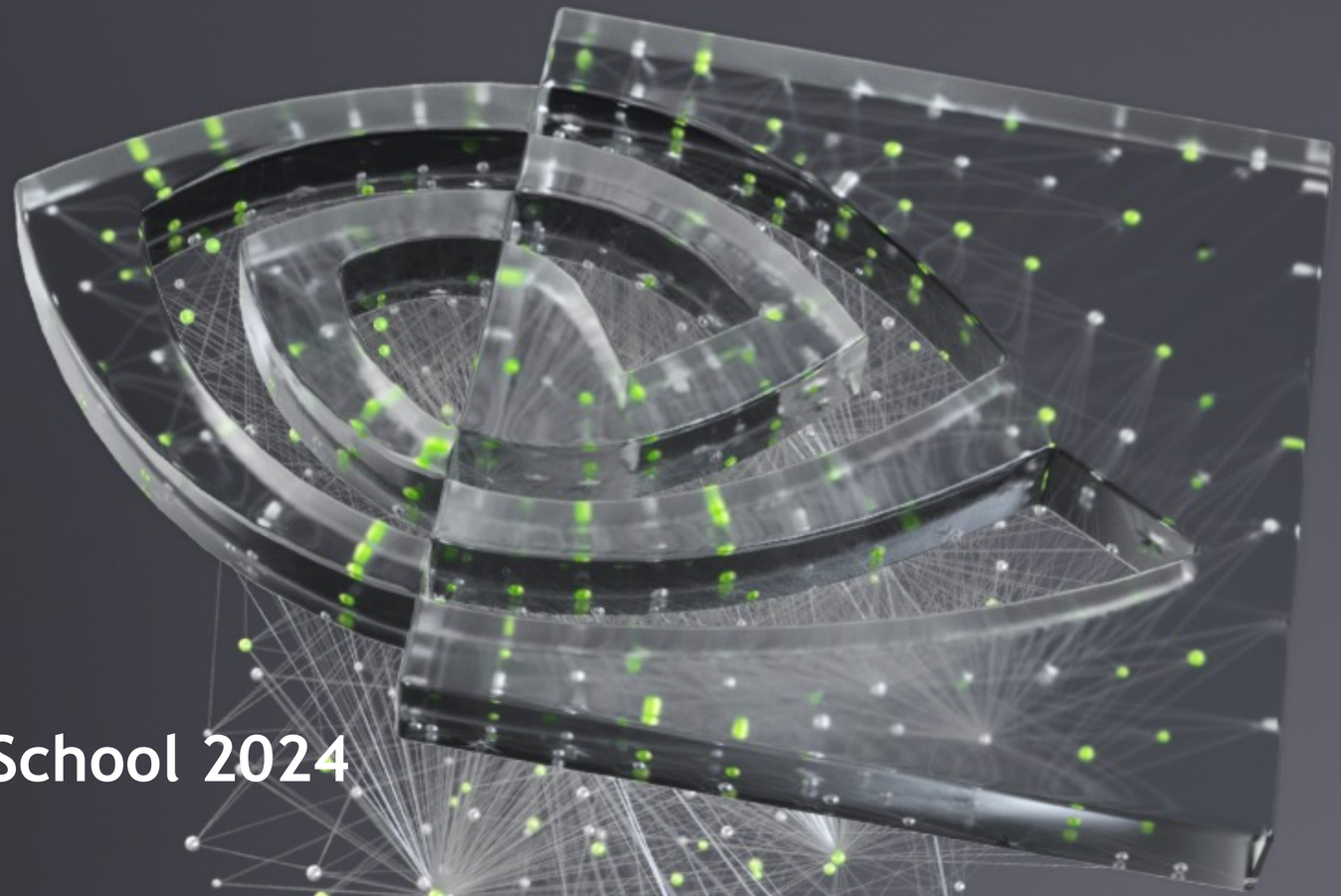


Oxford Machine Learning Summer School 2024
MLx Fundamentals

Generative AI (Vision)

Karsten Kreis

Senior Research Scientist, NVIDIA

@karsten_kreis

https://karstenkreis.github.io/

Teaching Assistant

Seung Wook Kim

- Senior Research Scientist at NVIDIA
- Expert in Generative AI for image, video, 3D/4D synthesis
- Available on Slack during lecture. Do not hesitate to ask questions!

Website: https://seung-kim.github.io/seungkim/
Twitter: @seungkim0123

- “Align Your Gaussians: Text-to-4D with Dynamic 3D Gaussians and Composed Diffusion Models”, CVPR, 2024
- “EmerDiff: Emerging Pixel-level Semantic Knowledge in Diffusion Models”, ICLR, 2024
- “WildFusion: Learning 3D-Aware Latent Diffusion Models in View Space”, ICLR, 2024
- “NeuralField-LDM: Scene Generation with Hierarchical Latent Diffusion Models”, CVPR, 2023
- “Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, CVPR, 2023
- “PolymorphicGAN: Generating Aligned Samples Across Multiple Domains With Learned Morph Maps”, CVPR, 2022
- …

https://stability.ai/news/stable-diffusion-sdxl-1-announcement

Dai et al., “Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack”, *arXiv*, 2023

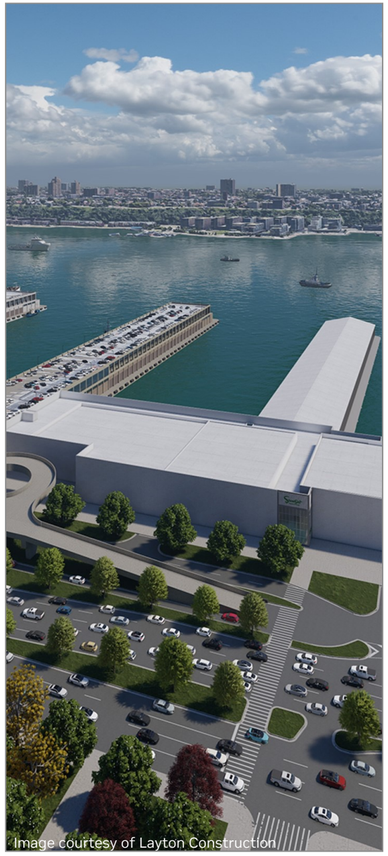


Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR* 2023
Emu Video, https://emu-video.metademolab.com/

”A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.”

Brooks, Peebles, et al. https://openai.com/sora

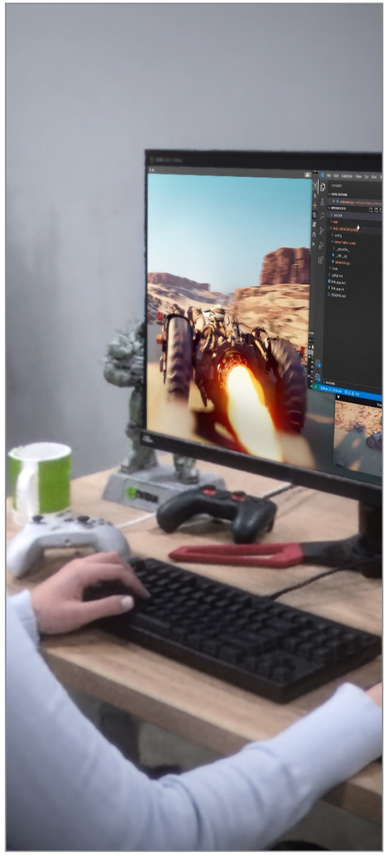
Generative AI Applications



Architecture / Design

Feature Learning

Film / Video



3D FX / Game Dev

Marketing

Photography

Slide: Courtesy of Ashlee Martino-Tarr and Daniela Flamm Jackson

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*

Overview

1. History: From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. Video Diffusion Models
4. 3D and 4D Generation: *From 2D to 3D & 4D with Score Distillation*

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*

Image Generation Timeline

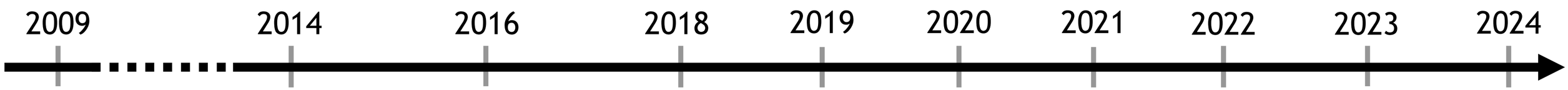


Image Generation Timeline

2009 2014 2016 2018 2019 2020 2021 2022 2023 2024

Boltzmann Machines

Deep Boltzmann Machines

Ruslan Salakhutdinov
Department of Computer Science
University of Toronto
rsalakhu@cs.toronto.edu

Geoffrey Hinton
Department of Computer Science
University of Toronto
hinton@cs.toronto.edu

Abstract

We present a new learning algorithm for Boltzmann machines that contain many layers of hidden variables. Data-dependent expectations are estimated using a variational approximation that tends to focus on a single mode, and data-independent expectations are approximated using persistent Markov chains. The use of two quite different techniques for estimating the two types of expectation that enter into the gradient of the log-likelihood makes it practical to learn Boltzmann machines with multiple hidden layers and millions of parameters. The learning can be made more efficient by using a layer-by-layer "pre-training" phase that allows variational inference to be initialized with a single bottom-up pass. We present results on the MNIST and NORB datasets showing that deep Boltzmann machines learn good generative models and perform well on handwritten digit and visual object recognition tasks.

1 Introduction

The original learning algorithm for Boltzmann machines (Hinton and Sejnowski, 1983) required randomly initialized Markov chains to approach their equilibrium distributions in order to estimate the data-dependent and data-independent expectations that a connected pair of binary variables would both be on. The difference of these two expectations is the gradient required for maximum likelihood learning. Even with the help of simulated annealing, this learning procedure was too slow to be practical. Learning can be made much more efficient in a restricted Boltzmann machine (RBM), which has no connections between hidden units (Hinton, 2002). Multiple hidden layers can be learned by treating the hidden activities of one RBM as the data for training a higher-level RBM (Hinton et al., 2006; Hinton and Salakhutdinov, 2006). However, if multiple layers are learned in this greedy, layer-by-layer way, the resulting composite model is *not* a multilayer Boltzmann machine (Hinton et al., 2006). It is a hybrid generative model called a "deep belief net" that has undirected connections between its top two layers and downward directed connections between all its lower layers.

In this paper we present a much more efficient learning procedure for fully general Boltzmann machines. We also show that if the connections between hidden units are restricted in such a way that the hidden units form multiple layers, it is possible to use a stack of slightly modified RBM's to initialize the weights of a deep Boltzmann machine before applying our new learning procedure.

2 Boltzmann Machines (BM's)

A Boltzmann machine is a network of symmetrically coupled stochastic binary units. It contains a set of visible units $\mathbf{v} \in \{0,1\}^D$, and a set of hidden units $\mathbf{h} \in \{0,1\}^P$ (see Fig. 1). The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ is defined as:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\frac{1}{2}\mathbf{v}^\top \mathbf{L}\mathbf{v} - \frac{1}{2}\mathbf{h}^\top \mathbf{J}\mathbf{h} - \mathbf{v}^\top \mathbf{W}\mathbf{h}, \quad (1)$$

where $\theta = \{\mathbf{W}, \mathbf{L}, \mathbf{J}\}$ are the model parameters[1]: $\mathbf{W}, \mathbf{L}, \mathbf{J}$ represent visible-to-hidden, visible-to-visible, and hidden-to-hidden symmetric interaction terms. The diagonal elements of $\mathbf{L}$ and $\mathbf{J}$ are set to 0. The probability that the model assigns to a visible vector $\mathbf{v}$ is:

$$p(\mathbf{v}; \theta) = \frac{p^*(\mathbf{v}; \theta)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right),$$

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right)$$

where p^* denotes unnormalized probability, and Z is the partition function. The *conditional* distribu

[1]We have omitted the bias terms for clarity of presen

Appearing in Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

Department of Computer Science
University of Toronto
http://learning.cs.toronto.edu

6 King's College Rd, Toronto
M5S 3G4, Canada
fax: +1 416 978 1455

June 26, 2008

UTML TR 2008–002

Learning and Evaluating Boltzmann Machines

Ruslan Salakhutdinov
Department of Computer Science, University of Toronto

Abstract

We provide a brief overview of the variational framework for obtaining deterministic approximations or upper bounds for the log-partition function. We also review some of the Monte Carlo based methods for estimating partition functions of arbitrary Markov Random Fields. We then develop an annealed importance sampling procedure for estimating partition functions of restricted Boltzmann machines (SRBM's), and Boltzmann machines (BM's). Our empirical results indicate that the AIS procedure provides much better estimates of the partition function than some of the popular variational-based methods. Finally, we develop a new learning algorithm for training general Boltzmann machines and show that it can be successfully applied to learning good generative models.

Image Generation Timeline

2009 | 2014 | 2016 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024

Boltzmann Machines

VAEs, GANs

Kingma and Welling et al., “Auto-Encoding Variational Bayes”, ICLR, 2014
Rezende et al., “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”, ICML, 2014
Goodfellow et al., “Generative Adversarial Nets”, NeurIPS, 2014

Image Generation Timeline

2009 2014 **2016** 2018 2019 2020 2021 2022 2023 2024

Boltzmann Machines

VAEs, GANs

PixelRNN, PixelCNN

van den Oord et al., “Pixel Recurrent Neural Networks”, ICML, 2016
van den Oord et al., “Conditional Image Generation with PixelCNN Decoders”, NeurIPS, 2016

Image Generation Timeline

2009 — Boltzmann Machines

2014 — VAEs, GANs

2016 — PixelRNN, PixelCNN

2018 — Glow

2019 2020 2021 2022 2023 2024

Kingma and Dhariwal, “Glow: Generative Flow with Invertible 1x1 Convolutions”, NeurIPS, 2018

Image Generation Timeline

2009 — Boltzmann Machines

2014 — VAEs, GANs

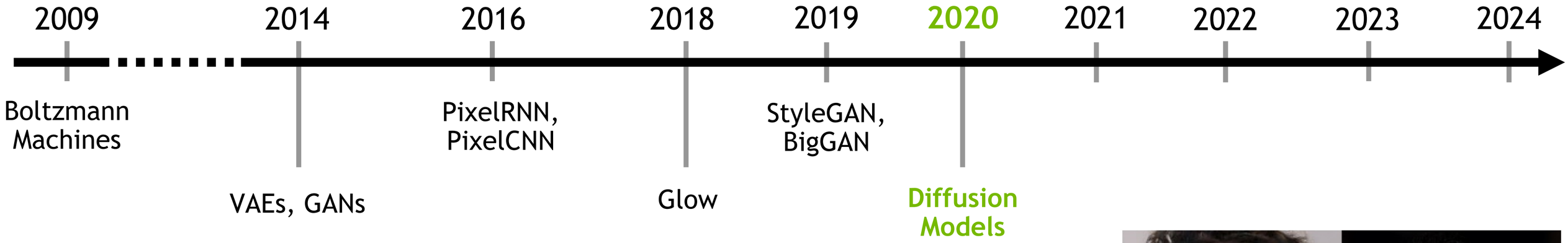
2016 — PixelRNN, PixelCNN

2018 — Glow

2019 — StyleGAN, BigGAN

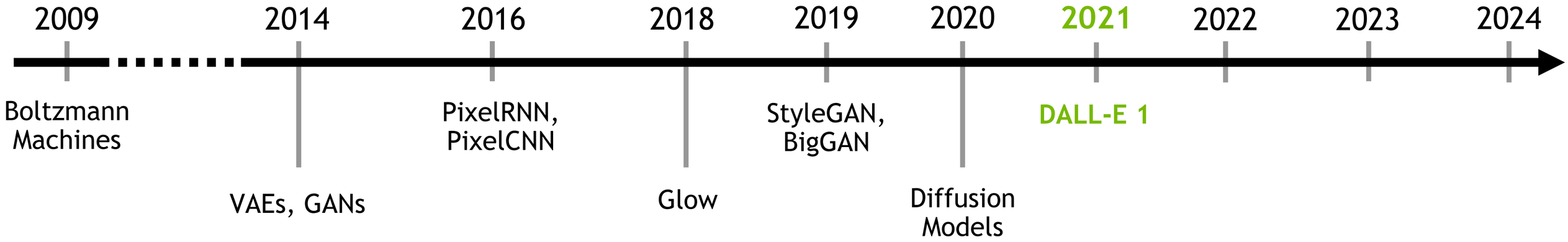
Brock et al., “Large Scale GAN Training for High Fidelity Natural Image Synthesis”, ICLR, 2019
Karras et al., “A Style-Based Generator Architecture for Generative Adversarial Networks”, CVPR, 2019

Image Generation Timeline



Ho et al., “Denoising Diffusion Probabilistic Models”, NeurIPS, 2020

Image Generation Timeline



an armchair in the shape of an avocado. . . .

Ramesh et al., “Zero-Shot Text-to-Image Generation”, ICML, 2021

(a) a tapir made of accordion. a tapir with the texture of an accordion.

(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

Image Generation Timeline

2009 — Boltzmann Machines

2014 — VAEs, GANs

2016 — PixelRNN, PixelCNN

2018 — Glow

2019 — StyleGAN, BigGAN

2020 — Diffusion Models

2021 — DALL-E 1

2022 — Stable Diffusion, DALL-E 2, Imagen, Midjourney

2023

2024

Ramesh et al., “Hierarchical Text-Conditional Image Generation with CLIP Latents”, 2022
Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, NeurIPS, 2022
Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR, 2022

Image Generation Timeline

2009 — Boltzmann Machines

2014 — VAEs, GANs

2016 — PixelRNN, PixelCNN

2018 — Glow

2019 — StyleGAN, BigGAN

2020 — Diffusion Models

2021 — DALL-E 1

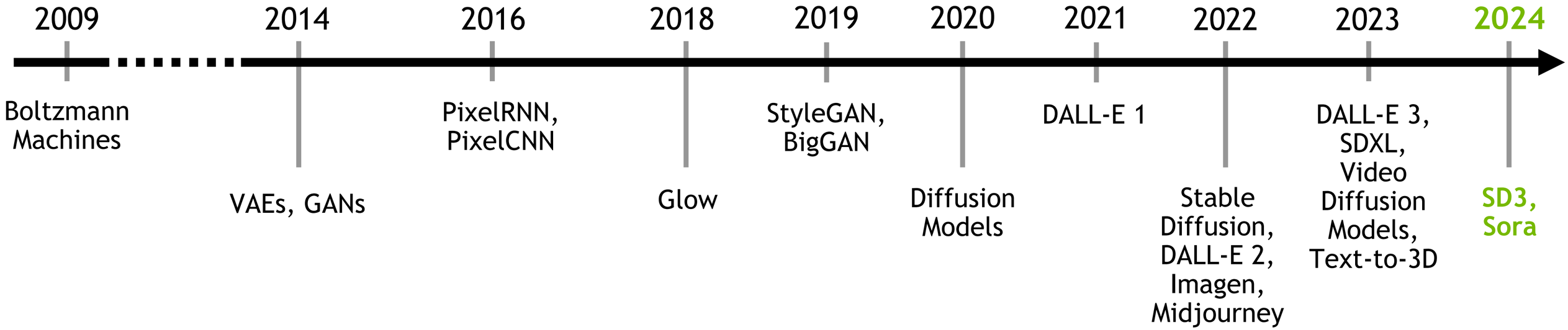
2022 — Stable Diffusion, DALL-E 2, Imagen, Midjourney

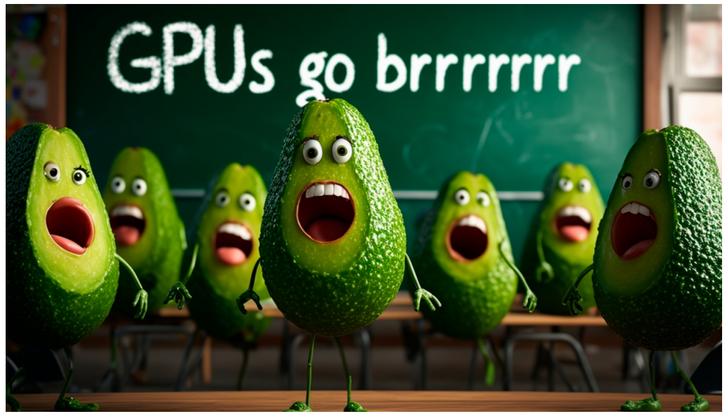
2023 — DALL-E 3, SDXL, Video Diffusion Models, Text-to-3D

2024

Betker et al., “Improving Image Generation with Better Captions”, 2023
Lin et al., “Magic3D: High-Resolution Text-to-3D Content Creation”, CVPR, 2023
Podell et al., “SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis”, 2023
Blattmann et al., “Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, CVPR, 2023

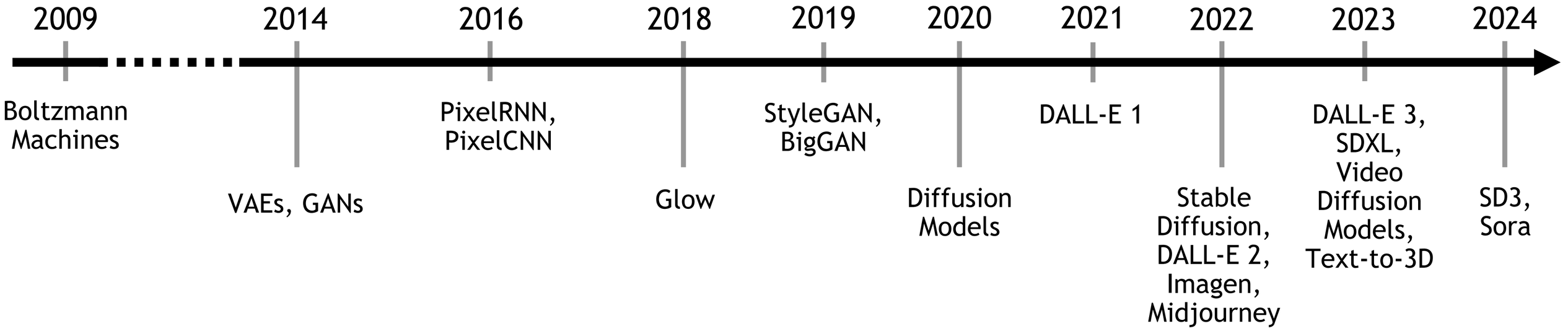
Image Generation Timeline

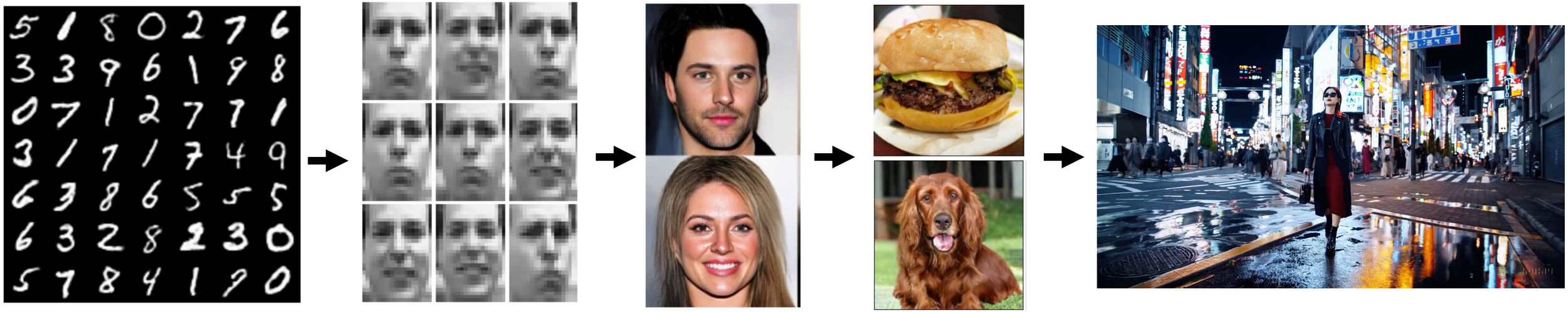




Esser et al. ”Scaling Rectified Flow Transformers for High-Resolution Image Synthesis”, 2024
Brooks, Peebles, et al., https://openai.com/sora, 2024

Image Generation Timeline





Overview

1. History: From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. Video Diffusion Models
4. 3D and 4D Generation: *From 2D to 3D & 4D with Score Distillation*

Tutorials on Diffusion Models

CVPR 2022: *Denoising Diffusion-based Generative Modeling: Foundations and Applications*

Website (~4 hours long, over 100,000 views on Youtube):
https://cvpr2022-tutorial-diffusion-models.github.io/


Karsten Kreis
NVIDIA


Ruiqi Gao
Google Brain


Arash Vahdat
NVIDIA

CVPR 2023: *Denoising Diffusion Models: A Generative Learning Big Bang*

Website:
https://cvpr2023-tutorial-diffusion-models.github.io/


Jiaming Song
NVIDIA


Chenlin Meng
Stanford


Arash Vahdat
NVIDIA

NeurIPS 2023: *Latent Diffusion Models: Is the Generative AI Revolution Happening in Latent Space?*

Website:
https://neurips2023-ldm-tutorial.github.io/


Karsten Kreis
NVIDIA


Ruiqi Gao
Google Brain

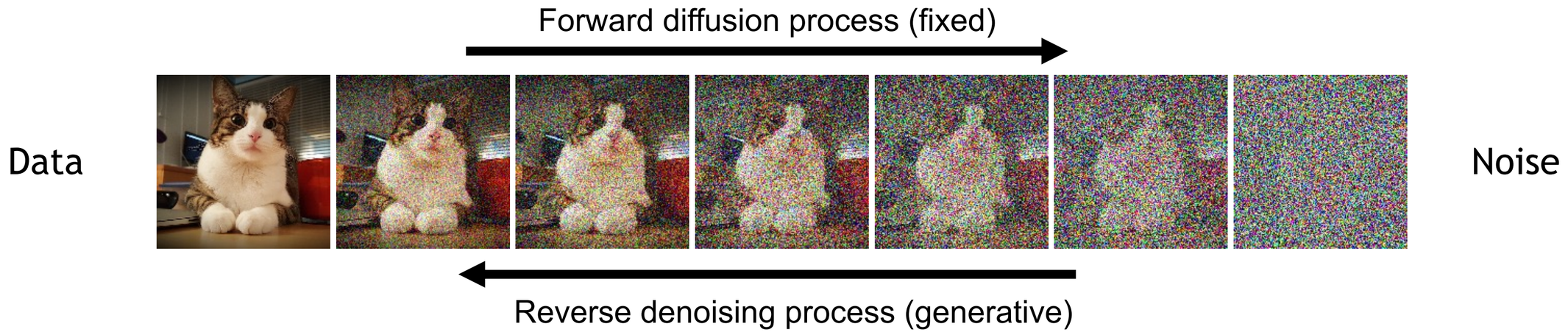

Arash Vahdat
NVIDIA

Diffusion Models

Learning to Generate by Denoising

Diffusion models consist of two processes:

