setup. The user provides a full resolution image $\mathbf{x}^{(g)}$ in a form of manipulating RGB pixels, which we call a "*guide*". The guide may contain different levels of details; a high-level guide contains only coarse colored strokes, a mid-level guide contains colored strokes on a real image, and a low-level guide contains image patches on a target image. We illustrate these guides in Fig. 1, which can be easily provided by non-experts. Our goal is to produce full resolution images with two desiderata:

Realism. The image should appear realistic (*e.g.*, measured by humans or neural networks).

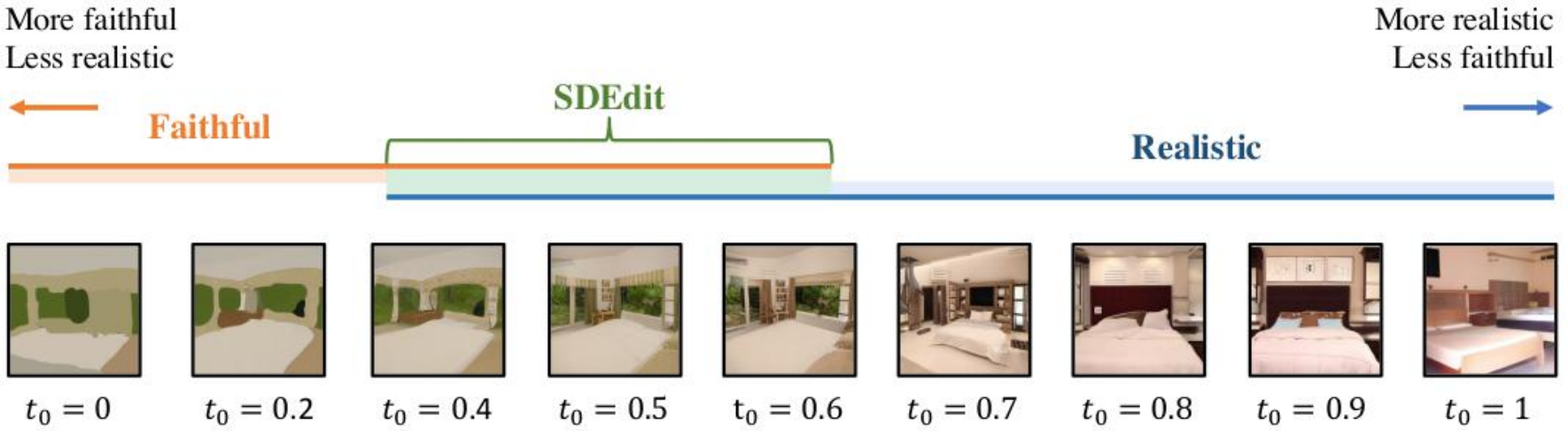
Faithfulness. The image should be similar to the guide $\mathbf{x}^{(g)}$ (*e.g.*, measured by L_2 distance).

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}(0) + \sigma(t)\mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

SDEdit



(a) KID and L_2 norm **squared** plot with respect to t_0.



(b) We illustrate synthesized images of SDEdit with various t_0 initializations. $t_0 = 0$ indicates the guide itself, whereas $t_0 = 1$ indicates a random sample.

Figure 3: Trade-off between faithfulness and realism for stroke-based generation on LSUN. As t_0 increases, the generated images become **more realistic** while **less faithful**. Given an input, SDEdit aims at generating an image that is both faithful and realistic, which means that we should choose t_0 appropriately ($t_0 \in [0.3, 0.6]$ in this example).

