Classifier-Free Diffusion Guidance

Ho et al., 2021

Why Classifier-‘free’ Guidance? (vs. Classifier Guidance)

- How the guidance works?
  - A pure generative model (vs. a pre-trained classifier needed)
  - Training conditional and unconditional models jointly (vs. 無)
  - Sampling using a linear combination of the conditional and unconditional models (vs. adding the gradient of the pre-trained classifier to the diffusion model)
- Pros and Cons
  - Simplified data pipeline (vs. complicated)
  - Extremely simple implementation
  - Detour the ‘adversarial attack’
  - X Slower sampling speed

Motivation: Truncation

- Truncation Trick
  - Taking a model trained with z ~ N(0, I)
  - Truncating a z vector by resampling the values with magnitude above a chosen threshold
  - Leading to improvement in individual sample quality at the cost of reduction in overall sample variety

(a) (b)

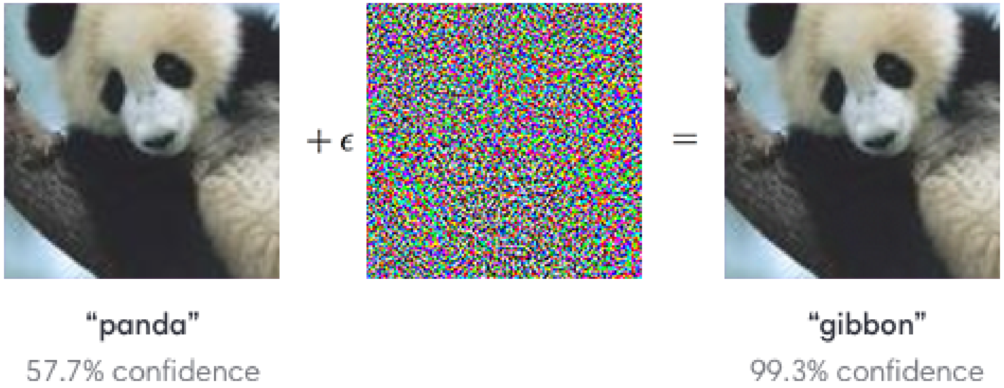
Figure 2: (a) The effects of increasing truncation. From left to right, the threshold is set to 2, 1, 0.5, 0.04. (b) Saturation artifacts from applying truncation to a poorly conditioned model.

Motivation: Classifier Guidance

- An adversarial attack

Recall…

  - $\hat{\epsilon}(x_t) := \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\,\nabla_{x_t} \log p_\phi(y|x_t)$
  - $p_\phi(y|x_t)$: The classifier is pre-trained with noise-added images.
  - Classifier-based evaluations metrics: IS (Inception Score), FID (Frechet Inception Score)



Nomenclature

Ho et al., 2021

$$q(\mathbf{z}_\lambda|\mathbf{x}) = \mathcal{N}(\alpha_\lambda \mathbf{x}, \sigma_\lambda^2 \mathbf{I}), \text{ where } \alpha_\lambda^2 = 1/(1+e^{-\lambda}),\ \sigma_\lambda^2 = 1-\alpha_\lambda^2$$

$$\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$$

Ho et al., 2020

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$\mathbf{x}_t(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$$

DDPM	**Classifier-free Guidance**
$\mathbf{x_0}$	$\mathbf{x}$
$\mathbf{x_t}$	$\boldsymbol{z_\lambda}$
$\sqrt{\overline{\alpha_t}}$	α_λ
$1-\overline{\alpha_t}$	σ_λ^2

Classifier Guidance

Recall…

DDPM $\quad \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) \approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda | \mathbf{c})$

Classifier Guidance

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c} | \mathbf{z}_\lambda)$$
$$\approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} [\log p(\mathbf{z}_\lambda | \mathbf{c}) + w \log p_\theta(\mathbf{c} | \mathbf{z}_\lambda)]$$

Song et al., 2020 $\quad p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{1-\beta_t}}(x_t + \beta_t\, s_\theta(x_t, t)), \beta_t I)$

