Classifier-Free Diffusion Guidance

NeurIPS 2021 Workshop

Jonathan Ho, Tim Salimans

Presented by Beomsoon Park

Contents

Introduction

Introduction

❑ Fidelity vs Diversity

Diversity ↓
Sample Quality ↑

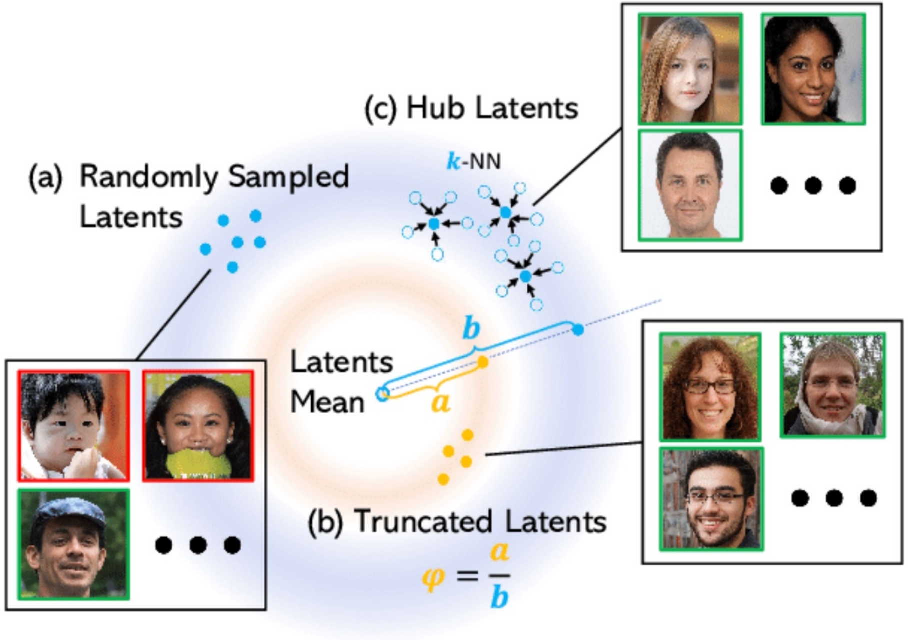
Trade-off

Diversity ↑
Sample Quality ↓

Introduction

- ❑ **Fidelity vs Diversity**

Truncation Trick (GAN)



Introduction

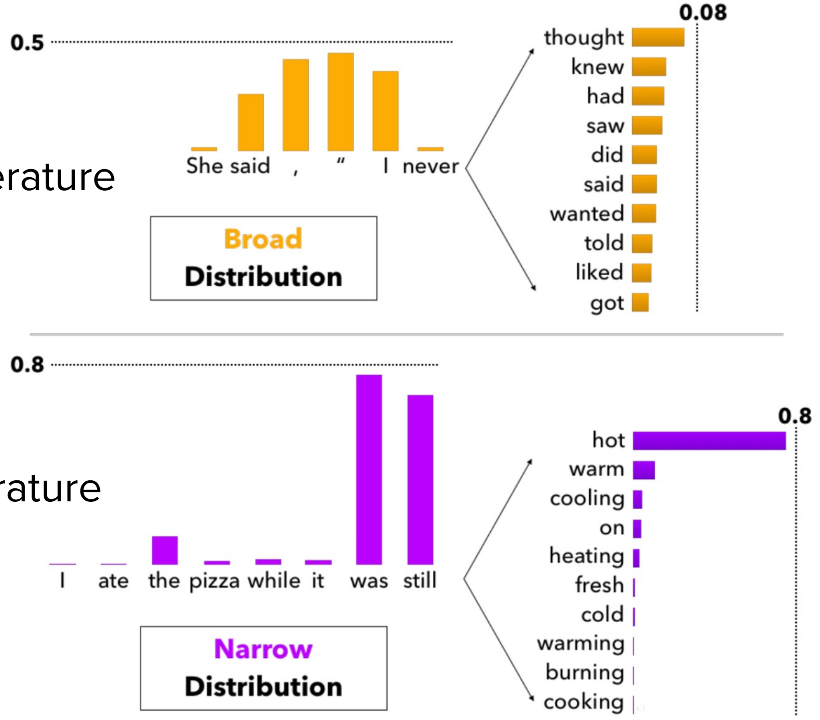
❑ **Fidelity vs Diversity**

Low Temperature Sampling

τ : Temperature ParameterX

High Temperature
$\tau > 1$

High Temperature
$\tau < 1$



https://towardsdatascience.com/how-to-sample-from-language-models-682bceb97277

Introduction

❑ Truncation-like effect on Diffusion

Truncation Trick
(GAN)

