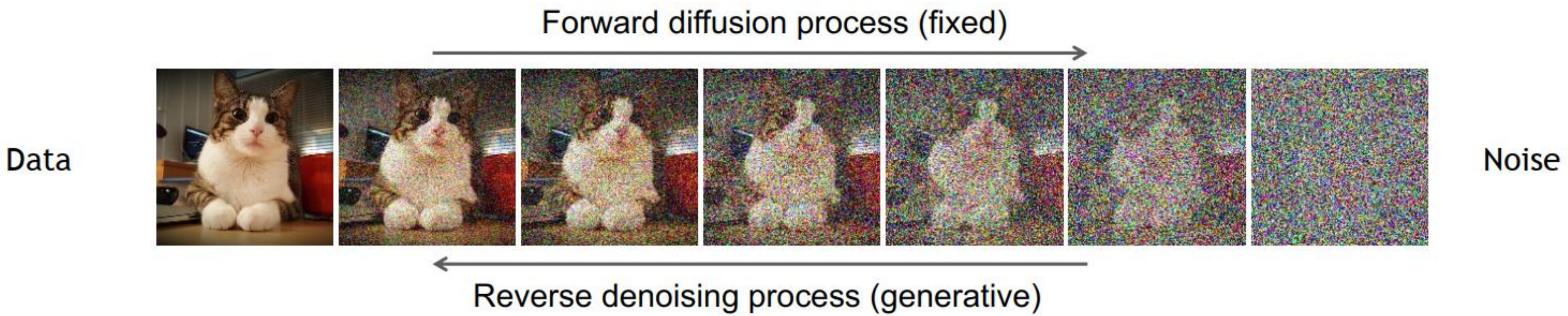
Diffusion Models

Denoising Diffusion Probabilistic Models

Learning to generate by denoising

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

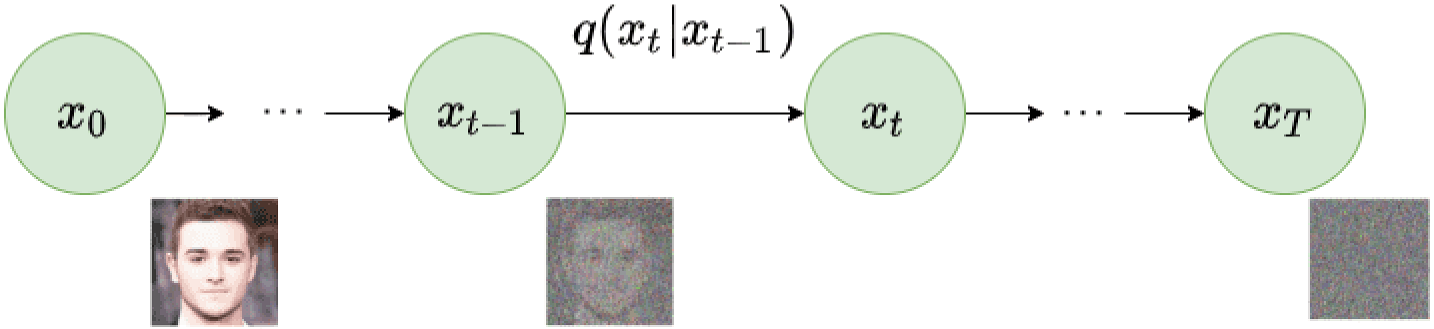


[Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015]
[Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020]
[Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021]

Diffusion Process

- Gradually add noise to input image $\boldsymbol{x_0}$ in a series of $\boldsymbol{T}$ time steps

- Neural network trained to recover original data



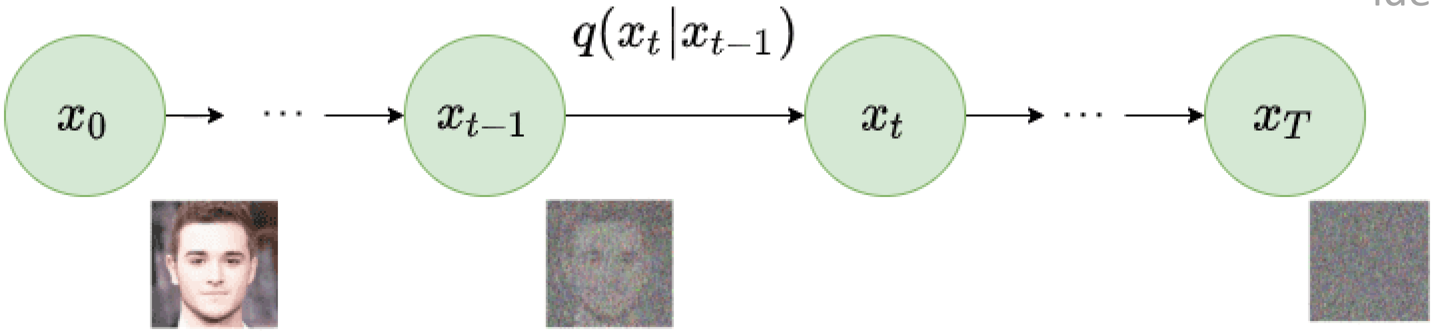
[Ho et al. ’20] Denoising Diffusion Probabilistic Models

Forward Diffusion

- Markov chain of $\boldsymbol{T}$ steps
  - Each step depends only on previous
- Adds noise to $\boldsymbol{x_0}$ sampled from real distribution $\boldsymbol{q(x)}$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \boldsymbol{\mu}_t = \sqrt{1-\beta_t}x_{t-1}, \boldsymbol{\Sigma}_t = \beta_t\mathbf{I})$$

mean variance identity matrix



[Ho et al. '20] Denoising Diffusion Probabilistic Models

Forward Diffusion

- Go from $\boldsymbol{x_0}$ to $\boldsymbol{x_T}$:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

- Efficiency?

Reparameterization

- Define $\alpha_t = 1 - \beta_t$, $\overline{\alpha_t} = \prod_{s=0}^{t} \alpha_s$, $\epsilon_0, \ldots, \epsilon_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$
\begin{aligned}
x_t &= \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \epsilon_{t-1} \\
&= \sqrt{\alpha_t} x_{t-2} + \sqrt{1 - \alpha_t} \epsilon_{t-2} \\
&= \cdots \\
&= \sqrt{\overline{\alpha_t}} x_0 + \sqrt{1 - \overline{\alpha_t}} \epsilon_0
\end{aligned}
$$

$$
x_t \sim q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha_t}} x_0, (1 - \overline{\alpha_t})\mathbf{I})
$$

Reverse Diffusion

- $x_{T\rightarrow\infty}$ becomes a Gaussian distribution

- Reverse distribution $q(x_{t-1}|x_t)$
  - Sample $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and run reverse process
  - Generates a novel data point from original distribution

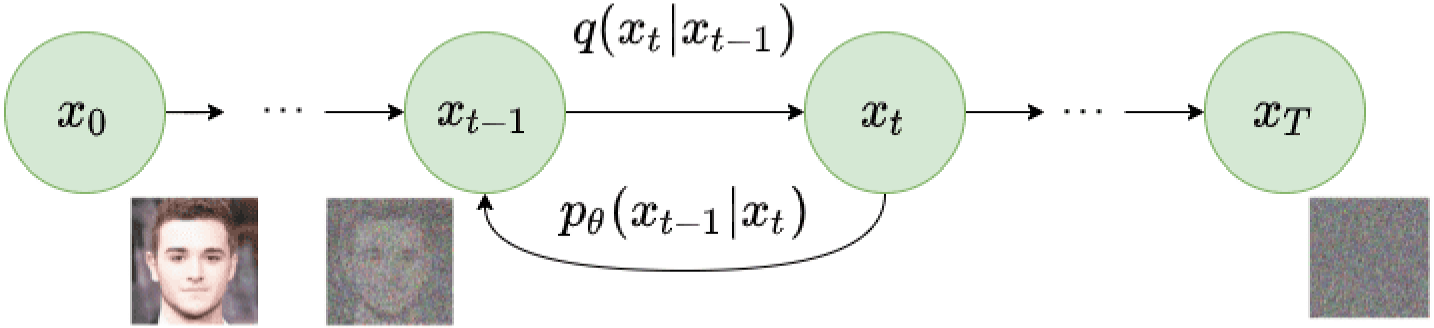
- How to model reverse process?

Approximate Reverse Process

- Approximate $q(x_{t-1}|x_t)$ with parameterized model p_θ (e.g., deep network)

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$



[Ho et al. '20] Denoising Diffusion Probabilistic Models

Training a Diffusion Model

- Optimize negative log-likelihood of training data

$$L_{VLB} = \underbrace{\mathbb{E}_q[D_{KL}(q(x_T|x_0||p_\theta(x_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{KL}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \underbrace{- \log p_\theta(x_0|x_1)]}_{L_0}$$

- Nice derivations: https://lilianweng.github.io/posts/2021-07-11-diffusion-models

