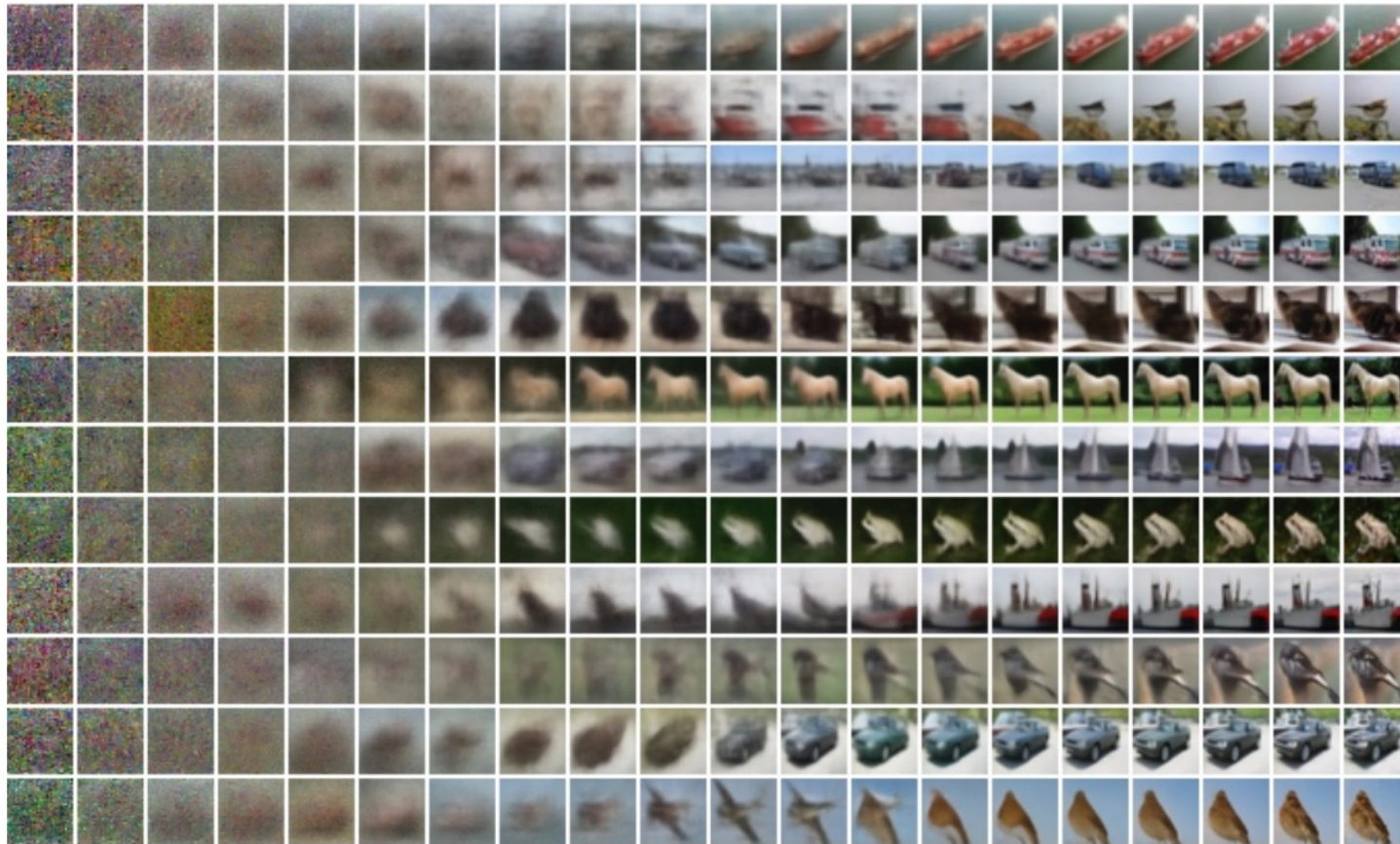


Diffusion models



Diffusion models



<https://www.nytimes.com/2023/04/08/technology/ai-photos-pope-francis.html>

Outline

Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

Part 2: Recent Advances

- Denoising diffusion implicit models (DDIMs)
- Stable Diffusion XL, Stable Diffusion 3
- Progressive Distillation
- Latent Consistency Models (LCM)

(next class) Part 3: Implementation, Applications, and Ethical issues

Denoising diffusion probabilistic models (DDPMs)

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley

jonathanho@berkeley.edu

Ajay Jain
UC Berkeley

ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley

pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at <https://github.com/hojonathanho/diffusion>.

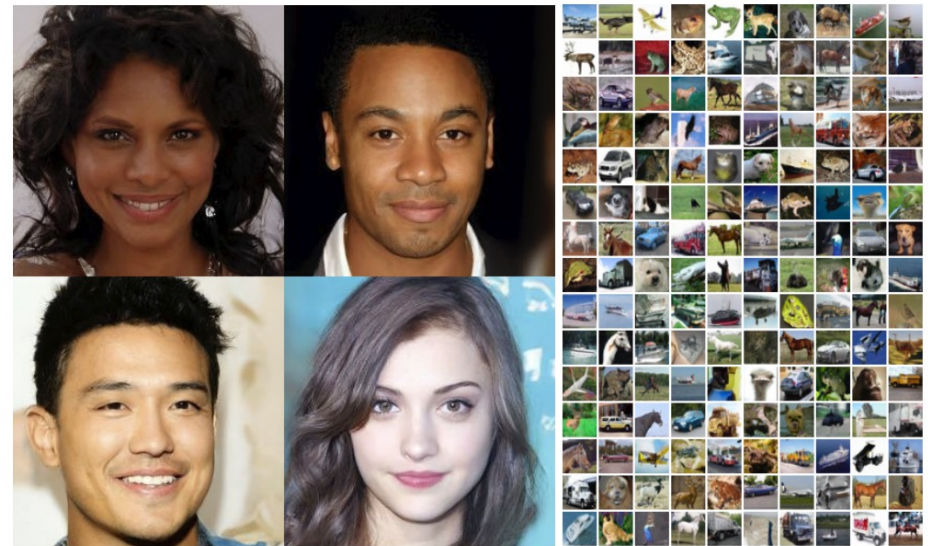
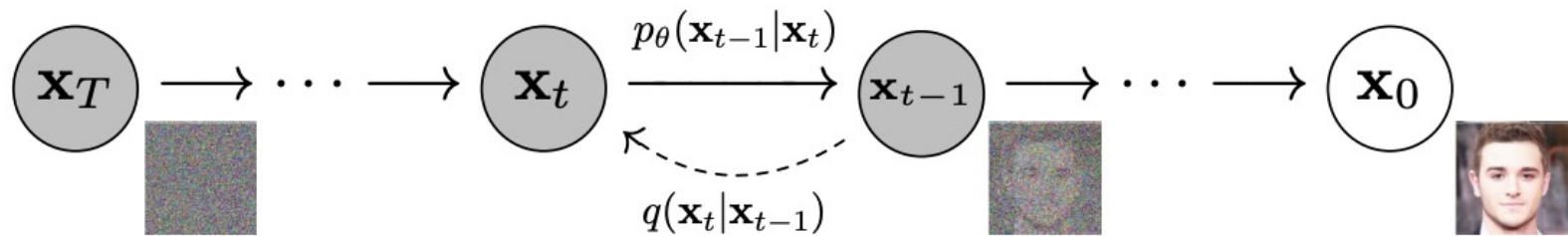


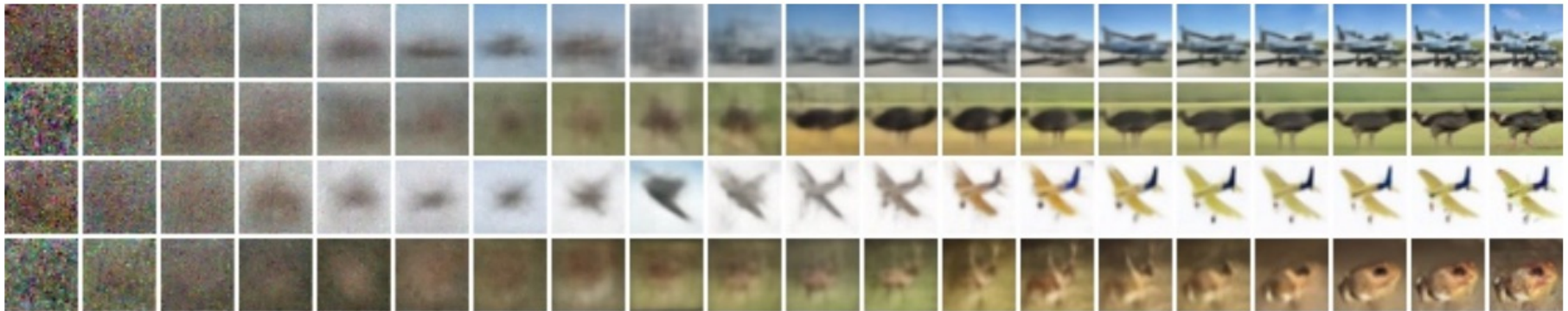
Figure 1: Generated samples on CelebA-HQ 256×256 (left) and unconditional CIFAR10 (right)

J. Ho et al. [Denoising diffusion probabilistic models](#). NeurIPS 2020

DDPMs: Basic idea



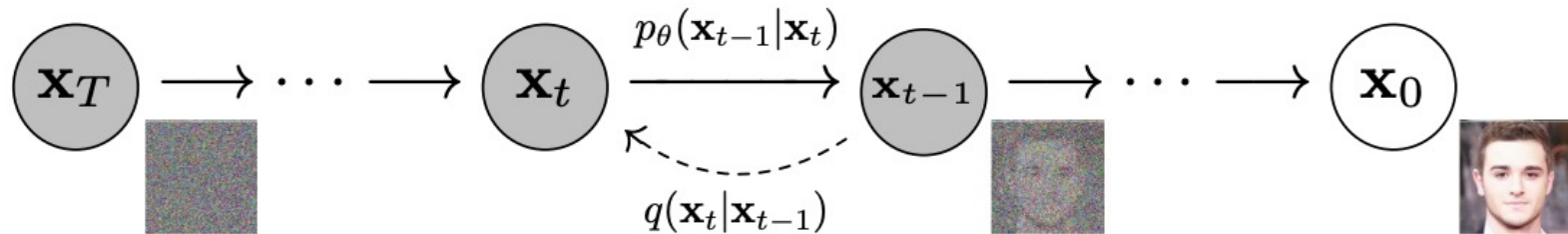
Unconditional CIFAR10 sample generation



J. Ho et al. [Denoising diffusion probabilistic models](#). NeurIPS 2020

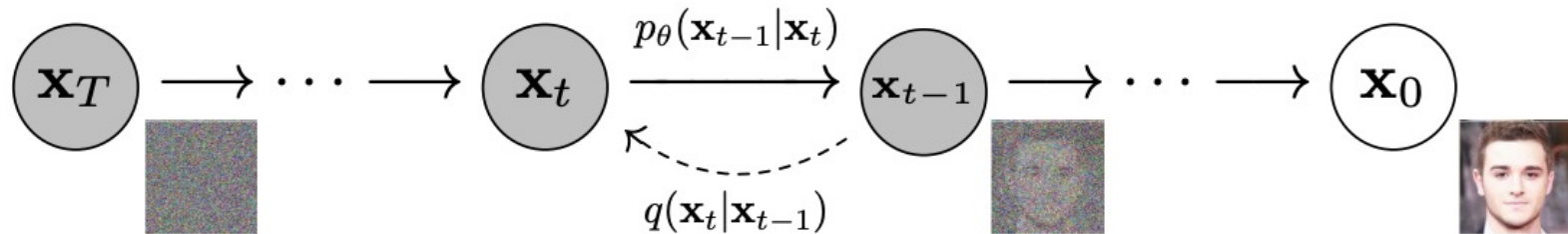
Blog introduction: <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>
[CVPR 2022 tutorial](#)

DDPMs: Basic idea



- *Forward process* q turns images into Gaussian noise
- *Reverse process* p turns noise into images
- Provided the increments of t are small enough, $p_\theta(x_{t-1} | x_t)$ is Gaussian and we can train a neural network to estimate the mean of x_{t-1} given x_t

DDPMs: Basic idea

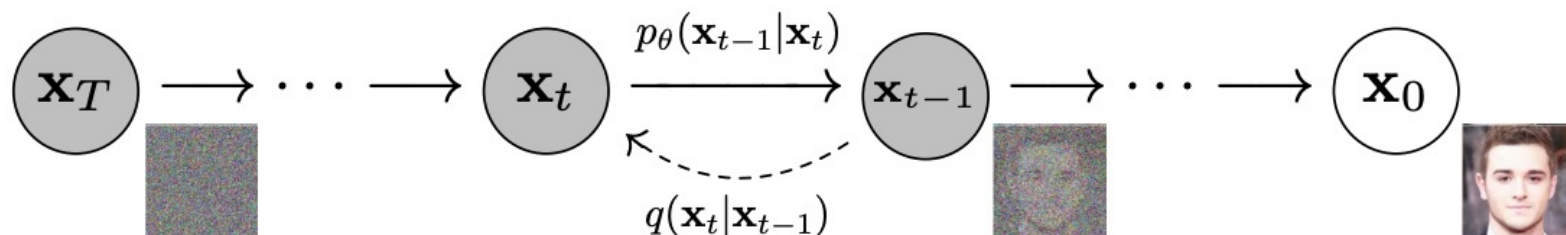


Algorithm 1 Training

- 1: **repeat**
 - 2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
 - 3: $t \sim \text{Uniform}(\{1, \dots, T\})$
 - 4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 5: Take gradient descent step on
 $\nabla_\theta \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2$
 - 6: **until** converged
-

- $\epsilon_\theta(x_t, t)$ is the predicted noise component of image x_t given noise level t
- Network parameters θ are updated to reduce L2 error between actual noise ϵ and predicted noise $\epsilon_\theta(x_t, t)$

