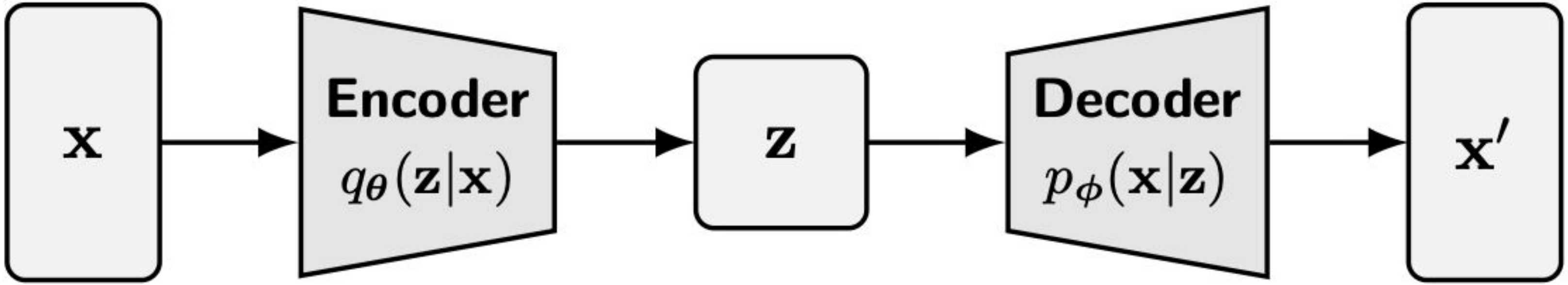
Diffusion

Background: from VAE to Diffusion Models



VAE comes from Bayesian statistics and Information theory. Two main problems:

1. Both encoder and decoder have the difficult task of transforming very complex probability distributions, in very high dimensional spaces.
2. Encoder and decoder are trained jointly, blurriness, mode collapse, hierarchical VAE improves but at cost of great training problems

Can we fix on of the two directions?

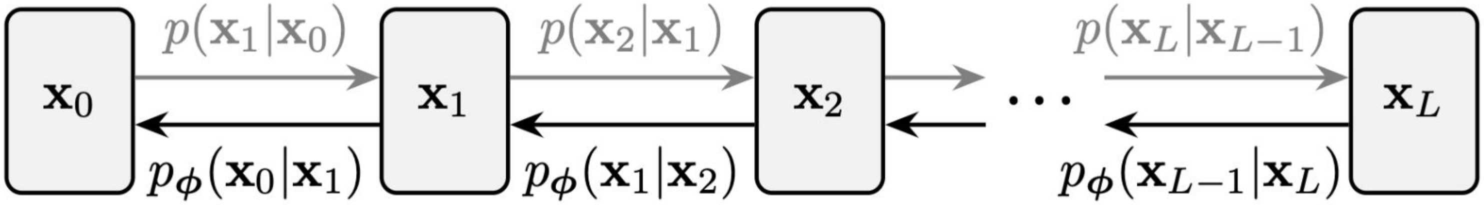
Diffusion models overcome VAE limitation by

- Fixing a rule of encoding
- Optimizing the decoding

How? Using a (controlled) diffusion model. The idea is rather simple

- Forward pass: specify a dynamic to “corrupt” an image and make it become white noise. How? Why?
- Backward pass: reverse the process (**one step at the time**) and get back an image.

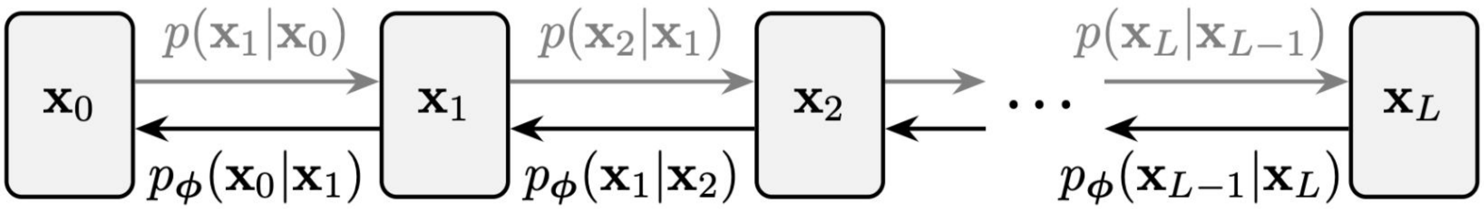
Sohl-Dickstein et. al, Deep Unsupervised Learning using Nonequilibrium Thermodynamics.



Forward pass:

- How? Through a gaussian semi-fixed kernel → it can be seen as a OU process → convergence to a gaussian (white noise) in the limit.
- Why? It is a lot easier computationally to make steps to perturbe!
- Next paper: Ho&al: there is no benefit in making the kernel semi-fixed, let's fix it and have an approximate (but more stable and easily trainable) backward

Sohl-Dickstein et. al, Deep Unsupervised Learning using Nonequilibrium Thermodynamics.



Backward pass in steps:

- Why?
  a. It is a lot easier computationally to make steps to perturbe!
  b. Turns out that backward steps are also gaussian kernels (Feller, 1949) (or binomial in the discrete setting).
- How?
  a. We can learn mean vector and covariance matrix of each step approximating with MLP.
- Advantage wrt VAE is that instead of trying to approximate a whole distribution, we just have to approximate two moments at each step (drastically easier)!

Similar to VAE, what are we optimizing? ELBO!

