GGDM: Gradient-Guided Diffusion Models

Junru Lin **Aditya Mehrotra** **Neil Gupta**

Abstract

Diffusion models are a powerful class of generative models, capable of generating high-quality images. In this paper, we explore a method of conditioning diffusion models using differentiable loss functions and present a general framework for doing so. We pick two losses as examples to study this framework. We analyze the effect of a discriminative loss on the backward diffusion process. In addition, we investigate the effect of conditioning duration and conditioning strength on the quality of the generated images. Finally, we experiment with our method on a CLIP loss for text conditional image generation. Our code is provided in `https://github.com/AditMeh/GGDM`.

1 Introduction

Over the past few years, diffusion models [1][7][22] have been successful in producing photo-realistic images. One class of these models, known as conditional diffusion models [2][6], allows people to control the diffusion models generation process via text prompts [14][17][18], images [16][21][23], and other modalities [5][19][20].

A recent popular work, known as classifier guidance [2], puts forward a conditional diffusion model that utilizes a discriminative signal from a classifier as a method of conditioning. However, this model has shortcomings in the strength of the discriminative signal due to high levels of noise in the image inputs. In this paper, we try to tackle the problems with classifier guidance using denoised sample estimation [10]. We present a general framework for guiding diffusion models with any differentiable loss function.

2 Related Work

Diffusion Models Diffusion models are a new class of generative models that map a known distribution that we can easily sample from (such as standard normal distribution) to a data distribution (such as images). There are many different architectures, but we will primarily focus on Denoising Diffusion Probabilistic Models (DDPM) [7]. A detailed treatment of DDPMs is provided in [13].

Manifold Constrained Gradient (MCG) Guided Diffusion Models In [10], the authors introduced a method of conditioning diffusion models inspired by Manifold Constrained Gradient (MCG) for inverse problems. Their method guides the reverse diffusion process by using an estimate of the denoised image using noise2score [9] at each timestep to compute a loss function. Their choice of loss function is suited toward style transfer. In this paper, we further explore this method on different tasks and loss functions, to test its effectiveness.

Classifier Guidance In [2], the authors drew a connection between the diffusion model objective and score functions to build a conditional diffusion model using the gradient of a classifier. Ho *et al.* [6] built on this work by removing the need for an explicit classifier. This method was used in GLIDE [14] for text-guided image generation.

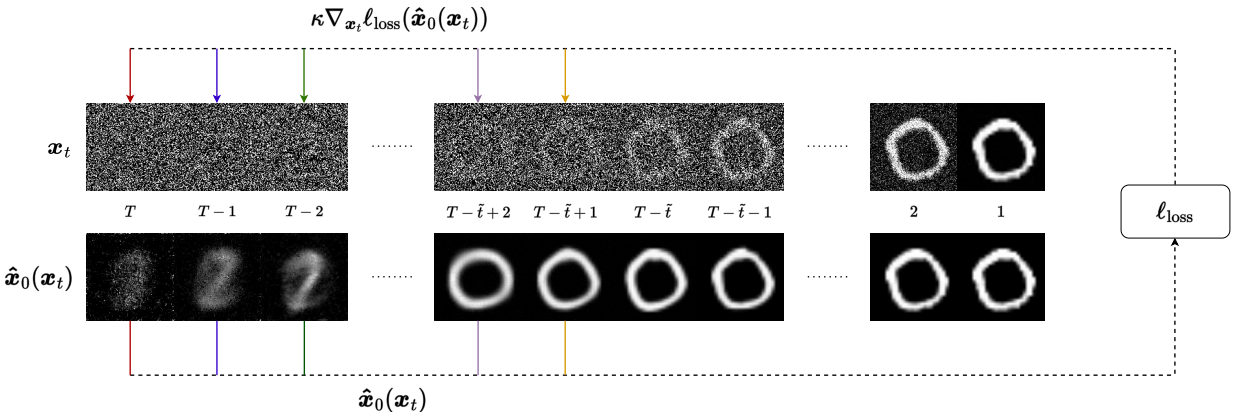

Figure 1: Gradient-guided results on MNIST [11] class zero. The first row contains the images $\boldsymbol{x}_t$ at each backward diffusion timestep t, and the second row contains the predicted denoised output $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ from each $\boldsymbol{x}_t$.