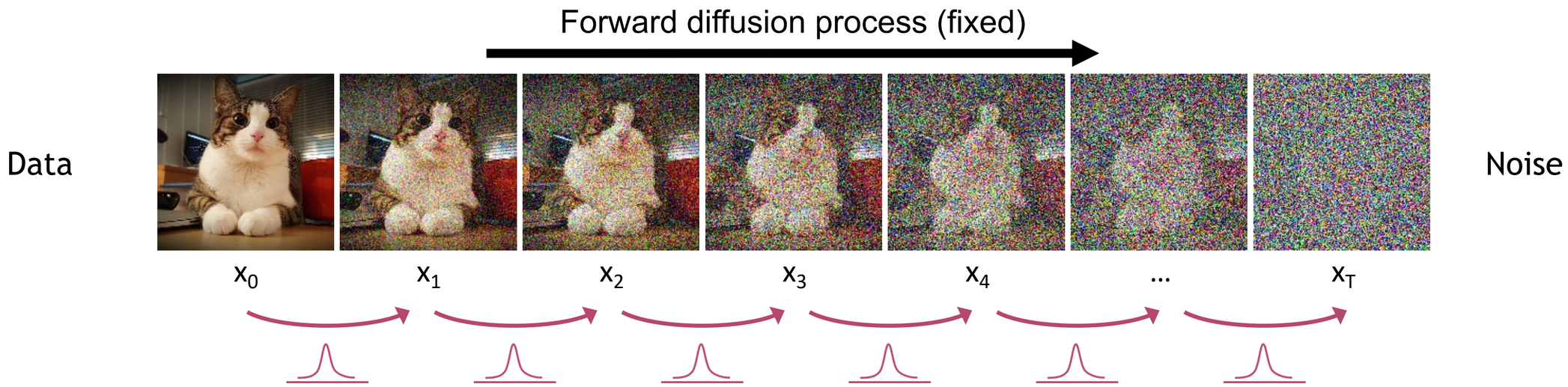
- (Fixed) forward diffusion process that gradually adds noise to input
- (Learnt) reverse denoising process that learns to generate data by denoising



Sohl-Dickstein et al., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”, ICML 2015
Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS* 2020

Diffusion Models

The Fixed Forward Diffusion Process

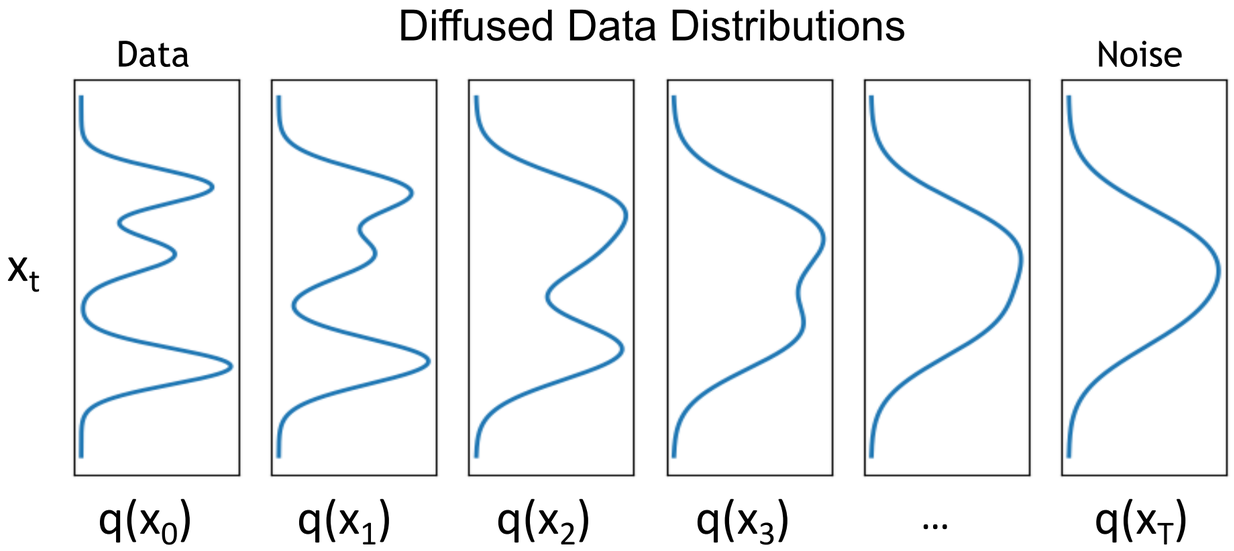


$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

β_t (the noise schedule) such that $q(\mathbf{x}_T|\mathbf{x}_0) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$

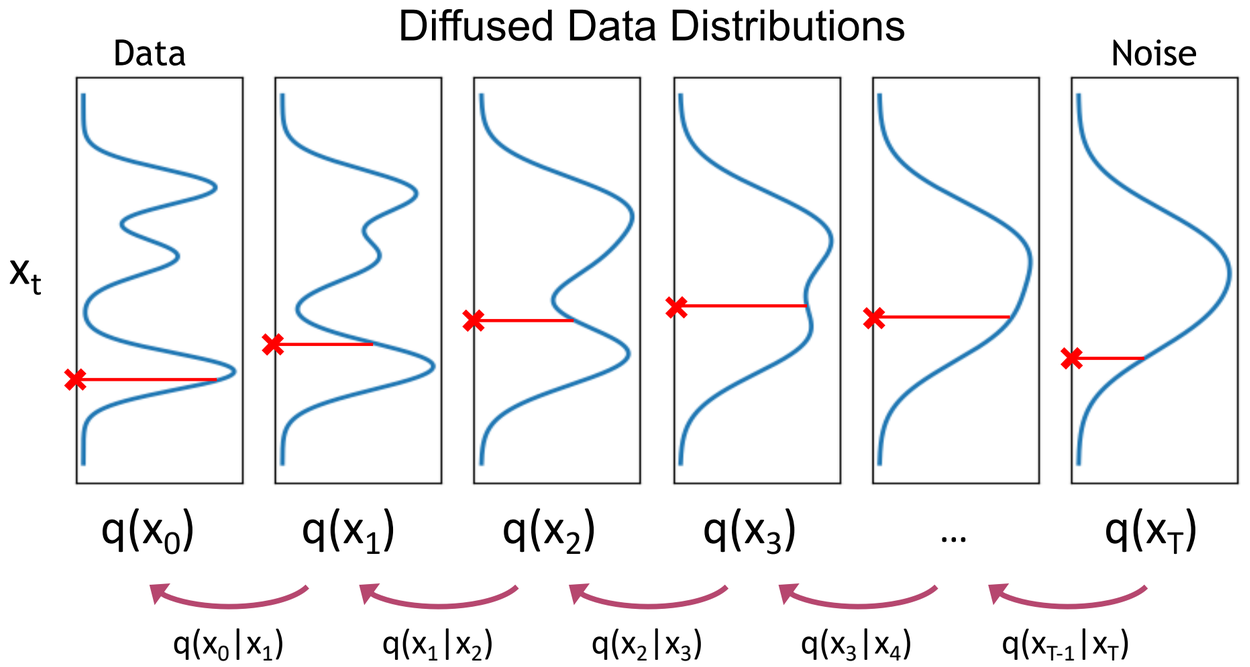
Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS* 2020

What happens to a Distribution in the Forward Diffusion?



The diffusion kernel is Gaussian convolution. We can sample $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_0)$ directly.

Generative Learning by Denoising



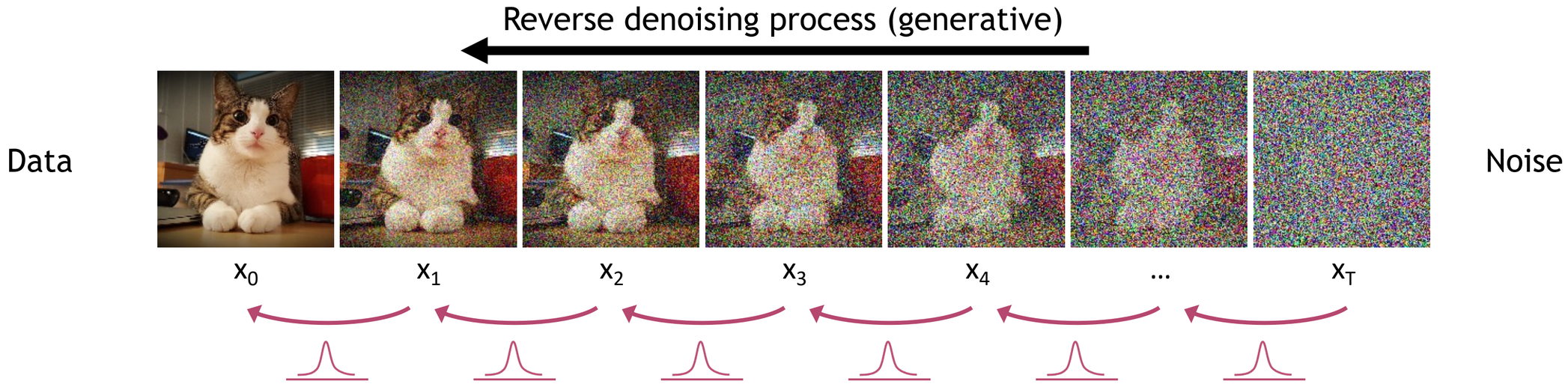
Generation:

- Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T, \mathbf{0}, \mathbf{I})$
- Iteratively sample $\underbrace{\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1}|\mathbf{x}_t)}_{\text{True Denoising Dist.}}$ ➡ **Intractable!** ➡ Approx. Gaussian for small steps. (β_t)!

➡ Learn Gaussian denoiser
(and use many small steps)!

Diffusion Models

The Learnt Reverse Generative Process



$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \underbrace{\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)}, \sigma_t^2 \mathbf{I})$$

Trainable network
(U-Net, Denoising Autoencoder)

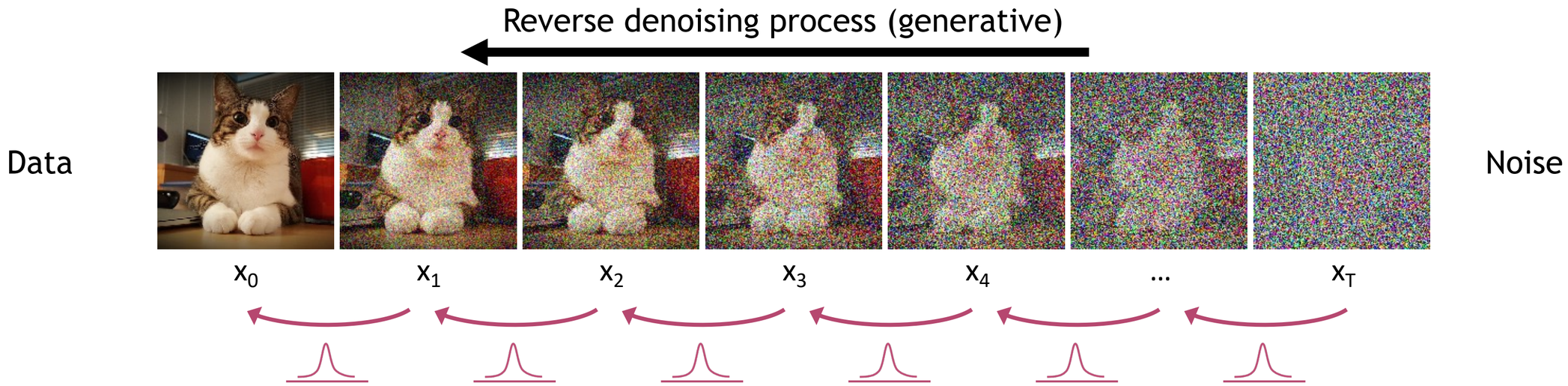
$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Trainable with maximum likelihood
(evidence lower bound/ELBO)

Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS* 2020

Diffusion Models

Evidence Lower Bound Objective



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

Constant

Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS* 2020

Diffusion Models

Evidence Lower Bound Objective

Reverse denoising process (generative)

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

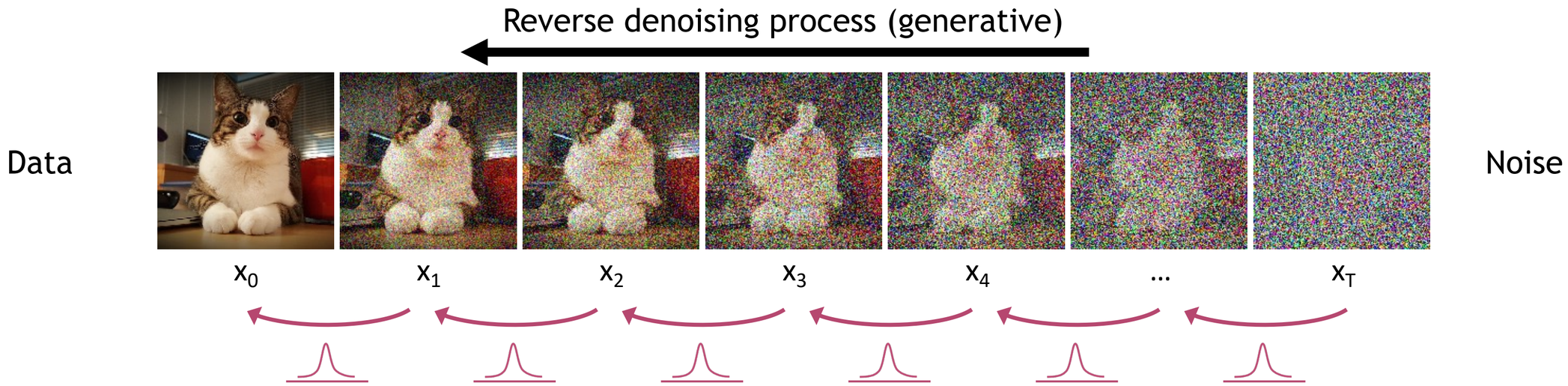
Constant

Matching Denoiser to Forward Diffusion

Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS* 2020

Diffusion Models

Evidence Lower Bound Objective



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

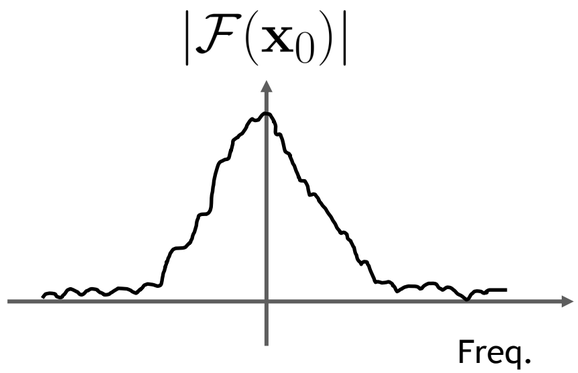
Constant

Matching Denoiser to Forward Diffusion

Final step clean data prediction.

Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS* 2020

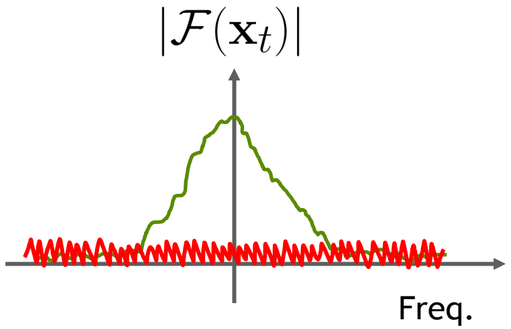
What happens to an image in the forward diffusion process?

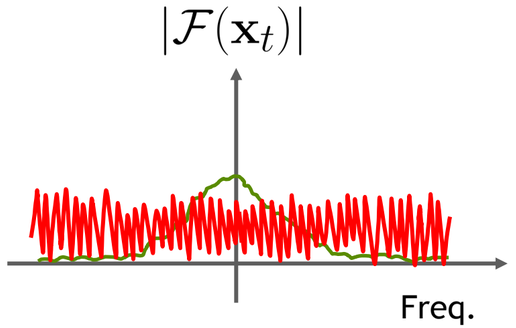


Spatial frequency spectrum
of natural images.

Small t

Large t





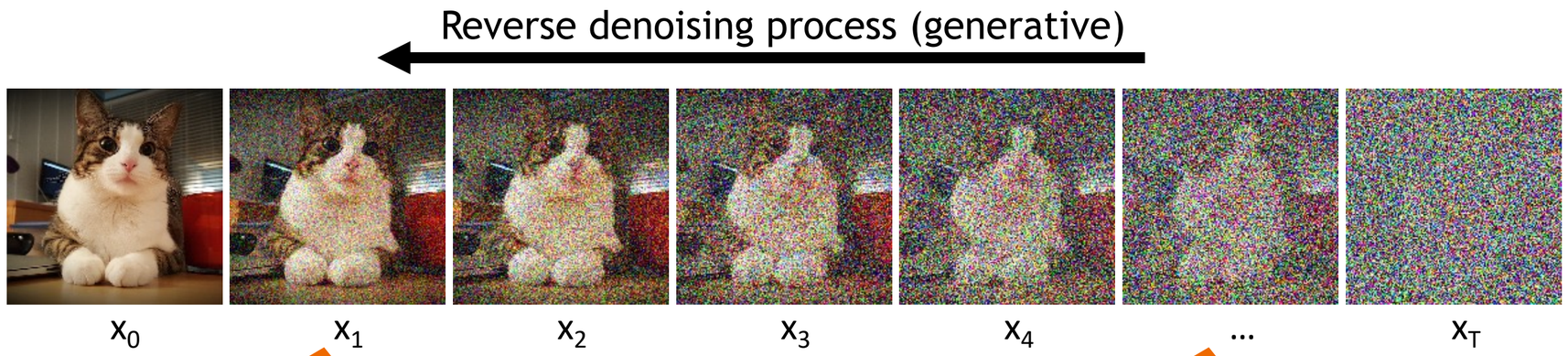
In forward diffusion, high frequency content is perturbed faster.

Data

x_0 x_1 x_2 x_3 x_4 ... x_T

Noise

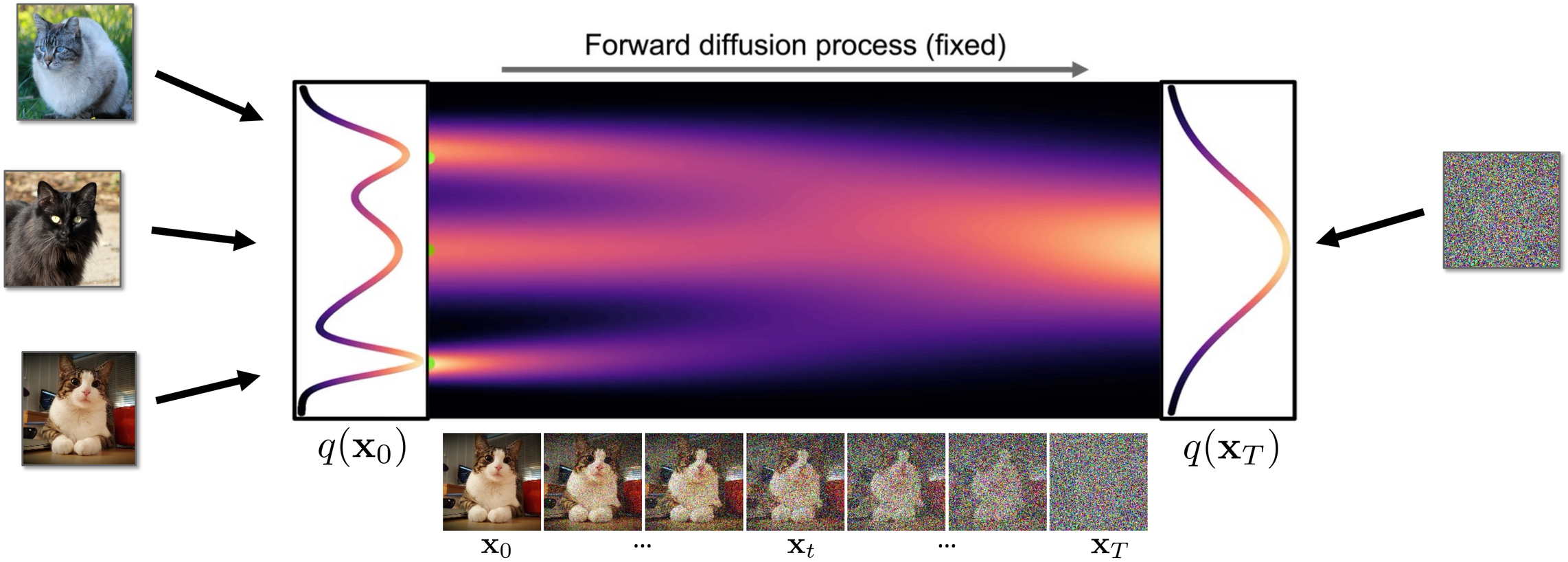
Content-Detail Tradeoff



The weighting of the training objective for different timesteps is important!

Diffusion Models

A Stochastic Differential Equation-based Perspective

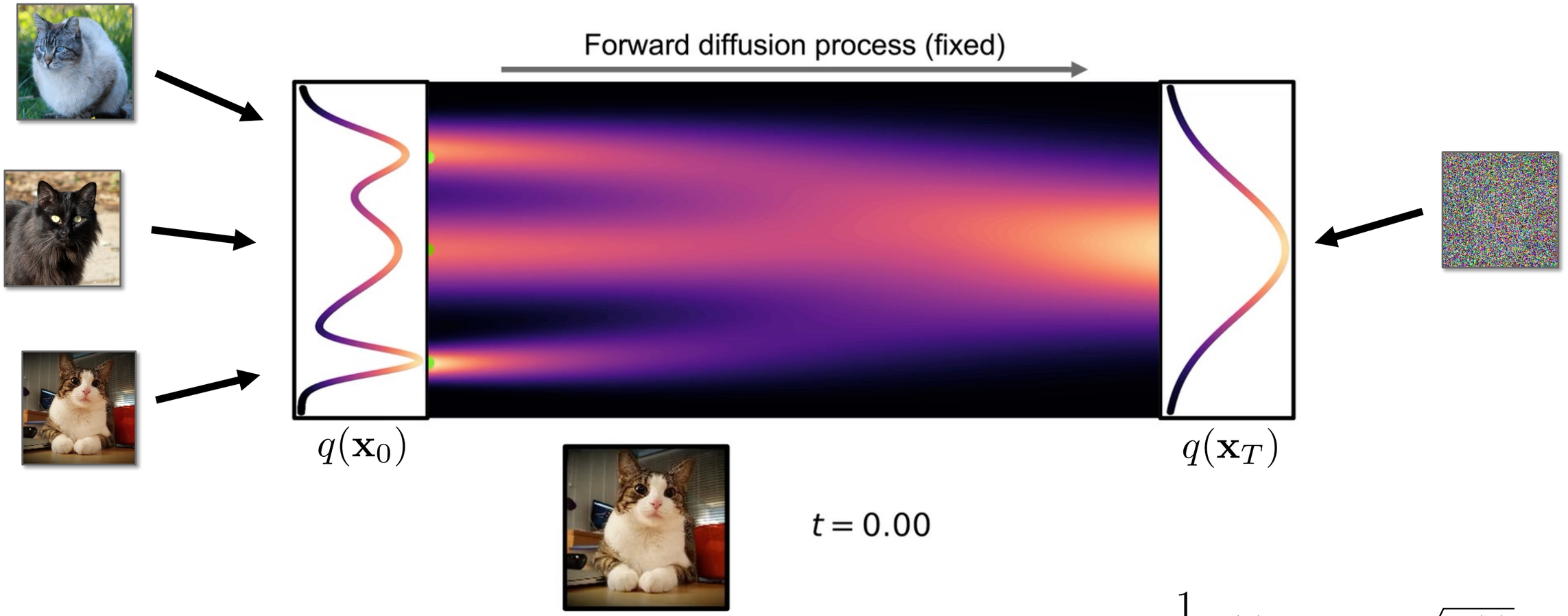


$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021
Anderson, “Reverse-time diffusion equation models”, *Stochastic Processes and their Applications*,1982

Diffusion Models

A Stochastic Differential Equation-based Perspective

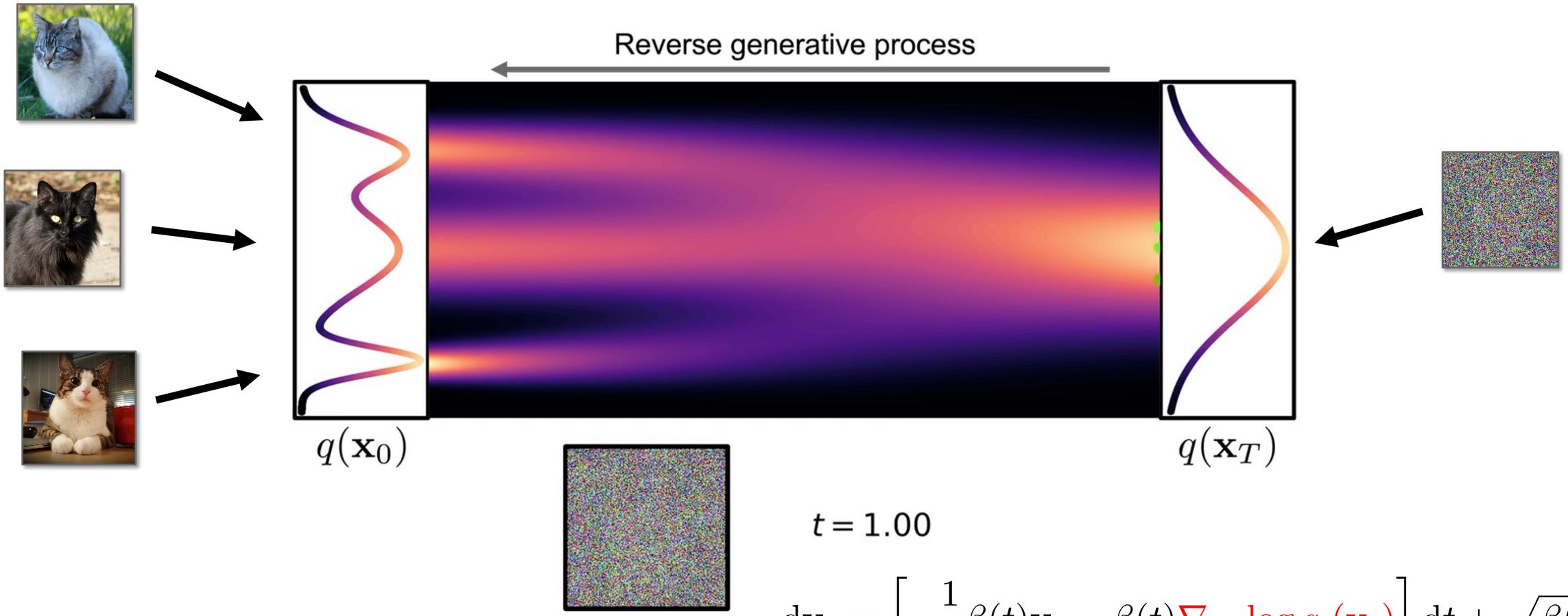


$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021
Anderson, “Reverse-time diffusion equation models”, *Stochastic Processes and their Applications*,1982

