NeurIPS 2023 Tutorial

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

Karsten Kreis

Ruiqi Gao

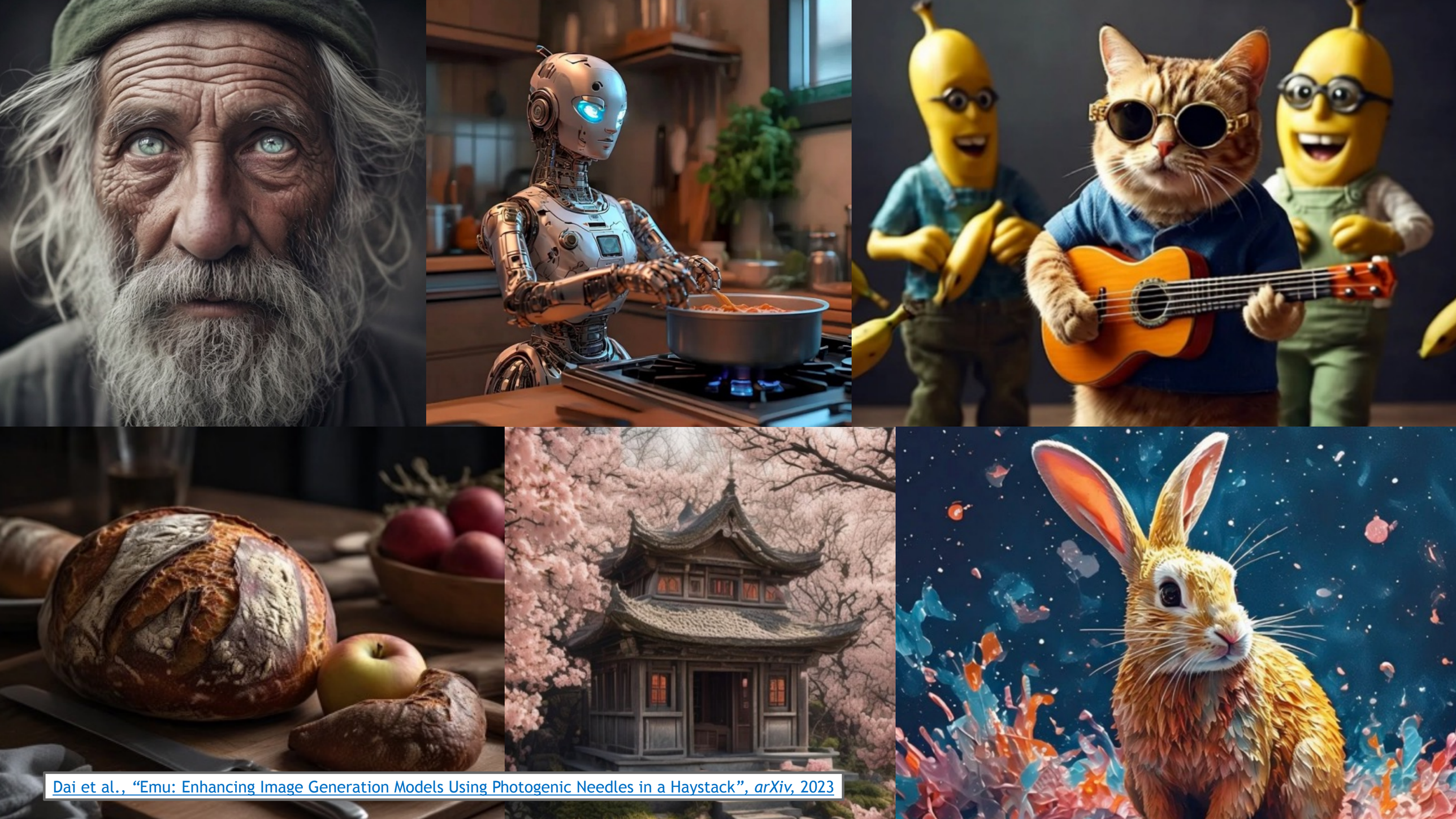
Arash Vahdat

NVIDIA

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

Google DeepMind

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



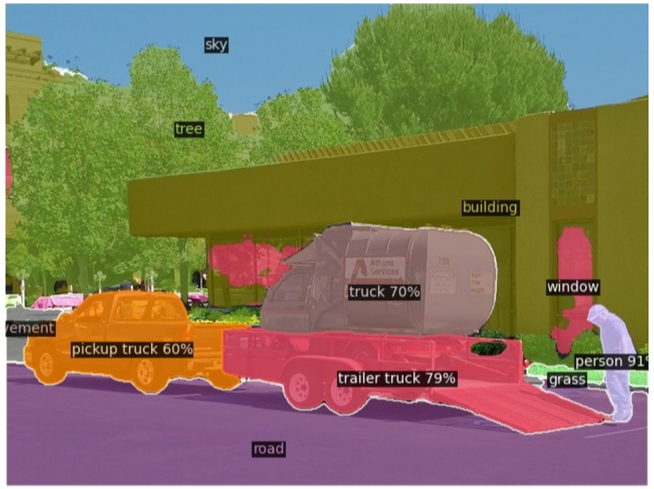
Dai et al., “Emu: Enhancing Image Generation Models Using Photogenic Needles in a Haystack”, *arXiv*, 2023

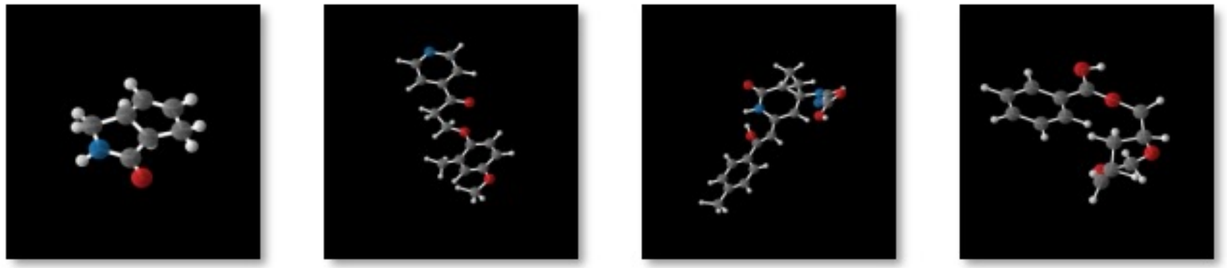


Blattmann et al., “Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models”, *CVPR* 2023
Emu Video, https://emu-video.metademolab.com/

Latent Diffusion Models:

Image, Video, 3D Generation. Graph and Molecule Synthesis. Downstream Discriminative Tasks (e.g. segmentation). …





Zeng et al., “LION: Latent Point Diffusion Models for 3D Shape Generation”, *NeurIPS*, 2022

Kim et al., “NeuralField-LDM: Scene Generation with Hierarchical Latent Diffusion Models”, *CVPR*, 2023

Wang et al., “ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation”, *NeurIPS*, 2023

Xu et al., “ODISE: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models”, *CVPR*, 2023

Xu et al., “Geometric Latent Diffusion Models for 3D Molecule Generation”, *ICML*, 2023

Today's Program

Title	Speaker	Time
Part (1): Introduction to Latent Diffusion Models *Diffusion models, autoencoding, compression, latent diffusion, architectures, image generation*	Karsten	40 min
Part (2): Advanced Design and Controllability *End-to-end training, maximum likelihood, accelerated sampling, distillation, control and editing*	Arash	40 min
Part (3): Latent Diffusion Models beyond Image Generation *Video generation, 3D object and scene synthesis, segmentation, language & molecule generation*	Ruiqi	40 min
Panel Discussion: *Robin Rombach, Durk Kingma, Chenlin Meng, Sander Dieleman, Ying Nian Wu*	Panelists	30 min

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

Panel Discussion

Durk Kingma
Google Deepmind

Chenlin Meng
Pika

Robin Rombach
Stability AI

Sander Dieleman
Google Deepmind

Ying Nian Wu
University of California, Los Angeles

Today's Program

Title	Speaker	Time
Part (1): Introduction to Latent Diffusion Models *Diffusion models, autoencoding, compression, latent diffusion, architectures, image generation*	Karsten	40 min
Part (2): Advanced Design and Controllability *End-to-end training, maximum likelihood, accelerated sampling, distillation, control and editing*	Arash	40 min
Part (3): Latent Diffusion Models beyond Image Generation *Video generation, 3D object and scene synthesis, segmentation, language & molecule generation*	Ruiqi	40 min
Panel Discussion: *Robin Rombach, Durk Kingma, Chenlin Meng, Sander Dieleman, Ying Nian Wu*	Panelists	30 min

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

Part (1):
Introduction to Latent Diffusion Models

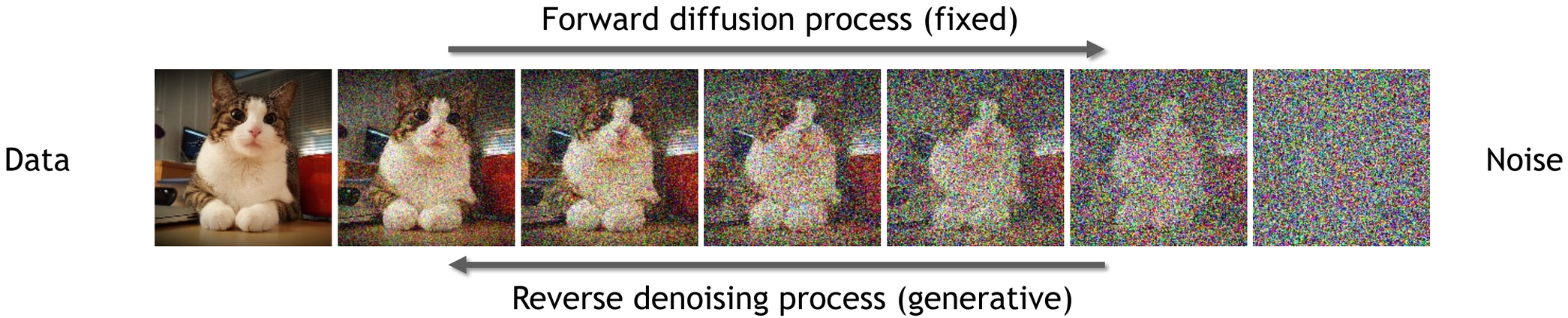
Part (1):
Introduction to Latent Diffusion Models

Diffusion Models

How do they work?

Denoising diffusion models consist of two processes:

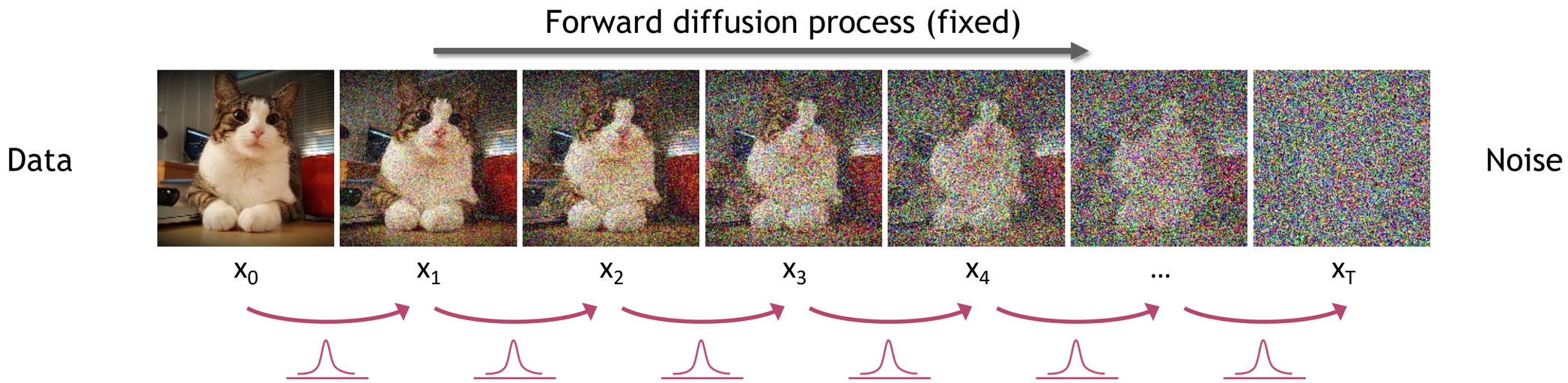
- Forward diffusion process that gradually adds noise to input
- Reverse denoising process that learns to generate data by denoising



Sohl-Dickstein et al., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”, *ICML*, 2015
Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS*, 2020
Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021

Diffusion Models

The Fixed Forward Diffusion Process



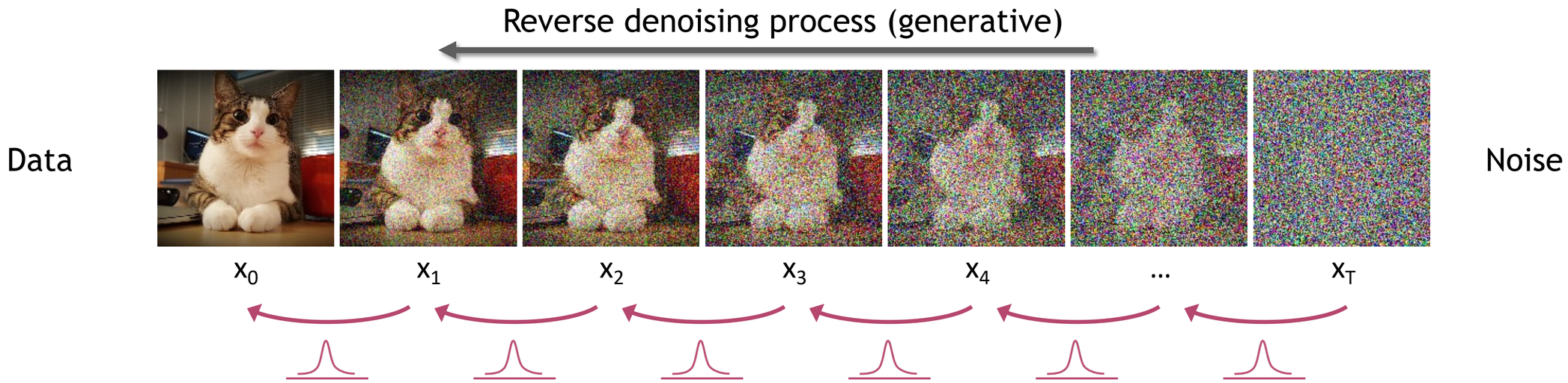
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad \Rightarrow \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad \text{(joint)}$$

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

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)

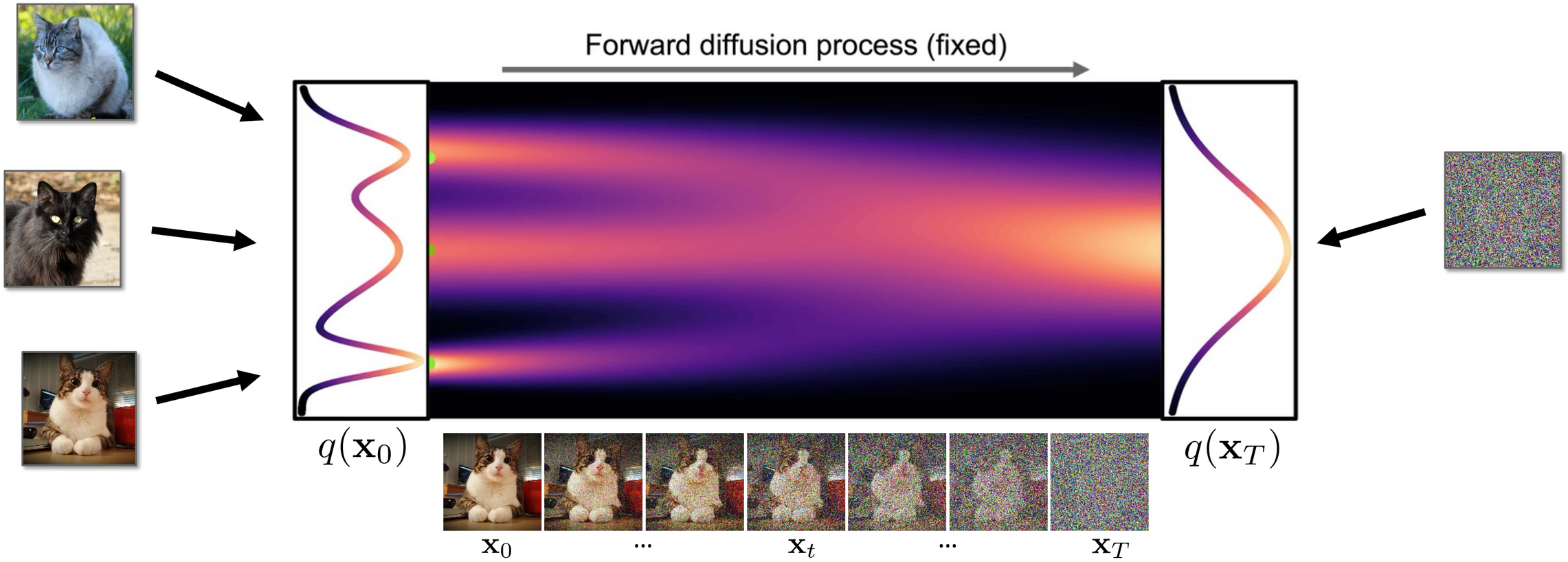
$$\Rightarrow \quad p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Trainable via evidence lower bound (ELBO)
or denoising score matching.

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

Diffusion Models

A Stochastic Differential Equation-based Perspective

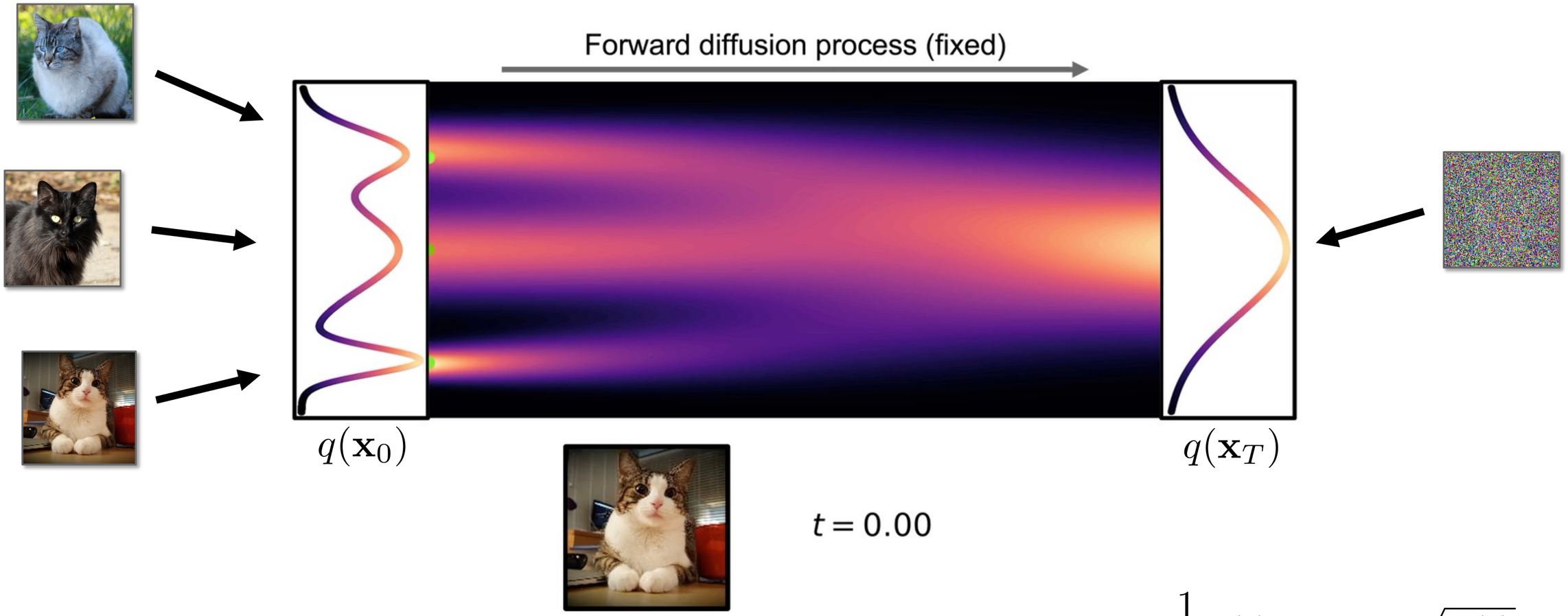


$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

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

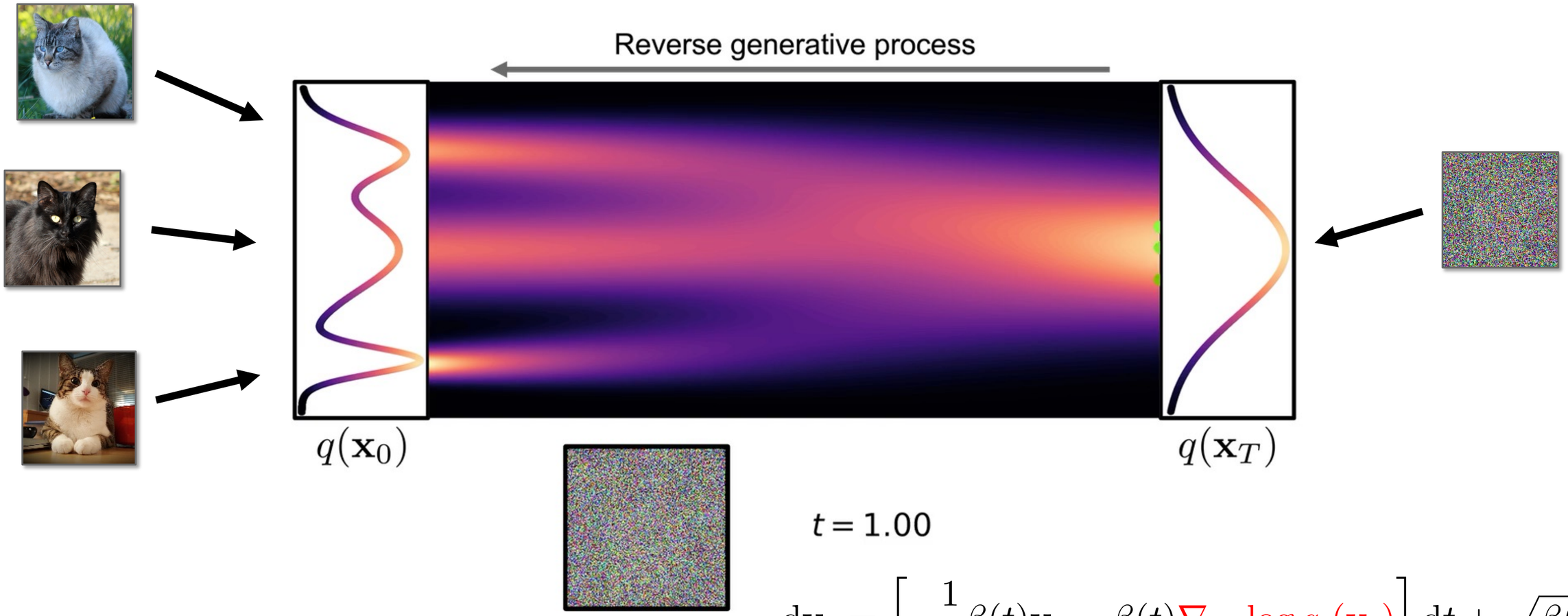


$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

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



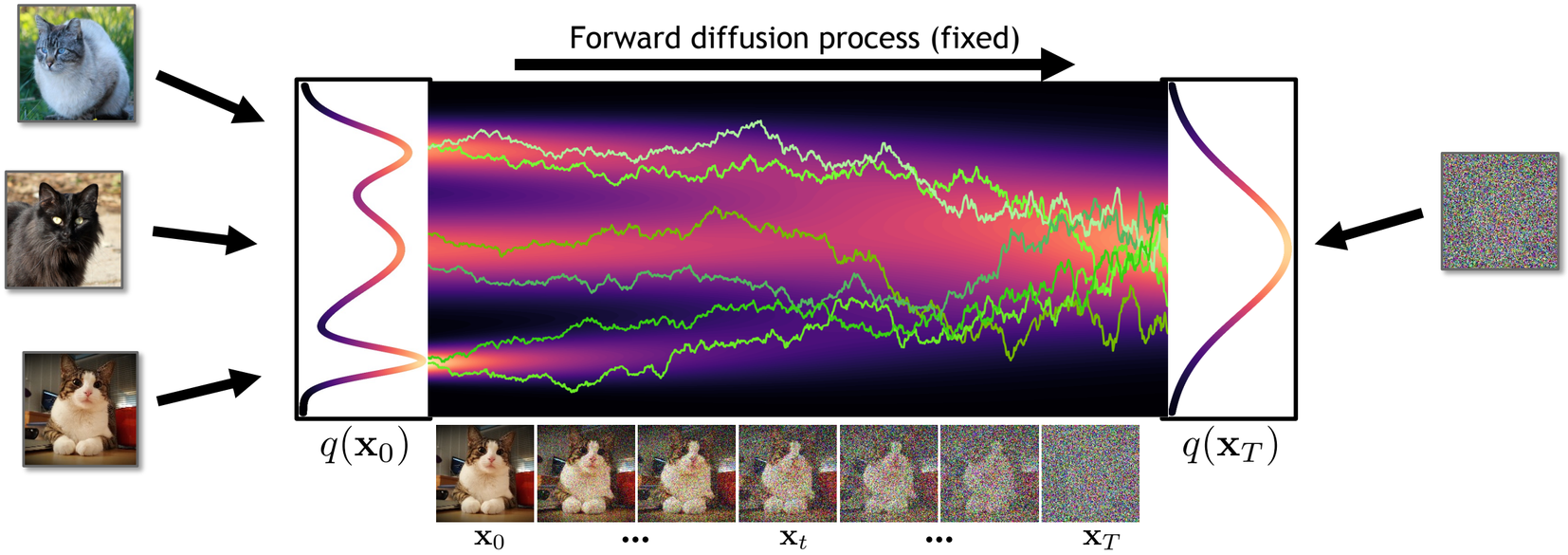
$t = 1.00$

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

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



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

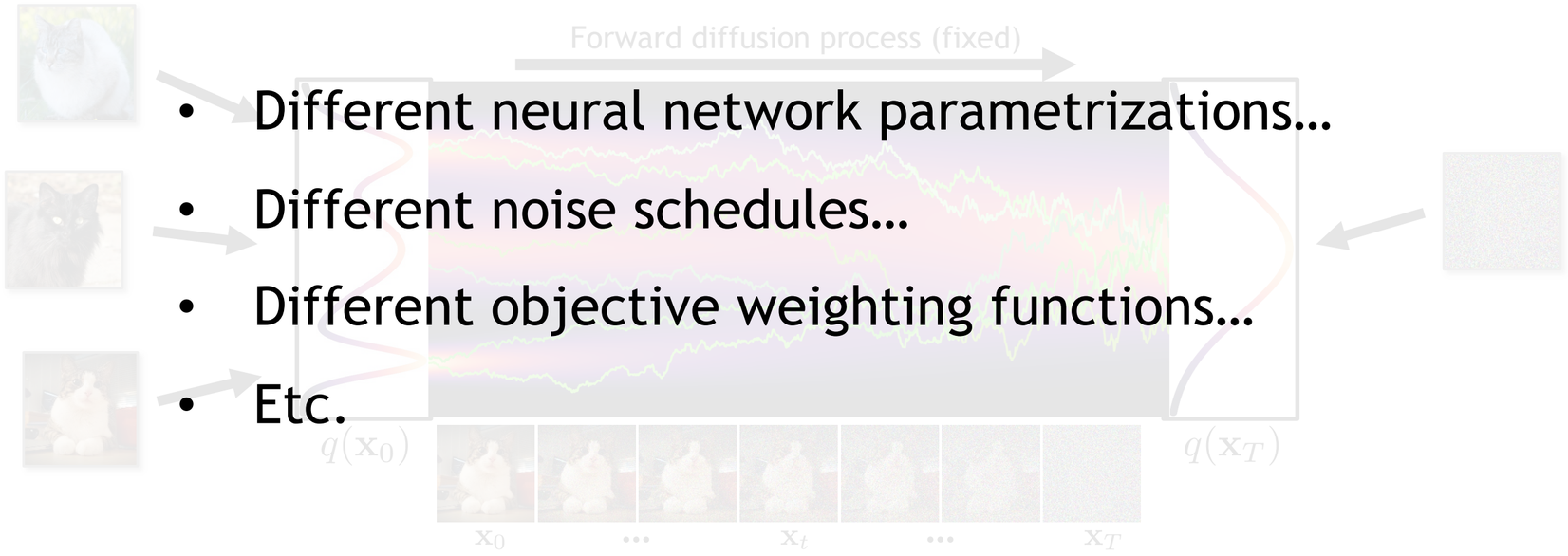
diffusion time t	data sample $\mathbf{x}_0$	diffused data sample $\mathbf{x}_t$	neural network	score of diffused data sample

➡ **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

Training Diffusion Models with Denoising Score Matching



- Different neural network parametrizations…
- Different noise schedules…
- Different objective weighting functions…
- Etc.