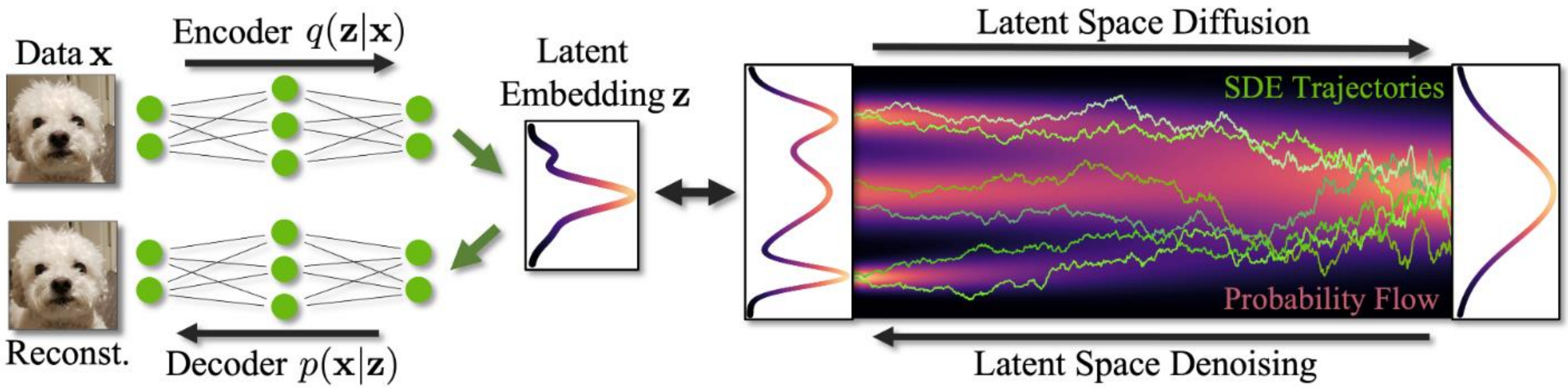
Training a Diffusion Model

- $L_{t-1} = D_{KL}\big(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)\big)$
- Comparing two Gaussian distributions
- $L_{t-1} \propto ||\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2$
- Predicts diffusion posterior mean

Diffusion Model Architecture

- Input and output dimensions must match
- Highly flexible to architecture design
- Commonly implemented with U-Net architecture

Latent Diffusion



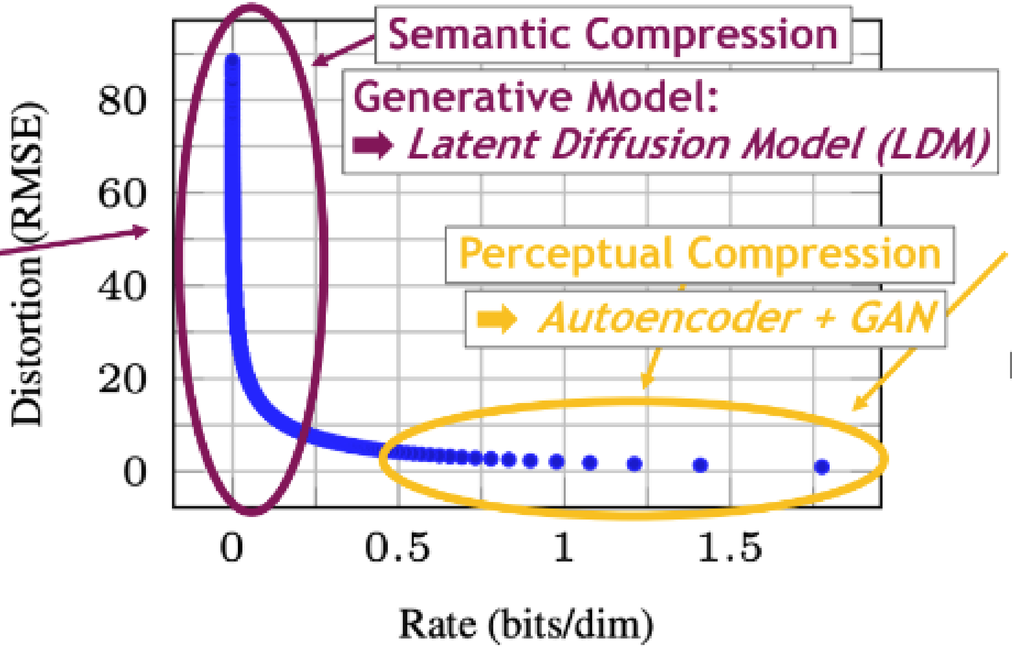
Stage 1: Train AE / VAE / VQ-VAE / VQ-GAN