Some math into the optimization problem

Notation:

- q for forward related, p for backward probabilities, i.e. $q(x_t|x_{t-1})$, $p(x_{t-1}|x_t)$
- $p(x^{(0,...,T)})$, $q(x^{(0,...,T)})$ are the probabilities across the whole sequence and

$$q(x^{(0,\ldots,T)}) = q(x^0)\prod_{t=1}^{T} q(x^t|x^{t-1}) \qquad p(x^{(0,\ldots,T)}) = p(x^T)\prod_{t=1}^{T} p(x^{t-1}|x^t)$$

- The probability of a sample can be written as

$$\begin{aligned} p(\mathbf{x}^{(0)}) &= \int d\mathbf{x}^{(1:T)}\, p(\mathbf{x}^{(0:T)}) \frac{q(\mathbf{x}^{(1:T)} \mid \mathbf{x}^{(0)})}{q(\mathbf{x}^{(1:T)} \mid \mathbf{x}^{(0)})} \\ &= \int d\mathbf{x}^{(1:T)}\, q(\mathbf{x}^{(1:T)} \mid \mathbf{x}^{(0)})\, p(\mathbf{x}^{(T)}) \prod_{t=1}^{T} \frac{p(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)})} \end{aligned}$$

MLE and ELBO

The log-likelihood function is given by

$$L = \int d\mathbf{x}^{(0)}\, q(\mathbf{x}^{(0)}) \log p(\mathbf{x}^{(0)}).$$

- As in VAE, it is actually computationally expensive to compute.
- Instead, by using Jensen's inequality, we maximise a lower bound

$$L \geq \underbrace{\boxed{\int d\mathbf{x}^{(0:T)}\, q(\mathbf{x}^{(0:T)}) \log\left[p(\mathbf{x}^{(T)}) \prod_{t=1}^{T} \frac{p(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)} \mid \mathbf{x}^{(t-1)})}\right]}}_{K}$$

We will optimize the backward kernels to maximise K, and since they are gaussian, we maximize mean and covariances!

Interpretation of K (ELBO)

$$K = \boxed{-\sum_{t=2}^{T} \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)}\, q(\mathbf{x}^{(0)}, \mathbf{x}^{(t)})\, D_{\mathrm{KL}}\left(q(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)}, \mathbf{x}^{(0)}) \,\|\, p(\mathbf{x}^{(t-1)} \mid \mathbf{x}^{(t)})\right)}$$

$$+ \boxed{H_q(\mathbf{X}^{(T)} \mid \mathbf{X}^{(0)})} - \boxed{H_q(\mathbf{X}^{(1)} \mid \mathbf{X}^{(0)})} - \boxed{H_p(\mathbf{X}^{(T)})}$$

$$H_q(X) = -\int q(x) \log q(x)\, dx \qquad D_{\mathrm{KL}}(q \,\|\, p) = \sum_x q(x) \log \frac{q(x)}{p(x)}$$

- In blue it is a weighted average of the distance between the forward and backward kernel at the stage t (reconstruction).
- In orange we want to make sure that no matter the initial distribution, we have as high as possible entropy. This is because we want to have a fully noisy end of the diffusion (randomization).
- In green we have a penalty for abrupt changes in the first step (which has a fixed beta).
- Finally, in red we have the entropy wrt to p (simple gaussian), so we want to avoid the end of the diffusion process to be too far away from the simple prior we will be sampling from.

Reverse Diffusion Process

🎯 This is our Generative Model!

📐 Mathematical Definition

Initial distribution : $p(x^{(T)}) = \mathcal{N}(0, I)$

Single step:

$p(x^{(t-1)}|x^{(t)}) = \mathcal{N}(x^{(t-1)}; f\mu(x^{(t)},t), f\Sigma(x^{(t)},t))$

Full trajectory: $p(x^{(0...T)}) = p(x^{(T)}) \prod_{t=1}^{T} p(x^{(t-1)}|x^{(t)})$

💡 Theoretical Foundation

Kolmogorov Forward-Backward Equations: For small β_t, forward and reverse processes have the SAME functional form!

What to Learn

- $f_\mu(x^{(t)}, t)$: Mean function

- $f_\Sigma(x^{(t)}, t)$: Covariance function

- Parameterized by neural networks

🎲 Sampling

$x^{(T)} \sim \mathcal{N}(0,I) \rightarrow x^{(T-1)} \rightarrow \ldots \rightarrow x^{(0)}$

Training Objective

Key Advantages

- Analytically computable!

Both distributions are Gaussian

- Reduces to regression problem

Learn mean and variance functions

- Tight bound when $\beta_t \to 0$

Quasi-static limit from physics

Five Key Advantages

1️⃣ Flexibility

• Can model ANY smooth distribution

• Theoretical guarantee exists

2️⃣ Exact Sampling

• No MCMC needed

• Deterministic procedure

3️⃣ Tractable Likelihood

• Closed-form evaluation

• Reliable model comparison

4️⃣ Easy Conditioning

Posterior computation is simple and Useful for inpainting, denoising. Deep Architecture:• Thousands of layers/time steps • Each step is simple

Four Key Theoretical Results

- Kolmogorov Equations

Justifies using Gaussian kernels

- Entropy Bounds
- Quasi-static Limit

Physics analogy is rigorous

More steps = better approximation

- Posterior Multiplication

From Overlooked to Revolutionary

2015 ————→ 2020 ————→ 2021 ————→ 2022 ————→ 2024

Published Explosion Quality Commercial Everywhere

(~50/yr) (1000+/yr) (3000+/yr) (5000+/yr) (>15K total)

Why This Paper Matters

1. Introduced complete diffusion framework
2. Connected physics and deep learning
3. Achieved flexibility + tractability
4. Foundation for modern generative AI

Limitations & Future Directions

In 2015, diffusion models were limited by slow sampling (about 1,000 forward passes), high training costs due to modeling many time steps, and lower image quality compared to GANs. Later advances solved these issues with faster samplers like DDIM, improved quality through better parameterization and guidance, and greater efficiency using distillation methods and optimized architectures.

