Classifier-Free Diffusion Guidance

Anonymous Author(s)
Affiliation
Address
email

Abstract

Classifier guidance is a recently introduced method to trade off mode coverage and sample fidelity in conditional diffusion models post training, in the same spirit as low temperature sampling or truncation in other types of generative models. This method combines the score estimate of a diffusion model with the gradient of an image classifier and thereby requires training an image classifier separate from the diffusion model. We show that guidance can be performed by a pure generative model without such a classifier: we jointly train a conditional and an unconditional diffusion model, and find that it is possible to combine the resulting conditional and unconditional scores to attain a trade-off between sample quality and diversity similar to that obtained using classifier guidance.

1 Introduction

Diffusion models have recently emerged as an expressive and flexible family of generative models, delivering competitive sample quality and likelihood scores on image and audio synthesis tasks [14, 15, 5, 16, 8]. These models have delivered audio synthesis performance rivaling the quality of autoregressive models with substantially fewer inference steps [2, 9], and they have delivered ImageNet generation results outperforming BigGAN-deep [1] and VQ-VAE-2 [11] in terms of FID score and classification accuracy score [6, 3].

Dhariwal and Nichol [3] proposed *classifier guidance*, a technique to boost the sample quality of a diffusion model using an extra trained classifier. Using classifier guidance, they generate high fidelity, non-diverse ImageNet samples that match or exceed the Inception scores of truncated BigGAN, and by varying the strength of the classifier gradient, they can trade off Inception score [13] and FID score [4] (or precision and recall) in a manner similar to varying the truncation parameter of BigGAN.

Prior to classifier guidance, it was not known how to generate "low temperature" samples from a diffusion model similar to those produced by truncated BigGAN: naive ways of doing so, such as scaling the model score vectors or decreasing the amount of Gaussian noise added during sampling, do not work. Classifier guidance resolves this issue but raises more questions: (1) Is it possible to achieve the same effect using a pure generative model without any classifier? (2) Is it necessary to use a classifier gradient to achieve this effect, and is classifier guidance able to boost classifier-based metrics such as Inception score and FID score simply because classifier guidance is adversarial to image classifiers and because classifier gradients have special structure? (3) What is an intuitive explanation for what is going on during guided sampling?

By presenting and analysing *classifier-free guidance*, we provide some answers to these questions.

2 Background

Let $\mathbf{x}$ be data drawn from a data distribution $p(\mathbf{x})$. We train a diffusion model in continuous time [16, 2, 8]: letting $\mathbf{z} = \{\mathbf{z}_\lambda \,|\, \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$ for hyperparameters $\lambda_{\min} < \lambda_{\max} \in \mathbb{R}$, the forward process $q(\mathbf{z}|\mathbf{x})$ is the variance-preserving Markov process [14] specified as

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

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

We will use the notation $p(\mathbf{z})$ (or $p(\mathbf{z}_\lambda)$) to denote the marginal of $\mathbf{z}$ (or $\mathbf{z}_\lambda$) when $\mathbf{x} \sim p(\mathbf{x})$. Note that $\lambda = \log \alpha_\lambda^2/\sigma_\lambda^2$, so λ can be interpreted as the log signal-to-noise ratio of $\mathbf{z}_\lambda$, and the forward process runs in the direction of decreasing λ. Conditioned on $\mathbf{x}$, the forward process can be described in reverse by the transitions $q(\mathbf{z}_{\lambda'}|\mathbf{z}_\lambda, \mathbf{x}) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}), \tilde{\sigma}_{\lambda'|\lambda}^2 \mathbf{I})$, where

$$\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}) = e^{\lambda-\lambda'}(\alpha_{\lambda'}/\alpha_\lambda)\mathbf{z}_\lambda + (1-e^{\lambda-\lambda'})\alpha_{\lambda'}\mathbf{x}, \quad \tilde{\sigma}_{\lambda'|\lambda}^2 = (1-e^{\lambda-\lambda'})\sigma_{\lambda'}^2 \tag{3}$$