Stage 2: Diffusion in Latent Space

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

Latent Diffusion

Large-scale Image Structure

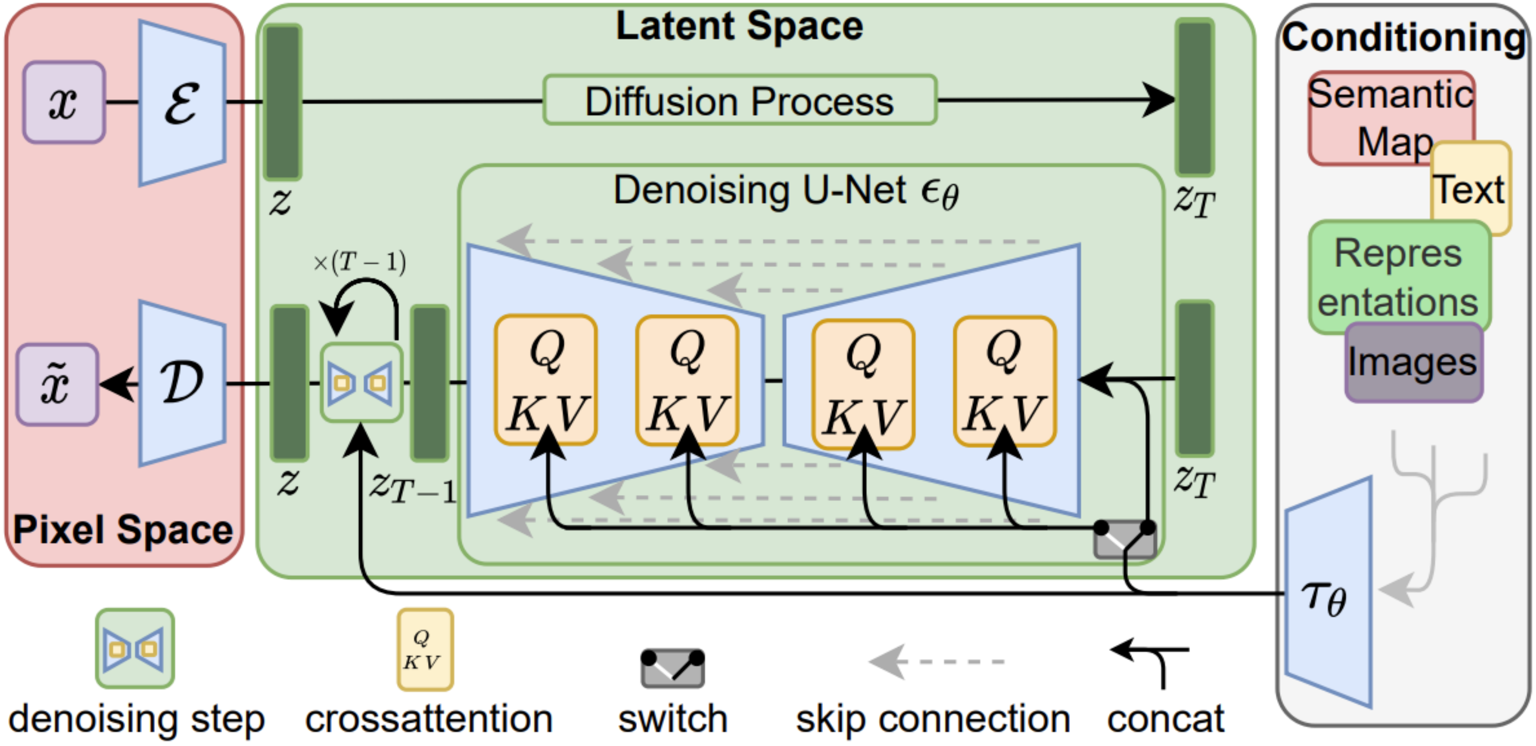


Local, Imperceptible Details

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

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

Latent Diffusion

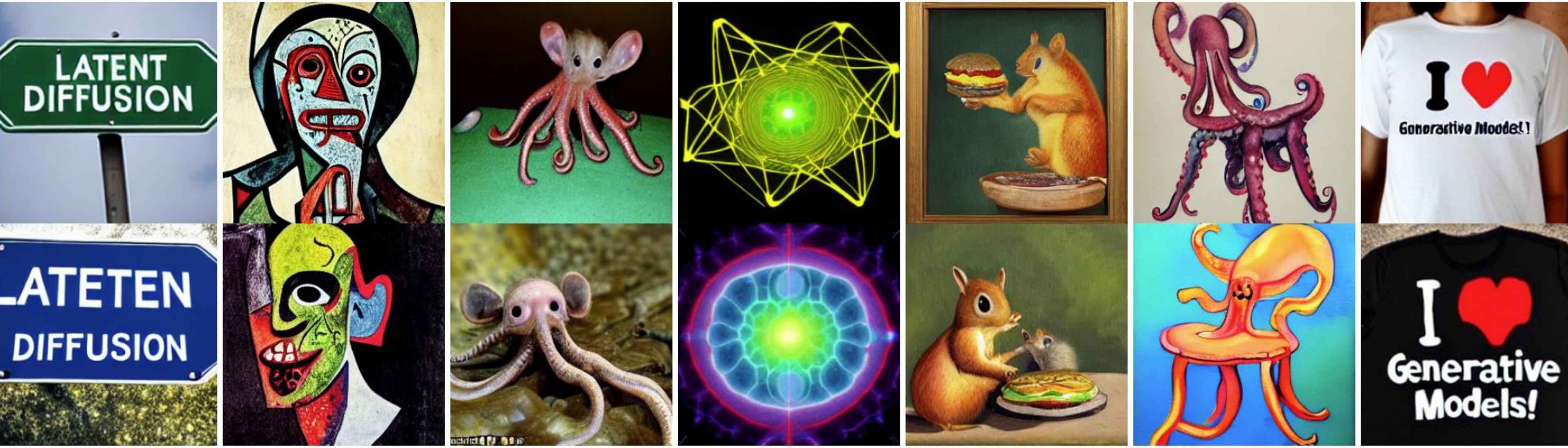


[Rombach et al. 21] Latent Diffusion

Latent Diffusion

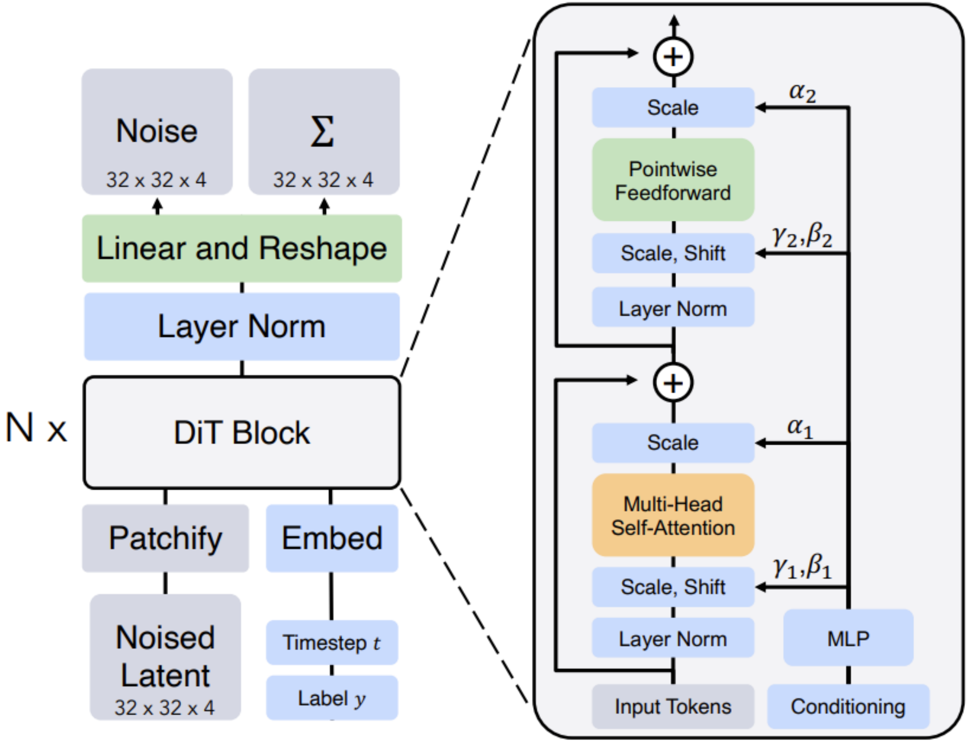
Text-to-Image Synthesis on LAION. 1.45B Model.

'A street sign that reads "Latent Diffusion" '	*'A zombie in the style of Picasso'*	*'An image of an animal half mouse half octopus'*	*'An illustration of a slightly conscious neural network'*	*'A painting of a squirrel eating a burger'*	*'A watercolor painting of a chair that looks like an octopus'*	*'A shirt with the inscription: "I love generative models!" '*



[Rombach et al. 21] Latent Diffusion

Diffusion Transformers (DiT)



Latent Diffusion Transformer

DiT Block with adaLN-Zero

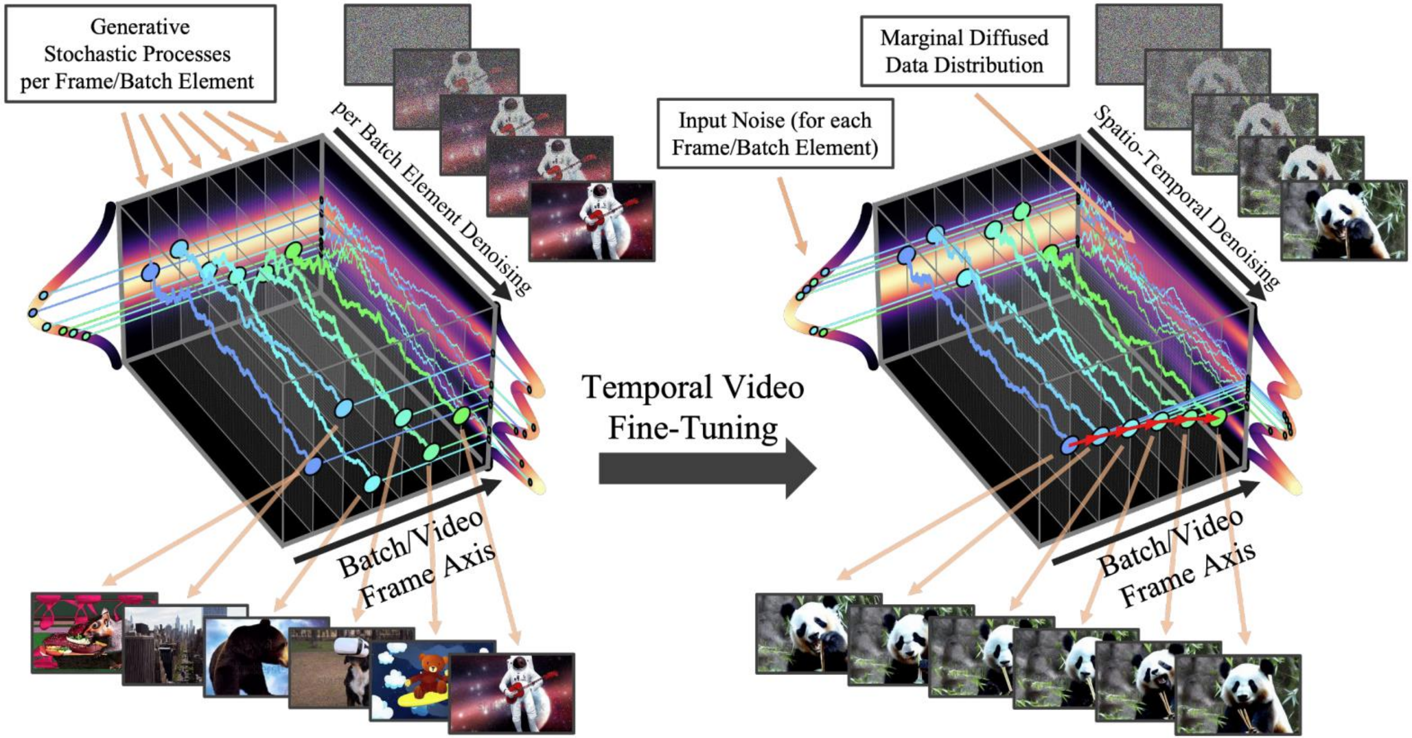
[Peebles and Xie 23] Diffusion Transformers

Diffusion Transformers (DiT)

[Peebles and Xie 23] Diffusion Transformers

Video Diffusion Models

Align your Latents

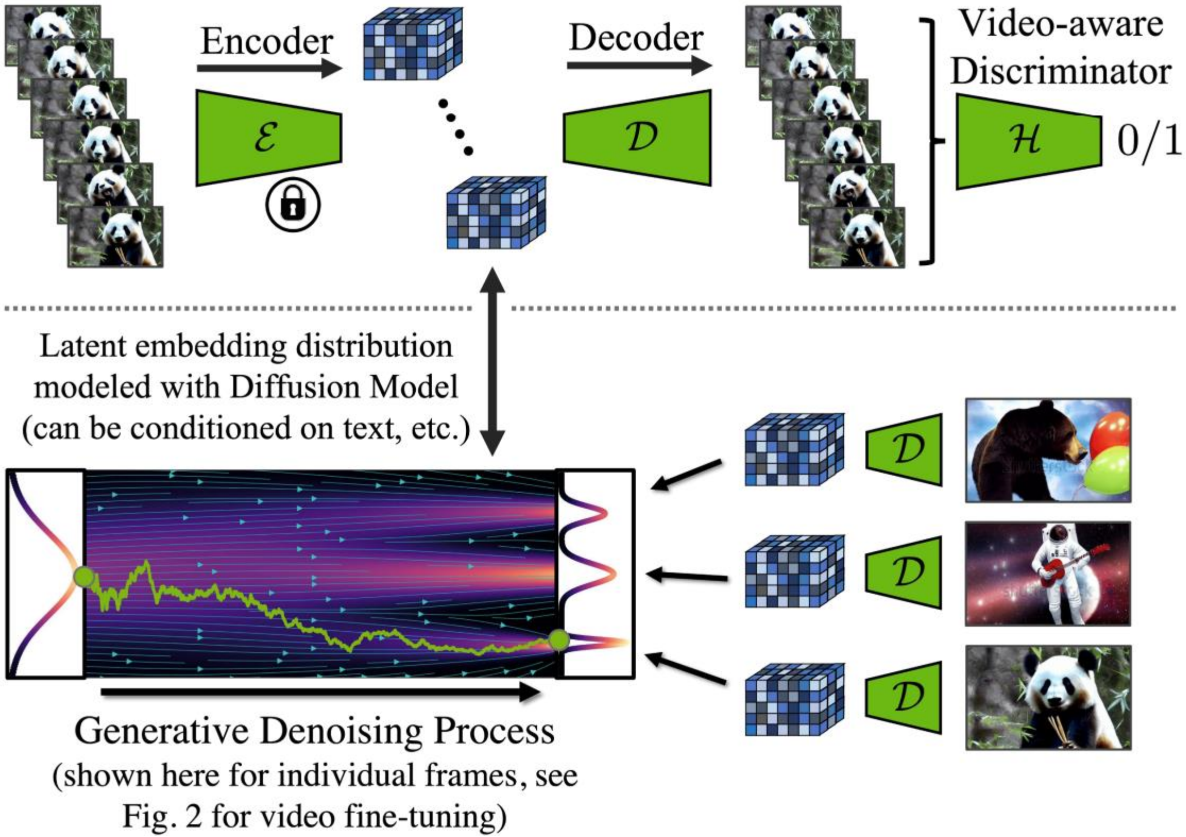


Before temporal video fine-tuning, different batch samples are independent.