A following work is DDPM – next paper(new idea, do not predict the mean, predict the noise we added)

Ho et. al, Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems (2020).

Reverse distribution is Gaussian → can we learn its mean & variance?

$$q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\, x_{t-1},\ \beta_t I)$$

$$q(x_t \mid x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t) I), \quad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(\tilde{\mu}_t,\ \tilde{\beta}_t I)$$

$$\tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\, \beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\, \beta_t$$

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

In practice, no. This is why early diffusion models failed.

- Scaling mismatch: the reverse means have very different magnitudes across timesteps, making the parameterization poorly conditioned.
- Mean–variance interaction: at large t, the true mean becomes very small while the true variance becomes large. This causes vanishing gradients for the mean and exploding gradients for the variance.

Ho et. al, Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems (2020).

Revival of diffusion models: reparameterize in terms of the noise.

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 \; + \; \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t),\, \sigma_t^2 I) \quad (\text{fix } \sigma_t^2 \text{ to known posterior variance})$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\, \epsilon_\theta(x_t, t) \right)$$

$$L_{\text{simple}}(\theta) = \mathbb{E}_{x_0, t, \epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right]$$

- ε has fixed scale across timesteps, so learning remains well-conditioned.
- Fixing the reverse variance removes timestep-dependent weighting, giving a simple uniform loss.
- From ε we can recover a unbiased estimate x_0 and then compute the reverse mean deterministically

Ho et. al, Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems (2020).

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	**10.06**	**2.67**	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			**2.80**
PixelIQN [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]		31.75	
NCSN [55]	8.87 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic $\boldsymbol{\Sigma}$)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 3.75 (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\boldsymbol{\mu}}$ prediction (baseline)		
L, learned diagonal $\boldsymbol{\Sigma}$	7.28 ± 0.10	23.69
L, fixed isotropic $\boldsymbol{\Sigma}$	8.06 ± 0.09	13.22
$\lVert \tilde{\boldsymbol{\mu}} - \tilde{\boldsymbol{\mu}}_\theta \rVert^2$	–	–
$\boldsymbol{\epsilon}$ prediction (ours)		
L, learned diagonal $\boldsymbol{\Sigma}$	–	–
L, fixed isotropic $\boldsymbol{\Sigma}$	7.67 ± 0.13	13.51
$\lVert \tilde{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}_\theta \rVert^2$ (L_{simple})	$\mathbf{9.46 \pm 0.11}$	**3.17**

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
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** $\mathbf{x}_0$

Ho et. al, Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems (2020).

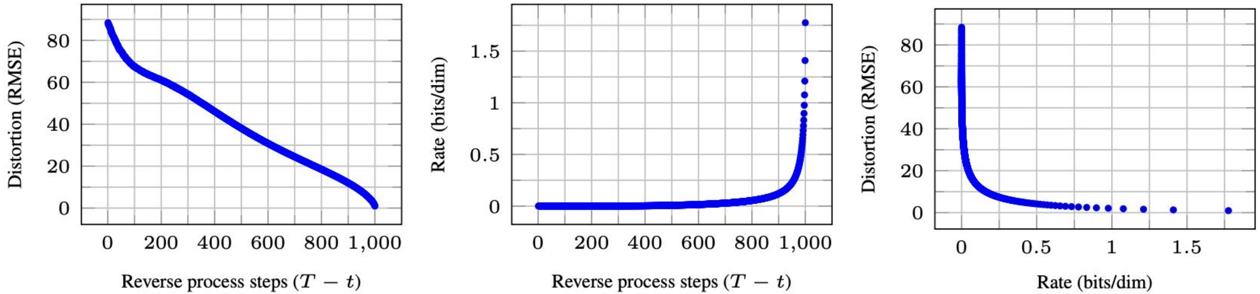


Figure 5: Unconditional CIFAR10 test set rate-distortion vs. time. Distortion is measured in root mean squared error on a $[0, 255]$ scale. See Table 4 for details.