Low Temperature
Sampling

Not applicable directly to
the diffusion models

Classifier-guided
Diffusion Model

Introduction

❑ Drawbacks of Classifier Guidance

Overhead to train the extra classifier

Must be trained with noisy data at every training procedure

Complicate Diffusion Model Training Pipeline

Introduction

❑ Drawbacks of Classifier Guidance

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

Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, 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**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log p_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0

Sampling
→ Mixture of **score estimate**
& **classifier gradient**

Introduction

❑ Drawbacks of Classifier Guidance

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**
$\quad \mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
$\quad x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, 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**
$\quad \hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1 - \bar{\alpha}_t}\, \nabla_{x_t} \log p_\phi(y|x_t)$
$\quad x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0

Image classifier might be confused as **adversarial attack** on sampling

Whether this method is eligible for classifier-based metrics is **questionable**

Introduction

❑ **Classifier-Free Guidance (CFG) Diffusion**

Not much overhead for training extra classifier

Extreme Simplicity (Only a change of one line)

Similar FID/IS tradeoff to that of classifier guidance diffusion

Background

Background

❑ Notation

Forward Process

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

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

$$\mathbf{z} = \{\mathbf{z}_\lambda \,|\, \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$$

Noisy Data

$$\lambda_{\min} < \lambda_{\max} \in \mathbb{R}$$

Hyperparameter for variance scheduling

Background

❑ Notation

Forward Process

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

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

$\lambda = \log(\alpha_\lambda^2/\sigma_\lambda^2)$

: log SNR of z_λ

Formularization of forward process feature

Background

❑ Notation

Forward Process

$$\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}) = e^{\lambda-\lambda'}(\alpha_{\lambda'}/\alpha_\lambda)\mathbf{z}_\lambda + (1 - e^{\lambda-\lambda'})\alpha_{\lambda'}\mathbf{x}, \quad \tilde{\sigma}^2_{\lambda'|\lambda} = (1 - e^{\lambda-\lambda'})\sigma^2_{\lambda'}$$

$$q(\mathbf{z}_{\lambda'}|\mathbf{z}_\lambda, \mathbf{x}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}), \tilde{\sigma}^2_{\lambda'|\lambda}\mathbf{I})$$

Final Sample

$$p_\theta(\mathbf{z}_{\lambda_{\min}}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\boxed{p_\theta(\mathbf{z}_{\lambda'}|\mathbf{z}_\lambda) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}_\theta(\mathbf{z}_\lambda)), (\tilde{\sigma}^2_{\lambda'|\lambda})^{1-v}(\sigma^2_{\lambda|\lambda'})^v)}$$

Reverse Process

Background

❑ Notation

$$\boxed{p_\theta(\mathbf{z}_{\lambda'}|\mathbf{z}_\lambda) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}_\theta(\mathbf{z}_\lambda)), (\tilde{\sigma}^2_{\lambda'|\lambda})^{1-v}(\sigma^2_{\lambda|\lambda'})^v)}$$

DDPM

$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$ for $1 < t \leq T$

$$\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t) \begin{cases} \sigma_t^2 = \beta_t \\ \sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \end{cases}$$

Similar Results
(1st option is adopted for this experiment)

$$\underbrace{(\tilde{\sigma}^2_{\lambda'|\lambda})}_{\tilde{\beta}_t}{}^{1-v}\underbrace{(\sigma^2_{\lambda|\lambda'})}_{\beta_t}{}^{v}$$

: log-space linear interpolation

v : **constant hyperparameter** controlling posterior variance

Background

❑ Training Objective

$$\boxed{\mathbb{E}_{\boldsymbol{\epsilon},\lambda}\left[\left\|\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) - \boldsymbol{\epsilon}\right\|_2^2\right]}$$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$, $\lambda \sim p(\lambda)$ over $[\lambda_{min}, \lambda_{max}]$

⇩

Denoising score matching for all λ (noise)

Denoising Score Matching

$$\boxed{\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\right\|_2^2\right]}$$

With this resemblance... ⇨ $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) \approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda)$