Diffusion Models

A Stochastic Differential Equation-based Perspective



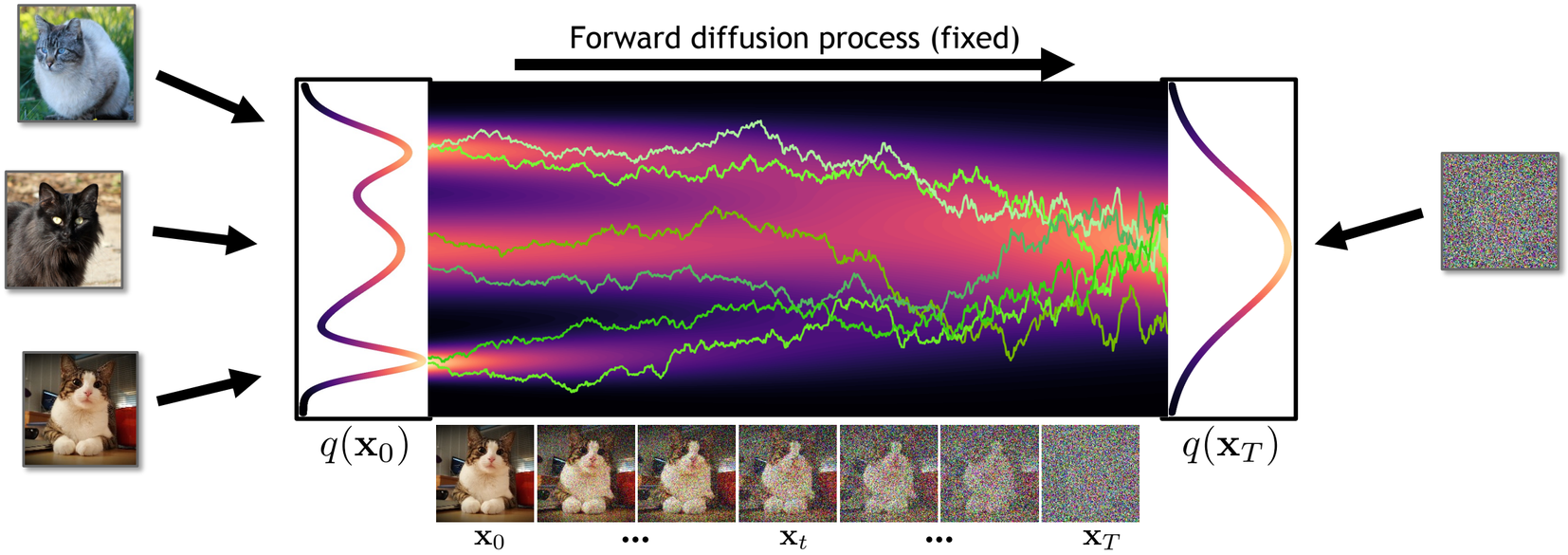
$t = 1.00$

$$\mathrm{d}\mathbf{x}_t = \left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)}_{\text{“Score Function” – modeled by neural network}} \right] \mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t$$

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021
Anderson, “Reverse-time diffusion equation models”, *Stochastic Processes and their Applications*,1982

Diffusion Models

Training Diffusion Models with Denoising Score Matching



$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)}\mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)}||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t|\mathbf{x}_0)||_2^2$$

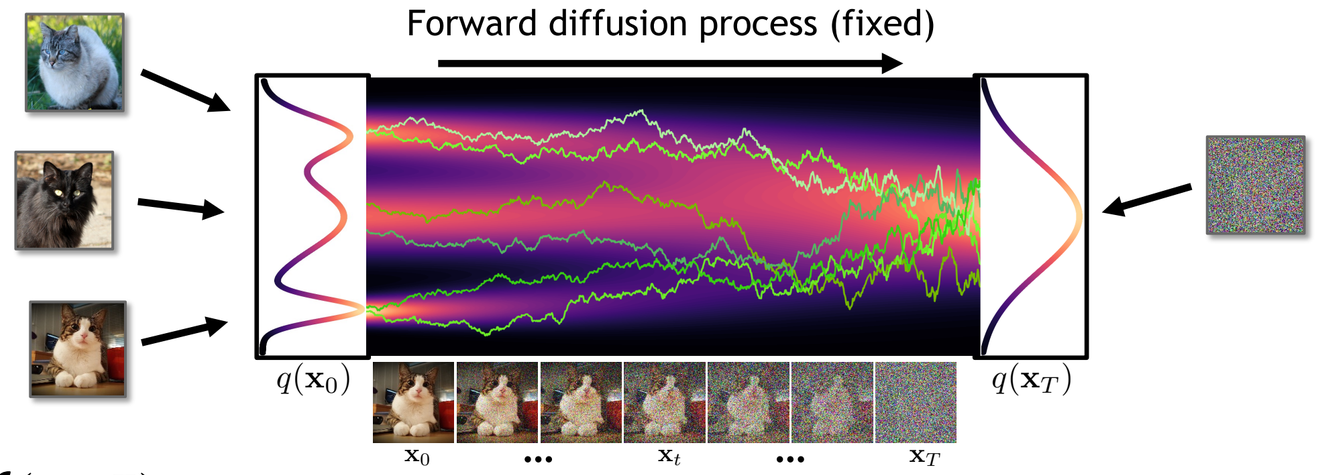
diffusion time t	data sample $\mathbf{x}_0$	diffused data sample $\mathbf{x}_t$	neural network	score of diffused data sample

➡ **After expectations,** $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t) \approx \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)$**!**

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021

Diffusion Models

General Training Objective in Practice



Diffusion noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

Data perturbation $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$ (α_t, σ_t define "noise schedule")

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t \sim \mathcal{U}(0,T)} \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0)} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \; w(t) \; ||\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2$$

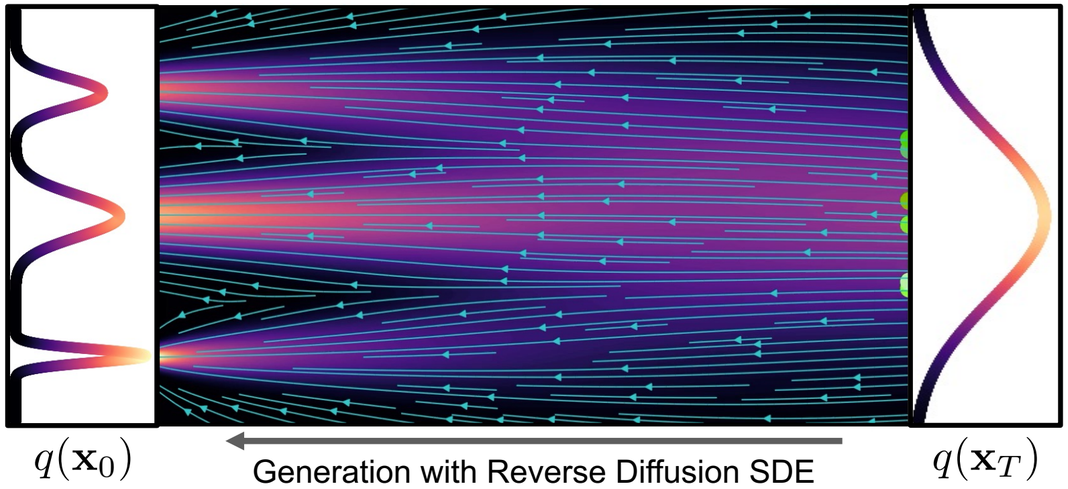
- $\mathbb{E}_{t \sim \mathcal{U}(0,T)}$: diffusion time t
- $\mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0)}$: data sample $\mathbf{x}_0$
- $\mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})}$: diffusion noise $\boldsymbol{\epsilon}$
- $w(t)$: loss weighting function
- $||\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon}||_2^2$: noise prediction objective

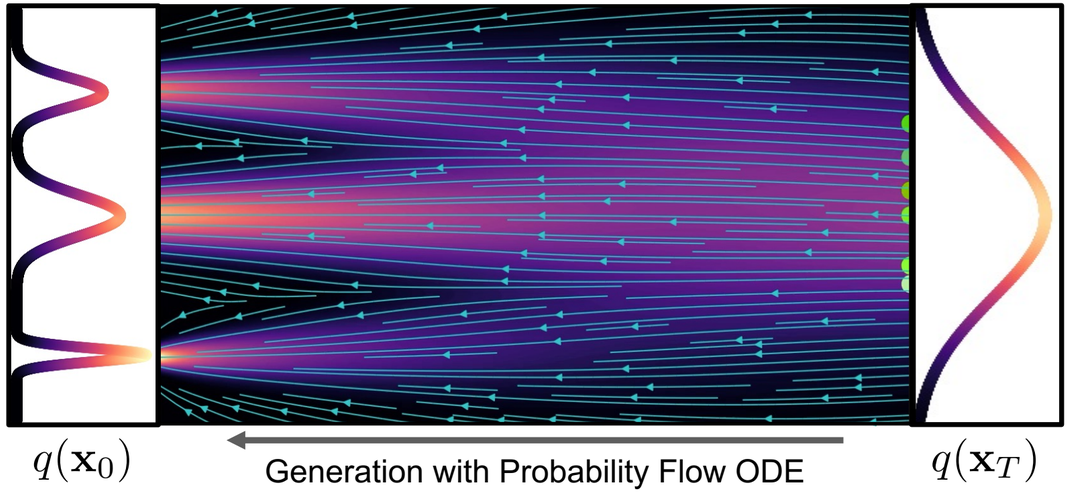
$$\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t) = -\frac{\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sigma_t}$$

Ho et al., "Denoising Diffusion Probabilistic Models", *NeurIPS* 2020

Kingma and Gao, "Understanding Diffusion Objectives as the ELBO with Simple Data Augmentation", NeurIPS, 2023

Synthesis with SDE vs. ODE





- **Generative Reverse Diffusion SDE (stochastic):**

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\left[\mathbf{x}_t + 2\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\right]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t$$

- **Generative Probability Flow ODE (deterministic):**

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\left[\mathbf{x}_t + \nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)\right]\mathrm{d}t$$

Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021

Flow Matching

The vector field between noise and data distribution can be learnt directly via ***Flow Matching***.

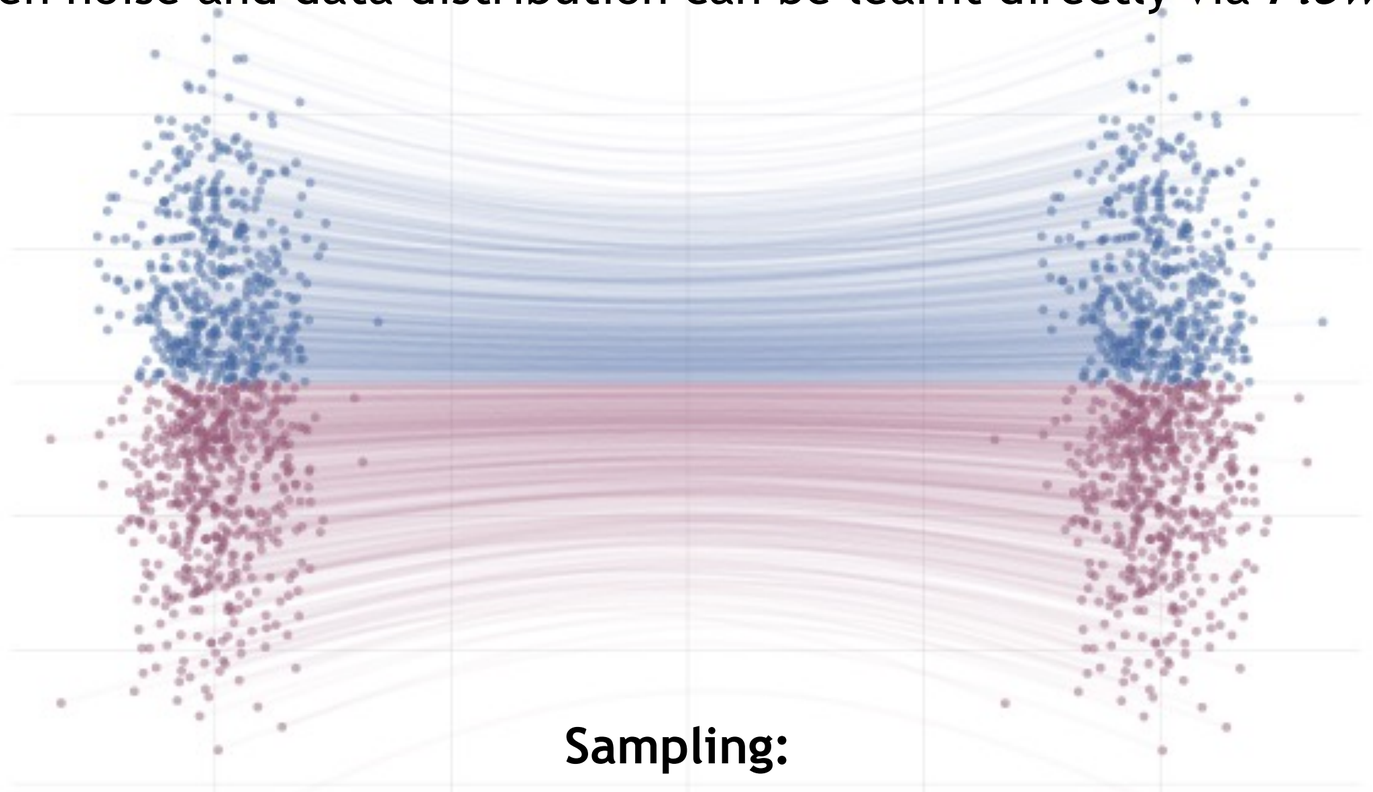
Training:

Interpolate between random pairs of noise and data distributions and learn vector field.

Lipman et al., “Flow Matching for Generative Modeling”, *ICLR*, 2023
Albergo et al., “Stochastic Interpolants: A Unifying Framework for Flows and Diffusions”, arXiv, 2023
https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Flow Matching

The vector field between noise and data distribution can be learnt directly via ***Flow Matching***.



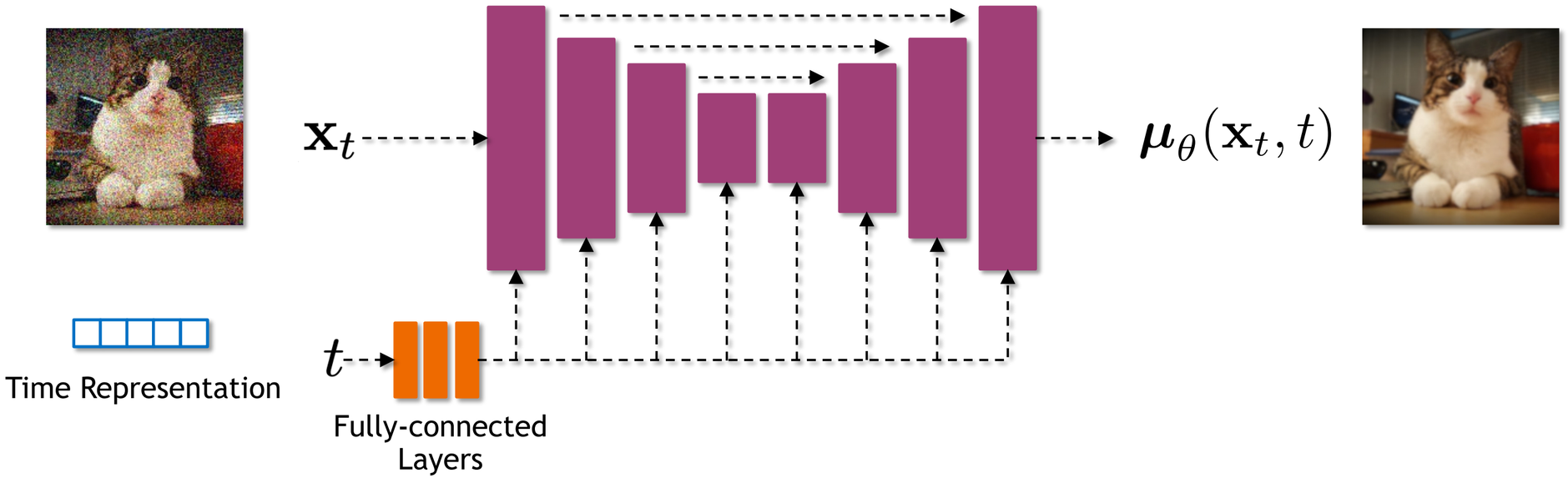
Sampling:

After training, we obtain smooth vector field connecting the distributions.

Lipman et al., “Flow Matching for Generative Modeling”, *ICLR*, 2023
Albergo et al., “Stochastic Interpolants: A Unifying Framework for Flows and Diffusions”, arXiv, 2023
https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Diffusion Model Architectures

Diffusion models often use U-Net architectures with Conv. ResNet blocks, skip connections, and self-attention layers.



Time representation: sinusoidal positional embeddings or random Fourier features.

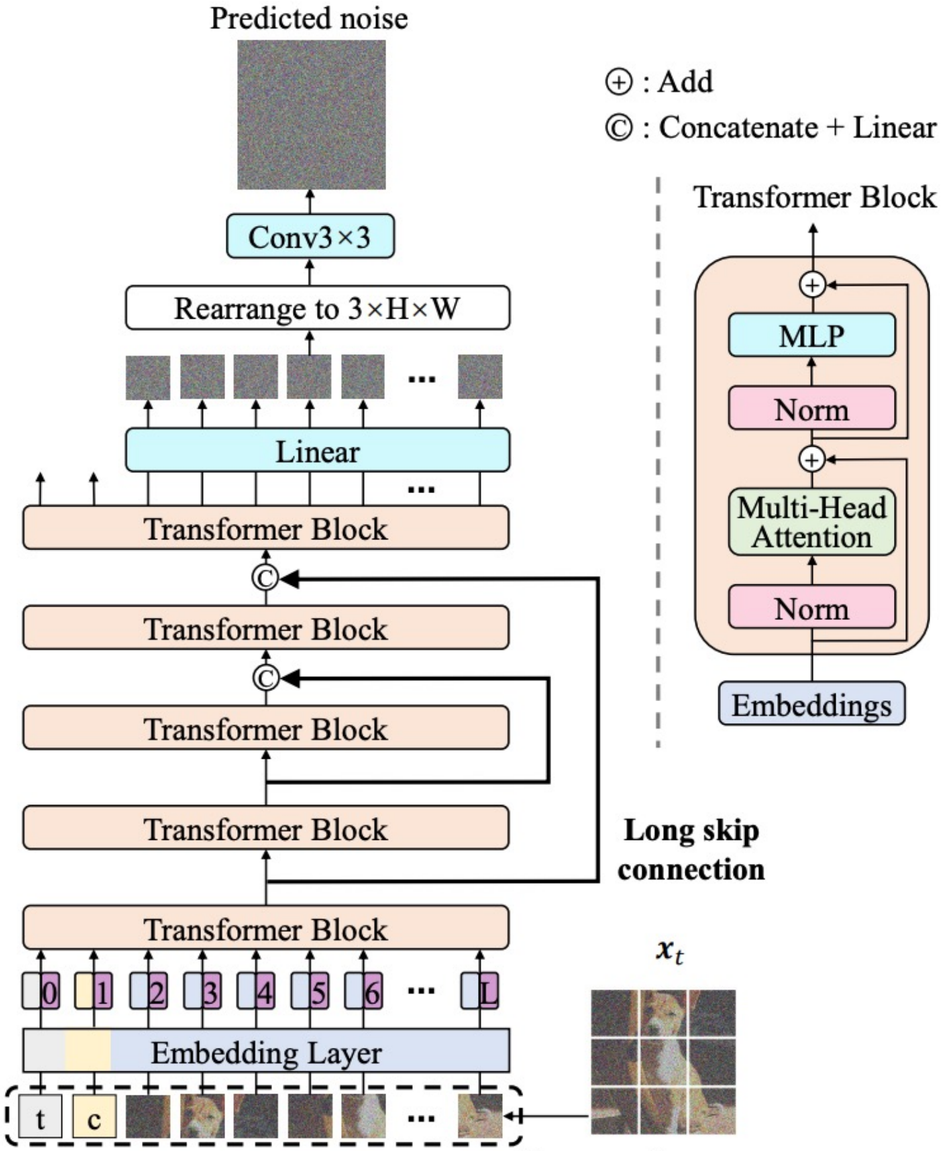
Time features are fed to residual blocks using either simple spatial addition or adaptive group normalization layers.

Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS*, 2020
Dhariwal and Nichol, “Diffusion Models Beat GANs on Image Synthesis”, *NeurIPS*, 2021
Karras et al., “Elucidating the Design Space of Diffusion-Based Generative Models”, *NeurIPS*, 2022
Hoogeboom et al., “simple diffusion: End-to-end diffusion for high resolution images”, *ICML*, 2023
Karras et al., “Analyzing and Improving the Training Dynamics of Diffusion Models”, *CVPR*, 2024

Diffusion Transformers

Vision transformer backbone for diffusion models:

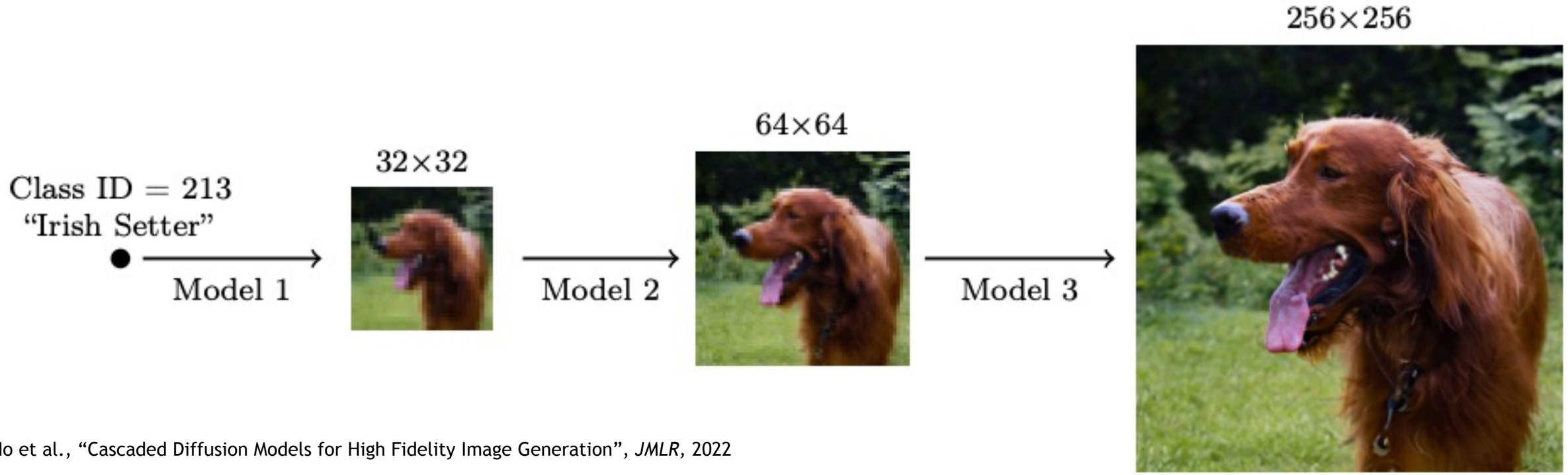
- Noise and denoise tokenized image patches.
- Similar architectures as in LLMs.



Bao et al., “All are Worth Words: A ViT Backbone for Diffusion Models”, *CVPR*, 2023
Peebles and Xie, “Scalable Diffusion Models with Transformers”, *ICCV*, 2023
Hatamizadeh et al., “DiffiT: Diffusion Vision Transformers for Image Generation”, 2023

Cascaded Diffusion Models

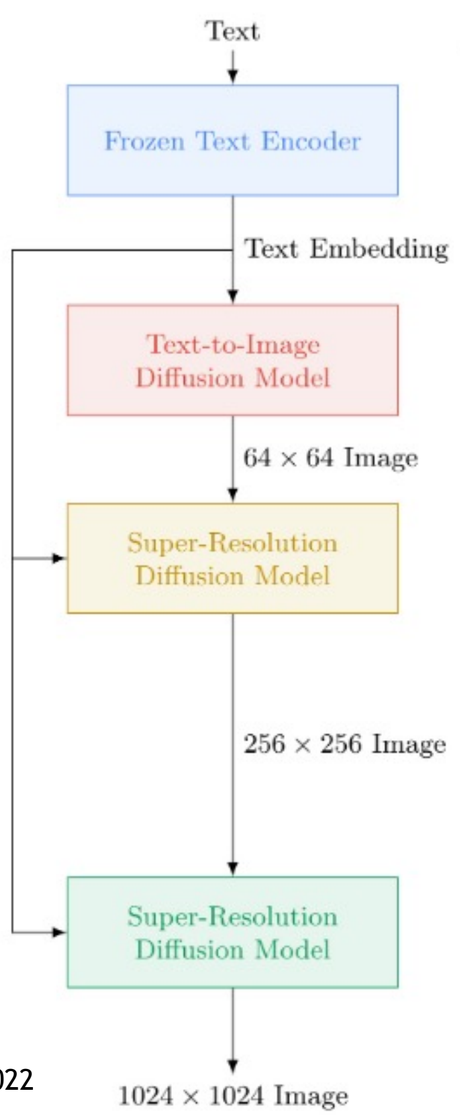
- Directly generating high-resolution images is hard.
- Partition generation into base model (focus on semantics) and upsamplers (focus on quality)
- Implemented by concatenating low-res. conditioning image to noise inputs of denoiser network

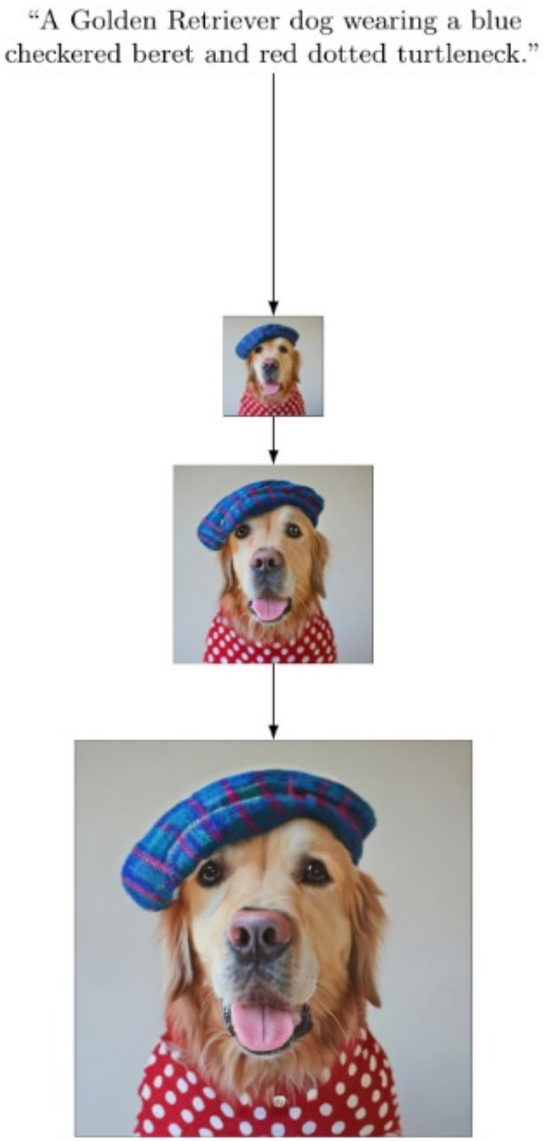


Ho et al., "Cascaded Diffusion Models for High Fidelity Image Generation", *JMLR*, 2022

Text-Guided Diffusion Models

- Create text embedding with text encoder (CLIP, T5, etc.)
- Condition diffusion model on text.
- Cross attention: Attention layer that attend from pixels to text (Q: pixels, K & V: text)





Saharia et al., "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding", *NeurIPS*, 2022
Balaji et al., "eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers", *arXiv*, 2022

Text-Guided Diffusion Models

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

A highly detailed digital painting of a portal in a mystic forest with many beautiful trees. A person is standing in front of the portal.