After temporal video fine-tuning, samples are aligned to form a video sequence (after applying the LDM decoder).

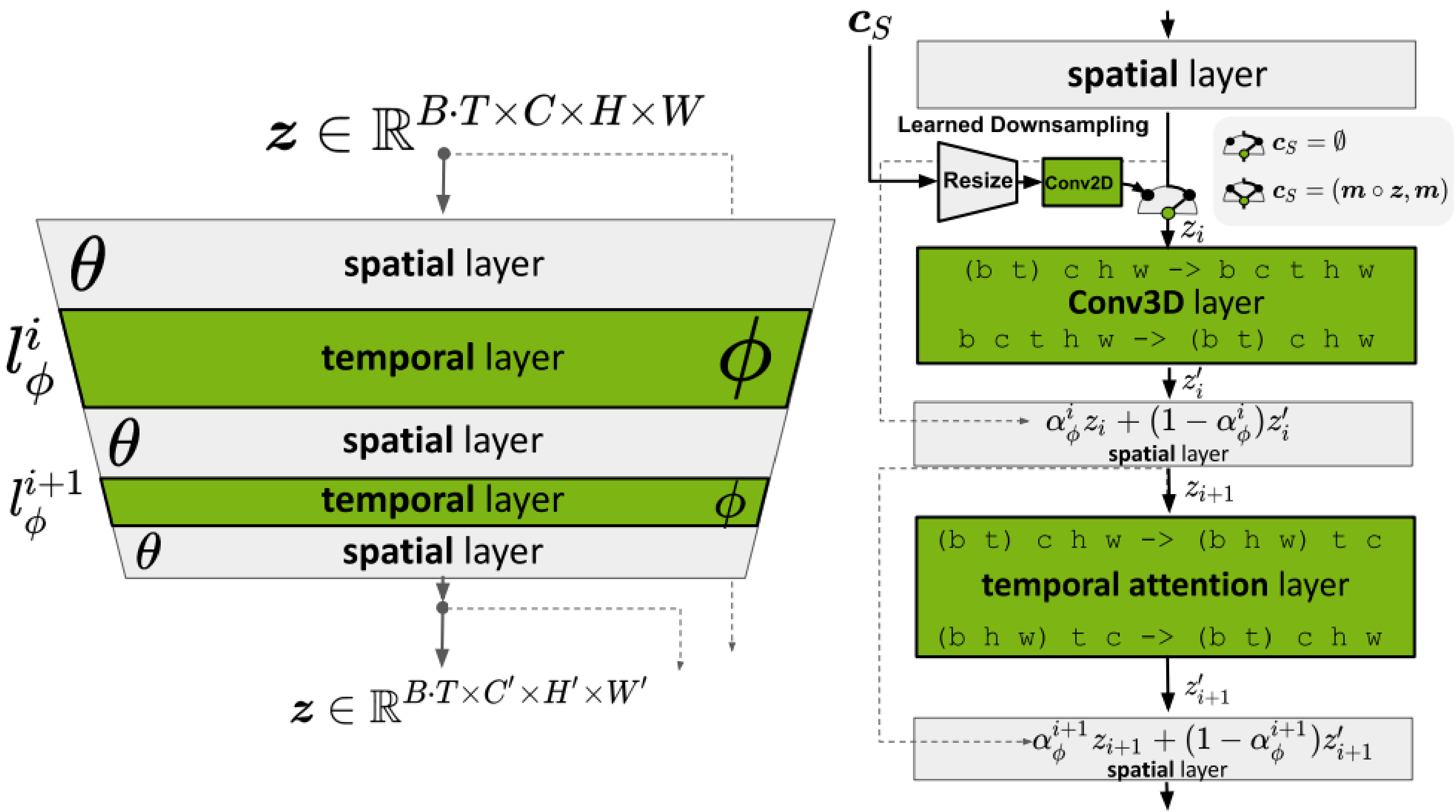
[Blattmann et al. 22] Latent Align your Latents

Align your Latents



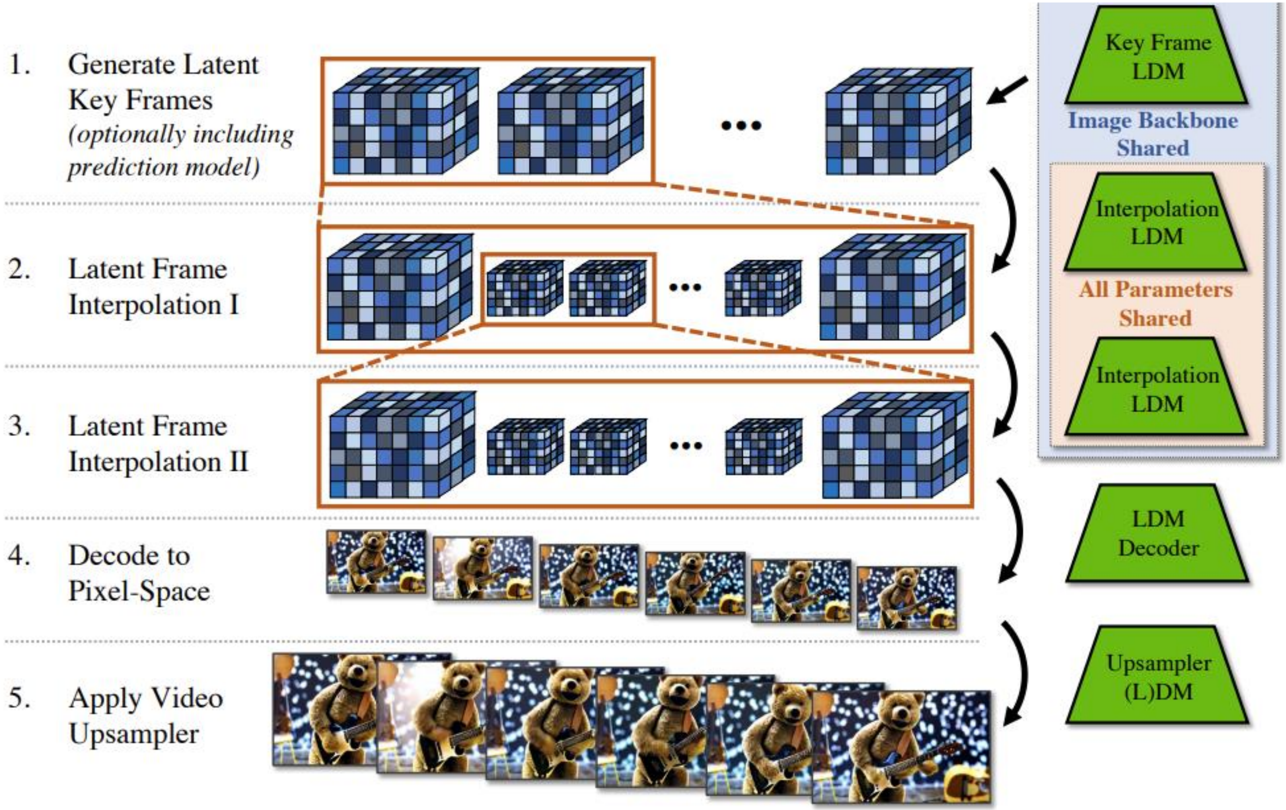
[Blattmann et al. 22] Latent Align your Latents

Align your Latents



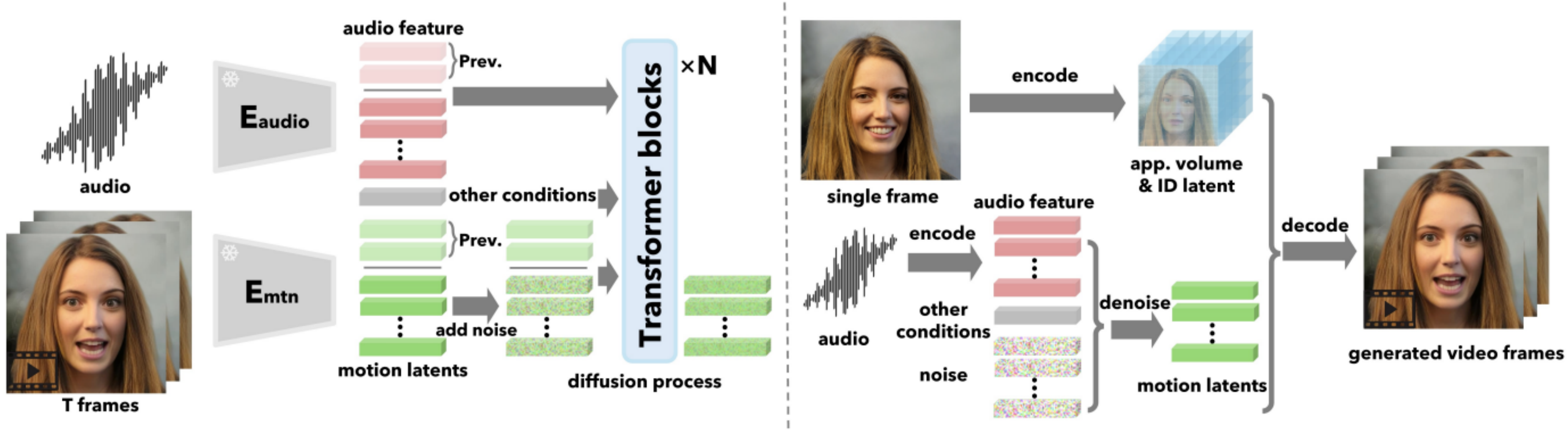
[Blattmann et al. 22] Latent Align your Latents

