Diffusion Models for Visual Content Creation

Niloy Mitra, Duygu Ceylan, Paul Guerrero,

Daniel Cohen-Or, Or Patashnik, Chun-Hao Huang, Minhyuk Sung

Part 1: Introduction to Diffusion Models

https://geometry.cs.ucl.ac.uk/courses/diffusion4ContentCreation_sigg24/

People

Niloy Mitra

Duygu Ceylan

Paul Guerrero

Daniel Cohen-Or

Or Patashnik

Chun-Hao Huang

Minhyuk Sung

Why do we need this Tutorial?

What are diffusion model?

What are the many design choices?

Interpretation, controls and adaptation in the context of Visual Computing

Many Related Materials

- Survey papers
- Past tutorials and courses
- Blogs and recorded videos

Presentation Schedule

Introduction to Diffusion Models

Guidance and Conditioning Sampling

Attention

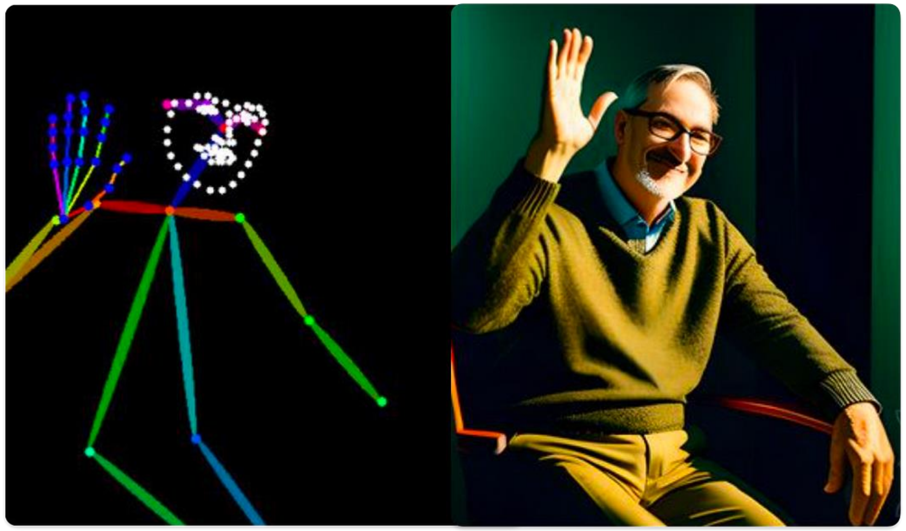
Break

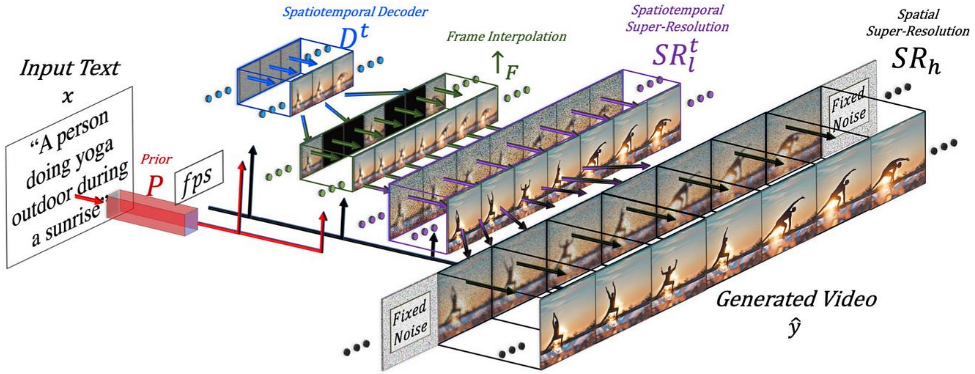
Personalization and Editing

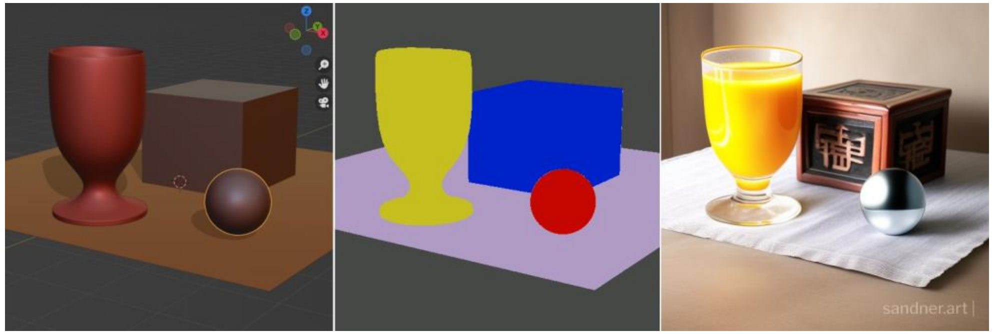
Beyond Single (RGB) Image Generation

Diffusion Models for 3D Generation

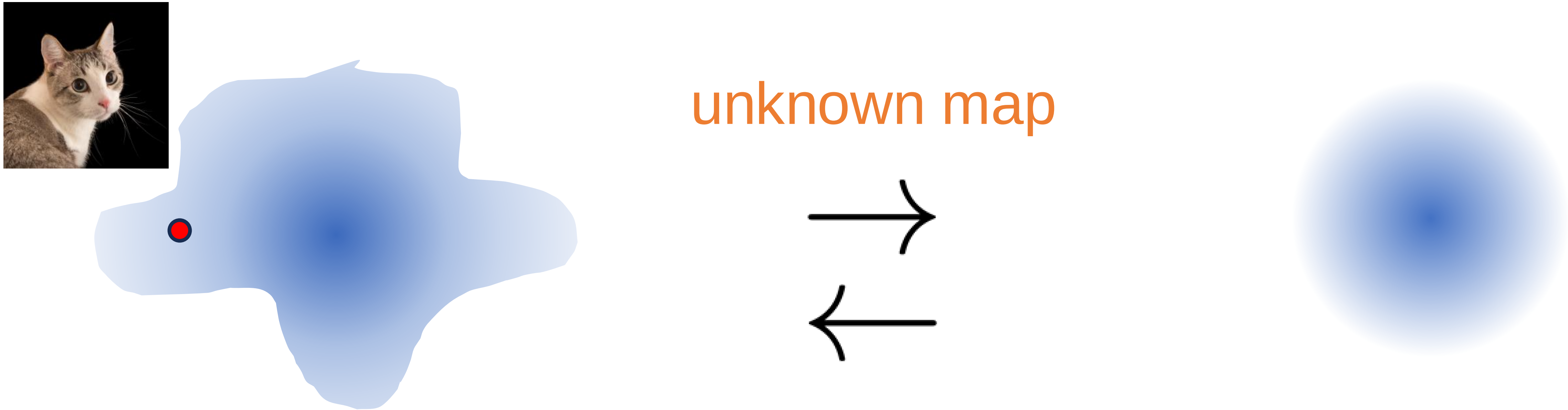
Images, Video, and Beyond







What is a Diffusion Process?



(unknown) data distribution

known distribution

Sampling $\Leftrightarrow$ *(Unconditional generation)*

Mapping between Distributions

$\mathbf{x}_0$ $\longrightarrow$ $\mathbf{x}_T$

$\mathcal{N}(\mathbf{0}, \mathbb{I})$

data distribution

known distribution