DDPMs: Basic idea



Algorithm 1 Training

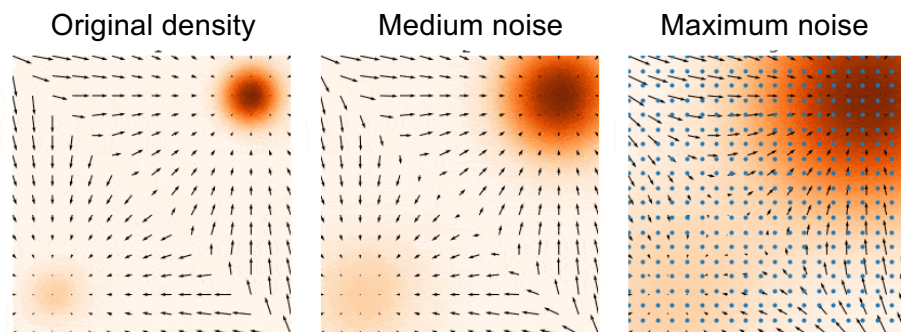
- 1: **repeat**
 - 2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
 - 3: $t \sim \text{Uniform}(\{1, \dots, T\})$
 - 4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 5: Take gradient descent step on
$$\nabla_\theta \left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|^2$$
 - 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: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
 - 5: **end for**
 - 6: **return** \mathbf{x}_0
-

Alternate viewpoint: Score-based generative modeling

- It can be shown that $\epsilon_{\theta}(x_t, t) \approx -\nabla_{x_t} \log q(x_t)$, where $\nabla_{x_t} \log q(x_t)$ is the *score function* of the (noisy) data distribution
- To sample from the original data density $q(x_0)$, we can use *annealed Langevin dynamics*, i.e., start by sampling from noise-perturbed versions of the data distribution and gradually reduce the amount of noise

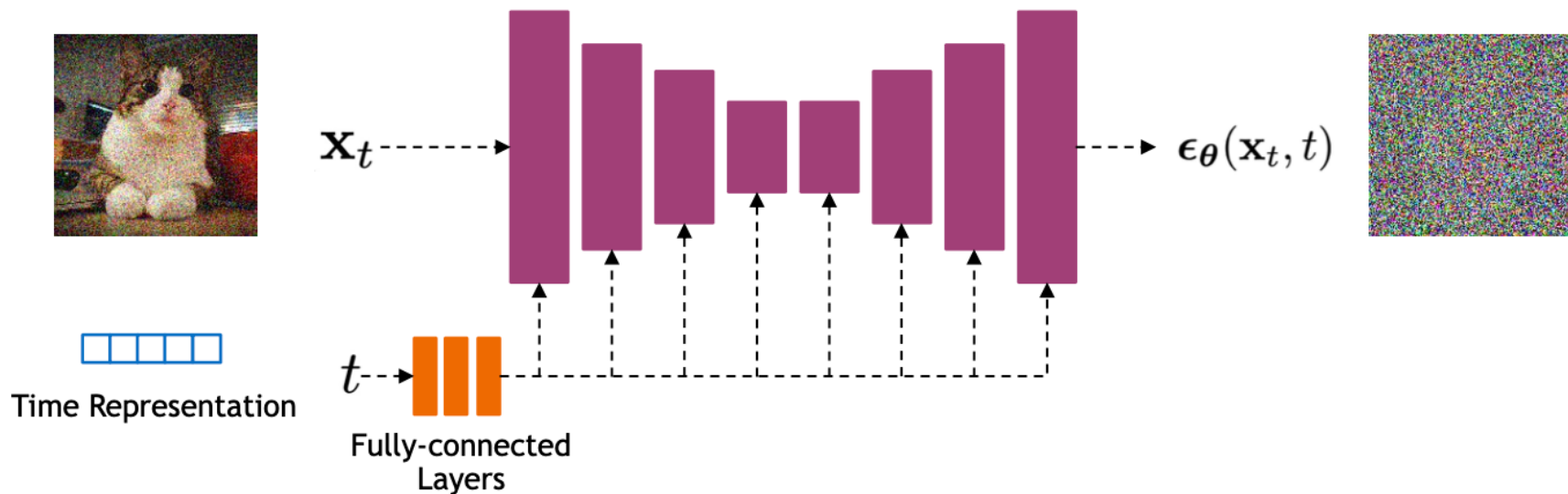


Algorithm 2 Sampling

- 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: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
 - 5: **end for**
 - 6: **return** \mathbf{x}_0
-

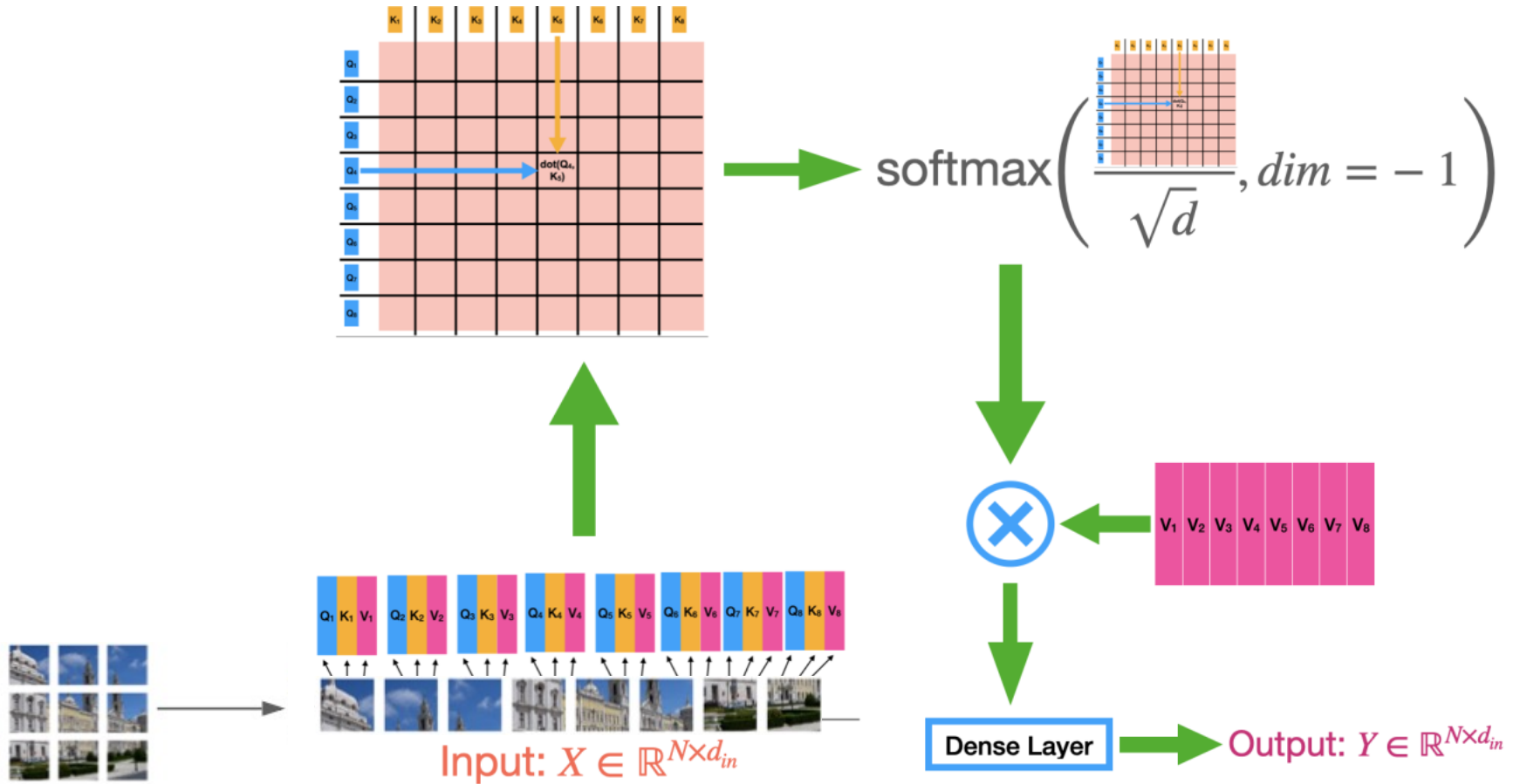
DDPMs: Implementation

- U-Net architectures are typically used to represent $\epsilon_{\theta}(x_t, t)$
 - Bells and whistles: residual blocks, self-attention



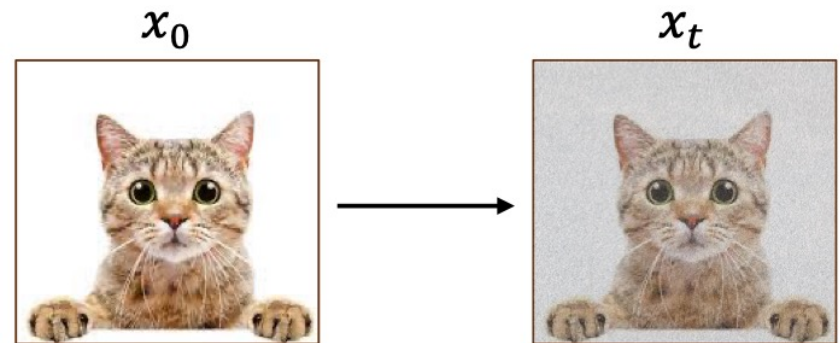
- Time is encoded using sinusoidal positional embeddings or random Fourier features, fed into the U-Net using addition or adaptive normalization

Attention Module

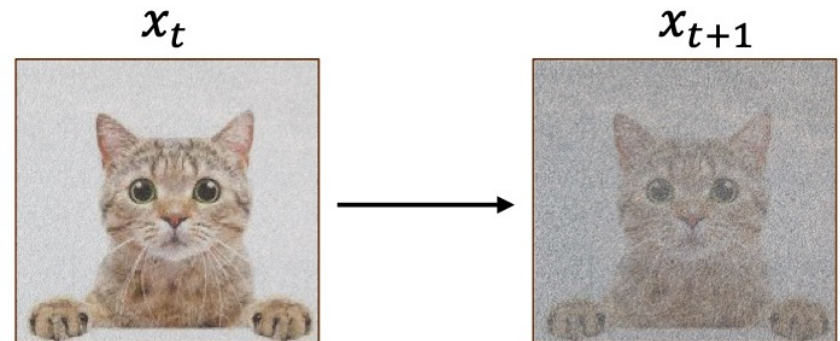


How do we do this in practice?

Step 1: Sample image from the dataset, generate noisy image using forward process



Step 2: Given noisy image, generate slightly noisier image



How do we do this in practice?

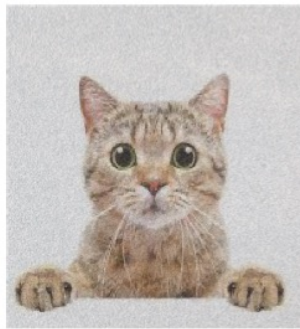
During Training

x_{t+1}



U-net

\hat{x}_t



Loss: $MSE(x_t, \hat{x}_t)$

During Inference

Input



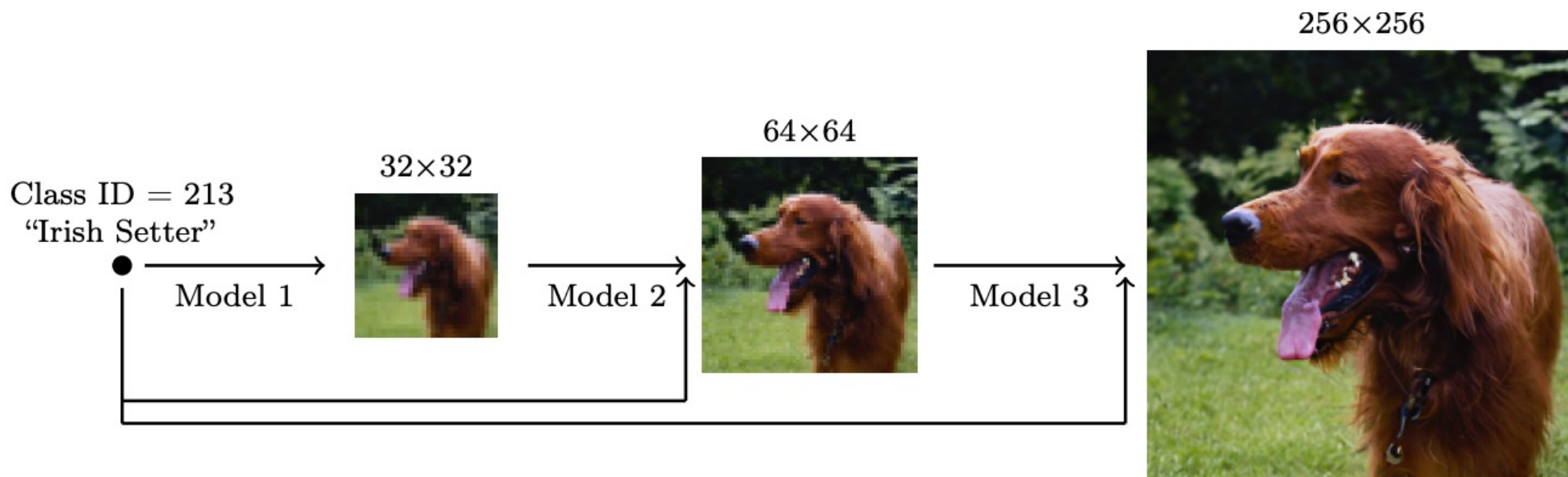
U-net

Output



U-net predicts the noise

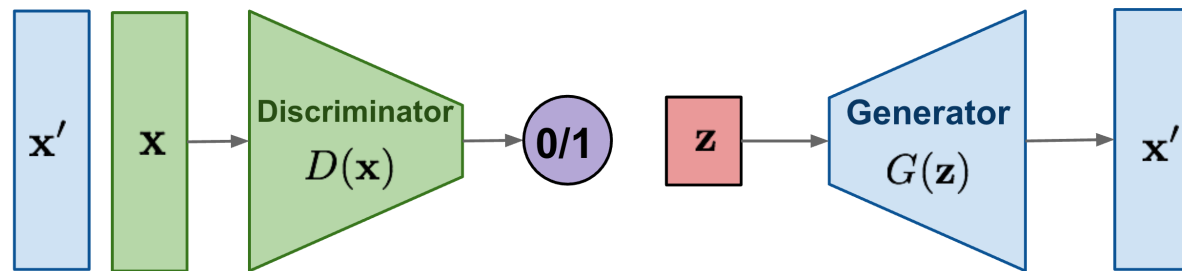
Efficient sampling at high resolutions: Cascaded generation