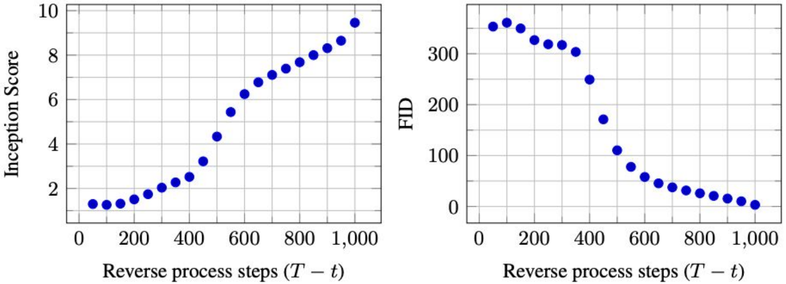
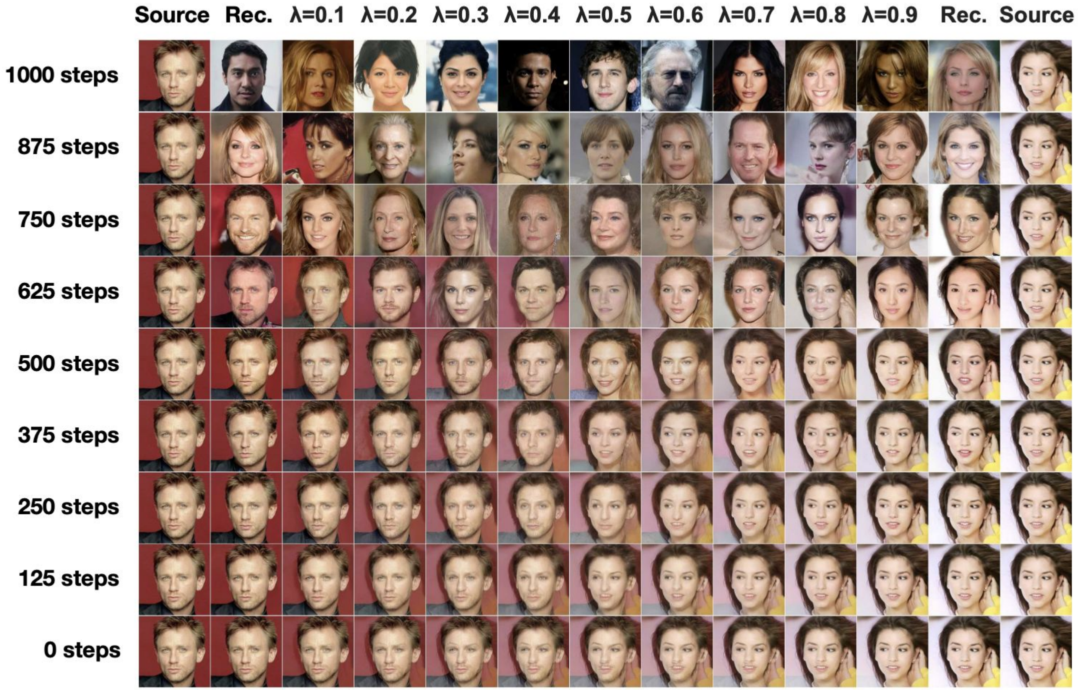


Figure 10: Unconditional CIFAR10 progressive sampling quality over time

Ho et. al, Denoising Diffusion Probabilistic Models. Advances in Neural Information Processing Systems (2020).



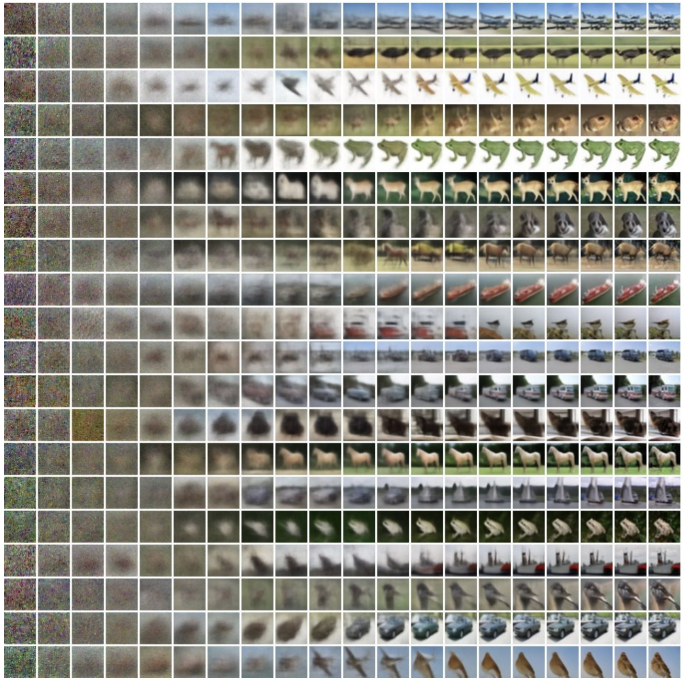


Figure 14: Unconditional CIFAR10 progressive generation

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Main problem with diffusion models: **High computational cost**

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Main problem with diffusion models: **High computational cost**

Example: OpenAI (Dhariwal-Nichol 2021) trained a series of diffusion models over 150 - 1000 GPU days (V100)

Comparison: training ResNet-50 took about 4 GPU days

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Main problem with diffusion models: **High computational cost**

Example: OpenAI (Dhariwal-Nichol 2021) trained a series of diffusion models over 150 - 1000 GPU days (V100)

Comparison: training ResNet-50 took about 4 GPU days

Evaluation is also costly: The above OpenAI model takes ~5 days on a single A100 to product 50,000 samples.

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Main problem with diffusion models: **High computational cost**

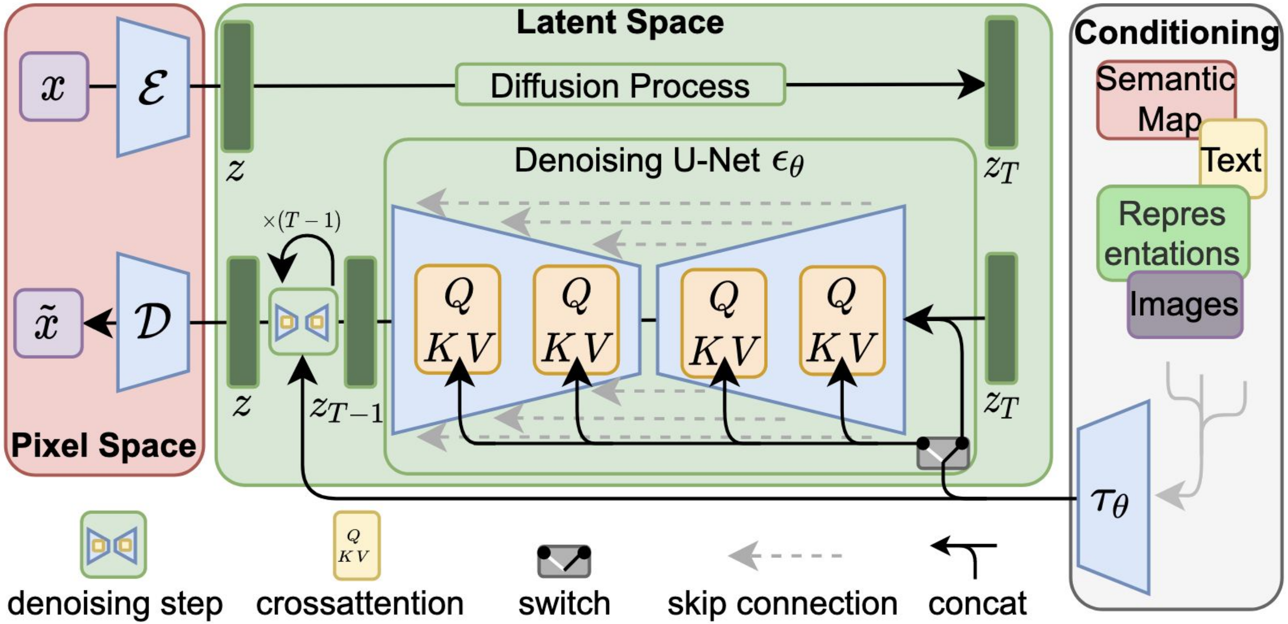
Example: OpenAI (Dhariwal-Nichol 2021) trained a series of diffusion models over 150 - 1000 GPU days (V100)