Gaussian (Normal) Distribution

- Uniquely defined by Mean and Variance

$$\mu, \mathbf{\Sigma}$$

- Reparameterization 'trick'

$$\mathcal{N}(\mu, \mathbf{\Sigma})$$

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

$$y_i = \mu_i x_i + \sigma_i$$

- Many results on combining Gaussian distributions

Mapping in Many Steps

$\longrightarrow$

ward mapping

$$\mathbf{x}_0 \rightarrow \quad \mathbf{x}_{t-1} \rightarrow \mathbf{x}_t \quad \rightarrow \quad \rightarrow \mathbf{x}_T$$

$$\mathcal{N}(\mathbf{0}, \mathbb{I})$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)\mathbb{I})$$

Mapping in Many Steps

$\longrightarrow$

forward mapping

$$\mathbf{x}_0 \rightarrow \qquad \mathbf{x}_{t-1} \rightarrow \mathbf{x}_t \qquad \rightarrow \qquad \rightarrow \mathbf{x}_T$$

$$\mathcal{N}(\mathbf{0}, \mathbb{I})$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)\mathbb{I})$$

Mapping in Many Steps

$\longrightarrow$

forward mapping

$$\mathbf{x}_0 \rightarrow \qquad \mathbf{x}_{t-1} \rightarrow \mathbf{x}_t \qquad \rightarrow \qquad \rightarrow \mathbf{x}_T$$

$$\mathcal{N}(\mathbf{0}, \mathbb{I})$$

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t} x_{t-1}, (1-\alpha_t)\mathbb{I})$$

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t} x_0, (1-\overline{\alpha}_t)\mathbb{I}) \qquad \overline{\alpha}_t = \Pi_{i=1}^{t} \alpha_i$$