Ho et al., 2020 $\quad x_{t-1} \sim p_\theta(x_{t-1} | x_t)$ where $\mu_\theta = \frac{1}{\sqrt{\alpha}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t))$

$$\nabla_{x_t} \log p_\theta(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t)$$

Dhariwal et al., 2021

$$\nabla_{x_t} \log(p_\theta(x_t) p_\phi(y | x_t)) = \nabla_{x_t} \log p_\theta(x_t) + \nabla_{x_t} \log p_\phi(y | x_t)$$
$$= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t) + \nabla_{x_t} \log p_\phi(y | x_t)$$
$$\hat{\epsilon}(x_t) := \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log p_\phi(y | x_t)$$

Classifier-free Guidance

Recall…

DDPM $\quad \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) \approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda | \mathbf{c})$

Classifier Guidance
$$\begin{aligned}\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) &= \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c} | \mathbf{z}_\lambda) \\ &\approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} [\log p(\mathbf{z}_\lambda | \mathbf{c}) + w \log p_\theta(\mathbf{c} | \mathbf{z}_\lambda)]\end{aligned}$$

Training
$$\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c})$$
$$\epsilon_\theta(\mathbf{z}_\lambda) = \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c} = \mathbf{0})$$

Sampling (Inference)
$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\epsilon_\theta(\mathbf{z}_\lambda)$$

Classifier-free Guidance: an Intuition

Recall…

DDPM $\quad \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) \approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda | \mathbf{c})$

Classifier Guidance

$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c} | \mathbf{z}_\lambda)$$
$$\approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} [\log p(\mathbf{z}_\lambda | \mathbf{c}) + w \log p_\theta(\mathbf{c} | \mathbf{z}_\lambda)]$$

Training

$$\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c})$$
$$\epsilon_\theta(\mathbf{z}_\lambda) = \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c} = \mathbf{0})$$

Sampling (Inference)

$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\epsilon_\theta(\mathbf{z}_\lambda)$$

Classifier-free Guidance: an Implicit Classifier

Recall…

DDPM $\qquad \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) \approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda | \mathbf{c})$

Classifier Guidance

$$\begin{aligned}\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) &= \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda) \\ &\approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda}[\log p(\mathbf{z}_\lambda|\mathbf{c}) + w \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)]\end{aligned}$$

Training

$$\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c})$$

$$\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) = \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c} = \mathbf{0})$$

Sampling (Inference)

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1 + w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$$

Implicit classifier

$$p^i(\mathbf{c}|\mathbf{z}_\lambda) \propto p(\mathbf{z}_\lambda|\mathbf{c})/p(\mathbf{z}_\lambda)$$

$$\nabla_{\mathbf{z}_\lambda} \log p^i(\mathbf{c}|\mathbf{z}_\lambda) = -\frac{1}{\sigma_\lambda}[\boldsymbol{\epsilon}^*(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon}^*(\mathbf{z}_\lambda)]$$

Classifier-free Guidance: an Implicit Classifier (cont.)

Classifier Guidance

$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)$$
$$\approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda}[\log p(\mathbf{z}_\lambda|\mathbf{c}) + w \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)]$$

Sampling (Inference)

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$$

Implicit classifier

$$p^i(\mathbf{c}|\mathbf{z}_\lambda) \propto p(\mathbf{z}_\lambda|\mathbf{c})/p(\mathbf{z}_\lambda)$$

$$\nabla_{\mathbf{z}_\lambda} \log p^i(\mathbf{c}|\mathbf{z}_\lambda) = -\frac{1}{\sigma_\lambda}[\boldsymbol{\epsilon}^*(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon}^*(\mathbf{z}_\lambda)]$$

$$\tilde{\epsilon}_\theta(z_\lambda, c) = \epsilon_\theta(z_\lambda, c) - w\sigma_\lambda\left(-\frac{1}{\sigma_\lambda}[\epsilon^*(z_\lambda, c) - \epsilon^*(z_\lambda)]\right)$$
$$= \epsilon_\theta(z_\lambda, c) + w\big(\epsilon^*(z_\lambda, c) - \epsilon^*(z_\lambda)\big)$$
$$= (1+w)\epsilon_\theta(z_\lambda, c) - w\epsilon_\theta(z_\lambda) \ (\because \epsilon_\theta \text{ estimates } \epsilon^*)$$

Experiments

- Dataset: 64x64 area-downsampled ImageNet
- The model on unconditional generation jointly trained with probability 0.1
- FID and Inception Scores calculated with 50000 samples for each value using T = 256 sampling steps.

