DENOISING DIFFUSION IMPLICIT MODELS

Jiaming Song, Chenlin Meng & Stefano Ermon
Stanford University
{tsong,chenlin,ermon}@cs.stanford.edu

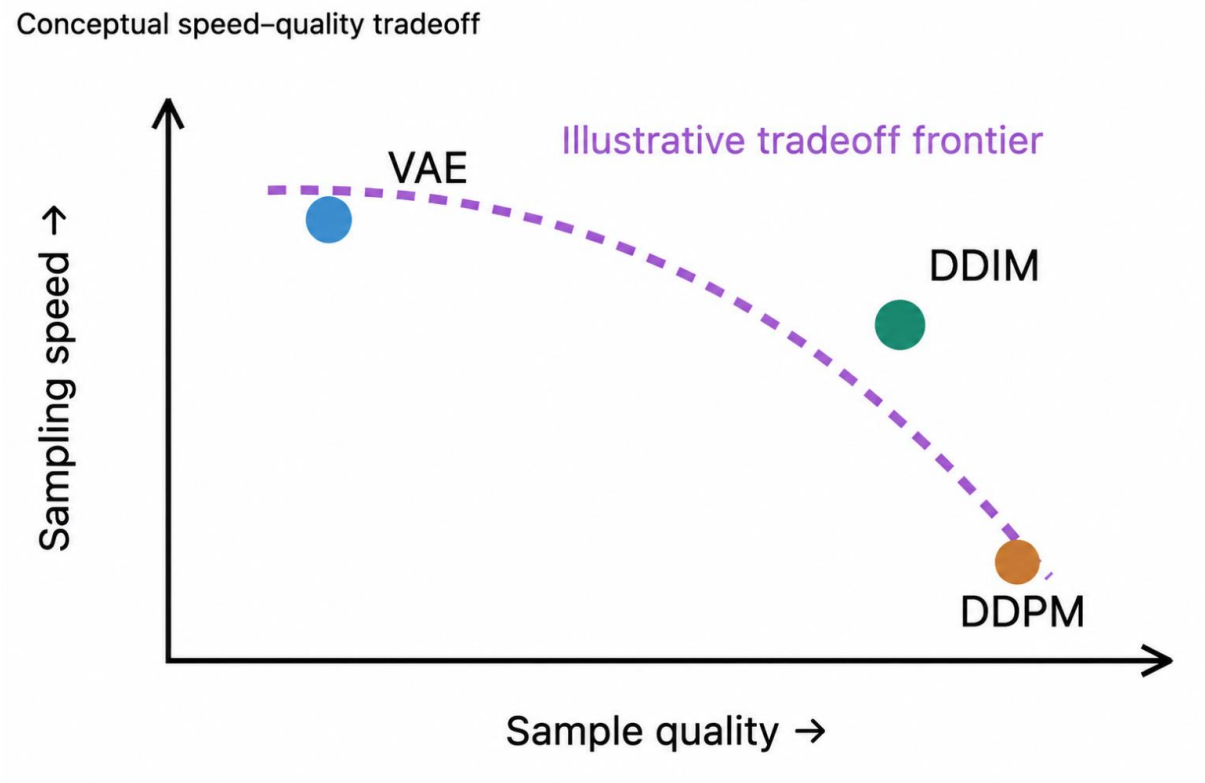
ICLR 2021

Motivation: High Quality, High Cost

Generative models face a tradeoff between sample quality and sampling speed.