Saharia et al., “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding”, *NeurIPS*, 2022
Balaji et al., “eDiff-I: Text-to-Image Diffusion Models with an Ensemble of Expert Denoisers”, *arXiv*, 2022

A Crucial Trick: Classifier(-free) Guidance

- **Unconditional generation:** $\nabla_{\mathbf{x}_t} \log\, p(\mathbf{x}_t)$ used score during iterative generation.
- **Conditional generation:** $\nabla_{\mathbf{x}_t} \log\, p(\mathbf{x}_t|c)$
- **Classifier guidance:** Train classifier $p(c|\mathbf{x}_t)$ and use $\nabla_{\mathbf{x}_t} \left[\log\, p(\mathbf{x}_t|c) + \omega \log\, p(c|\mathbf{x}_t)\right]$

  → approximately samples from $\tilde{p}(\mathbf{x}_t|c) \propto p(\mathbf{x}_t|c)\, p(c|\mathbf{x}_t)^{\omega}$

- **Classifier-free guidance:** Construct implicit classifier $p(c|\mathbf{x}_t) \propto \dfrac{p(\mathbf{x}_t|c)}{p(\mathbf{x}_t)}$

  ← *Conditional diffusion model*

  ← *Unconditional diffusion model*

→ $\nabla_{\mathbf{x}_t} \left[\log p(\mathbf{x}_t|c) + \omega \log p(c|\mathbf{x}_t)\right] = \nabla_{\mathbf{x}_t} \left[(1+\omega) \log p(\mathbf{x}_t|c) - \omega \log p(\mathbf{x}_t)\right]$

Dhariwal and Nichol, “Diffusion Models Beat GANs on Image Synthesis”, *NeurIPS*, 2021
Ho and Salimans, “Classifier-Free Diffusion Guidance”, arXiv, 2021

A Crucial Trick: Classifier(-free) Guidance

Non-guided samples

Guided samples $(\omega = 3.0)$

Dhariwal and Nichol, “Diffusion Models Beat GANs on Image Synthesis”, *NeurIPS*, 2021
Ho and Salimans, “Classifier-Free Diffusion Guidance”, arXiv, 2021

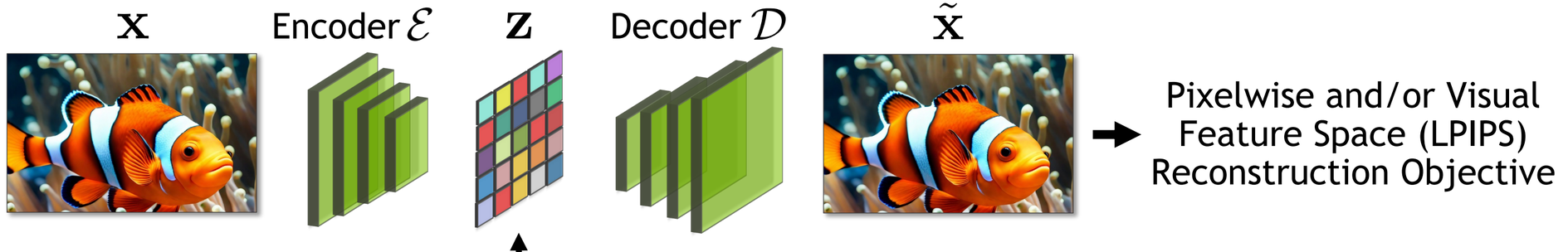
Latent Diffusion Models

Map Data into Compressed Latent Space. Train Diffusion Model efficiently in Latent Space.

- Stage 1:

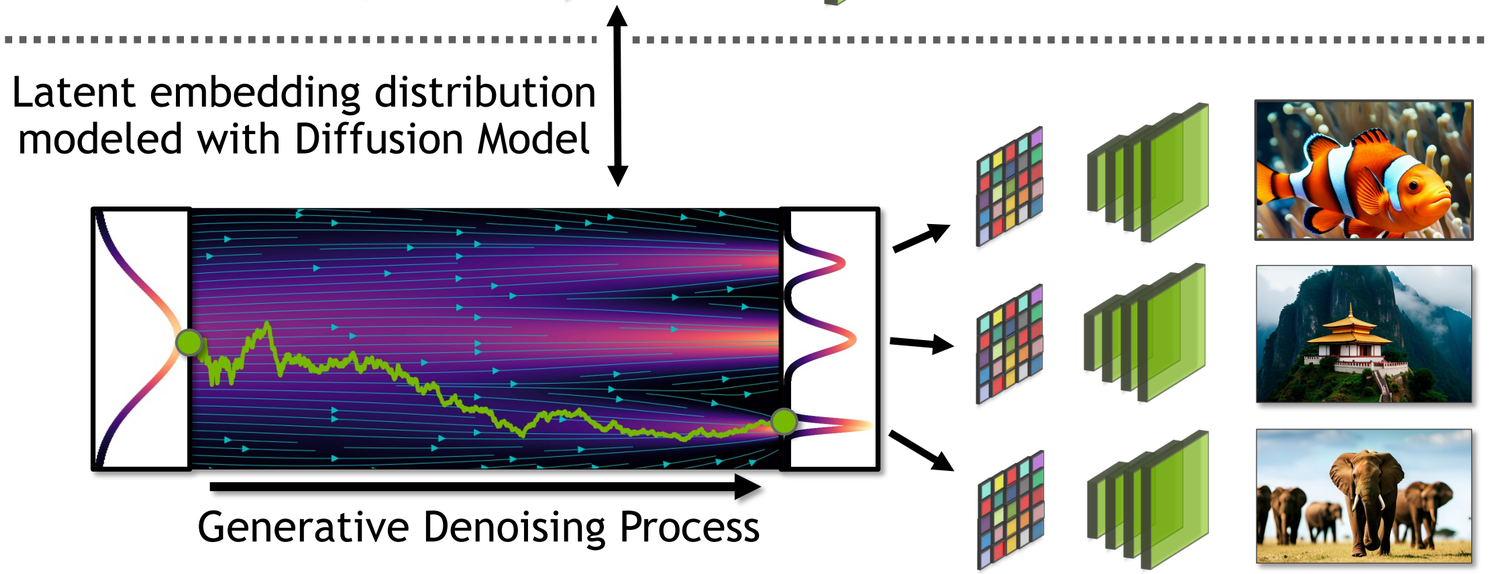
Train Autoencoder

$$\tilde{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$$



- Stage 2:

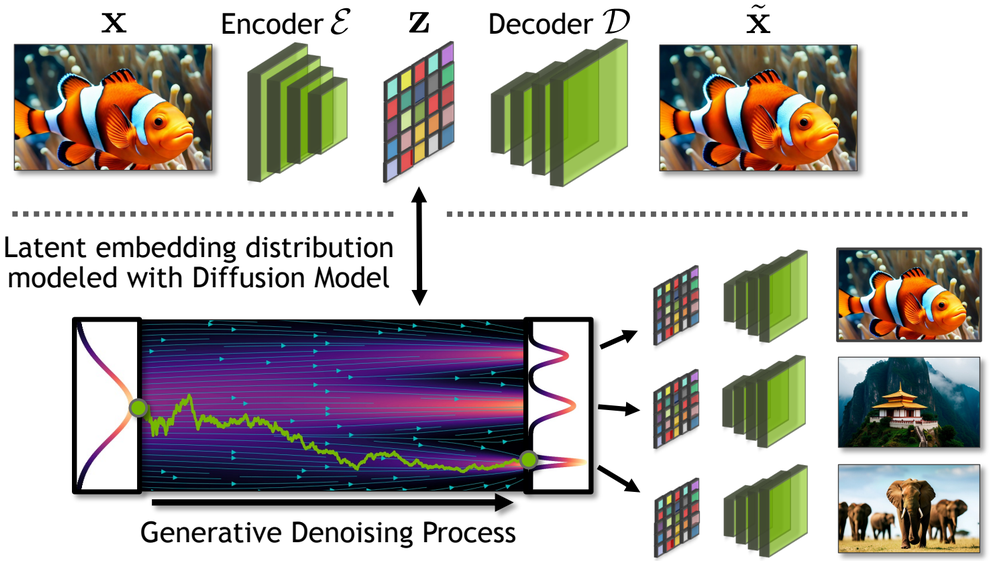
Train **Latent** Diffusion Model



Vahdat et al., “Score-based Generative Modeling in Latent Space”, *NeurIPS*, 2021
Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, *CVPR*, 2022
Sinha et al., “D2C: Diffusion-Denoising Models for Few-shot Conditional Generation”, *NeurIPS*, 2021
Mittal et al., “Symbolic Music Generation with Diffusion Models”, *ISMIR*, 2021

Latent Diffusion Models

Map Data into Compressed Latent Space. Train Diffusion Model efficiently in Latent Space.



Advantages:

1. *Compressed latent space:* Train diffusion model in **lower resolution** latent space ➡ **computationally more efficiently**
2. *Regularized smooth/compressed latent space:* **Easier task** for diffusion model and **faster sampling**
3. *Flexibility*: **Autoencoder can be tailored to data** (images, video, text, graphs, 3D point clouds, meshes, etc.)

Vahdat et al., "Score-based Generative Modeling in Latent Space", *NeurIPS*, 2021
Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", *CVPR*, 2022
Sinha et al., "D2C: Diffusion-Denoising Models for Few-shot Conditional Generation", *NeurIPS*, 2021
Mittal et al., "Symbolic Music Generation with Diffusion Models", *ISMIR*, 2021

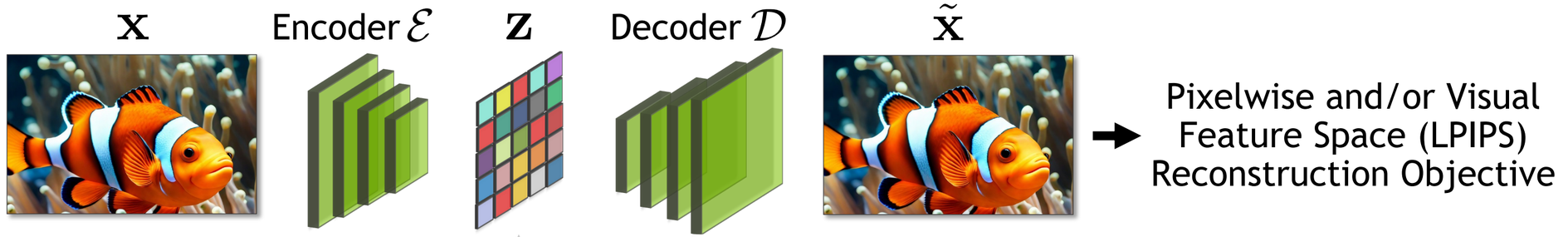
Latent Diffusion Models

Add Adversarial Patch-based Discriminator on top of Reconstruction Loss for Perceptual Compression

- Stage 1:

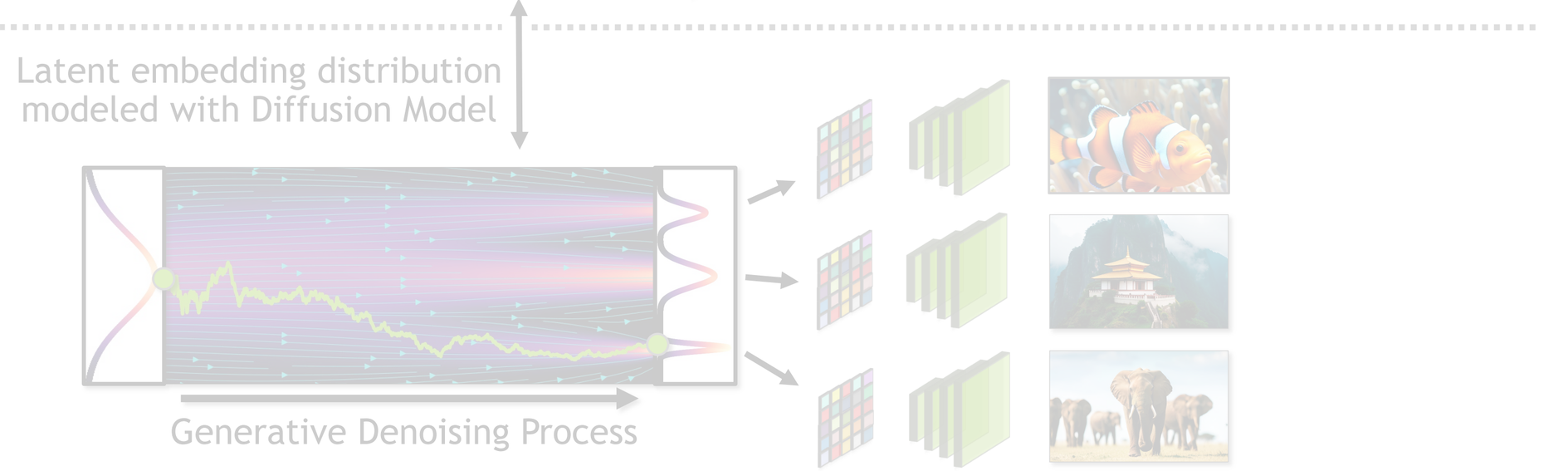
Train Autoencoder

$$\tilde{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$$



- Stage 2:

Train Latent Diffusion Model



Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, *CVPR*, 2022

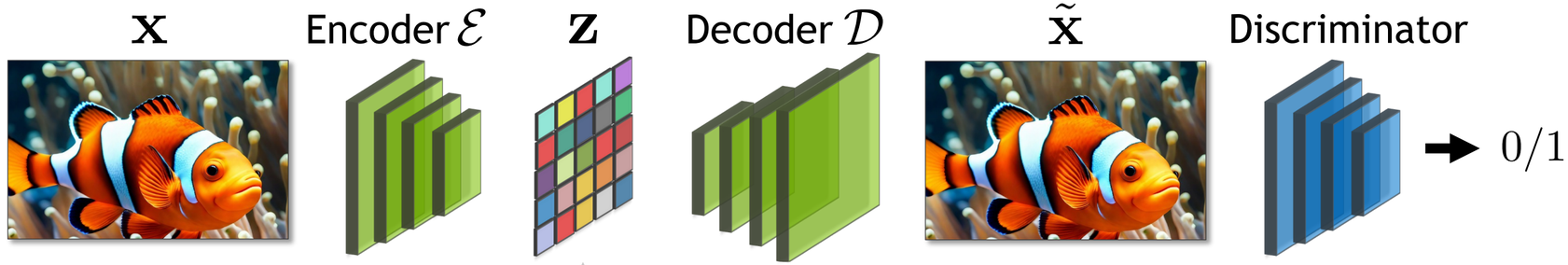
Latent Diffusion Models

Add Adversarial Patch-based Discriminator on top of Reconstruction Loss for Perceptual Compression

- Stage 1:

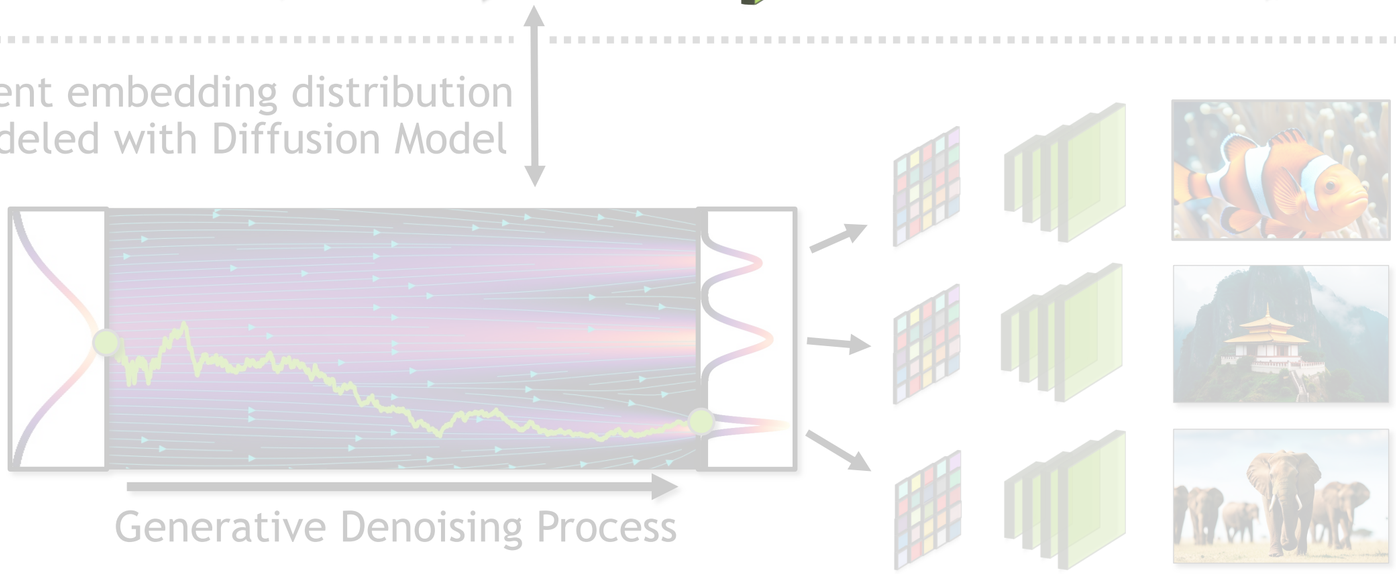
Train Autoencoder

$$\tilde{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$$



- Stage 2:

Train Latent Diffusion Model



Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, *CVPR*, 2022

Input

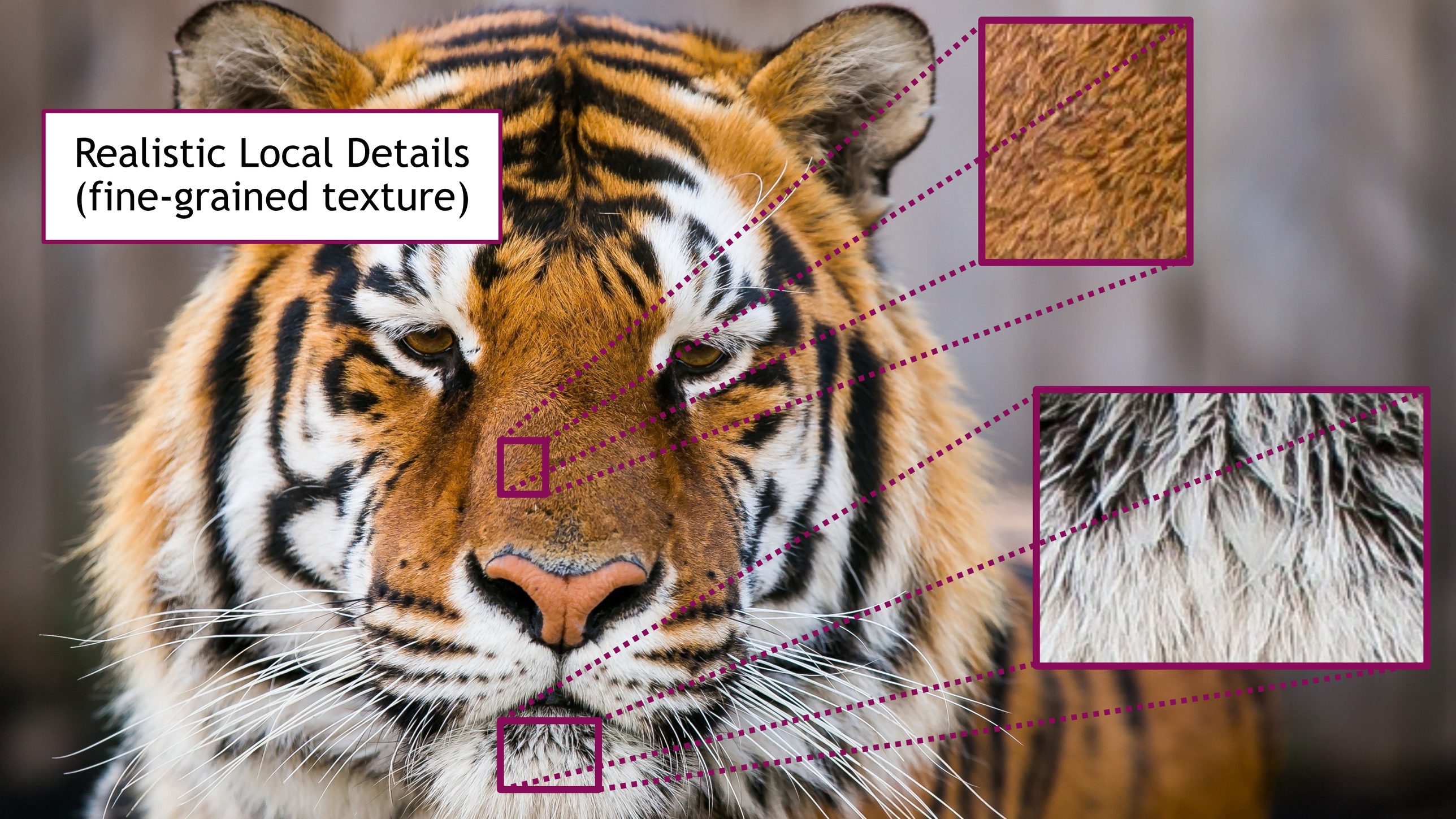
Reconstruction without Discriminator

Reconstruction with Discriminator

Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, *CVPR*, 2022

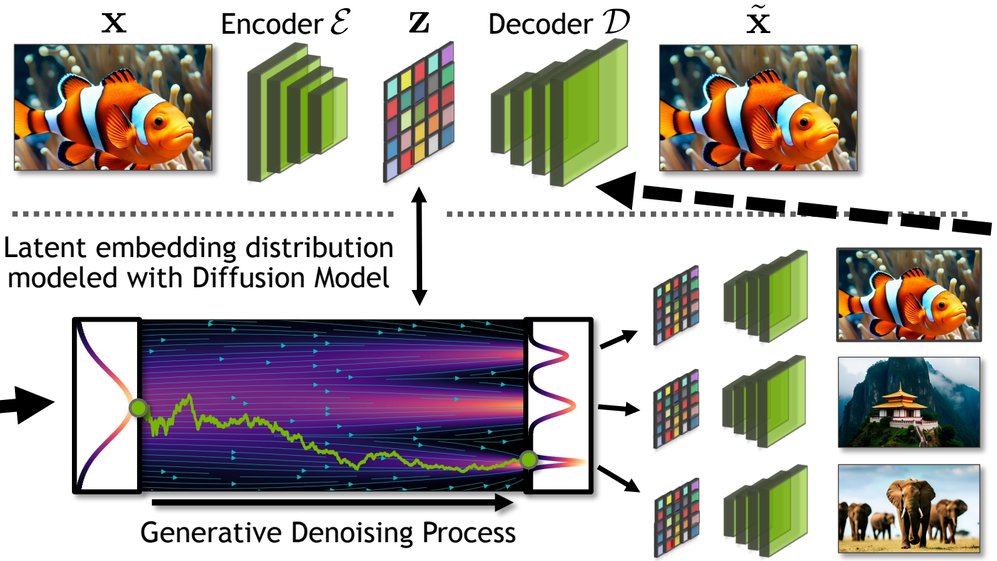

What makes an image look realistic and high-quality?


Realistic Global Structure
(correct placement of ears,
eyes, fur pattern, etc.)

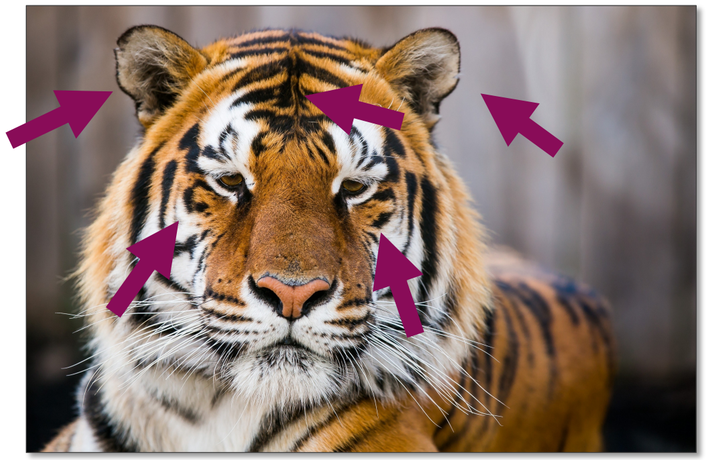

Realistic Local Details
(fine-grained texture)

Latent Diffusion Models

Map Data into Compressed Latent Space. Train Diffusion Model efficiently in Latent Space.



Global semantic structure modeled by latent **diffusion model.**



Local "imperceptible" details generated by **decoder** (with adversarial objective).

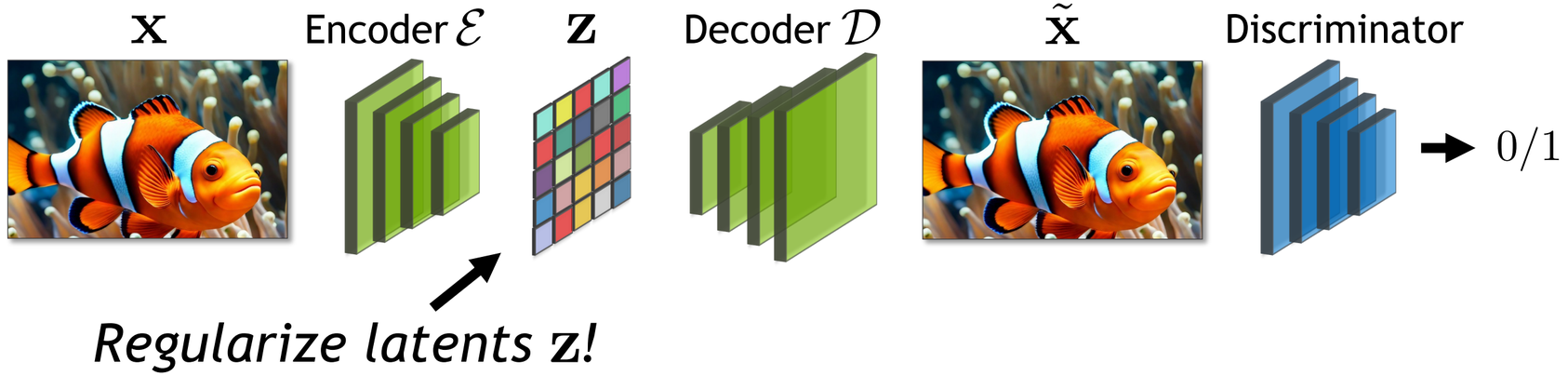
Latent space compression needs to be tuned carefully!

Balance between what content the latent DM needs to model and what is generated by decoder!

Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", *CVPR*, 2022

Latent Space Regularization

Regularize Latent Space for better Compression and easier Training of Latent Space Diffusion Models



- **Option 1: Kullback-Leibler (KL) regularization**

  Parametrize encoder by diagonal Gaussian, regularize towards standard normal distribution (as in regular VAEs).

  Use very small weight for KL regularization term (weak regularization).

- **Option 2: Vector Quantization (VQ) regularization**

  Discretize latent encodings using finite-sized learnable codebook as in VQ-VAEs (implemented by vector-quantization layer in decoder).

  Use large codebook size (weak regularization).