Background

❑ Training Objective

$$\mathbb{E}_{\boldsymbol{\epsilon},\lambda}\left[\left\|\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) - \boldsymbol{\epsilon}\right\|_2^2\right] \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}),\ \mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon},\ \boxed{\lambda \sim p(\lambda) \text{ over } [\lambda_{min}, \lambda_{max}]}$$

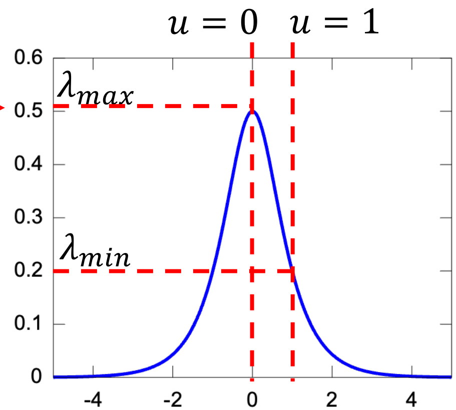
$$\boxed{\lambda = -2\log\tan(au + b)}$$

$$a = \arctan\left(e^{-\lambda_{min}/2}\right) - b$$

$$b = \arctan\left(e^{-\lambda_{max}/2}\right)$$

$$u \in [0,1]$$

⇨ Sample λ



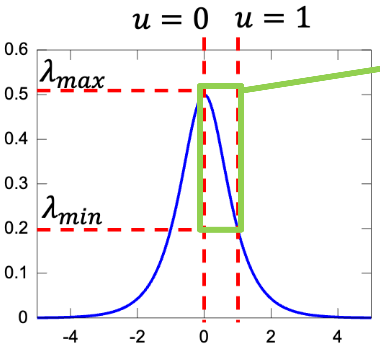
Hyperbolic secant distribution

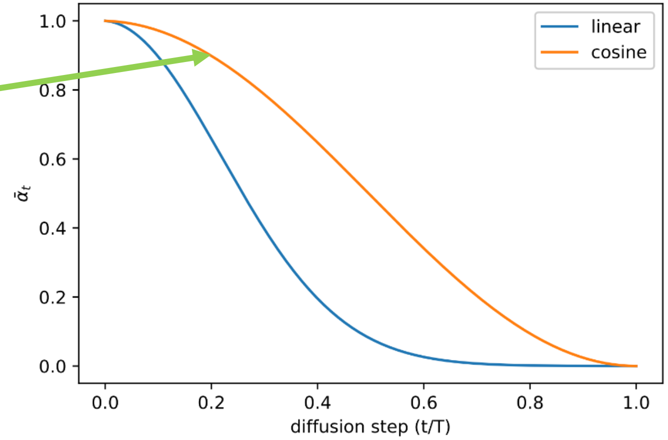
https://en.wikipedia.org/wiki/Hyperbolic_secant_distribution

Background

❑ Training Objective

$$\mathbb{E}_{\boldsymbol{\epsilon},\lambda}\left[\|\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) - \boldsymbol{\epsilon}\|_2^2\right] \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{I}),\ \mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon},\ \boxed{\lambda \sim p(\lambda) \text{ over } [\lambda_{min}, \lambda_{max}]}$$





Weighted variational lower bound

More sophisticated variance scheduling
➔ Improve sample quality

https://en.wikipedia.org/wiki/Hyperbolic_secant_distribution

Background

❑ Sampling Procedure

$$\boxed{\epsilon_\theta(\mathbf{z}_\lambda) \approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda)}$$

Score

Learned noise ≈ Estimation of $\nabla_{\boldsymbol{z}_\lambda} \log p(\boldsymbol{z}_\lambda)$

⇩

Correlation between **sampling from learned diffusion model** and **sampling with Langevin Dynamics**

Guidance

Guidance

- **Classifier Guidance**

Truncation-like effect in diffusion models

Conditional generative modeling

Conditions have impacts during training and sampling

Guidance

❑ Classifier Guidance

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

Diffusion Score during conditional generative modeling

Parameter determining the strength of classifier

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

$$\boxed{\tilde{p}_\theta(\mathbf{z}_\lambda|\mathbf{c}) \propto p_\theta(\mathbf{z}_\lambda|\mathbf{c}) p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^w}$$

Classifier-guided distribution (model)

Guidance

❑ **Classifier Guidance**

$$\tilde{p}_\theta(\mathbf{z}_\lambda|\mathbf{c}) \propto p_\theta(\mathbf{z}_\lambda|\mathbf{c})\boxed{p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^w}$$