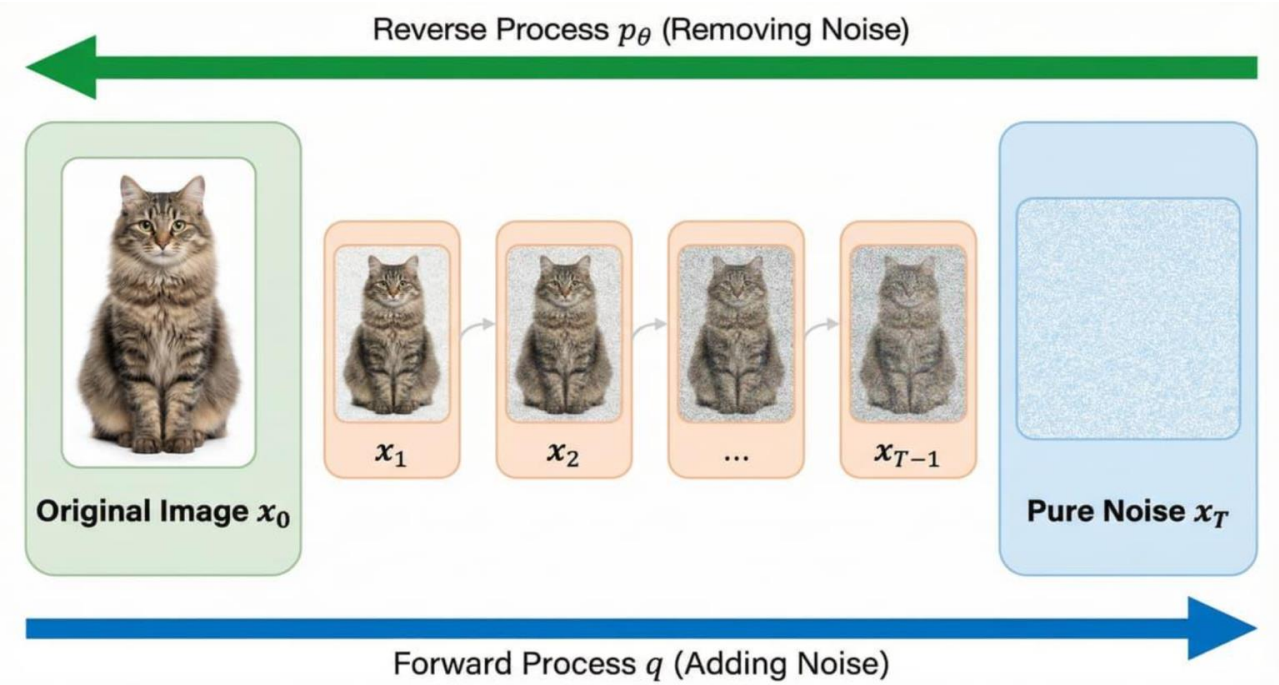
- VAEs: Fast sampling, but historically lower image quality than leading diffusion models.
- DDPMs: High-quality samples, but slow sequential generation.
- Traditional DDPM sampling typically requires 1,000 denoising steps.
- DDIM aims to improve the speed- quality tradeoff without retraining the model.

Motivation: High Quality, High Cost



How does a diffusion model generate an image?

A diffusion model learns to reverse a gradual noising process.



Background: Denoising Diffusion Probabilistic Models

Goal: Learn a generative distribution $p_\theta(x_0)$ that approximates the real data distribution $q(x_0)$

DDPM as a latent variable model (Equation 1)

$$p_\theta(x_0) = \int p_\theta(x_{0:T})\, dx_{1:T}$$

where

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

Training objective: Variational lower bound (Equation 2)

$$\max_\theta \mathbb{E}_{q(x_0)}[\log p_\theta(x_0)] \geq \max_\theta \mathbb{E}_{q(x_{0:T})}[\log p_\theta(x_{0:T}) - \log q(x_{1:T} \mid x_0)]$$

- x_0 : Original image.
- $x_1, \cdots, x_T$: Latent variables representing progressively noisier images.
- $q(x_{1:T} \mid x_0)$: Fixed inference process.
- $p_\theta(x_{0:T})$: Learned reverse generative process.

DDPM Forward Process: Adding Gaussian Noise

Markovian forward process (Equation 3)

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

Direct sampling at timestep t (Equation 4)

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\epsilon$$

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

DDPM Training Objective and Sampling Limitation

Noise prediction objective (Equation 5)

$$L_\gamma(\epsilon_\theta) = \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\substack{x_0 \sim q(x_0) \\ \epsilon_t \sim \mathcal{N}(0,I)}} \left[\left\| \epsilon_\theta^{(t)}(\sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon_t) - \epsilon_t \right\|_2^2 \right]$$

Why is DDPM sampling slow?

$$x_T \rightarrow x_{T-1} \rightarrow \cdots \rightarrow x_0$$

- DDPM commonly uses T=1000.
- Each reverse transition requires a sequential denoising operation.
- A large T makes Gaussian reverse-transition approximations more accurate, but increases generation time.