Method	FID (↓)	IS (↑)
ADM [3]	2.07	-
CDM [6]	**1.48**	67.95
Ours, no guidance	1.80	53.71
Ours, with guidance		
$w = 0.1$	1.55	66.11
$w = 0.2$	2.04	78.91
$w = 0.3$	3.03	92.8
$w = 0.4$	4.30	106.2
$w = 0.5$	5.74	119.3
$w = 0.6$	7.19	131.1
$w = 0.7$	8.62	141.8
$w = 0.8$	10.08	151.6
$w = 0.9$	11.41	161
$w = 1.0$	12.6	170.1
$w = 2.0$	21.03	225.5
$w = 3.0$	24.83	250.4
$w = 4.0$	26.22	**260.2**

Figure 1: ImageNet 64x64 results

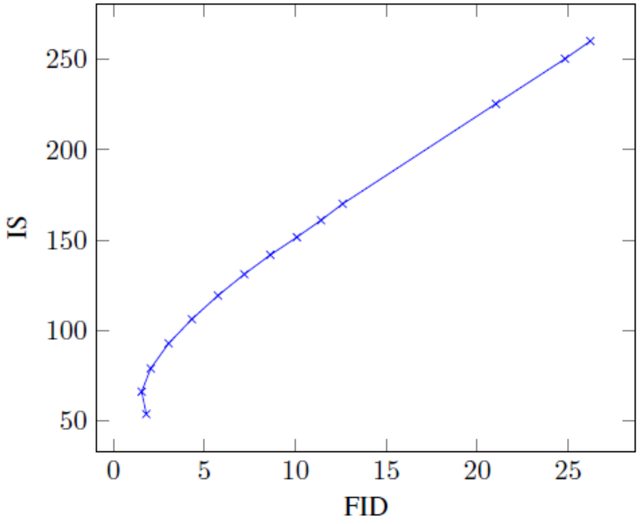


Figure 2: ImageNet 64x64 FID vs. IS

[3] Diffusion models beat GANs on image synthesis (Dhariwal et al., 2021)
[6] Cascaded diffusion models for high fidelity image generation (Ho et al., 2021)

Experiments (cont.)