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

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

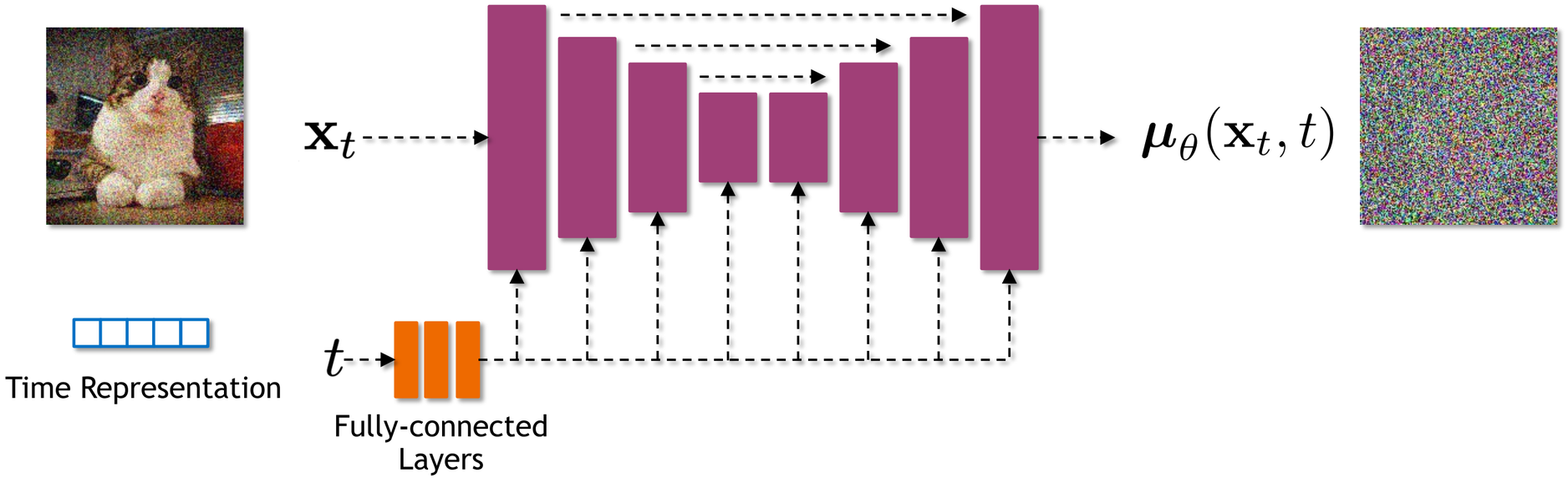
$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)$!

Ho et al., “Denoising Diffusion Probabilistic Models”, *NeurIPS*, 2020
Song et al., “Score-Based Generative Modeling through Stochastic Differential Equations”, *ICLR*, 2021
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
Kingma and Gao, “Understanding Diffusion Objectives as the ELBO with Simple Data Augmentation”, *NeurIPS*, 2023
Hoogeboom et al., “simple diffusion: End-to-end diffusion for high resolution images”, *ICML*, 2023

Implementation Considerations

Network Architectures

Diffusion models often use U-Net architectures with 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", *arXiv*, 2023

Previous Tutorials on Diffusion Models

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

YouTube (~4 hours long, over 100,000 views):
https://www.youtube.com/watch?v=cS6JQpEY9cs

Website:
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*

YouTube:
https://www.youtube.com/watch?v=1d4r19GEVos

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


Jiaming Song
NVIDIA


Chenlin Meng
Stanford


Arash Vahdat
NVIDIA

Part (1):
Introduction to Latent Diffusion Models

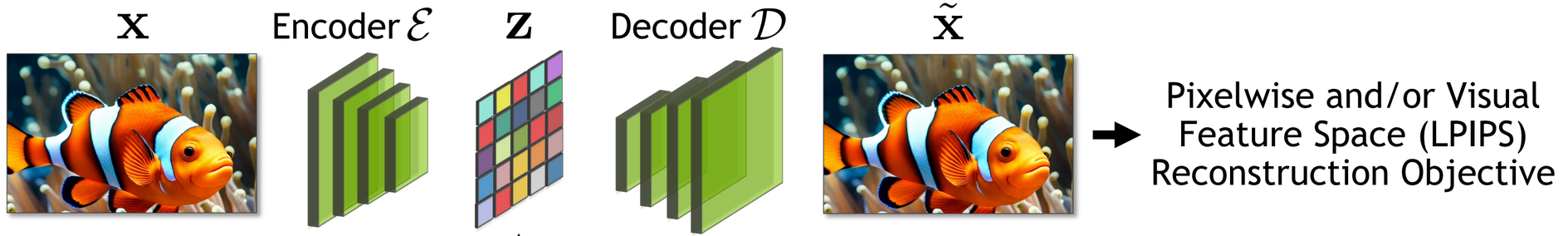
Latent Diffusion Models

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

- Stage 1:

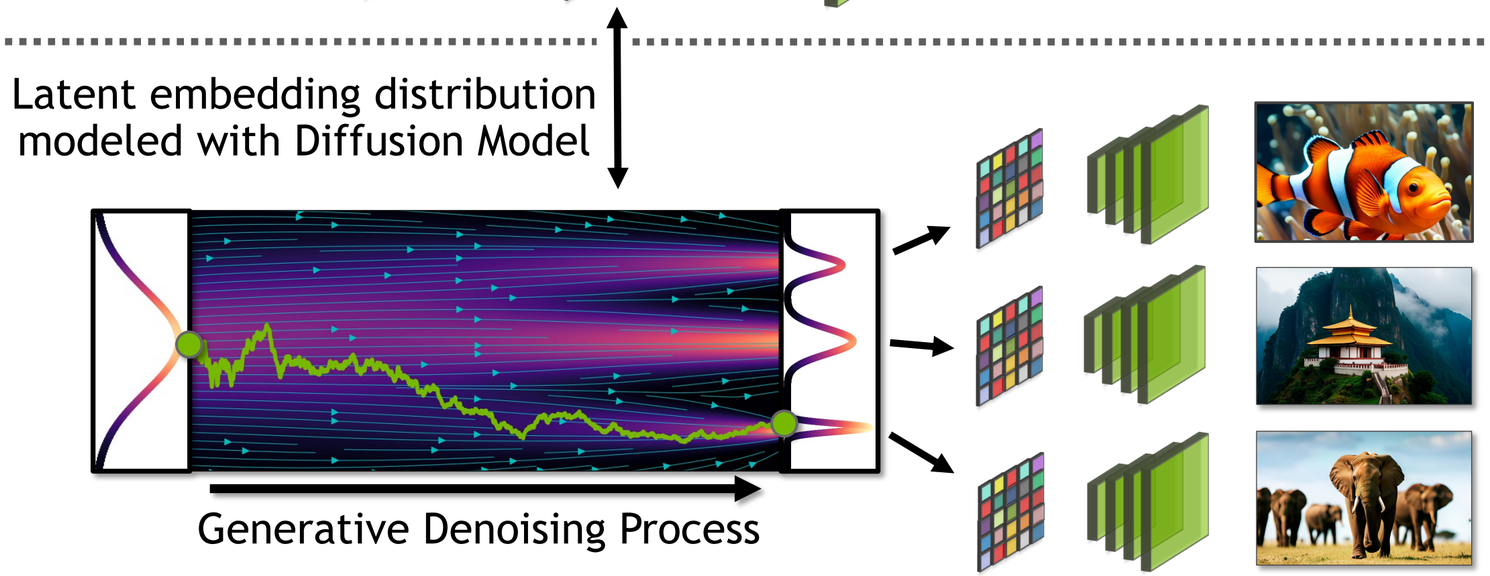
Train Autoencoder

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



- Stage 2:

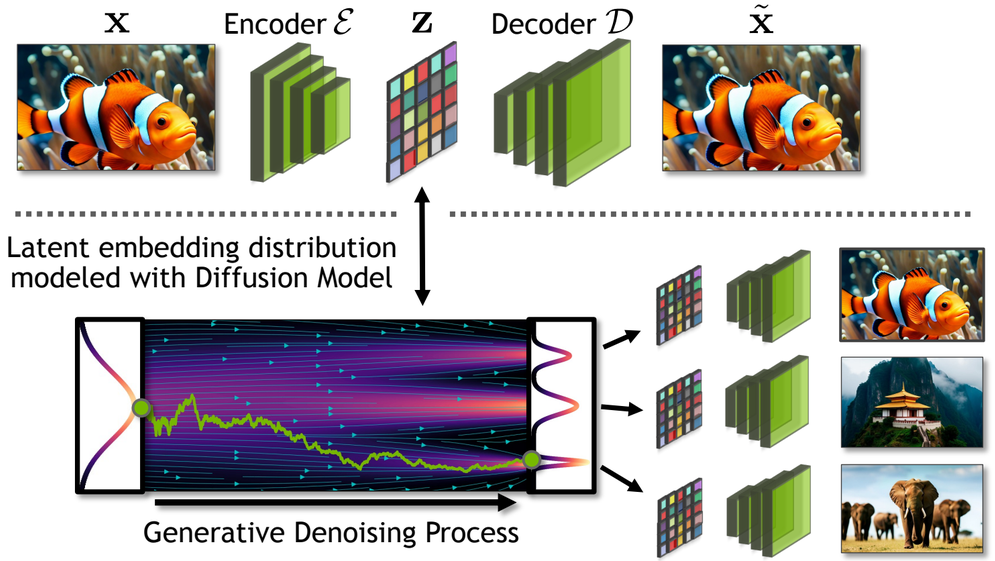
Train **Latent** Diffusion Model



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.



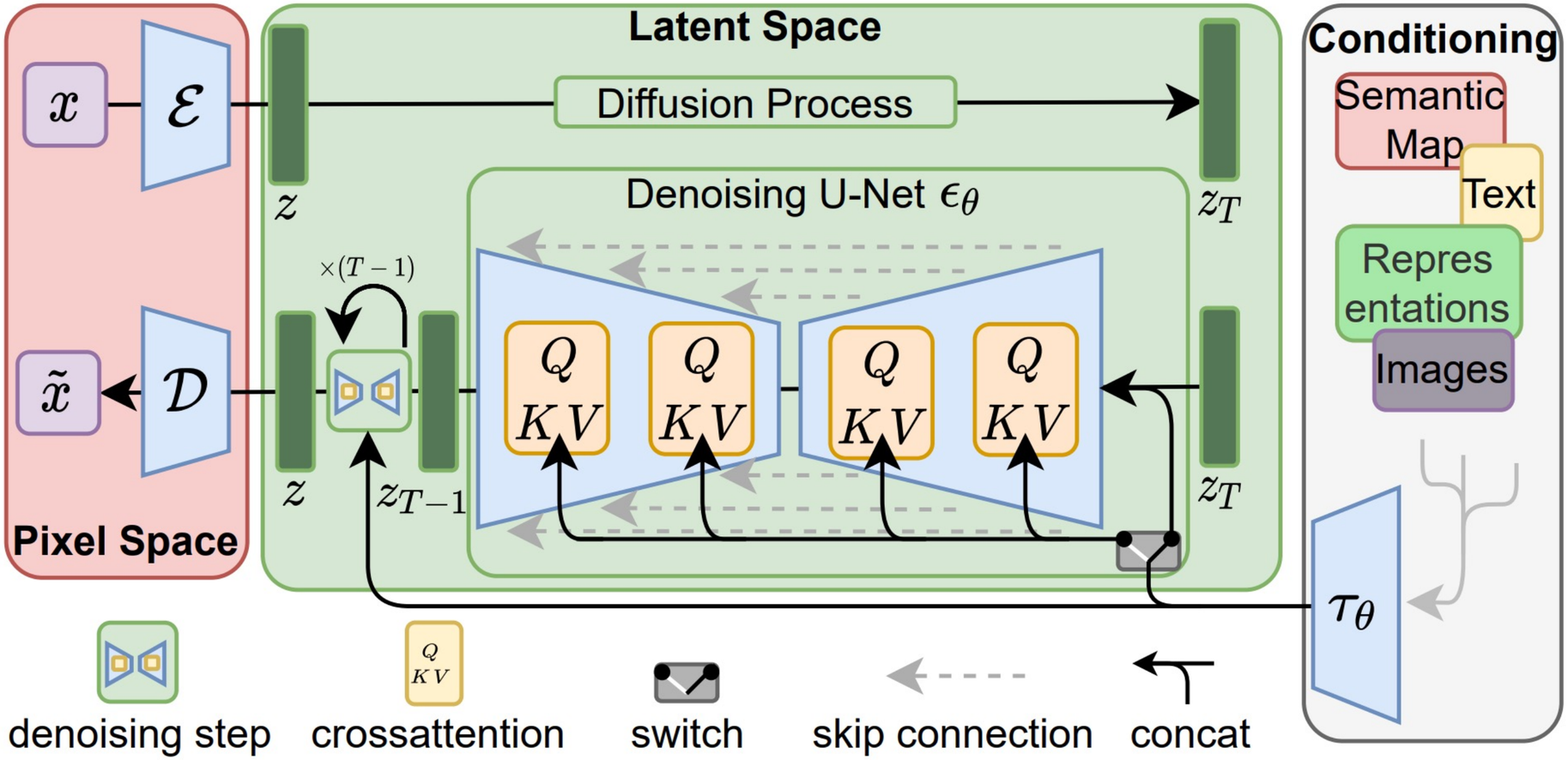
Advantages:

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

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



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

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

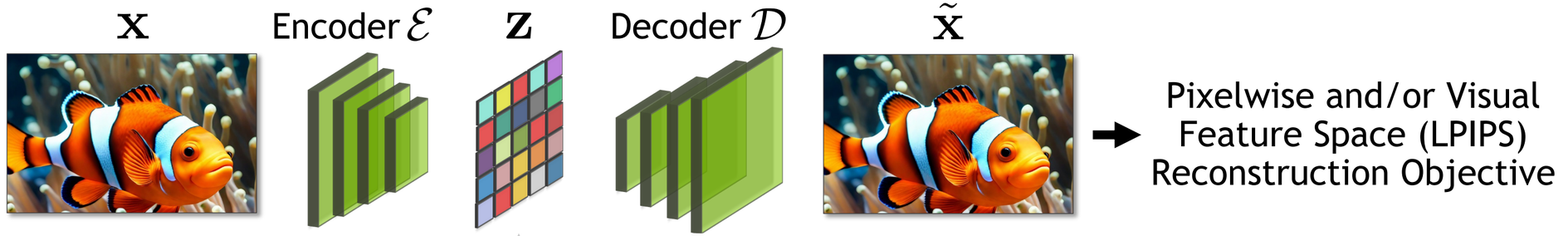
Latent Diffusion Models

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

- Stage 1:

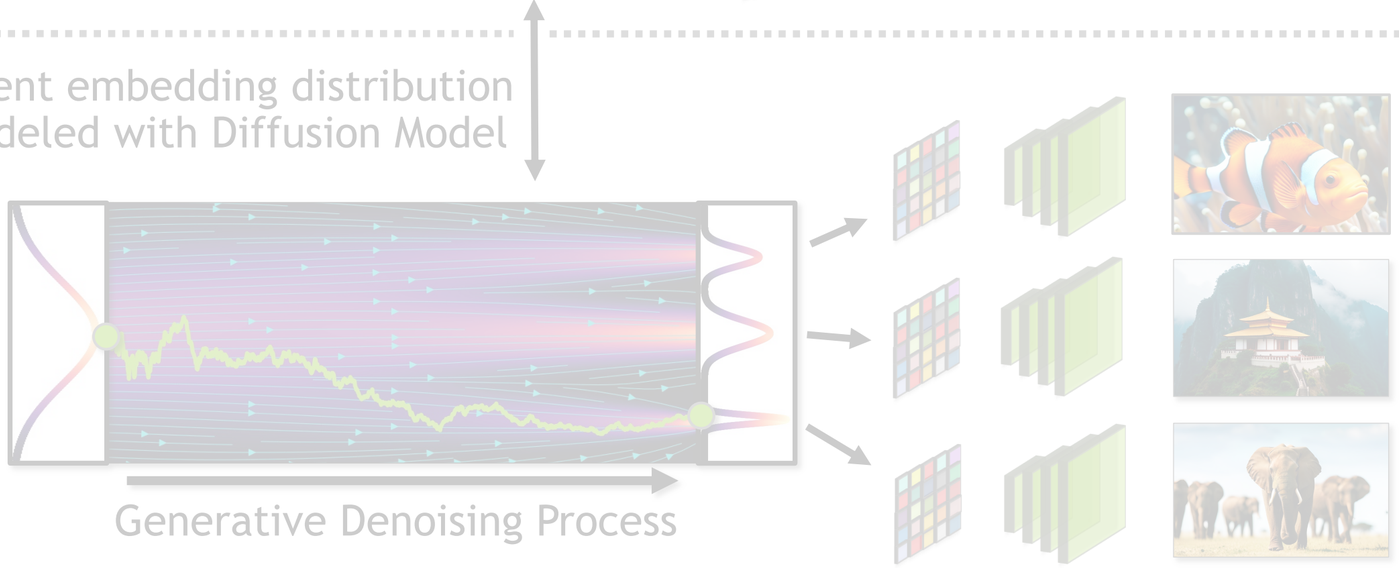
Train Autoencoder

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



- Stage 2:

Train **Latent** Diffusion Model



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

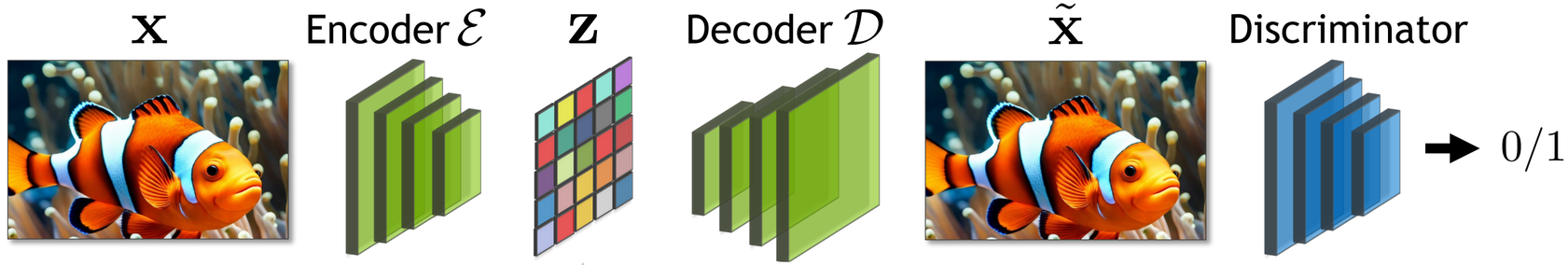
Latent Diffusion Models

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

- Stage 1:

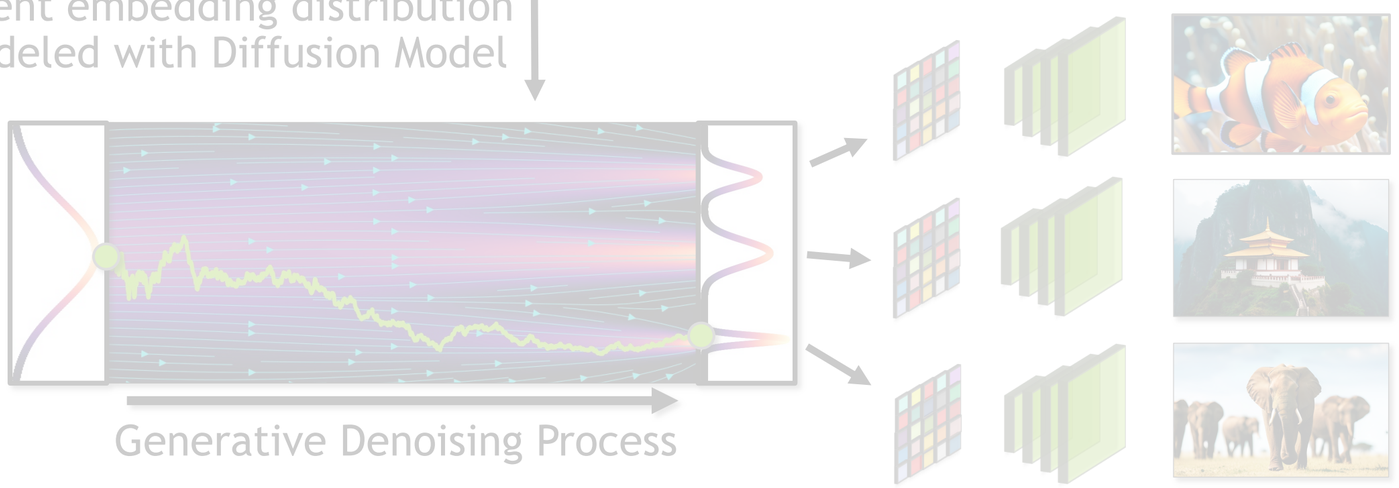
Train Autoencoder

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



- Stage 2:

Train **Latent** Diffusion Model



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

Input

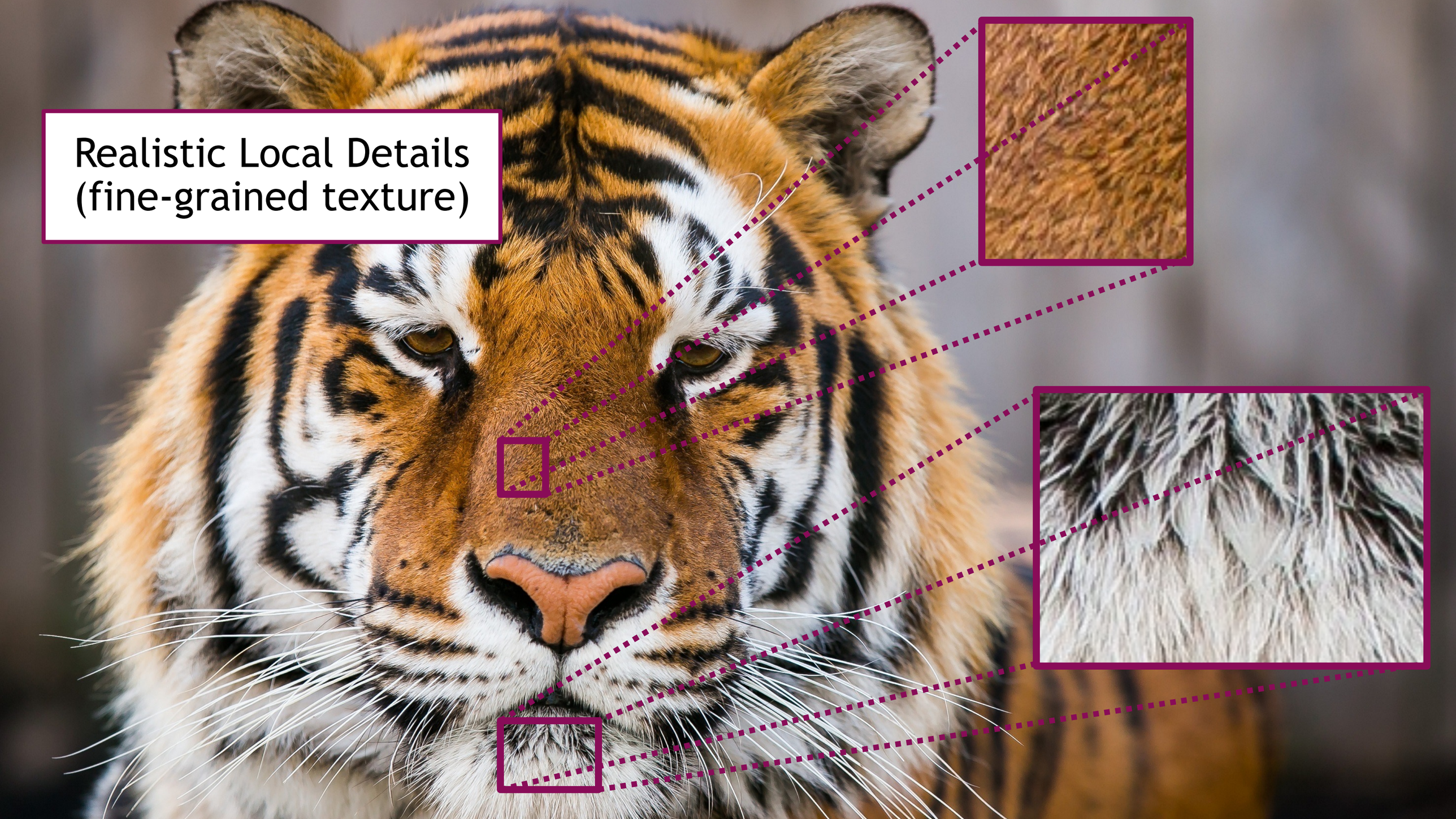
Reconstruction without Discriminator

Reconstruction with Discriminator

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

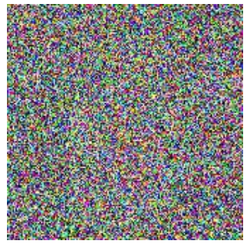

What makes an image look realistic and high-quality?