Comparison: training ResNet-50 took about 4 GPU days

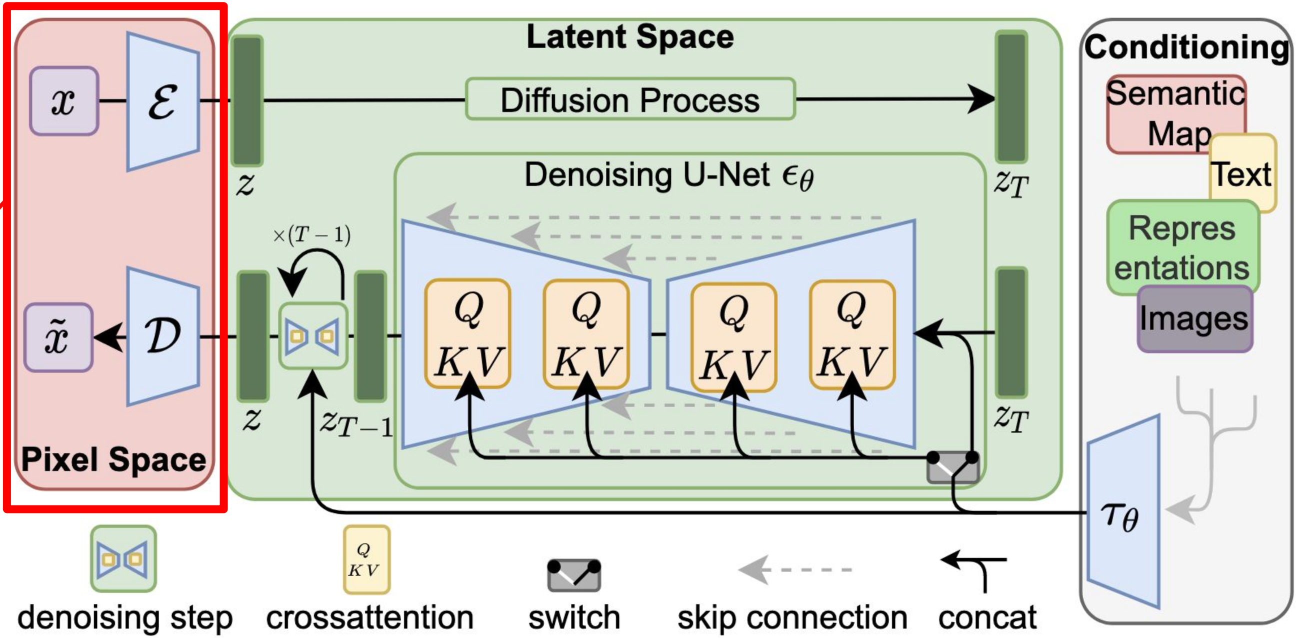
Evaluation is also costly: The above OpenAI model takes ~5 days on a single A100 to product 50,000 samples.

Solution: use autoencoder to obtain lower-dimensional representation of images, and train diffusion model on lower-dimensional space

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).



Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).