Align your Latents



[Blattmann et al. 22] Latent Align your Latents

VASA-1



Training pipeline for motion latent diffusion

Test pipeline for audio to video generation

[Xu et al. 24] VASA-1

VASA-1

[Xu et al. 24] VASA-1

VASA-1

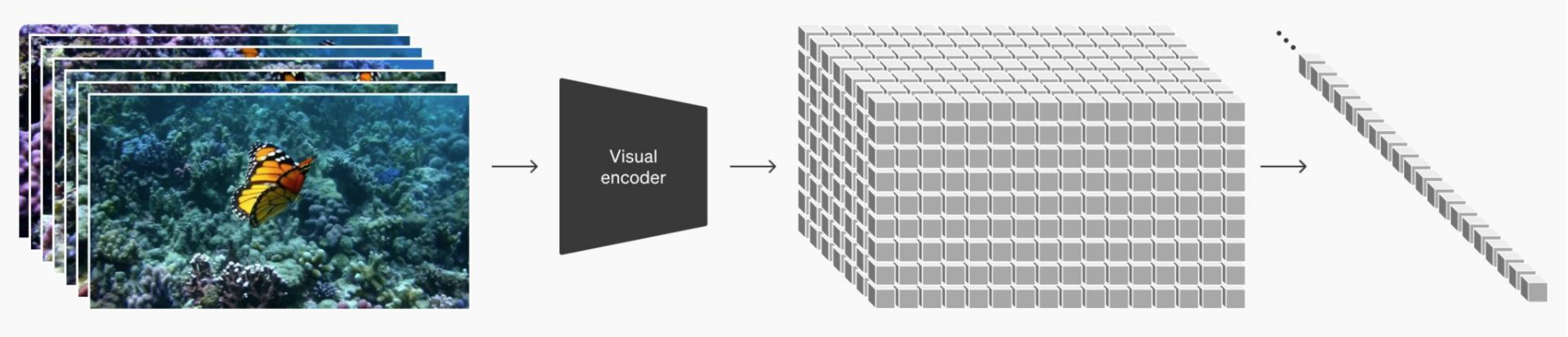
[Xu et al. 24] VASA-1

Sora

Sora

Sora

- Temporal Diffusion Transformer
- VAE vs VQ-GAN vs … ?
- Temporal window?
- Pre-trained on any images?



“Video Compressor”

Luma Dream Machine

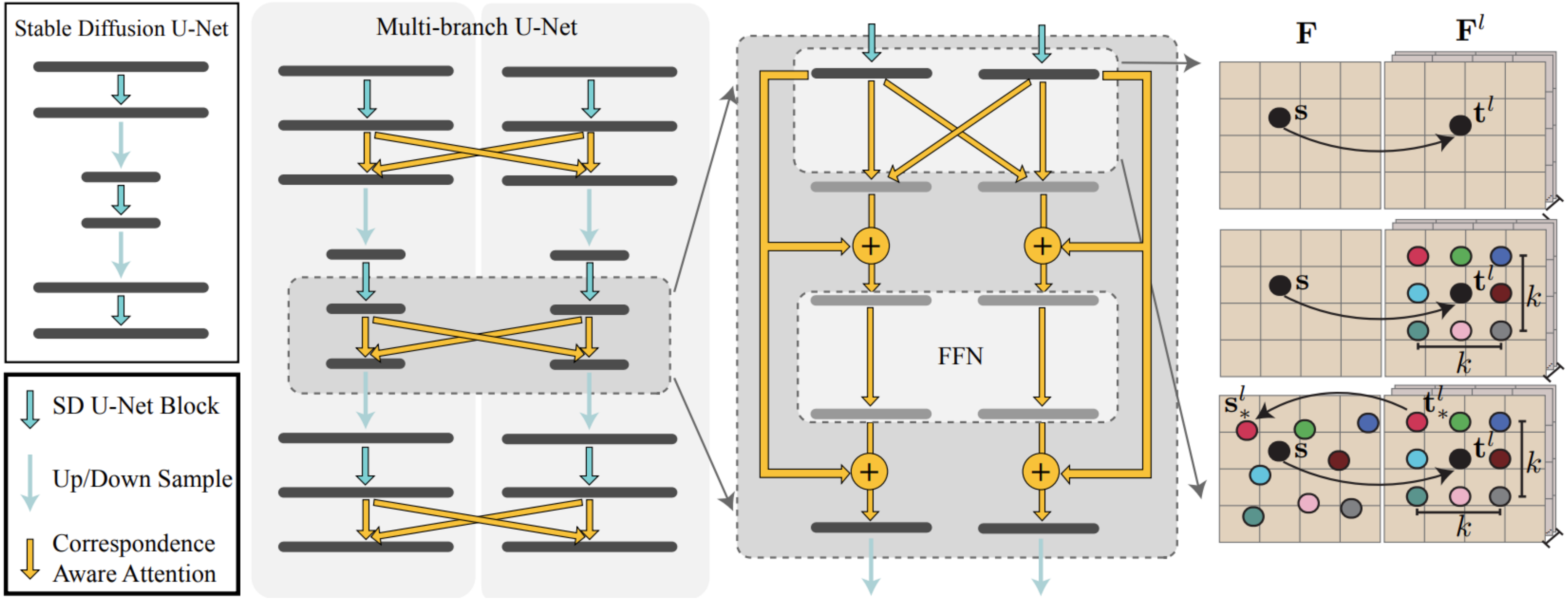
Jiaming Song's Talk https://youtube.com/live/oilWwsXZamA

Latent Design

- Pre-trained image-based LDM
  - Can leverage massive amounts of pre-training
  - Issues in low-level temporal stability
- Training directly on videos
  - Can't use any image-based pre-training
  - Potentially better low-level stability

Multi-view Diffusion Models

MVDiffusion



[Tang et al. 23] Tang et al.

MVDiffusion

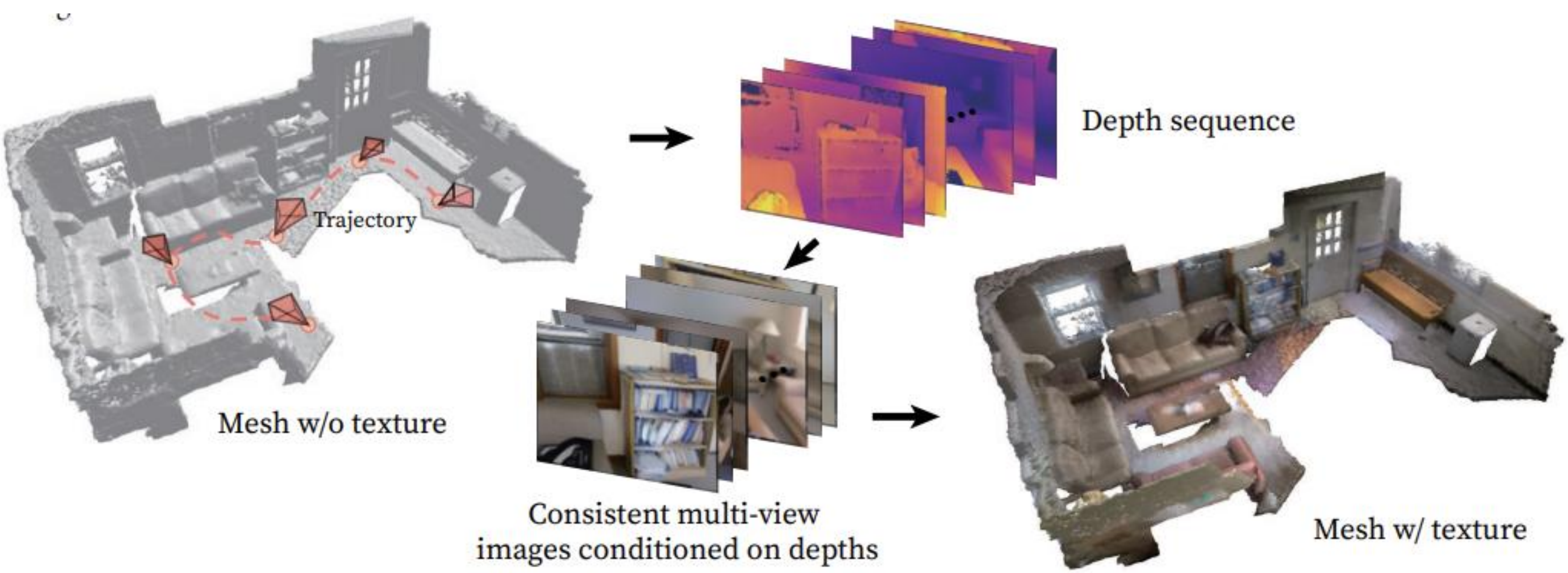
"This kitchen is a charming blend of rustic and modern, featuring a large reclaimed wood island with marble countertop, a sink surrounded by cabinets. A stainless-steel refrigerator stands tall. To the right of the sink, built-in wooden cabinets painted in a muted."

Consistent Multi-view Images → Closed-Loop Panoramic Image

"A living room with multiple couches and a coffee table. A wooden book shelf filled with lots of books next to a door. A white refrigerator sitting next to a wooden bench."

[Tang et al. 23] Tang et al.