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

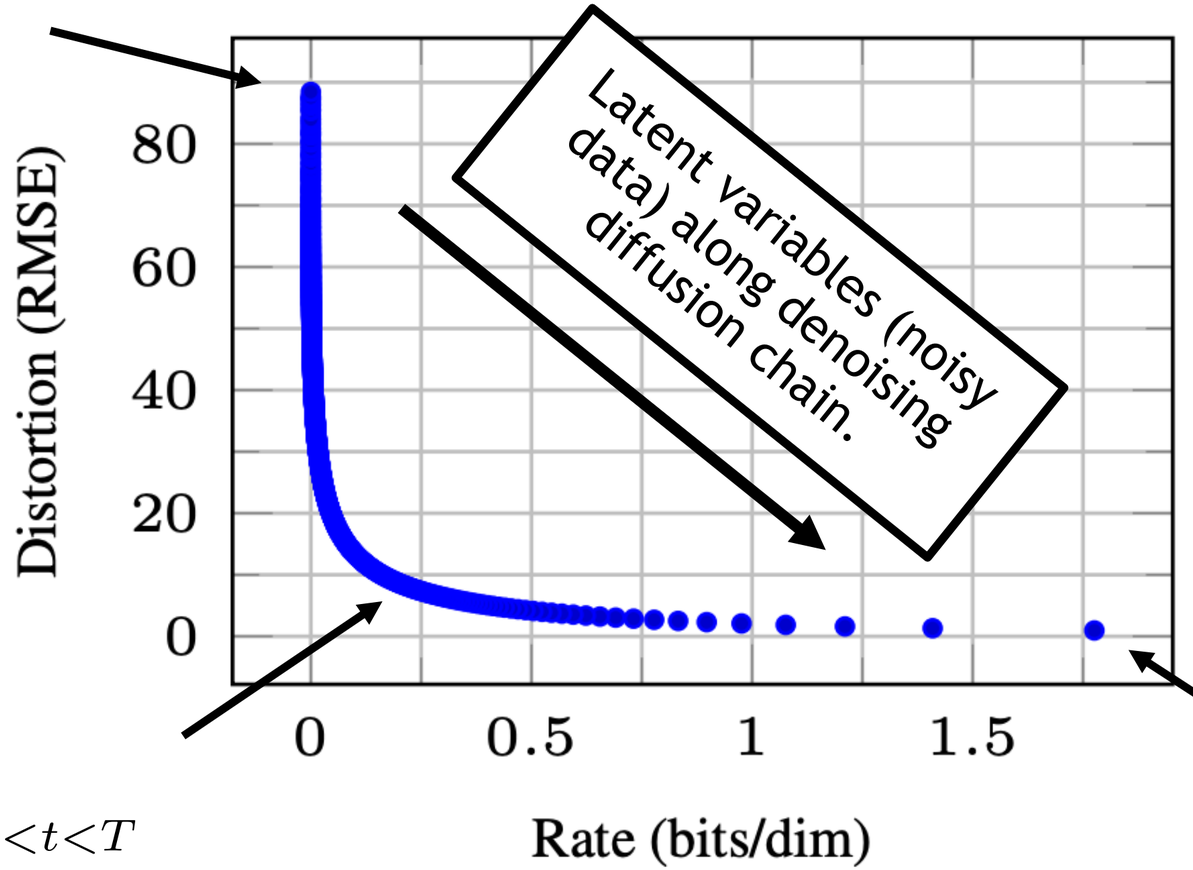

Realistic Local Details
(fine-grained texture)

Compression/Encoding in Diffusion Models

Only $\mathbf{x}_T$ (pure noise)



Intermediate $\mathbf{x}_{0<t<T}$



(Almost) $\mathbf{x}_0$

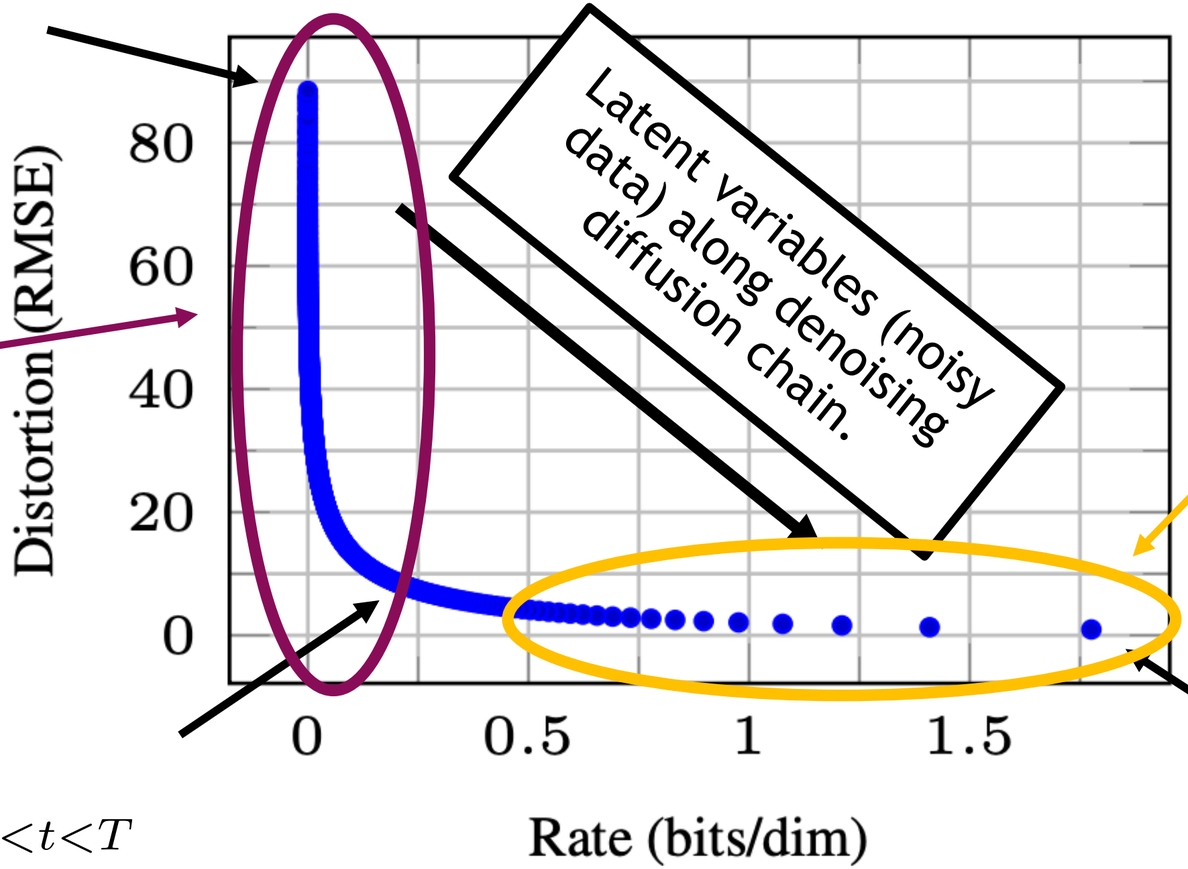
Diffusion models encode images in their noisy latent states.

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

Compression/Encoding in Diffusion Models

Only $\mathbf{x}_T$
(pure noise)

Large-scale
Image Structure



Local, Imperceptible
Details

Intermediate $\mathbf{x}_{0<t<T}$

(Almost) $\mathbf{x}_0$

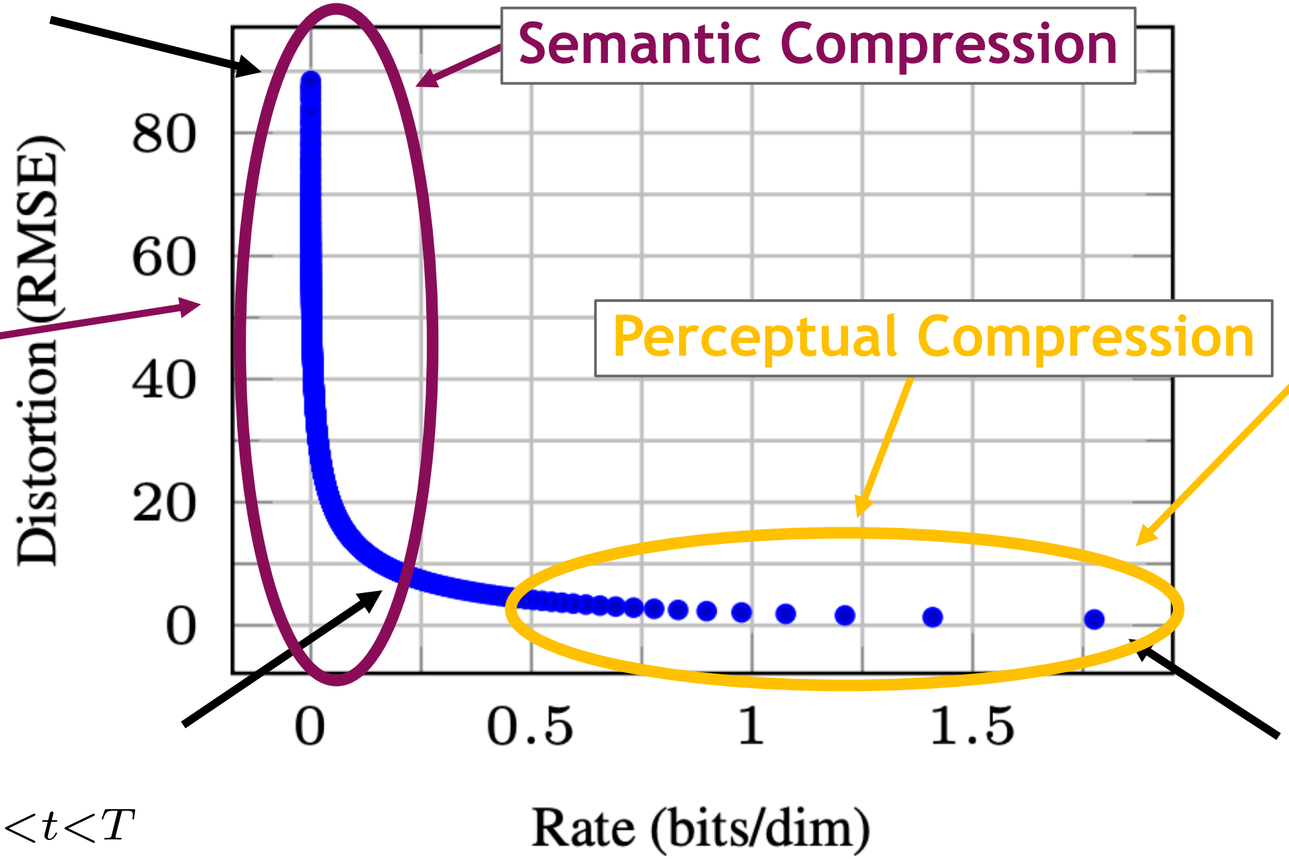
Diffusion models encode images in their noisy latent states.

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

Perceptual and Semantic Compression in Diffusion Models

Only $\mathbf{x}_T$
(pure noise)

Large-scale
Image Structure



Local, Imperceptible
Details

Intermediate $\mathbf{x}_{0<t<T}$

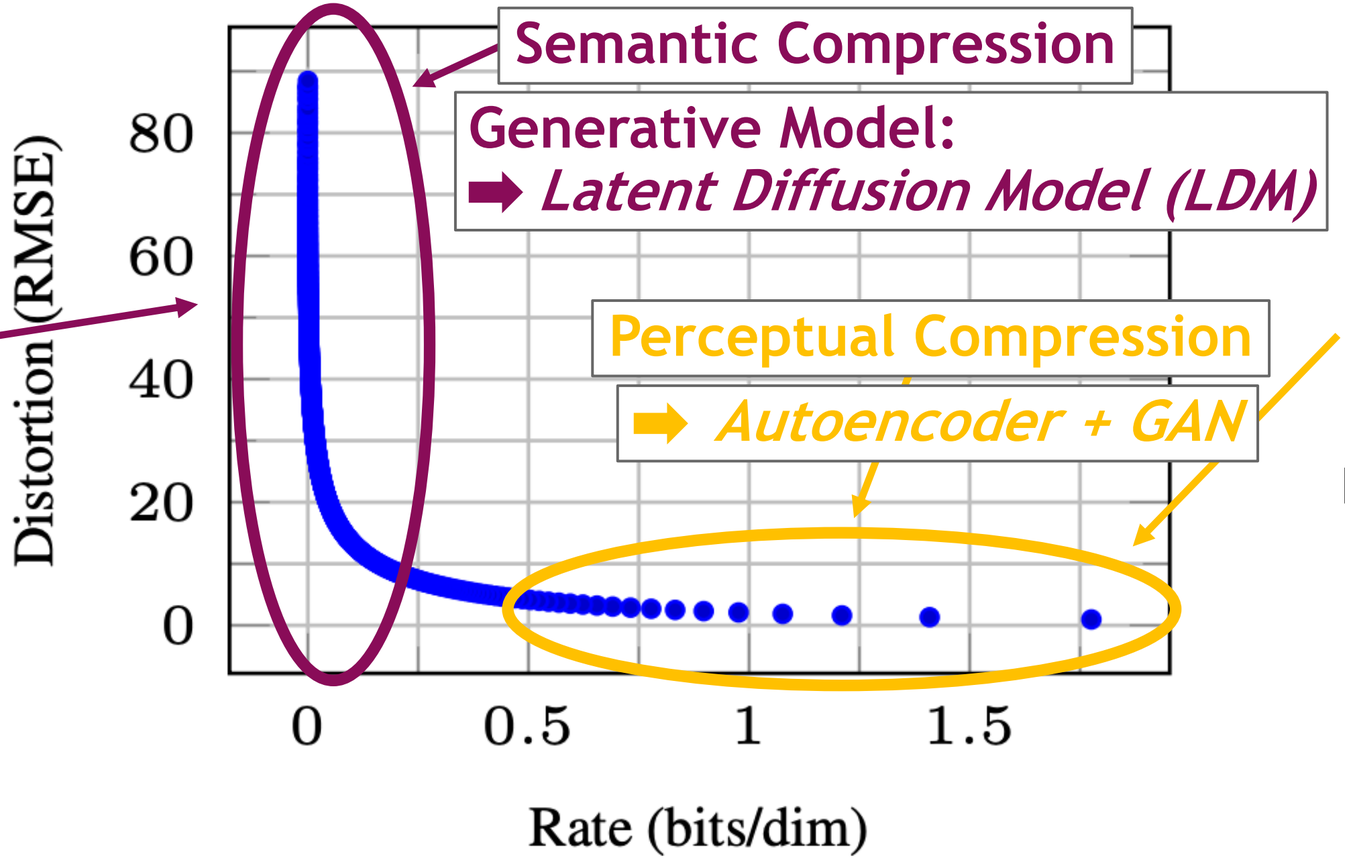
(Almost) $\mathbf{x}_0$

Diffusion models encode images in their noisy latent states.

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

Perceptual and Semantic Compression in *Latent* Diffusion Models

Large-scale
Image Structure



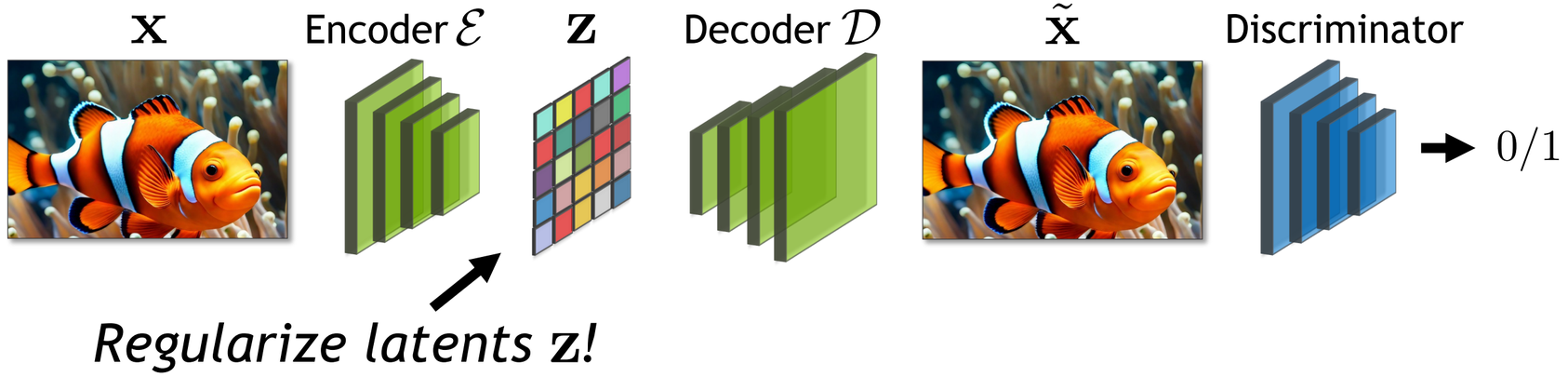
Local, Imperceptible
Details

LDMs: **Latent diffusion model** for **large-scale structure**, **Autoencoder/GAN** for **local details**.

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).

Encoder distribution:

$$q_{\mathcal{E}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mathcal{E}_{\mu}, \mathcal{E}_{\sigma}^2)$$

KL regularization in latent space:

$$\mathrm{KL}\left(q_{\mathcal{E}}(\mathbf{z}|\mathbf{x})||\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})\right)$$

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

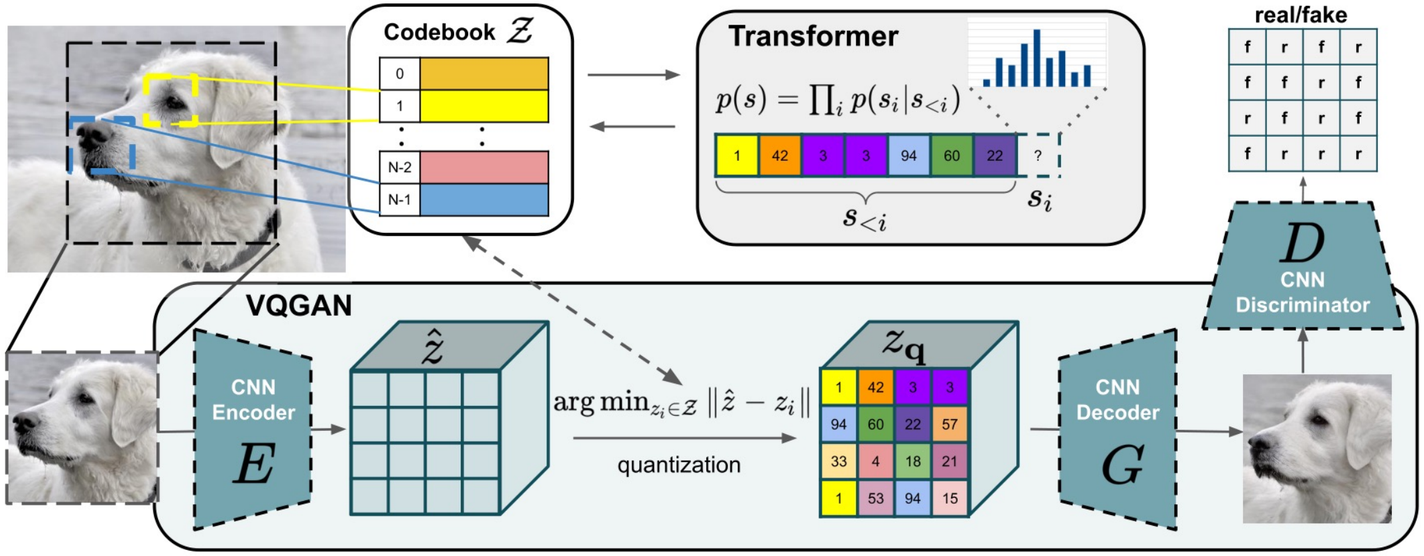
Latent Space Regularization

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

- **Option 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).



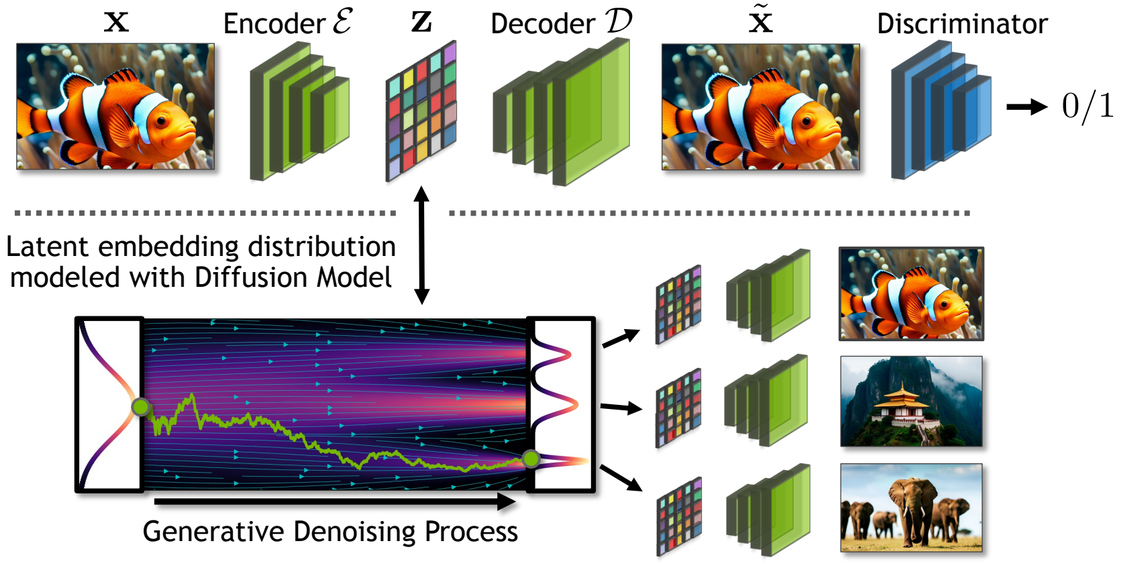
Esser et al., "*Taming Transformers for High-Resolution Image Synthesis*", *CVPR*, 2021

Latent Diffusion Models

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

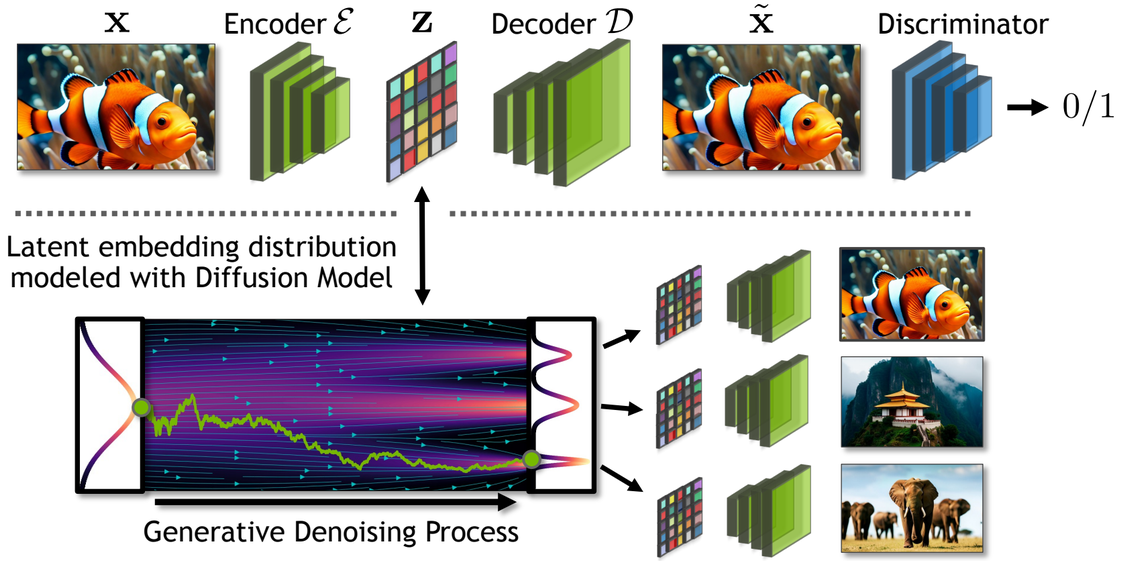
LDM "Recipe":

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



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**?

Common observation:

- Latent diffusion model **technology is mature** for practical image generation.
- The above models all achieve their high-performance 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

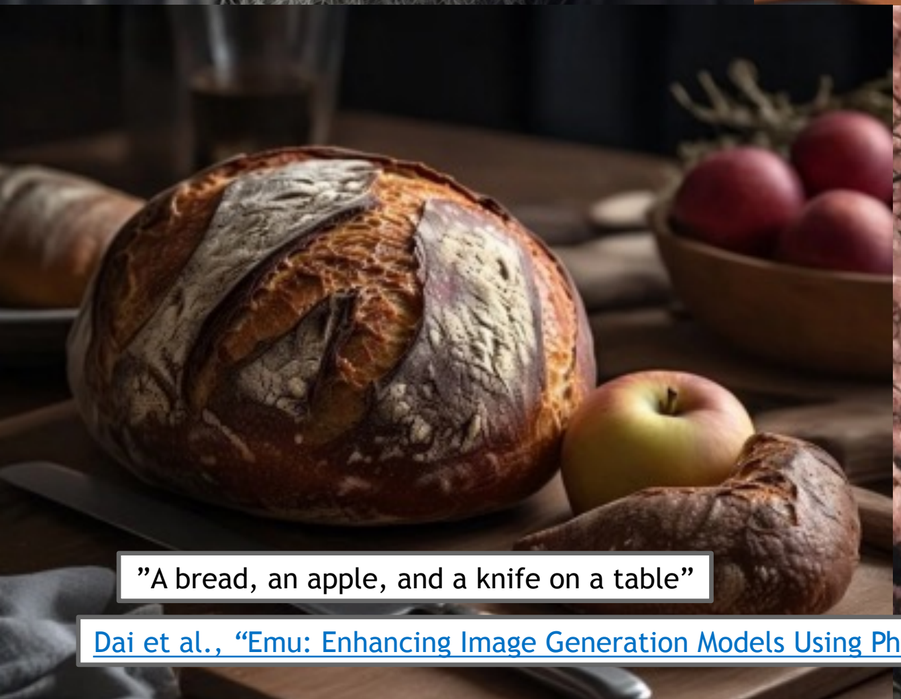
https://stability.ai/news/stable-diffusion-sdxl-1-announcement

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

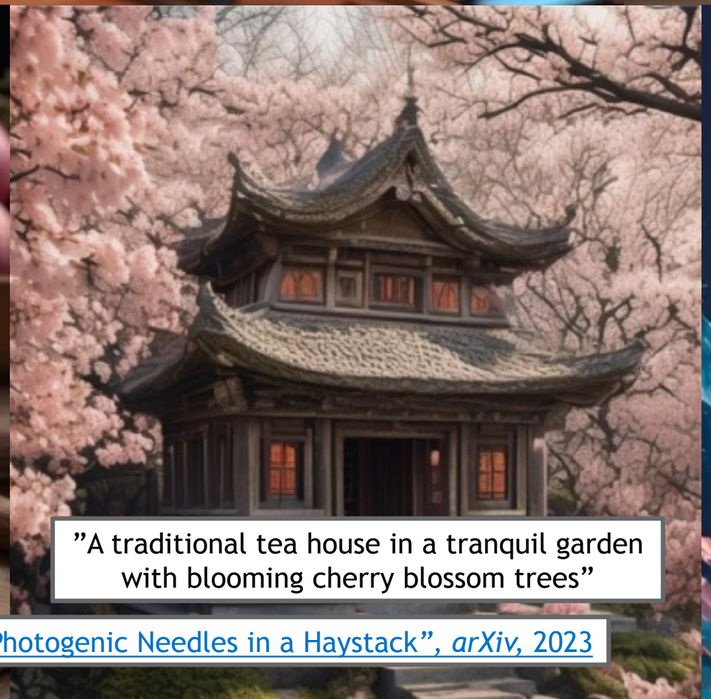
”An old man with green eyes and a long grey beard”

”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

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*

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

Today's Program

Title	Speaker	Time
Part (1): Introduction to Latent Diffusion Models *Diffusion models, autoencoding, compression, latent diffusion, architectures, image generation*	Karsten	40 min
Part (2): Advanced Design and Controllability *End-to-end training, maximum likelihood, accelerated sampling, distillation, control and editing*	Arash	40 min
Part (3): Latent Diffusion Models beyond Image Generation *Video generation, 3D object and scene synthesis, segmentation, language & molecule generation*	Ruiqi	40 min
Panel Discussion: *Robin Rombach, Durk Kingma, Chenlin Meng, Sander Dieleman, Ying Nian Wu*	Panelists	30 min

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

NeurIPS 2023 Tutorial

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

Karsten Kreis

Ruiqi Gao

Arash Vahdat

NVIDIA

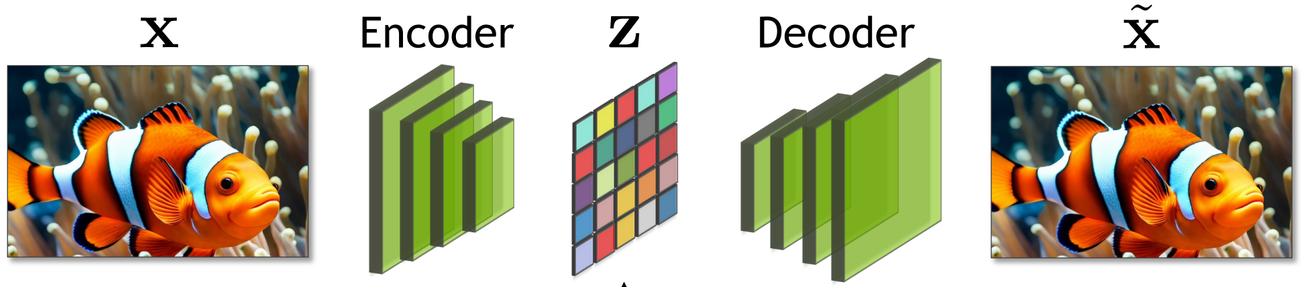
Google DeepMind

Recap: Two-Stage Latent Diffusion Training

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

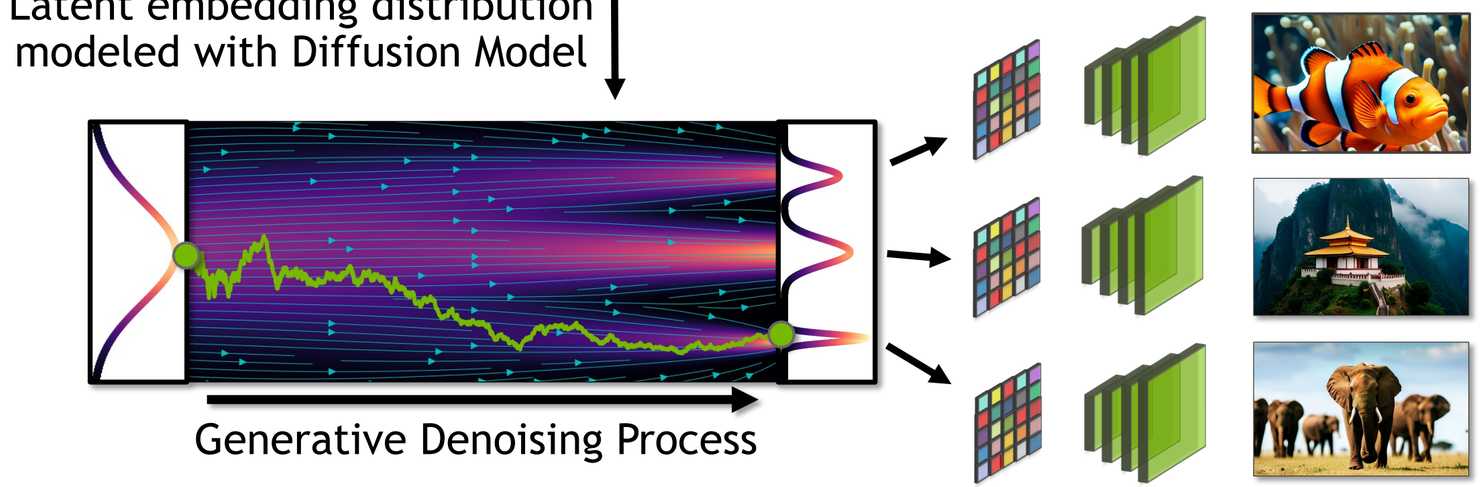
Stage 1:

Train Autoencoder



Stage 2:

Train **Latent** Diffusion Model



Agenda

End-to-End Training

Accelerated Sampling

Inverse Problems

Controllability & Manipulation

Agenda

End-to-End Training

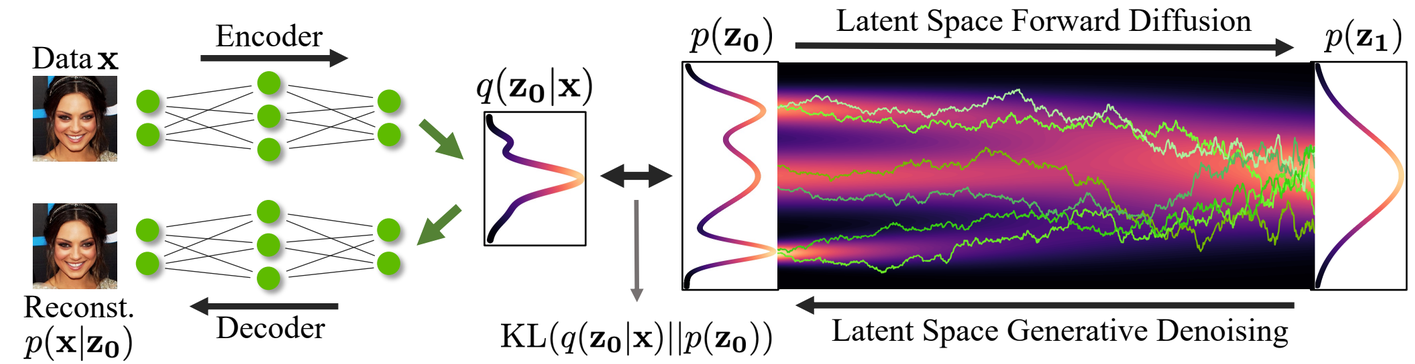
Accelerated Sampling

Inverse Problems

Controllability & Manipulation

End-to-end training of latent diffusion models

Variational autoencoder + score-based prior



Main Idea

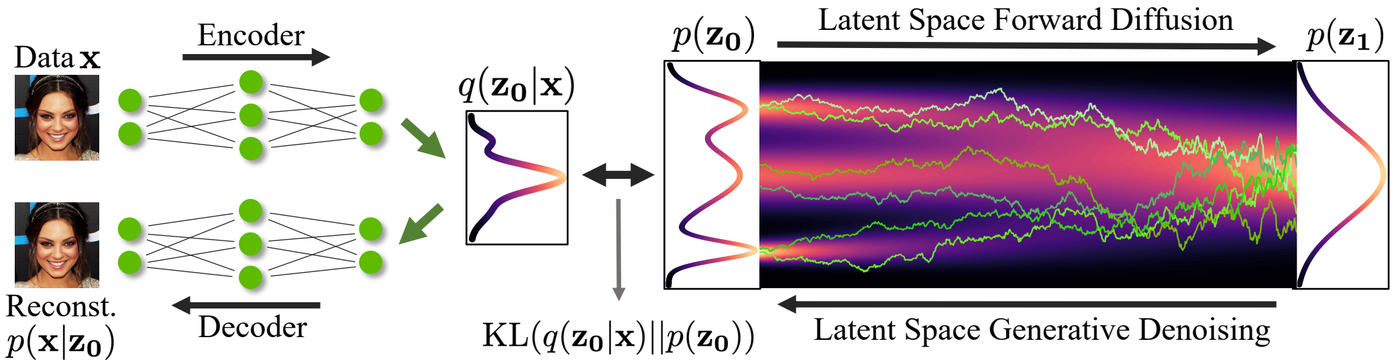
Pre-train the variational autoencoder model (VAE) with a standard Normal prior

Train both the VAE and diffusion prior jointly

Vahdat et al., “Score-based generative modeling in latent space”, NeurIPS 2021.

Latent-space diffusion models

Variational autoencoder + score-based prior



(1) The distribution of latent embeddings is close to Normal distribution → *Simpler denoising, Faster Synthesis!*

(2) Augmented latent space → *More expressivity & Flexibility!*

(3) Tailored Autoencoders → *Application to any data type (graphs, text, 3D data, etc.)!*

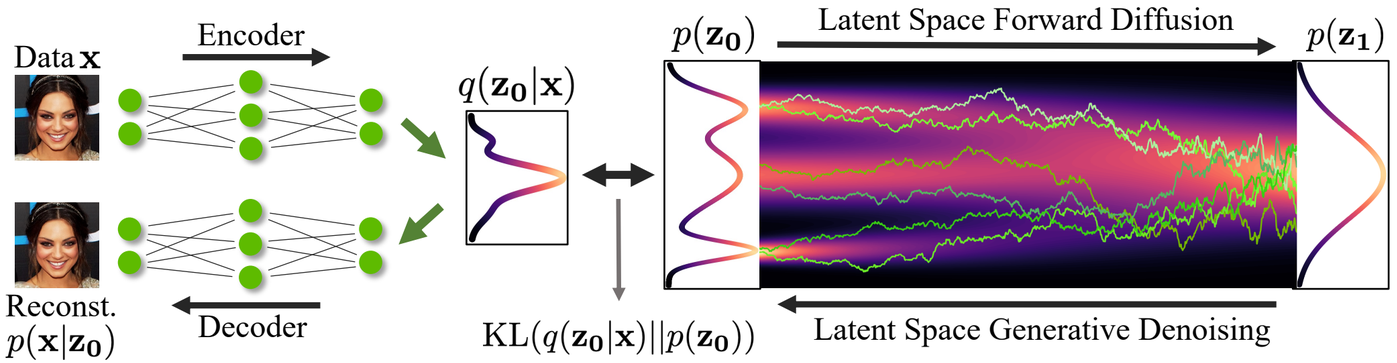
(4) Autoencoder and diffusion prior work together → *Trade complexity between encoder and latent space*

(1) Scalability → *Training encoder and prior jointly is expensive*

Vahdat et al., “Score-based generative modeling in latent space”, NeurIPS 2021.

Latent-space diffusion models

Training objective: score-matching for cross entropy



$$\mathcal{L}(\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x})}\left[-\log p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}_0)\right] + \mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}_0))$$

$$= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x})}\left[-\log p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}_0)\right]}_{\text{reconstruction term}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x})}\left[\log q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x})\right]}_{\text{negative encoder entropy}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_0|\mathbf{x})}\left[-\log p_{\boldsymbol{\theta}}(\mathbf{z}_0)\right]}_{\text{cross entropy}}$$

$$CE(q(\mathbf{z}_0|\mathbf{x})||p(\mathbf{z}_0)) \leq \underbrace{\mathbb{E}_{t\sim\mathcal{U}[0,1]}}_{\text{time sampling}}\left[\frac{g(t)^2}{2}\underbrace{\mathbb{E}_{q(\mathbf{z}_t,\mathbf{z}_0|\mathbf{x})}}_{\text{Forward diffusion}}\left[||\underbrace{\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t|\mathbf{z}_0)}_{\text{Diffusion kernel}} - \underbrace{\nabla_{\mathbf{z}_t}\log p(\mathbf{z}_t)}_{\text{Trainable score function}}||_2^2\right]\right] + \underbrace{\frac{D}{2}\log\left(2\pi e\sigma_0^2\right)}_{\text{Constant}}$$

Vahdat et al., “Score-based generative modeling in latent space”, NeurIPS 2021.

Normal Distribution assumption

Inductive biases for end-to-end training

Denoising score matching objective has high variance!

Recall that the distribution of latent variables is close to a Normal distribution:

$$CE(q(\mathbf{z}_0|\mathbf{x})||p(\mathbf{z}_0)) = \mathbb{E}_{t\sim\mathcal{U}[0,1]}\left[\frac{g(t)^2}{2}\mathbb{E}_{q(\mathbf{z}_t,\mathbf{z}_0|\mathbf{x})}\left[||\nabla_{\mathbf{z}_t}\log q(\mathbf{z}_t|\mathbf{z}_0) - \nabla_{\mathbf{z}_t}\log p(\mathbf{z}_t)||_2^2\right]\right]$$

Reduce the variance of the objective function using importance sampling for the Normal assumption

Design score function that is close to a Normal score function

Agenda

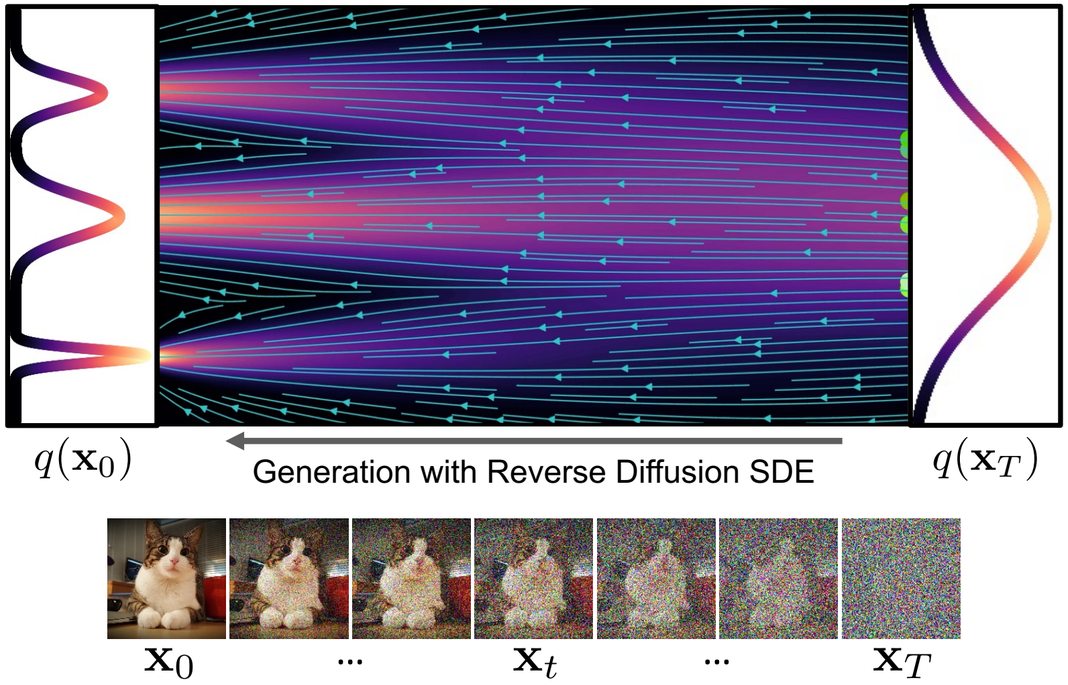
End-to-End Training

Accelerated Sampling

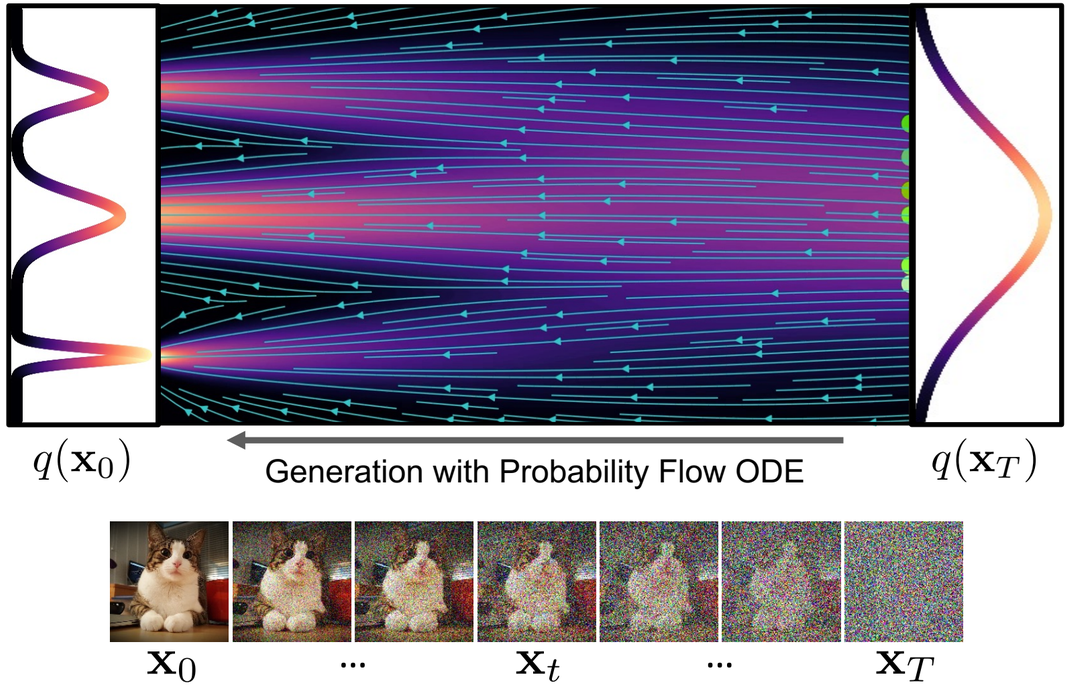
Inverse Problems

Controllability & Manipulation

Generation with SDE vs. ODE



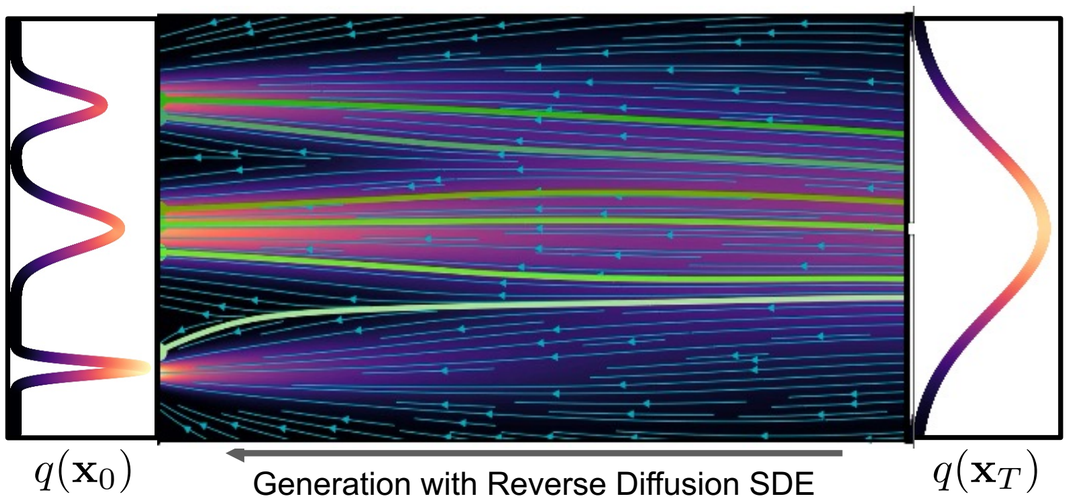
- **Generative Reverse Diffusion SDE (stochastic)**
- **Slow generation - Many function calls**



- **Generative Probability Flow ODE (deterministic)**
- **Easier to accelerate**

Song et al., *ICLR*, 2021

Accelerated Sampling with Numerical ODE Solvers



Trajectories in the latent space tend to be smoother

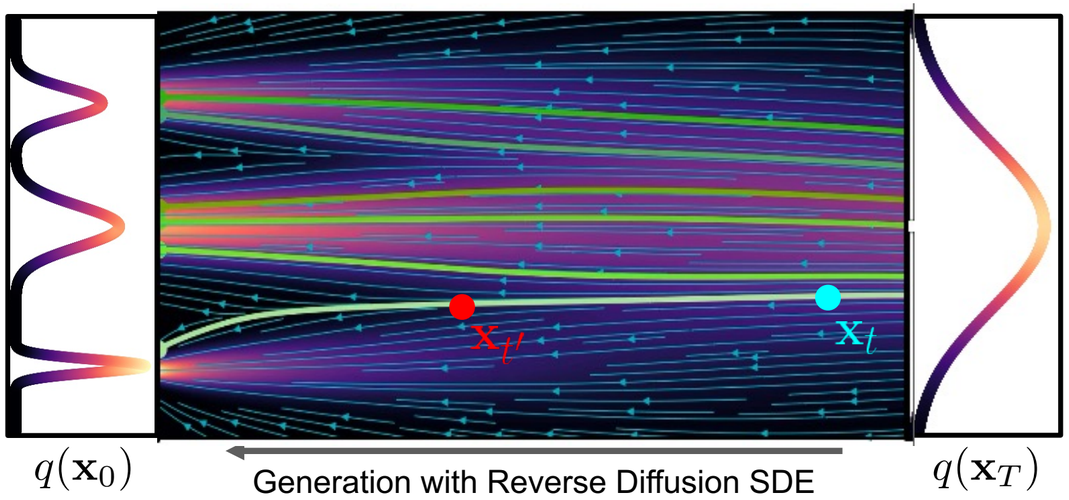
Use off-the-shelf SDE/ODE solvers to accelerate sampling

CelebA-HQ 256	FID ↓	# Fun. Calls
Data-space Diffusion (Song et al. ICLR 2021)	7.22	4000
Latent-space Diffusion (Vahdat et al. NeurIPS 2021)	7.23	23

A Rich Body of Work on ODE/SDE Solvers for Diffusion Models

- 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

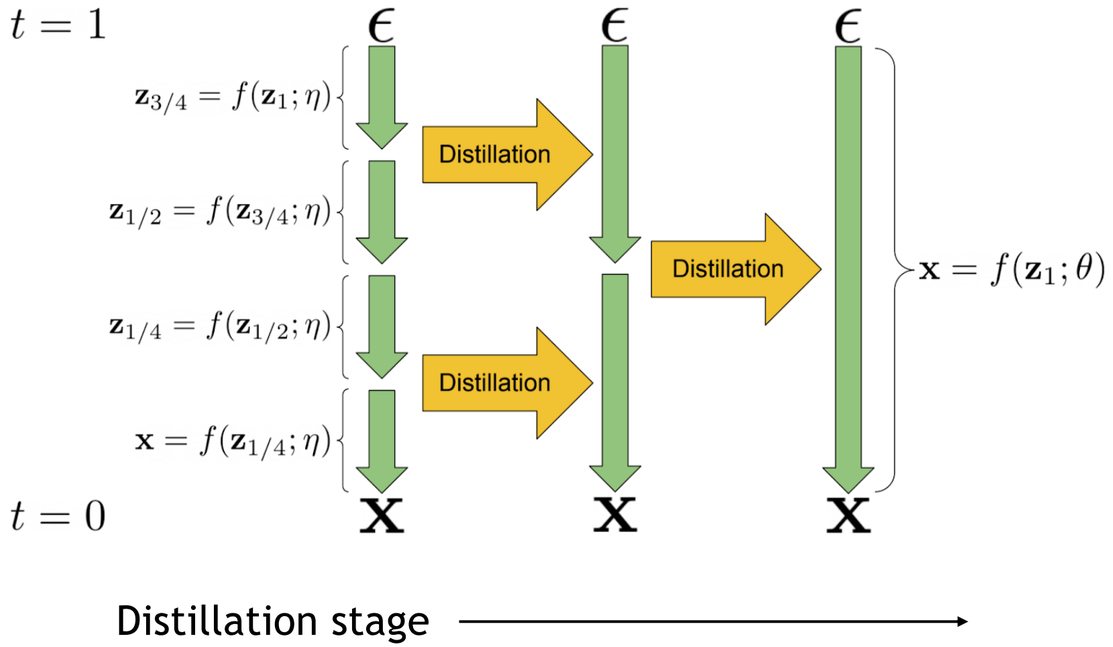
ODE Distillation



Can we train a neural network to directly predict $\mathbf{x}_{t'}$ given $\mathbf{x}_t$?

Progressive Distillation

- A student-teacher model for accelerating sampling.
- At each stage, a “student” model is learned to distill two adjacent sampling steps of the “teacher” model to one sampling step.
- At the next stage, the “student” model from the previous stage will serve as the new “teacher” model.



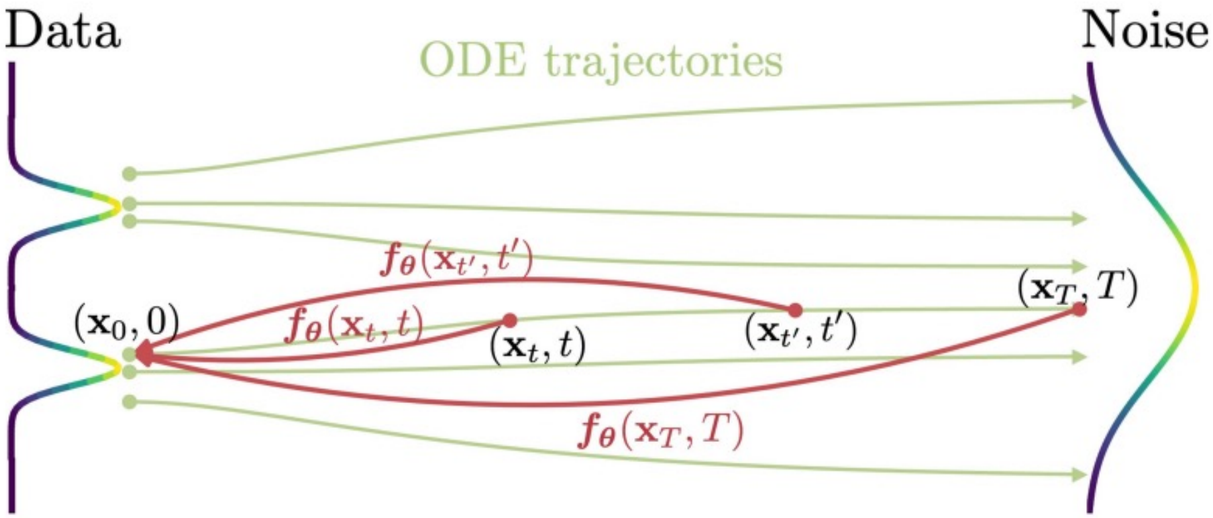
Salimans & Ho, “Progressive distillation for fast sampling of diffusion models”, ICLR 2022.

Progressive Distillation in Latent Space

(a) 2 denoising steps (b) 4 denoising steps (c) 8 denoising steps

Meng et al., "On Distillation of Guided Diffusion Models", CVPR 2023

Consistency Distillation



Points on the same trajectory should generate the same $\mathbf{x}_0$

Assume $f_\theta(\mathbf{x}_t, t)$ is the current estimation of $\mathbf{x}_0$

Basic idea:

- Find $\mathbf{x}_t$ and $\mathbf{x}_{t'}$ on a trajectory by solving generative ODE in $[t, t']$
- Minimize self-consistency loss: $\min_\theta \; ||f_{\text{EMA}}(\mathbf{x}_t, t) - f_\theta(\mathbf{x}'_t, t')||_2^2$

Song et al., Consistency Models, ICML 2023
Song & Dhariwal, Improved Techniques for Training Consistency Models, arXiv 2023

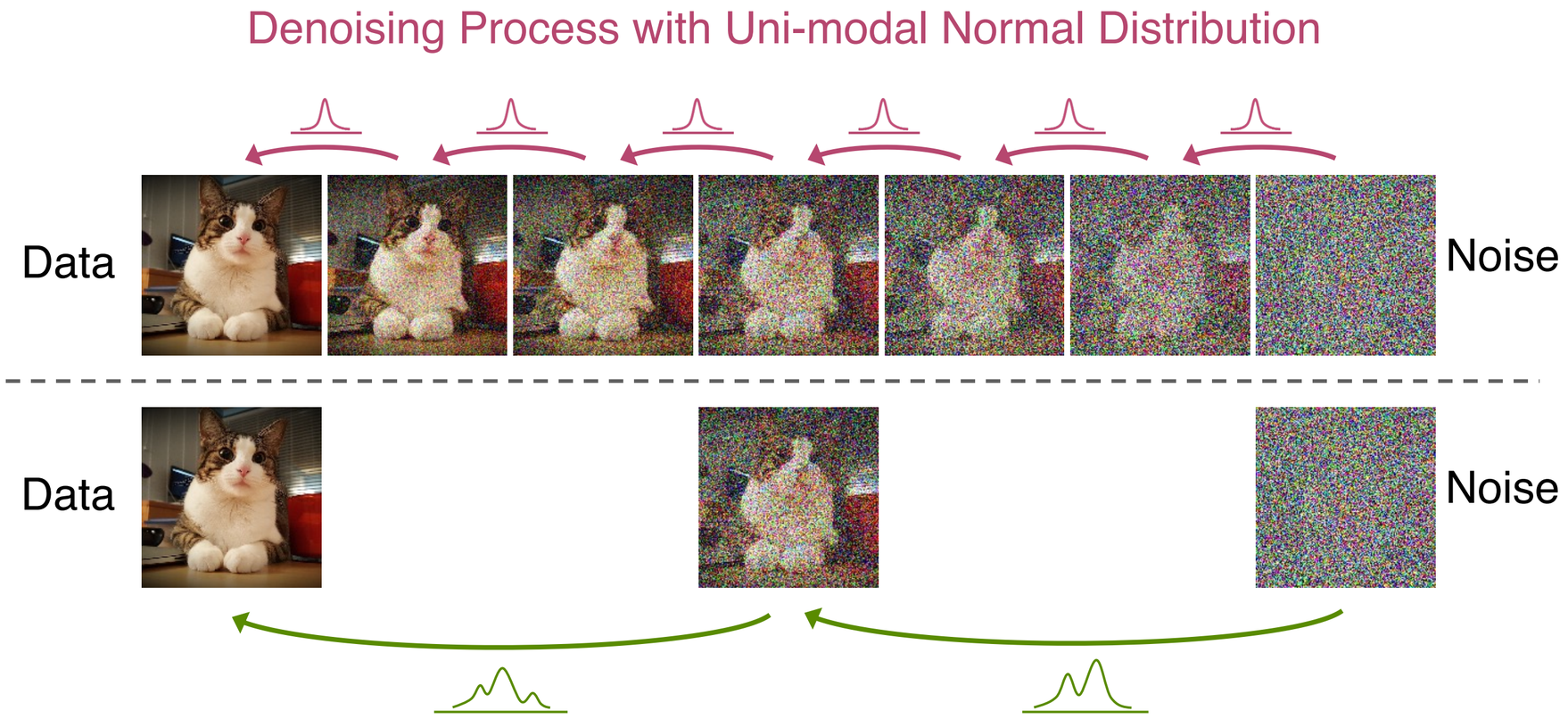
Latent Consistency Distillation

4-Steps Inference

Luo et al., "Latent Consistency Models", arXiv 2023

Adversarial methods for accelerated sampling

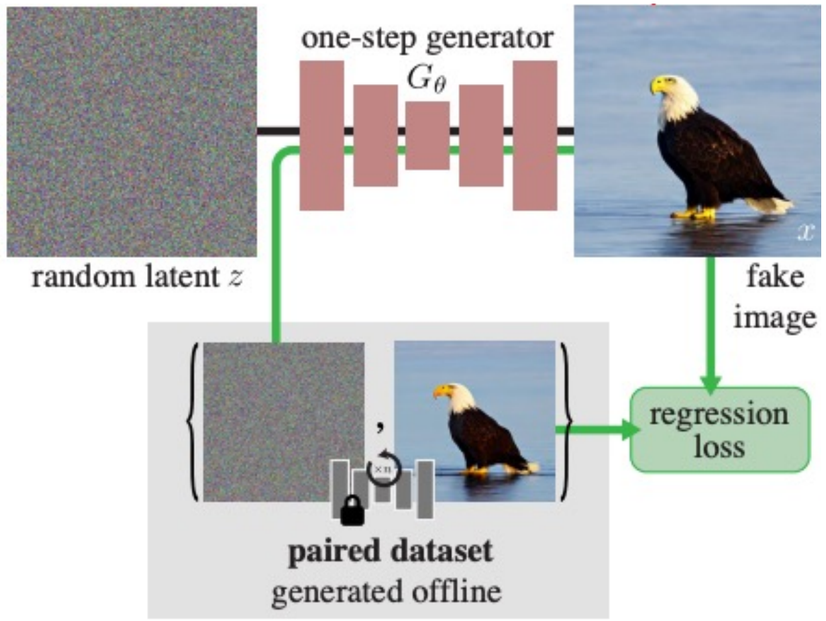
Normal assumption in denoising distribution holds only for small step



Xiao et al., “Tackling the Generative Learning Trilemma with Denoising Diffusion GANs”, ICLR 2022.
Gao et al., “Learning energy-based models by diffusion recovery likelihood”, ICLR 2021.