➡ **Up-weighting the effect of classifier**

➡ **Higher likelihood to the correct label**

➡ **Improve IS/FID trade-off by setting $w > 0$**

Guidance

❑ **Simple Classifier Guidance Experiment**

Densities of Mixtures of 3 Gaussians

Guidance

❑ **Classifier Guidance with unconditional model**

$$\xrightarrow{\text{Bayes Rule}}$$

$$p_\theta(\mathbf{z}_\lambda|\mathbf{c})p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^w \propto p_\theta(\mathbf{z}_\lambda)p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^{w+1}$$

$$\begin{aligned}\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) - (w+1)\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda) &\approx -\sigma_\lambda \nabla_{\mathbf{z}_\lambda}[\log p(\mathbf{z}_\lambda) + (w+1)\log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)] \\ &= \boxed{-\sigma_\lambda \nabla_{\mathbf{z}_\lambda}[\log p(\mathbf{z}_\lambda|\mathbf{c}) + w \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)]}\end{aligned}$$

Same Diffusion score

Guidance

❑ Classifier Guidance

Better Performance

Conditional Model

$$\boxed{\tilde{p}_\theta(\mathbf{z}_\lambda|\mathbf{c}) \propto p_\theta(\mathbf{z}_\lambda|\mathbf{c})p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^w}$$

Unconditional Model

$$p_\theta(\mathbf{z}_\lambda|\mathbf{c})p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^w \propto p_\theta(\mathbf{z}_\lambda)p_\theta(\mathbf{c}|\mathbf{z}_\lambda)^{w+1}$$

⇨ Classifier-guided conditional model is used for comparison

Guidance

- Classifier-Free Guidance (CFG)

Much simpler implementation

Classifier-guided effect without extra classifier

Guidance

- ❑ **Training Classifier-Free Guidance (CFG) diffusion model**

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training

1: **repeat**
2: $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$
3: $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond}
4: $\lambda \sim p(\lambda)$
5: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6: $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$
7: Take gradient step on $\nabla_\theta \left\| \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon} \right\|^2$
8: **until** converged

Similar training process to that of DDPM

Guidance

❑ Training Classifier-Free Guidance (CFG) diffusion model

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: $\boxed{p_{\text{uncond}}\text{: probability of unconditional training}}$

```
repeat
    (x, c) ~ p(x, c)
    c ← ∅ with probability p_uncond
    λ ~ p(λ)
    ε ~ N(0, I)
    z_λ = α_λ x + σ_λ ε
    Take gradient step on ∇_θ ‖ε_θ(z_λ, c) − ε‖²
until converged
```

Repeat
- $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$
- $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond}
- $\lambda \sim p(\lambda)$
- $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$
- Take gradient step on $\nabla_\theta \left\| \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon} \right\|^2$

until converged

Unconditional model & Conditional model is being trained with 1 neural network

Hyperparameter p_{uncond} decides **which model to train** during the current iteration

Guidance

❑ Training Classifier-Free Guidance (CFG) diffusion model

Algorithm 1 Joint training a diffusion model with classifier-free guidance

Require: p_{uncond}: probability of unconditional training

repeat
  $(\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$
  $\mathbf{c} \leftarrow \varnothing$ with probability p_{uncond}
  $\lambda \sim p(\lambda)$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \epsilon$
  Take gradient step on $\nabla_\theta \left\| \epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \epsilon \right\|^2$
until converged

$1 - p_{uncond}$: Conditional Model is trained

Assign null token $\emptyset$ for $\boldsymbol{c}$ **(class identifier)**

p_{uncond} : Unconditional Model is trained

Guidance

❑ **Training Classifier-Free Guidance (CFG) diffusion model**

Algorithm 1 Joint training a diffusion model with classifier-free guidance

---

Require: p_{uncond}: probability of unconditional training

repeat

$\quad (\mathbf{x}, \mathbf{c}) \sim p(\mathbf{x}, \mathbf{c})$

$\quad \mathbf{c} \leftarrow \varnothing$ with probability p_{uncond}

$\quad \lambda \sim p(\lambda)$

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

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

$\quad$ Take gradient step on $\nabla_\theta \left\| \boxed{\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c})} - \boldsymbol{\epsilon} \right\|^2$

until converged

$$\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) \to \begin{cases} 1 - p_{uncond} : \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c}) \\ p_{uncond} : \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda) \end{cases}$$

