Improving Conditional Score-Based Generation with Calibrated Classification and Joint Training

Paul Kuo-Ming Huang
National Taiwan University
b08902072@csie.ntu.edu.tw

Si-An Chen
National Taiwan University
d09922007@csie.ntu.edu.tw

Hsuan-Tien Lin
National Taiwan University
htlin@csie.ntu.edu.tw

Abstract

Score-based Generative Model (SGM) is a popular family of deep generative models that can achieve leading image generation quality. Earlier works have extended SGMs to tackle class-conditional generation with the guidance of well-trained classifiers. Nevertheless, we find that the classifier-guided SGMs actually do not achieve accurate conditional generation when evaluated with class-conditional measures. We argue that the lack of control roots from inaccurate gradients within the classifiers. We then propose to improve classifier-guided SGMs by calibrating classifiers using principles from energy-based models. In addition, we design a joint-training architecture to further enhance the conditional generation performance. Empirical results on CIFAR-10 demonstrate that the proposed model improves the conditional generation accuracy significantly while maintaining similar generation quality. The results support the potential of memory-efficient SGMs for conditional generation based on classifier guidance.

1 Introduction

Score-based generative models (SGMs) capture the data distribution by learning the gradient function of the log-likelihood on data, also known as the score function, providing a new way to estimate probability distributions. While the ground-truth score function is typically unknown and cannot be directly learned, previous work [7, 9, 13, 16] transformed the learning objective to train a model without knowing the ground-truth. Among all, denoising score matching (DSM) [16] is arguably one of the most computationally effective and inspired many successive works on score-based generation.

SGMs can perform both unconditional and class-conditional generation. For class-conditional generation, one family of methods estimates the conditional score as a mixture of an unconditional score and gradients of an auxiliary classifier [2, 14]. This extra classifier has been shown to improve generation quality and can further be controlled to trade off between sample diversity and fidelity [2]. While classifiers trained with cross-entropy loss are known to be over-confident [4, 8, 10, 11], they may lead to inaccurate score functions that deteriorate the conditional generation. In this work, we proposed *Score Calibration loss* to solve this problem. We derive the loss function in a principled manner. Then, we demonstrate the helpfulness of the loss with thorough experiments. The results demonstrate that the loss can indeed improve class-conditional image generation. Given the recent success of joint network learning [1, 6, 15] in various fields, we hypothesize that joint network learning can be integrated into the current framework for better performance. Our empirical study on CIFAR-10 shows that the new loss improves intra-FID from 16.11 to 11.70 and generation accuracy from 66.4% to 82.9%. Integrating joint training further improves generation accuracy to 94.3%.

NeurIPS 2022 Workshop on Score-Based Methods.

2 Background

2.1 Classifier guidance for score-based generation

The goal of classifier guidance is to estimate the conditional score $\nabla_x p(x|y)$ from a dataset (x_n, y_n) where $x_n \in \mathbb{R}^N$ and y_n is the class corresponding to x_n. Previous works [2, 14] showed how to decompose the conditional score function using Bayes' theorem:

$$\nabla_x \log p(x|y) = \nabla_x[\log p(x) + \log p(y|x) - \log p(y)] = \nabla_x \log p(x) + \nabla_x \log p(y|x) \quad (1)$$

where y is the target category and term $\log p(y)$ can be circumvented because it is not a function of x and its gradient evaluates to 0. The formulation shows that conditional generation can be achieved by an unconditional SGM guided by an additional classifier.

The term $\log p(y|x)$ is estimated using an extra classifier trained with cross-entropy loss because of its success in probabilistic discriminative modeling. Such classifiers are notorious to be overconfident [4, 8, 10, 11] on its predictions hence leads to inaccurate gradients to the conditional SGM. Therefore, we hypothesized that the classifier requires calibration to provide more accurate gradients.