Distribution Matching Distillation

One-step Sample Generation

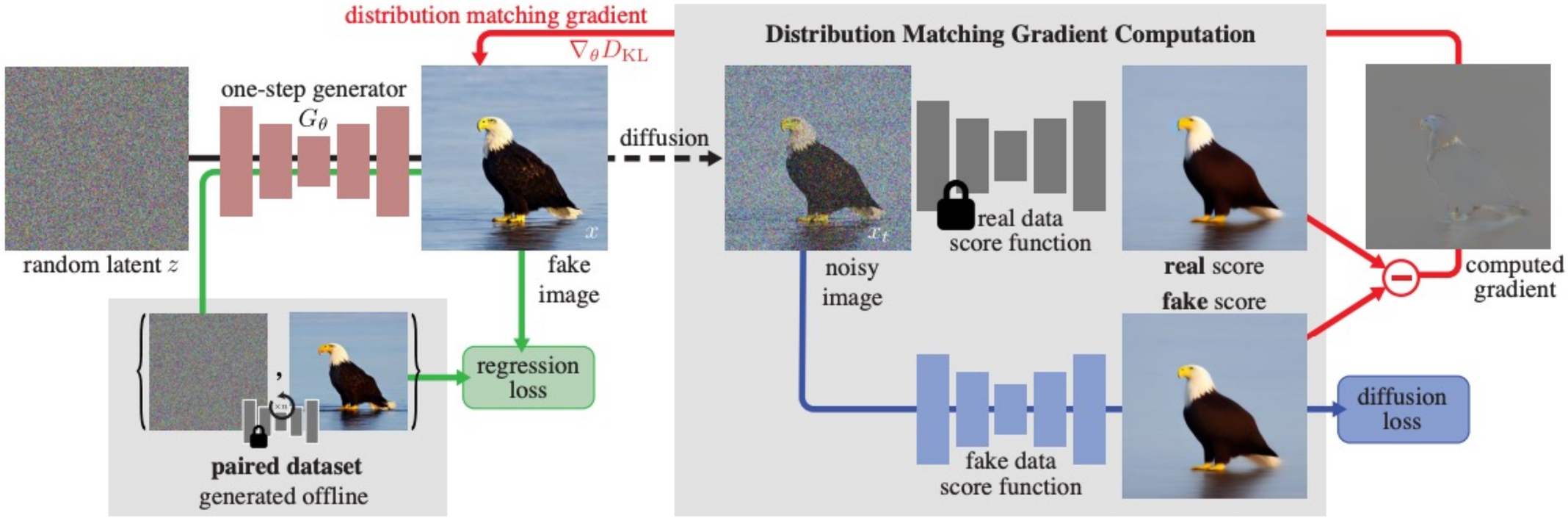


$$D_{KL}\left(p_{\text{fake}} \parallel p_{\text{real}}\right) = \mathop{\mathbb{E}}_{x\sim p_{\text{fake}}}\left(\log\left(\frac{p_{\text{fake}}(x)}{p_{\text{real}}(x)}\right)\right) \quad \Rightarrow \quad \nabla_\theta D_{KL} = \mathop{\mathbb{E}}_{\substack{z\sim\mathcal{N}(0;\mathbf{I}) \\ x=G_\theta(z)}}\Big[-\left(s_{\text{real}}(x) - s_{\text{fake}}(x)\right)\nabla_\theta G_\theta(z)\Big]$$

Yin et al., "One-step Diffusion with Distribution Matching Distillation", arXiv 2023

Distribution Matching Distillation

One-step Sample Generation

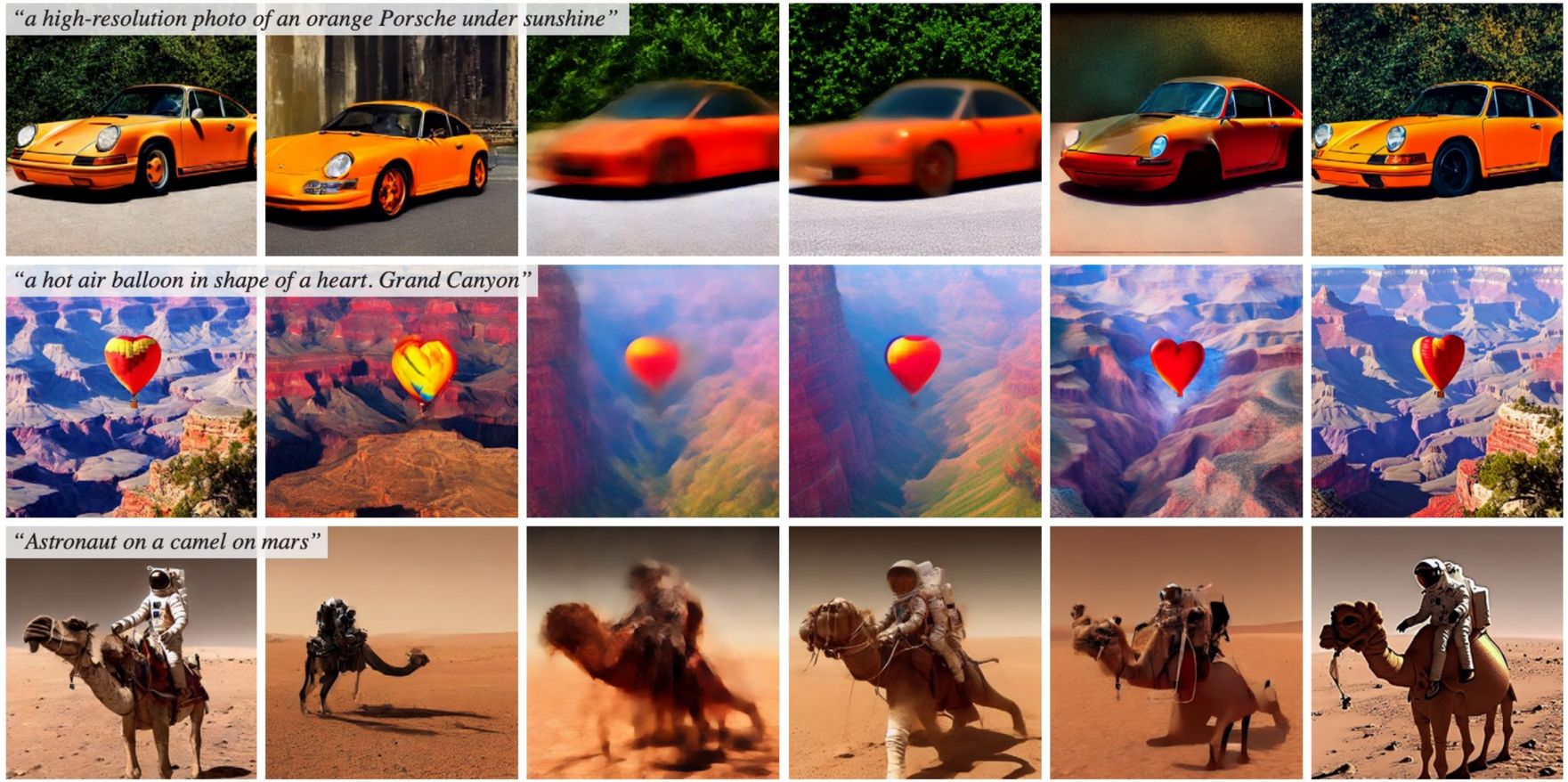


$$D_{KL}\left(p_{\text{fake}} \parallel p_{\text{real}}\right) = \mathop{\mathbb{E}}_{x\sim p_{\text{fake}}}\left(\log\left(\frac{p_{\text{fake}}(x)}{p_{\text{real}}(x)}\right)\right) \quad \Rightarrow \quad \nabla_\theta D_{KL} = \mathop{\mathbb{E}}_{\substack{z\sim\mathcal{N}(0;\mathbf{I}) \\ x=G_\theta(z)}}\Big[-\left(s_{\text{real}}(x) - s_{\text{fake}}(x)\right)\nabla_\theta G_\theta(z)\Big]$$

Yin et al., "One-step Diffusion with Distribution Matching Distillation", arXiv 2023

Distribution Matching Distillation

One-step Sample Generation



Yin et al., "One-step Diffusion with Distribution Matching Distillation", arXiv 2023

Other works

Adversarial:

- Xu et al., “UFOGen: You Forward Once Large Scale Text-to-Image Generation via Diffusion GANs”, arXiv 2023
- Sauer et al., "Adversarial Diffusion Distillation", arXiv 2023

Flow-based:

- Liu et al., “InstaFlow: One Step is Enough for High-Quality Diffusion-Based Text-to-Image Generation”, arXiv 2023

Distillation based:

- Zheng et al., “Fast Sampling of Diffusion Models via Operator Learning”, ICML 2023
- Luhman & Luman, “Knowledge distillation in iterative generative models for improved sampling speed”, arXiv 2021

Consistency:

- Berthelot et al., “TRACT: Denoising diffusion models with transitive closure time-distillation”, arXiv 2023
- Gu et al., “Boot: Data-free distillation of denoising diffusion models with bootstrapping”, ICML 2023

Agenda

End-to-End Training

Accelerated Sampling

Inverse Problems

Controllability & Manipulation

Solving Inverse Problems with Diffusion Models

Given noisy observation y, recover x_0:

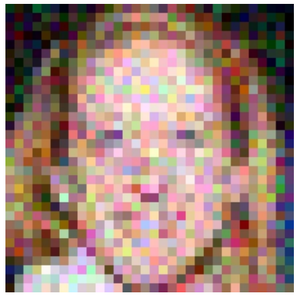
[Linear Degradation]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noisy observation] [Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]

Observations (Inputs) $\mathbf{y}$

Super-resolution



Inpainting

Deblurring

Solving Inverse Problems with Diffusion Models

Given noisy observation y, recover x_0:

[Linear Degradation]

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$$

[Noisy observation] [Noise, Gaussian stddev = $\sigma_{\mathbf{y}}$]

Diffusion Prior $p(x_0)$

Degradation Model $p(y|x_0)$

Solving inverse problems is

sampling from posterior $p(x_0|y)$

Reconstruction Guidance

Can we introduce a guidance term to solve inverse problems?

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

Conditional score | Prior diffusion model | This is not known!

We often approximate this with *Reconstruction Guidance*

$$-\nabla_{\mathbf{x}} ||\mathbf{y} - H\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)||_2^2$$

Denoising model

Ho et al., “Video Diffusion Models”, NeurIPS 2022
Chung et al., “Diffusion Posterior Sampling for General Noisy Inverse Problems”, ICLR 2023
Song et al., “Pseudoinverse-Guided Diffusion Models for Inverse Problems”, ICLR 2023

Variational Inference

We can also formulate solving inverse problems as variational inference:

$$\arg\min_q \; KL(q(x_0|y)||p(x_0|y)) = \arg\min_q \; \underbrace{\mathbb{E}_{q(x_0|y)}[-\log p(y|x_0)]}_{\text{Reconstruction Term}} + \underbrace{KL(q(x_0|y)||p(x_0))}_{\text{Regularization Term}}$$

Assuming $q(x_0|y) = \delta(x_0 - x)$, the MAP estimate is obtained by minimizing:

$$\min_x \underbrace{||y - Hx||_2^2}_{\text{Reconstruction Term}} + \underbrace{\mathbb{E}_{t,x_t}\left[\lambda_t(x - \hat{x}_\theta(x_t, t))^T x\right]}_{\text{Score Distillation Loss}}$$

Mardani et al., “A Variational Perspective on Solving Inverse Problems with Diffusion Models”, arXiv 2023
Feng et al., “Score-Based Diffusion Models as Principled Priors for Inverse Imaging”, ICCV 2023
Zhu et al., “Denoising Diffusion Models for Plug-and-Play Image Restoration”, CVPR Workshops 2023

More on inverse problems

Choi et al., “ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models”, ICCV 2021

Ho et al., “Video Diffusion Models”, NeurIPS 2022

Chung et al., “Diffusion Posterior Sampling for General Noisy Inverse Problems”, ICLR 2023

Song et al., “Pseudoinverse-Guided Diffusion Models for Inverse Problems”, ICLR 2023

Song et al., “Loss-Guided Diffusion Models for Plug-and-Play Controllable Generation”, ICML 2023

Kawar et al., “SNIPS: Solving Noisy Inverse Problems Stochastically”, NeurIPS 2021

Chung et al., “Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction”, CVPR 2022

Song et al., “Solving Inverse Problems in Medical Imaging with Score-Based Generative Models”, ICLR 2022

Kawar et al., “Denoising Diffusion Restoration Models”, NeurIPS 2022

Chung et al., “Improving Diffusion Models for Inverse Problems using Manifold Constraints”, NeurIPS 2022

Agenda

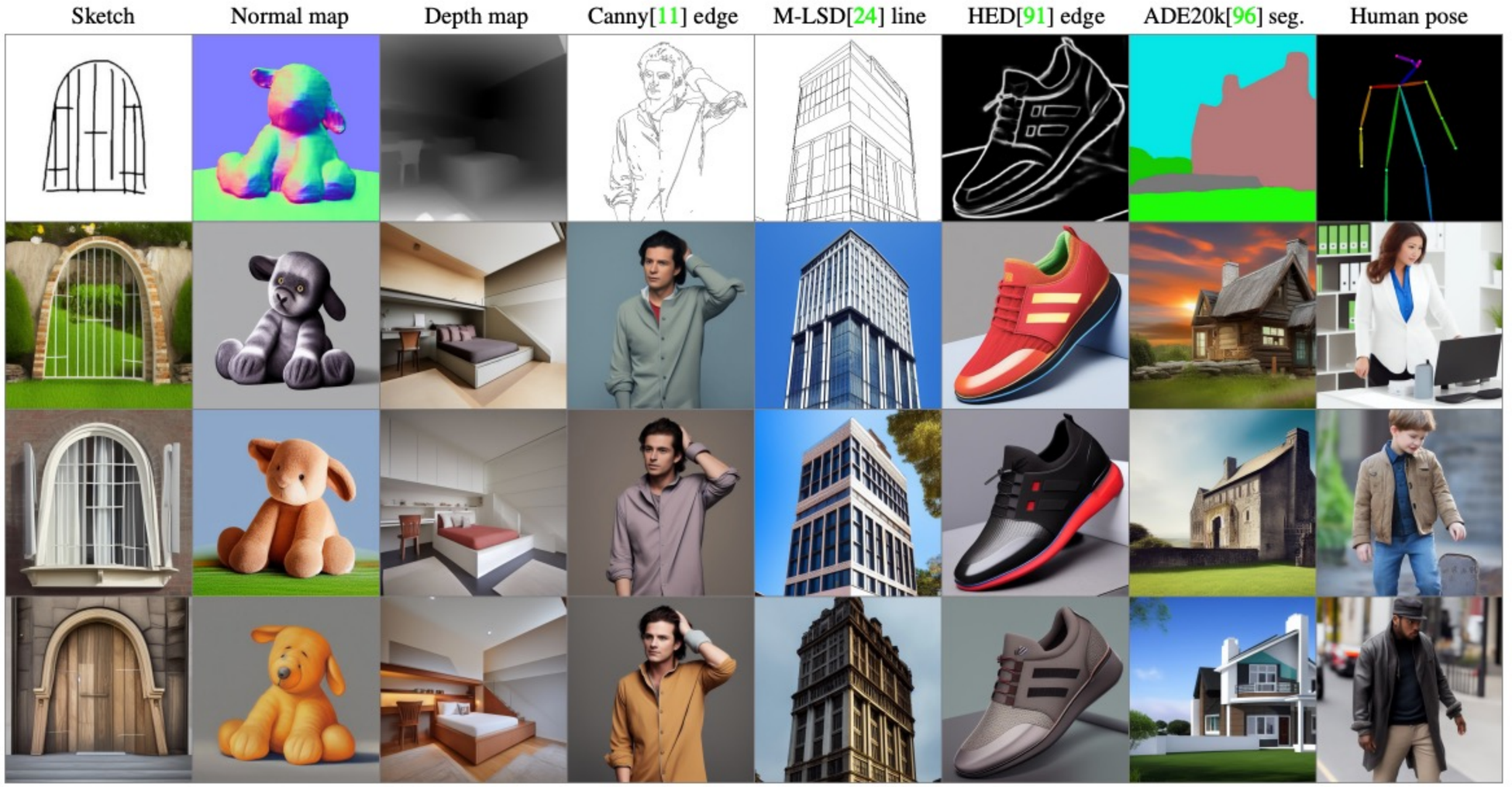
End-to-End Training

Accelerated Sampling

Inverse Problems

Controllability & Manipulation

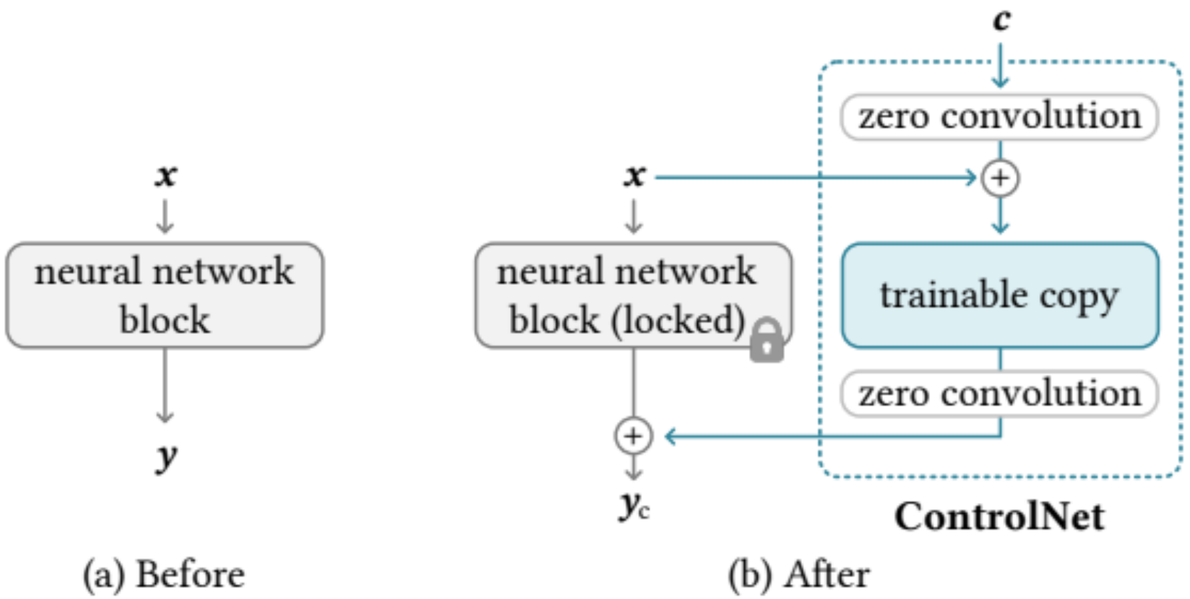
Controlling Latent Diffusion Models



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

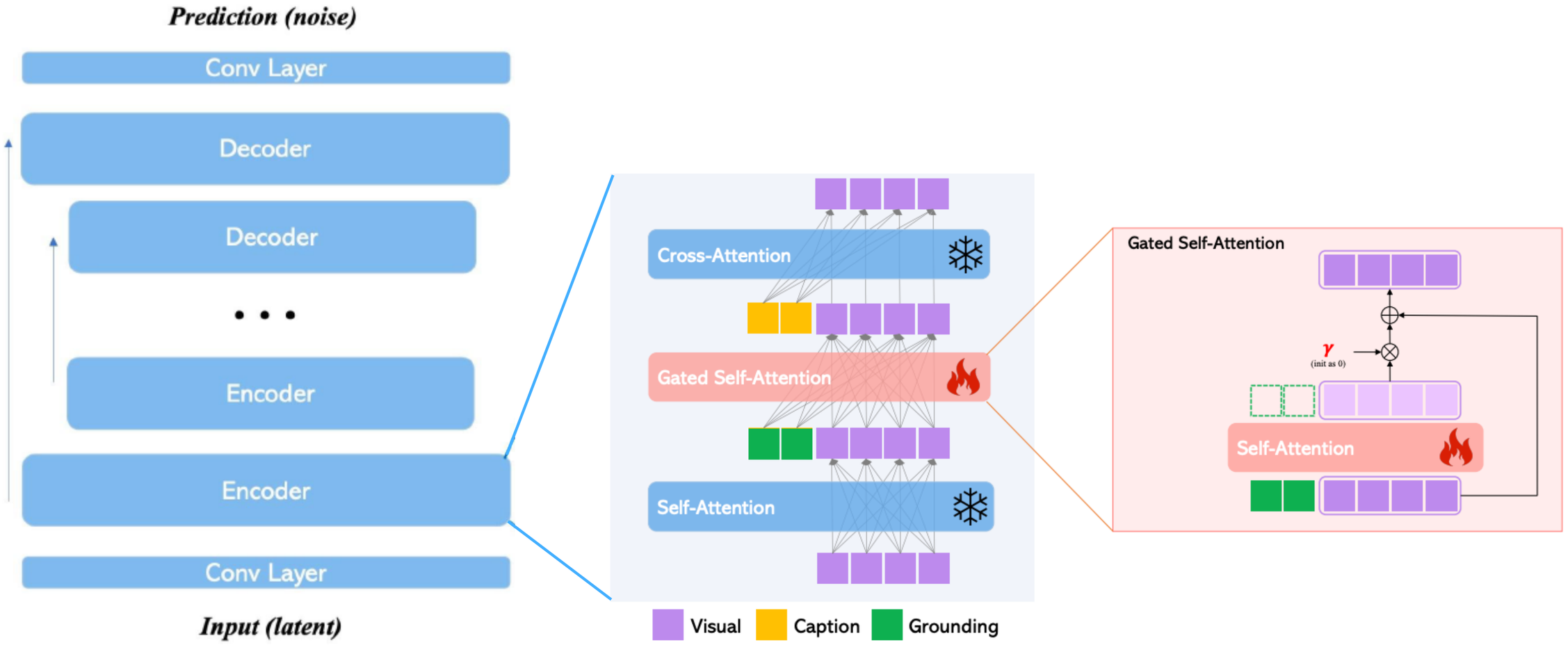
ControlNet

Turn unconditional diffusion models to conditional models:



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

Grounded Generation



Li et al., "GLIGEN: Open-Set Grounded Text-to-Image Generation", arXiv 2023

Grounded Generation

Caption: “golden hour, a pekingese is on the beach with an umbrella”
Grounded text: Pekingese, umbrella, sea

Caption: “a hen is hatching a huge egg”
Grounded text: hen, egg

Caption: “an apple and a same size dog”
Grounded text: apple, dog

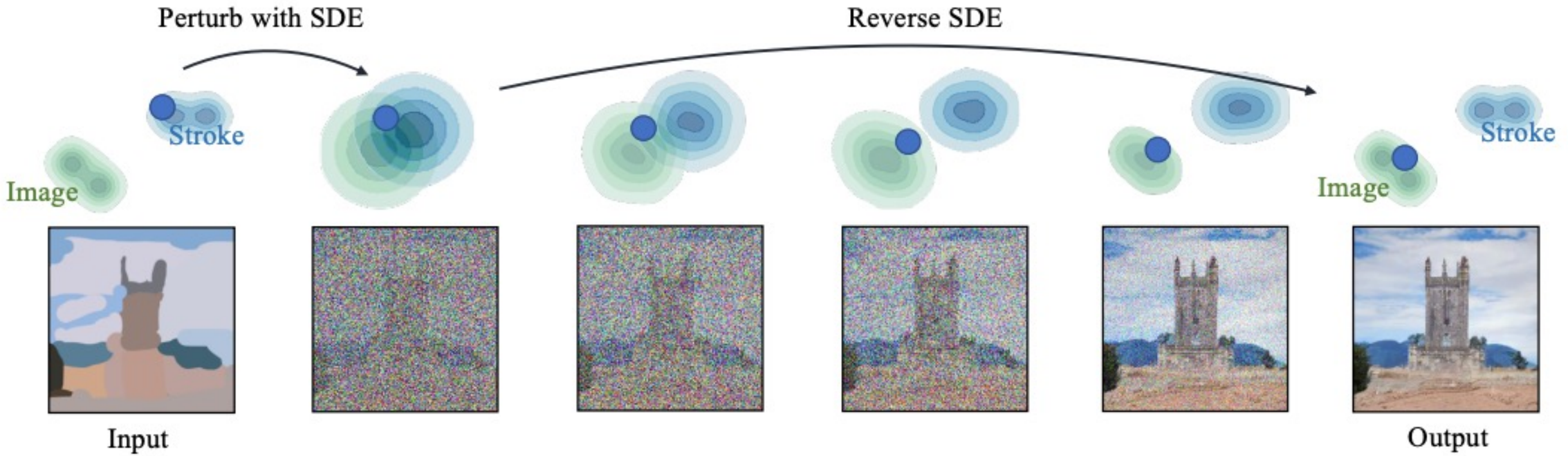
Li et al., "GLIGEN: Open-Set Grounded Text-to-Image Generation", arXiv 2023

Other Works

- Su et al., "Dual diffusion implicit bridges for image-to-image translation", ICLR 2023
- Couairon et al., "DiffEdit: Diffusion-based semantic image editing with mask guidance", ICLR 2023
- Kawar et al., "Imagic: Text-Based Real Image Editing with Diffusion Models", CVPR 2023
- Avrahami et al., “Blended Latent Diffusion”, SIGGRAPH 2023
- Avrahami et al., “Blended Diffusion for Text-driven Editing of Natural Images”, CVPR 2023
- Ge et al, “Expressive Text-to-Image Generation with Rich Text”, ICCV 2023.