Guidance

❑ Sampling from Classifier-Free Guidance (CFG) diffusion model

Algorithm 2 Conditional sampling with classifier-free guidance

Require: w: guidance strength
Require: $\mathbf{c}$: conditioning information for conditional sampling ⇨ **Conditions utilized during training**
Require: $\lambda_1, \ldots, \lambda_T$: increasing log SNR sequence with $\lambda_1 = \lambda_{\min}$, $\lambda_T = \lambda_{\max}$

Sampling Order

1: $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = 1, \ldots, T$ **do**
▷ Form the classifier-free guided score at log SNR λ_t
3: $\boxed{\tilde{\boldsymbol{\epsilon}}_t = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)}$ **Main point**
▷ Sampling step (could be replaced by another sampler, e.g. DDIM)
4: $\tilde{\mathbf{x}}_t = (\mathbf{z}_t - \sigma_{\lambda_t}\tilde{\boldsymbol{\epsilon}}_t)/\alpha_{\lambda_t}$
5: $\mathbf{z}_{t+1} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda_{t+1}|\lambda_t}(\mathbf{z}_t, \tilde{\mathbf{x}}_t), (\tilde{\sigma}^2_{\lambda_{t+1}|\lambda_t})^{1-v}(\sigma^2_{\lambda_t|\lambda_{t+1}})^v)$ if $t < T$ else $\mathbf{z}_{t+1} = \tilde{\mathbf{x}}_t$
6: **end for**
7: **return** $\mathbf{z}_{T+1}$

Also resemble DDPM sampling procedure

Guidance

- ❑ **Suggested score from Classifier-Free Guidance**

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

$$\tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c}) = w\left(\boxed{\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda)}\right) + \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c})$$

Implicit classifier guidance

Parameter determines the **strength**

Guidance

❑ **Classifier-guided Effect**

$$\left(\boxed{\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \epsilon_\theta(\mathbf{z}_\lambda)}\right)$$

Implicit classifier guidance

Recap – Classifier-guided Score

$$\tilde{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = \underbrace{\boxed{\epsilon_\theta(\mathbf{z}_\lambda, \mathbf{c})}}_{\text{Score (Conditional generative model)}} - w\underbrace{\boxed{\sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p_\theta(\mathbf{c}|\mathbf{z}_\lambda)}}_{\text{Score (Classifier)}}$$

Guidance

❑ Classifier-guided Effect

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

Implicit classifier guidance

Just estimation, not actual classifier gradient

$$\left[\ \tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c}) = w\big(\underline{\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda)}\big) + \underline{\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_\lambda, \boldsymbol{c})}\ \right]$$

Classifier Guidance

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

Score (Conditional generative model)

Score (Classifier)

Experiments

Experiments

❑ Experiment

Main Purpose

Demonstrating attaining IS/FID trade-off similar to that of classifier guidance

⇩

Same model architecture & hyperparameter settings from classifier guidance

Suboptimal for classifier-free guidance diffusion model

Experiments

❑ Experiment Settings

IS/FID score calculation with 50K Samples

$\lambda_{min} = -20, \lambda_{max} = 20$

64 X 64 Conditional ImageNet Model

Sampler noise interpolation coefficient $v = \mathbf{0.3}$, **400K** Training steps

128 X 128 Conditional ImageNet Model

Sampler noise interpolation coefficient $v = \mathbf{0.2}$, **2.7M** Training steps