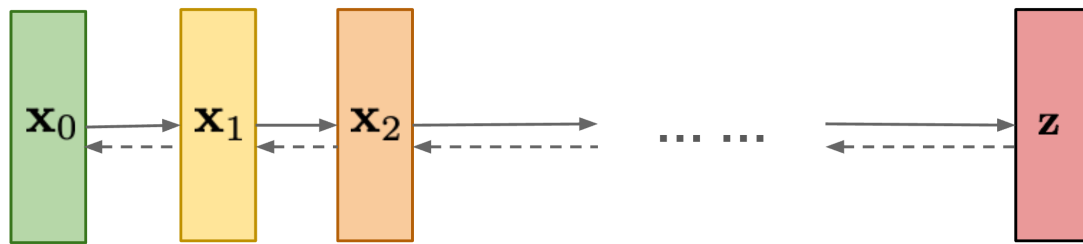
- In practice, data augmentation for inputs to upsampling models is crucial (esp. adding Gaussian noise or early stopping for base model)

GANs vs. VAEs vs. Diffusion Models

GAN: Adversarial training



Diffusion models:
Gradually add Gaussian noise and then reverse



Outline

Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models

Class-conditioned DDPMs

- “We can sample with as few as 25 forward passes while maintaining FIDs comparable to BigGAN”

Abstract

We show that diffusion models can achieve image sample quality superior to the current state-of-the-art generative models. We achieve this on unconditional image synthesis by finding a better architecture through a series of ablations. For conditional image synthesis, we further improve sample quality with classifier guidance: a simple, compute-efficient method for trading off diversity for fidelity using gradients from a classifier. We achieve an FID of 2.97 on ImageNet 128×128 , 4.59 on ImageNet 256×256 , and 7.72 on ImageNet 512×512 , and we match BigGAN-deep even with as few as 25 forward passes per sample, all while maintaining better coverage of the distribution. Finally, we find that classifier guidance combines well with upsampling diffusion models, further improving FID to 3.94 on ImageNet 256×256 and 3.85 on ImageNet 512×512 . We release our code at <https://github.com/openai/guided-diffusion>.

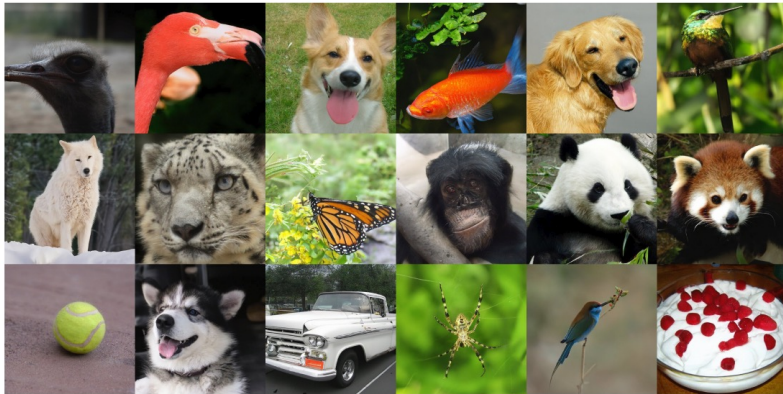


Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

Classifier guidance

- We can sample from the class-conditional density $q(x_t|c)$ with the help of a pre-trained classifier $P(c|x_t)$
- Bayes rule:

$$q(x_t|c) \propto P(c|x_t)q(x_t)$$

$$\log q(x_t|c) = \log P(c|x_t) + \log q(x_t) + \text{const.}$$

$$\nabla_{x_t} \log q(x_t|c) = \nabla_{x_t} \log P(c|x_t) + \nabla_{x_t} \log q(x_t)$$

conditional score
function

obtained from classifier
output

unconditional score
function (pre-trained)

- To sample from class c , steer sample in the modified direction $\nabla_{x_t} [\log q(x_t) + w \log P(c|x_t)]$

Classifier-free guidance

- Instead of training an additional classifier, get an “implicit classifier” by jointly training a conditional and unconditional diffusion model: $P(c|x_t) \propto q(x_t|c)/q(x_t)$
- Both $q(x_t|c)$ and $q(x_t)$ are represented using the same network, trained by dropping out c with some probability (corresponding to the unconditional case)
- The modified score function corresponding to this implicit classifier is

$$\begin{aligned} & \nabla_{x_t} [\log q(x_t) + w \log P(c|x_t)] \\ = & \nabla_{x_t} [\log q(x_t) + w(\log q(x_t|c) - \log q(x_t))] \end{aligned}$$

Sample is steered away from the unconditional distribution in the direction of the conditional one

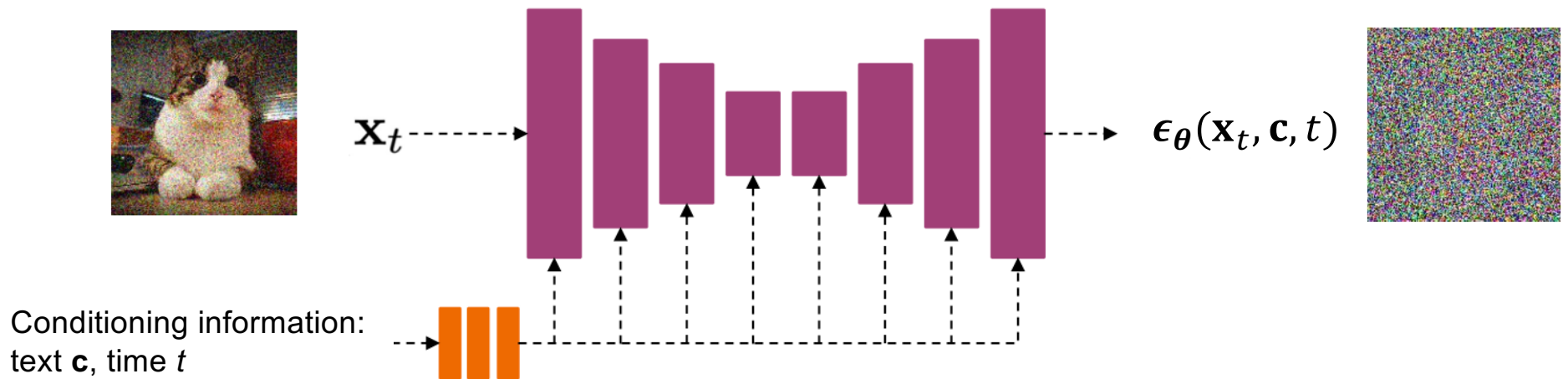
Classifier-free guidance



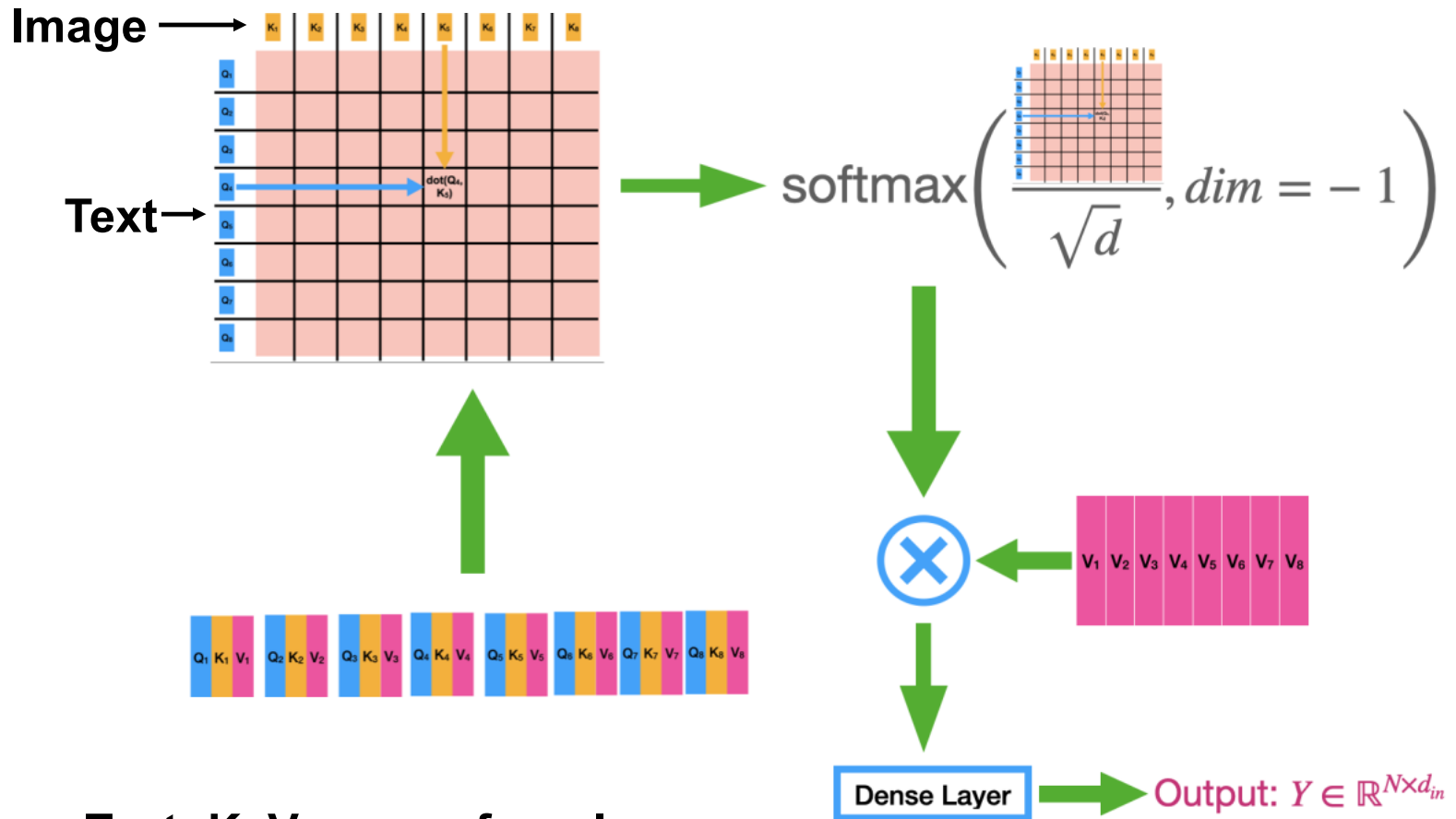
Figure 1: Classifier-free guidance on the malamute class for a 64x64 ImageNet diffusion model. Left to right: increasing amounts of classifier-free guidance, starting from non-guided samples on the left.

Text-guided diffusion

- Instead of a class label, c can be an encoded text prompt, injected into the U-Net using *cross-attention*



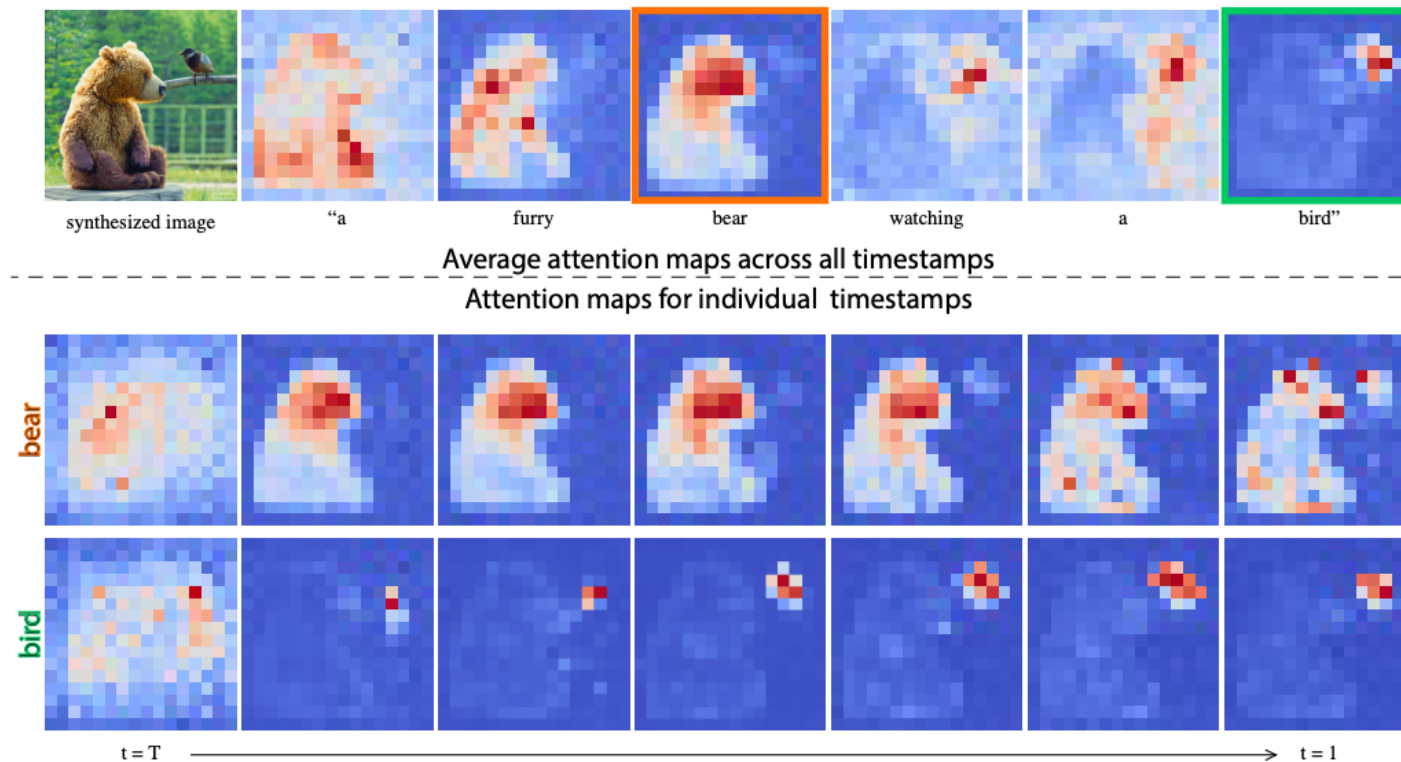
Cross Attention Module



Q comes from Text; K, V comes from Image

Text-guided diffusion

- Instead of a class label, c can be an encoded text prompt, injected into the U-Net using *cross-attention*