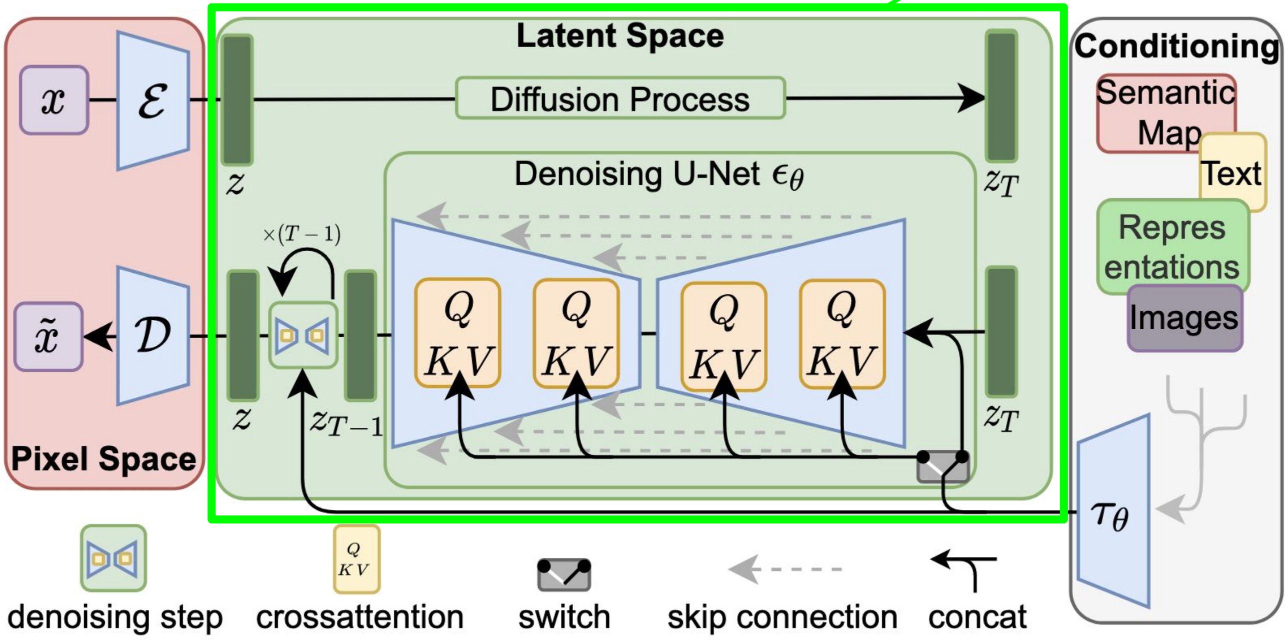
Images occur as both input and output of model.

They are given in terms of their pixels.

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

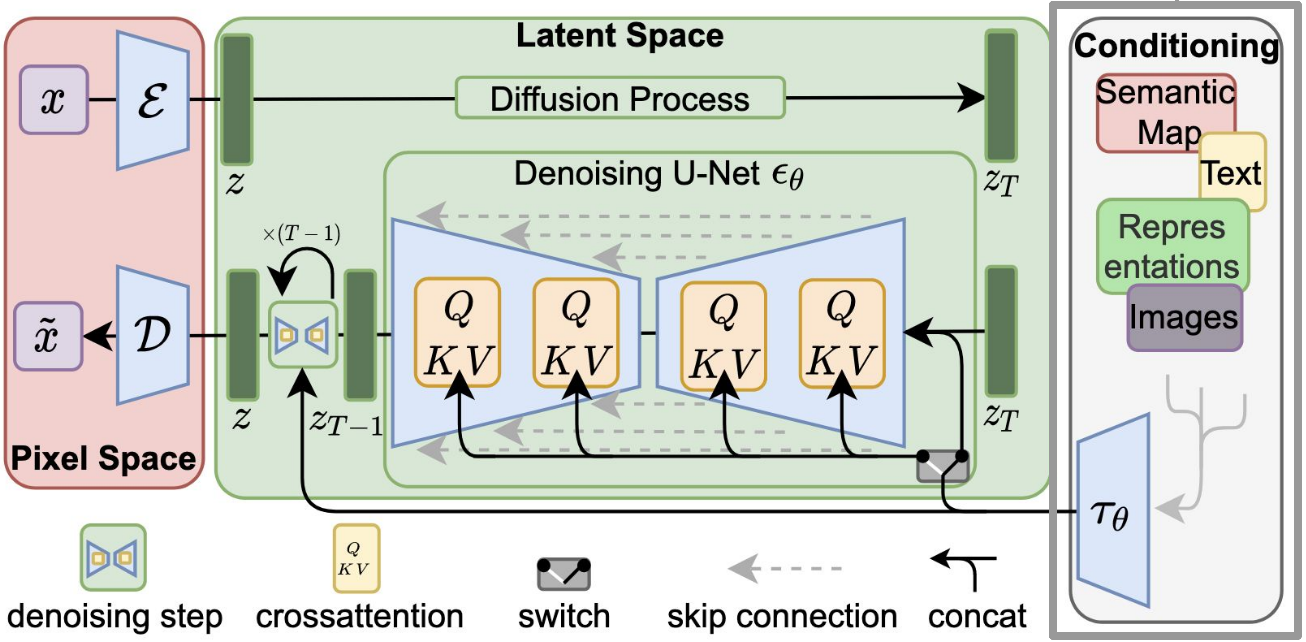
Diffusion model is trained and evaluated in “latent space”, given by downsampling images

factor f = H/h = W/w = 2^m



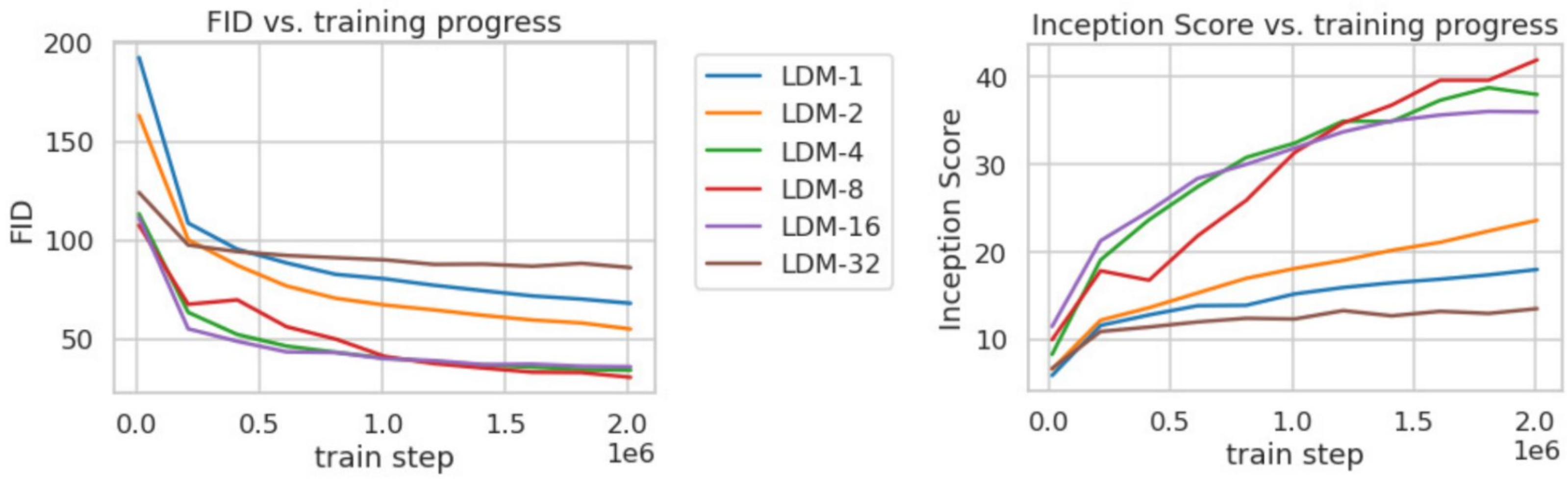
Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