2.2 Reinterpreting classifiers as energy-based models

JEM [4] has shown that reinterpreting and calibrating classifiers as energy-based models is beneficial for classifiers to capture more accurate probability distribution. The authors first demonstrated how classifiers can be reinterpreted as energy-based models, which are models that estimate energy functions (negative log-likelihood) of distributions. They observed that given the logits of a classifier to be $f_\phi(x)$, the joint distribution estimated by the classifier can be written as $p_\phi(x, y) = \frac{\exp(f_\phi(x)[y])}{Z(\phi)}$, where $\exp(\cdot)$ means exponential, $f_\phi(x)[y]$ is the y-th logit, and $Z(\phi) = \int_{x',y'} \exp(f_\phi(x')[y'])\, dx'\, dy'$ is a normalizing constant. The energy function can be obtained by:

$$E_\phi(x) = -\log p_\phi(x) = -\log \sum_y \frac{\exp(f_\phi(x)[y])}{Z(\phi)} = -\mathrm{LogSumExp}_y(f_\phi(x)[y]) + \log Z(\phi) \quad (2)$$

where $\mathrm{LogSumExp}_y(\cdot) = \log \Sigma_y \exp(\cdot)$.

The authors solved the optimization of JEM by training the classifier with an auxiliary loss derived from Eq. 2 and demonstrated that JEM is a well-calibrated classifier in their empirical study.

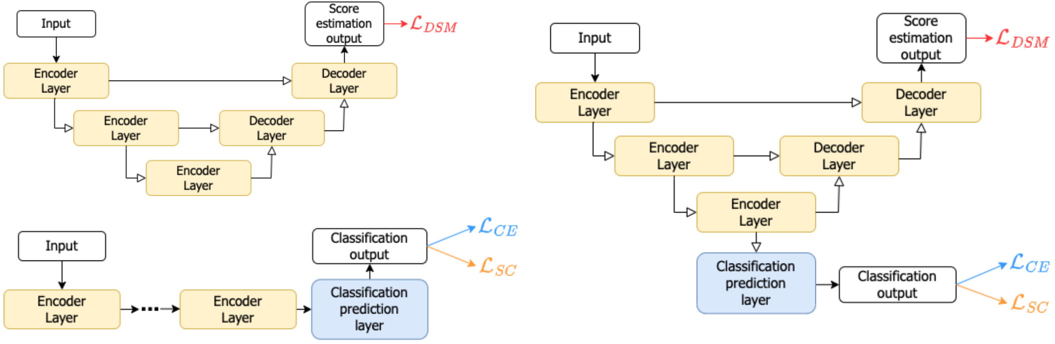
3 Calibrated classifier guidance

In our work, we adopted the framework of score-based generative modeling using stochastic differential equations (SDEs) [14]. Given a target distribution $p_0(x)$ and a known prior distribution $p_T(x)$ (typically a Gaussian distribution) where the transition between them is a diffusion process with timestep $0 \leq t < T$, we can describe the diffusion process and its reverse process using SDEs. To incorporate results of Section 2 into this framework, we introduce the time-dependent version of $\nabla_x \log p(x)$ and $\nabla_x \log p(y|x)$. That is $\nabla_x \log p_t(x(t))$ and $\nabla_x \log p_t(y|x(t))$, respectively, where $x(t) \sim p_t$. Denoising score matching (DSM) [16] is often utilized to train the score-based model under this framework due to its close relationship with diffusion process modeling. To train the classifier, we can adopt a time-generalized cross-entropy loss.