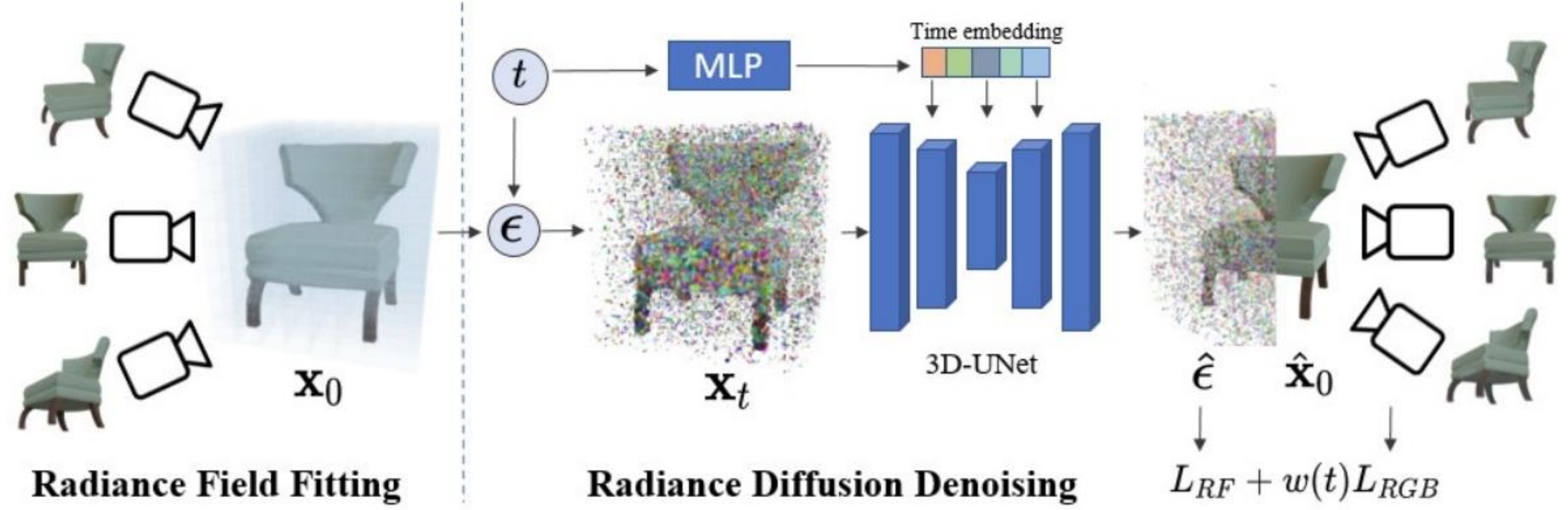
MVDiffusion



[Tang et al. 23] Tang et al.

