Diffusion Models Beat GANs on Image Synthesis

Authors: **Prafulla Dhariwal and Alex Nichol**
Presenter: Xiaoxiao Ma

22 March 2024

Introduction

- Generative models have gained ability to generate human-like language, high-quality images, etc..
- There still much room for improvement and better generative models have wide-ranging impacts on various applications.

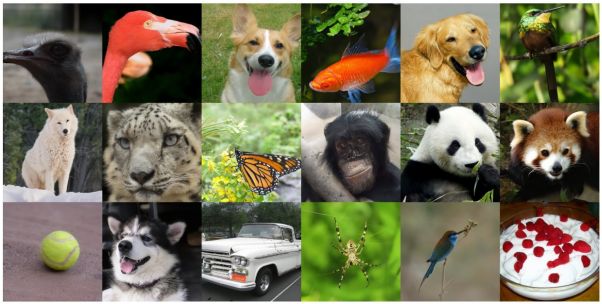


Figure 1: Selected samples from our best ImageNet 512×512 model (FID 3.85)

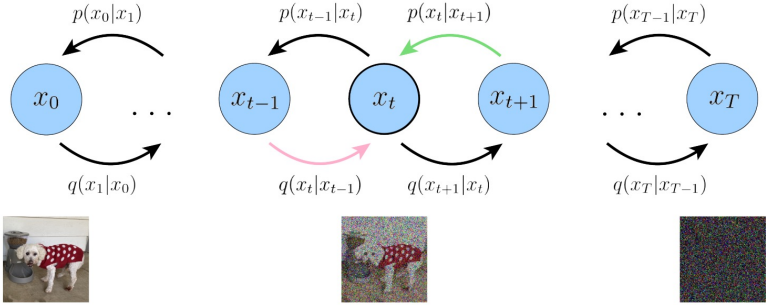
GANs

GANs[1] and likelihood-based models hold the state-of-the-art on image generation tasks before the burst of diffusion models.

- GANs
  - GANs capture less diversity
  - Difficult to train, collapsing without carefully selected hyperparameters and regularizers
  - Difficult to scale and apply to new domains
  - Fast sampling speed
- Conventional Likelihood-based models
  - Capture more diversity
  - Easier to scale and train than GANs
  - Still fall short in terms of visual sample quality
  - Sampling from these models is slower than GANs (except VAE)

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

Diffusion Models



Diffusion models are a class of likelihood-based models.
These models generate samples by gradually removing noise from a signal.
But still lags behind BigGAN-deep[2] on difficult generation datasets like LSUN and ImageNet.

[2] Brock, A., Donahue, J., & Simonyan, K. (2018, September). Large Scale GAN Training for High Fidelity Natural Image Synthesis. In International Conference on Learning Representations.

Factors Behind the Gap

- F1. the model architectures used by recent GAN literature have been heavily explored and refined
- F2. GANs are able to trade off diversity for fidelity, producing high quality samples but not covering the whole distribution

The authors bring these benefits to diffusion models by:

- A1. improving model architecture
- A2. devising a scheme for trading off diversity for fidelity

DDPM Training Objectives

The variational lower bound (VLB) of DDPM model can be written as:

$$L_{vlb} = L_0 + L_1 + \cdots + L_{T-1} + L_T \tag{1}$$
$$L_0 = -logp_\theta(x_0|x_1) \tag{2}$$
$$L_{t-1} = D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) \tag{3}$$
$$L_T = D_{KL}(q(x_T|x_0)||p(x_T)) \tag{4}$$
$$\tag{5}$$

L_0 is reconstruction term that can be approximated and optimized using a Monte Carlo estimate.
L_T is independent of θ, which can be ignored.
L_{t-1} is the denoising matching term and it is minimized when two denoising steps match as closely as possible.