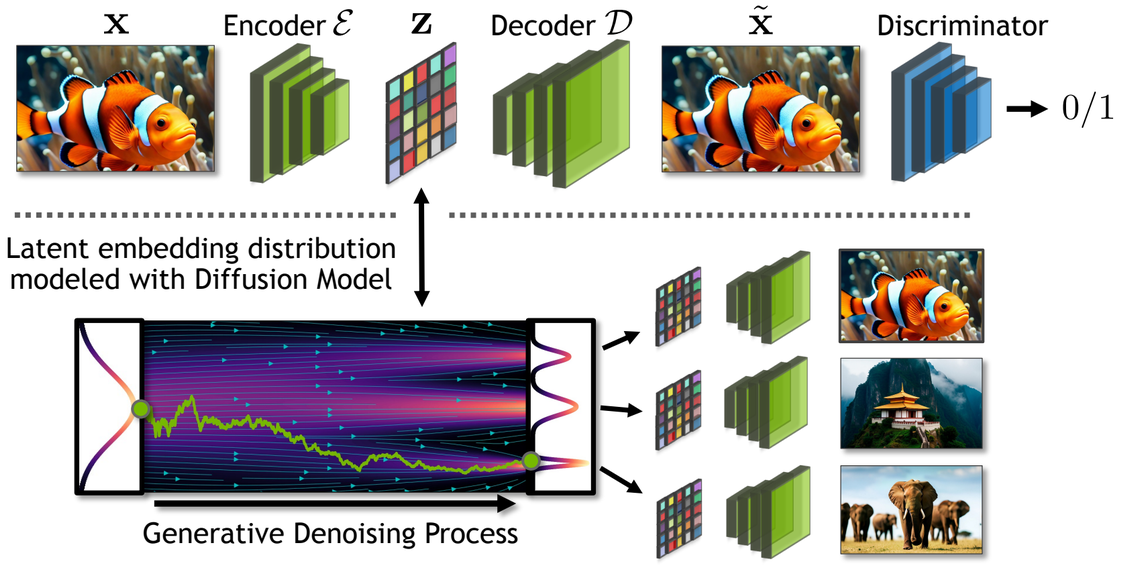
Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", *CVPR*, 2022
Esser et al., "*Taming Transformers for High-Resolution Image Synthesis*", *CVPR*, 2021

Latent Diffusion Models

Latent Diffusion Models offer Excellent Trade-off between Performance and Compute Demands

LDM "Recipe":

1. Train strong autoencoder
    - Compress…
      (downsampling factor / latent space regularization)
    - …while ensuring high visual quality on reconstructions
      ("upper bound" on synthesis quality)



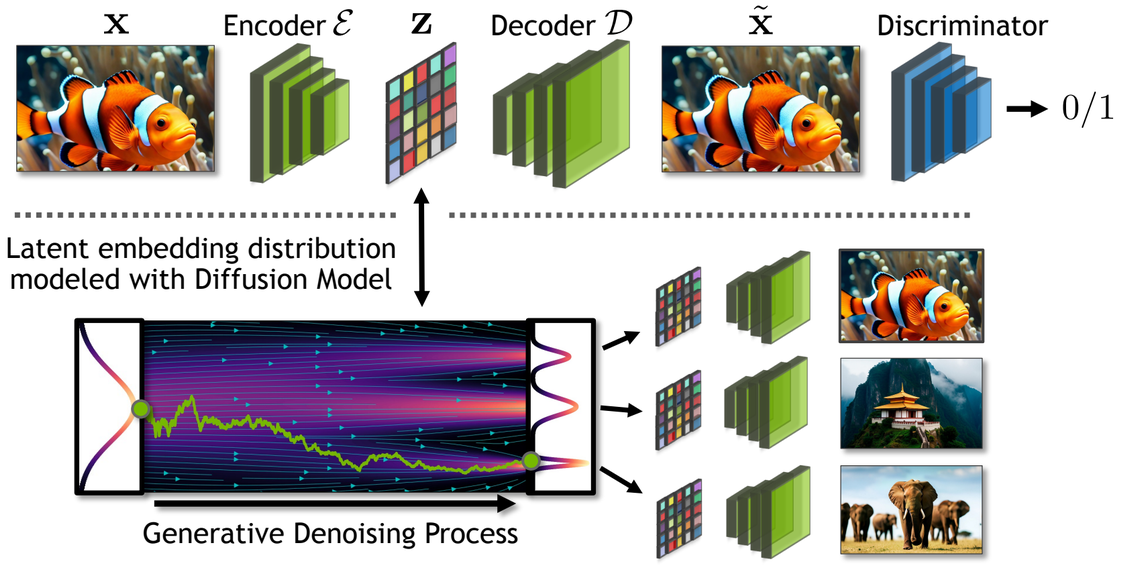
Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", *CVPR*, 2022

Latent Diffusion Models

Latent Diffusion Models offer Excellent Trade-off between Performance and Compute Demands

LDM "Recipe":

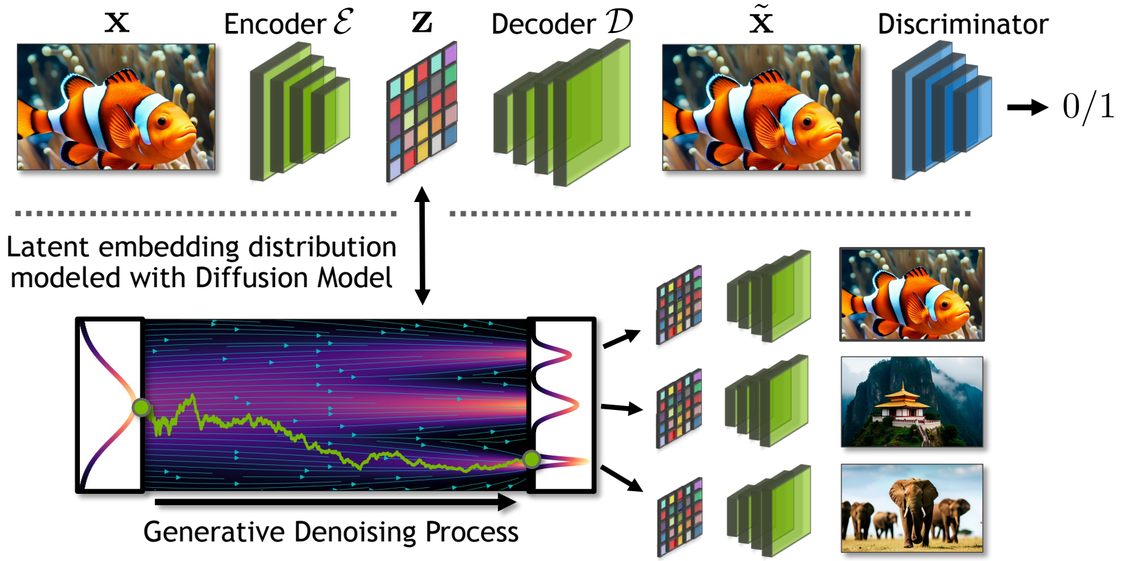
1. Train strong autoencoder
   - Compress…
     (downsampling factor / latent space regularization)
   - …while ensuring high visual quality on reconstructions
     ("upper bound" on synthesis quality)
2. Train efficient latent diffusion model
   - Latent space compression/regularization makes diffusion model training easier ➡ but trade-off with respect quality? Not really…
   - …because discriminator ➡ high quality despite compression (re-generate details, not encode)!



Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", *CVPR*, 2022

Latent Diffusion Models

Latent Diffusion Models offer Excellent Trade-off between Performance and Compute Demands



➡ LDM with appropriate regularization, compression, downsampling ratio and strong autoencoder reconstruction:

- **Computationally efficient** diffusion model in latent space (compression & lower resolution).
- Yet **very high-performance** (latent diffusion + autoencoder + discriminator = ❤️).
- **Highly flexible** (can adjust autoencoder for different tasks and data).

Image Generation with Latent Diffusion Models

Many state-of-the-art large-scale text-to-image models are latent diffusion models:

- Stability AI's **Stable Diffusion**
- Meta's **Emu**
- OpenAI's **Dall-E 3** and **Sora**

Common observation:

- (Latent) diffusion model **technology is mature** for practical image generation.
- The above models all achieve their high-performance mostly by **sophisticated data captioning** and **filtering** and **fine-tuning** strategies.

Rombach et al., "High-Resolution Image Synthesis with Latent Diffusion Models", *CVPR*, 2022
Dai et al., "Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack", *arXiv*, 2023
Betker et al., "Improving Image Generation with Better Captions" *(DALL-E 3), 2023*

https://stability.ai/news/stable-diffusion-sdxl-1-announcement

https://stability.ai/news/stable-diffusion-sdxl-1-announcement

”An old man with green eyes and a long grey beard”

”A robot cooking dinner in the kitchen”

”A cool orange cat wearing sunglasses playing a guitar with a group of dancing bananas”

”A bread, an apple, and a knife on a table”

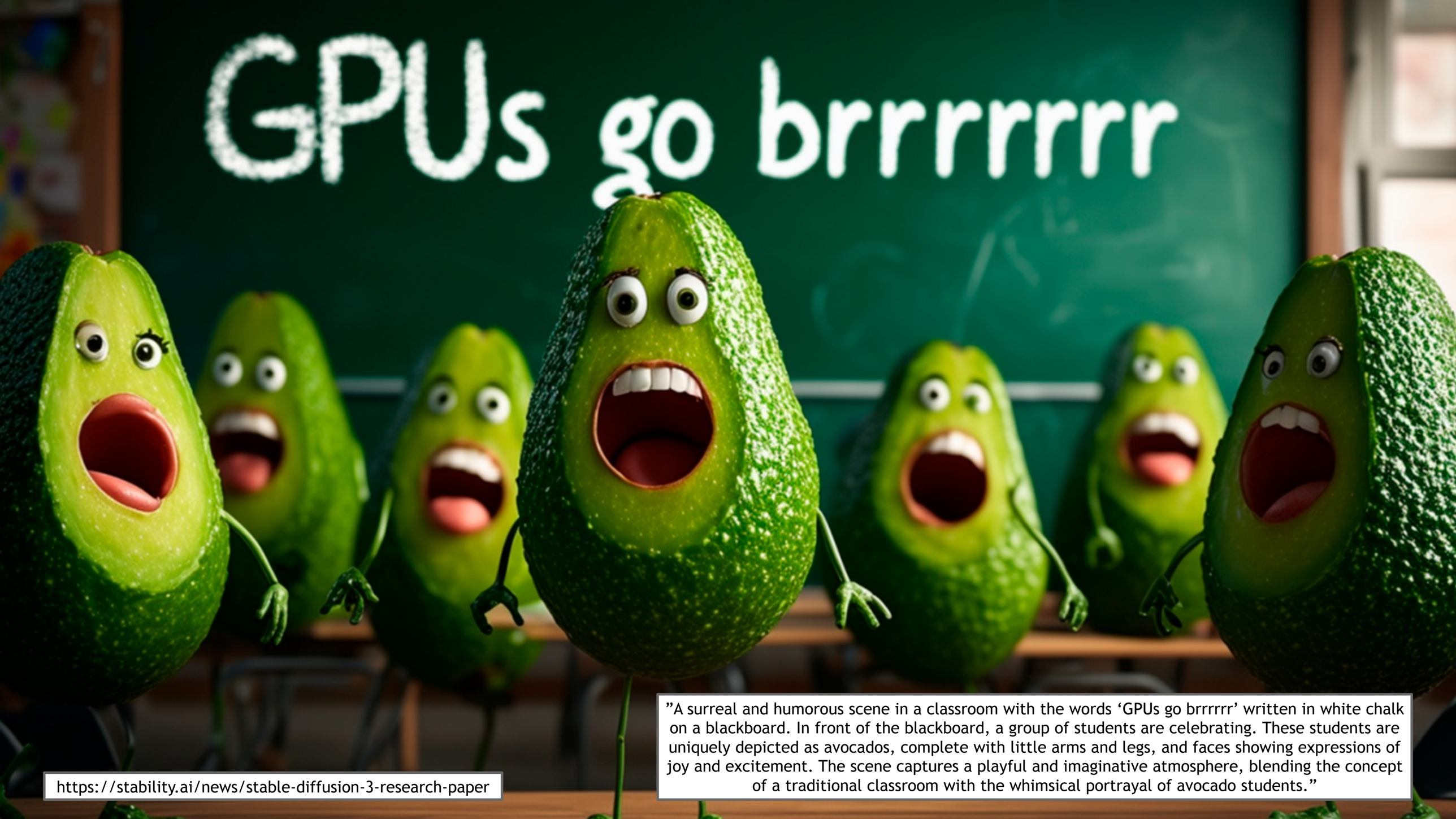
”A traditional tea house in a tranquil garden with blooming cherry blossom trees”

”A painting of an adorable rabbit sitting on a colorful splash”

Dai et al., “Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack”, *arXiv,* 2023

”Frog sitting in a 1950s diner wearing a leather jacket and a top hat. On the table is a giant burger and a small sign that says ”Froggy Fridays””

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



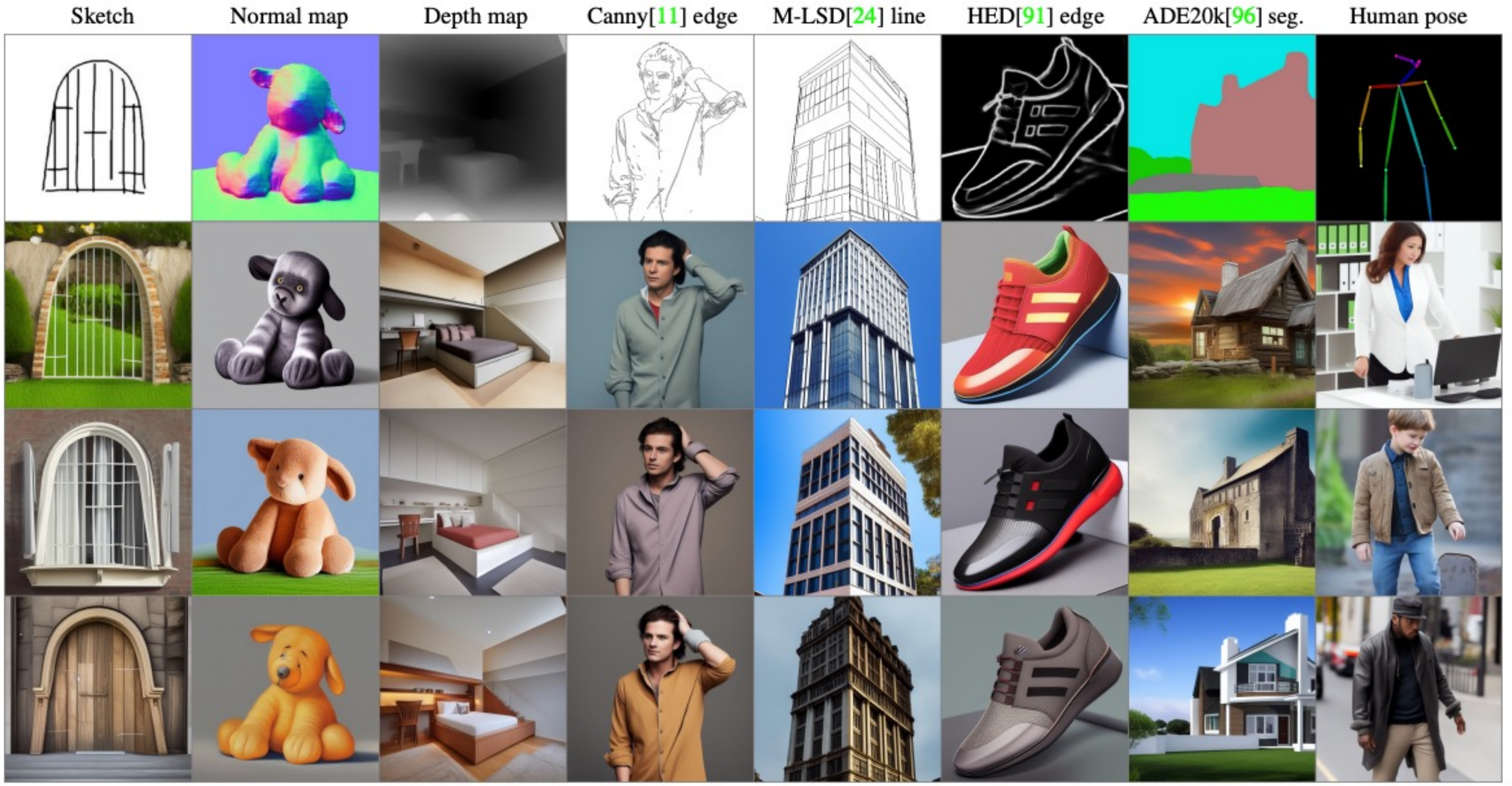
https://stability.ai/news/stable-diffusion-3-research-paper

”A surreal and humorous scene in a classroom with the words ‘GPUs go brrrrrr’ written in white chalk on a blackboard. In front of the blackboard, a group of students are celebrating. These students are uniquely depicted as avocados, complete with little arms and legs, and faces showing expressions of joy and excitement. The scene captures a playful and imaginative atmosphere, blending the concept of a traditional classroom with the whimsical portrayal of avocado students.”

Betker et al., “Improving Image Generation with Better Captions” *(DALL-E 3), 2023*

Betker et al., “Improving Image Generation with Better Captions” *(DALL-E 3), 2023*

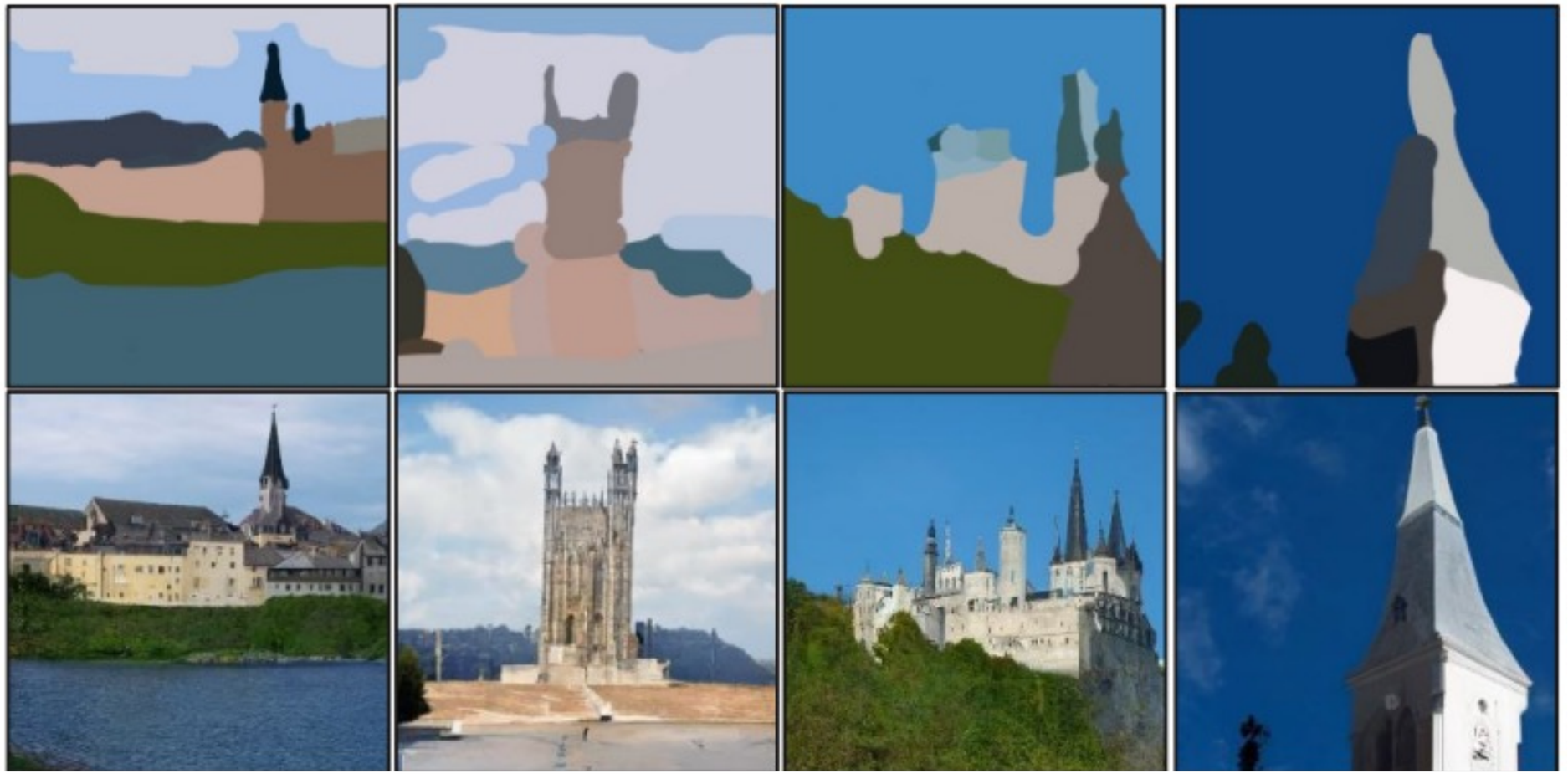
Controlling Latent Diffusion Models



Zhang et al., "Adding Conditional Control to Text-to-Image Diffusion Models", *ICCV*, 2023

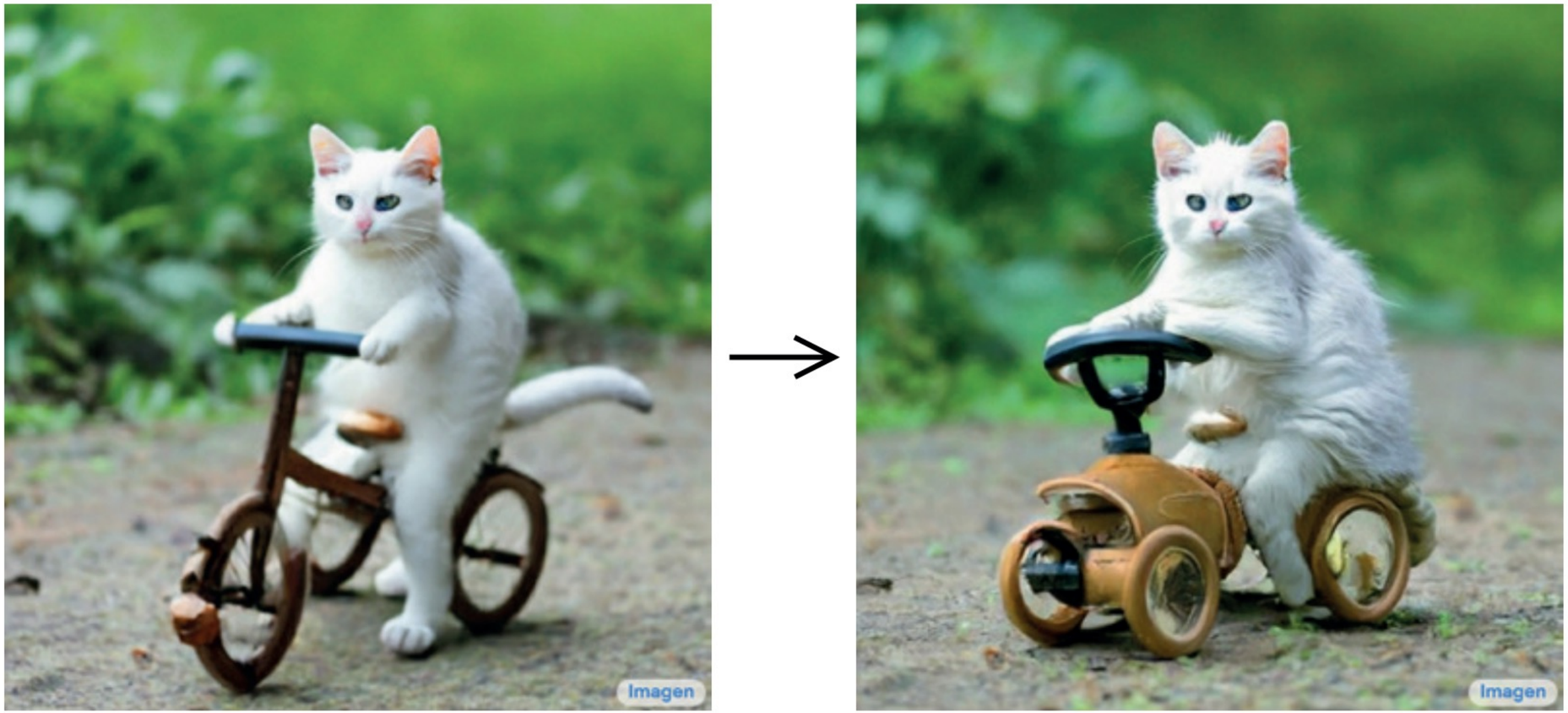
SDEdit

Use pre-trained diffusion models for editing (image-to-image)



Meng et al., "SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations", ICLR, 2022

Prompt-to-Prompt Editing



“Photo of a cat riding on a bicycle.”

car

Hertz et al., "Prompt-to-Prompt Image Editing with Cross-Attention Control", ICLR, 2023

Inpainting

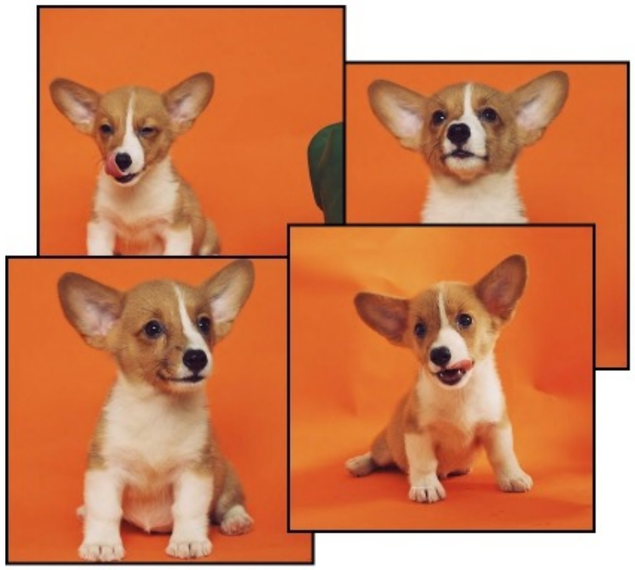
Rombach et al., “High-Resolution Image Synthesis with Latent Diffusion Models”, CVPR, 2022

Personalization

Input images

Ruiz et al, “DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”, CVPR, 2023

Personalization



Input images

in the Acropolis

swimming

in a doghouse

sleeping

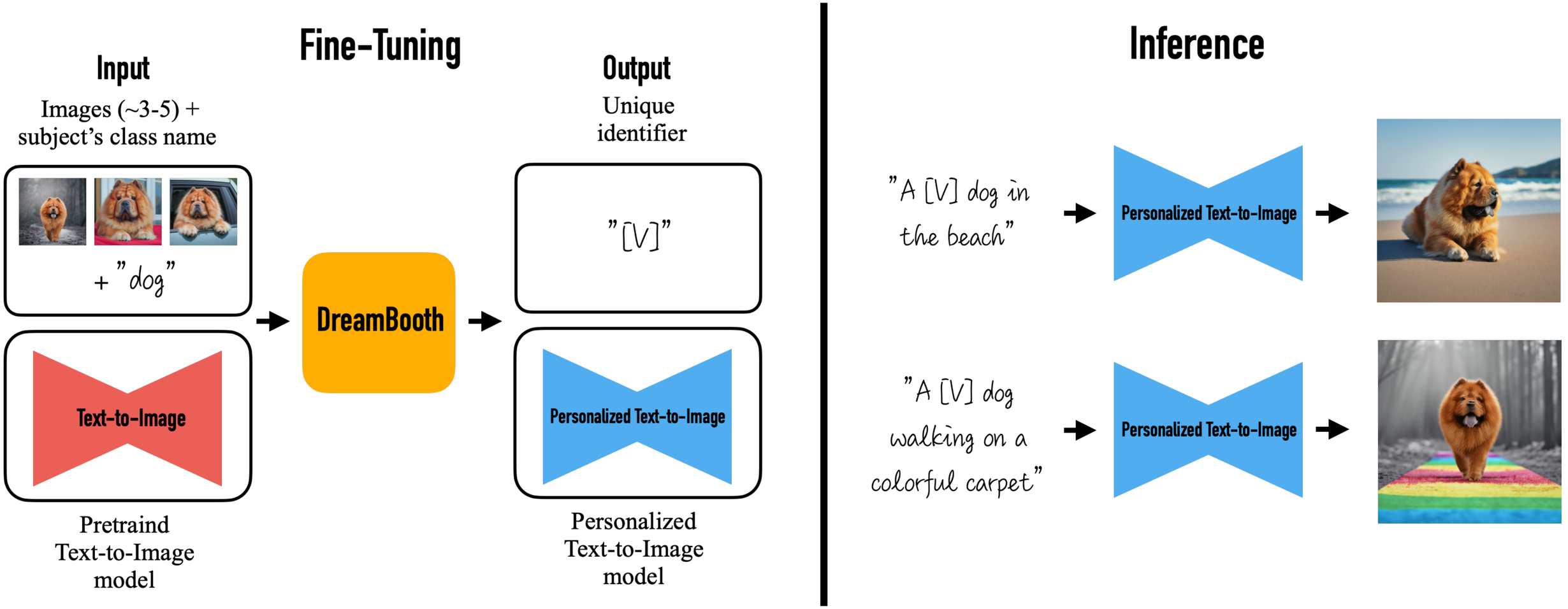
in a bucket

getting a haircut

Generated images by *personalized* diffusion model

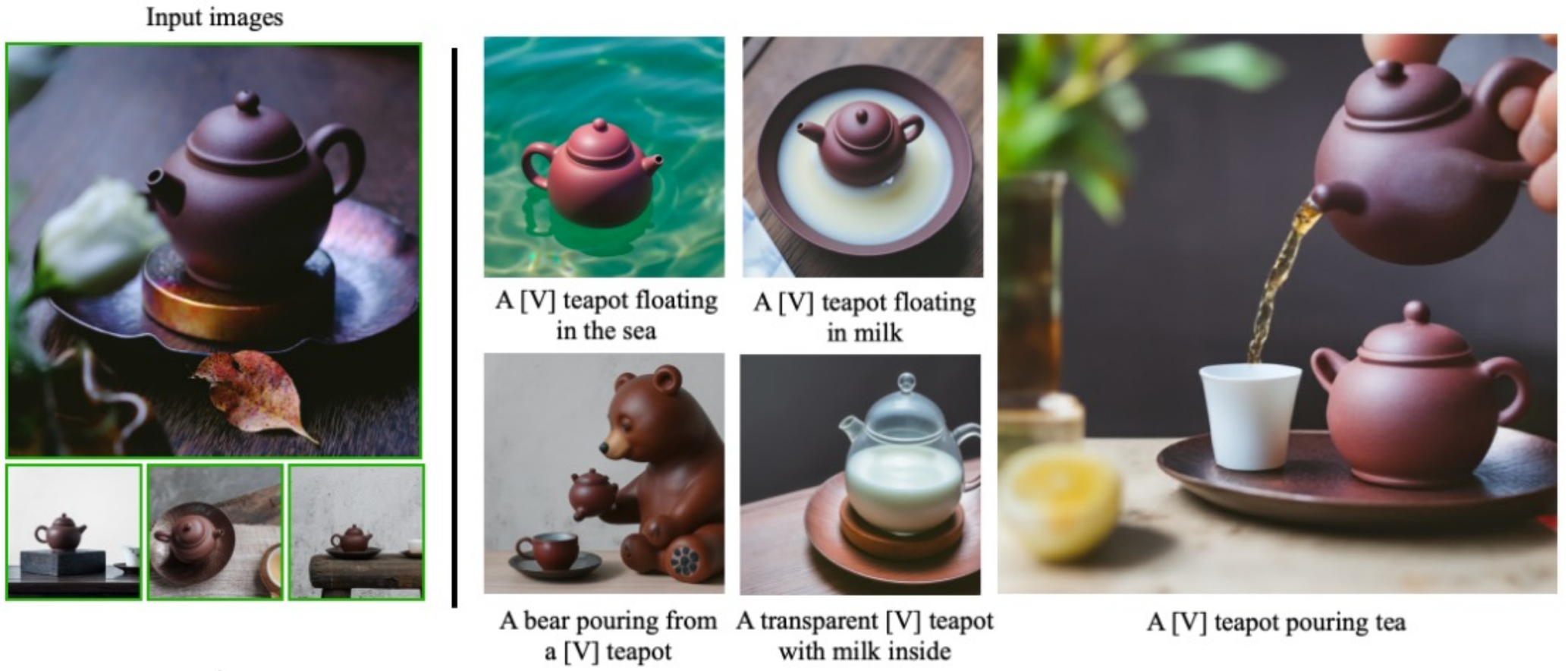
Ruiz et al, “DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”, CVPR, 2023

Personalization



Ruiz et al, "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation", CVPR, 2023

Personalization



Ruiz et al, “DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”, CVPR, 2023

Personalization

Ruiz et al, “DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”, CVPR, 2023

Personalization

Ruiz et al, “DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation”, CVPR, 2023

Why do Diffusion Models work so well?