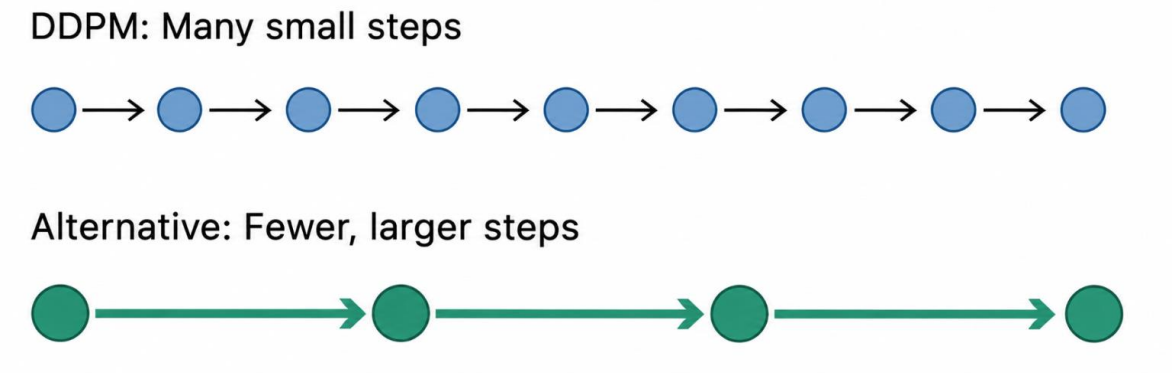
Can we preserve the same noise-prediction training objective while changing the generative process to require fewer sampling steps?

Can We Change the Sampling Process?

The problem with DDPM

- DDPM generates images through many small, sequential denoising steps.
- Conventional sampling also adds fresh noise during the reverse process.
- Using fewer steps with the original sampler can significantly reduce image quality.

What if we could take a different path?



Non-Markovian Forward Processes

The DDPM training objective depends on the marginals $q(x_t \mid x_0)$, not directly on the joint distribution $q(x_{1:T} \mid x_0)$

Non-Markovian inference process (Equation 6):

$$q_\sigma(x_{1:T} \mid x_0) = q_\sigma(x_T \mid x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1} \mid x_t, x_0)$$

Unlike DDPM, the forward process is non-Markovian: transitions can depend on both x_{t-1} and x_0

σ_t controls the stochasticity of the process.

Different choices of σ produce different joint inference distributions.

All these processes preserve the same marginals :

$$q_\sigma(x_t \mid x_0) = \mathcal{N}(\sqrt{\alpha_t} x_0, (1 - \alpha_t) I)$$

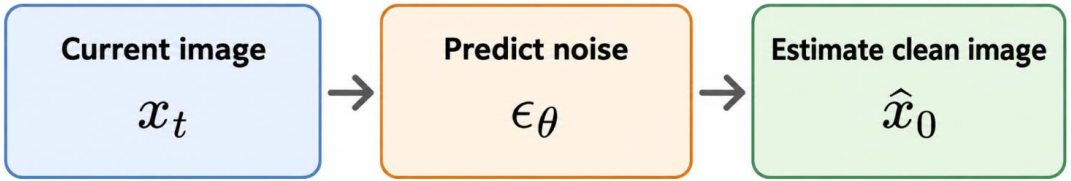
Generative Process: Predicting the Clean Image

Given x_t, predict the original image x_0, then use that prediction to generate x_{t-1}

Predict x_0 using the noise prediction network (Equation 9)

$$f_\theta^{(t)}(x_t) = \frac{x_t - \sqrt{1-\alpha_t}\,\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}}$$

- $\epsilon_\theta^{(t)}(x_t)$: Predicted noise.
- $f_\theta^{(t)}(x_t)$: Predicted clean image.