DDPM Training Objectives

Ho et al. (2020) simplifies the training objective as to predict the noise, ϵ, been added at arbitrary step t.

$$L_{simple} = \mathbb{E}_{t,x_0,\epsilon}[||\epsilon - \epsilon_\theta(x_t, t)||^2] \tag{6}$$

DDPM Training Objectives

Ho et al. (2020) simplifies the training objective as to predict the noise, ϵ, been added at arbitrary step t by setting the variance as a constant.

$$
\begin{aligned}
&\arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \\
&= \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}(\mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\left(t\right)\right) \parallel \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}, \boldsymbol{\Sigma}_q\left(t\right)\right)) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \left[\left\| \frac{1}{\sqrt{\alpha_t}} \boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) - \frac{1}{\sqrt{\alpha_t}} \boldsymbol{x}_t + \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \boldsymbol{\epsilon}_0 \right\|_2^2 \right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \left[\left\| \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \boldsymbol{\epsilon}_0 - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \right\|_2^2 \right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \left[\left\| \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} (\boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)) \right\|_2^2 \right] \\
&= \arg\min_{\boldsymbol{\theta}} \frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t} \left[\left\| \boldsymbol{\epsilon}_0 - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \right\|_2^2 \right]
\end{aligned}
$$

Previous Improvements on DDPM

Nichol and Dhariwal[3] find that fixing the variance $\Sigma_\theta(x_t, t)$ to a constant as done in Ho et al. (2020) is sub-optimal for sampling with fewer diffusion steps, and reformat the training objective to learn the variance following[4]:

$$L_{hybrid} = L_{simple} + \lambda L_{vlb} \tag{7}$$

[3] Nichol, A. Q., & Dhariwal, P. (2021, July). Improved denoising diffusion probabilistic models. In International conference on machine learning (pp. 8162-8171). PMLR.

[4] the authors set $\lambda = 0.001$ to prevent L_{vlb} from overwhelming L_simple.

Architecture Improvements

The authors explore the following changes to the UNet architecture:

- Increasing depth versus width, holding model size relatively constant
- Increasing the number of attention heads
- Using attention at 32×32, 16×16, and 8×8 resolutions rather than only at 16×16
- Using the BigGAN residual block for upsampling and downsampling the activations
- Rescaling residual connections with $\frac{1}{\sqrt{2}}$

Empirical Results

Channels	Depth	Heads	Attention resolutions	BigGAN up/downsample	Rescale resblock	FID 700K	FID 1200K
160	2	1	16	✗	✗	15.33	13.21
128	4					-0.21	-0.48
		4				-0.54	-0.82
			32,16,8			-0.72	-0.66
				✓		-1.20	-1.21
					✓	0.16	0.25
160	2	4	32,16,8	✓	✗	**-3.14**	**-3.00**

Table 1: Ablation of various architecture changes, evaluated at 700K and 1200K iterations

Aside from rescaling residual connections, all of the other modifications improve performance and have a positive compounding effect.

Empirical Results

Number of heads	Channels per head	FID
1		14.08
2		-0.50
4		-0.97
8		-1.17
	32	-1.36
	64	-1.03
	128	-1.08

Table 2: Ablation of various attention configurations. More heads or lower channels per heads both lead to improved FID.

More heads or fewer channels per head improves FID.

Empirical Results - Adaptive Group Normalization

Operation	FID
AdaGN	13.06
Addition + GroupNorm	15.08

Table 3: Ablating the element-wise operation used when projecting timestep and class embeddings into each residual block. Replacing AdaGN with the Addition + GroupNorm layer from Ho et al. [25] makes FID worse.

$$time_{emb} = AdaGN(h, y) = SiLU(Linear(time_{emb} + label_{emb}))$$
$$time_{emb} = GaroupNorm(h) + y = SiLU(Linear(time_{emb})) + label_{emb}$$

Classifier Guidance

In addition to the architectural improvement, the authors also explore different ways to condition diffusion models on class labels.
Specifically, they exploit a classifier $p(y|x)$ to improve a diffusion generator.

Conditional Reverse Noising Process

The class-conditional reverse transition can be formulated as:

$$p_{\theta,\phi}(x_t|x_{t+1}, y) = Zp_\theta(x_t|x_{t+1})p_\phi(y|x_t) \tag{8}$$

Here, Z is a constant independent of x_t.