SDEdit

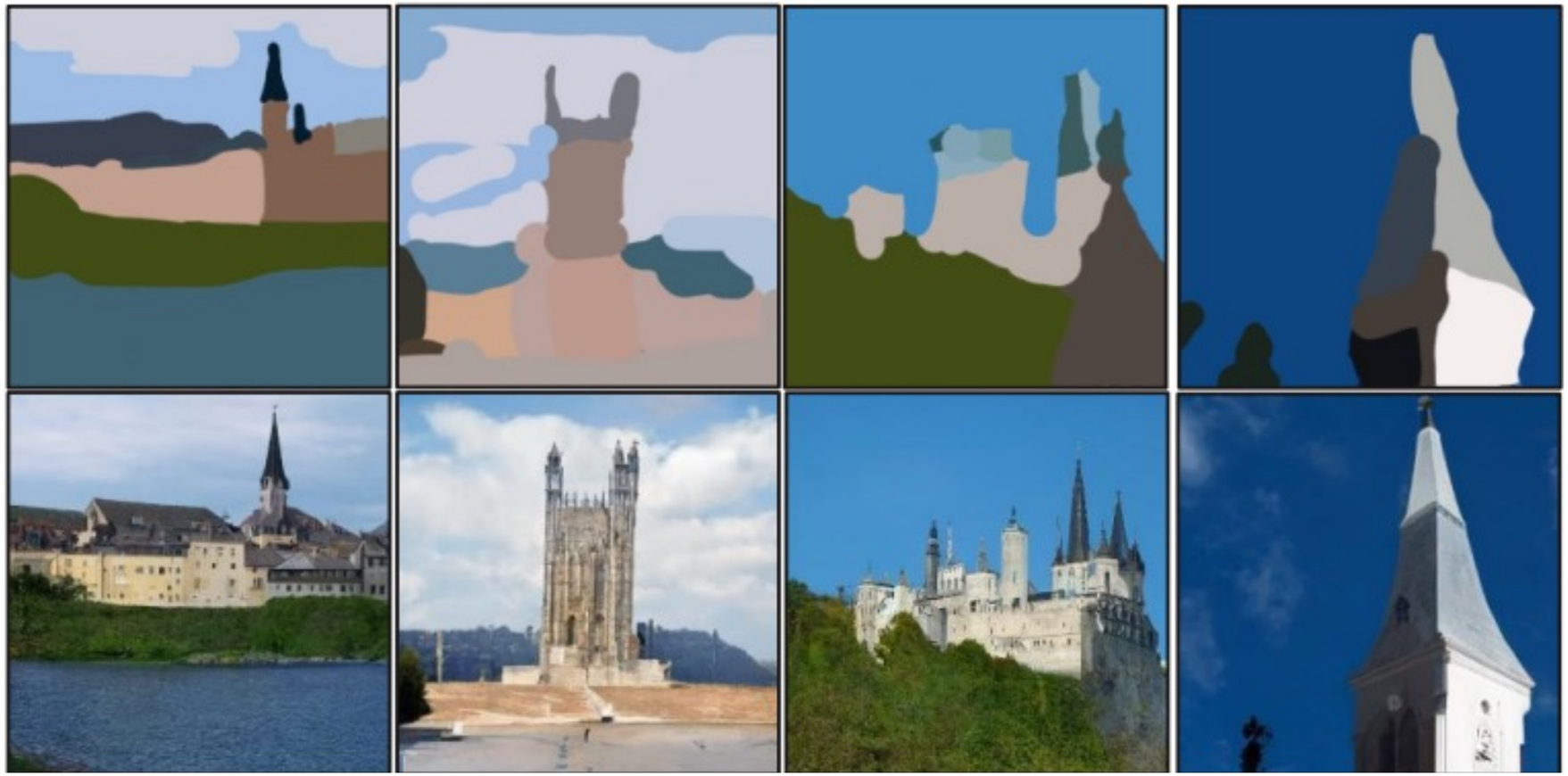
Use pre-trained diffusion models for editing



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

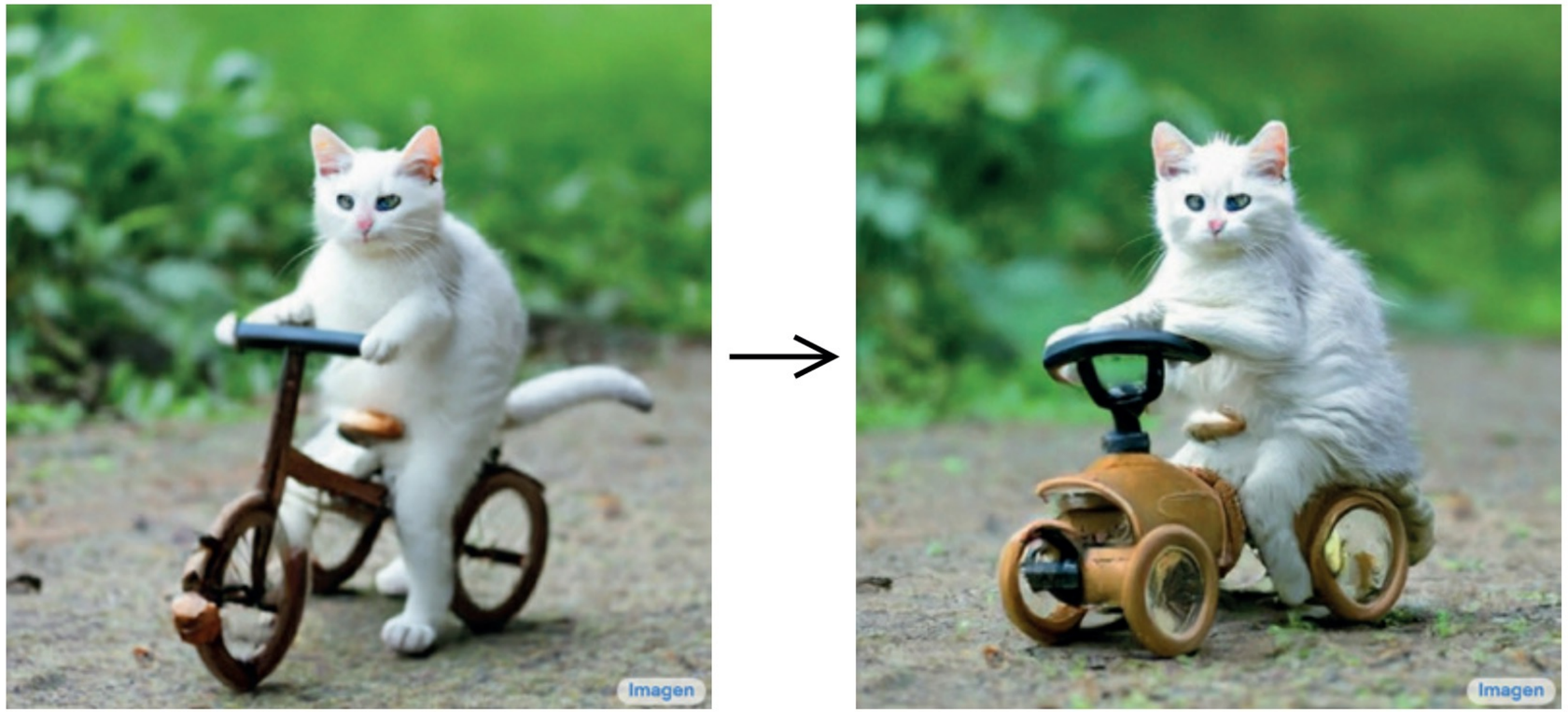
SDEdit

Use pre-trained diffusion models for editing



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

Prompt-to-Prompt

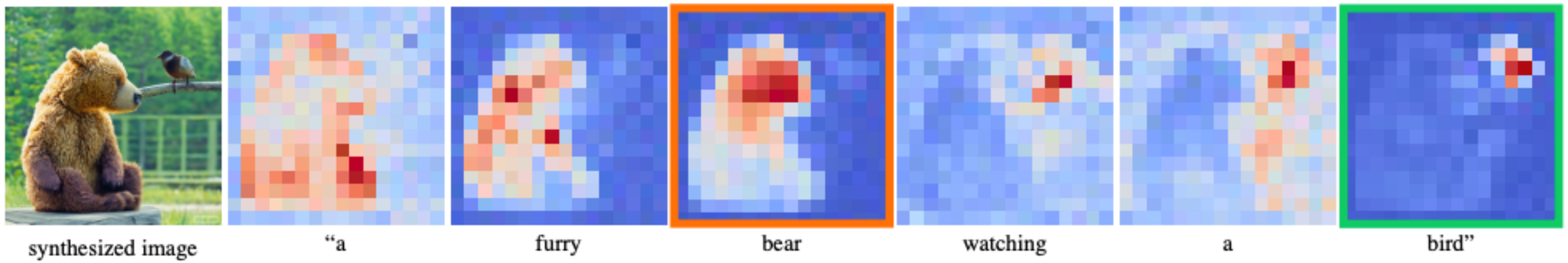


“Photo of a cat riding on a bicycle.”

car

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

Prompt-to-Prompt



The spatial layout of the generated image depends on the cross-attention maps.

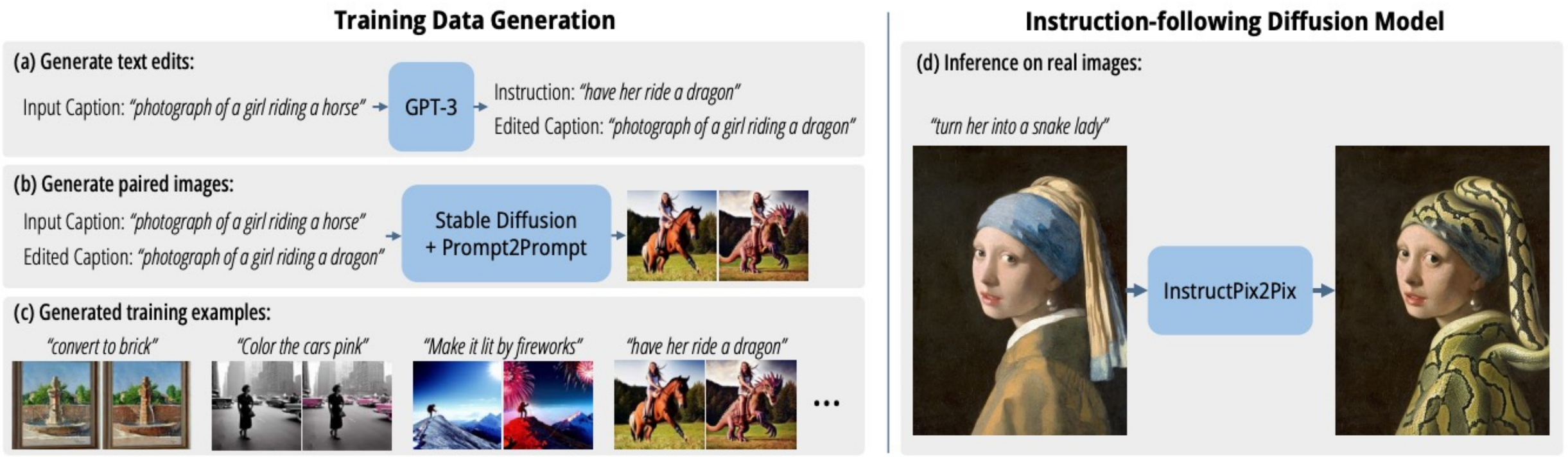
Reuse cross-attention maps between two prompts.

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

Prompt-to-Prompt

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

InstructPix2Pix



Brooks et al., "InstructPix2Pix: Learning to Follow Image Editing Instructions", CVPR 2023

Summary

End-to-end training enables the training of latent diffusion models and their autoencoding architecture jointly.

Acceleration diffusion models in latent space seems to be more promising.

One can consider inverse problems as a variational inference problem with diffusion priors.

Latent diffusion models are now foundation models that allow many downstream applications with limited training.

Thanks!

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

@ArashVahdat

Today's Program

Title	Speaker	Time
Part (1): Introduction to Latent Diffusion Models *Diffusion models, autoencoding, compression, latent diffusion, architectures, image generation*	Karsten	40 min
Part (2): Advanced Design and Controllability *End-to-end training, maximum likelihood, accelerated sampling, distillation, control and editing*	Arash	40 min
Part (3): Latent Diffusion Models beyond Image Generation *Video generation, 3D object and scene synthesis, segmentation, language & molecule generation*	Ruiqi	40 min
Panel Discussion: *Robin Rombach, Durk Kingma, Chenlin Meng, Sander Dieleman, Ying Nian Wu*	Panelists	30 min

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

Part (3):
Latent Diffusion Models beyond Image Generation

Motivation

- A promise of LDMs is that it is agnostic to data modalities, as long as we can find an auto-encoder that lifts the data to a continuous latent space.
- In this part, we will see how LDMs can be applied to modalities beyond images.

Contents

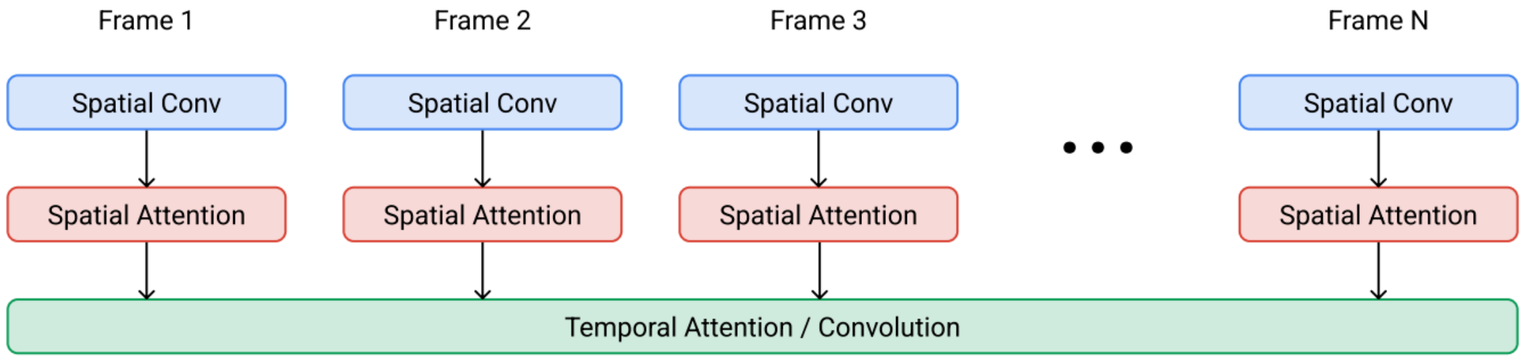
Not intended as a complete review of all recent work!

Contents

Video Diffusion Models

2D Diffusion models with temporal connections

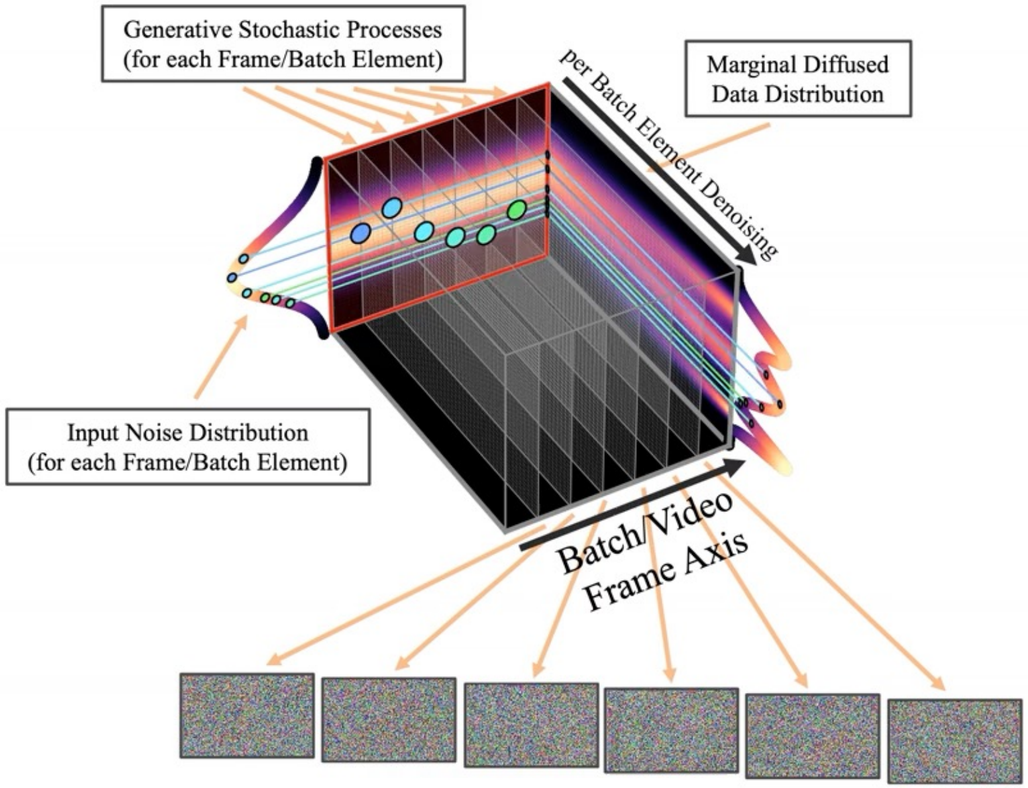
- Image → Video: Just add another dimension to the data tensor
- Image architecture: 2D UNet
- Video architecture: 3D Unet, space-time separable
    - repeat the 2D UNet over frames
    - additional layers to mix over time using attention or convolution



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

Latent Video Diffusion Models

Align your latents: finetune an image LDM to a video LDM

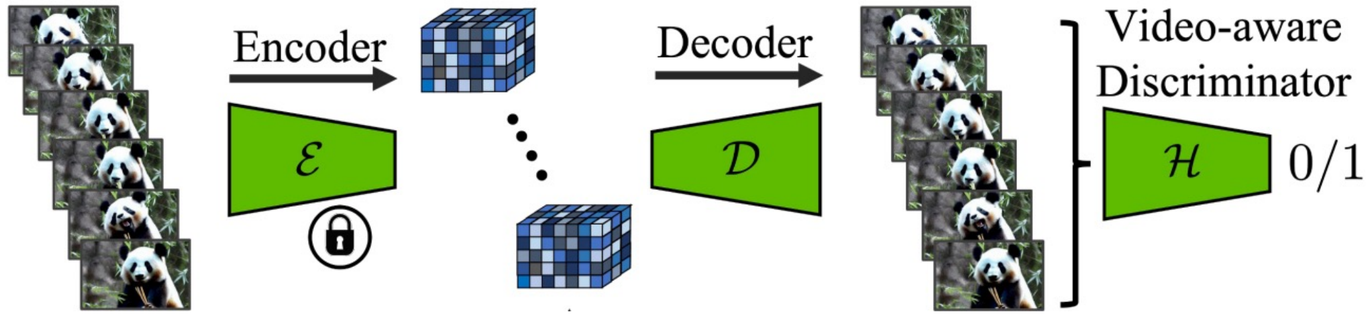


Before temporal video fine-tuning,
different batch samples are independent.

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

Align Your Latents

Encoder fixed; Decoder finetuning

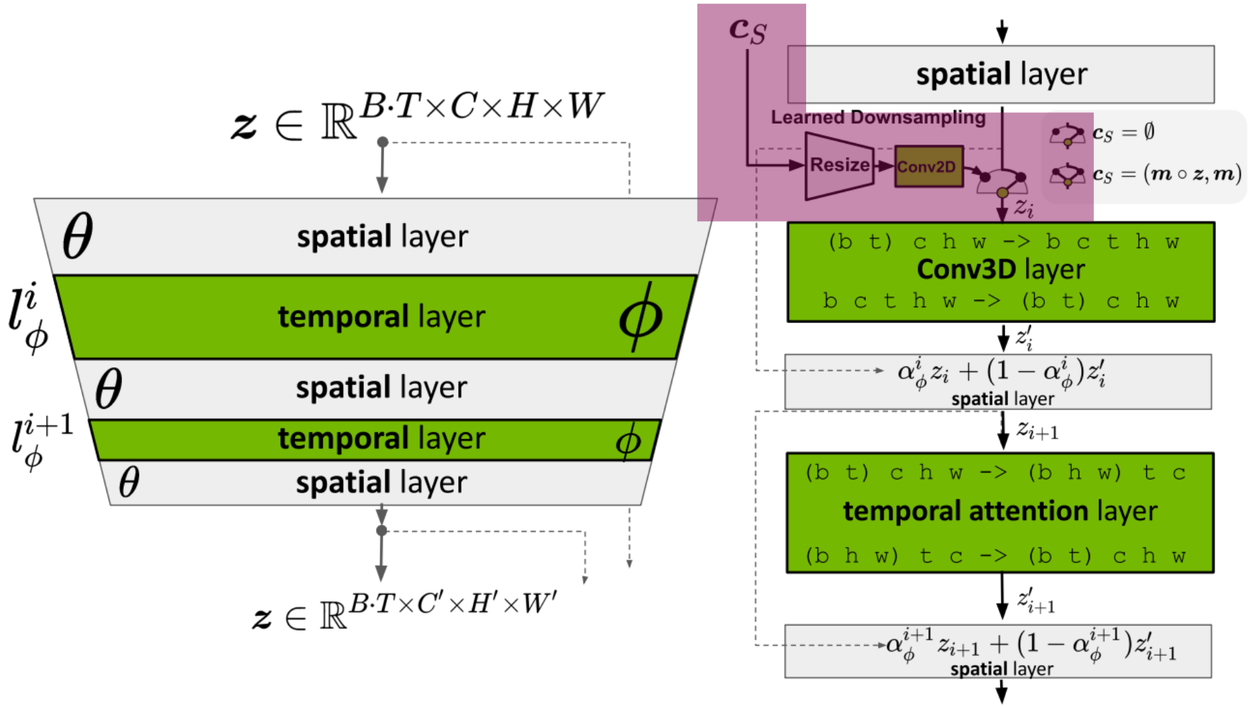


- To maintain the latent distribution captured by the image LDM, the encoder remains as an image encoder and frozen.
- The decoder is finetuned with additional temporal layers, and a patch-wise temporal discriminator.
- Critical for outputting temporal consistent videos without flickering.

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

Align Your Latents

Diffusion model part: only finetune new temporal layers



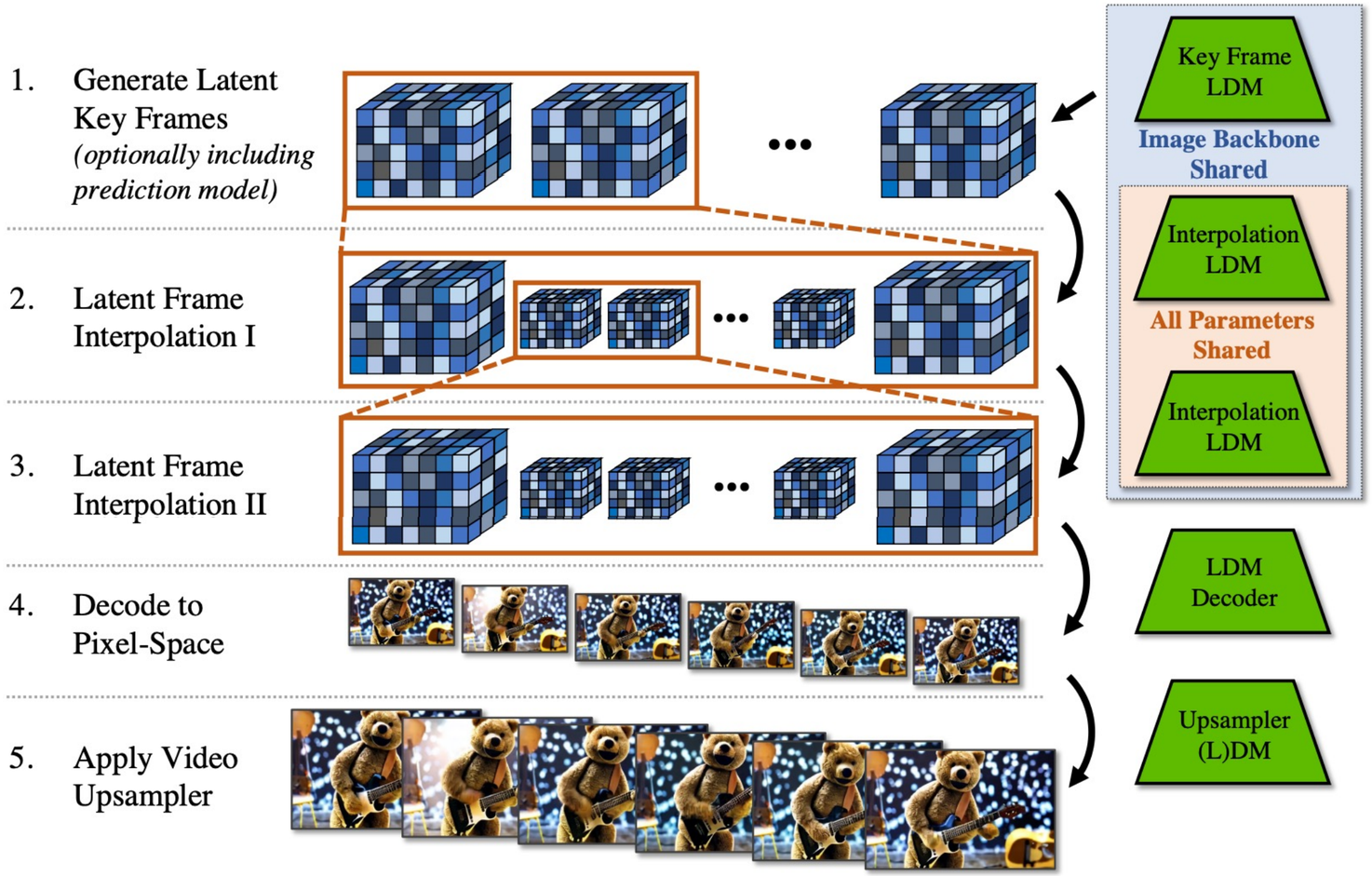
- After every spatial layer, a temporal layer is inserted, either (1) a residual block with 3D conv, or (2) a temporal attention layer.
- Only new temporal layers are learned. Spatial layers are frozen.
- Additional context can be added, to support autoregressive video generation.

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

Align Your Latents

Video LDM cascaded pipeline for high res/fps generation

- Similar to pixel-space diffusion models, we can build cascaded pipeline for video LDMs.
- A key frame LDM with low fps and spatial resolution.
- Two temporal super-res models sharing weights.
- A spatial video up-sampler finetuned from the image up-sampler LDM.



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

Stable Video Diffusion

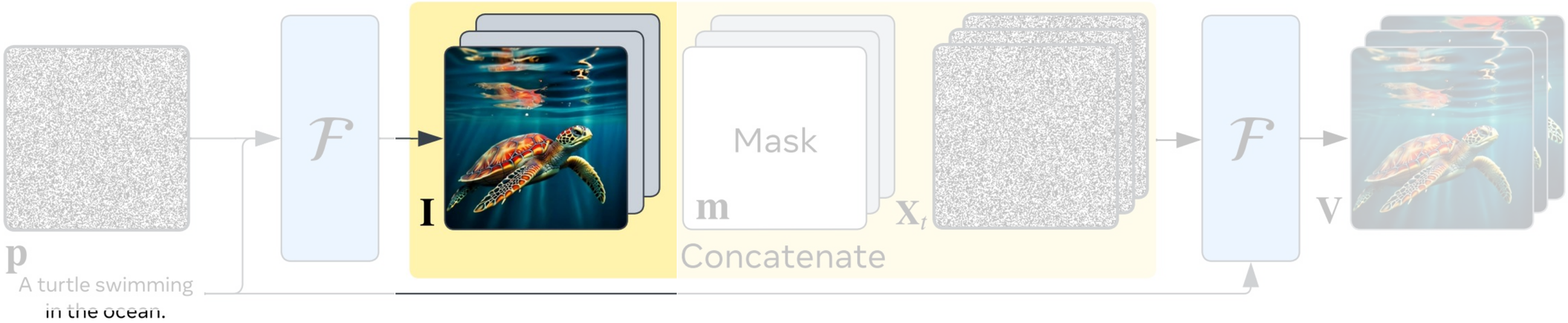
Scaling up align your latents; detailed recipe

- Finetune the whole model, instead of only temporal layers.
- Three stages: text-to-image pretraining, video pretraining, and high-quality video finetuning.
- Well-curated dataset helps.
- Shifting the noise schedule is important.

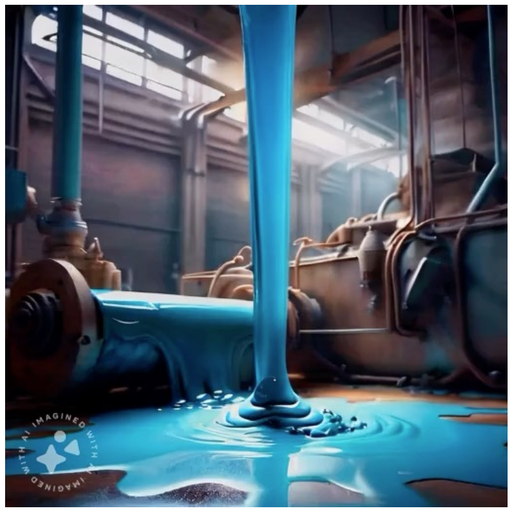
Blattmann et. al, Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, 2023

EMU Video

Factorizing the task to two steps: text-to-image and image-to-video



- No cascaded pipeline is required.



Girdhar et. al, [EMU VIDEO: Factorizing Text-to-Video Generation by Explicit Image Conditioning](), 2023

Open Problems for Video LDMs (1/2)

- How to train and generate long high-res video sequences, given the current hardware constraint?
  - Efficient model architecture
  - Higher compression rate
  - Divide-and-conquer: autoregressive generation, cascaded pipeline, text->image->video
- How to capture motion in video?
  - Current video LDMs tend to generate static scenes, lacking complex and large motions.
  - The model capacity is still largely devoted to generating realistic images per frame.
  - How to build good inductive bias into the model, to put more emphasis on capturing motion?
- How to maintain content consistency?
  - How to generate a consistent and persistent character over the course of long-range generation?
  - How to maintain scene consistency under different camera poses and lighting?

Open Problems for Video LDMs (2/2)

- How to improve controllability in video LDMs?
  - Fine-grained temporal / motion control?
  - More conditioning information => ease the burden of the model, increase interpretability.
  - Possible conditioning signals: first frame / key frames, fps, camera poses, human sketches, optical flow…
- How to make generation faster?
  - Video is of much higher dimension than images.
  - Divide and conquer approaches make sampling even slower.
- How to evaluate video LDMs?
  - Metrics emphasizing more on motions and frame consistency?
  - Alignment with text?

Contents

- LDM for 3D representations.
- Leveraging 2D LDMs for 3D generation.