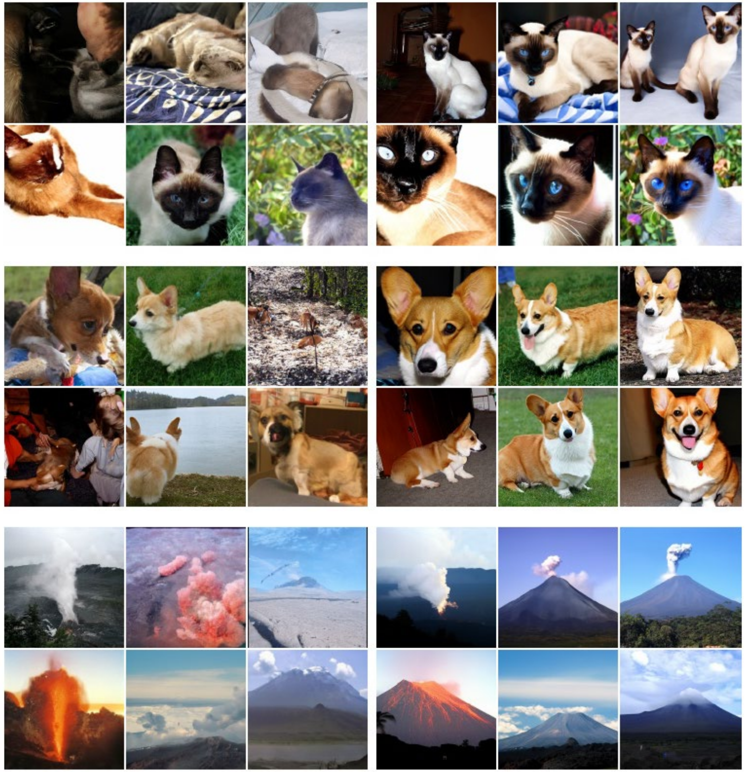
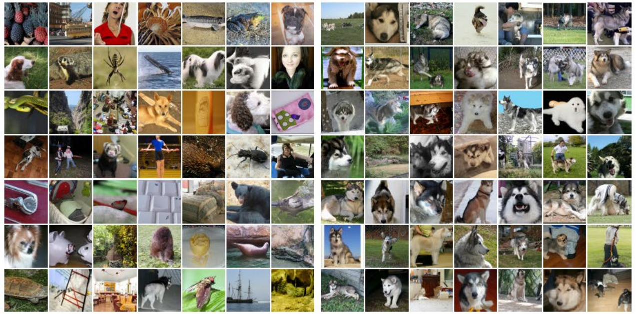
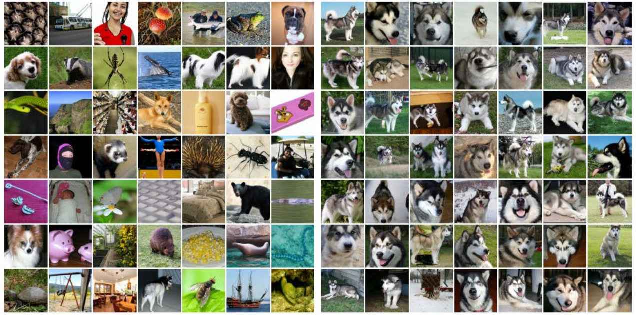


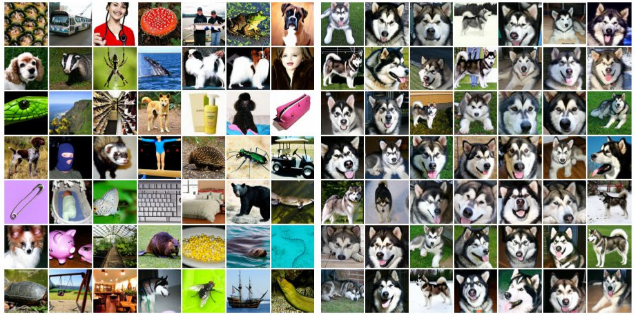
Figure 4: Classifier-free guidance on 128x128 ImageNet. Left: non-guided samples, right: guided samples with $w = 3.0$. Interestingly, strongly guided samples such as these display saturated colors.



(a) Non-guided conditional sampling: FID=1.80, IS=53.71



(b) Classifier-free guidance with $w = 1.0$: FID=12.6, IS=170.1



(c) Classifier-free guidance with $w = 3.0$: FID=24.83, IS=250.4

Figure 3: Classifier-free guidance on ImageNet 64x64. Left: random classes. Right: single class (malamute). Same random seeds used for sampling in each subfigure.

Why Classifier-‘free’ Guidance? (vs. Classifier Guidance)

- How the guidance works?
  - A pure generative model (vs. a pre-trained classifier needed)
  - Training conditional and unconditional models jointly (vs. 無)
  - Sampling using a linear combination of the conditional and unconditional models (vs. adding the gradient of the pre-trained classifier to the diffusion model)
- Pros and Cons
  - Simplified data pipeline (vs. complicated)
  - Detour the ‘adversarial attack’
  - Extremely simple implementation
  - X Slower sampling speed

Reference

- Classifier-Free Diffusion Guidance (Ho et al., 2021)
- Denoising Diffusion Probabilistic Models (Ho et al., 2020)
- Score-Based Generative Modeling through Stochastic Differential Equations (Song et al., 2021)
- Diffusion Models Beat GANs on Image Synthesis (Dhariwal et al., 2021)
- Large Scale GAN Training for High Fidelity Natural Image Synthesis (Brock et al., 2018)
- Explaining and Harnessing Adversarial Examples (Goodfellow et al., 2014)

Thanks ☺