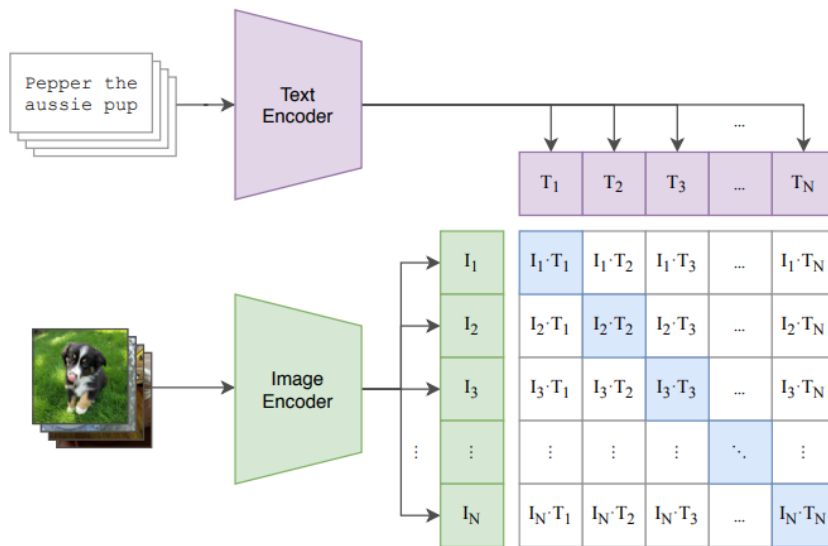


Text-guided diffusion

- Instead of a class label, c can be an encoded text prompt, injected into the U-Net using *cross-attention*
- Classifier-free guidance works the same way as before, by training both conditional and unconditional models using text dropout
- CLIP guidance: steer samples in the direction of $\nabla_{x_t} \text{CLIP}(x_t, c)$
- Note: both classifier and CLIP must be *noise-aware* (trained on noised images)

CLIP

(1) Contrastive pre-training



Contrastive objective: in a batch of N image-text pairs, classify each text string to the correct image and vice versa

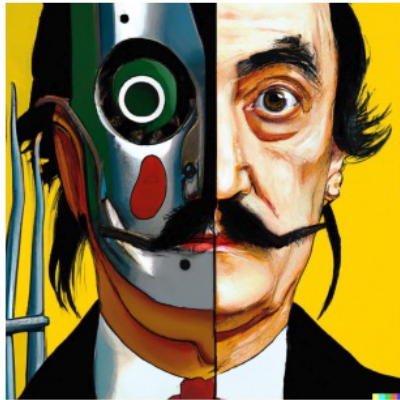
A. Radford et al., [Learning Transferable Visual Models From Natural Language Supervision](https://openai.com/blog/clip/), ICML 2021
<https://openai.com/blog/clip/>

Outline

Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

DALL-E 2



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



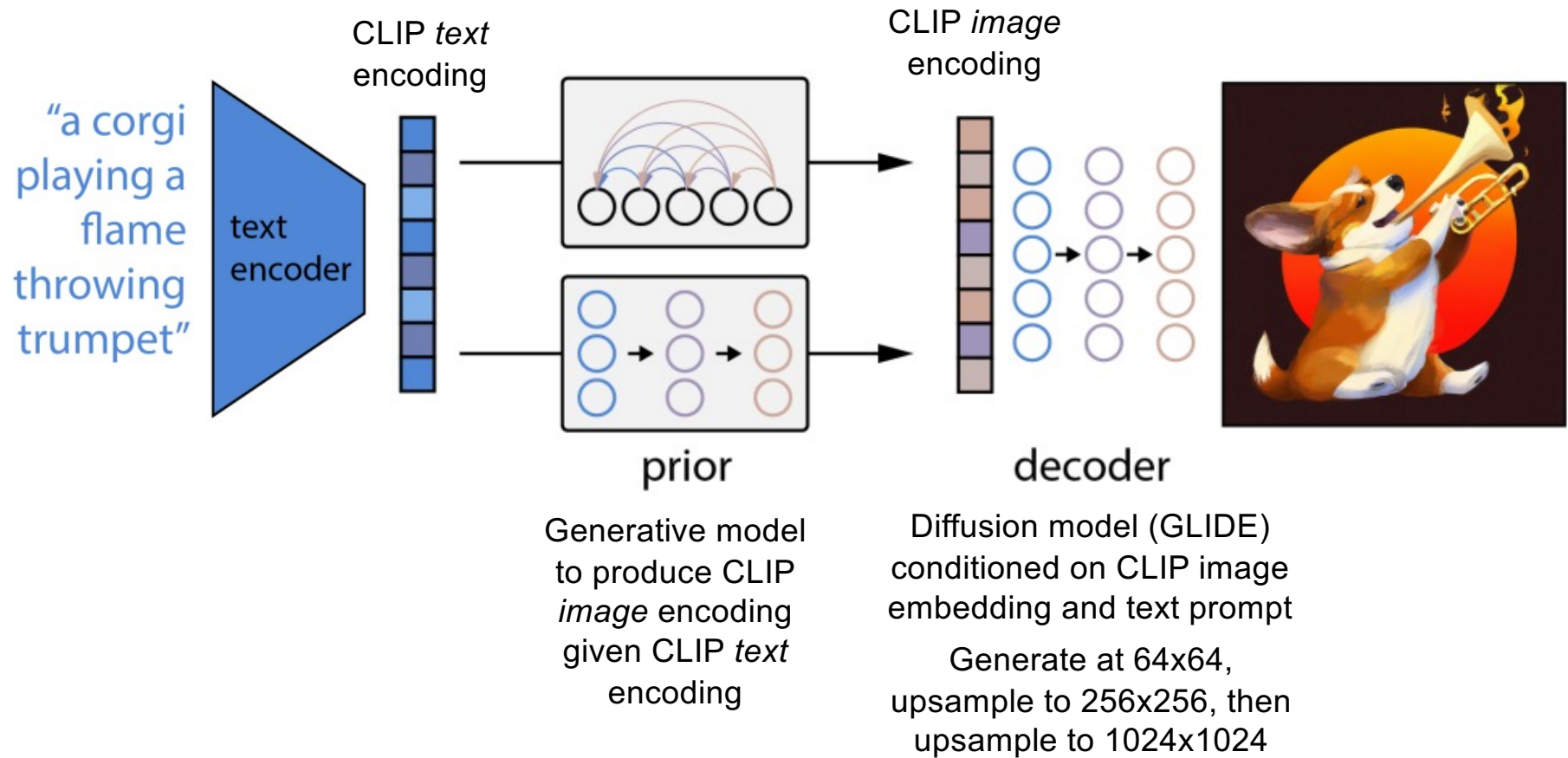
panda mad scientist mixing sparkling chemicals, artstation



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

A. Ramesh et al. [Hierarchical text-conditional image generation with CLIP latents](#). Preprint 2022

DALL-E 2



DALL-E 2: Results



Figure 19: Random samples from unCLIP for prompt "A close up of a handpalm with leaves growing from it."



Figure 18: Random samples from unCLIP for prompt "Vibrant portrait painting of Salvador Dali with a robotic half face"

DALL-E 2: Results



Figure 3: Variations of an input image by encoding with CLIP and then decoding with a diffusion model. The variations preserve both semantic information like presence of a clock in the painting and the overlapping strokes in the logo, as well as stylistic elements like the surrealism in the painting and the color gradients in the logo, while varying the non-essential details.

DALL-E 2: Results

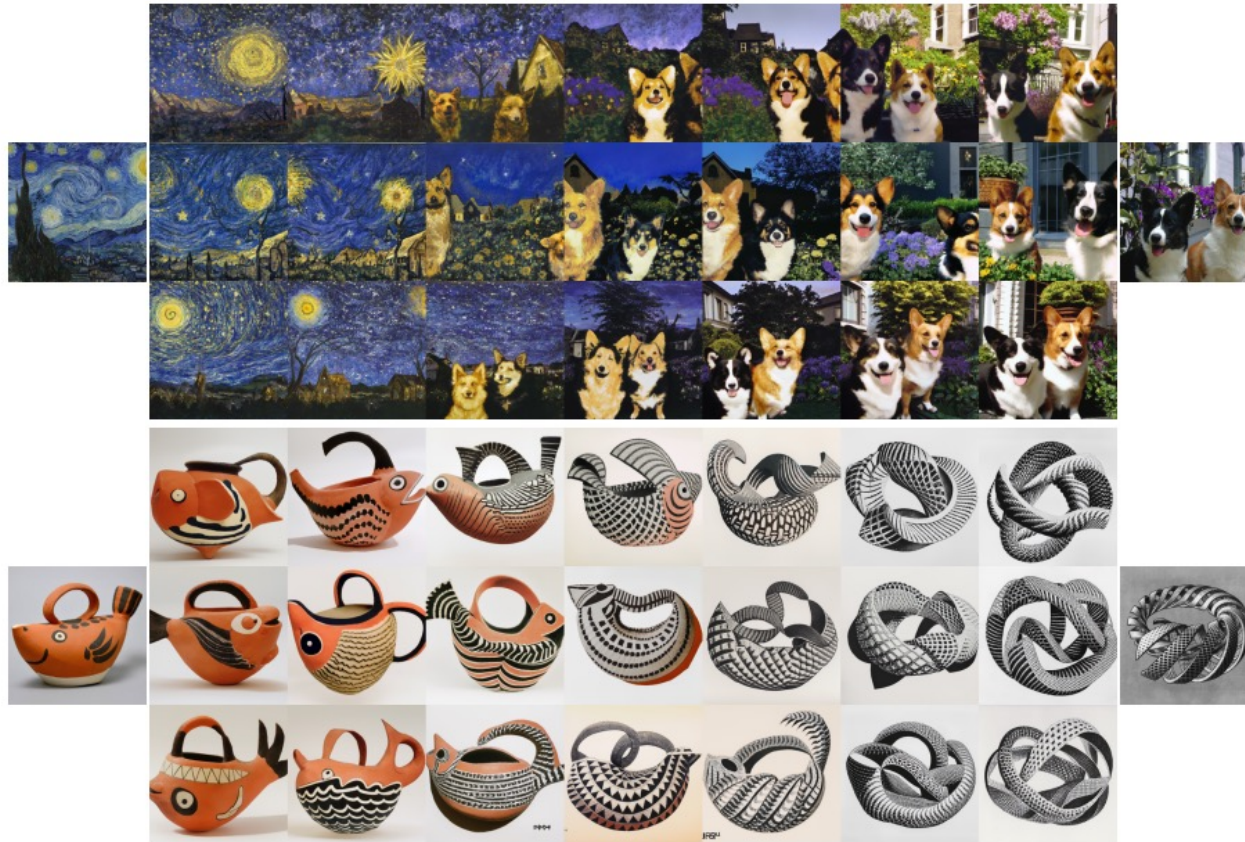


Figure 4: Variations between two images by interpolating their CLIP image embedding and then decoding with a diffusion model. We fix the decoder seed across each row. The intermediate variations naturally blend the content and style from both input images.

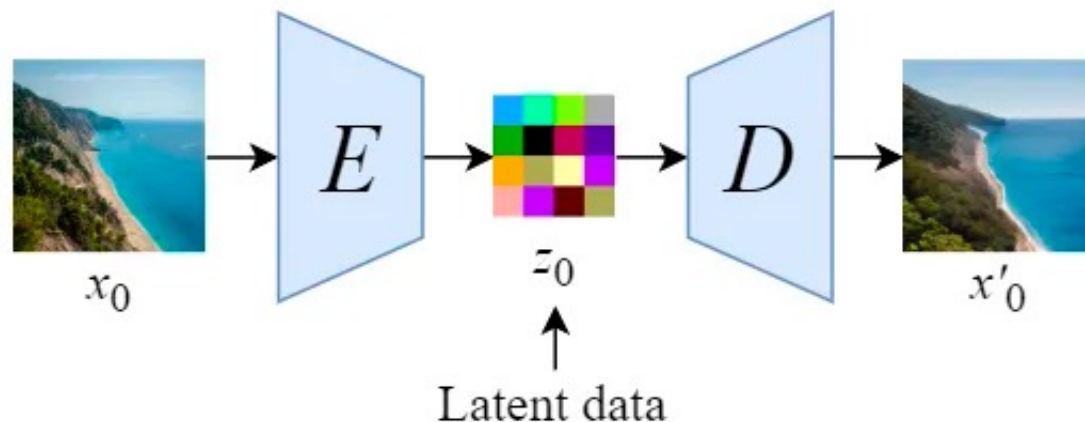
DALL-E 2: Limitations



Figure 15: Reconstructions from the decoder for difficult binding problems. We find that the reconstructions mix up objects and attributes. In the first two examples, the model mixes up the color of two objects. In the rightmost example, the model does not reliably reconstruct the relative size of two objects.