In section 2.1, we have mentioned that classifiers need to be calibrated to provide accurate gradients for classifier guidance. In our work, we show that classifier guidance can be further improved by calibration during the training stage. Inspired by JEM [4], we reinterpret the classifier as a time-dependent energy-based model and obtain the score function by calculating the gradient. Given the close relationship between energy function $-\log p(x)$ and score function $\nabla_x \log p(x)$, we hypothesize that integrating a similar objective into classifier training can be beneficial to classifier guidance. To incorporate the energy function into our framework, we used a time-dependent version of Eq. 2:

$$E_{\phi,t}(x) = -\log \sum_y \frac{\exp(f_{\phi,t}(x)[y])}{Z_t(\phi)} = -\mathrm{LogSumExp}_y(f_{\phi,t}(x)[y]) + \log Z_t(\phi)$$

Figure 1: The image on the left and right shows the model architecture without and with joint training correspondingly. The encoder of SBM and the classifier are shared in the joint model. Cross-entropy loss: $\mathcal{L}_{\mathrm{CE}}$. Score calibration loss: $\mathcal{L}_{\mathrm{SC}}$. DSM loss: $\mathcal{L}_{\mathrm{DSM}}$.



The score function can then be computed like the following:

$$s_\phi(x,t) = \nabla_x \log(p_{\phi,t}(x)) = \nabla_x \mathrm{LogSumExp}_y(f_{\phi,t}(x)[y])$$

To calibrate this score estimated by the classifier, we utilize DSM to calculate the *Score Calibration Loss (SC loss)*:

$$\mathcal{L}_{\mathrm{SC}}(\phi) = \mathbb{E}_t\left[\lambda(t)\mathbb{E}_{p_0(x)}\mathbb{E}_{p_t(x(t)|x(0))}\left[\frac{1}{2}\left\|s_\phi(x(t),t) - \nabla_{x(t)}\log p_t(x(t)|x(0))\right\|_2^2\right]\right] \tag{3}$$

After the score calibration loss is obtained, it is summed with the cross-entropy loss to train the classifier. The total loss can be written as:

$$\mathcal{L}_{\mathrm{CLS}}(\phi) = \mathcal{L}_{\mathrm{CE}}(\phi) + \lambda_{\mathrm{SC}}\mathcal{L}_{\mathrm{SC}}(\phi) \tag{4}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss and λ_{SC} is a hyperparameter. The calibrated classifier then can be used to guide an unconditional SGM to achieve conditional generation.

4 Joint network learning

Joint training has been shown to improve the performance and robustness of models by using the domain information of multiple related tasks to train a model in parallel [1, 6, 15]. Part of the network is shared to perform all tasks, and some task-specific parameters are integrated. The shared architecture learns a more general representation by leveraging the information contained in different tasks. Besides improvement in performance, another benefit of joint training is that number of trainable parameters needed to perform multiple tasks can be greatly reduced.

In this work, we take NCSN++ [14], as our basic architecture for the score-based models. We also develop a new joint version of NCSN++ that share the encoders between the score model and the classifier. As illustrated in Fig 1. When a data instance is passed through the encoders, the output is fed to a classifier to train the JEM, and decoders to train the score-base model, respectively. The losses for both tasks are then summed as follow.

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{DSM}} + \lambda_{\mathrm{joint}}(\mathcal{L}_{\mathrm{CE}} + \mathcal{L}_{\mathrm{SC}}) \tag{5}$$

Here, $\mathcal{L}_{\mathrm{DSM}}$ is the denoising score matching loss, and λ_{joint} is a hyperparameter to balance between the losses.

5 Experiments

We tested our methods on the CIFAR-10 dataset for image generation. We demonstrate that our methods are able to improve generation quality both conditionally and unconditionally.

Table 1: Sample quality comparison of all methods. **NCSN++** refers to the result of previous work [14] and **CNCSN++** replaces the normalization layers of the previous model with conditional normalization layers. **G** additionally uses classifier guidance and **GC** uses calibrated classifier.