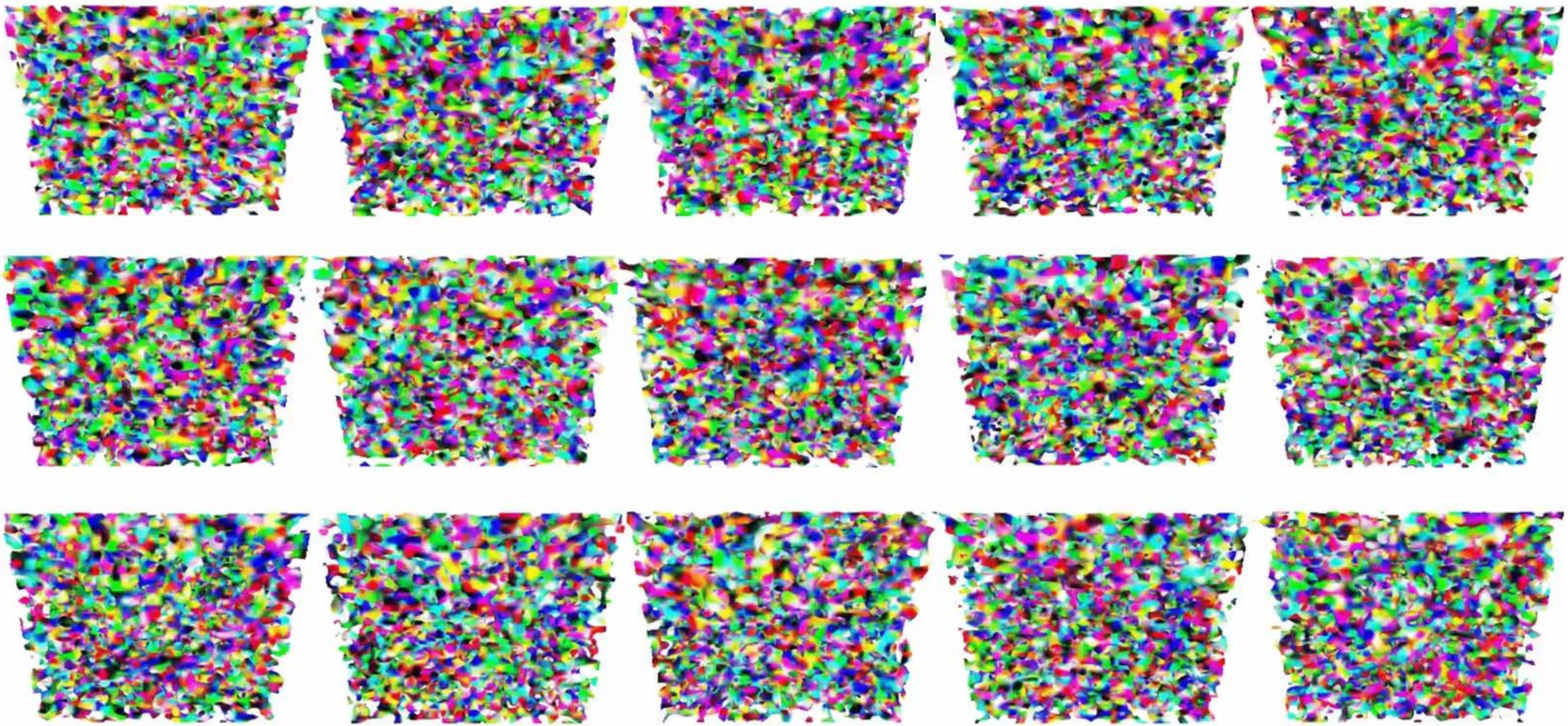
3D Aware Diffusion

DiffRF: Train with 3D Ground Truth



[Mueller et al. 23] DiffRF

DiffRF: Results

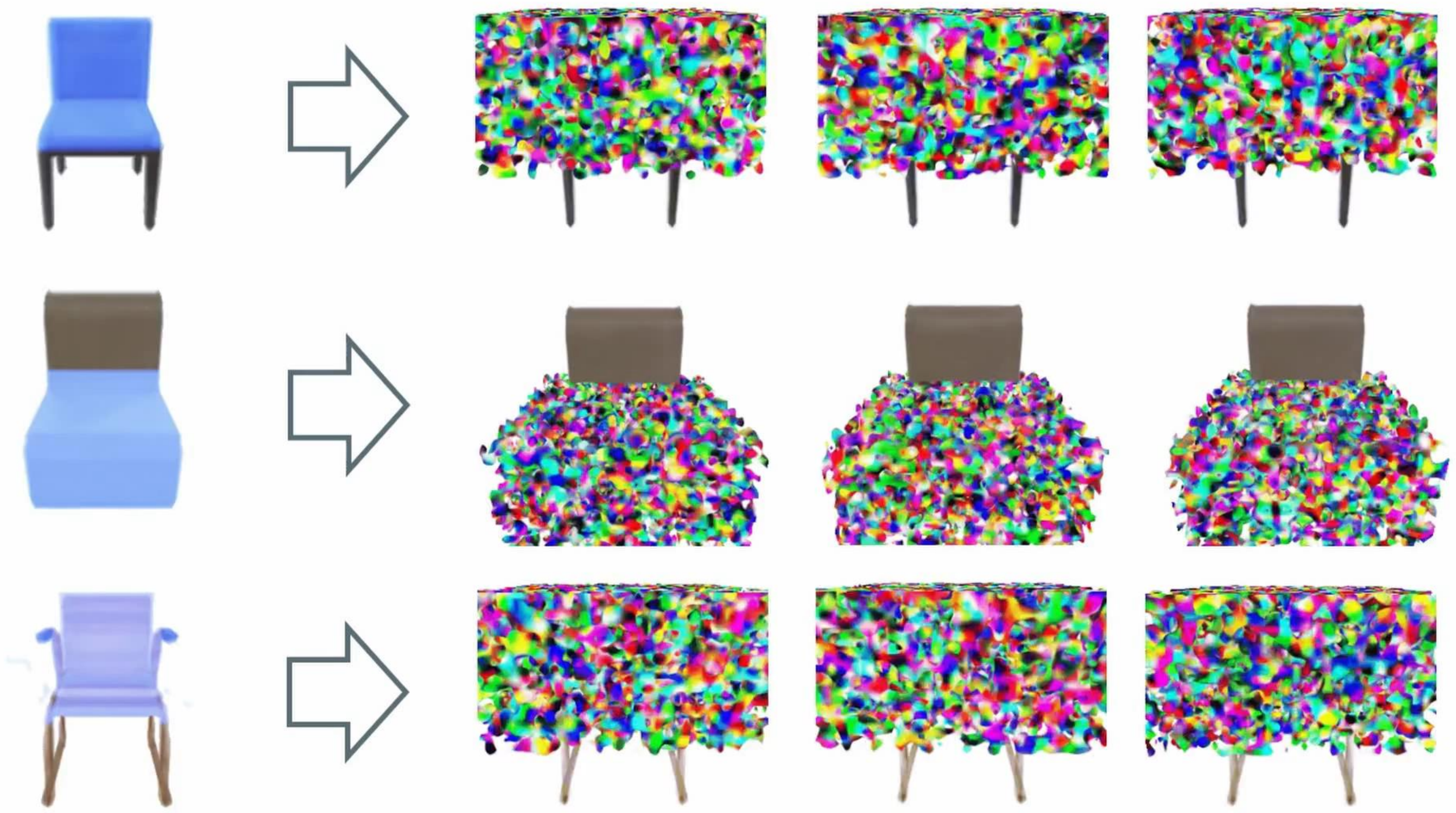


[Mueller et al. 23] DiffRF

DiffRF: Masked Predictions

[Mueller et al. 23] DiffRF

DiffRF: Masked Predictions



[Mueller et al. 23] DiffRF

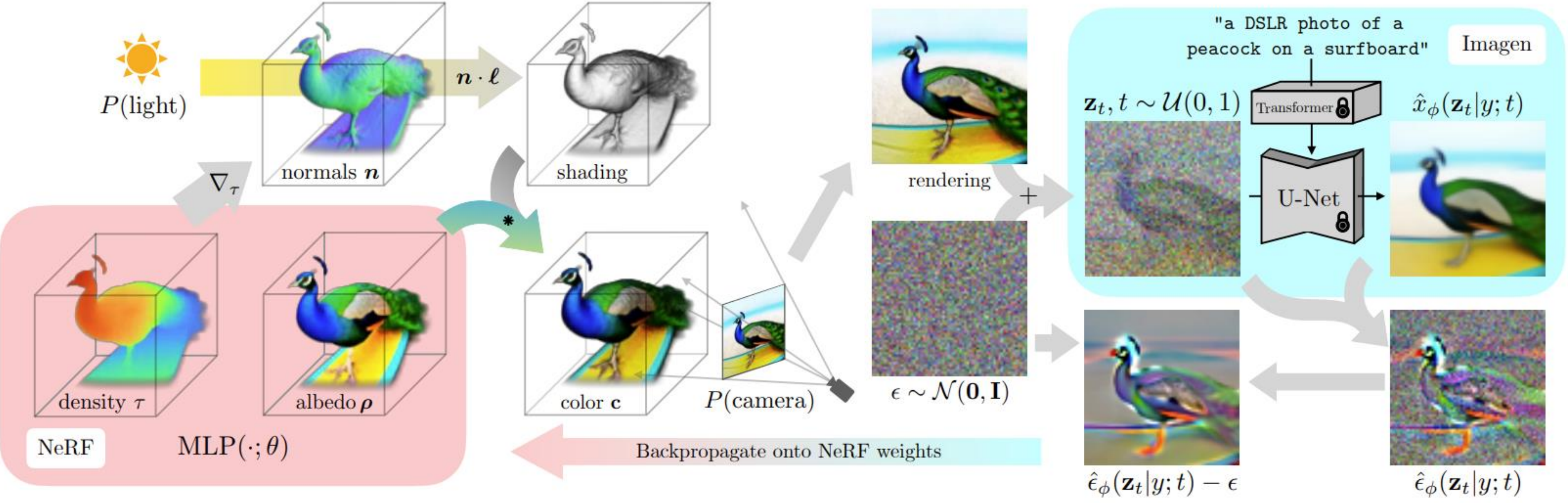
Discussion: Diffusion vs GANs

Problem is always training data…

- Diffusion: input vs output need same dimensionality

- GANs: partial information feasible (e.g., reprojection, similar to GRAF, PiGAN, EG3D)

DreamFusion



Score Distillation Sampling (SDS)

[Poole et al. 22] DreamFusion

Score Distillation Sampling (SDS)

Loss functions for diffusion models

$$\mathcal{L}_{\text{Diff}}(\phi, \mathbf{x}) = \mathbb{E}_{t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[w(t) \| \epsilon_\phi(\alpha_t \mathbf{x} + \sigma_t \epsilon; t) - \epsilon \|_2^2 \right]$$

Training a diffusion model: $\phi^* = \arg\min_{\phi} \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x})$

Sampling from a diffusion model? $\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x})$

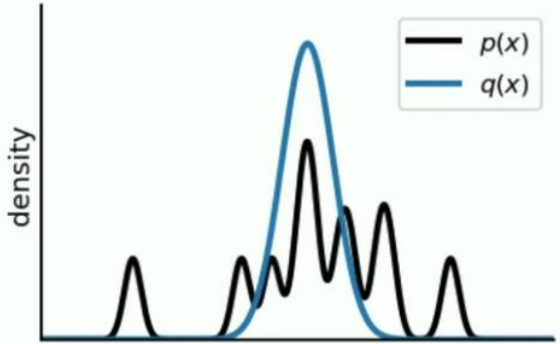
$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x} = g(\theta)) = \mathbb{E}_{t,\epsilon} \Bigg[w(t) \underbrace{\left(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon\right)}_{\text{Noise Residual}} \underbrace{\cancel{\frac{\partial \hat{\epsilon}_\phi(\mathbf{z}_t; y, t)}{\mathbf{z}_t}}}_{\text{U-Net Jacobian}} \underbrace{\frac{\partial \mathbf{x}}{\partial \theta}}_{\text{Generator Jacobian}} \Bigg]$$

Score Distillation Sampling (SDS)

Score distillation sampling

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

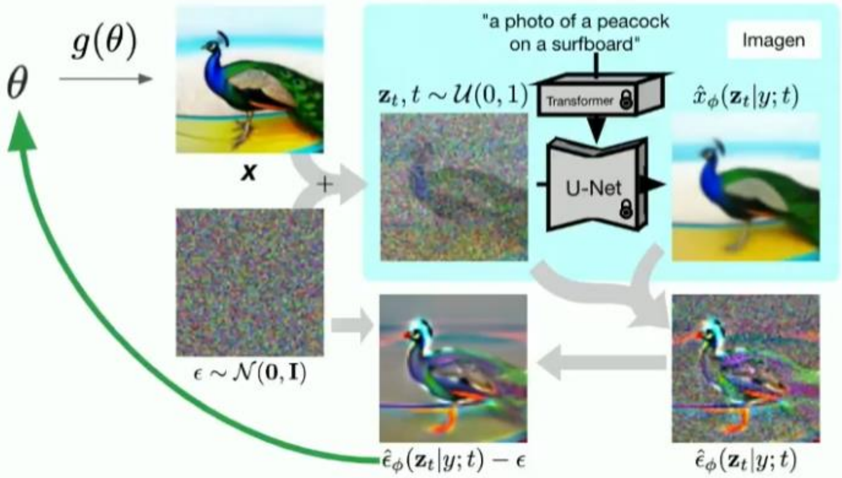
$$\mathcal{L}_{\text{SDS}}(\theta) = \mathbb{E}_t\left[w(t)\text{KL}(q(z_t; \theta, y, t) \| p_\phi(z_t; y, t))\right]$$



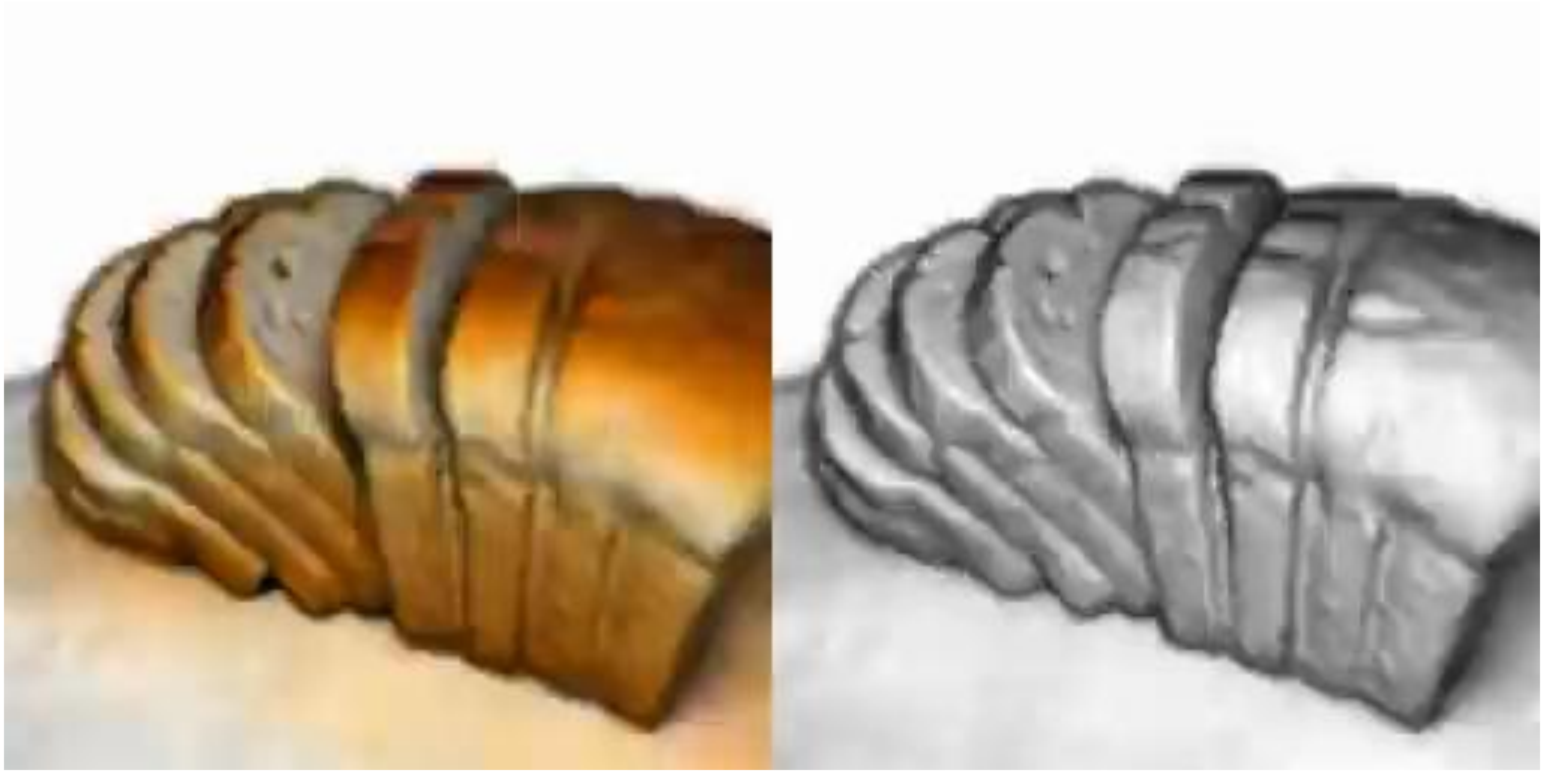
Score Distillation Sampling (SDS)