Let’s look at other, previous generative frameworks and compare…

Generative Adversarial Networks

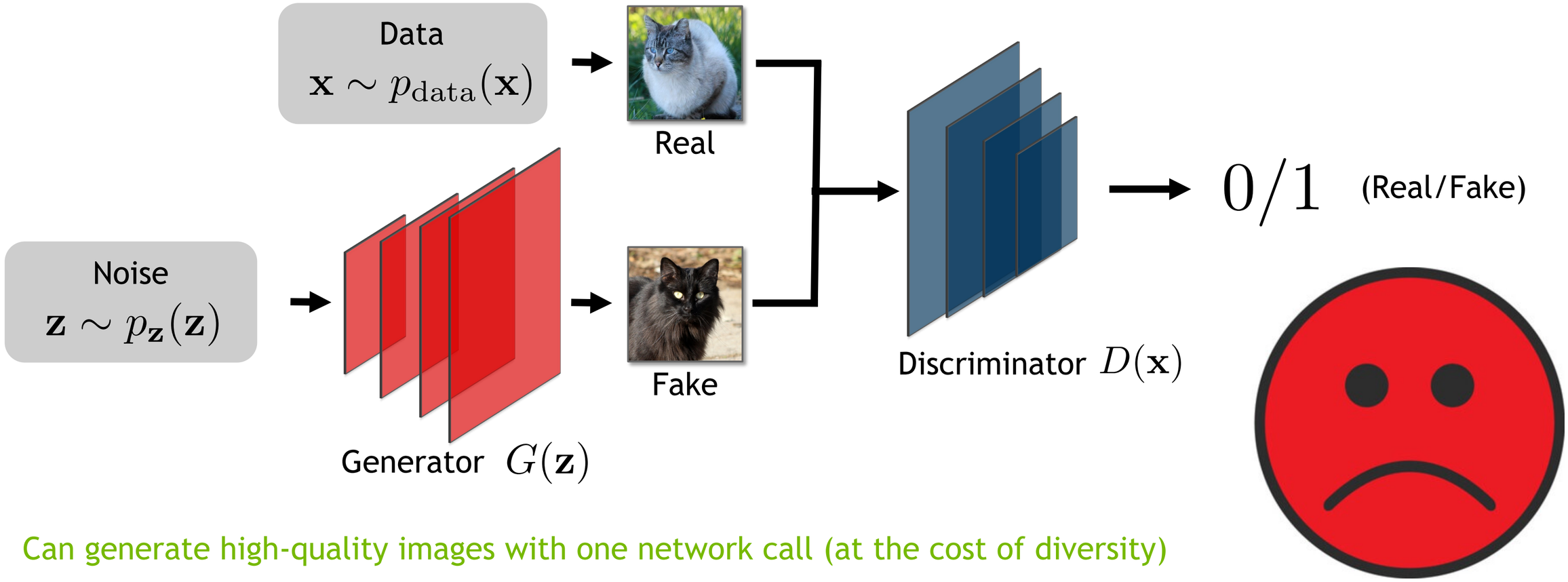
Energy-based Models

Normalizing Flows

Autoregressive Models

Variational Autoencoders

Generative Adversarial Networks



- Can generate high-quality images with one network call (at the cost of diversity)
- Adversarial training is not stable
- Not scalable to diverse data
- Prone to mode collapse

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Goodfellow et al., "Generative Adversarial Nets", NeurIPS 2023

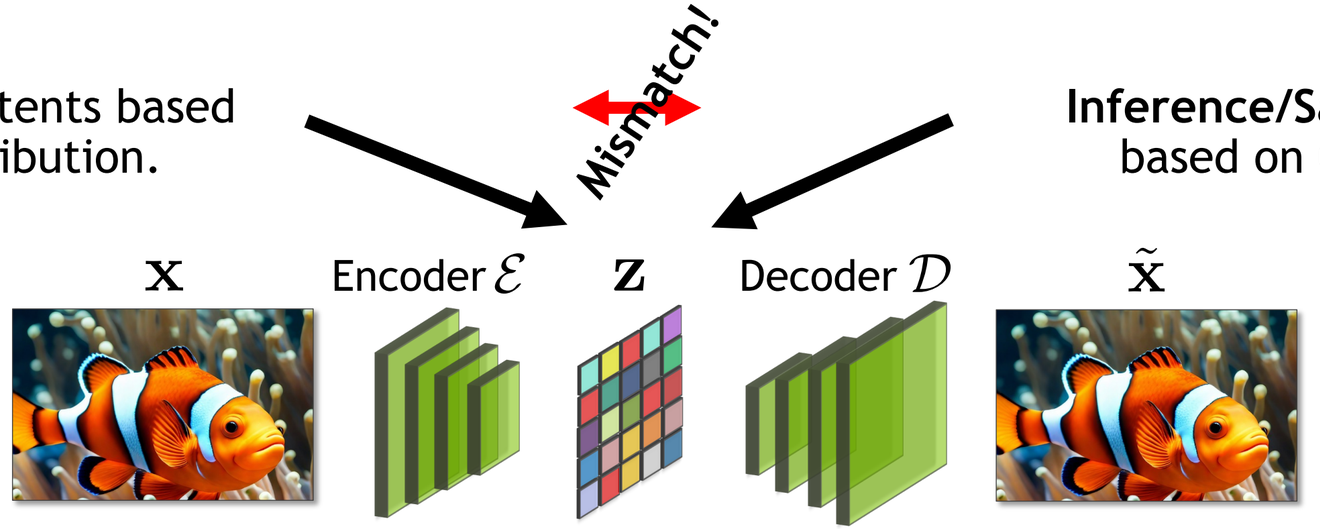
Old Likelihood-based Generative Models

Variational Autoencoders (VAEs):

- Prior hole problem: Mismatch between encoding (training) and prior (sampling) distributions.
- Very-deep VAEs are hard to train (need to backprop. through very deep model with stochastic latents).

Training: Sample latents based on encoder distribution.

Inference/Sampling: Sample latents based on Gaussian distribution.



Old Likelihood-based Generative Models

Variational Autoenco

- Prior hole problem
- Very-deep VAEs ar

Normalizing Flows (N

- Prior hole problem
- Very-deep NFs are

Training: Sample later
on encoder distribu

distributions.

with stochastic latents).

distributions.

ampling: Sample latents
Gaussian distribution.

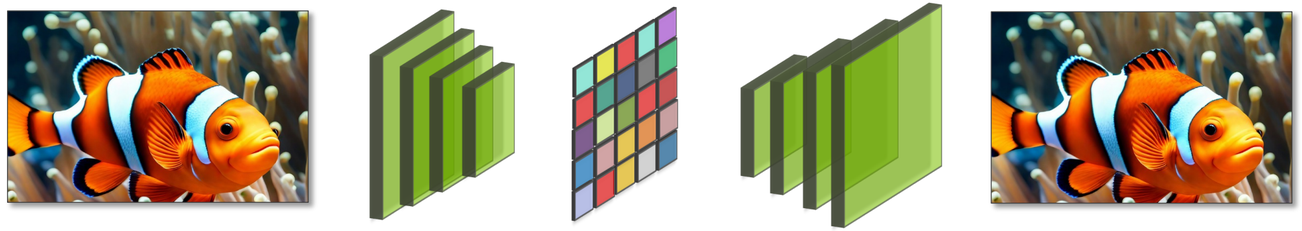
Diffusion Models:

Encoder = fixed forward diffusion that exactly diffuses towards Gaussian. No mismatch!

+

Scalable “layer-wise” training objective!

Both problems solved!



Normalizing Flows:

Same as VAE, but decoder exact inverse of encoder.

Old Likelihood-based Generative Models

Variational Autoencoders (VAEs):

- Prior hole problem: Mismatch between encoding (training) and prior (sampling) distributions.
- Very-deep VAEs are hard to train (need to backprop. through very deep model with stochastic latents).

Normalizing Flows (NFs):

- Prior hole problem: Mismatch between encoding (training) and prior (sampling) distributions.
- Very-deep NFs are hard to train (need to backprop. through very deep model).

Energy-based Models (EBMs):

- No scalable training algorithms (often cumbersome Markov Chain Monte Carlo required during training).

Autoregressive vs. Diffusion Models

Autoregressive models actually work (see language models). Let’s compare…

Autoregressive Models

- Iterative generation
- Training objective breaks complex task (sequence generation) down into scalable simple objective: *Next-token-prediction*.

  → Allows to learn extremely complex generative model.
- Ideal inductive bias for ”sequential” problems (like text).

Comes at a cost! Slow iterative generation in both autoregressive and diffusion models!

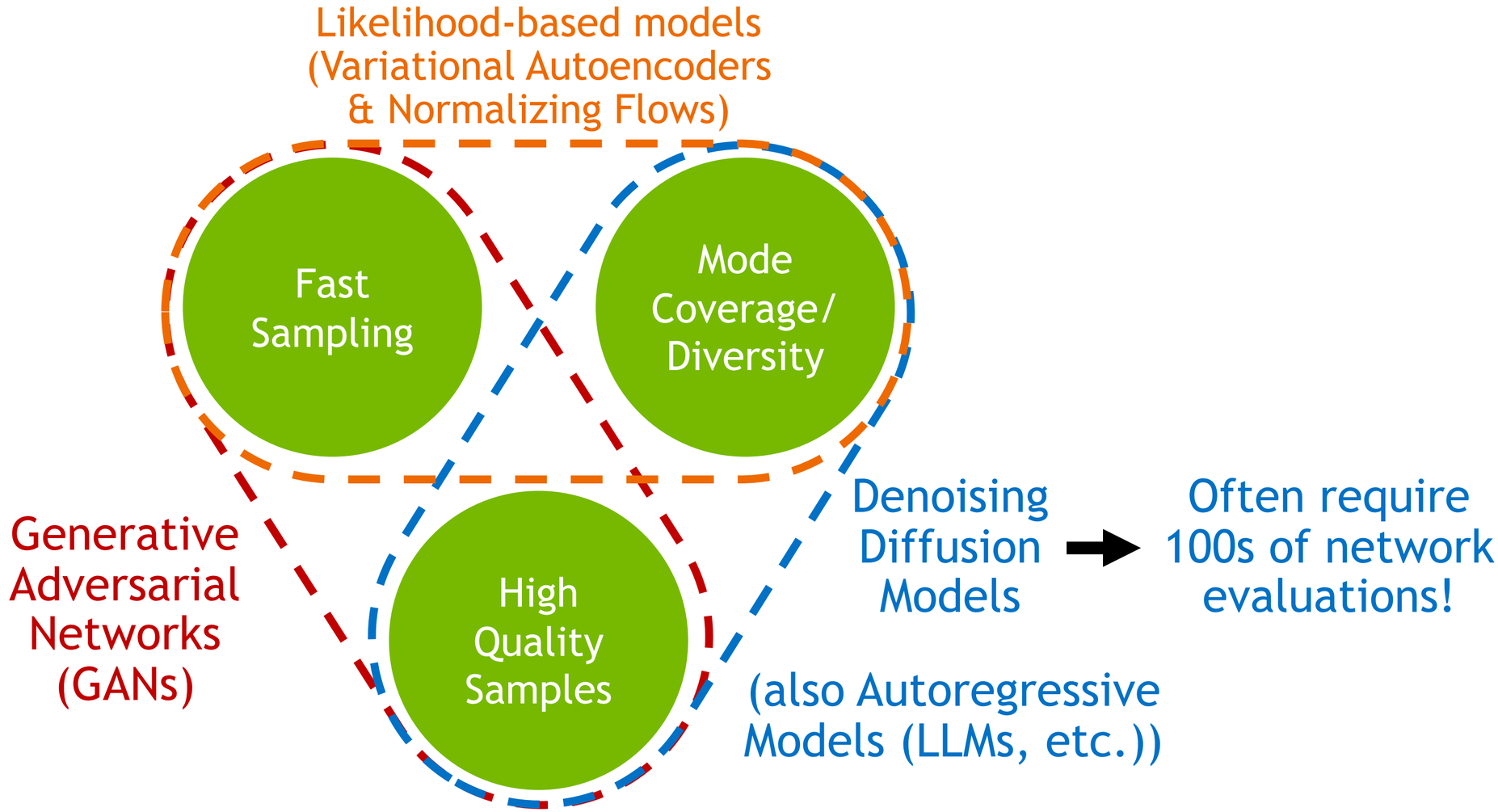
Diffusion Models

- Iterative generation
- Training objective breaks complex task (noise-to-data) down into scalable simple objective: *Small denoising step*.

  → Allows to learn extremely complex generative model.
- Ideal inductive bias for ”global coarse-to-fine” problems (like images and most visual data).
- Objective reweighting helps focus model on relevant information (crucial for images)
- Guidance extremely successful, mostly special to diffusion models.

What makes a Good Generative Model?

The Generative Learning Trilemma

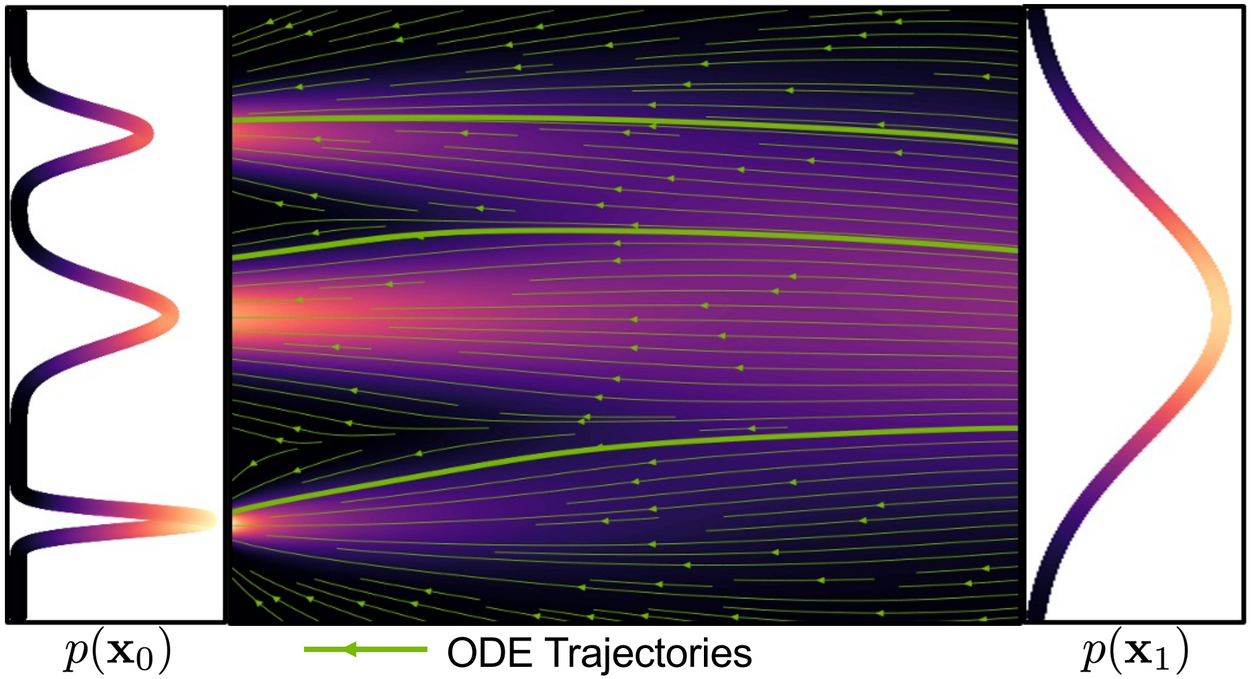


Xiao et al., ”Tackling the Generative Learning Trilemma with Denoising Diffusion GANs”, ICLR 2022

Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

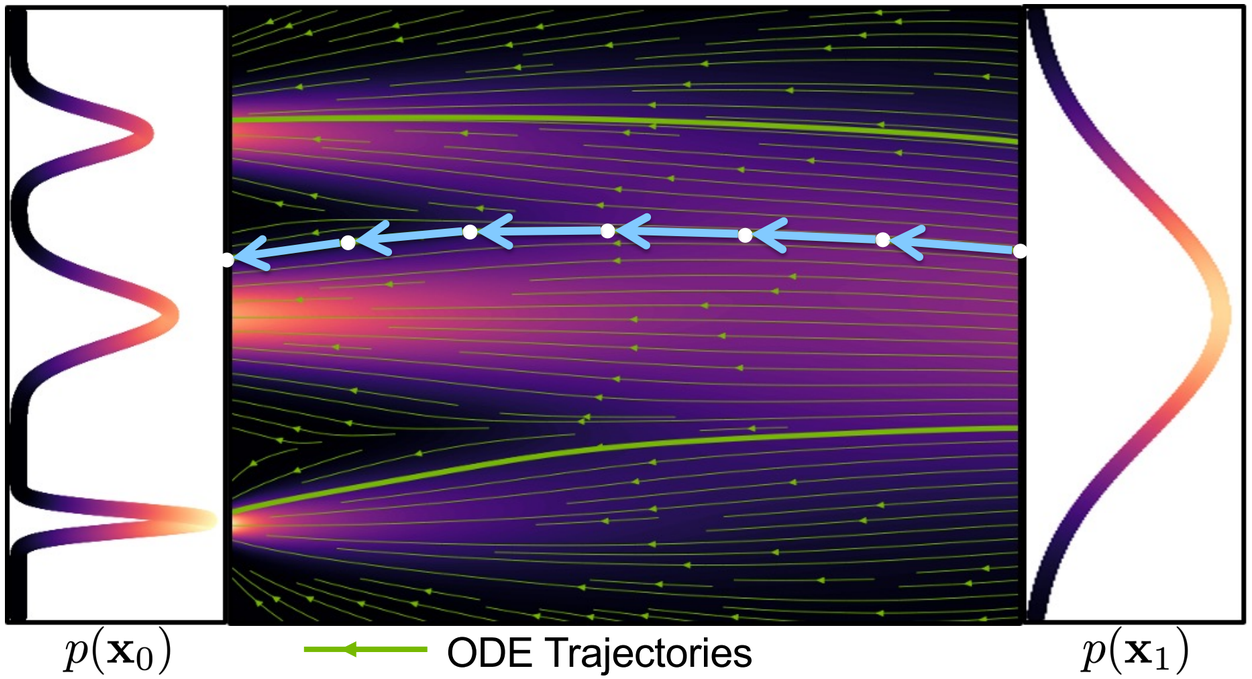
Fast samplers and solvers based on ODE/SDE literature.



Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

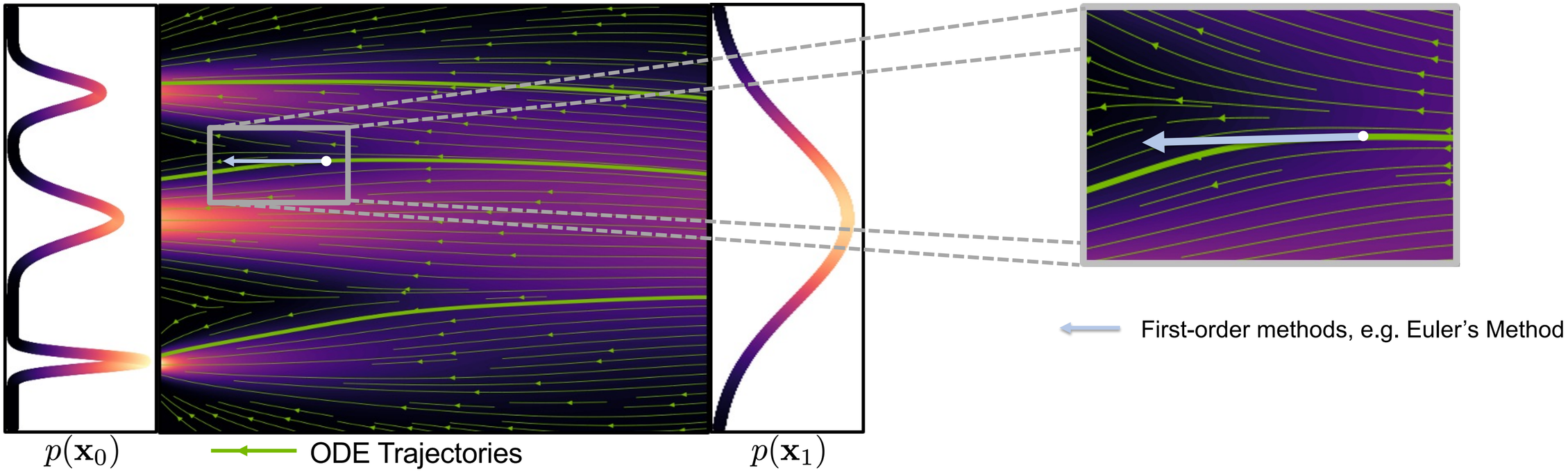
Fast samplers and solvers based on ODE/SDE literature.



Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

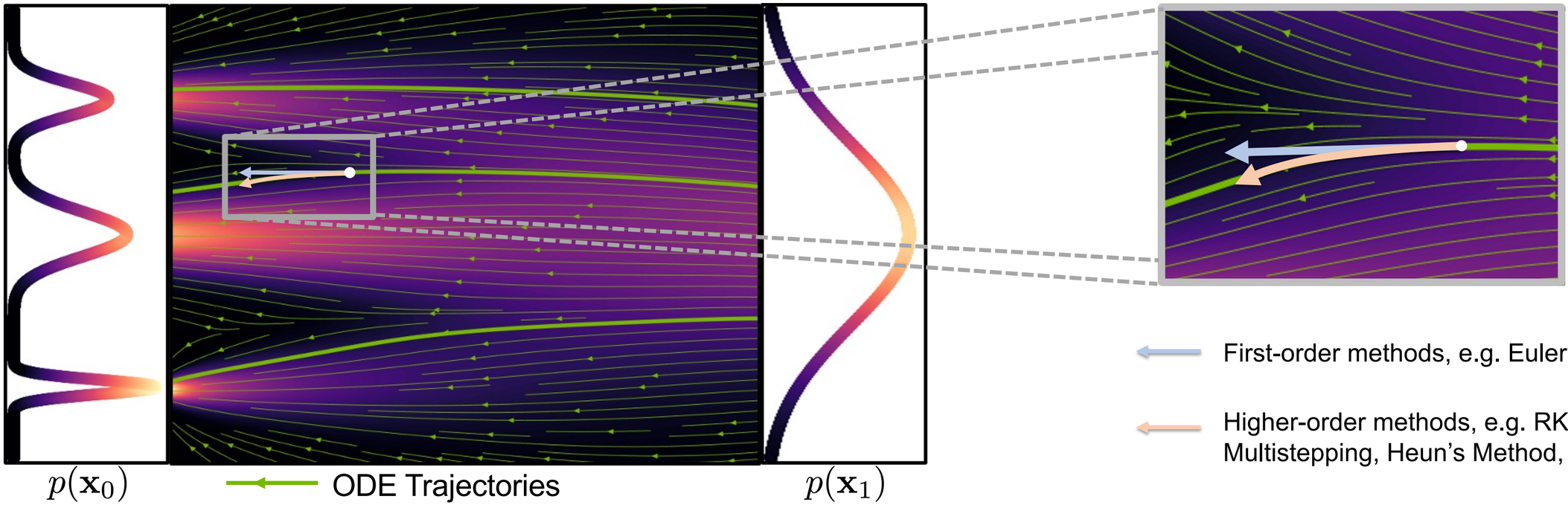
Fast samplers and solvers based on ODE/SDE literature.



Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

Fast samplers and solvers based on ODE/SDE literature.



Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

Fast samplers and solvers based on ODE/SDE literature.

DPM-Solver++(2M)
$(N = 15)$

DPM-Solver++(2M)
$(N = 20)$

DPM-Solver++(2M)
$(N = 50)$

Lu et al., ”DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models”, 2022

Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

Fast samplers and solvers based on ODE/SDE literature.