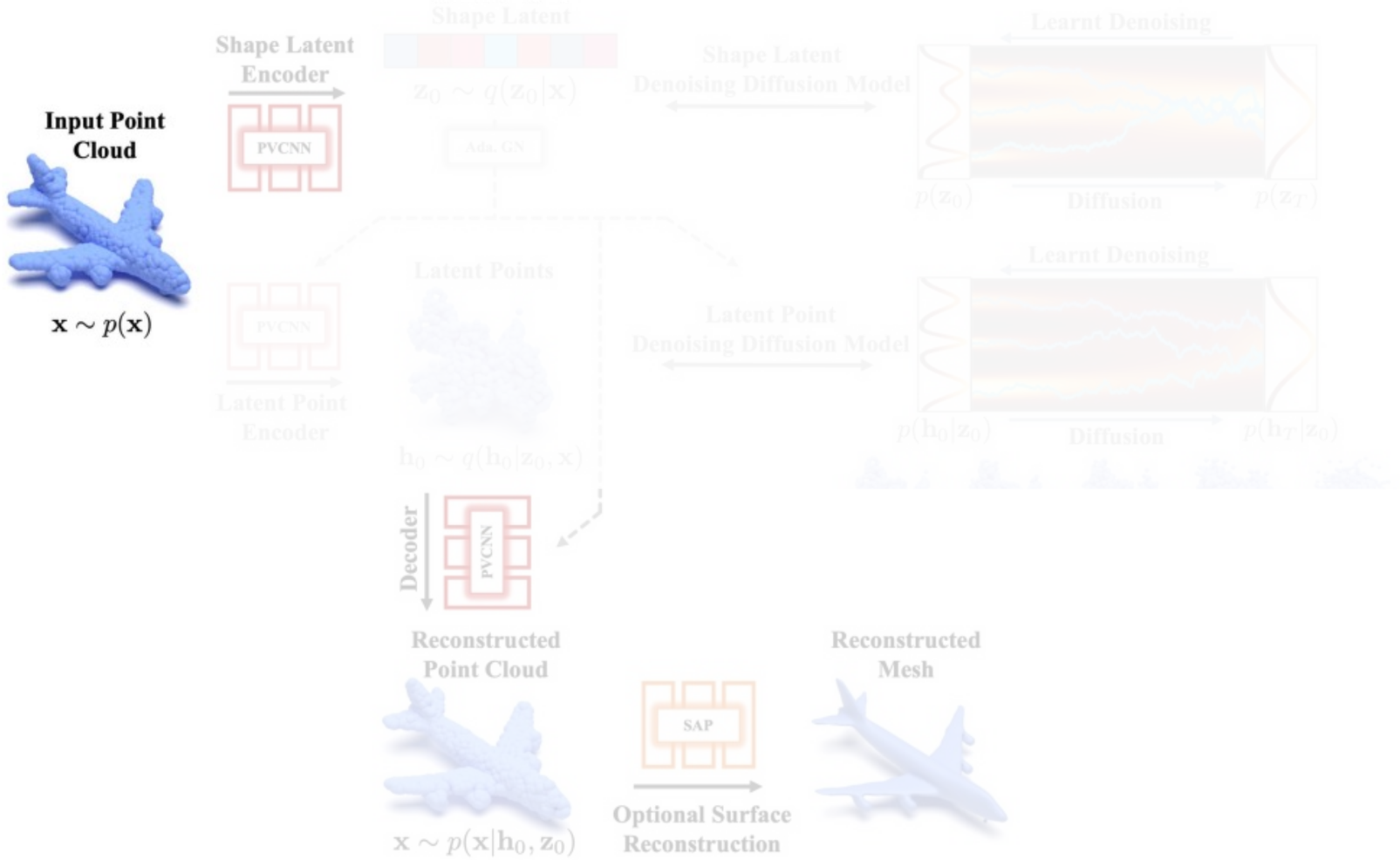
Contents

- LDM for 3D representations.
- Leveraging 2D LDMs for 3D generation.

LION

Point cloud generation with LDM

- Encoder / decoder: Point-Voxel CNNs (PVCNN)
- Hierarchical latents:
  - Vector shape latents
  - Latent points
- Two-stage training, no GAN loss.



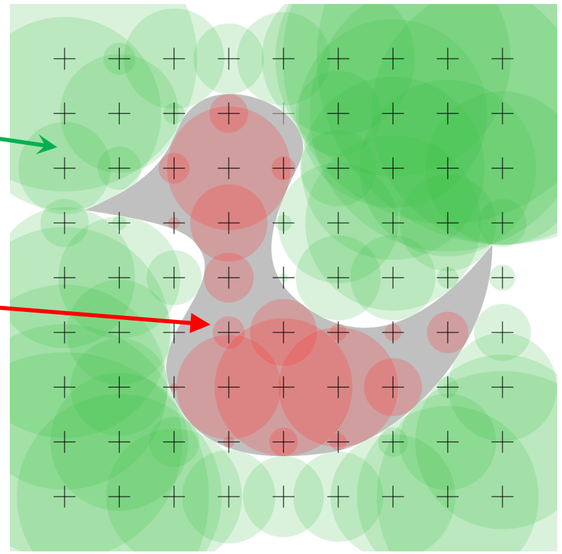
Zeng et. al, LION: Latent Point Diffusion Models for 3D Shape Generation, NeurIPS 2022

Signed Distance Functions (SDF)

- SDF is a function representation of a surface.
- For each location x, |SDF(x)| = smallest distance to any point on the surface.

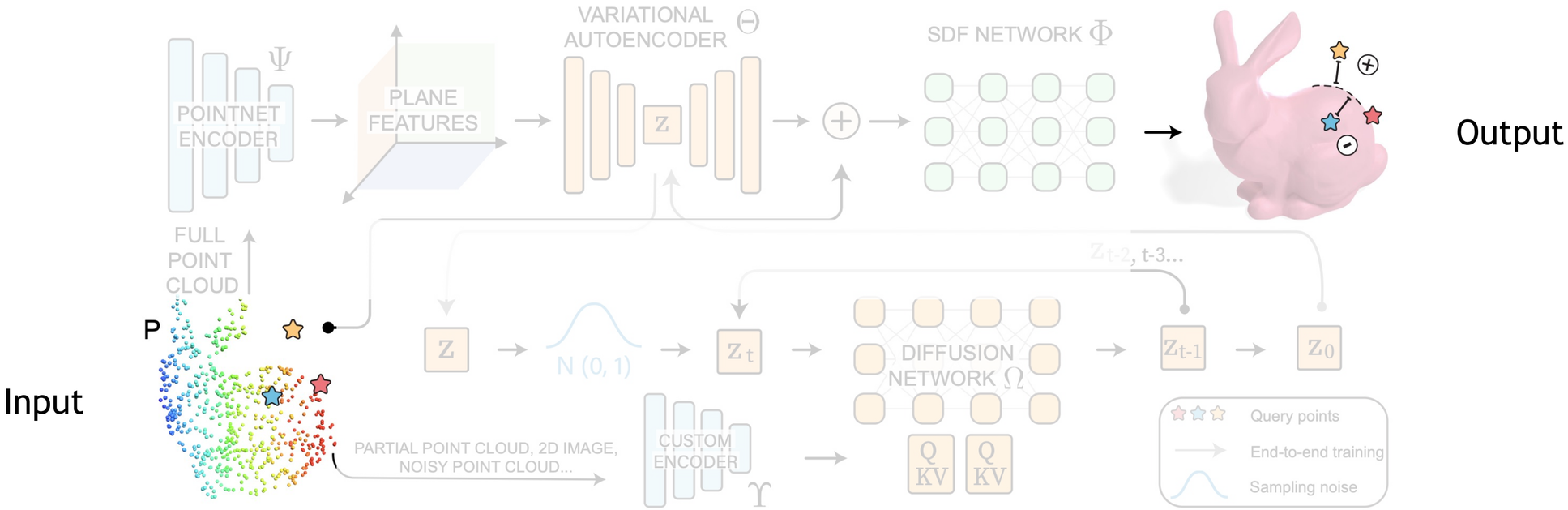
Green: outside of surface, positive

Red: inside of surface, negative



Diffusion-SDF

SDF generation with LDM



- Input: a set of query points; Output: SDF of the query points
- Encoder: pointnet + latent encoder
- Decoder: latent decoder + SDF network
- Diffusion models on vector z. Optionally conditioning information through cross attention.

Chou et. al, Diffusion-SDF: Conditional Generative Modeling of Signed Distance Functions, ICCV 2023

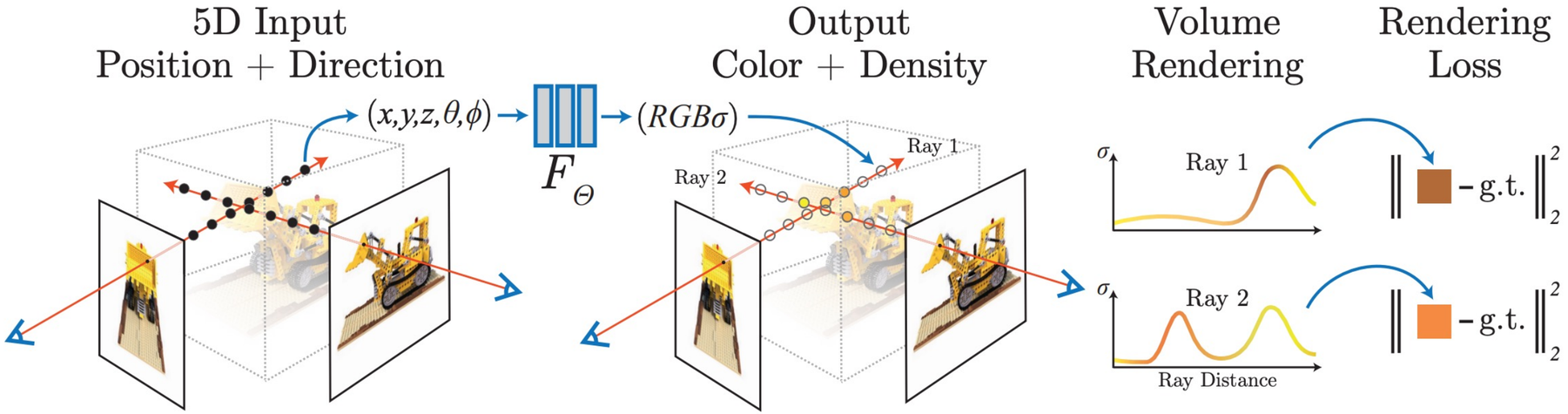
Neural Radiance Fields (NeRF)

An implicit neural representation

- For a 3D scene, given (world coordinates + viewing direction), learn an MLP that outputs RGB + density.

$$(x,y,z,\theta,\phi) \rightarrow \boxed{\quad} \rightarrow (RGB\sigma)$$
$$F_\Theta$$

- To generate images, sampling (world coordinates + viewing direction) along camera rays, and use volume rendering (differentiable)

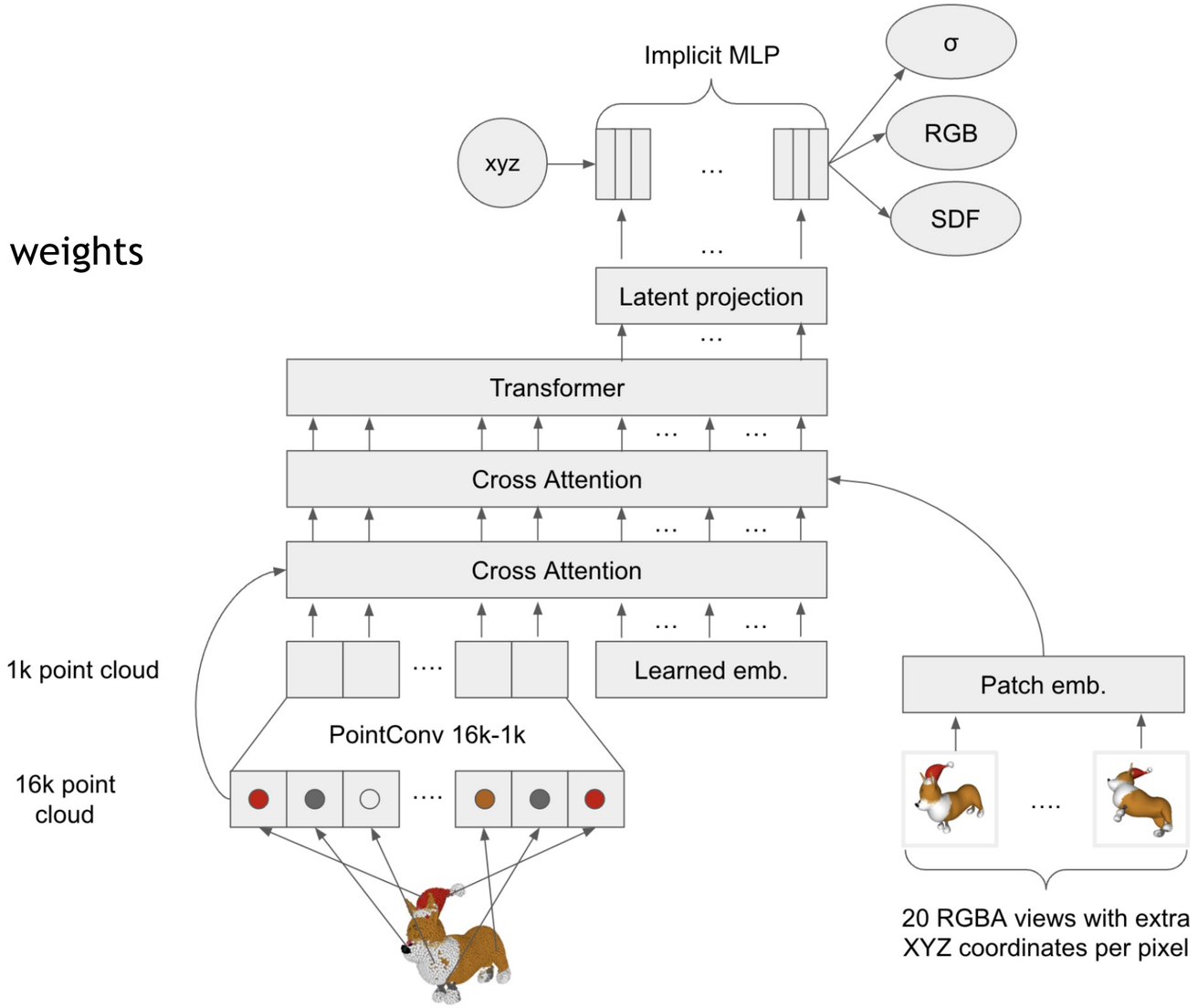


Mildenhall et. al, NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV 2020

Shap·E

NeRF / STF generation with LDM

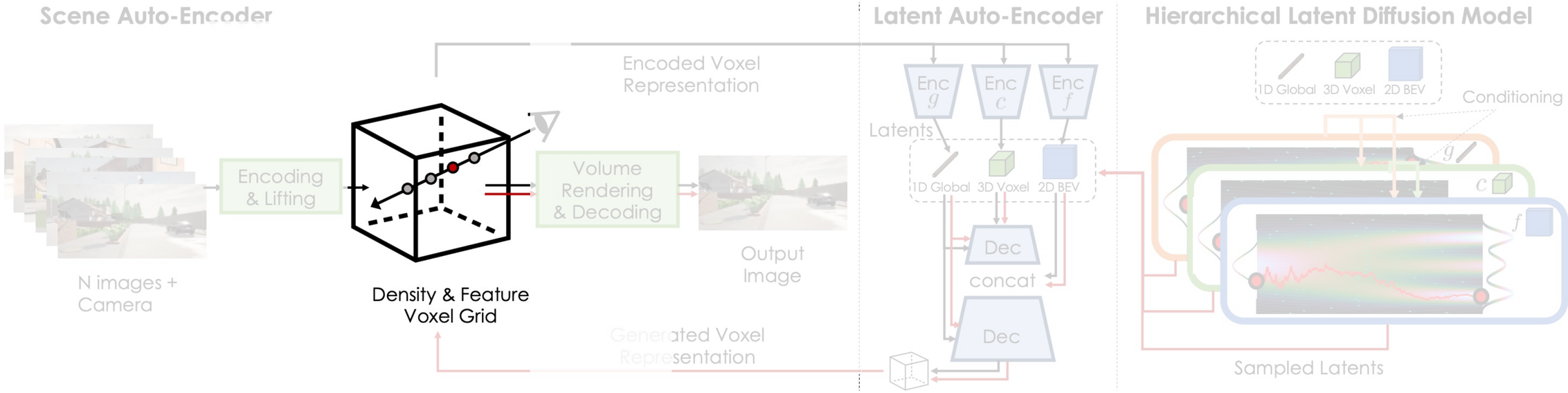
- Input: point cloud + rendered views;
  Output: novel view images
- Encoder: PointConv + transformer
- Decoder: projection layer outputting MLP weights of NeRF + NeRF rendering
- Diffusion model on the latents before the projection.



Jun et. al, [Shap·E: Generating Conditional 3D Implicit Functions]

NeuralField-LDM

3D scene generation with voxel grid + LDM



- An alternative 3D representation to NeRF (fully implicit) is voxel grid.
- Input: A set of (images + camera poses); Output: novel view images.
- Scene autoencoder: input -> voxel grid -> output.
- Latent autoencoder: voxel grid -> hierarchical latents -> voxel grid.
- Hierarchical latent diffusion models.

Kim et. al, NeuralField-LDM: Scene Generation with Hierarchical Latent Diffusion Models, CVPR 2023

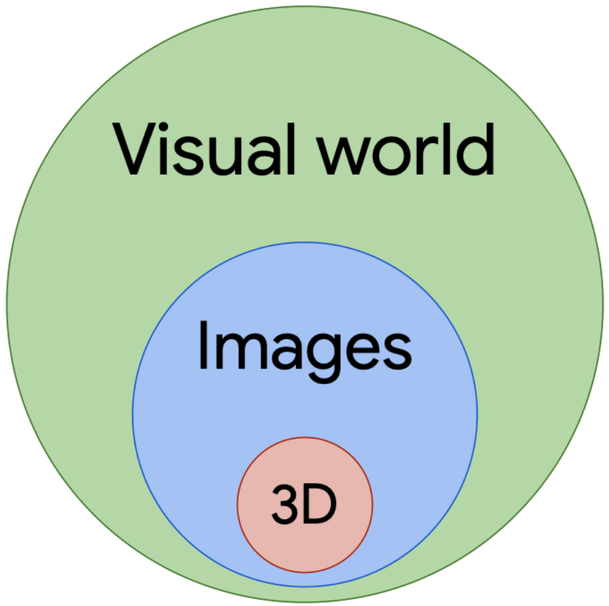
Contents

- LDM for 3D representations.
- Leveraging 2D LDMs for 3D generation.

2D LDMs for 3D Generation

motivation

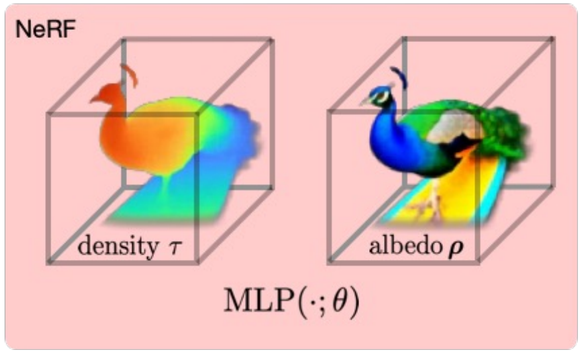
- Typical deep learning recipe: big model trained on big datasets.
- However, high quality 3D data are quite limited, especially compared with image / video data.

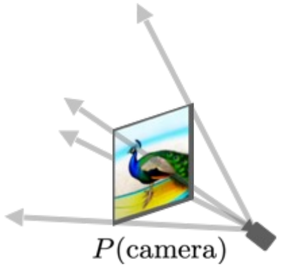


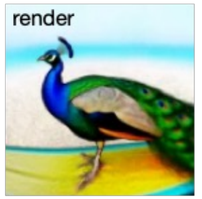
Use 2D priors for 3D generation!

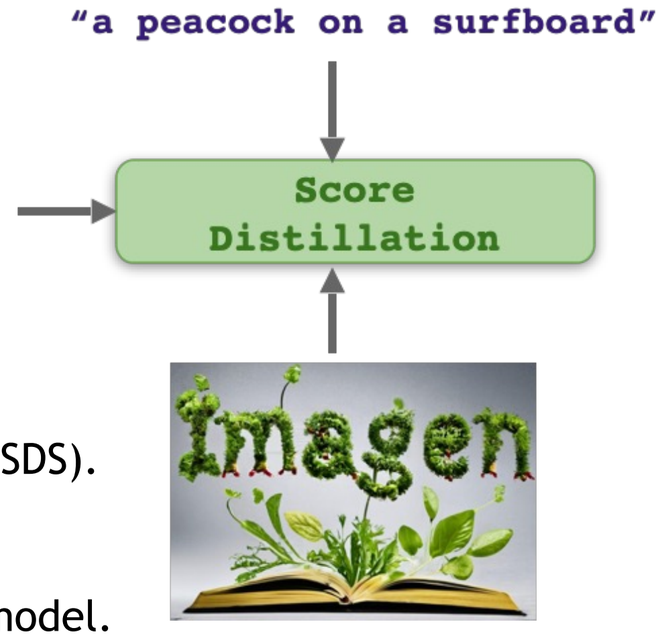
DreamFusion

Leverage 2D diffusion models for text-to-3D generation









- DreamFusion = NeRF + text-to-image diffusion model + score distillation sampling (SDS).
- The pretrained text-to-image diffusion model serves as a "critic".
- The NeRF learns to render images that can receive high scores from the diffusion model.
- Readily applicable with 2D latent diffusion models.

Poole et. al, DreamFusion: Text-to-3D using 2D Diffusion, ICLR 2023

DreamFusion

Challenge: multi-view consistency

Multi-face Janus Problem

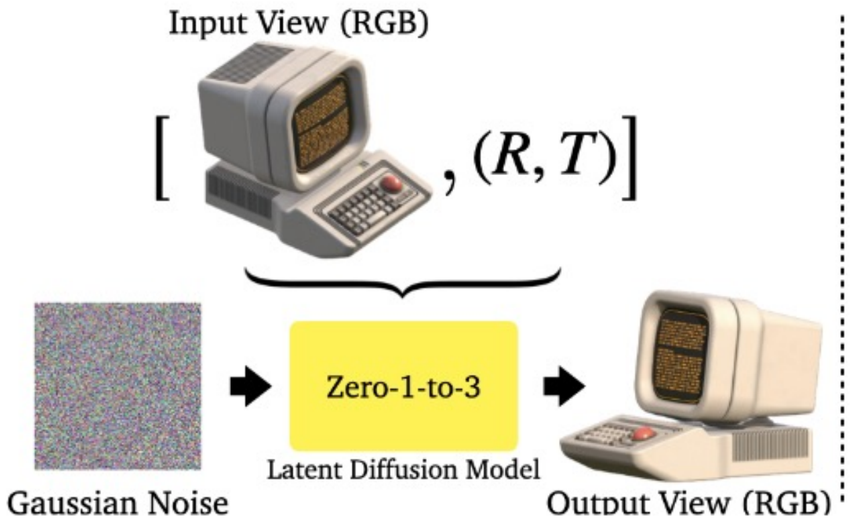
Content Drift Problem

- A 2D diffusion model can give high score to multiple plausible samples at a certain angle.
- Such uncertainty leads to inconsistent 3D generation.
- How to reduce uncertainty?

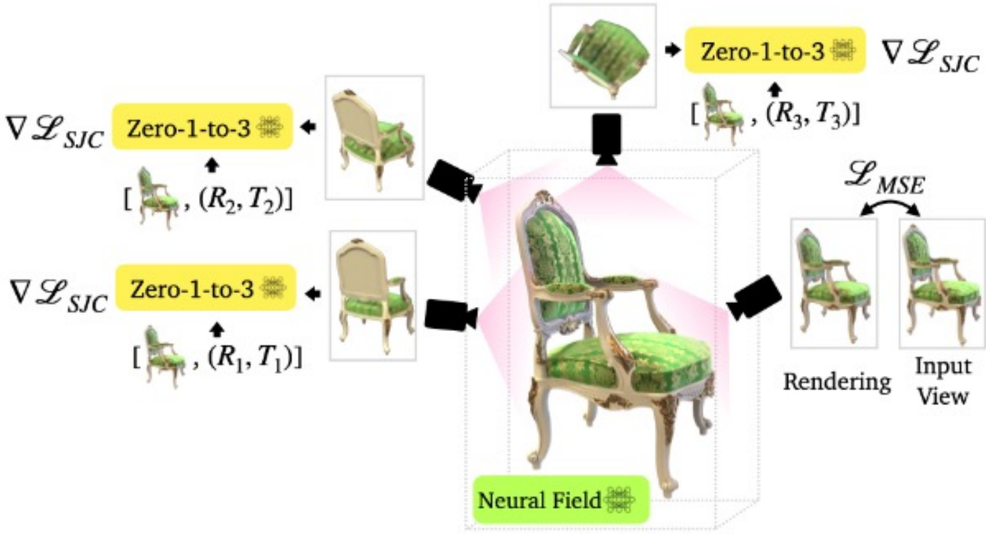
Poole et. al, DreamFusion: Text-to-3D using 2D Diffusion, ICLR 2023

Zero-1-to-3

Leverage image-to-image LDMs for image-to-3D generation



Novel View Synthesis



3D Reconstruction

- Text-to-3D => text-to-image + image-to-3D.
- Finetune Stable Diffusion: (input image, relative camera poses) => image at target view.
- Learn a NeRF with score distillation + reconstruction on the input image.

Liu et. al, Zero-1-to-3: Zero-shot One Image to 3D Object, ICCV 2023

ReconFusion

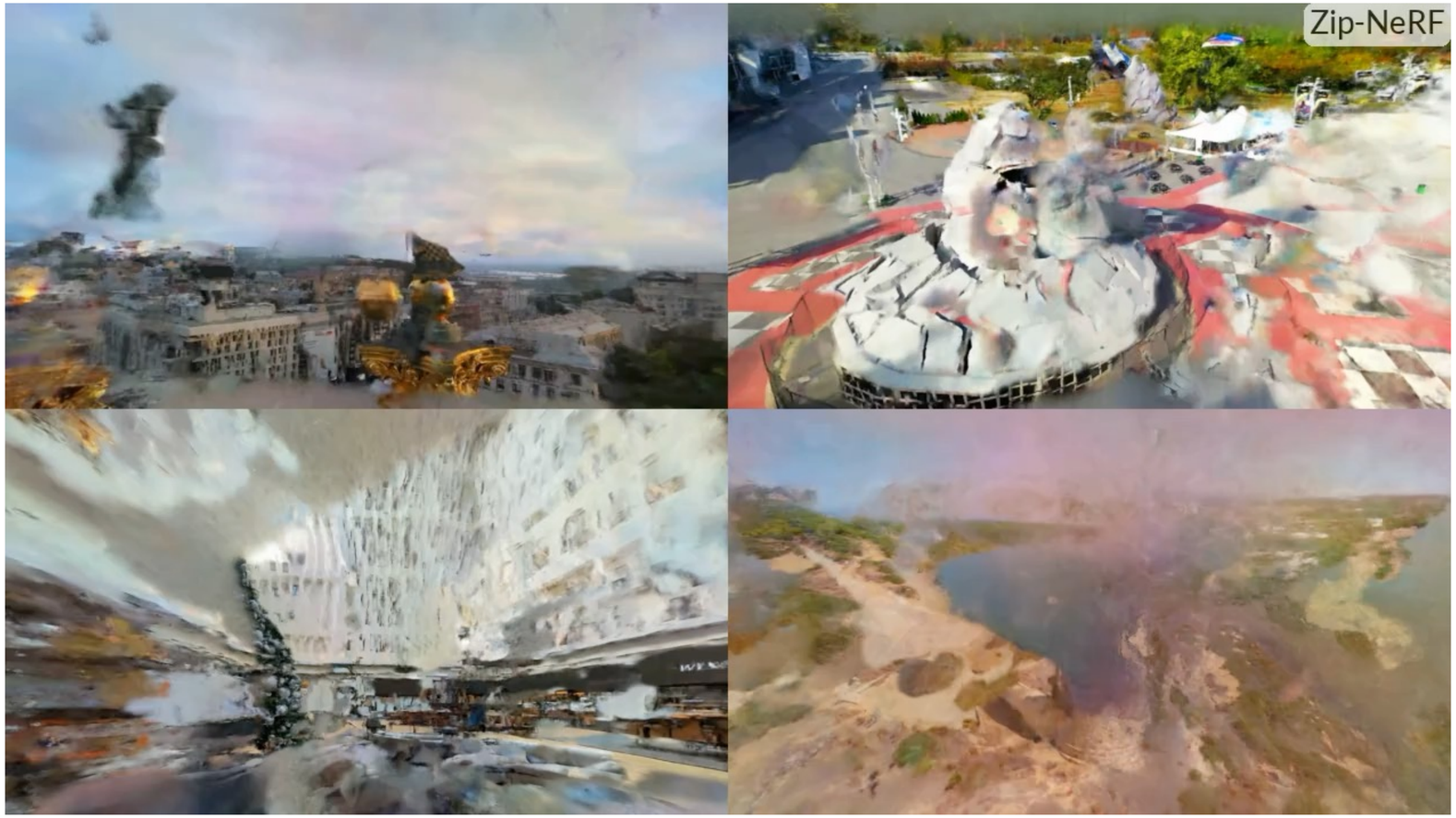
Multi-view conditioning with 3D-aware structure

- To further reduce the uncertainty, we can make the LDM conditional on multi-view inputs.
- The multi-view inputs are fused in a latent space with 3D-aware structure (pixelNeRF).
- Finetuned LDM serves as the prior for 3D reconstruction.