Method	FID (↓)	IS (↑)	Intra-FID (↓)	Accuracy (↑)	Number of Parameters
NCSN++	**2.20**	9.89			107.6 M
NCSN++ (G)	2.25	9.81	16.11	0.664	151.4 M
NCSN++ (GC)	2.23	9.87	11.70	0.829	151.4 M
Joint Training (GC)	2.48	**9.91**	**11.03**	**0.943**	107.6 M
CNCSN++	2.13	10.05	10.29	0.970	107.9 M
CNCSN++ (G)	3.74	10.12	13.15	0.985	151.7 M
CNCSN++ (GC)	3.83	10.08	13.38	0.988	151.7 M

5.1 Experimental setup

Implementation details: We follow NCSN++ [14] to implement the unconditional score estimation model. We also adapted the encoder part of NCSN++ as the classifier used in classifier guidance [2].

Sampling method: We used Predictor-Corrector (PC) samplers [14] with 1000 sampling steps.

Evaluation metrics: Besides two commonly used metrics Frechet Inception Distance (FID) [5] and Inception Score (IS) [12], we also evaluated class-conditional performance of our methods using two different methods. The first one is intra-FID, which measures the average FID for each class. The second one is generation accuracy, which uses a pre-trained ViT [3] classifier to check whether the samples are generated in the correct class. The test accuracy of the pre-trained ViT is 98.52%. Besides, we also show number of trainable parameters for each method.

5.2 Results

Table 1 shows the result of all methods. **NCSN++** is the unconditional score-based model proposed in previous work [14]. **CNCSN++** is another approach for conditional generation. It adopts conditional score-based generative models by conditional normalization techniques [2] rather than using classifier guidance. We consider it a strong competitor for conditional generation. The suffix **G** means classifier guidance is applied. The suffix **GC** means SC loss is applied. We tuned λ_{joint} and λ_{SC} in $\{10, 1, 0.1, 0.01\}$ and set them to 1. Although the original classifier guidance **NCSN++ (G)** has great performance when evaluated with FID and IS, its intra-FID and generation accuracy dropped from 11.70 to 16.11 and 0.970 to 0.664 respectively. This supports our statement that the classifier does not provide accurate gradients to guide SGMs for conditional generation.

After calibration by SC loss, **NCSN++ (GC)** is able to improve all class-conditional metrics. While achieving similar unconditional generation performance, NCSN++ (GC) generates much more samples in the correct class compared with NCSN++ (G). Although integrating joint training caused a small drop in FID, all other metrics improved. Both SC loss and joint training are able to make the classifier provide more accurate gradients for classifier guidance.

CNCSN++ performs the best compared to all other methods for all metrics. There is a huge performance gap between CNCSN++ and NCSN++ (G) originally. After applying our methods, the difference in intra-FID and generation accuracy decreased from 5.82 and 30.6% to 0.74 and 2.7% respectively. Previous work [2] also demonstrated that the performance of score-based models with conditional normalization layers can be further improved by applying classifier guidance to the ImageNet dataset. However, a similar result does not happen in our experiments on CIFAR-10.

6 Conclusion

In this work, we showed that although classifier guidance is able to produce high-quality images, it is not a good class-conditional generation method as it generates many images from the incorrect class. To resolve this problem, we proposed to train the classifier with *Score Calibration Loss* in addition to cross-entropy loss and integrate joint training. The experimental results show that our method drastically improved class-conditional metrics compared to the original classifier guidance.

References

[1] Si-An Chen, Chun-Liang Li, and Hsuan-Tien Lin. A unified view of cgans with and without classifiers. In *Advances in Neural Information Processing Systems*, 2021.

[2] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[4] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020.

[5] 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 *NeurIPS*, 2017.

[6] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839, 2017. doi: 10.1109/ICASSP.2017.7953075.

[7] Durk P Kingma and Yann Cun. Regularized estimation of image statistics by score matching. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

[8] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

[9] James Martens, Ilya Sutskever, and Kevin Swersky. Estimating the hessian by back-propagating curvature. In *ICML*, 2012.

[10] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, pages 15288–15299, 2020.

[11] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2019.

[12] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[13] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 204, 2019.

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

[15] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*, 2018.

[16] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.