Conditional Reverse Noising Process

$$
\begin{aligned}
\hat{q}(x_t|x_{t+1}, y) &= \frac{\hat{q}(x_t, x_{t+1}, y)}{\hat{q}(x_{t+1}, y)} \\
&= \frac{\hat{q}(x_t, x_{t+1}, y)}{\hat{q}(y|x_{t+1})\hat{q}(x_{t+1})} \\
&= \frac{\hat{q}(x_t|x_{t+1})\hat{q}(y|x_t, x_{t+1})\hat{q}(x_{t+1})}{\hat{q}(y|x_{t+1})\hat{q}(x_{t+1})} \\
&= \frac{\hat{q}(x_t|x_{t+1})\hat{q}(y|x_t, x_{t+1})}{\hat{q}(y|x_{t+1})} \\
&= \frac{\hat{q}(x_t|x_{t+1})\hat{q}(y|x_t)}{\hat{q}(y|x_{t+1})} \\
&= \frac{q(x_t|x_{t+1})\hat{q}(y|x_t)}{\hat{q}(y|x_{t+1})}
\end{aligned}
$$

Here $\hat{q}(y|x_{t+1}) = \frac{1}{Z}$ is independent of x_t.

Conditional Reverse Noising Process

We can approximate $log(p_\phi(y|x_t))$ using a Taylor expansion around $x_t = \mu$ as:

$$log(p_\phi(y|x_t)) \approx log(p_\phi(y|x_t))|_{x_t=\mu} + (x_t - \mu)\nabla_{x_t} log(p_\phi(y|x_t))|_{x_t=\mu} \quad (9)$$

$$= (x_t - \mu)g + C_1, \quad (10)$$

here $g = \nabla_{x_t} log(p_\phi(y|x_t))|_{x_t=\mu}$

Conditional Reverse Noising Process

Taylor expansion of $f(x)$ around $x = a$

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n$$

Conditional Reverse Noising Process

Recall that the denoising process predicts x_t from x_{t+1} following:

$$p_\theta(x_t|x_{t+1}) = \mathcal{N}(\mu, \Sigma) \tag{11}$$

$$log(p_\theta(x_t|x_{t+1})) = -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) + C \tag{12}$$

We can then write $p_{\theta,\phi}(x_t|x_{t+1}, y)$ as:

$$logp_{\theta,\phi}(x_t|x_{t+1}, y) \approx -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) + (x_t-\mu)g + C_2 \tag{13}$$

$$= -\frac{1}{2}(x-\mu-\Sigma g)^T\Sigma^{-1}(x-\mu-\Sigma g) + \frac{1}{2}g^T\Sigma g + C_2 \tag{14}$$

$$= -\frac{1}{2}(x-\mu-\Sigma g)^T\Sigma^{-1}(x-\mu-\Sigma g) + C_3 \tag{15}$$

$$= logp(z) + C_4, \text{with} \quad z \sim \mathcal{N}(\mu+\Sigma g, \Sigma) \tag{16}$$

Conditional Reverse Noising Process

Recall that

$$p_{\theta,\phi}(x_t|x_{t+1}, y) = Z p_\theta(x_t|x_{t+1}) p_\phi(y|x_t), \tag{17}$$

then C_4 in Eq.(16) corresponds to the coefficient Z and can be safely ignored.

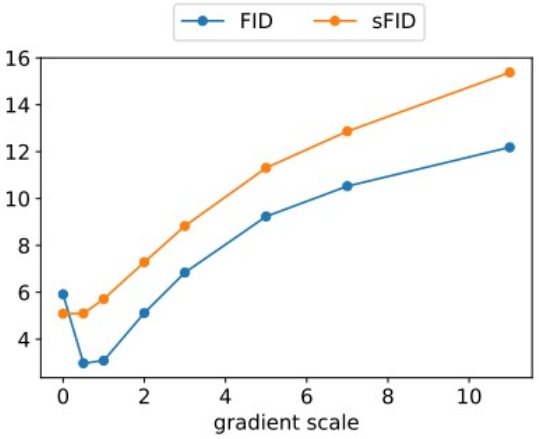
Classifier Guided Diffusion Sampling

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

In the sampling process, we can see a factor s is introduced to scale the effect of classifier gradients. The sampled x_t with classifier guidance can be treated as shifting the mean of unconditional sampling by $s\Sigma g$.
A larger s will stress more on the classifier signal, which is potentially desirable for producing high fidelity (but less diverse) samples.

Impact of the Scaling Factor



Something to Take Away

Section 7 Limitations and Future Work is very valuable and pinpoints the potential of image generator with text caption.

Thanks very much for your attention.