Mapping in Many Steps

$\longrightarrow$

forward mapping

$$\mathbf{x}_0 \rightarrow \qquad \mathbf{x}_{t-1} \rightarrow \mathbf{x}_t \qquad \rightarrow \qquad \rightarrow \mathbf{x}_T$$

$$\mathcal{N}(\mathbf{0}, \mathbb{I})$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)\mathbb{I})$$

$$x_t = \hat{\epsilon}(x_0) = \sqrt{\overline{\alpha}_t}x_0 + \sqrt{(1-\overline{\alpha}_t)}\epsilon_t \qquad \bar{\alpha}_t = \Pi_{i=1}^{t}\alpha_i$$

Generative Modeling: Sampling

$$\longleftarrow$$

reverse mapping

$$\mathbf{x}_0 \rightarrow \qquad \mathbf{x}_{t-1} \rightarrow \mathbf{x}_t \qquad \rightarrow \qquad \rightarrow \mathbf{x}_T$$

$$\mathcal{N}(\mathbf{0}, \mathbb{I})$$

$$q(x_{t-1}|x_t)$$

$$x_t = \sqrt{\overline{\alpha}_t} x_0 + \sqrt{(1 - \overline{\alpha}_t)} \epsilon_t$$

$$\approx$$

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \sqrt{\overline{\alpha}_{t-1}} D_\theta(x_t, t), (1 - \overline{\alpha}_{t-1})\mathbb{I})$$

Loss Functions

$$\mathcal{L}_{simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}[C_t \parallel \epsilon_\theta(x_t, t) - \epsilon \parallel^2]$$

$$\mathcal{L}(\theta) = \mathbb{E}_{t,\epsilon,x_0}\left[C_t \parallel D_\theta(\hat{\epsilon}_t(x_0), t) - x_0 \parallel^2\right]$$

$$p(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \sqrt{\overline{\alpha}_{t-1}} D_\theta(x_t, t), (1 - \overline{\alpha}_{t-1})\mathbb{I})$$

Algorithm (How to Train?)

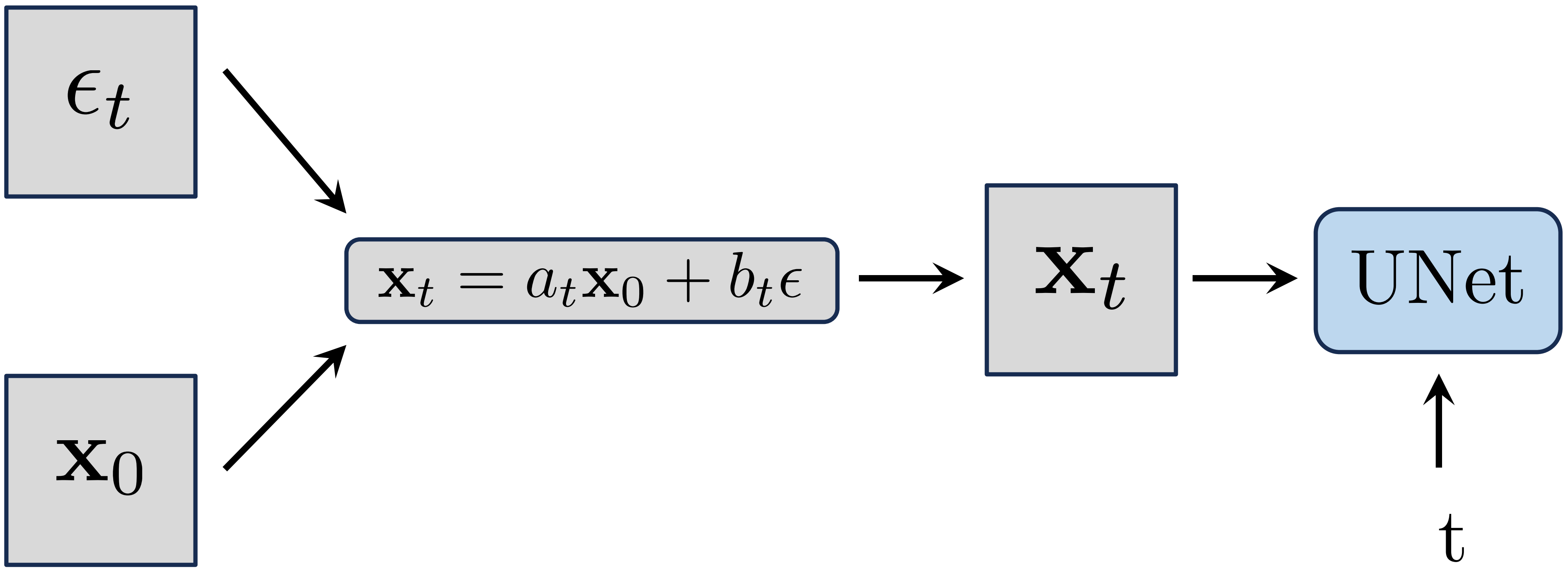
Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