3 Method

3.1 Denoising Diffusion Probabilistic models (DDPM)

The mean of the reverse process is defined in Eq (1), where $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is our estimate of the noise $\boldsymbol{\epsilon}$ added to $\boldsymbol{x}_0$ to reach timestep $\boldsymbol{x}_t$:

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\Big). \tag{1}$$

$\boldsymbol{x}_{t-1}$ can be sampled from $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x})$ using the reparamaterization trick:

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) + \sigma_t \boldsymbol{\epsilon} = \frac{1}{\sqrt{\alpha_t}}\Big(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\Big) + \sigma_t \boldsymbol{z}. \tag{2}$$

This formulation lends itself to an evidence lower bound we want to minimize:

$$L(\theta) := \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}}\Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\Big]. \tag{3}$$

For rudimentary information about diffusion models, see Appendix Details of Diffusion Models.

3.2 Gradient-Guided Diffusion

Algorithm 1 Gradient Guided Diffusion Model

Input: total denoising steps T, loss function ℓ, scale κ, conditioning duration $\tilde{t}$
Output: Generate image $\boldsymbol{x}_0$

1: $\boldsymbol{x}_T \sim \mathcal{N}(0, \mathbf{I})$
2: **for all** t from T to 1 **do**
3: $\quad \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t) \leftarrow \frac{\boldsymbol{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$
4: $\quad \boldsymbol{x}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\Big(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\Big) + \sigma_t \boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I})$
5: $\quad$ **if** $t > (T - \tilde{t})$ **then**
6: $\quad\quad \boldsymbol{x}_{t-1} \leftarrow \boldsymbol{x}_{t-1} - \kappa \cdot \nabla_{\boldsymbol{x}_t} \ell(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))$
7: $\quad$ **end if**
8: **end for**
9: **return** $\boldsymbol{x}_0$

In [9][10], the authors used Tweedie's formula to compute an estimate of the final denoised $\boldsymbol{x}_0$ from any arbitrary step $\boldsymbol{x}_t$, denoted as $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$:

$$\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t) := \frac{\boldsymbol{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t). \tag{4}$$

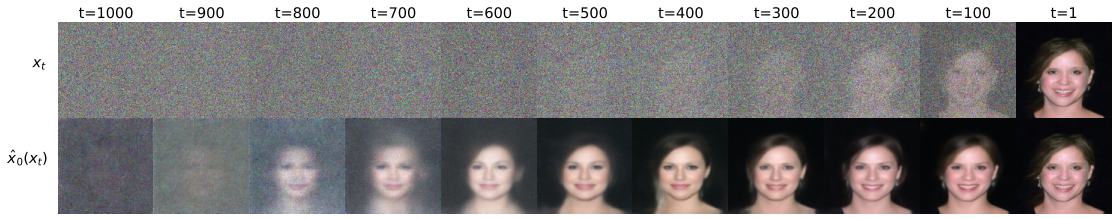

Figure 2: Visualization of backward diffusion on the dataset CelebA [12]. At the top are the intermediate images $\boldsymbol{x}_t$ by diffusion; at the bottom are the estimated final denoised samples $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ calculated by Tweedie's formula [9][10].

Using $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$, we can get a sense of the diffusion model's final output, as shown in Figure 2. We can use this to control the diffusion trajectory with a differentiable loss $\nabla_{\boldsymbol{x}_t}\ell(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))$:

$$\boldsymbol{x}_{t-1} = \left(\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \sigma_t \boldsymbol{z}\right) - \kappa \cdot \nabla_{\boldsymbol{x}_t}\ell(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)). \tag{5}$$

Our algorithm is described in Algorithm 1, where κ is a hyperparamter used to control how much the gradient affects our diffusion process at each timestep t, similar to a learning rate. Additionally, the conditioning duration $\tilde{t}$ refers to how many timesteps we use Eq (5) to compute $\boldsymbol{x}_{t-1}$ as opposed to Eq (2).