Experiments

❑ Varying Classifier-Free Guidance Strength

w ↑

FID ↑ & IS ↑

Model	FID (↓)	IS (↑)
ADM (Dhariwal & Nichol, 2021)	2.07	-
CDM (Ho et al., 2021)	**1.48**	67.95
Ours	$p_{\text{uncond}} = 0.1/0.2/0.5$	
$w = 0.0$	1.8 / 1.8 / 2.21	53.71 / 52.9 / 47.61
$w = 0.1$	1.55 / 1.62 / 1.91	66.11 / 64.58 / 56.1
$w = 0.2$	2.04 / 2.1 / 2.08	78.91 / 76.99 / 65.6
$w = 0.3$	3.03 / 2.93 / 2.65	92.8 / 88.64 / 74.92
$w = 0.4$	4.3 / 4 / 3.44	106.2 / 101.11 / 84.27
$w = 0.5$	5.74 / 5.19 / 4.34	119.3 / 112.15 / 92.95
$w = 0.6$	7.19 / 6.48 / 5.27	131.1 / 122.13 / 102
$w = 0.7$	8.62 / 7.73 / 6.23	141.8 / 131.6 / 109.8
$w = 0.8$	10.08 / 8.9 / 7.25	151.6 / 140.82 / 116.9
$w = 0.9$	11.41 / 10.09 / 8.21	161 / 150.26 / 124.6
$w = 1.0$	12.6 / 11.21 / 9.13	170.1 / 158.29 / 131.1
$w = 2.0$	21.03 / 18.79 / 16.16	225.5 / 212.98 / 183
$w = 3.0$	24.83 / 22.36 / 19.75	250.4 / 237.65 / 208.9
$w = 4.0$	26.22 / 23.84 / 21.48	**260.2** / 248.97 / 225.1

Table 1: ImageNet 64x64 results ($w = 0.0$ refers to non-guided models).

Experiments

❑ Varying Classifier-Free Guidance Strength

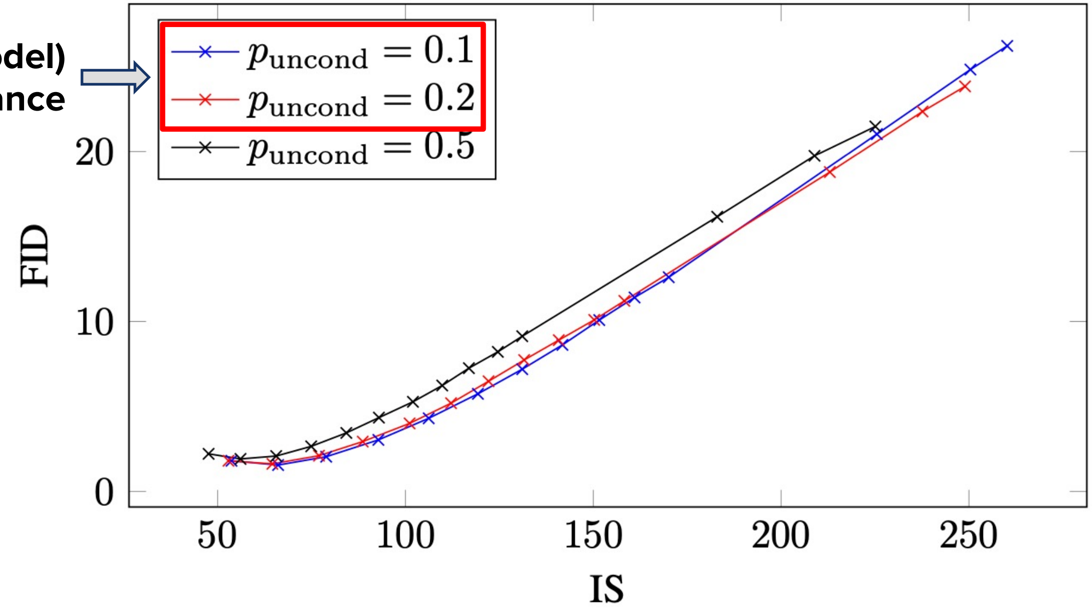
$w = 0$ (Non-guided)

$w = 3.0$

Experiments

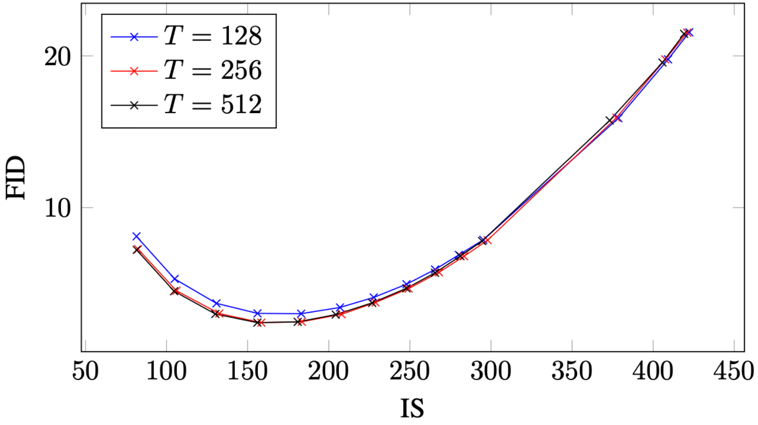
❑ Varying Unconditional Training Probability