- Runge-Kutta adaptive step-size ODE solver:
  - Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, ICLR, 2021
- Higher-Order adaptive step-size SDE solver:
  - Jolicoeur-Martineau et al., “Gotta Go Fast When Generating Data with Score-Based Models”, arXiv, 2021
- Reparametrized, smoother ODE:
  - Song et al., “Denoising Diffusion Implicit Models”, ICLR, 2021
  - Zhang et al., "gDDIM: Generalized denoising diffusion implicit models", arXiv 2022
- Higher-Order ODE solver with linear multistepping:
  - Liu et al., “Pseudo Numerical Methods for Diffusion Models on Manifolds”, ICLR, 2022
- Exponential ODE Integrators:
  - Zhang and Chen, “Fast Sampling of Diffusion Models with Exponential Integrator”, arXiv, 2022
  - Lu et al., “DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps”, NeurIPS, 2022
  - Lu et al., "DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models", NeurIPS 2022
- Higher-Order ODE solver with Heun’s Method:
  - Karras et al., “Elucidating the Design Space of Diffusion-Based Generative Models”, NeurIPS, 2022
- Many more:
  - Zhao et al., "UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models", arXiv 2023
  - Shih et al., "Parallel Sampling of Diffusion Models", arxiv 2023
  - Chen et al., "A Geometric Perspective on Diffusion Models", arXiv 2023

Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

Distillation into few-step or single-step generators.

Stable Diffusion
(50 steps, 2590 ms)

Distribution Matching
Distillation (1 step, 90 ms)

Stable Diffusion
(50 steps, 2590 ms)

Distribution Matching
Distillation (1 step, 90 ms)

Yin et al., ” One-step Diffusion with Distribution Matching Distillation”, CVPR, 2024

Fast Sampling and Diffusion Model Distillation

Diffusion Models are slow due to their iterative sampling process!

Distillation into few-step or single-step generators.

- Distillation:
  - Luhman and Luhman, “Knowledge Distillation in Iterative Generative Models for Improved Sampling Speed”, arXiv, 2021
  - Salimans and Ho, “Progressive Distillation for Fast Sampling of Diffusion Models”, ICLR, 2022
  - Meng et al., “On Distillation of Guided Diffusion Models”, CVPR, 2023
  - Gu et al., “BOOT: Data-free Distillation of Denoising Diffusion Models with Bootstrapping”, arXiv, 2023
  - Kohler et al., “Imagine Flash: Accelerating Emu Diffusion Models with Backward Distillation”, 2024
- Consistency Models:
  - Song et al., “Consistency Models”, ICML, 2023
  - Luo et al., “Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference”, arXiv, 2023
  - Kim et al., “Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion”, ICLR, 2024
  - Heek et al., “Multistep Consistency Models”, arXiv, 2024
- Adversarial Diffusion Distillation:
  - Sauer et al., “Adversarial Diffusion Distillation”, arXiv, 2023
  - Sauer et al., “Fast High-Resolution Image Synthesis with Latent Adversarial Diffusion Distillation”, arXiv, 2024
  - Yin et al., “One-step Diffusion with Distribution Matching Distillation”, CVPR, 2024
- Rectified Flow:
  - Liu et al., “Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow”, ICLR, 2023
  - Liu et al., “InstaFlow: One Step is Enough for High-Quality Diffusion-Based Text-to-Image Generation”, ICLR, 2024

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*



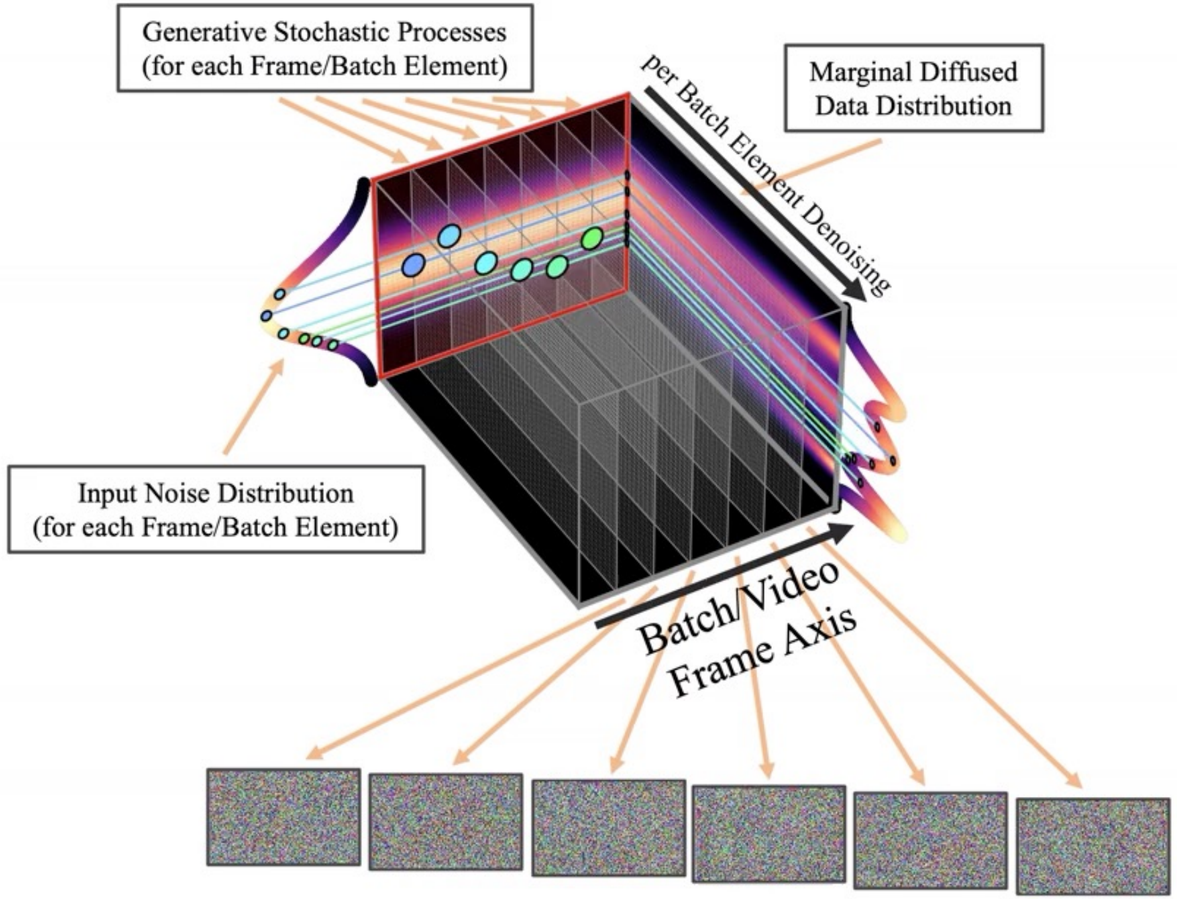
Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR* 2023
Emu Video, https://emu-video.metademolab.com/

”A stylish woman walks down a Tokyo street filled with warm glowing neon and animated city signage. She wears a black leather jacket, a long red dress, and black boots, and carries a black purse. She wears sunglasses and red lipstick. She walks confidently and casually. The street is damp and reflective, creating a mirror effect of the colorful lights. Many pedestrians walk about.”

Brooks, Peebles, et al. https://openai.com/sora

From Image to Video Diffusion Models

Common Idea: Temporally Align an Image Diffusion Model via Video Fine-tuning



Before temporal video fine-tuning,
different batch samples are independent.

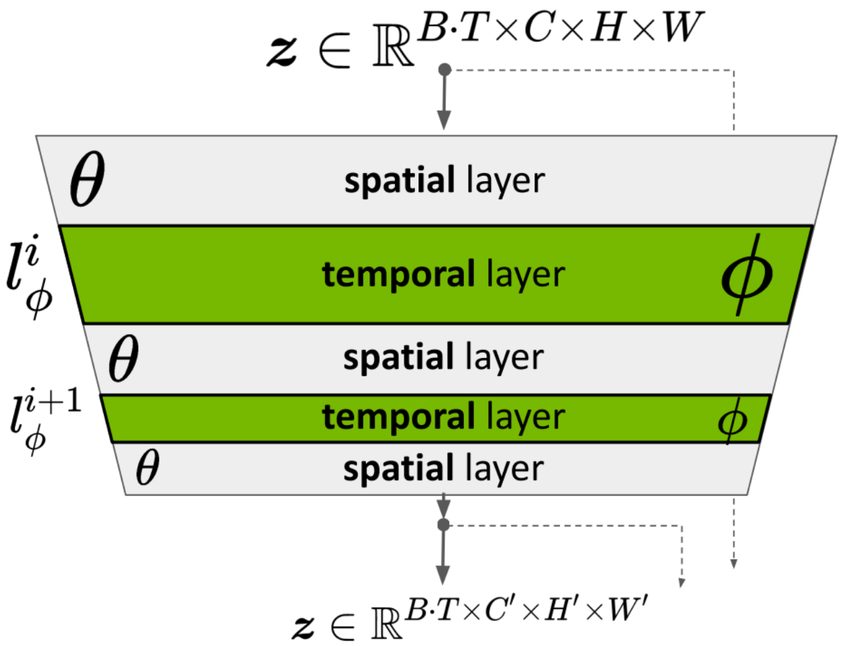
From Image to Video Diffusion Models

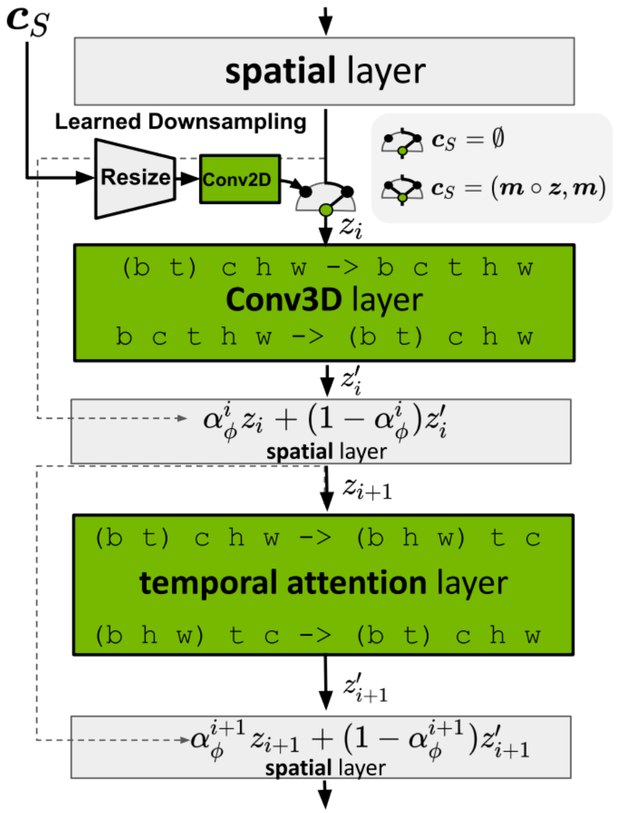
Common Idea: Temporally Align an Image Diffusion Model via Video Fine-tuning

- **Spatial layers** interpret frames as independent images.
- **Temporal layers** interpret frames as sequence and model temporal dynamics.
- Implemented by shifting temporal axis into batch dimension for spatial layers.

1. Can train both spatial and temporal layers jointly, using both image and video datasets.

2. Or can fix pre-trained spatial layers and only train temporal layers with video data.

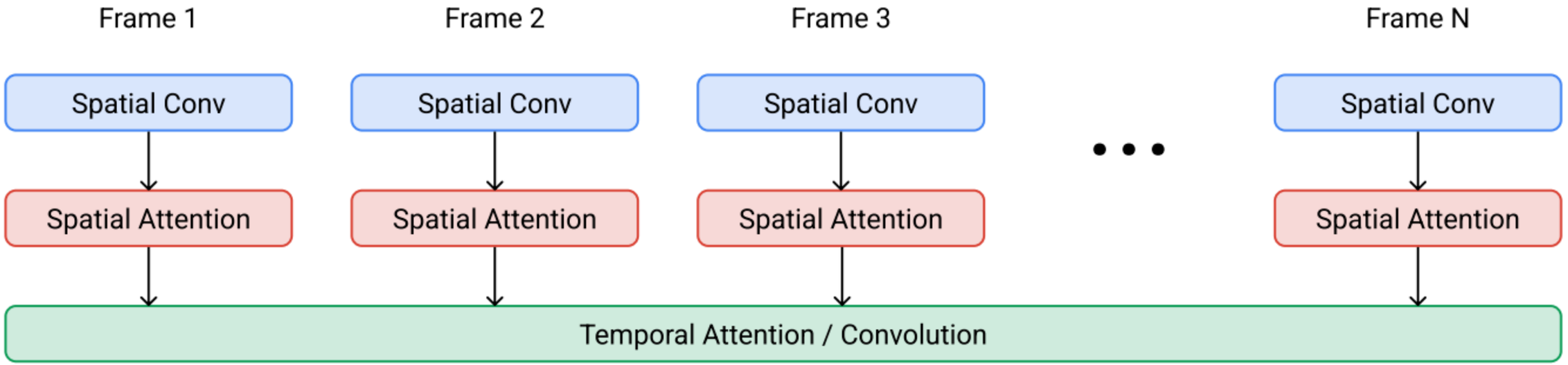




Blattmann et al., "Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models", *CVPR, 2023*

From Image to Video Diffusion Models

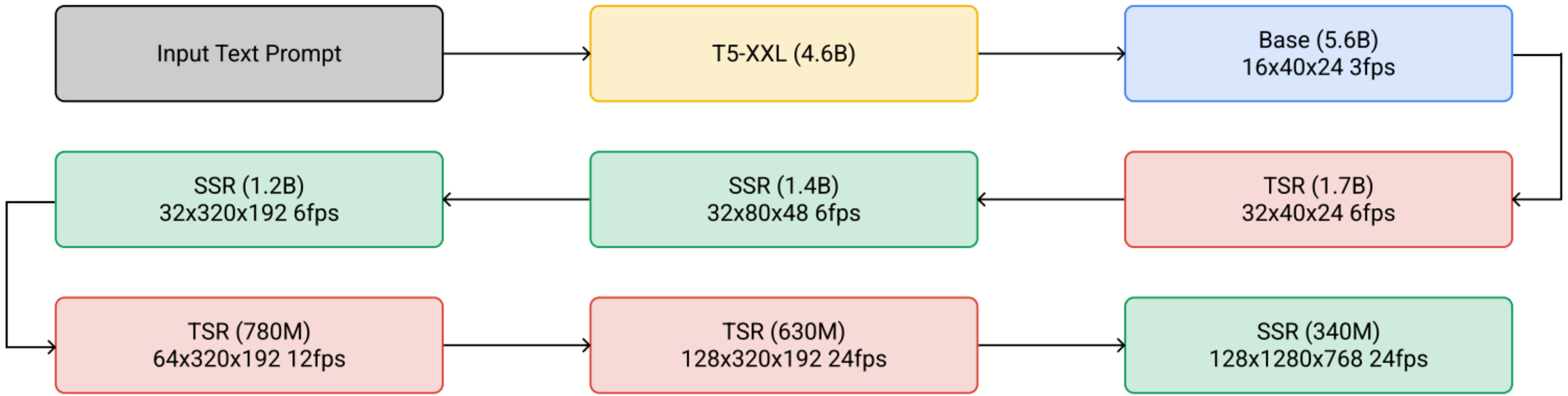
Common Idea: Temporally Align an Image Diffusion Model via Video Fine-tuning



Ho et al., “Imagen Video: High Definition Video Generation with Diffusion Models”, 2022

From Image to Video Diffusion Models

Spatial and Temporal Cascaded Diffusion Models



Ho et al., “Imagen Video: High Definition Video Generation with Diffusion Models”, 2022

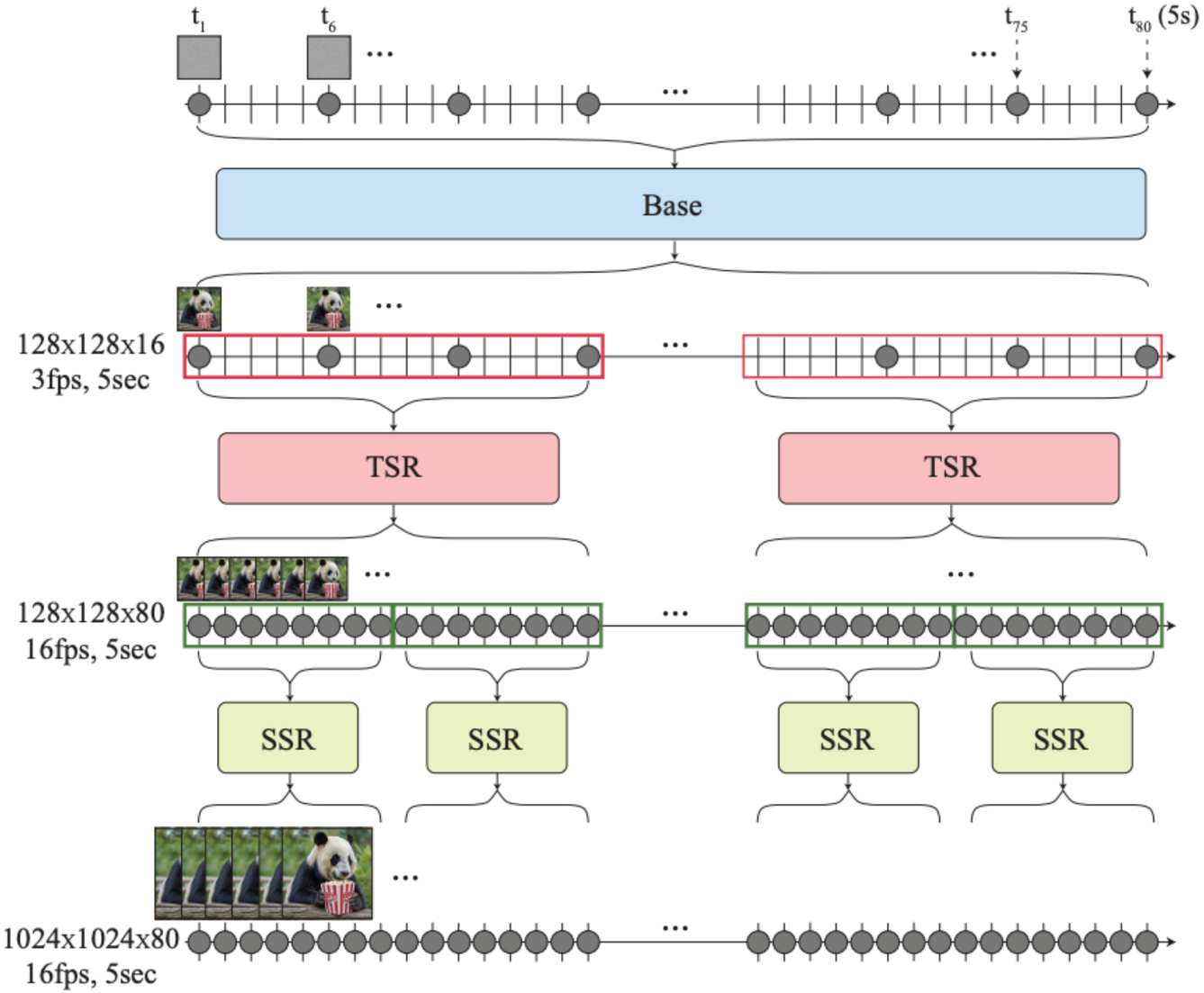
From Image to Video Diffusion Models

Spatial and Temporal Cascaded Diffusion Models

Cascade common for longer generation.

Typical stepwise approach:

1. Generate sparse key frames
2. Several rounds of temporal superresolution to generate in-between frames and get high fps, conditioned on keyframes
3. Upscale final video



Bar-Tal et al., “Lumiere: A Space-Time Diffusion Model for Video Generation”, 2024

Video Latent Diffusion Models

Text-to-Video

- Stable Diffusion Text-to-Image base LDM &
  Stable Diffusion 4x Text-to-Image Upscaler
- Video Training on WebVid10M
- 1280 x 2048 resolution, 113 frames, 4.7s, 24fps
- 2.7B trained parameter (4.1B in total)

“A teddy bear is playing the electric guitar, high definition, 4k.”

Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR, 2023*

Video Latent Diffusion Models

Text-to-Video

“A storm trooper vacuuming the beach.”

“Two pandas discussing an academic paper.”

Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR, 2023*

Video Latent Diffusion Models

Text-to-Video

“Close up of grapes on a rotating table. High definition.”

“Sunset time lapse at the beach with moving clouds and colors in the sky, 4k, high resolution.”

Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR, 2023*

Video Latent Diffusion Models

Real in-the-wild Driving Scene Video Generation

Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR, 2023*

Stable Video Diffusion

- Larger and carefully curated dataset and careful final fine-tuning on high-quality data
- End-to-end training including image backbone

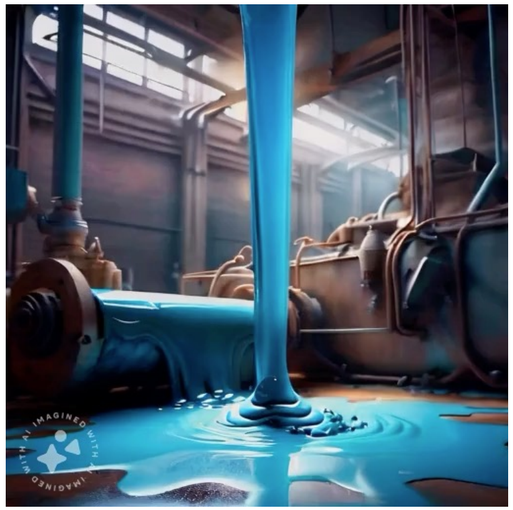
Blattmann et al., “Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets”, 2023

Emu Video

Factorized generation protocol:

1. Text-to-image

2. Text-&-image-to-video



Girdhar et al., “Emu Video: Factorizing Text-to-Video Generation by Explicit Image Conditioning”, 2023

Lumiere

Main technical contribution: Space-Time U-Net

- When inflating text-to-image U-Net do both spatial and temporal up- and down-sampling for more efficient temporal processing and longer-sequence modeling
- (pixel space video diffusion)

Bar-Tal et al., “Lumiere: A Space-Time Diffusion Model for Video Generation”, 2024

The camera directly faces colorful buildings in Burano Italy. An adorable dalmation looks through a window on a building on the ground floor. Many people are walking and cycling along the canal streets in front of the buildings.”

Brooks, Peebles, et al. https://openai.com/sora, 2024

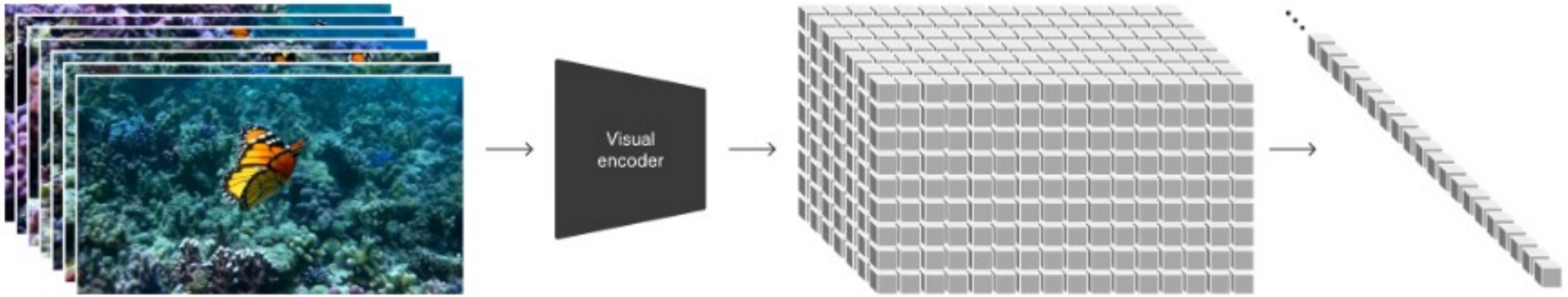
”An extreme close-up of an gray-haired man with a beard in his 60s, he is deep in thought pondering the history of the universe as he sits at a cafe in Paris, his eyes focus on people offscreen as they walk as he sits mostly motionless, he is dressed in a wool coat suit coat with a button-down shirt , he wears a brown beret and glasses and has a very professorial appearance, and the end he offers a subtle closed-mouth smile as if he found the answer to the mystery of life, the lighting is very cinematic with the golden light and the Parisian streets and city in the background, depth of field, cinematic 35mm film.”