Unified Variational Objective: Why Retraining Is Unnecessary

Variational inference objective (Equation 11)

$$J_\sigma(\epsilon_\theta) = \mathbb{E}_{x_{0:T} \sim q_\sigma} \left[\log q_\sigma(x_{1:T} \mid x_0) - \log p_\theta(x_{0:T}) \right]$$

Theorem 1

For every strictly positive sigma, there exist positive weights gamma and a constant C such that:

$$J_\sigma(\epsilon_\theta) = L_\gamma(\epsilon_\theta) + C$$

Recall the DDPM training objective from Equation 5:

$$L_\gamma(\epsilon_\theta) = \sum_{t=1}^{T} \gamma_t \, \mathbb{E} \left[\left\| \epsilon_\theta^{(t)}(x_t) - \epsilon_t \right\|_2^2 \right]$$

Denoising Diffusion Implicit Models (DDIM)

Use the same trained DDPM model but change the reverse sampling process by controlling σ_t

Generalized sampling update (Equation 12)

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

where $\epsilon_t \sim \mathcal{N}(0, I)$ and $\alpha_0 = 1$

Two special cases:

DDPM-style stochastic sampling $\sigma_t = \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\sqrt{1-\frac{\alpha_t}{\alpha_{t-1}}}$

Deterministic DDIM sampling $\sigma_t = 0$

Accelerated Generation: Sampling with Fewer Steps

Instead of generating through all T timesteps, sample using a shorter subsequence.

Original DDPM sampling $x_T \to x_{T-1} \to \cdots \to x_1 \to x_0$

Accelerated sampling, choose a subsequence:

$$\tau = [\tau_1, \tau_2, \ldots, \tau_S], \qquad S \ll T$$

Then sample:

$$x_{\tau_S} \to x_{\tau_{S-1}} \to \cdots \to x_{\tau_1} \to x_0$$

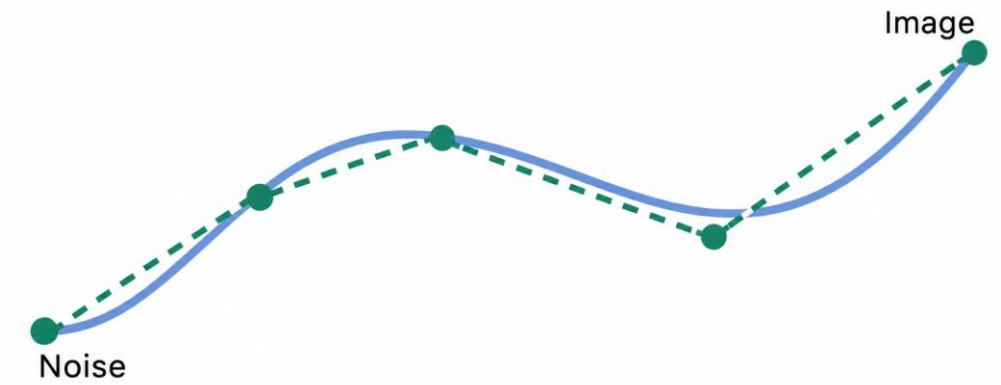
Preserving the marginals $q(x_{\tau_i} \mid x_0) = \mathcal{N}\left(\sqrt{\alpha_{\tau_i}} x_0, (1 - \alpha_{\tau_i}) I\right)$

Each reverse step requires a neural network evaluation. Reducing the number of steps reduces the number of sequential evaluations.