Latent diffusion model (basis of Stable Diffusion)

- Key idea: train a separate *encoder* and *decoder* to convert images to and from a lower-dimensional latent space, run conditional diffusion model in latent space

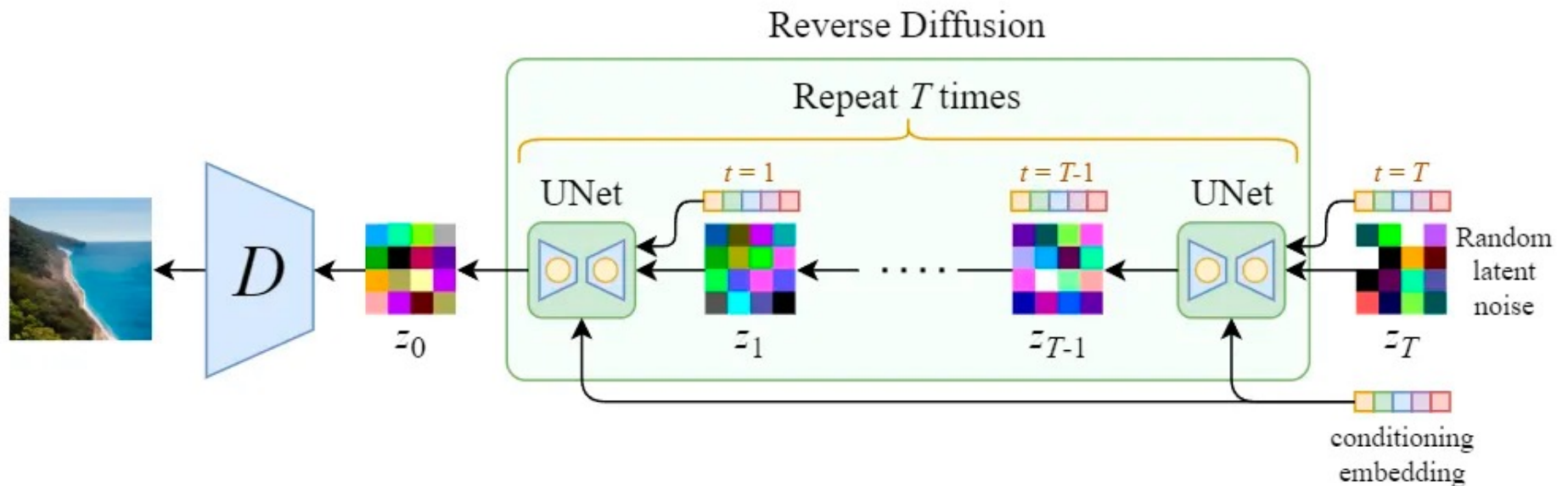


R. Rombach et al. [High-Resolution Image Synthesis with Latent Diffusion Models](#). CVPR 2022

<https://medium.com/@steinsfu/stable-diffusion-clearly-explained-ed008044e07e>

Latent diffusion model (basis of Stable Diffusion)

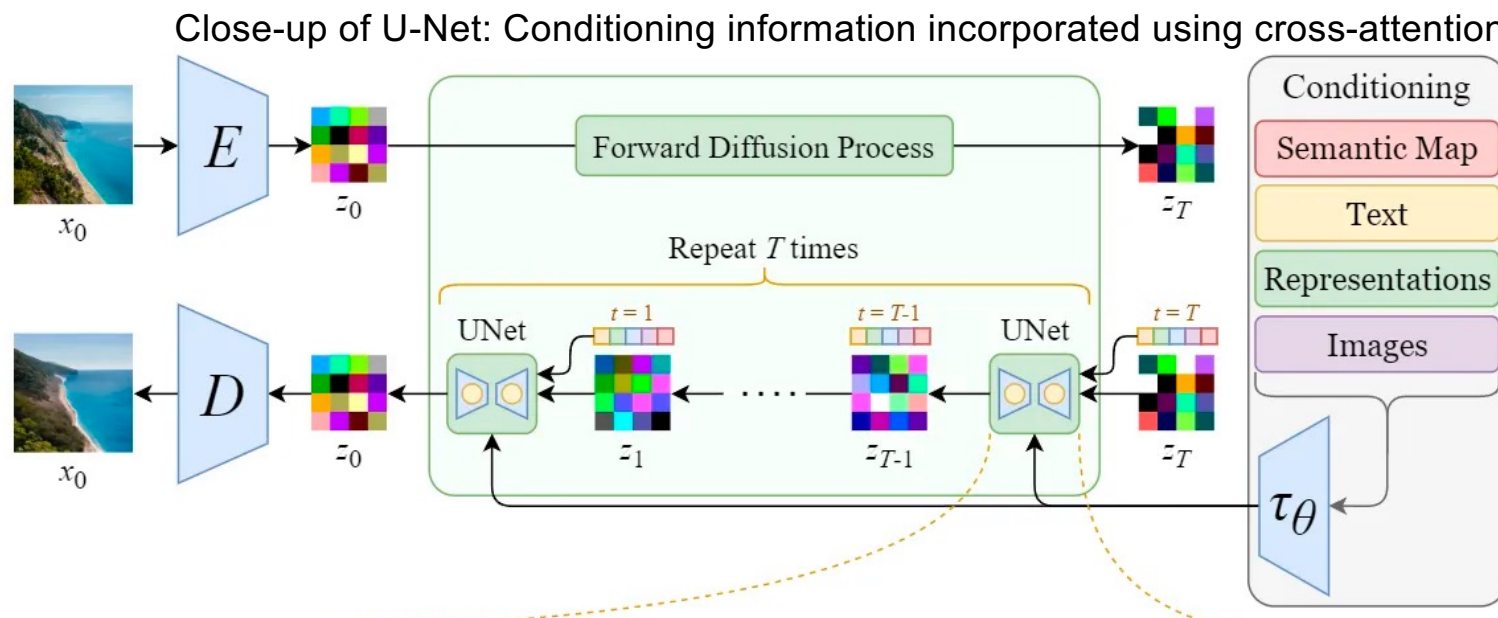
- Key idea: train a separate *encoder* and *decoder* to convert images to and from a lower-dimensional latent space, run conditional diffusion model in latent space



<https://medium.com/@steinsfu/stable-diffusion-clearly-explained-ed008044e07e>

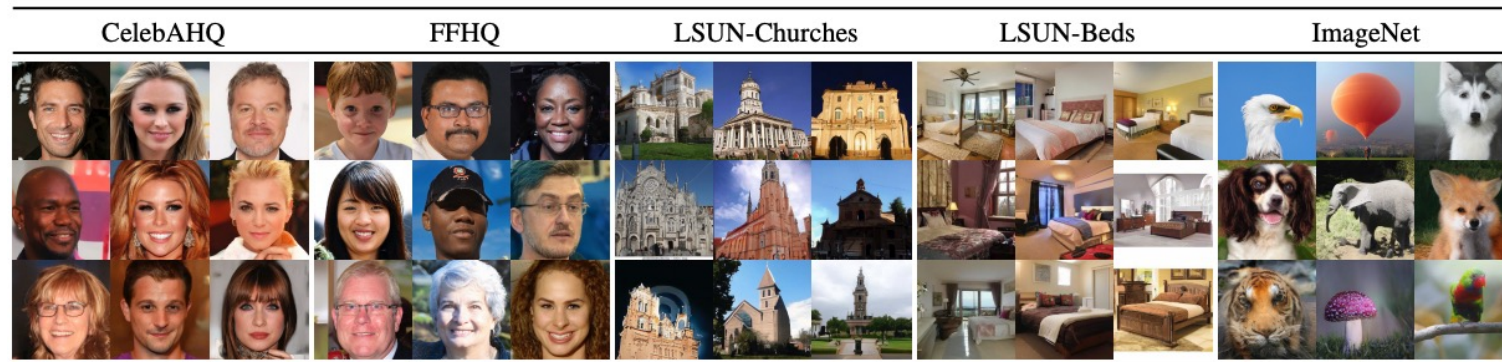
Latent diffusion model (basis of Stable Diffusion)

- Key idea: train a separate *encoder* and *decoder* to convert images to and from a lower-dimensional latent space, run conditional diffusion model in latent space

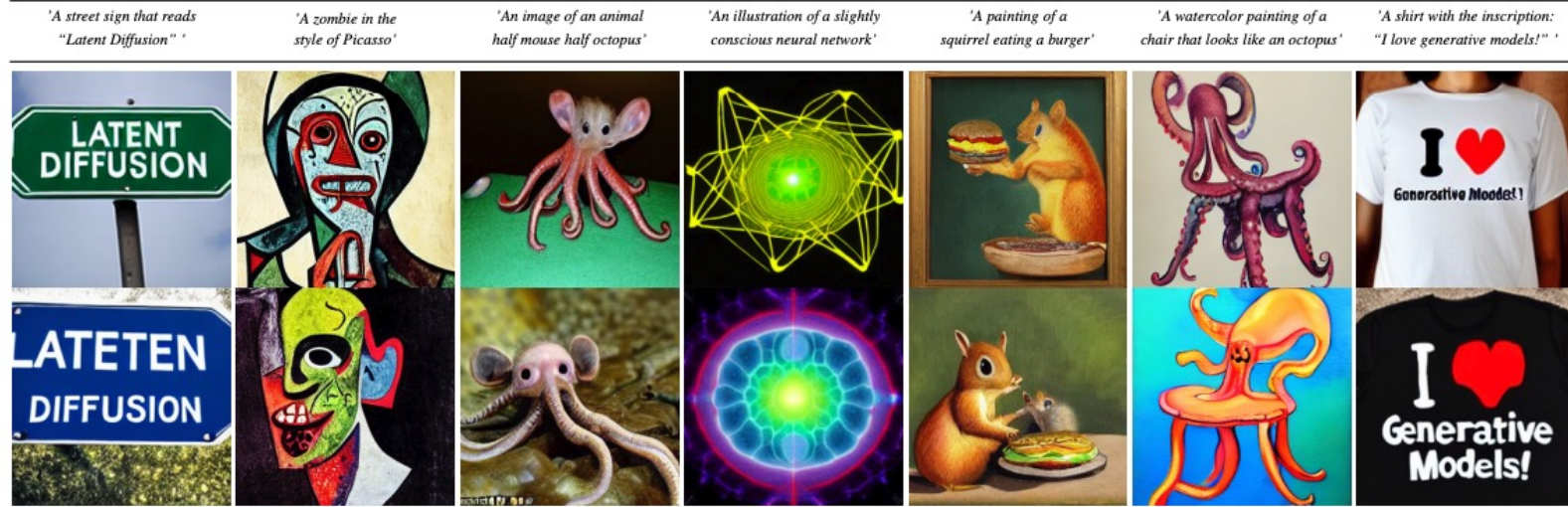


<https://medium.com/@steinsfu/stable-diffusion-clearly-explained-ed008044e07e>

Latent diffusion model (basis of Stable Diffusion)



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



Google Imagen (not public)



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.



A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.



A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.



Teddy bears swimming at the Olympics 400m Butterfly event.



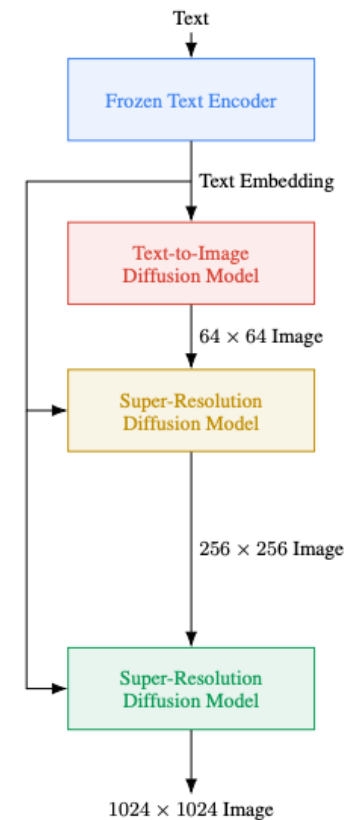
A cute corgi lives in a house made out of sushi.



A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

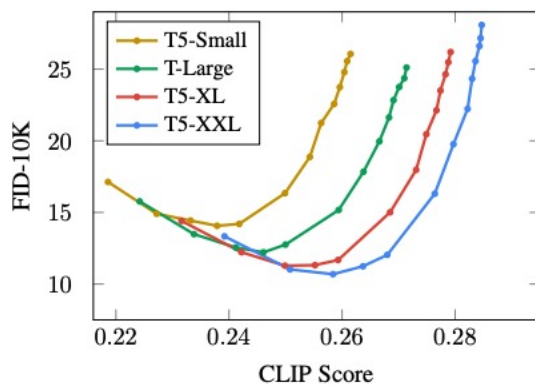
Google Imagen: Details

- Text encoder is a large language model (4.6B parameters) trained on text only
- Diffusion model to generate at 64x64, upsample to 256x256, then 1024x1024
 - Architecture: *efficient U-Net* (2B parameters): more parameters at lower resolutions, convolutions *after* downsampling and *before* upsampling
 - Classifier-free guidance with a *dynamic thresholding* technique, enabling good generation quality with high guidance weights
 - Training dataset: 460M image-text pairs (internally collected), 400M pairs from the [LAION dataset](#)

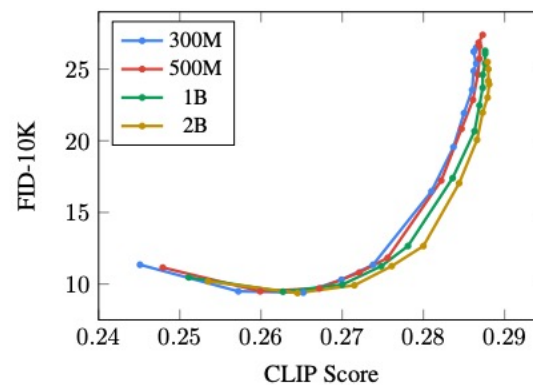


Google Imagen: Evaluation

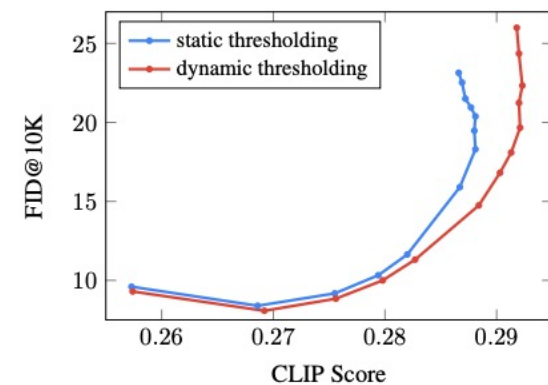
- Impact of model size, implementation choices



(a) Impact of encoder size.