The reverse process generative model $p_\theta(\mathbf{z})$ starts from $p_\theta(\mathbf{z}_{\lambda_{\min}}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. We specify the transitions:

$$p_\theta(\mathbf{z}_{\lambda'}|\mathbf{z}_\lambda) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_{\lambda'|\lambda}(\mathbf{z}_\lambda, \mathbf{x}_\theta(\mathbf{z}_\lambda)), (\tilde{\sigma}_{\lambda'|\lambda}^2)^{1-v}(\sigma_{\lambda|\lambda'}^2)^v) \tag{4}$$

During sampling, we apply this transition along an increasing sequence $\lambda_{\min} = \lambda_1 < \cdots < \lambda_T = \lambda_{\max}$ for T timesteps. If the model $\mathbf{x}_\theta$ is correct, then as $T \to \infty$, we obtain samples from an SDE whose sample paths are distributed as $p(\mathbf{z})$ [16]. The variance is a log-space interpolation of $\tilde{\sigma}_{\lambda'|\lambda}^2$ and $\sigma_{\lambda|\lambda'}^2$ as suggested by [10]; for simplicity we use a constant hyperparameter v rather than learned $\mathbf{z}_\lambda$-dependent v. Note that variances simplify to $\tilde{\sigma}_{\lambda'|\lambda}^2$ as $\lambda' \to \lambda$, so v has an effect only when sampling with non-infinitesimal timesteps as done in practice.

The reverse process mean comes from an estimate $\mathbf{x}_\theta(\mathbf{z}_\lambda) \approx \mathbf{x}$ plugged into $q(\mathbf{z}_{\lambda'}|\mathbf{z}_\lambda, \mathbf{x})$ [5, 8] ($\mathbf{x}_\theta$ also receives λ as input, but we suppress this to keep our notation clean). We parameterize $\mathbf{x}_\theta$ in terms of $\boldsymbol{\epsilon}$-prediction [5]: $\mathbf{x}_\theta(\mathbf{z}_\lambda) = (\mathbf{z}_\lambda - \sigma_\lambda \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda))/\alpha_\lambda$, and we train on the objective

$$\mathbb{E}_{\boldsymbol{\epsilon},\lambda}\left[\|\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) - \boldsymbol{\epsilon}\|_2^2\right] \tag{5}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}_\lambda = \alpha_\lambda \mathbf{x} + \sigma_\lambda \boldsymbol{\epsilon}$, and λ is drawn from a distribution $p(\lambda)$ over $[\lambda_{\min}, \lambda_{\max}]$. This objective is denoising score matching [17] over multiple noise scales [15], and when $p(\lambda)$ is uniform, the objective is proportional to the variational lower bound on the marginal log likelihood of the latent variable model $\int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$, ignoring the term for the unspecified $p_\theta(\mathbf{x}|\mathbf{z})$ and for the prior at $\mathbf{z}_{\lambda_{\min}}$ [8]. For a different distribution $p(\lambda)$, the objective can be interpreted as weighted variational lower bound whose weighting can be tuned for sample quality [5]. We use a $p(\lambda)$ inspired by the cosine noise schedule of [10]: sampling λ is given by $\lambda = -2\log\tan(au+b)$ for uniformly distributed $u \in [0,1]$, where $b = \arctan(e^{-\lambda_{\max}/2})$ and $a = \arctan(e^{-\lambda_{\min}/2}) - b$. This represents a hyperbolic secant distribution modified to be supported on a bounded interval. For finite timestep sampling, we use λ values corresponding to uniformly spaced $u \in [0,1]$.

Because the loss for $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$ is denoising score matching for all λ, the score $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$ learned by our model estimates the gradient of the log-density of the distribution of our noisy data $\mathbf{z}_\lambda$, that is $\boldsymbol{\epsilon}(\mathbf{z}_\lambda) \approx \sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda)$. Sampling from the learned diffusion model resembles using Langevin diffusion to sample from a sequence of distributions $p(\mathbf{z}_\lambda)$ that converges to the conditional distribution $p(\mathbf{x})$ of the original data $\mathbf{x}$.

In the case of conditional generative modeling, the data $\mathbf{x}$ is drawn jointly with conditioning information $\mathbf{c}$, i.e. a class label for class-conditional image generation. The only modification to the model is that the reverse process function approximator receives $\mathbf{c}$ as input, as in $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c})$.