6: **until** converged

Training Loss



Loss Functions: Three Interpretations

1. Predict Noise ϵ_t

2. Predict clean image $\mathbf{x}_0$

3. Score-based optimization $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_0) = -\frac{\epsilon}{\sqrt{1 - \bar{\alpha_t}}}$

they are equivalent!!

Algorithm (How to Sample?)

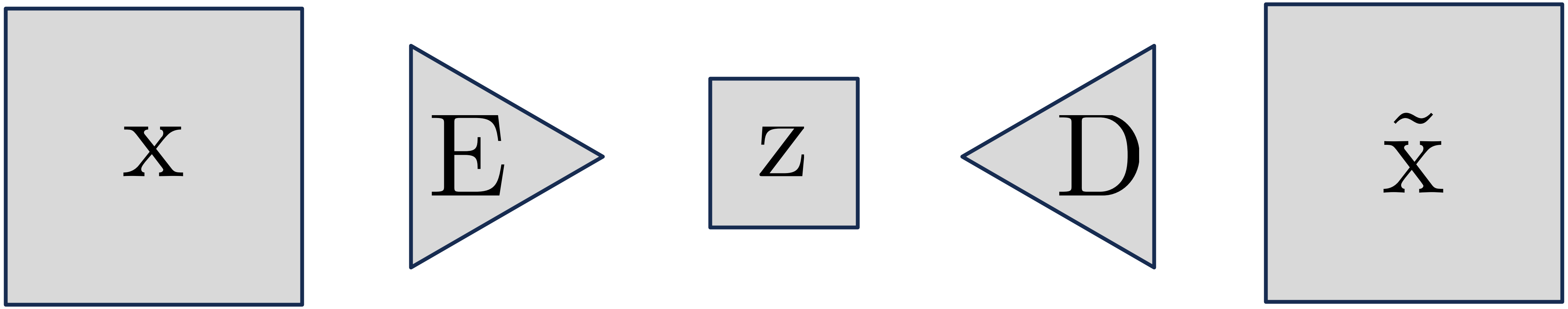
Algorithm 2 Sampling

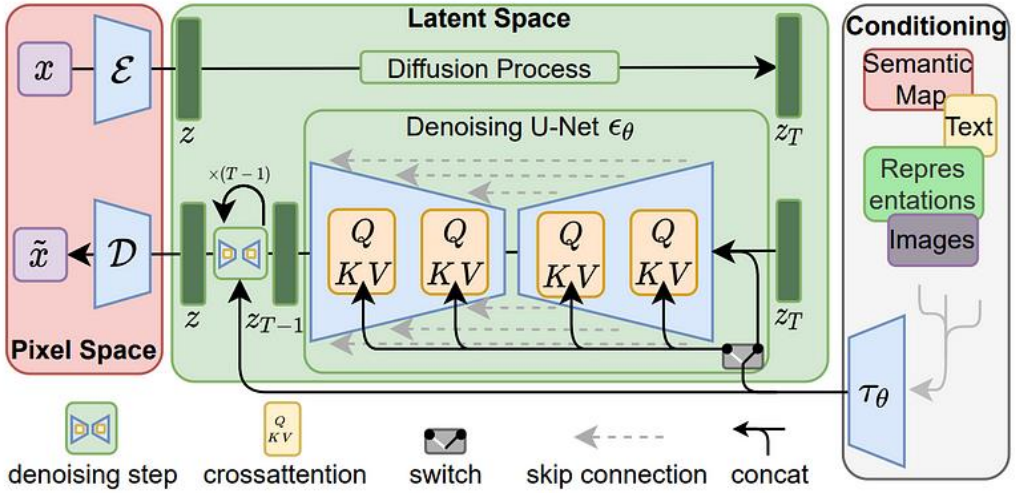
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

What’s Special about Visual Data?