(b) Impact of U-Net size.



(c) Impact of thresholding.

Curves are obtained by varying guidance weight

FID evaluated on COCO dataset by sampling prompts and generating images using the same prompts

Google Imagen: Evaluation

- Human evaluation on DrawBench (set of 200 prompts)

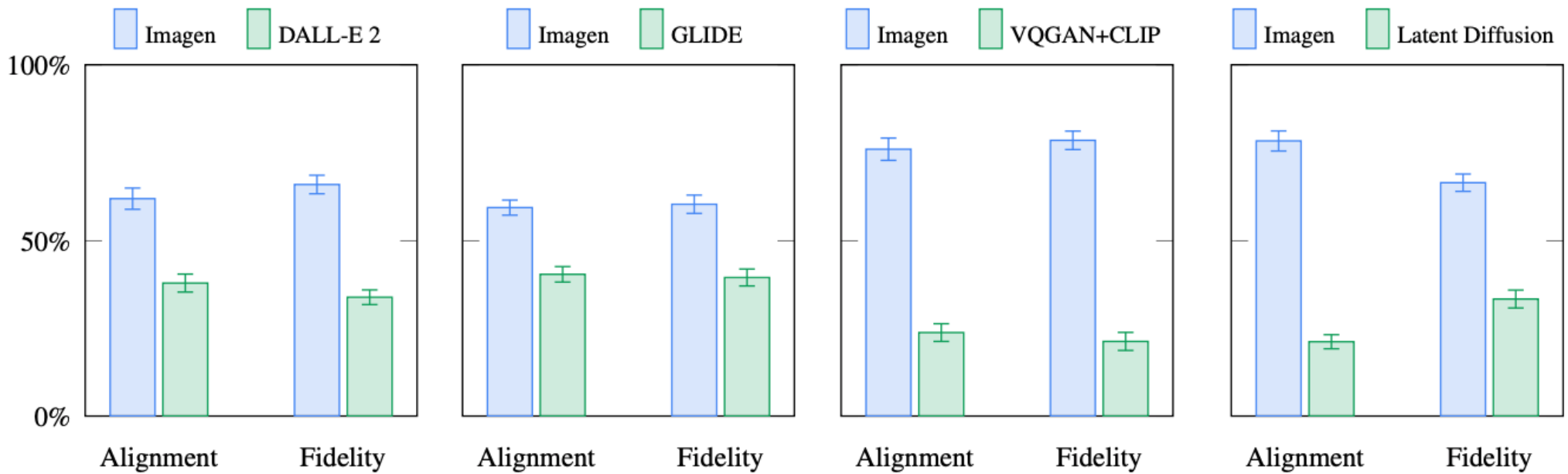
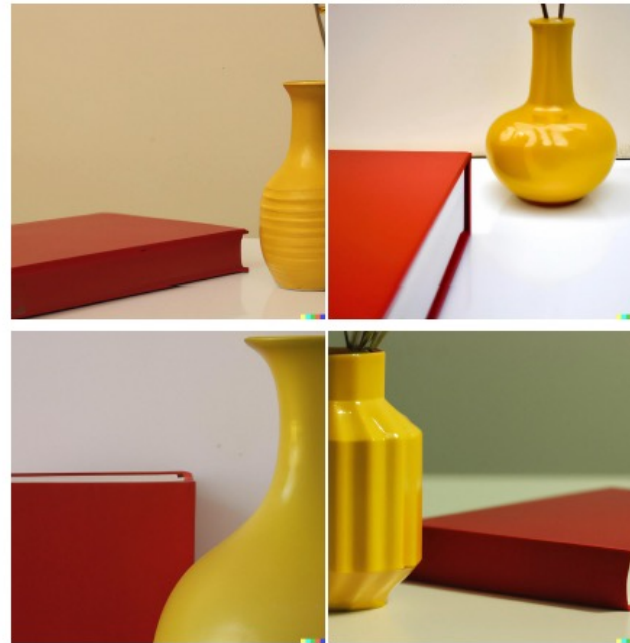


Imagen vs. DALL-E 2

Imagen (Ours)



DALL-E 2 [54]



“A yellow book and a red vase”

Imagen vs. DALL-E 2

Imagen (Ours)



DALL-E 2 [54]



“A black apple and a green backpack”

Imagen is better than DALL-E 2 in assigning the colors to the objects

Imagen vs. DALL-E 2

Imagen (Ours)



DALL-E 2 [54]



“A storefront with Text to Image written on it”

Imagen vs. DALL-E 2

Imagen (Ours)



DALL-E 2 [54]



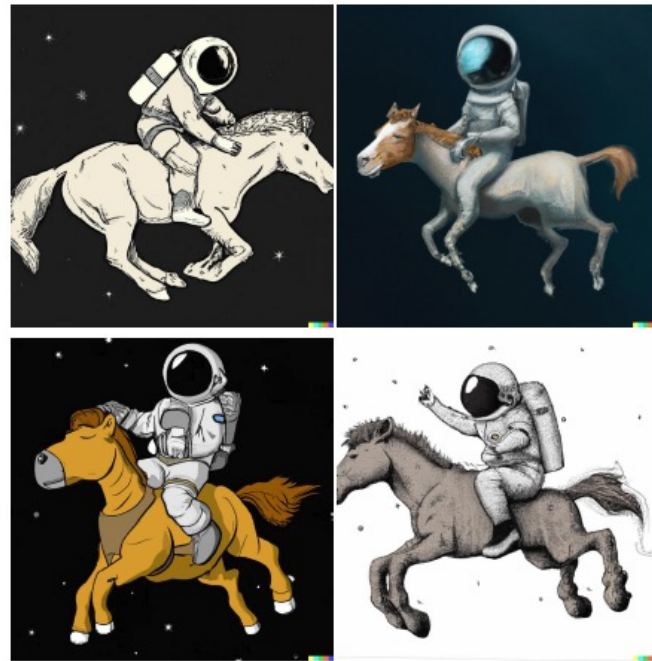
“A panda making latte art”

Imagen vs. DALL-E 2

Imagen (Ours)



DALL-E 2 [54]



“A horse riding an astronaut”

Outline

Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

Part 2: Recent Advances

- Denoising diffusion implicit models (DDIMs)

Denoising Diffusion Implicit Models (DDIMs)

- DDIM **roughly sketches** the final sample, then **refine** it with the reverse process
 - **Key idea:**
 - Given \mathbf{x}_t , generate the **rough sketch** \mathbf{x}_0 and **refine** $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)^1$
 - Unlike original diffusion model, it is not a Markovian structure

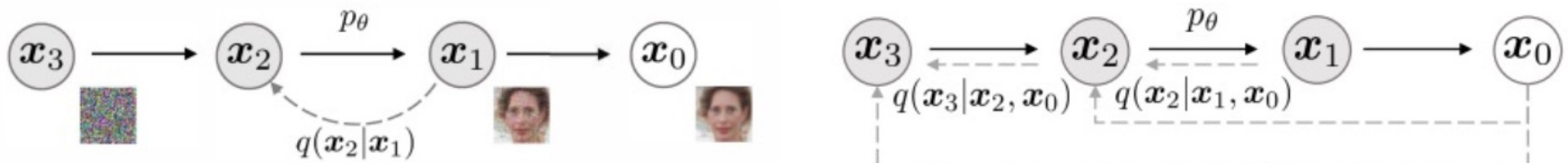


Figure 1: Graphical models for diffusion (left) and non-Markovian (right) inference models.

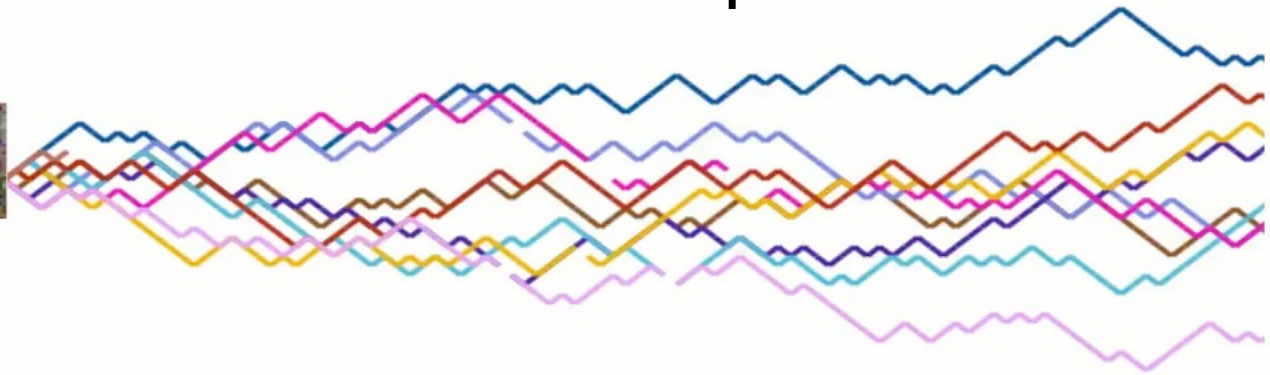
DDIM follows a deterministic process

DDPM follows stochastic process

DDPM



x_T



DDIM follows deterministic process

DDIM



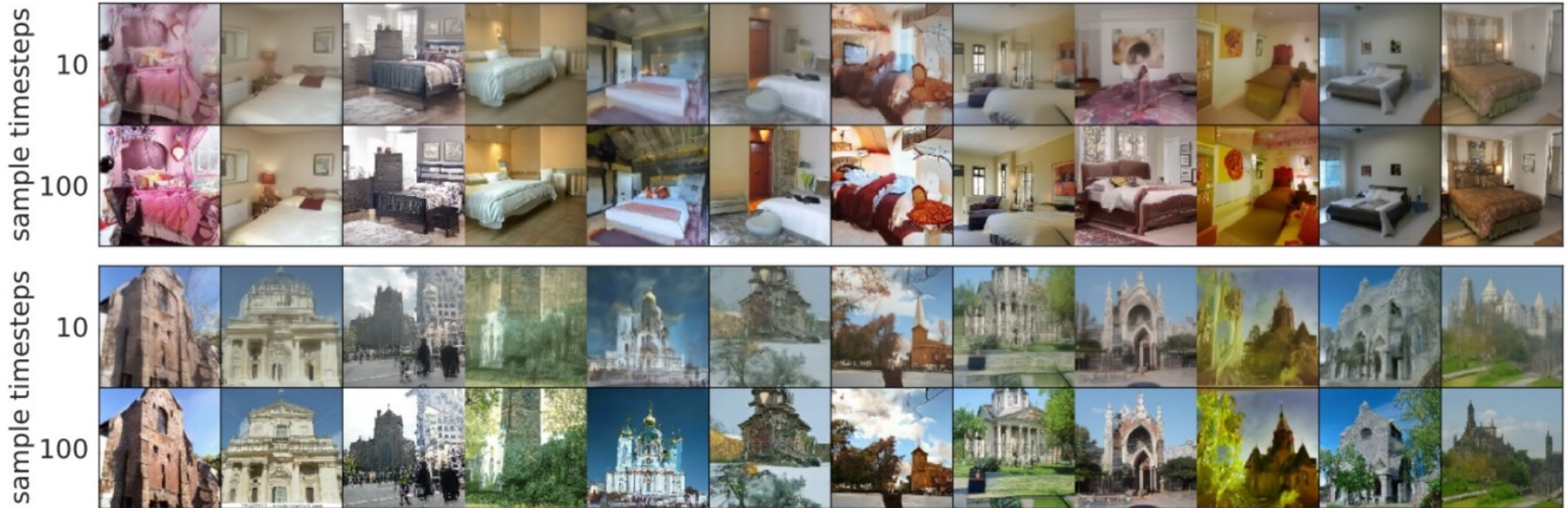
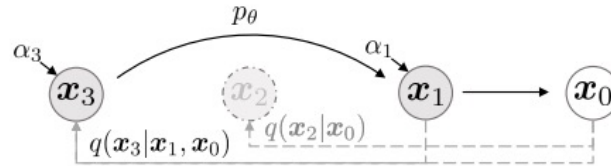
x_T



The same original noise x_T to same image x_0

DDIM reduces the sampling steps significantly

- Creates the outline of the sample after only 10 steps (DDPM needs hundreds)



Outline

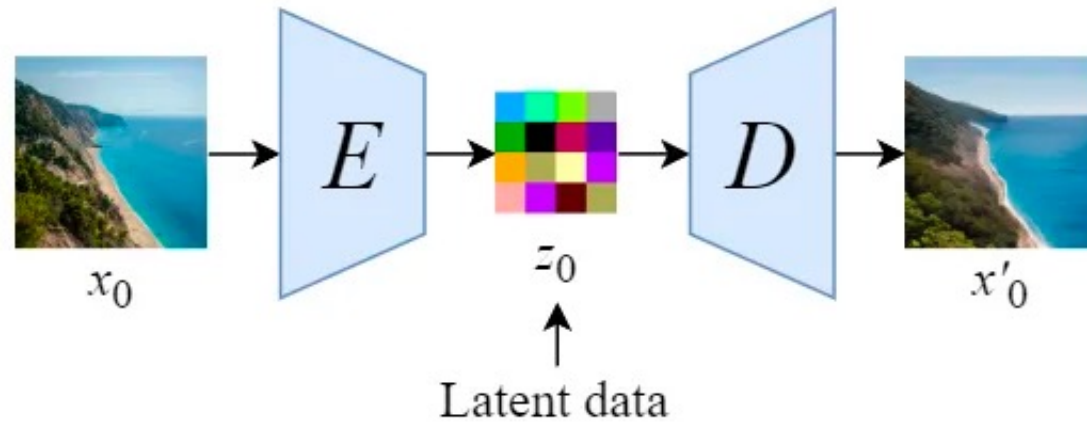
Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