Connection to Neural Ordinary Differential Equations

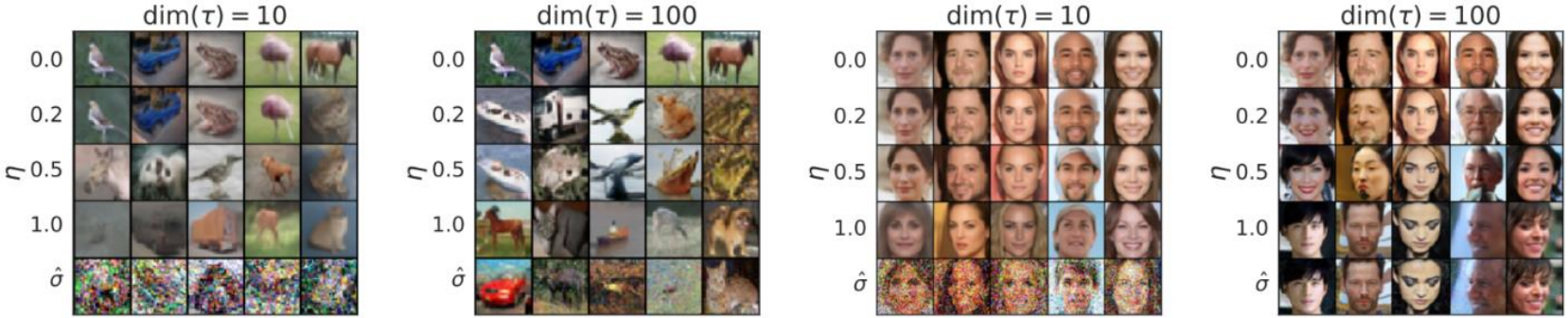
Deterministic DDIM sampling resembles numerical integration of an ODE.

- DDIM's deterministic update resembles Euler integration.
- The model defines a trajectory from noise to an image.
- Different step counts approximate the same continuous trajectory.
- The trajectory can also be followed in reverse for approximate image encoding.