UNet is augmented by cross-attention mechanism, allowing for influence from text, images, etc.



Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Results:

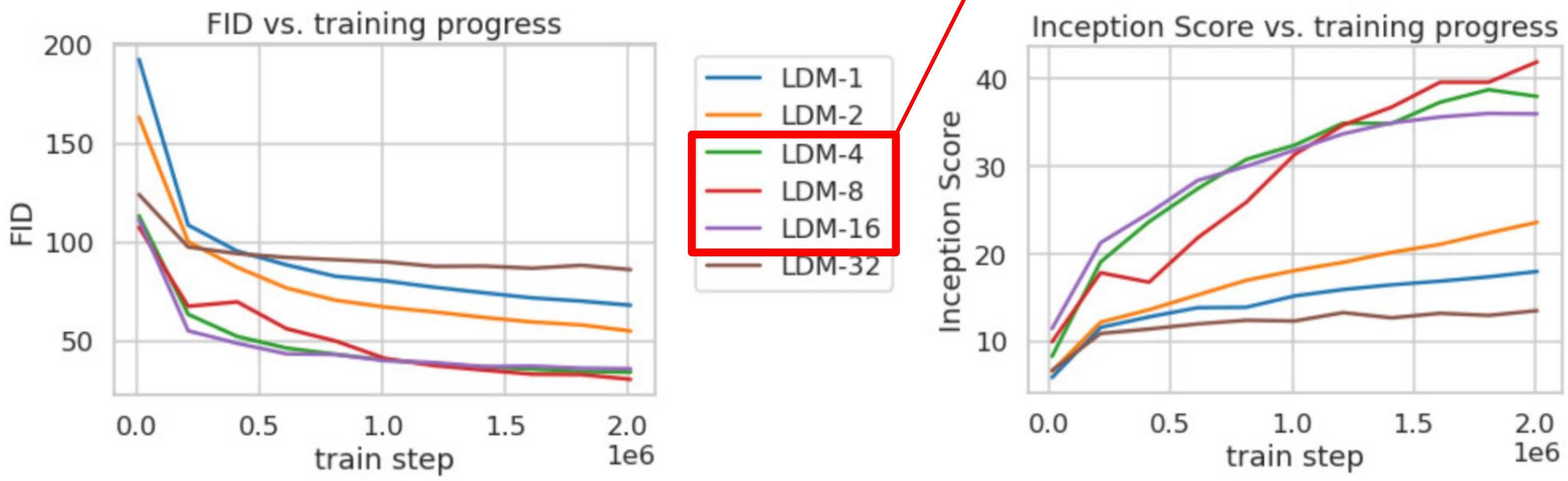


Good (low) FID: greater sample variety
Good (high) inception score: higher individual image quality

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Good balance of efficiency and faithfulness

Results:



Good (low) FID: greater sample variety
Good (high) inception score: higher individual image quality

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

CelebA-HQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
DC-VAE [63]	15.8	-	-
VQGAN+T. [23] (k=400)	10.2	-	-
PGGAN [39]	8.0	-	-
LSGM [93]	7.22	-	-
UDM [43]	7.16	-	-
LDM-4 (ours, 500-s†)	**5.11**	0.72	0.49

FFHQ 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
ImageBART [21]	9.57	-	-
U-Net GAN (+aug) [77]	10.9 (7.6)	-	-
UDM [43]	5.54	-	-
StyleGAN [41]	4.16	0.71	0.46
ProjectedGAN [76]	**3.08**	0.65	0.46
LDM-4 (ours, 200-s)	4.98	**0.73**	**0.50**

LSUN-Churches 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
DDPM [30]	7.89	-	-
ImageBART [21]	7.32	-	-
PGGAN [39]	6.42	-	-
StyleGAN [41]	4.21	-	-
StyleGAN2 [42]	3.86	-	-
ProjectedGAN [76]	**1.59**	0.61	0.44
*LDM-8** (ours, 200-s)	4.02	**0.64**	**0.52**

LSUN-Bedrooms 256 × 256			
Method	FID ↓	Prec. ↑	Recall ↑
ImageBART [21]	5.51	-	-
DDPM [30]	4.9	-	-
UDM [43]	4.57	-	-
StyleGAN [41]	2.35	0.59	0.48
ADM [15]	1.90	**0.66**	**0.51**
ProjectedGAN [76]	**1.52**	0.61	0.34
LDM-4 (ours, 200-s)	2.95	**0.66**	0.48