3 Guidance

An interesting property of certain generative models, such as GANs and flow-based models, is the ability to perform truncated or low temperature sampling by decreasing the variance or range of noise inputs to the generative model at sampling time. The intended effect is to decrease the diversity of the samples while increasing the quality of each individual sample. Truncation in BigGAN [1], for example, yields a tradeoff curve between FID score and Inception score for low and high amounts of truncation, respectively. Low temperature sampling in Glow [7] has a similar effect.

3.1 Classifier guidance

Unfortunately, straightforward attempts of implementing truncation or low temperature sampling in diffusion models are ineffective. For example, scaling model scores or decreasing the variance of Gaussian noise in the reverse process cause the diffusion model to generate blurry, low quality samples [3].

To obtain a truncation-like effect in diffusion models, Dhariwal and Nichol [3] introduce *classifier guidance*, where the diffusion score $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) \approx \sigma_\lambda \nabla_{\mathbf{z}_\lambda} \log p(\mathbf{z}_\lambda|\mathbf{c})$ is modified to include the gradient of the log likelihood of an auxiliary classifier model $p_\theta(\mathbf{c}|\mathbf{z}_\lambda)$ as follows:

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

where w is a parameter that controls the strength of the classifier guidance. This modified score $\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c})$ is then used in place of $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c})$ when sampling from the diffusion model, which has the effect of up-weighting the probability of data for which the classifier $p_\theta(\mathbf{c}|\mathbf{z}_\lambda)$ assigns high likelihood to the correct label: data that can be classified well scores high on the Inception score of perceptual quality [13], which rewards generative models for this by design. Dhariwal and Nichol [3] therefore find that by setting $w > 0$ they can improve the Inception score of their diffusion model, at the expense of decreased diversity in their samples. Interestingly, they obtain their best results when applying classifier guidance to an already class-conditional model as described above, and they find that applying guidance to an unconditional model performs less well: the effects of class-conditioning and guidance thus seem complimentary.

3.2 Classifier-free guidance

A downside of classifier guidance is that it requires an additional classifier model and thus complicates the training pipeline. This model has to be trained on noisy data $\mathbf{z}_\lambda$, so it is not possible to plug in a standard pre-trained classifier. We explore an alternative method of modifying $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c})$ to achieve the same effect of boosting the perceptual quality as measured by the Inception score without requiring an auxiliary classifier. We call this new method *classifier-free guidance*.

Instead of training a separate classifier model, we choose to train an unconditional denoising diffusion model $p_\theta(\mathbf{z})$ parameterized through a score estimator $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)$ together with the conditional model $p_\theta(\mathbf{z}|\mathbf{c})$ parameterized through $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c})$. We use a single neural network to parameterize both models, where for the unconditional model we can simply input zeros for the class identifier $\mathbf{c}$ when predicting the score, i.e. $\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) = \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c} = \mathbf{0})$. We jointly train the unconditional and conditional models simply by randomly setting $\mathbf{c}$ to the unconditional class identifier.

We can then apply Bayes' rule to obtain an *implicit classifier* as $p^i_\theta(\mathbf{c}|\mathbf{z}_\lambda) \propto p_\theta(\mathbf{z}_\lambda|\mathbf{c})/p_\theta(\mathbf{z}_\lambda)$. The score of this implicit classifier will then be given by $\nabla_{\mathbf{z}_\lambda} \log p^i_\theta(\mathbf{c}|\mathbf{z}_\lambda) \approx \frac{1}{\sigma_\lambda}[\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda)]$. Applying classifier guidance with this implicit classifier yields the following modification to the diffusion score estimator:

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c}) = (1+w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda, \mathbf{c}) - w\boldsymbol{\epsilon}_\theta(\mathbf{z}_\lambda) \approx \sigma_\lambda \nabla_{\mathbf{z}_\lambda}[\log p_\theta(\mathbf{z}_\lambda|\mathbf{c}) + w \log p^i_\theta(\mathbf{c}|\mathbf{z}_\lambda)]. \tag{6}$$

We then use $\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_\lambda, \mathbf{c})$ to sample from our diffusion model as usual, thus producing approximate samples from $\tilde{p}_\theta(\mathbf{z}_\lambda|\mathbf{c}) \propto p_\theta(\mathbf{z}_\lambda|\mathbf{c}) p^i_\theta(\mathbf{c}|\mathbf{z}_\lambda)^w$.