SDEdit

Algorithm 1 Guided image synthesis and editing with SDEdit (VE-SDE)

Require: $\mathbf{x}^{(g)}$ (guide), t_0 (SDE hyper-parameter), N (total denoising steps)

$\Delta t \leftarrow \frac{t_0}{N}$

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

$\mathbf{x} \leftarrow \mathbf{x} + \sigma(t_0)\mathbf{z}$

for $n \leftarrow N$ **to** 1 **do**

$t \leftarrow t_0 \frac{n}{N}$

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

$\epsilon \leftarrow \sqrt{\sigma^2(t) - \sigma^2(t - \Delta t)}$

$\mathbf{x} \leftarrow \mathbf{x} + \epsilon^2 \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t) + \epsilon\mathbf{z}$

end for

Return $\mathbf{x}$

SDEdit - Results

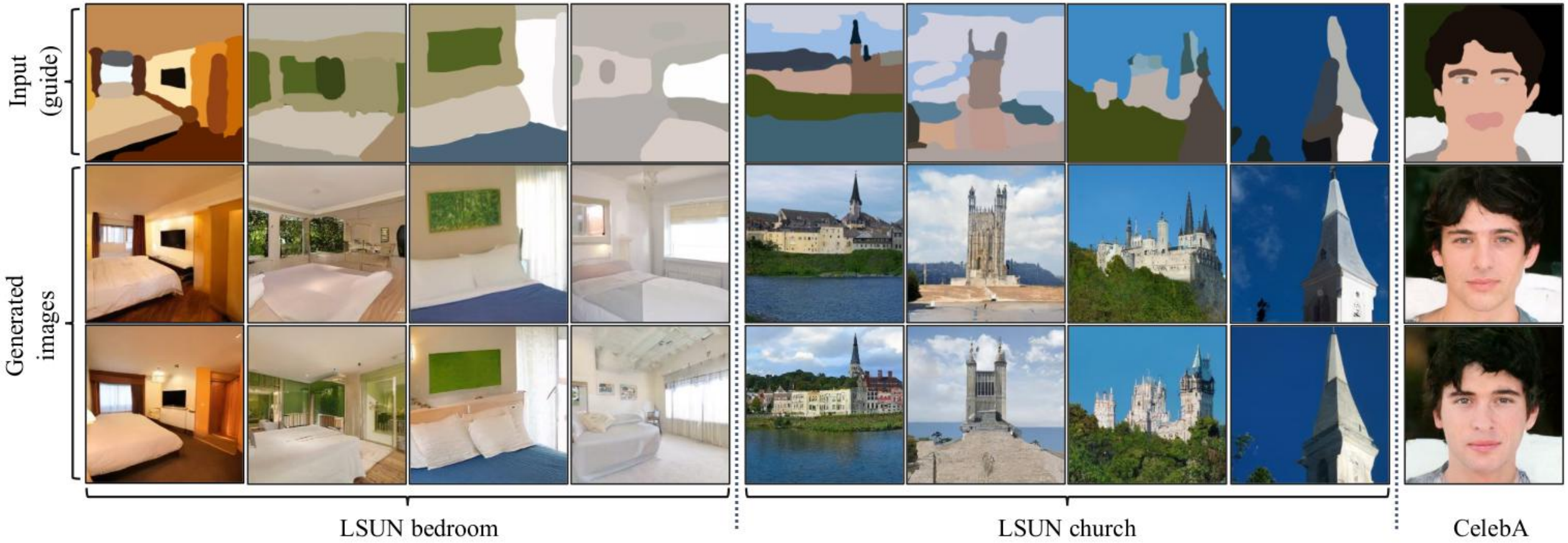
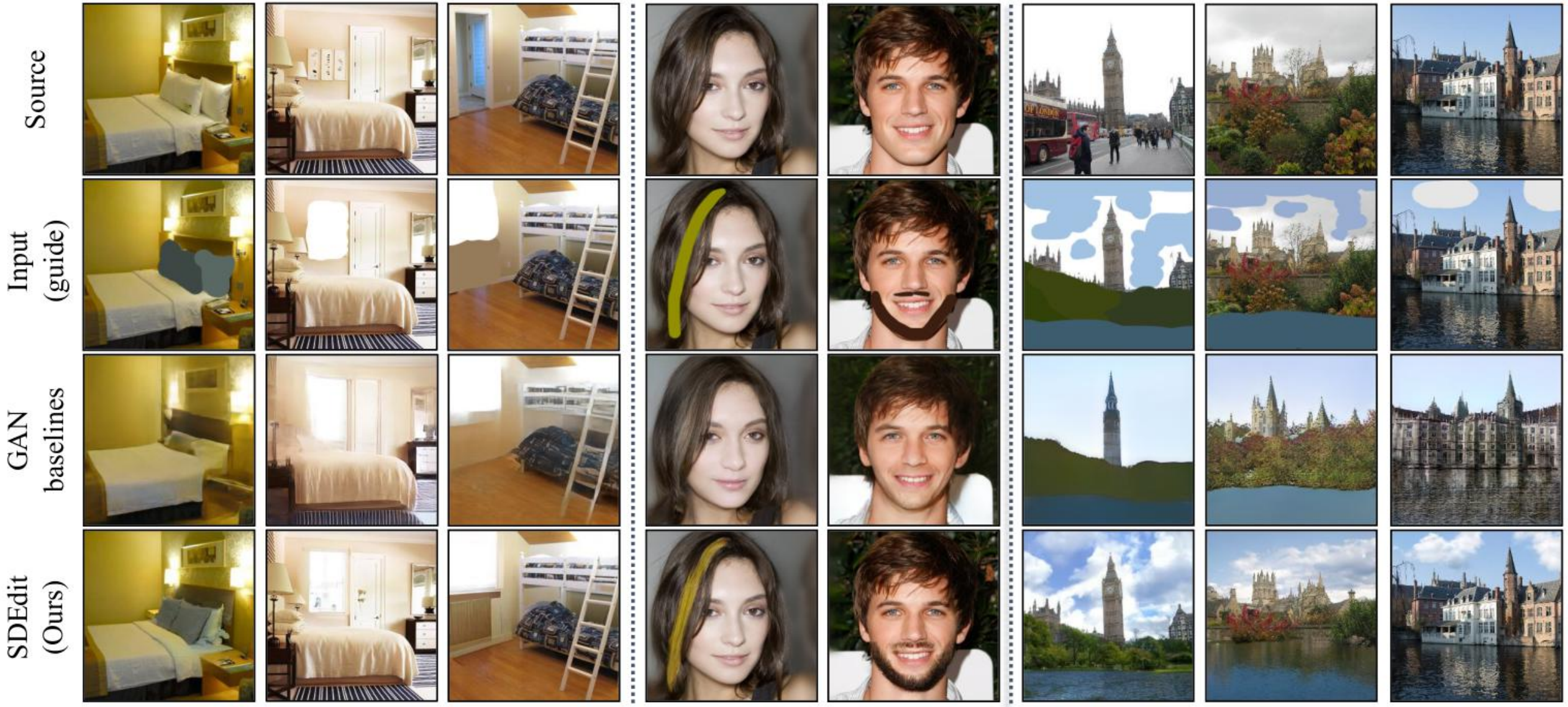
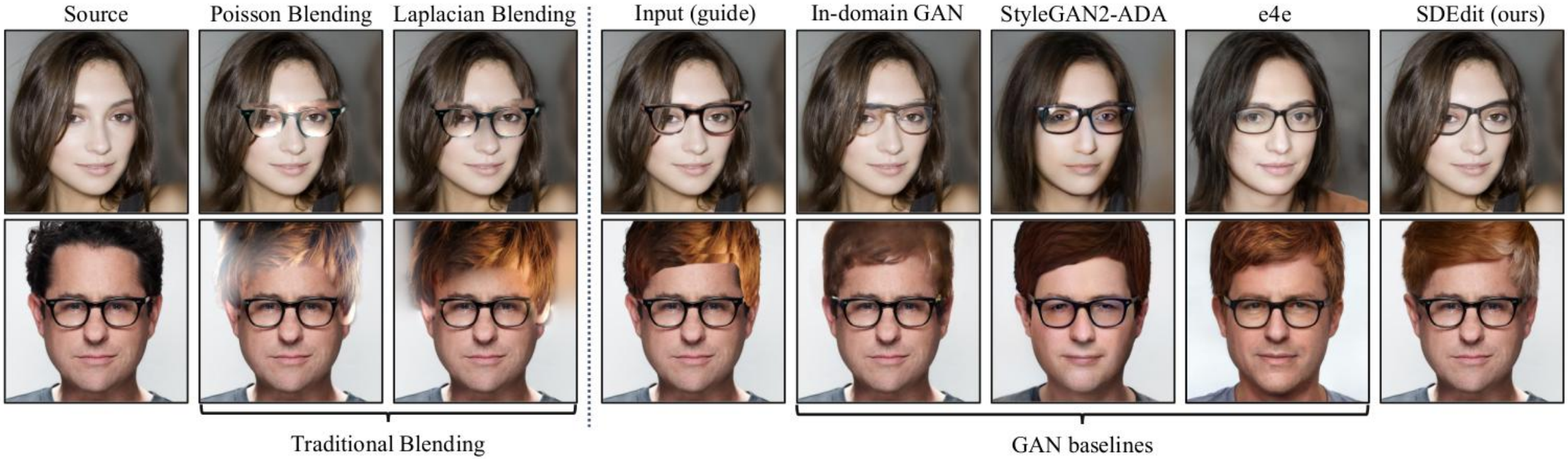