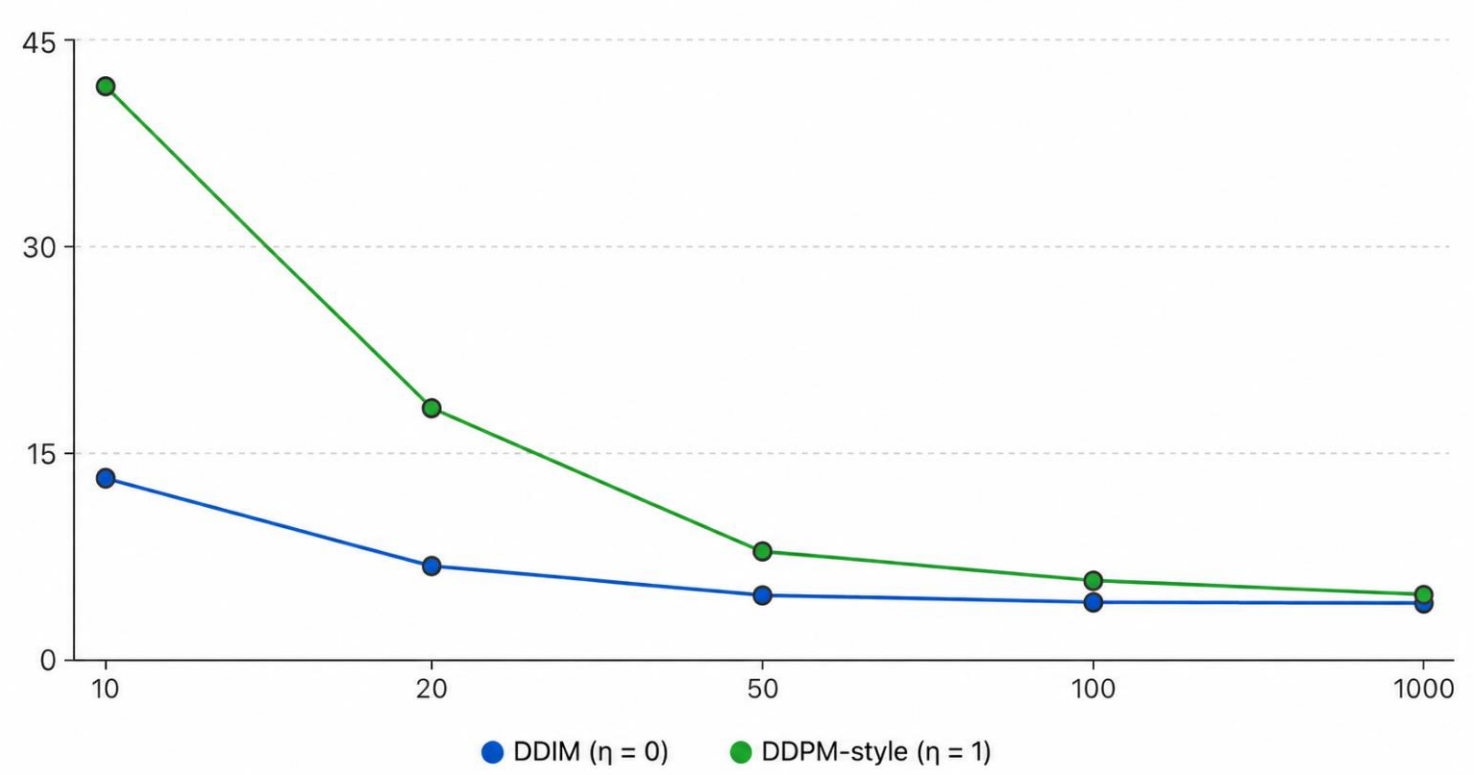


Experimental Setup : SAMPLE QUALITY AND EFFICIENCY

		CIFAR10 (32 × 32)					CelebA (64 × 64)				
	S	10	20	50	100	1000	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**