Wu et. al, ReconFusion: 3D Reconstruction with Diffusion Priors, arXiv 2023

ReconFusion

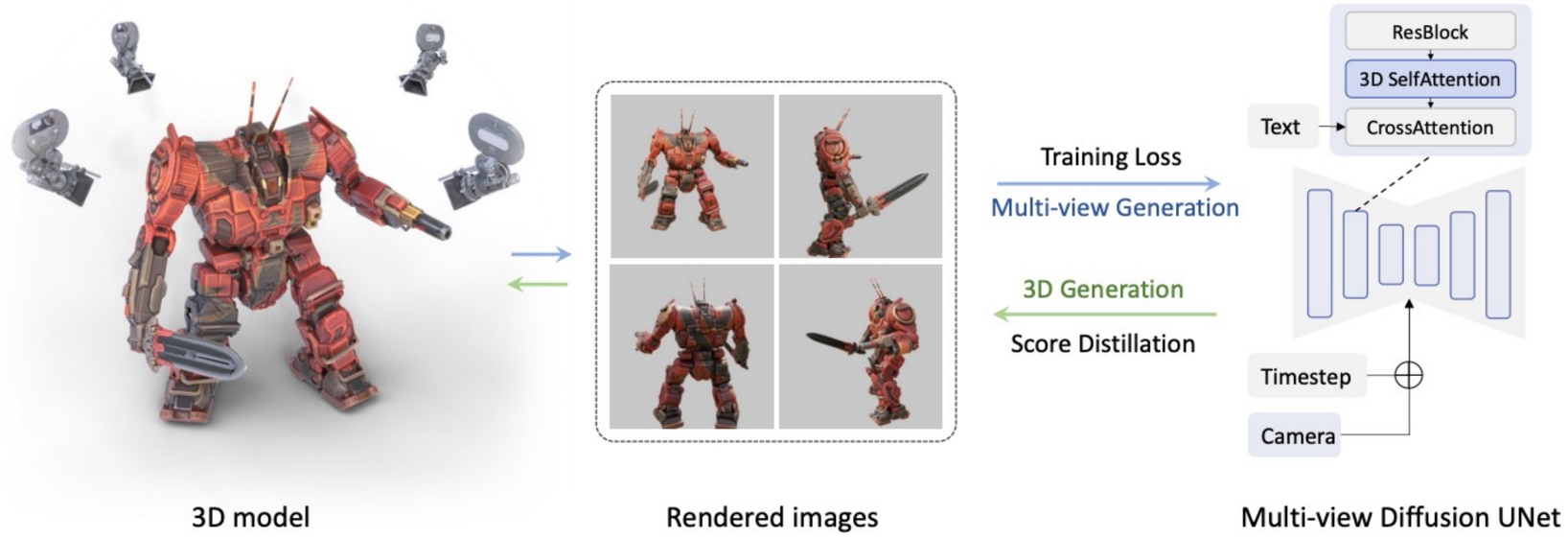
Enables high-quality 3D reconstruction from few views



Wu et. al, ReconFusion: 3D Reconstruction with Diffusion Priors, arXiv 2023

MVDream

Leverage multi-view LDMs for text-to-3D generation



- So far we have seen how to enhance conditioning signal to reduce uncertainty of the 2D diffusion.
- Can we do something better in the output front?
- Output single image => output multi-view image

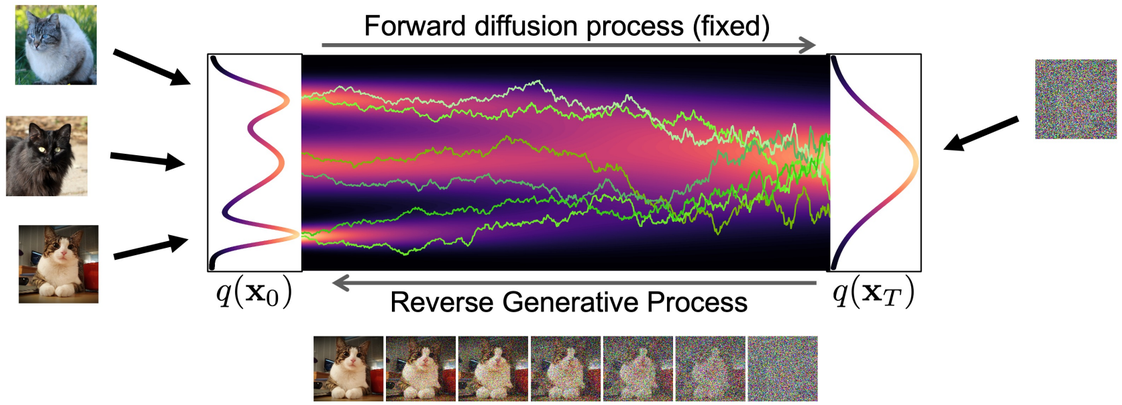
Shi et. al, MVDream: Multi-view Diffusion for 3D Generation, arXiv 2023

Contents

Diffusion Models vs. Auto-Regressive Models

Pro's and Con's

Both are (1) iterative refinement process in generation; (2) trained by supervised learning objectives.

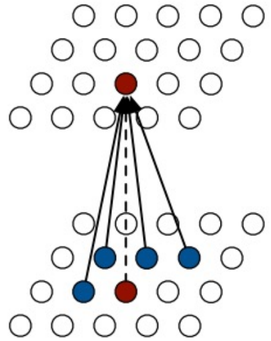


Pros:

- Flexibility in model architecture.
- Flexibility in sampling: advanced ODE / SDE solvers, distillation, guidance.

Cons:

- Worse training efficiency: a certain time step is selected per iteration.



Pros:

- Better training efficiency: a useful gradient signal from all steps is obtained through 1 pass of the model.
- Fit well with the discreate nature of text.

Cons:

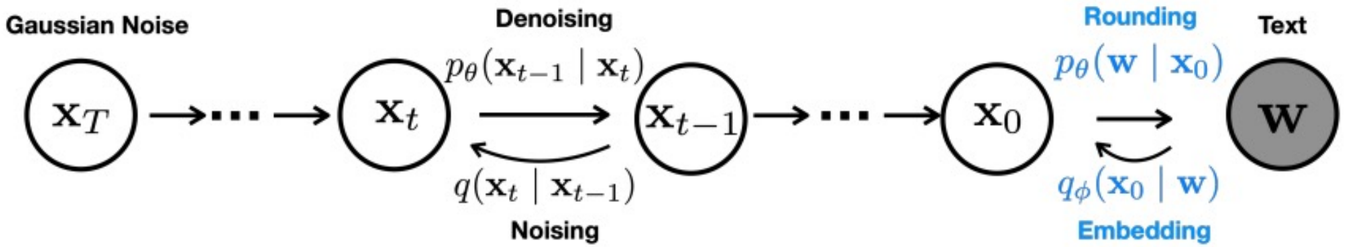
- Constraints in model architecture.
- Slow in sampling.

https://sander.ai/2023/01/09/diffusion-language.html

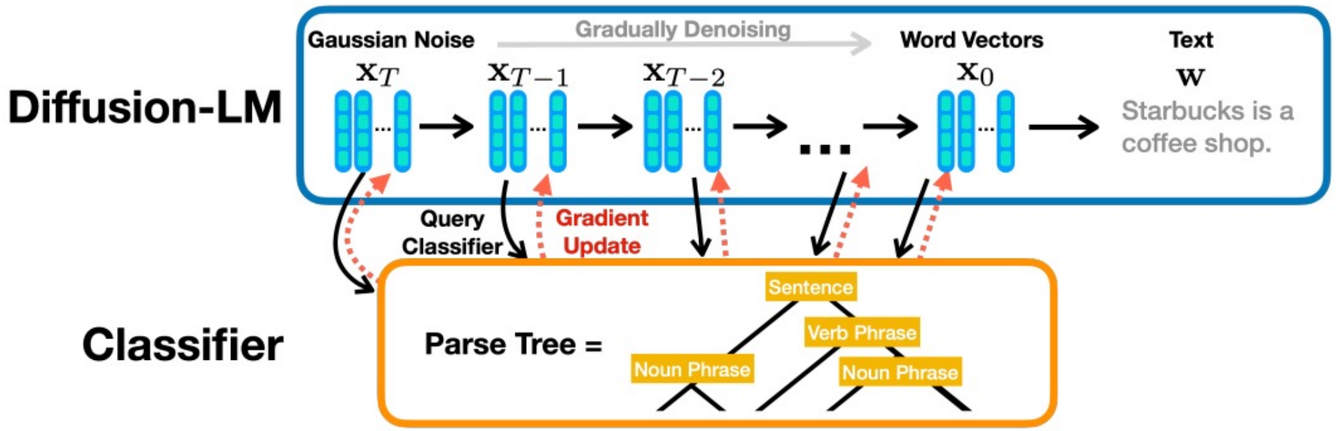
Diffusion-LM

Continuous diffusion models on text embeddings

- A minimally simple autoencoder to map text to the continuous latent space:



- Classifier guidance improves controllable generation:



Li et. al, Diffusion-LM Improves Controllable Text Generation, NeurIPS 2022

CDCD: Continuous Diffusion for Categorical Data

Training continuous diffusion models with cross-entropy loss

- Key observation: $\mathbf{x}_0$ is text embedding, only have finite possible values. $\mathbf{x}$ is noisy embedding at time t,

$$\hat{\mathbf{s}}(\mathbf{x}, t) = \sum_{i=1}^{V} p(\mathbf{x}_0 = \mathbf{e}_i | \mathbf{x}, t) \mathbf{s}(\mathbf{x}, t | \mathbf{x}_0 = \mathbf{e}_i)$$

$$\mathbf{s}(\mathbf{x}, t | \mathbf{x}_0) = \frac{\mathbf{x}_0 - \mathbf{x}}{t^2}$$

Learned by cross-entropy loss

- For sampling, leverage any sampler for continuous diffusion models.

Dieleman et. al, Continuous diffusion for categorical data, 2022

Contents

Representations of Molecules

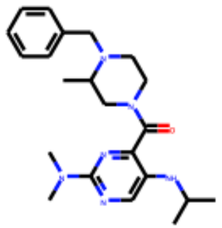
```
CC(=O)OCCC(/C)=C\C[C@H](C(C)=C)CCC=C
```

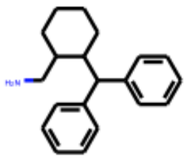
```
CC[C@H](O1)CC[C@@]12CCCO2
```

```
CC(C)[C@@]12C[C@@H]1[C@@H](C)C(=O)C2
```

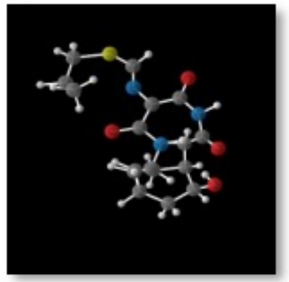
```
OCCc1c(C)[n+](cs1)Cc2cnc(C)nc2N
```

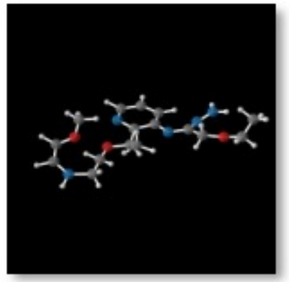
String-based representations (SMILES, SELFIES, …)



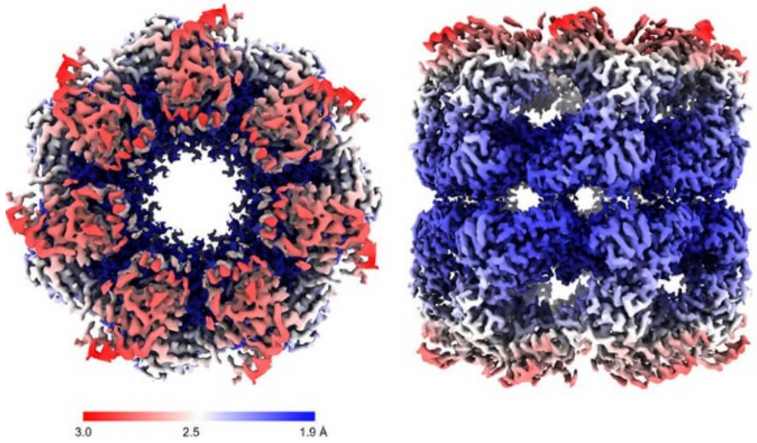


2D graphs





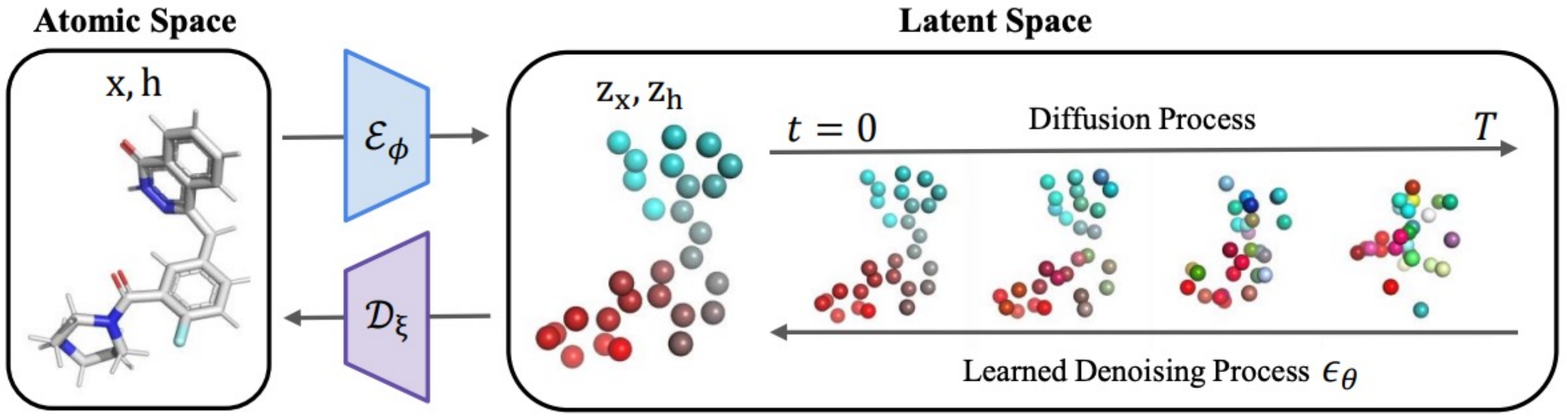
3D structure



Biological imaging

Geometric LDMs

3D molecule generation



- Data format: point clouds. x-coordinates matrix, h–node feature.
- *Equivariance* needs to be preserved: invariant to coordinate rotation / translation.
- Parametrizing encoder and decoder with equivariant GNNs.

$$\mathbf{R}\mathbf{z}_{\mathrm{x}}+\boldsymbol{t}, \mathbf{z}_{\mathrm{h}} = \mathcal{E}_\phi(\mathbf{R}\mathbf{x}+\boldsymbol{t}, \mathbf{h}); \mathbf{R}\mathbf{x}+\boldsymbol{t}, \mathbf{h} = \mathcal{D}_\xi(\mathbf{R}\mathbf{z}_{\mathrm{x}}+\boldsymbol{t}, \mathbf{z}_{\mathrm{h}}),$$

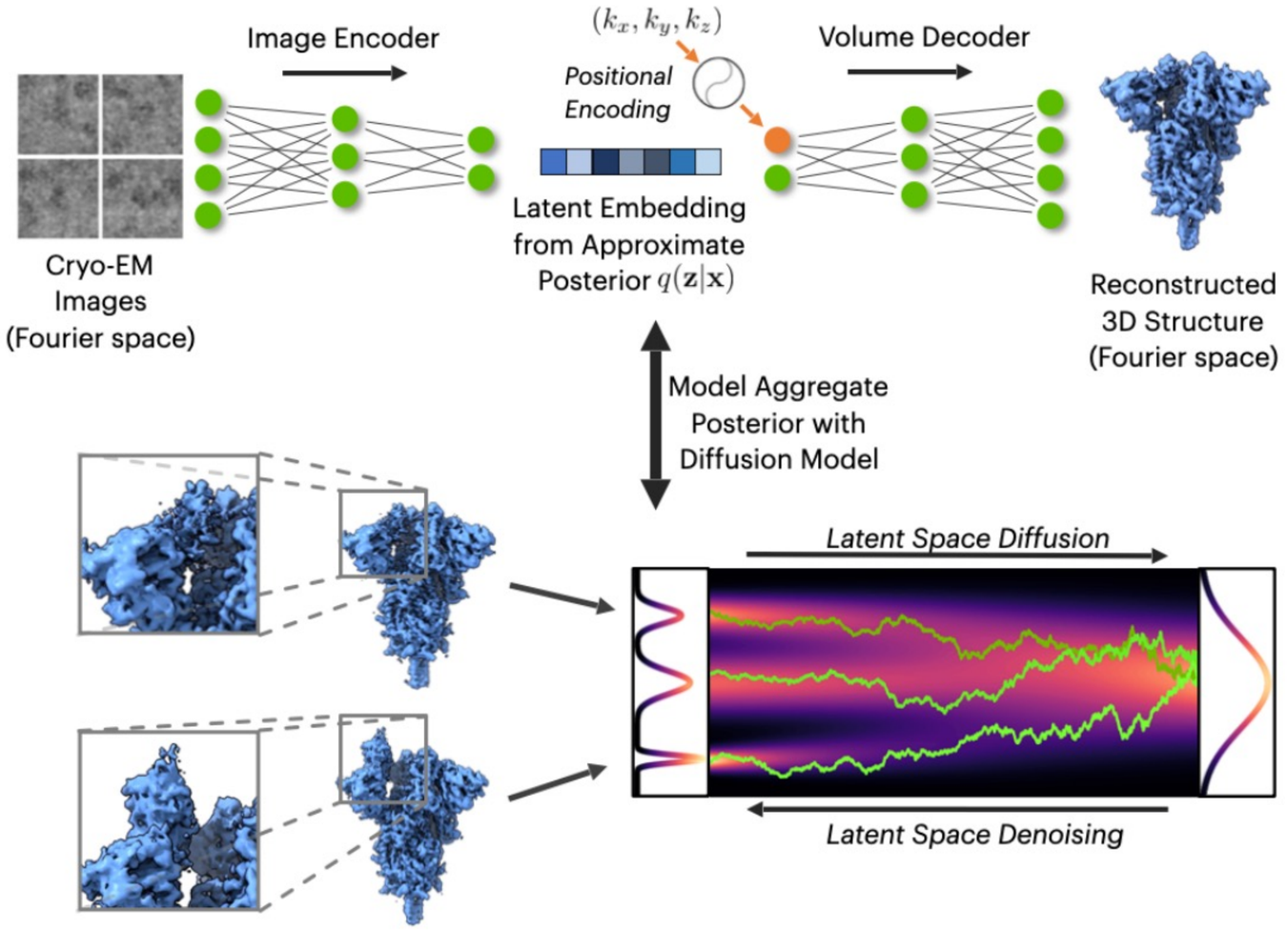
- ELBO is invariant to coordinate in the latent space:

$$p_\theta(\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}}) = p_\theta(\mathbf{R}\mathbf{z}_{\mathrm{x}}, \mathbf{z}_{\mathrm{h}})$$

Xu et. al, Geometric Latent Diffusion Models for 3D Molecule Generation, ICML 2023

LDMs for Cryo-EM Structures

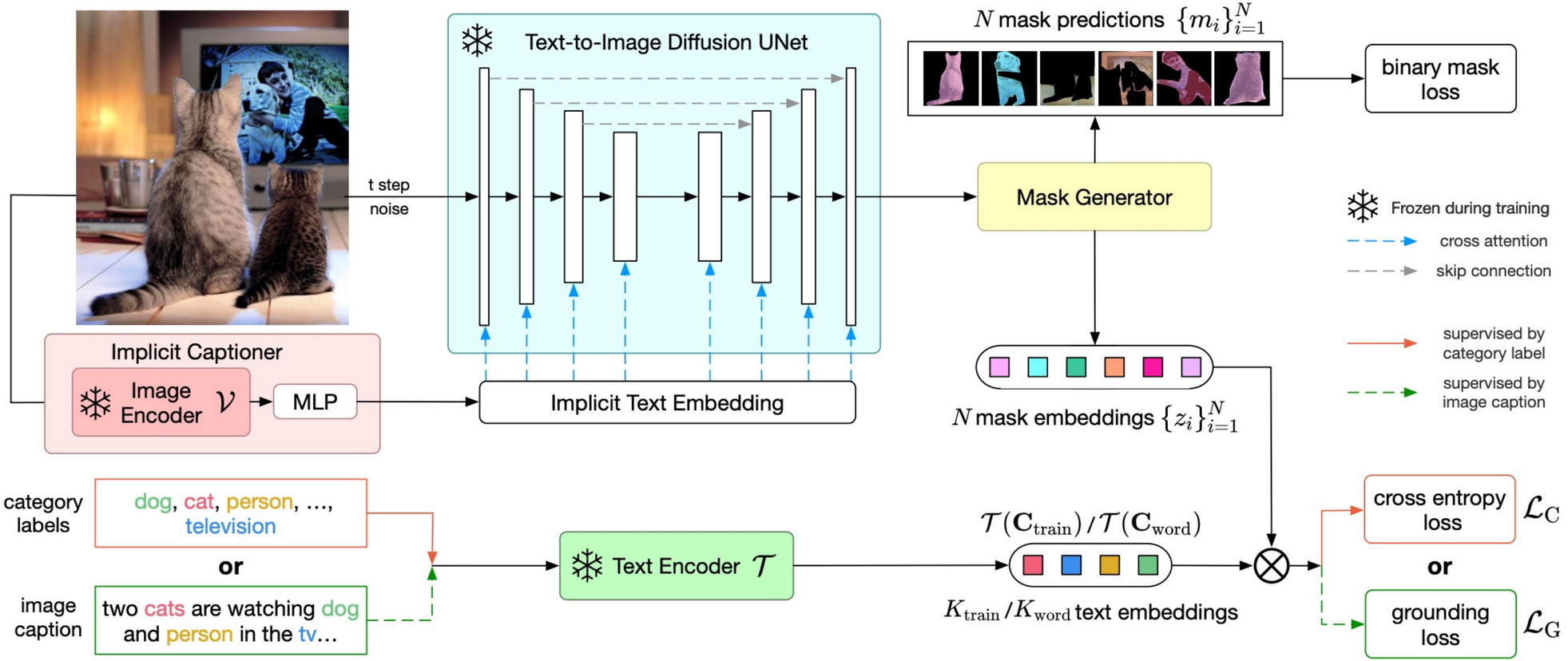
- Cryo-EM: a set of 2D projection of slices of the 3D molecule volume, captured in unknown poses.
- Latents z: different molecular configurations.
- An online optimization approach is applied to estimated the poses.



Kreis et. al, Latent Space Diffusion Models of Cryo-EM Structures, NeurIPS 2022

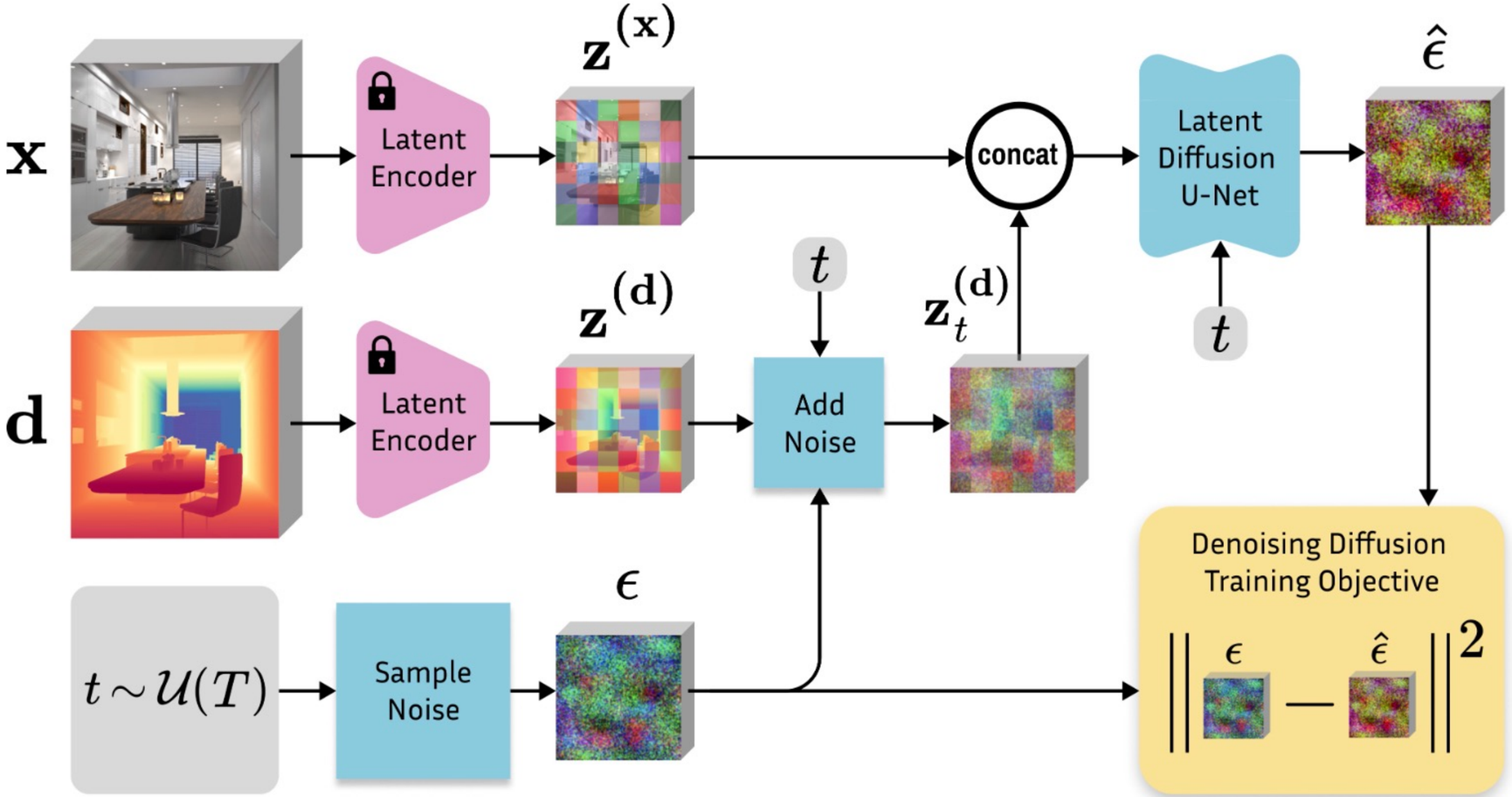
Contents

Segmentation



Xu et. al, Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models, CVPR 2023

Mono Depth Estimation



Ke et. al, [Repurposing Diffusion-Based Image Generators for Monocular Depth Estimation]

Summary

In this tutorial, we learned foundations and applications of latent diffusion models:

- The model first maps data into a latent space through a VAE, and a diffusion model is trained in the latent space.
- The latent space is compressed and regularized => easier and faster to learn / sample from.
- Advanced techniques of latent diffusion models:
    - End-to-end training, accelerating sampling, inverse problems, controllability.
- Latent diffusion models can be tailored to various data modalities:
    - We learn how it is applied to video, 3D, text, molecule generation, perception tasks.

Today's Program

Title	Speaker	Time
Part (1): Introduction to Latent Diffusion Models *Diffusion models, autoencoding, compression, latent diffusion, architectures, image generation*	Karsten	40 min
Part (2): Advanced Design and Controllability *End-to-end training, maximum likelihood, accelerated sampling, distillation, control and editing*	Arash	40 min
Part (3): Latent Diffusion Models beyond Image Generation *Video generation, 3D object and scene synthesis, segmentation, language & molecule generation*	Ruiqi	40 min
Panel Discussion: *Robin Rombach, Durk Kingma, Chenlin Meng, Sander Dieleman, Ying Nian Wu*	Panelists	30 min

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

Panel Discussion

Durk Kingma
Google Deepmind

Chenlin Meng
Pika

Robin Rombach
Stability AI

Sander Dieleman
Google Deepmind

Ying Nian Wu
University of California, Los Angeles

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