4 Experiments

We apply our proposed classifier-free guidance to 64×64 area-downsampled ImageNet [12]. We trained a model with architecture and hyperparameters identical to the 64×64 model in [3], and we jointly trained the model on unconditional generation with probability 0.1. We choose $\lambda_{\min} = -20$, $\lambda_{\max} = 20$, and $v = 0.3$. We consider implied-classifier weights $w \in \{0, 0.1, 0.2, \ldots, 5\}$ and calculate FID and Inception Scores with 50000 samples for each value using $T = 256$ sampling steps. Figure 1 and Fig. 2 list our results: we obtain the best FID result with a small amount of guidance ($w = 0.1$) and the best IS result with strong guidance ($w \geq 4$). These results compare favorably to [3, 6] and are currently state-of-the-art for this data set as far as we are aware for models that use $T \approx 256$ steps (the ADM result uses 250 steps, and the CDM result is a two-stage model with 4000 steps each). Between these two extremes we see a clear trade-off between these two metrics of perceptual quality, with FID monotonically decreasing and IS monotonically increasing with guidance weight w.

Figure 3 shows randomly generated samples from our model for different levels of guidance: here we clearly see that increasing guidance has the effect of decreasing sample variety and increasing individual sample fidelity.

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

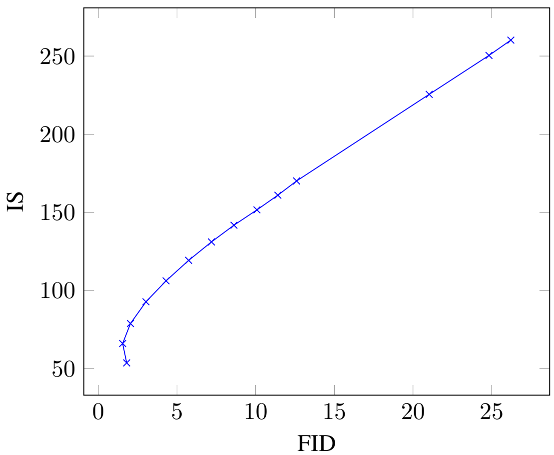


Figure 2: ImageNet 64x64 FID vs. IS

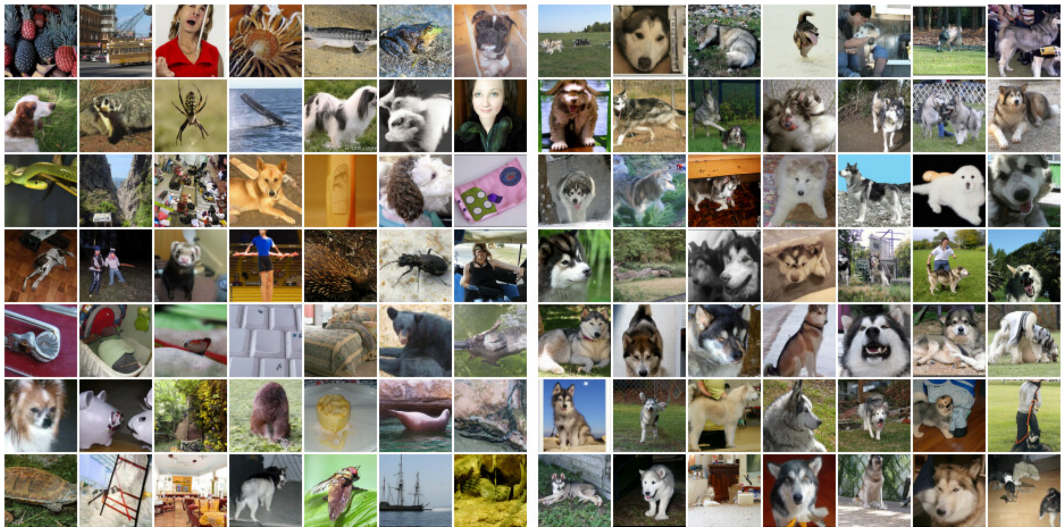
5 Conclusion

Returning to the questions we posed in the introduction: since classifier-free guidance is able to trade off IS and FID like classifier guidance without needing an extra trained classifier, we have resolved our question of whether guidance can be performed with a pure generative model. We confirm that it is possible to maximize Inception scores using classifier-free guidance (and FID score for a small amount of guidance), thus providing evidence that classifier-based sample quality metrics can be improved using methods that are not adversarial against ImageNet classifiers using classifier gradients. Finally also have an intuitive explanation for what guidance does: it decreases the unconditional likelihood of the sample while increasing the conditional likelihood. Our classifier-free guidance decreases the unconditional likelihood with a *negative* score term, which to our knowledge has not yet been explored and may find uses in other applications.