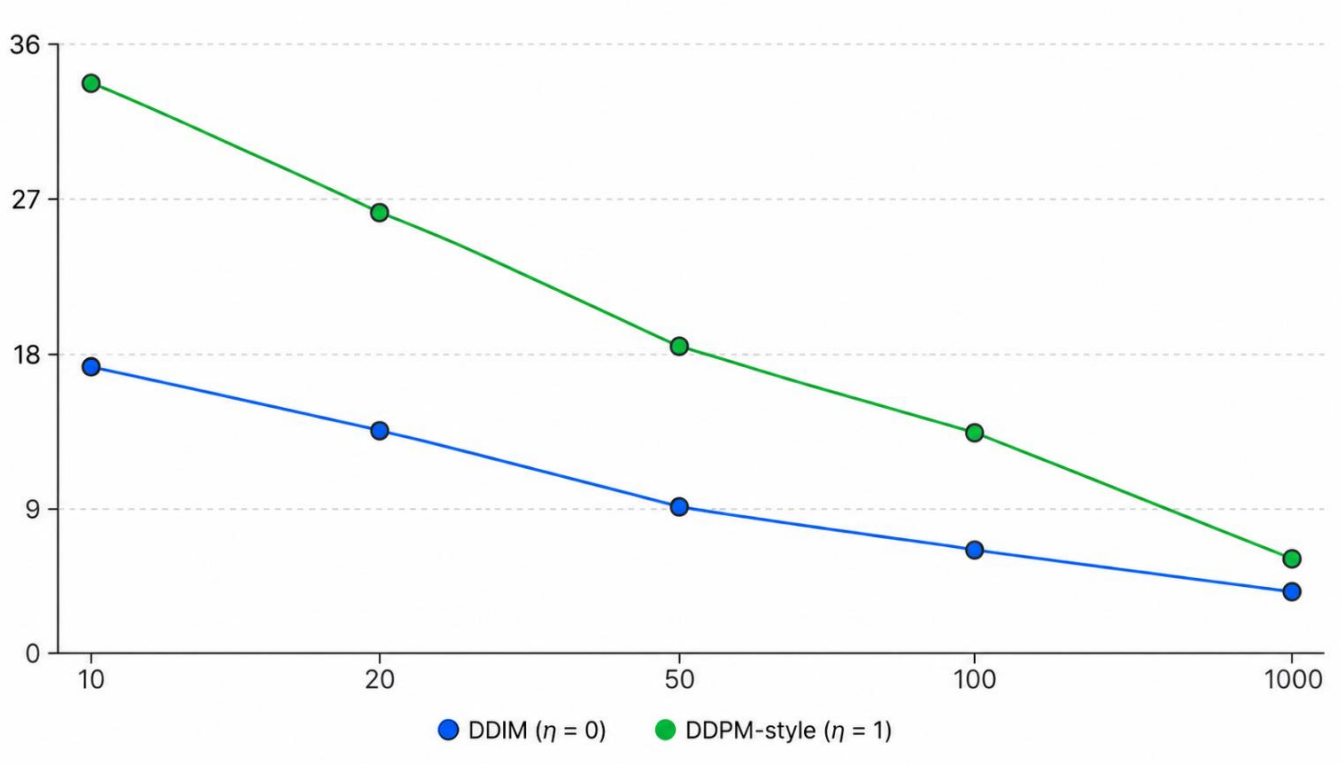


CIFAR-10 FID vs. sampling steps

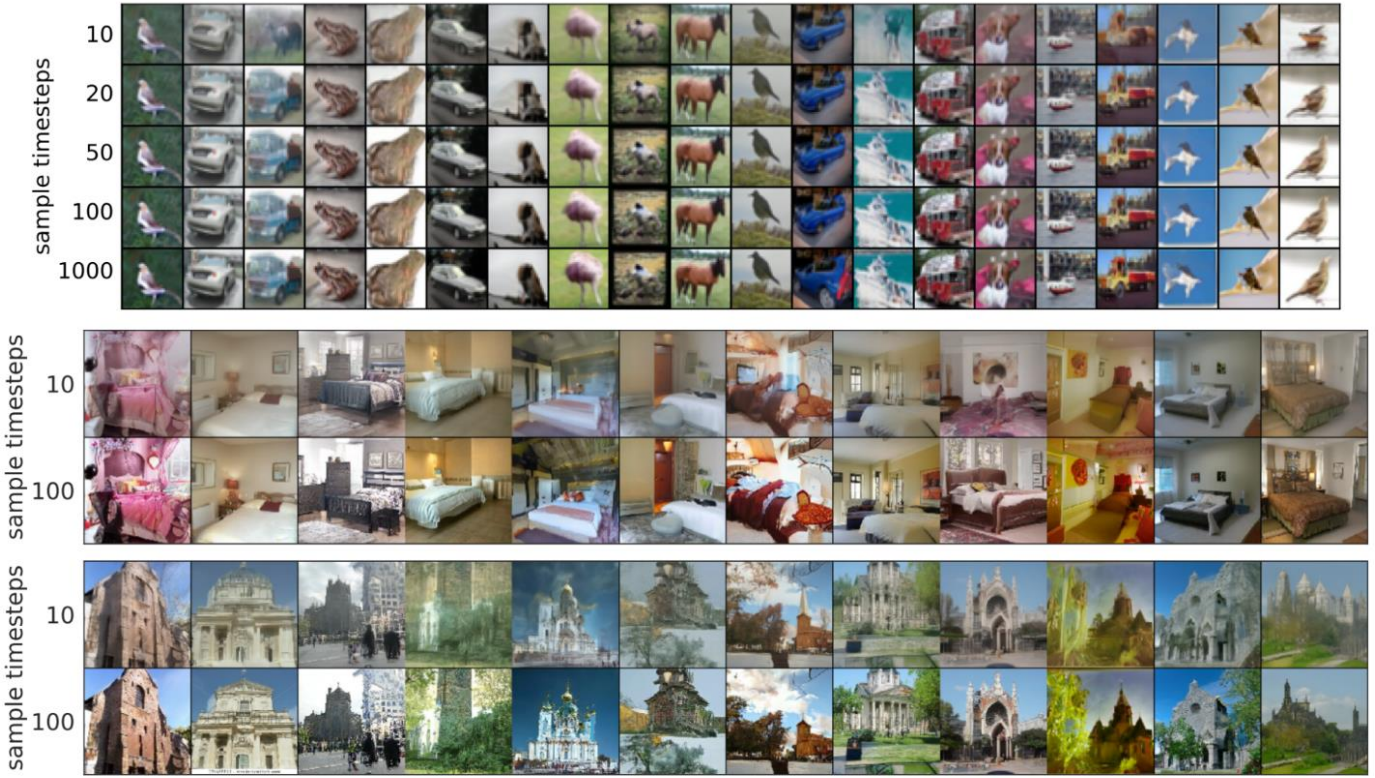


Lower FID is better. All methods use the same trained model.