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Text-to-speech

Text-Conditional Image Synthesis				
Method	FID ↓	IS↑	N_{params}	
CogView† [17]	27.10	18.20	4B	self-ranking, rejection rate 0.017
LAFITE† [109]	26.94	<u>26.02</u>	75M	
GLIDE* [59]	<u>12.24</u>	-	6B	277 DDIM steps, c.f.g. [32] $s = 3$
Make-A-Scene* [26]	**11.84**	-	4B	c.f.g for AR models [98] $s = 5$
LDM-KL-8	23.31	$20.03_{\pm 0.33}$	$\boxed{1.45\text{B}}$	250 DDIM steps
*LDM-KL-8-G**	12.63	$\mathbf{30.29}_{\pm \mathbf{0.42}}$	$\boxed{1.45\text{B}}$	250 DDIM steps, c.f.g. [32] $s = 1.5$

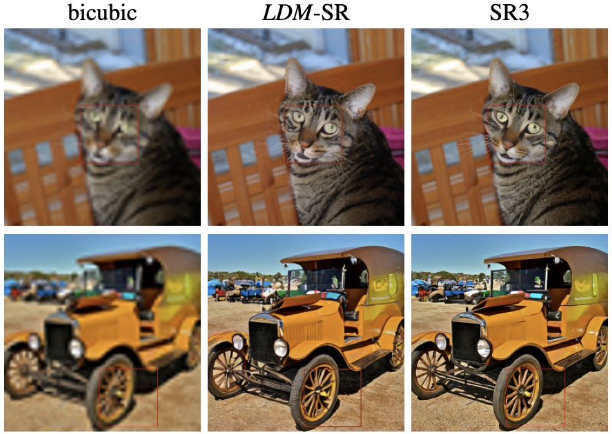
Classifier-free guidance

Substantially fewer parameters

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Super-resolution and inpainting

Super-resolution on ImageNet



Inpainting on Places

Task 1: blind preference between model and original picture

Task 2: blind preference between two different models

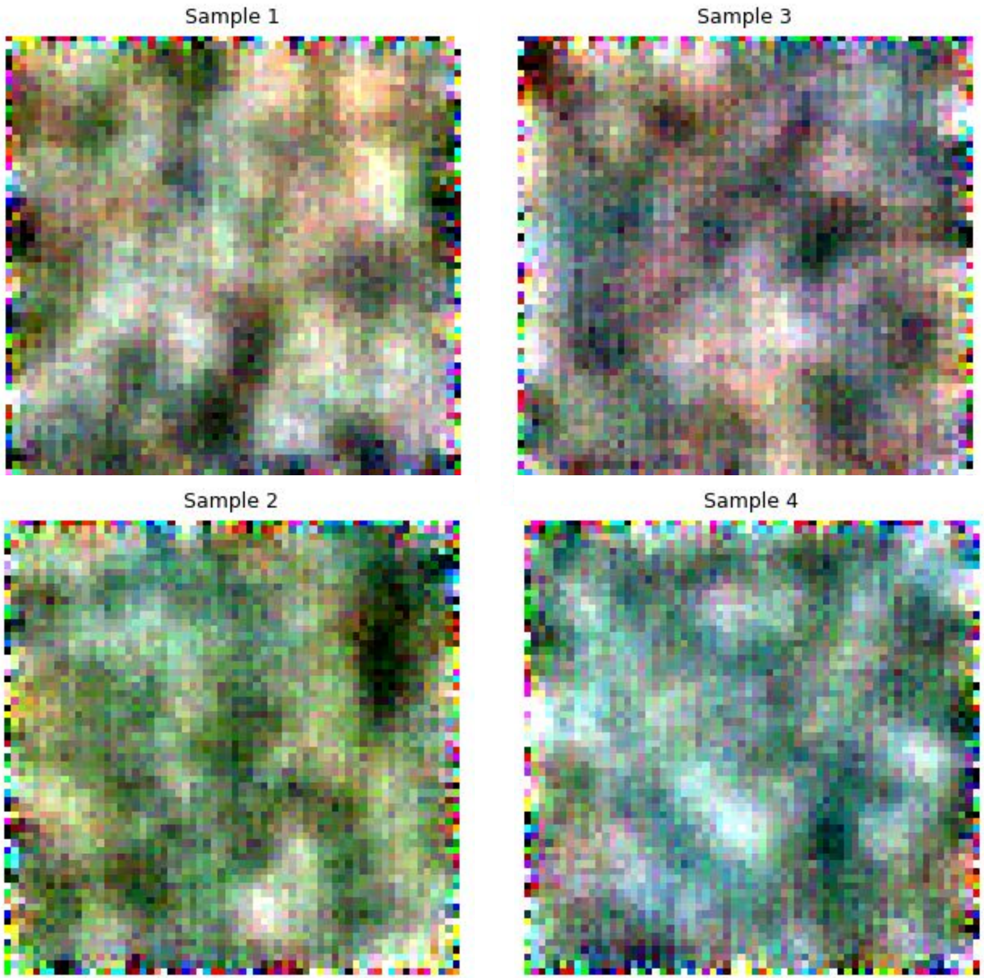
User Study	SR on ImageNet		Inpainting on Places	
	Pixel-DM ($f1$)	*LDM-4*	LAMA [88]	*LDM-4*
Task 1: Preference vs GT ↑	16.0%	**30.4%**	13.6%	**21.0%**
Task 2: Preference Score ↑	29.4%	**70.6%**	31.9%	**68.1%**

Rombach et. al, High-Resolution Image Synthesis with Latent Diffusion Models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).

Limitations

- Sequential sampling still slower than GANs
- Not as suitable for high-precision tasks
  - For instance, super-resolution

Guess who?



Sample 4