- Dimensionality of the problem
- Inference speed and diversity of generations
- Training data: we have many (differentiable) known functions
- Media specific-losses and semantics of data
- Types of controls

Latent Diffusion Model

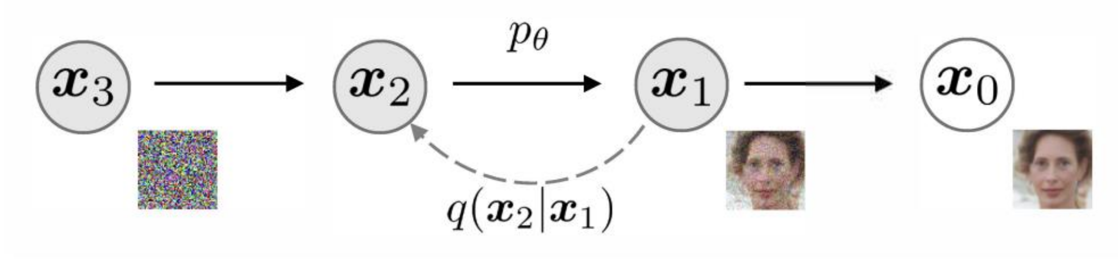




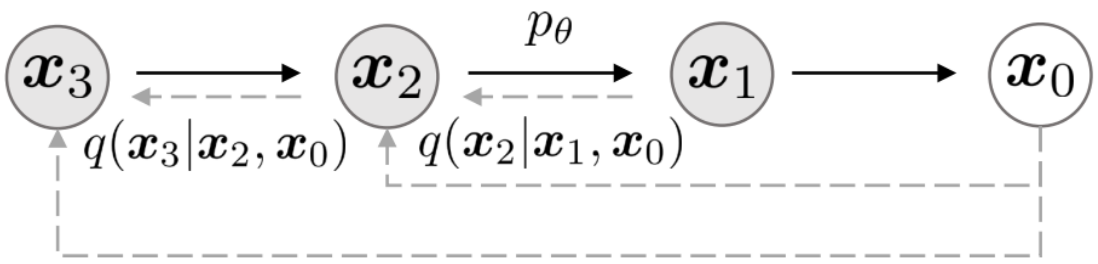
[High-Resolution Image Synthesis with Latent Diffusion Models, Rombach et al., Arxiv 2021]

Faster Inference: DDPM vs DDIM

- DDPM: Markovian process



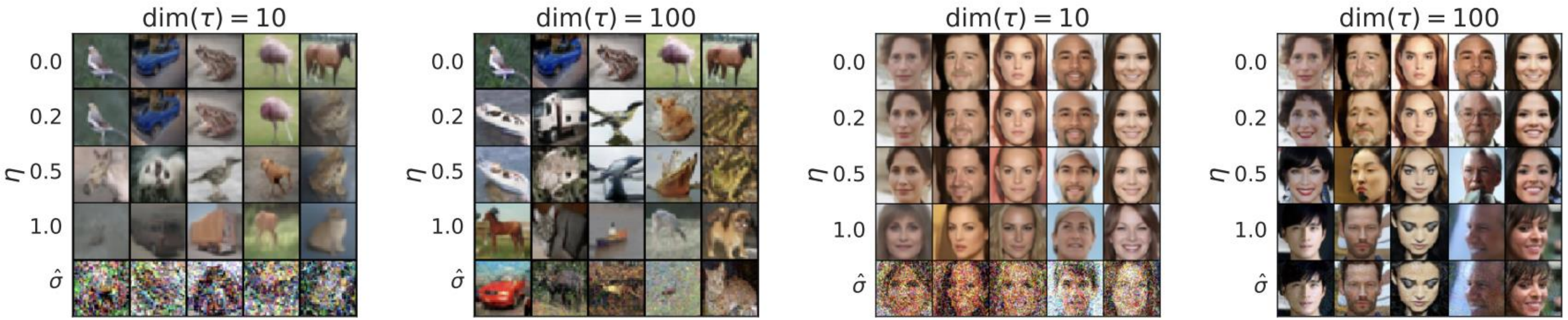
- DDIM: Non-Markovian process but 10-50x faster!!
  - Trained w/ pretrained DDPM diffusion



$$x_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{" predicted } x_0 \text{"}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{"direction pointing to } x_t \text{"}} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

DDPM vs DDIM

	S	CIFAR10 (32 × 32) 10	20	50	100	1000	CelebA (64 × 64) 10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
	$\hat{\sigma}$	367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**



Summary so far

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Skipped Concepts

- CLIP space (linking images with text)
- LORA (finetuning with limited data)
- Image inversion (DDIM inversion) for real images
- Training schedule

Presentation Schedule

Introduction to Diffusion Models

Guidance and Conditioning Sampling

Attention

Break

Personalization and Editing

Beyond Single (RGB) Image Generation

Diffusion Models for 3D Generation