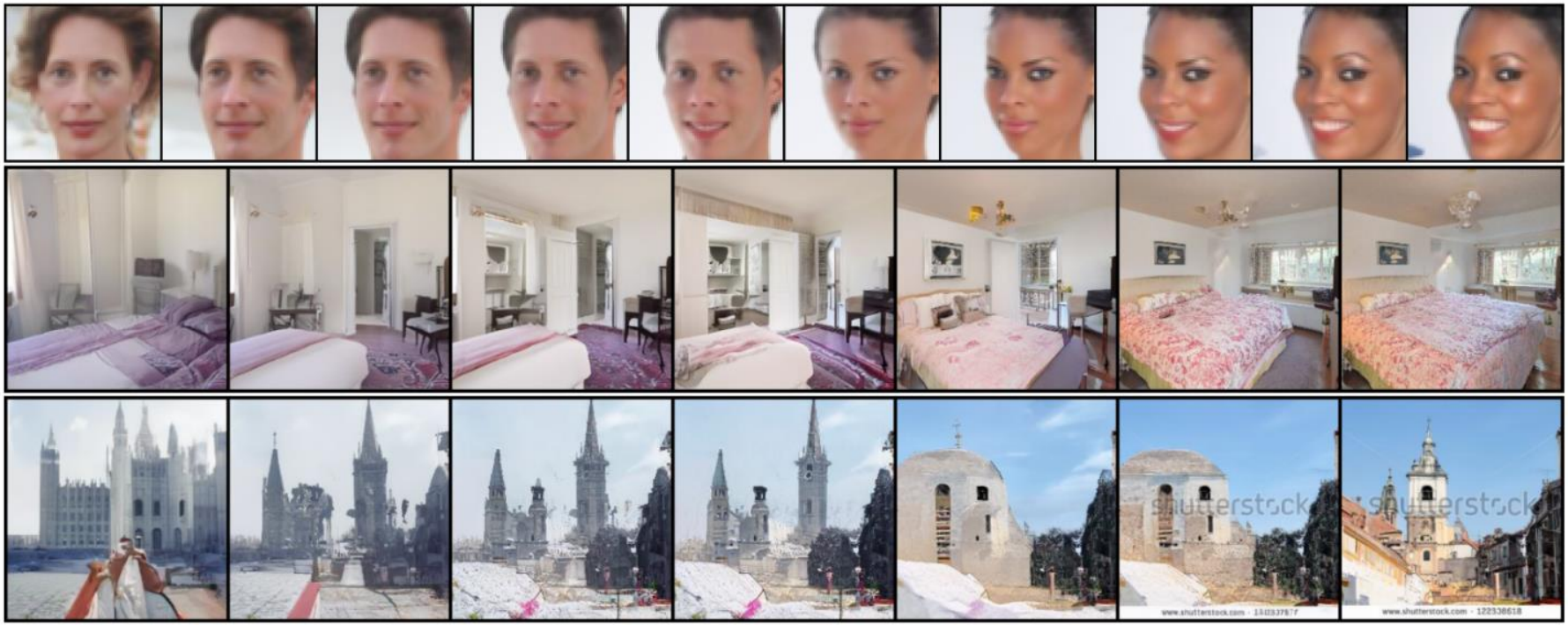
CelebA FID vs. sampling steps



SAMPLE CONSISTENCY IN DDIMS



INTERPOLATION IN DETERMINISTIC GENERATIVE PROCESSES



Strengths and Limitations

Strengths

- Reuses existing DDPM models without retraining.
- Strong improvements in low-step image quality.
- Flexible control over step count and stochasticity.
- Enables meaningful interpolation and approximate inversion.

Limitations

- Very few sampling steps can still degrade quality.
- Deterministic DDIM is not always best at 1,000 steps.
- Experiments focus on a limited set of image datasets and model architectures.

Questions

- Why does removing all randomness (eta = 0) give better quality at few steps? Is there a clear intuition for that? Also, how should we choose which subset of timesteps to use, and does that choice matter as much as the number of steps?

- Is DDIM in fact a kind of flow model? Since it is also deterministic.

- Why does DDIM perform better with fewer steps but not always with 1,000 steps?

- From my understanding, few-step diffusion models do not necessarily need DDIM. We can skip from timestep t to an earlier timestep t-k and derive q(x_t-k| x_t,x_0) by Bayes' rule, which is how Stable Diffusion implements. Then what might be the advantages of DDIM?