(Small portion for unconditional model) Best Performance



Experiments

❑ Varying Sampling Steps



Model	FID (↓)	IS (↑)
BigGAN-deep, max IS (Brock et al., 2019)	25	253
BigGAN-deep (Brock et al., 2019)	5.7	124.5
CDM (Ho et al., 2021)	3.52	128.8
LOGAN (Wu et al., 2019)	3.36	148.2
ADM-G (Dhariwal & Nichol, 2021)	2.97	-
Ours	$T = 128/256/1024$	
$w = 0.0$	8.11 / 7.27 / 7.22	81.46 / 82.45 / 81.54
$w = 0.1$	5.31 / 4.53 / 4.5	105.01 / 106.12 / 104.67
$w = 0.2$	3.7 / 3.03 / 3	130.79 / 132.54 / 130.09
$w = 0.3$	3.04 / **2.43** / **2.43**	156.09 / 158.47 / 156
$w = 0.4$	3.02 / 2.49 / 2.48	183.01 / 183.41 / 180.88
$w = 0.5$	3.43 / 2.98 / 2.96	206.94 / 207.98 / 204.31
$w = 0.6$	4.09 / 3.76 / 3.73	227.72 / 228.83 / 226.76
$w = 0.7$	4.96 / 4.67 / 4.69	247.92 / 249.25 / 247.89
$w = 0.8$	5.93 / 5.74 / 5.71	265.54 / 267.99 / 265.52
$w = 0.9$	6.89 / 6.8 / 6.81	280.19 / 283.41 / 281.14
$w = 1.0$	7.88 / 7.86 / 7.8	295.29 / 297.98 / 294.56
$w = 2.0$	15.9 / 15.93 / 15.75	378.56 / 377.37 / 373.18
$w = 3.0$	19.77 / 19.77 / 19.56	409.16 / 407.44 / 405.68
$w = 4.0$	21.55 / 21.53 / 21.45	**422.29** / 421.03 / 419.06

Table 2: ImageNet 128x128 results ($w = 0.0$ refers to non-guided models).

Experiments

❑ Varying Sampling Steps

Model	FID (↓)	IS (↑)
BigGAN-deep, max IS (Brock et al., 2019)	25	253
BigGAN-deep (Brock et al., 2019)	5.7	124.5
CDM (Ho et al., 2021)	3.52	128.8
LOGAN (Wu et al., 2019)	3.36	148.2
ADM-G (Dhariwal & Nichol, 2021)	2.97	-
Ours	$T = 128/256/1024$	
$w = 0.0$	8.11 / 7.27 / 7.22	81.46 / 82.45 / 81.54
$w = 0.1$	5.31 / 4.53 / 4.5	105.01 / 106.12 / 104.67
$w = 0.2$	3.7 / 3.03 / 3	130.79 / 132.54 / 130.09
$w = 0.3$	3.04 / **2.43** / **2.43**	156.09 / 158.47 / 156
$w = 0.4$	3.02 / 2.49 / 2.48	183.01 / 183.41 / 180.88
$w = 0.5$	3.43 / 2.98 / 2.96	206.94 / 207.98 / 204.31
$w = 0.6$	4.09 / 3.76 / 3.73	227.72 / 228.83 / 226.76
$w = 0.7$	4.96 / 4.67 / 4.69	247.92 / 249.25 / 247.89
$w = 0.8$	5.93 / 5.74 / 5.71	265.54 / 267.99 / 265.52
$w = 0.9$	6.89 / 6.8 / 6.81	280.19 / 283.41 / 281.14
$w = 1.0$	7.88 / 7.86 / 7.8	295.29 / 297.98 / 294.56
$w = 2.0$	15.9 / 15.93 / 15.75	378.56 / 377.37 / 373.18
$w = 3.0$	19.77 / 19.77 / 19.56	409.16 / 407.44 / 405.68
$w = 4.0$	21.55 / 21.53 / 21.45	**422.29** / 421.03 / 419.06

Table 2: ImageNet 128x128 results ($w = 0.0$ refers to non-guided models).

Classifier-Guidance Diffusion

$$T = 256$$

Best balance

between sample quality & sampling speed

CFG Diffusion model should go through **2 times of forward process**

(Conditional & Unconditional Model Training)

Leading to Slow Sampling Speed

Thank you