Brooks, Peebles, et al. https://openai.com/sora, 2024

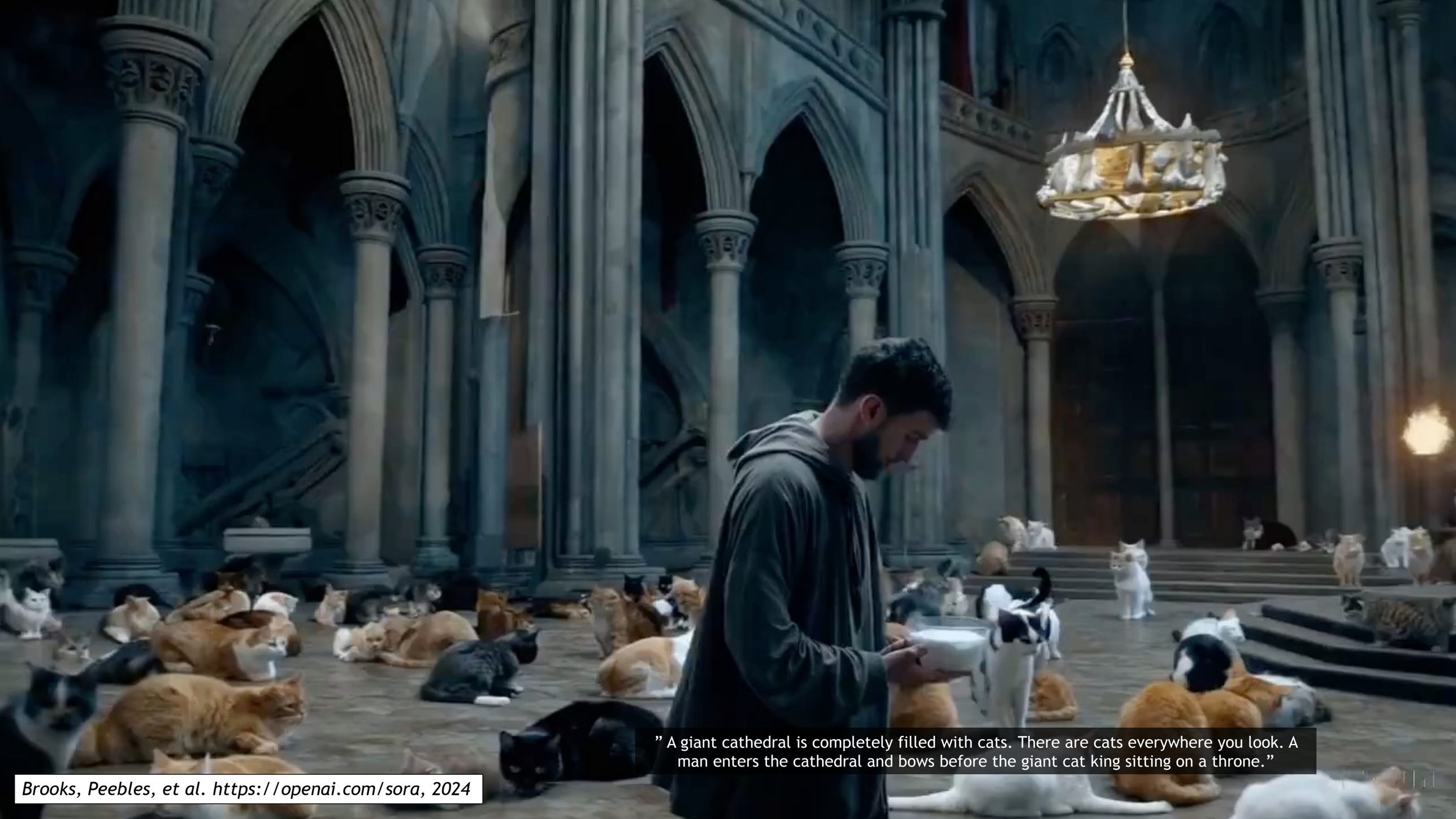
Sora

State-of-The-Art Text-to-Video Diffusion Model

- Spatio-temporal autoencoder
- Patch embeddings in latent space (like in vision transformers, but in latent space; similar to tokens in LLMs)
- Diffusion model over patches
- Neural Network likely based on large transformer (similar technology as in LLMs)
- Trained on much more video data

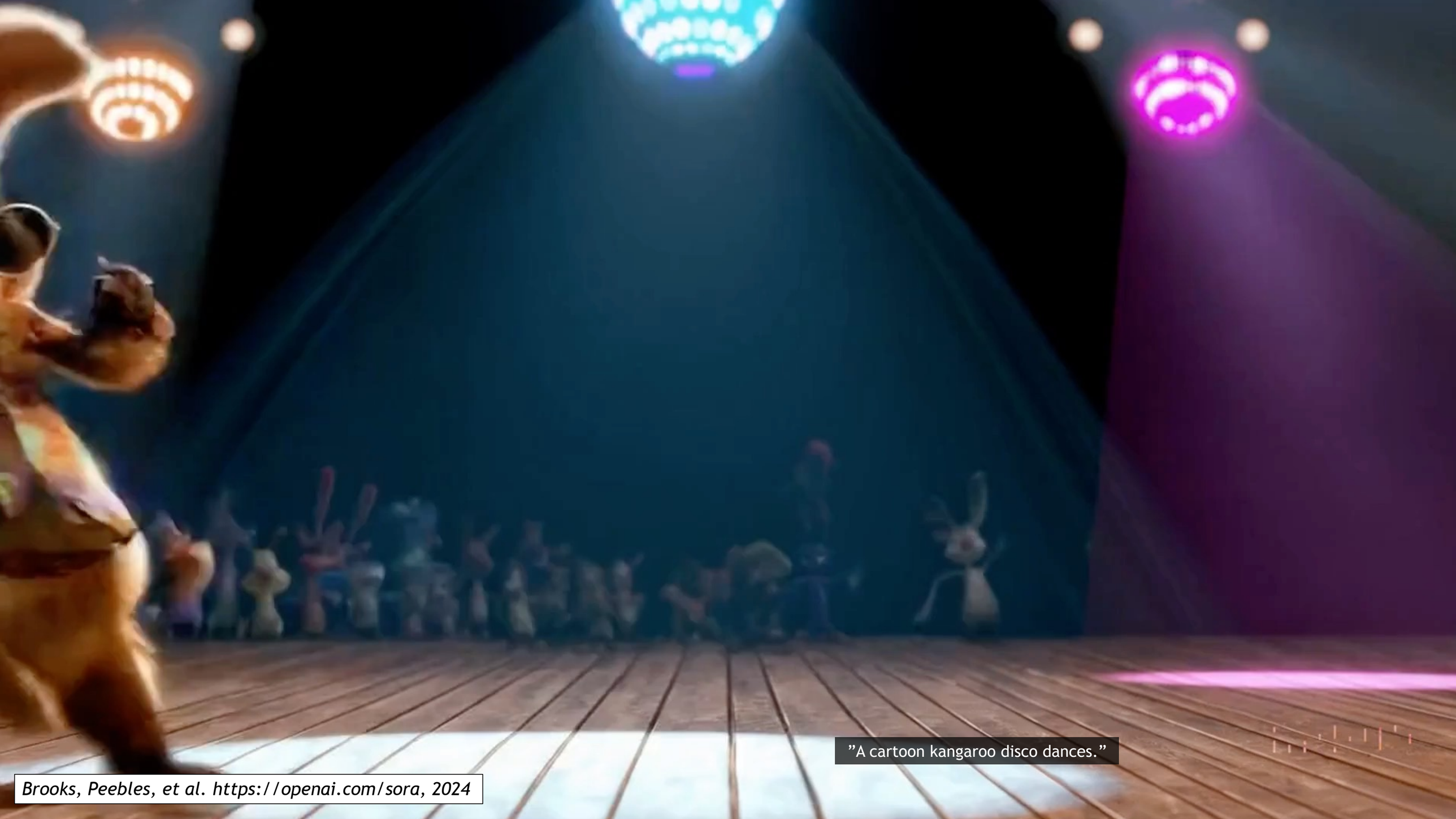


Brooks, Peebles, et al. https://openai.com/research/video-generation-models-as-world-simulators



” A giant cathedral is completely filled with cats. There are cats everywhere you look. A man enters the cathedral and bows before the giant cat king sitting on a throne.”

Brooks, Peebles, et al. https://openai.com/sora, 2024



Brooks, Peebles, et al. https://openai.com/sora, 2024

”Photorealistic closeup video of two pirate ships battling each other as they sail inside a cup of coffee.”

Brooks, Peebles, et al. https://openai.com/sora, 2024

Overview

1. **History:** From the Beginnings of Image Generation until Today
2. **Image Generation with Diffusion Models**
   - *Fundamentals:* Introduction to Diffusion Models
   - *Architectures, Pipelines and Tricks:* Building Diffusion Models in Practice
   - *Results:* Image Generation and Image Processing
   - *Framework Comparisons:* What makes Diffusion Models work so well? How are they different?
3. **Video Diffusion Models**
4. **3D and 4D Generation:** *From 2D to 3D & 4D with Score Distillation*

3D Diffusion Models?

- Large-scale image diffusion models require 100's million to billions of text-caption pairs for training.
  - ➞ Internet

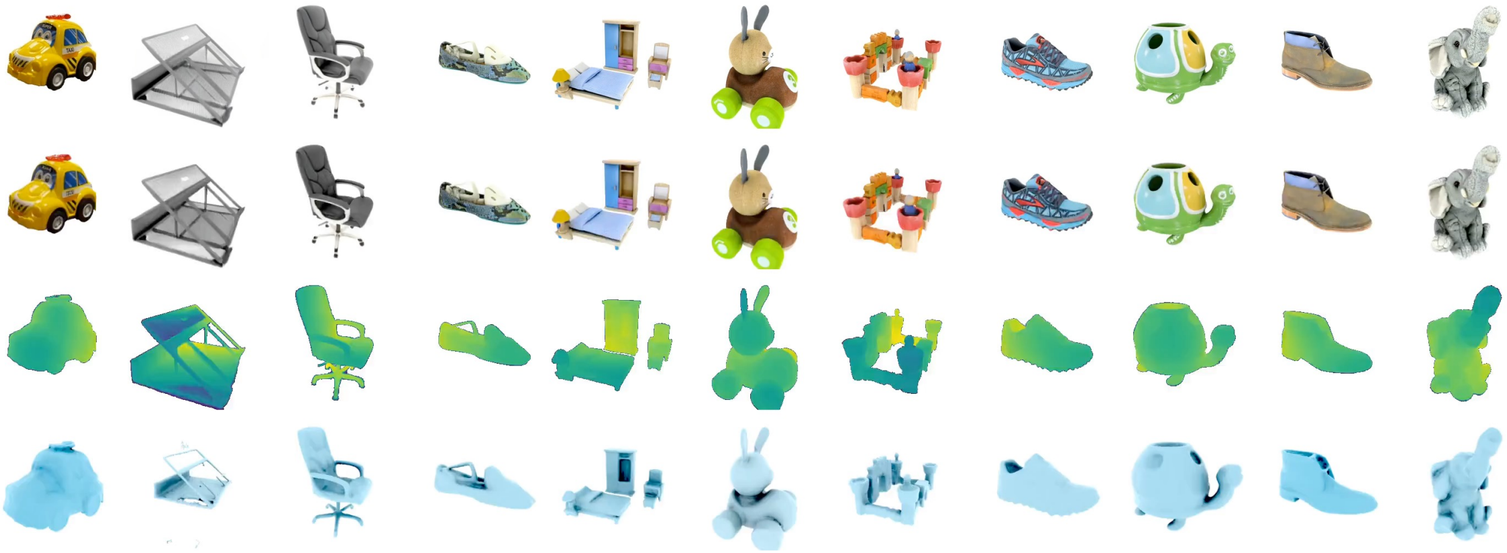
- 3D objects? Scarce. Objaverse-XL ~ 10M objects only.
  - ➞ Objaverse-XL ~ 10M objects only.

But can we also use large-scale text-to-image models for 3D generation?

↓

Score Distillation Sampling

We can train 3D generative models on this data…



Xu et al., "DMV3D: Denoising Multi-View Diffusion using 3D Large Reconstruction Model", 2023

Score Distillation Sampling

1. 3D object, parametrized through learnable 3D representation (radiance field, mesh, 3D Gaussians, etc.)

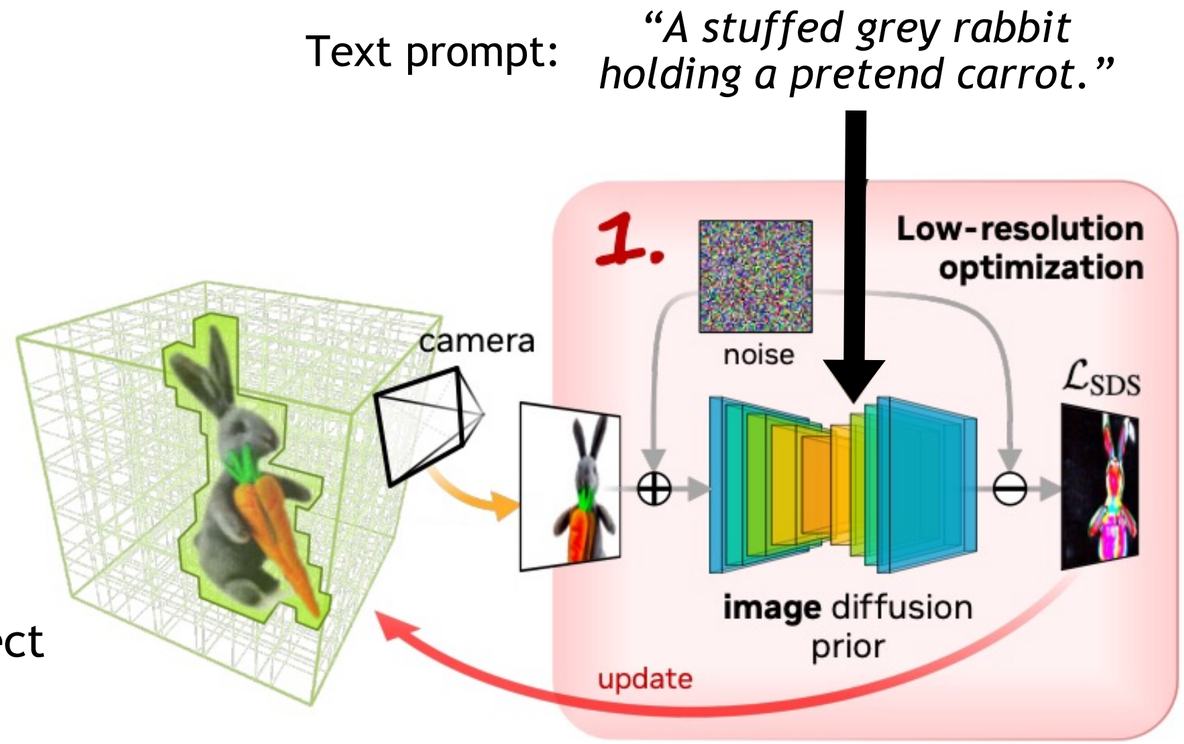
2. (Differentiably) render 3D object from different camera perspectives.

3. Provide 2D renderings to pre-trained large-scale text-to-image diffusion model.

4. Diffusion model has learnt what a good image of the object looks like for the text prompt ➡ feedback/gradient.

5. Backprop diffusion model gradient back into 3D representation to make it look realistic from all camera directions.

6. If it looks good from all directions, it is likely 3D consistent, too!



Poole et al., "DreamFusion: Text-to-3D using 2D Diffusion", ICLR 2023
Lin et al., "Magic3D: High-Resolution Text-to-3D Content Creation", CVPR 2023

Score Distillation Sampling

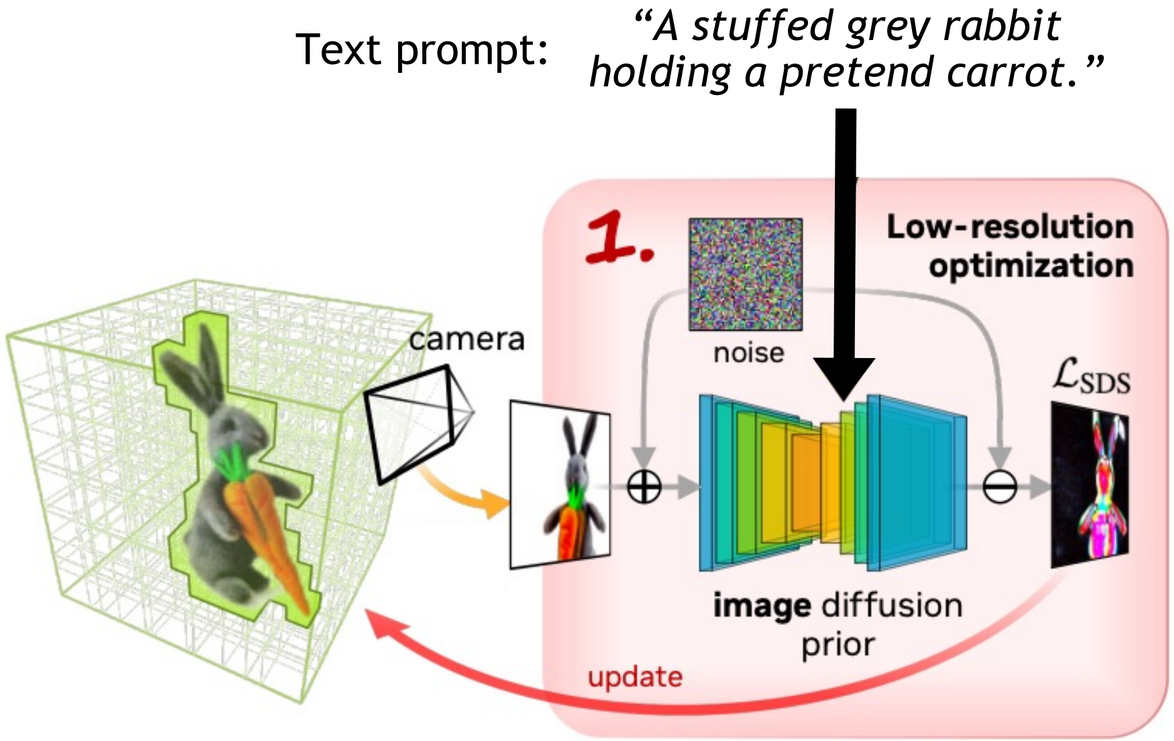
Formally, minimize reverse Kullback-Leibler divergence from rendering distribution to diffusion model distribution by optimizing 3D representation:

$$\nabla_{\boldsymbol{\theta}} \mathrm{KL}\left(q_{\boldsymbol{\theta}}(\mathbf{z}_t) || p_{\boldsymbol{\phi}}(\mathbf{z}_t | \mathrm{prompt})\right)$$

$\mathbf{x} = g(\boldsymbol{\theta})$ — Diff. rendering function g of 3D rep $\boldsymbol{\theta}$

$\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ — Diffusion

$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})$ — Noise



➡ Score Distillation gradient:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{SDS}}(\mathbf{x} = g(\boldsymbol{\theta})) = \mathbb{E}_{t,\boldsymbol{\epsilon}} \left[w(t) \left(\hat{\boldsymbol{\epsilon}}_{\phi}(\mathbf{z}_t, v, t) - \boldsymbol{\epsilon} \right) \frac{\partial \mathbf{x}}{\partial \boldsymbol{\theta}} \right]$$

Poole et al., "DreamFusion: Text-to-3D using 2D Diffusion", ICLR 2023
Lin et al., "Magic3D: High-Resolution Text-to-3D Content Creation", CVPR 2023

Text-to-3D with Score Distillation Sampling

Score Distillation Sampling with **video diffusion models**?

Generate **moving** & **dynamic 3D objects**?

Text-to-4D generation?

a beautiful dress made out of garbage bags

an imperial state crown of england

a blue poison-dart frog sitting on a water lily

neuschwanstein castle, aerial view

Lin et al., “Magic3D: High-Resolution Text-to-3D Content Creation”, CVPR 2023

Text-to-4D

“A corgi is running fast.”

“A llama running fast.”

“A bee fluttering its wings fast.”

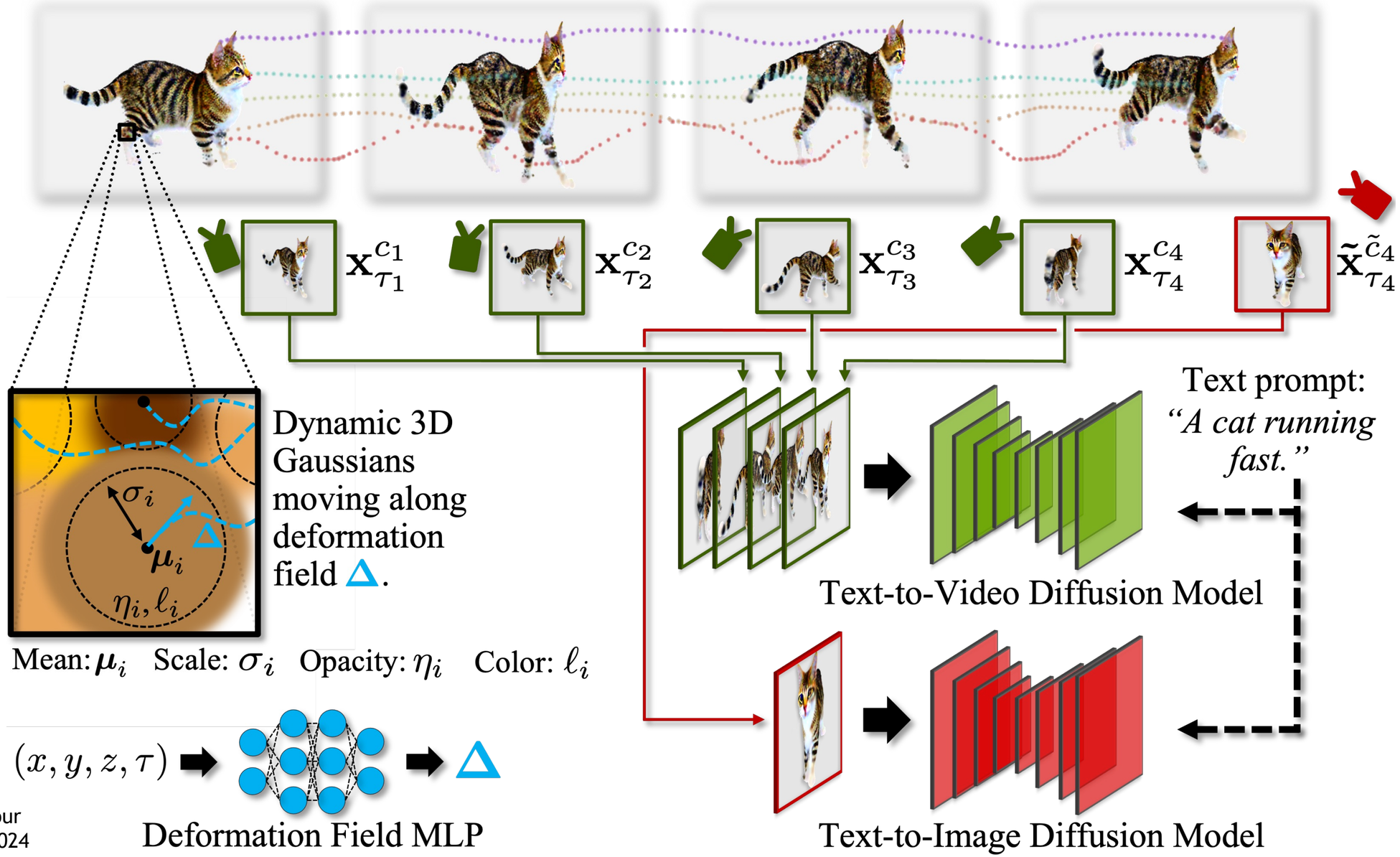
“A panda dancing.”

“Clown fish swimming.”

“A turtle swimming.”

Ling et al., “Align Your Gaussians”, CVPR, 2024

Text-to-4D with Dynamic 3D Gaussians and Composed Diffusion Models



Ling et al., "Align Your Gaussians", CVPR, 2024

Text-to-4D

“A monkey is playing bass.”

“A dog wearing a Superhero outfit with red cape flying through the sky.”

“A squirrel playing on a swing set.”

“A panda surfing a wave.”

“A cat singing.”

Ling et al., “Align Your Gaussians”, CVPR, 2024

Text-to-4D

“Volcano eruption.”

“Assassin with sword running fast.”

“Waves crashing against a lighthouse.”

“Beer pouring into a glass.”

“Wood on fire.”

Ling et al., “Align Your Gaussians”, CVPR, 2024

Text-to-4D

“Flying dragon on fire.”

“Poseidon holding his trident emerging from the sea.”

“An astronaut is playing the electric guitar.”

“A purple unicorn flying.”

“A storm trooper walking forward and vacuuming.”

Ling et al., “Align Your Gaussians”, CVPR, 2024

Autoregressively Extended Generation with Changing Prompts

“Running.” ➡ “Walking.” ➡ “Dancing.”

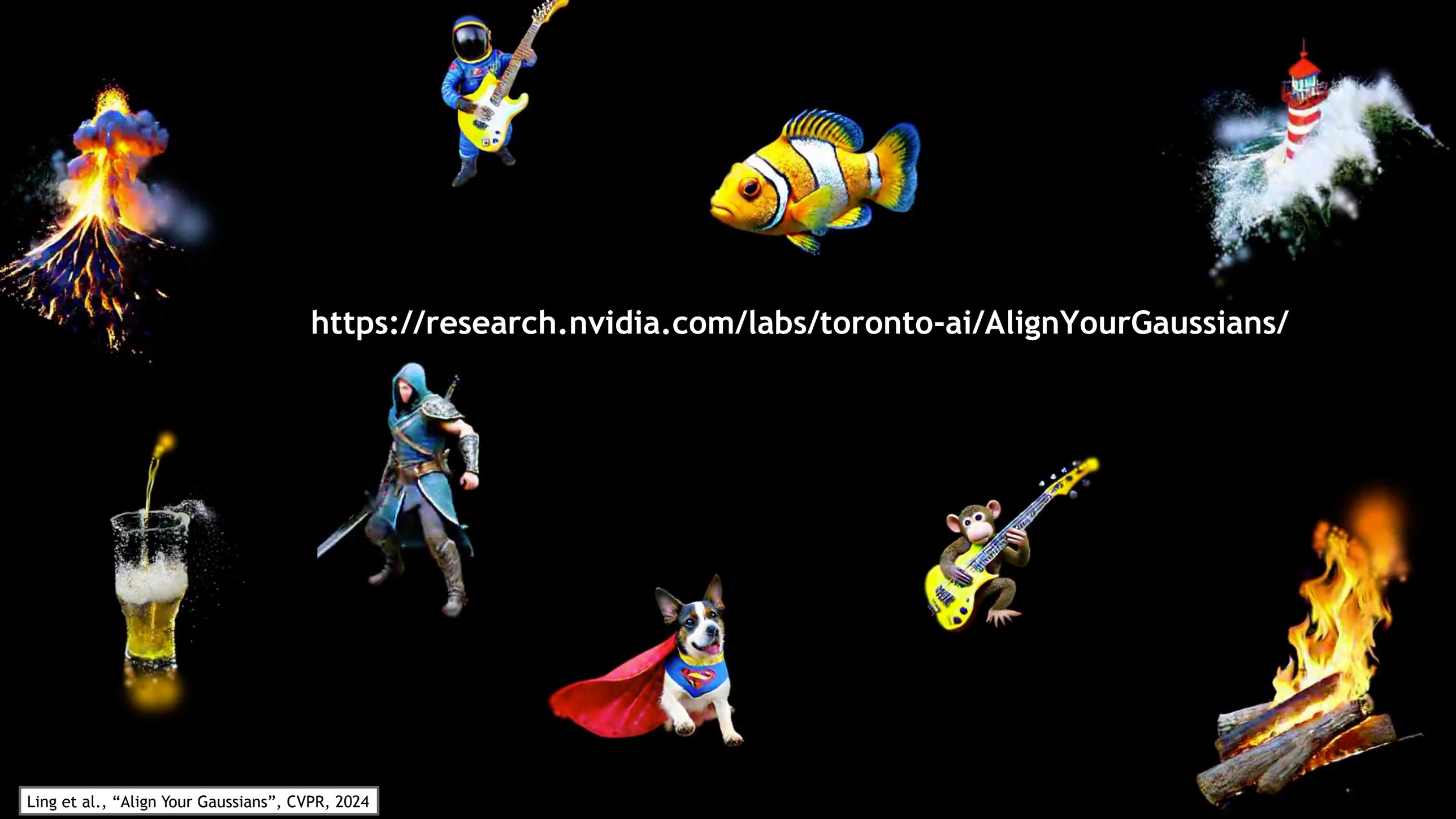
Ling et al., “Align Your Gaussians”, CVPR, 2024

Composing Dynamic 4D Assets in Scenes

Ling et al., “Align Your Gaussians”, CVPR, 2024

Composing Dynamic 4D Assets in Scenes

Ling et al., “Align Your Gaussians”, CVPR, 2024



Ling et al., “Align Your Gaussians”, CVPR, 2024

Key Take-Aways

- **Rapid Progress:** Generative AI for Vision advancing quickly. And will continue to do so!
- **Diffusion Models:** Robust and scalable generative modeling framework.
  - **Generation by Denoising:** Diffusion models synthesize data based on iterative denoising, inverting a fixed forward diffusion process.
  - **Scalability:** Modern neural network architectures + data + compute scale well.
  - **Latent Diffusion Models:** Powerful framework combining efficiency, flexibility and high expressivity.
- **Image Generation:** Very mature, with lots of applications (editing, control, personalization, ...)
- **Video Generation:** Developing rapidly. Currently, first photo-realistic models emerging.
- **3D and 4D Generation:** Not yet as robust and scalable. But promising and catching up quickly, powered by advances in image and video models (score distillation).

Many Collaborators – Thank You!


Sanja Fidler


Arash Vahdat

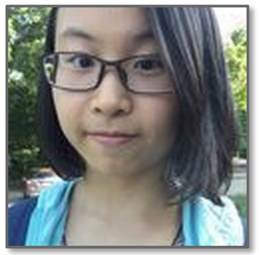

Huan Ling


Seung Wook Kim


Tim Dockhorn


Andreas Blattmann


Robin Rombach


Xiaohui Zeng


Antonia Torralba


Yilun Xu


Zhisheng Xiao

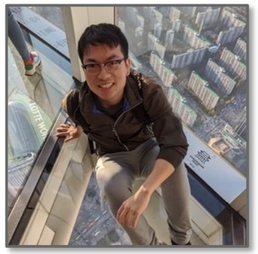

Tianshi Cao


Katja Schwarz


Ellen Zhong


Ming-Yu Liu


Or Litany


Daiqing Li


Amirmojtaba Sabour


Jun Gao


Davis Rempe


Amlan Kar


Masha Shugrina


David Acuna

...and more...

@karsten_kreis

https://karstenkreis.github.io/