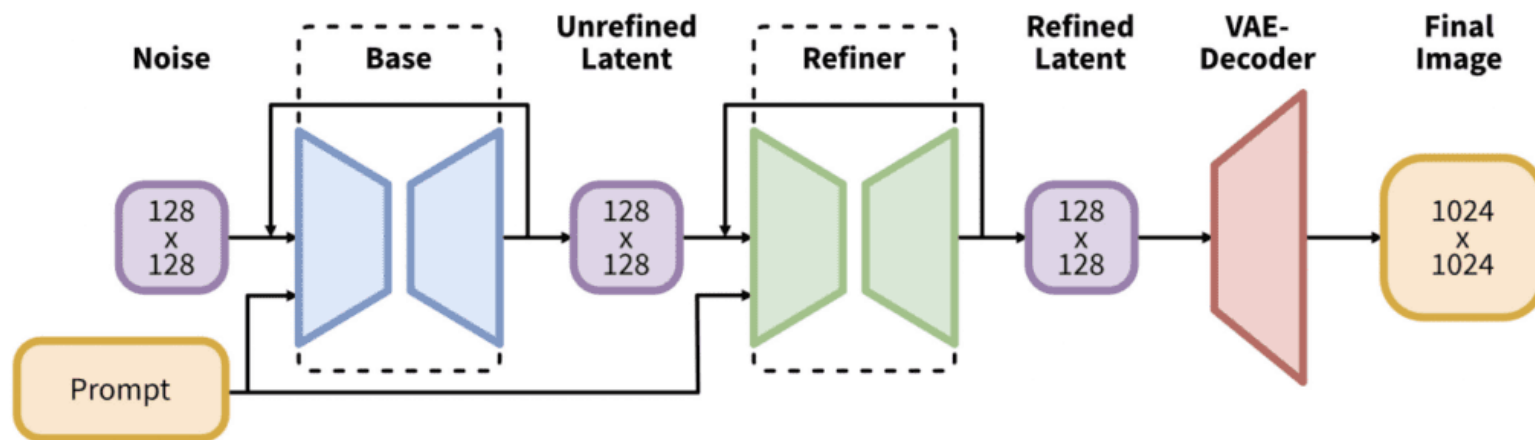
Part 2: Recent Advances

- Denoising diffusion implicit models (DDIMs)
- Stable Diffusion XL, Stable Diffusion 3

Recall Stable Diffusion (SD v1.4, SD v1.5, SD v2.1)



Improved SDXL pipeline

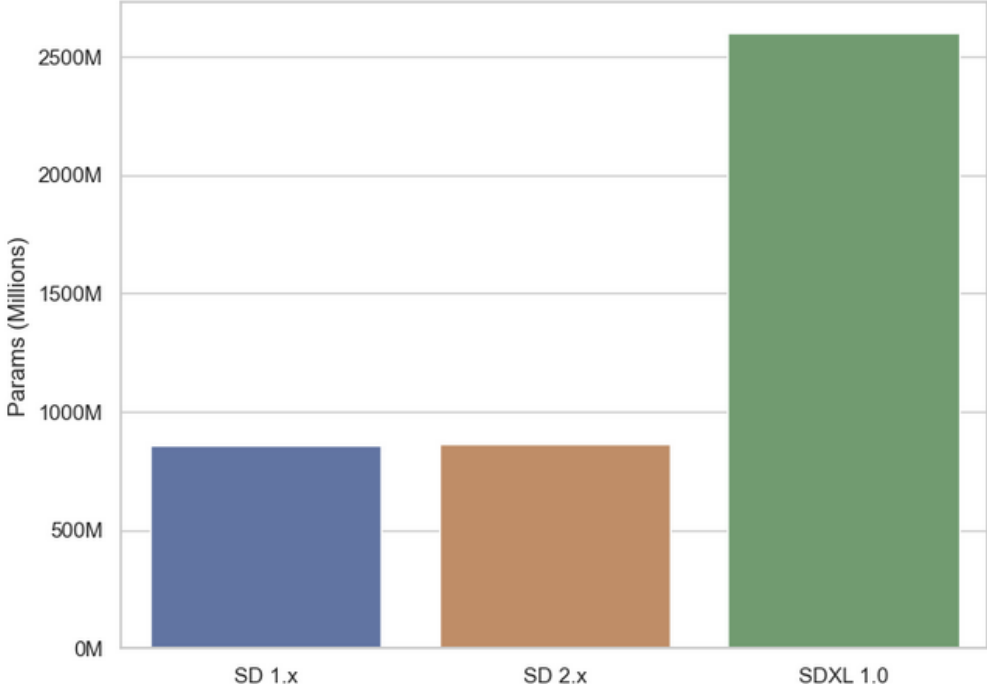


- Separate refiner model
- Two text-encoders
- Bigger U-net with more attention blocks and higher number of parameters

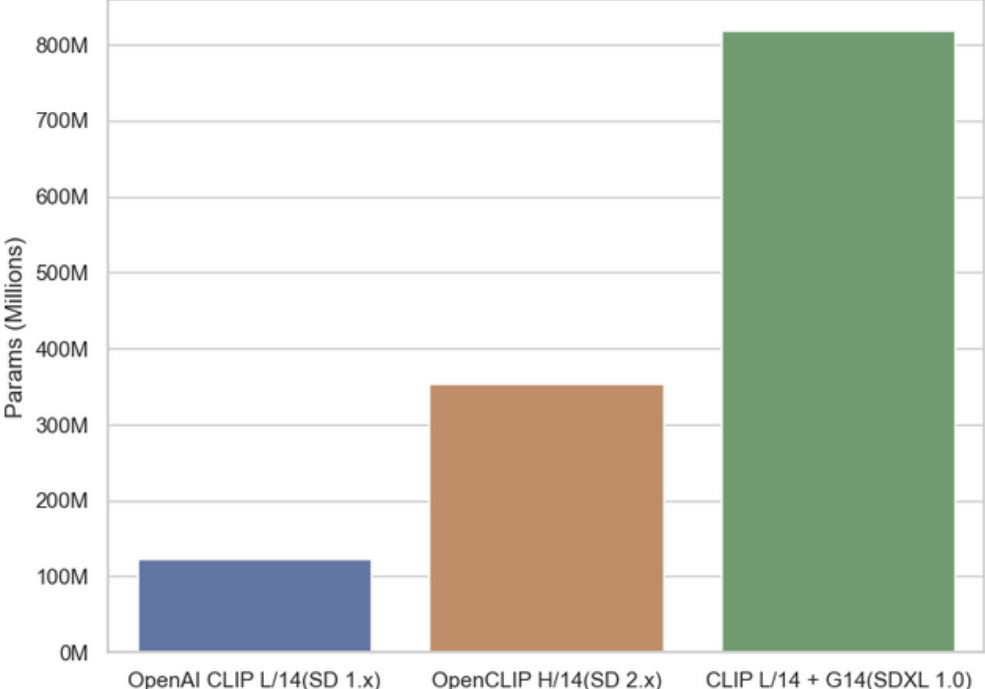
<https://towardsdatascience.com/the-arrival-of-sd-xl-1-0-4e739d5cc6c7>

SDXL

Number of UNeT Parameters

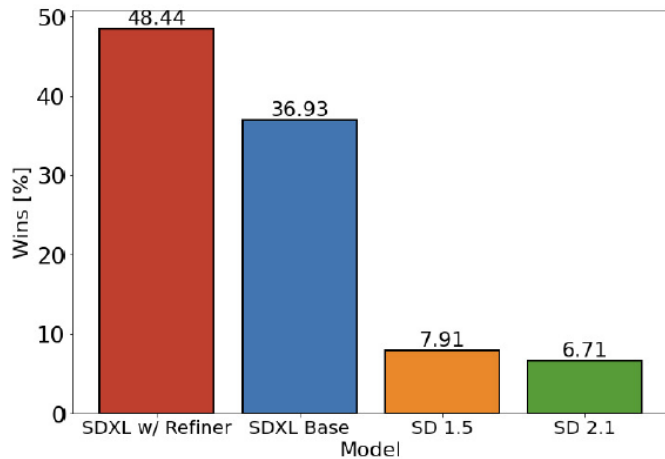


Number of Text Encoder Parameters



<https://towardsdatascience.com/the-arrival-of-sdxl-1-0-4e739d5cc6c7>

SDXL: Improvements in generation quality



Comparing user preferences between SDXL and previous models.

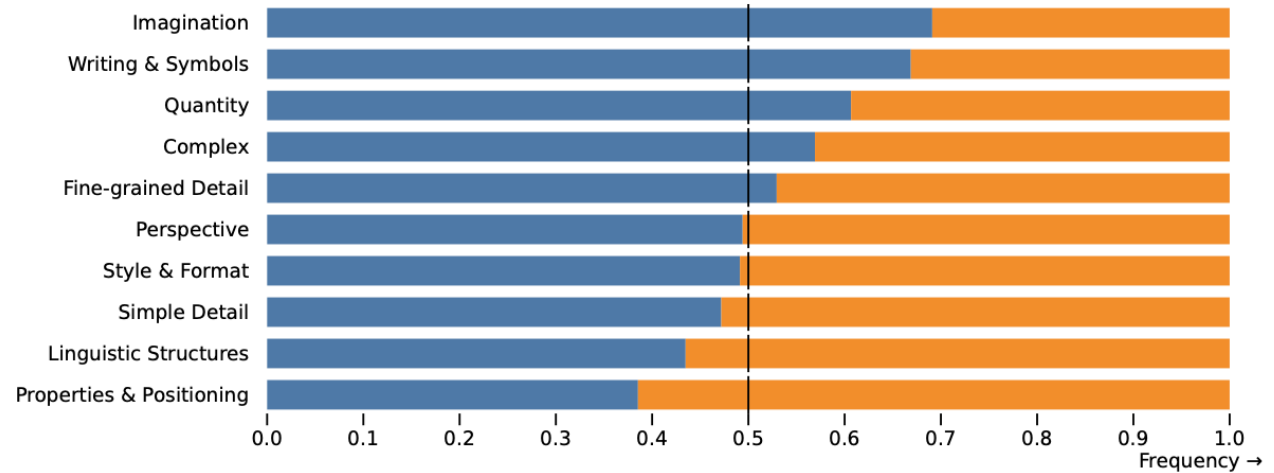


Figure 11: Preference comparisons of *SDXL* (with refinement model) to *Midjourney V5.1* on complex prompts. *SDXL* either outperforms or is statistically equal to *Midjourney V5.1* in 7 out of 10 categories.

[Improving Latent Diffusion Models for High-Resolution Image Synthesis](#)

SDXL: Results



Figure 4: Comparison of the output of *SDXL* with previous versions of *Stable Diffusion*. For each prompt, we show 3 random samples of the respective model for 50 steps of the DDIM sampler [46] and cfg-scale 8.0 [13]. Additional samples in Fig. 14.

SDXL: Results

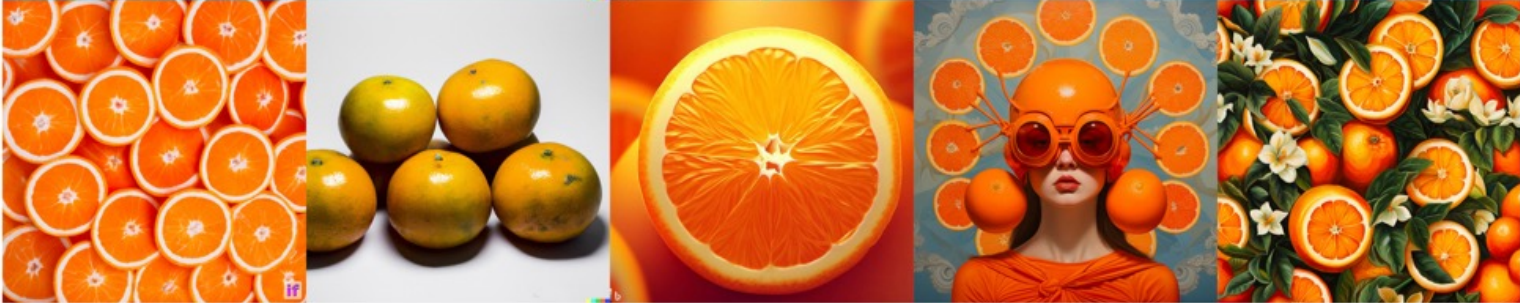
cat patting a crystal ball with the number 7 written on it in black marker



photograph of a red ball on a blue cube



orange



DEEPLYLOYD IF

DALLE-2

BING IMAGE CREATOR

MIDJOURNEY v5.2

SDXL v0.9

SD3



Prompt: A beautiful painting of flowing colors and styles forming the words "The SD3 research paper is here!", the background is speckled with drops and splashes of paint

<https://stability.ai/news/stable-diffusion-3-research-paper>

SD3: Results



Prompt: Translucent pig, inside is a smaller pig.



Prompt: A massive alien space ship that is shaped like a pretzel.



Prompt: A kangaroo holding a beer, wearing ski goggles and passionately singing silly songs.



Prompt: An entire universe inside a bottle sitting on the shelf at walmart on sale.



Prompt: A cheeseburger with juicy beef patties and melted cheese sits on top of a toilet that looks like a



Prompt: This dreamlike digital art captures a vibrant, kaleidoscopic bird in a lush rainforest



Prompt: A car made out of vegetables.