Using the score distillation loss

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



DreamFusion



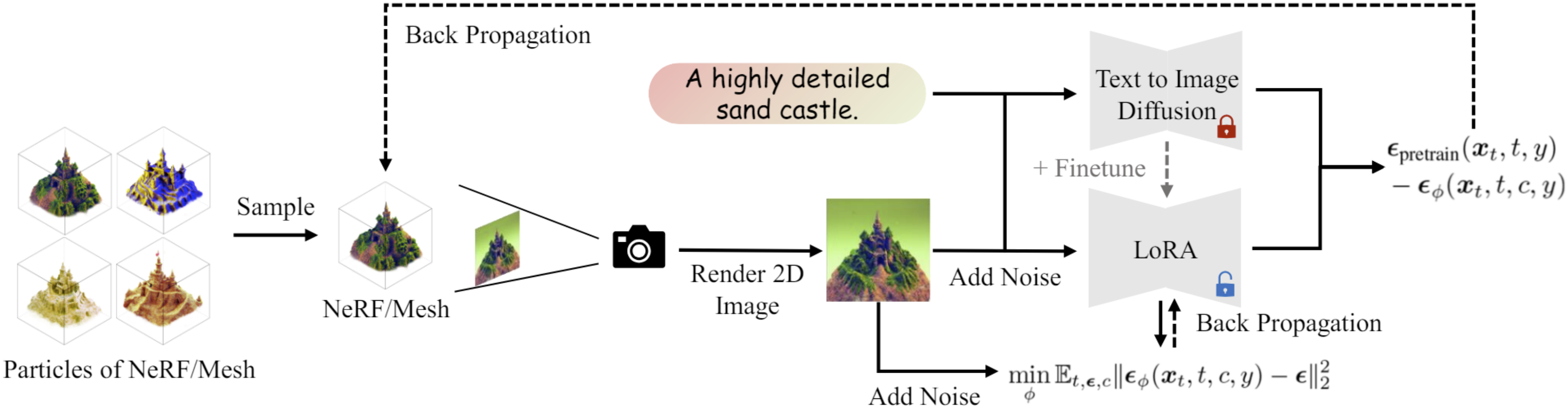
[Poole et al. 22] DreamFusion

DreamFusion

[Poole et al. 22] DreamFusion

SDS Follow Ups

- ProlificDreamer (variational SDS)



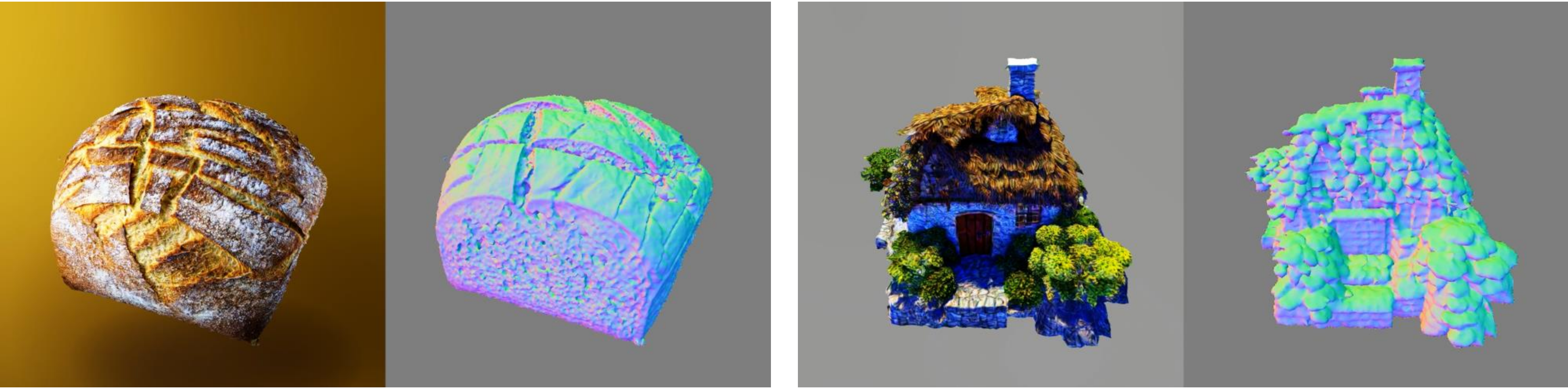
$$\nabla_\theta \mathcal{L}_{\text{VSD}}(\theta) \triangleq \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[\omega(t)\left(\boldsymbol{\epsilon}_{\text{pretrain}}(\boldsymbol{x}_t, t, y^c) - \boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t, t, c, y)\right)\frac{\partial \boldsymbol{g}(\theta, c)}{\partial \theta}\right],$$

where $\boldsymbol{x}_t = \alpha_t \boldsymbol{g}(\theta, c) + \sigma_t \boldsymbol{\epsilon}$.

[Wang et al. 23] ProlificDreamer

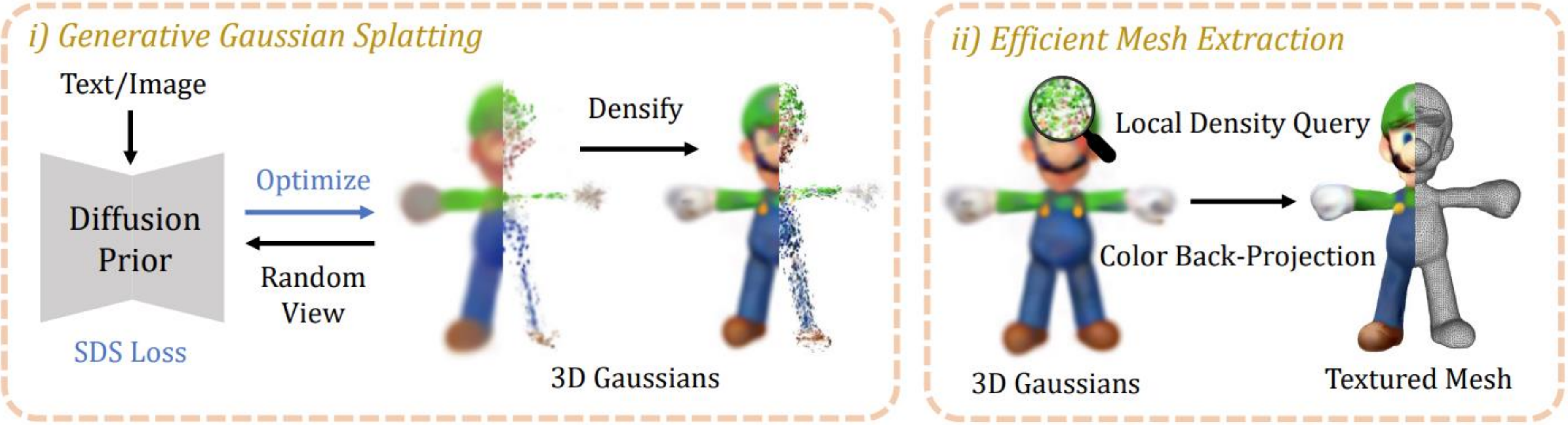
SDS Follow Ups

- ProlificDreamer (variational SDS)



[Wang et al. 23] ProlificDreamer

DreamGaussian

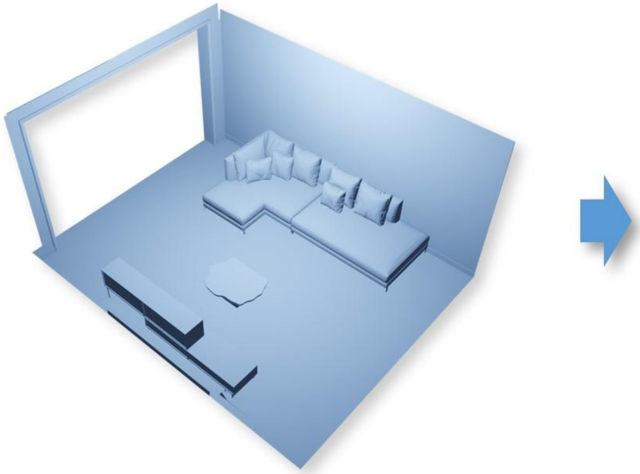


[Tang et al.. 24] Dream Gaussian

DreamGaussian

[Tang et al.. 24] Dream Gaussian

SceneTex


Scene geometry

Scene with generated texture

[Chen et al.. 24] SceneTex


Baroque
Bohemian
French
Japanese

Administrative: Lecture Evaluation

Diffusion Models

Reading Homework

- Denoising Diffusion Probabilistic Models.
  - https://arxiv.org/abs/2006.11239
- Classifier Guided Diffusion. Diffusion Models Beat GANs on Image Synthesis
  - https://arxiv.org/abs/2105.05233
- Classifier-Free Guidance. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models
  - https://arxiv.org/abs/2112.10741
- CLIP Guidance. Hierarchical Text-Conditional Image Generation with CLIP Latents
  - https://arxiv.org/abs/2204.06125

Literature

- CVPR 2022 Tutorial on Denoising Diffusion-based Generative Modeling
  - https://cvpr2022-tutorial-diffusion-models.github.io/
- Tackling the Generative Learning Trilemma with Denoising Diffusion GANs
  - https://arxiv.org/abs/2112.07804
- Deep Unsupervised Learning using Nonequilibrium Thermodynamics
  - https://arxiv.org/abs/1503.03585
- Denoising Diffusion Probabilistic Models
  - https://arxiv.org/abs/2006.11239
- Diffusion Models Beat GANs on Image Synthesis
  - https://arxiv.org/abs/2105.05233

Thanks for watching!