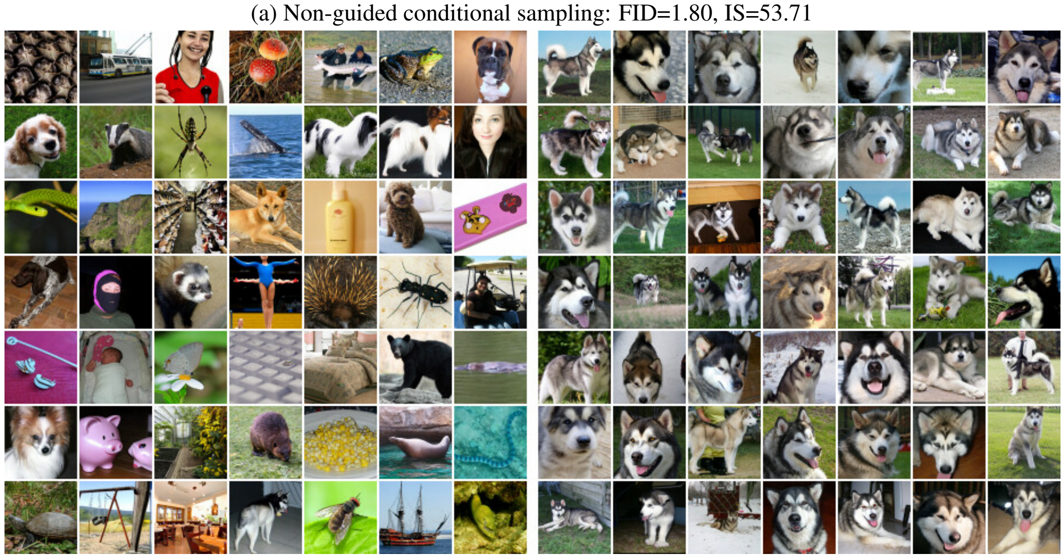
A potential disadvantage of classifier-free guidance is sampling speed. Generally, classifiers can be smaller and faster than generative models, so classifier guided sampling may be faster than classifier-free guidance because the latter needs to run two forward passes of the diffusion model, one for conditional score and another for the unconditional score. The necessity to run multiple passes of the diffusion model might be mitigated by changing the architecture to inject conditioning late in the network, but we leave this exploration for future work.

It may be possible to entirely avoid training an unconditional model. If we know the class distribution and there are only a few classes, we can use the fact that $\sum_c p(\mathbf{x}|c)p(c) = p(\mathbf{x})$ to obtain an unconditional score from conditional scores without explicitly training for the unconditional score. Of course, this would require as many forward passes as possible values of $\mathbf{c}$ and would be inefficient for high dimensional conditioning signals.

We have presented a method to increase sample quality while decreasing sample diversity, just like classifier guidance. There may be negative impacts of doing so in deployed models, since sample diversity is important to maintain in applications where certain parts of the data are underrepresented in the context of the rest of the data. It would be an interesting avenue of future work to try to boost sample quality while maintaining sample diversity.



(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.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[2] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating gradients for waveform generation. *International Conference on Learning Representations*, 2021.

[3] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020.

[6] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021.

[7] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[8] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

[9] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A Versatile Diffusion Model for Audio Synthesis. *International Conference on Learning Representations*, 2021.

[10] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, 2021.

[11] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[13] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[14] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

[15] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[16] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*, 2021.

[17] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]

    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
    (a) Did you state the full set of assumptions of all theoretical results? [N/A]
    (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Will be released later.
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [Yes] We used ImageNet.
    (b) Did you mention the license of the assets? [No] ImageNet is standard.
    (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]