Prompt: Heat death of the universe line art

Outline

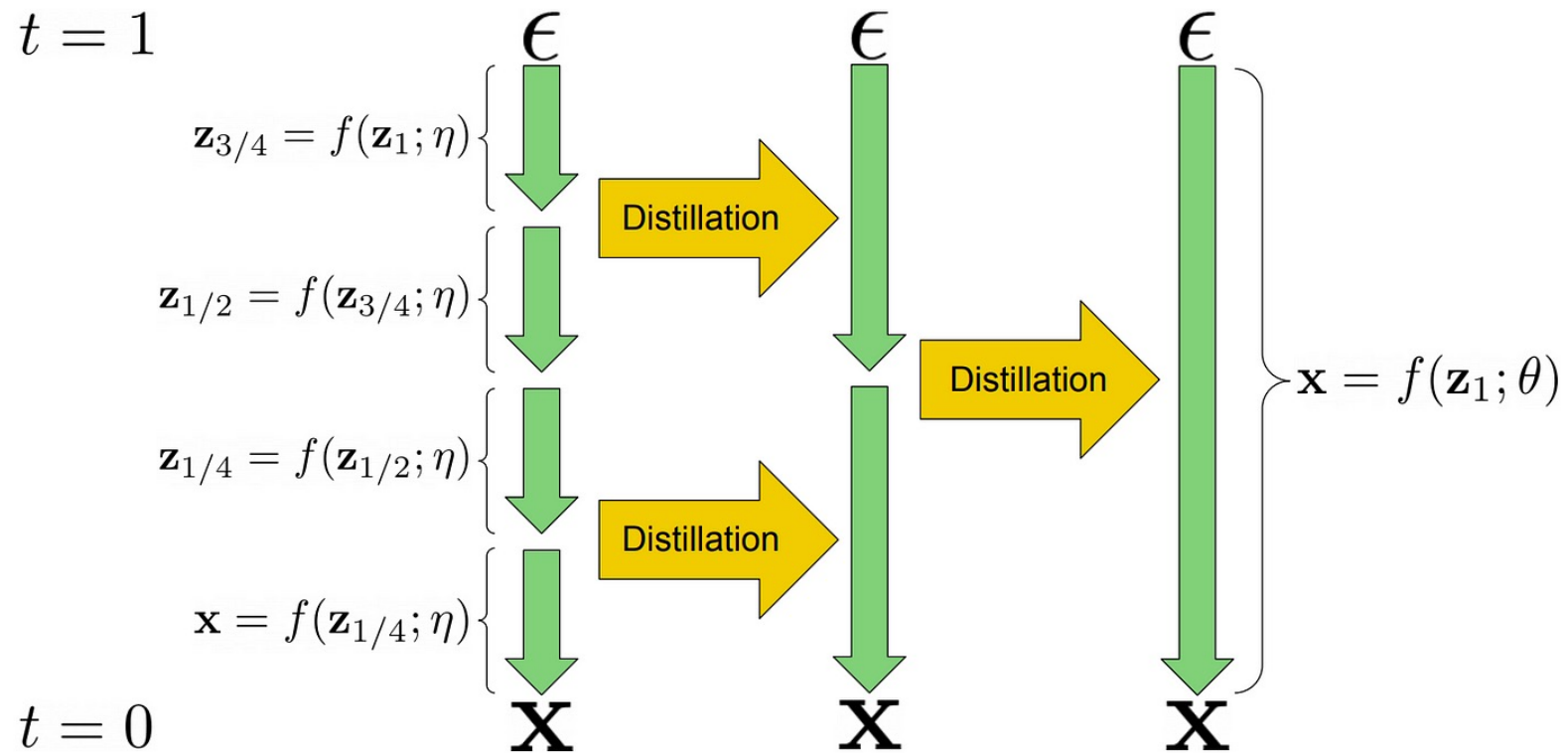
Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

Part 2: Recent Advances

- Denoising diffusion implicit models (DDIMs)
- Stable Diffusion XL, Stable Diffusion 3
- **Progressive Distillation**

Progressive Distillation



Progressive Distillation: Results

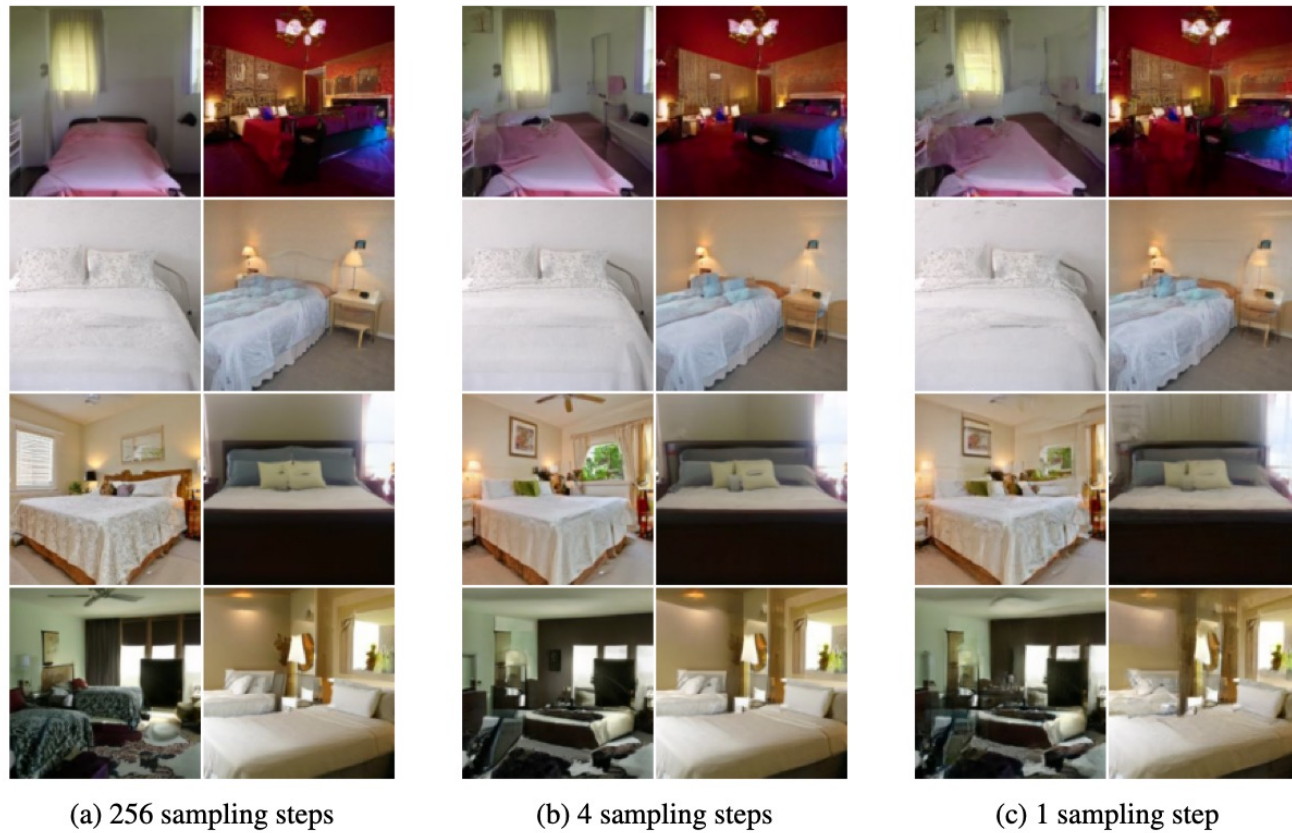


Figure 10: Random samples from our distilled LSUN bedrooms models, for fixed random seed and for varying number of sampling steps.

Outline

Part 1: Basics

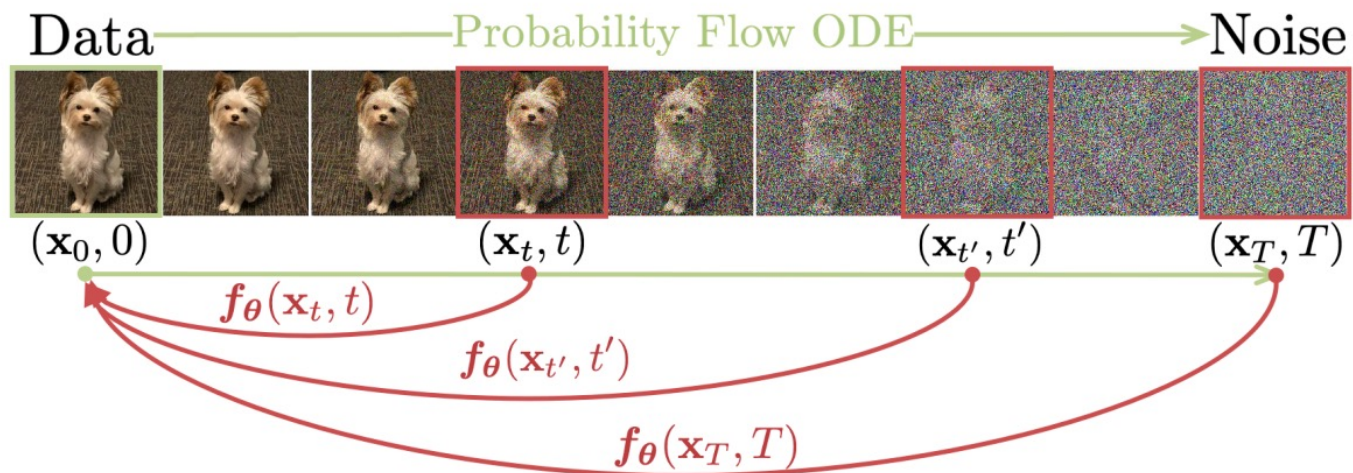
- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

Part 2: Recent Advances

- Denoising diffusion implicit models (DDIMs)
- Stable Diffusion XL, Stable Diffusion 3
- Model Distillation
- Latent Consistency Models (LCM)

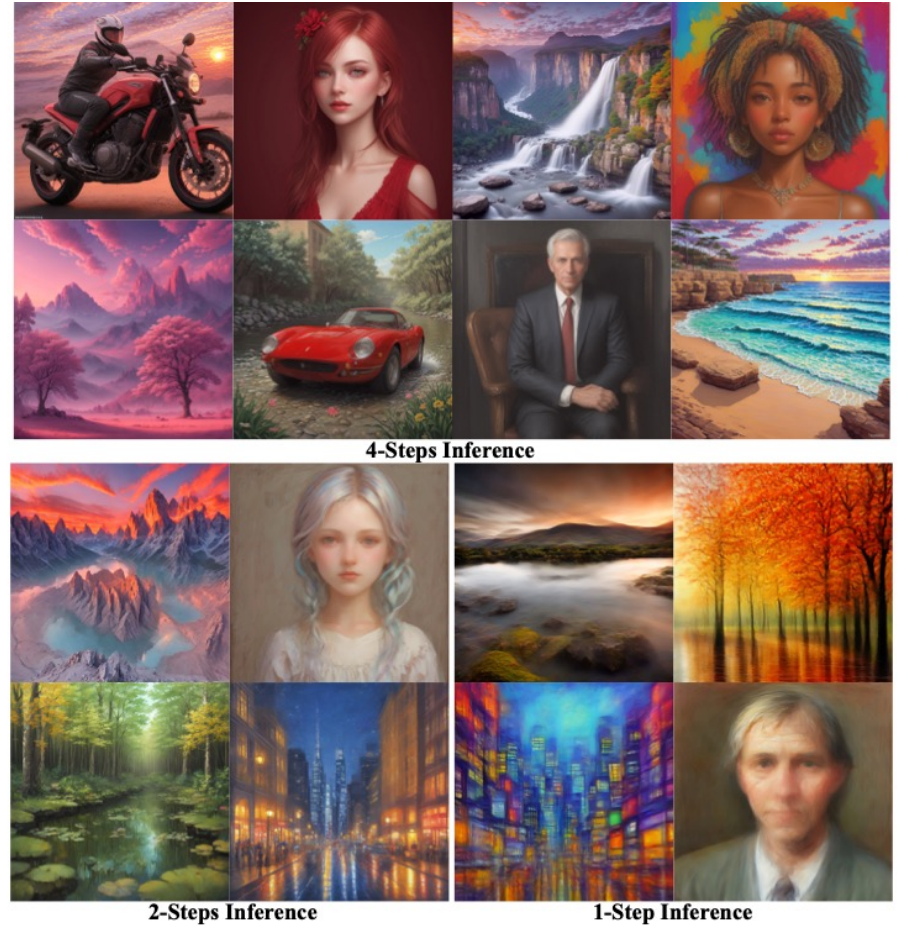
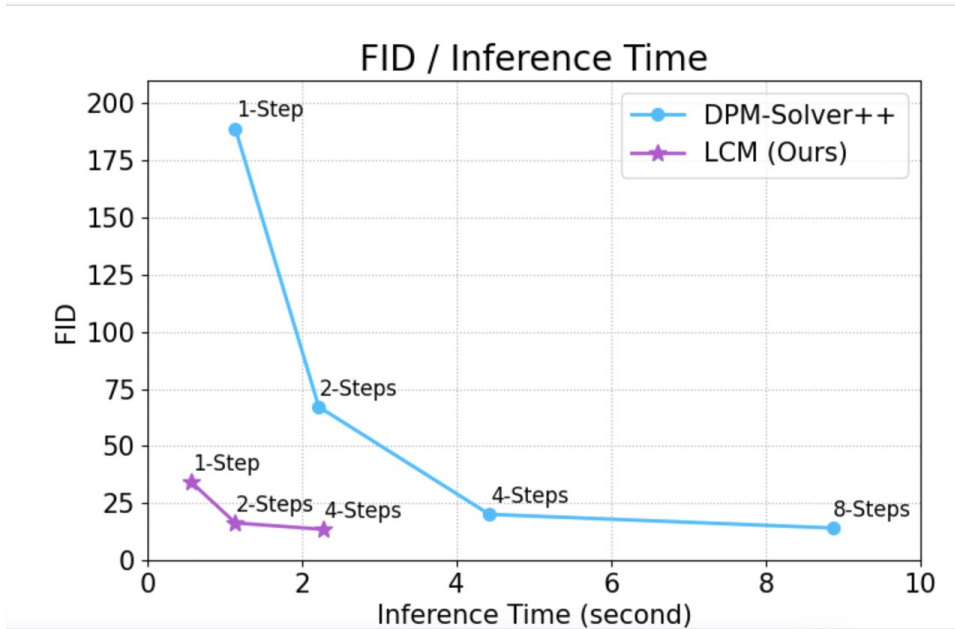
Latent Consistency Models

Consistency Models



Latent Consistency Models: combine the above idea with Latent Diffusion Models

Latent Consistency Models: Results



Outline

Part 1: Basics

- Denoising diffusion probabilistic models (DDPMs)
- Conditional diffusion models
- Large-scale models: DALL-E 2, Stable Diffusion, Imagen

Part 2: Recent Advances

- Denoising diffusion implicit models (DDIMs)
- Stable Diffusion XL, Stable Diffusion 3
- Progressive Distillation
- Latent Consistency Models (LCM)

(next class) Part 3: Implementation, Applications, and Ethical issues