Figure 5: SDEdit can generate realistic, faithful and diverse images for a given stroke input drawn by human.

SDEdit - Results



SDEdit - Results



SDEdit - Results

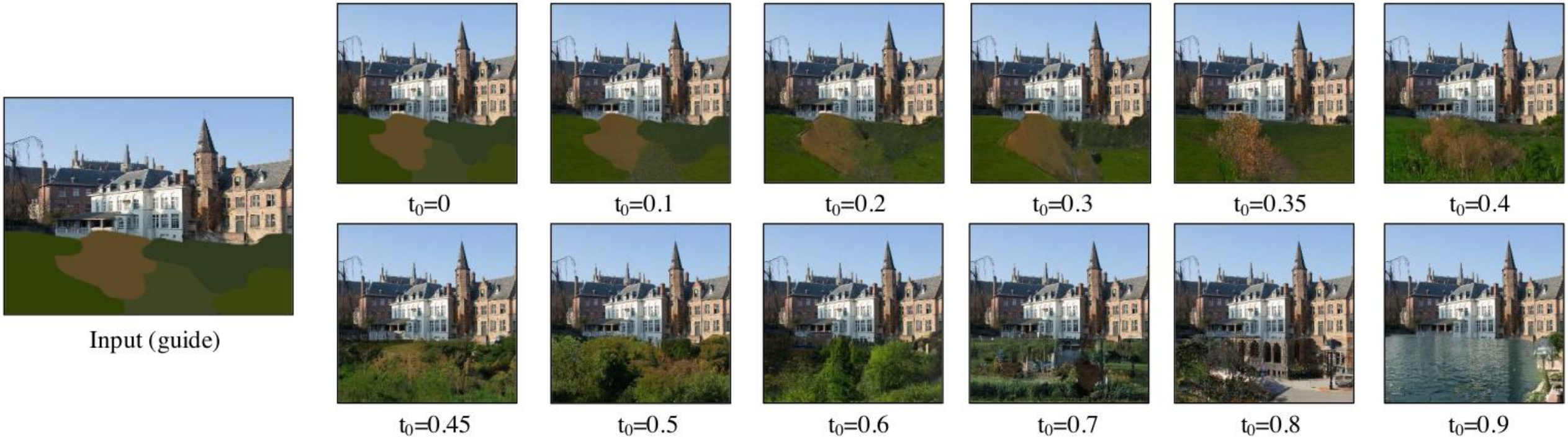


Figure 12: Extra analysis on t_0. As t_0 increases, the generated images become more realistic while less faithful.