The loss function $\ell(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))$ can take any form, and can be a linear combination of multiple losses, as long as they are differentiable. This brings additional versatility in controlling the diffusion process.

This is related to [2], where they used classifier guidance at each diffusion step:

$$\hat{\boldsymbol{\epsilon}}\left(\boldsymbol{x}_t\right) := \boldsymbol{\epsilon}_\theta\left(\boldsymbol{x}_t\right) - \sqrt{1-\bar{\alpha}_t}\nabla_{\boldsymbol{x}_t}\log p_\phi\left(y \mid \boldsymbol{x}_t\right). \tag{6}$$

However, their approach requires computing the gradient $\nabla_{\boldsymbol{x}_t}\log p_\phi\left(y \mid \boldsymbol{x}_t\right)$, which can be noisy if t is close to T. This is because each intermediate image $\boldsymbol{x}_t$ for each timestep t close to T is a very noisy version of $\boldsymbol{x}_0$. The gradient-guided method presented in [10] alleviates this problem by computing $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$, which is a lot less noisy as shown in Figure 2.

4 Experiments

In this section, we present the experimental results on Classifier-Gradient-Guided Diffusion on the dataset MNIST [11] and CLIP-Gradient-Guided Diffusion on the dataset CelebA [12].

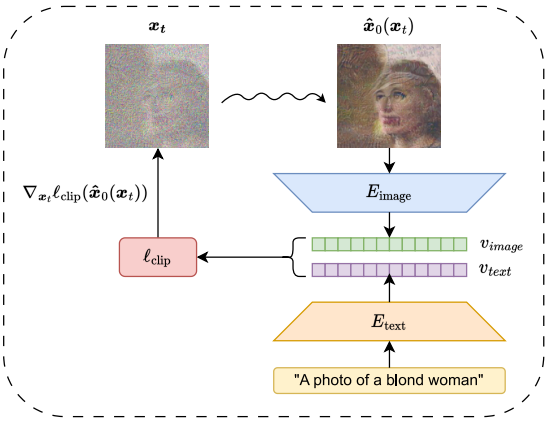

Figure 3: (left) An illustration of how we use CLIP to guide the diffusion process. (right) Generated images for different prompts. Prompts at each row: (1) "A photo of blonde woman" (2) "A photo of a man with black hair" (3) "A photo of a smiling woman" (4) "A photo of a bald man". We use $\kappa = 10$ and $\tilde{t} = 1000$ in our experiments.

4.1 Classifier-Gradient-Guided Diffusion on MNIST

Description of loss function We expect that classifiers provide a strong discriminative signal for what an image of a given class $\hat{y}$ "should" look like. High softmax scores should be given to images that perceptually match $\hat{y}$. Using this idea, the loss can be written as $\ell_{\text{loss}}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)) = -\log P(y = \hat{y} \mid \hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))$, which is computed through a softmax classifier trained on MNIST with heavy data augmentation. A visualization of this method is provided in Figure 1.

Effect of κ and $\tilde{t}$ For each value of κ and $\tilde{t}$, we generate 200 images of class 0 using the method in Algorithm 1, and get the accuracy on this set using a pretrained classfier, as shown in Figure 4. When $\kappa = 0$, it means we are not using the gradient guidance (so simply using Eq (2)), and the number of classified zeros is around 10%. For $\kappa > 0$, we can see that the accuracy increases with $\tilde{t}$, and roughly converges at step 500. Additionally, with enough backward diffusion steps, a higher value of κ can bring the final generated image closer to the desired class $\hat{y}$. This is most likely due to a stronger gradient signal, which has a significant influence on the diffusion process.

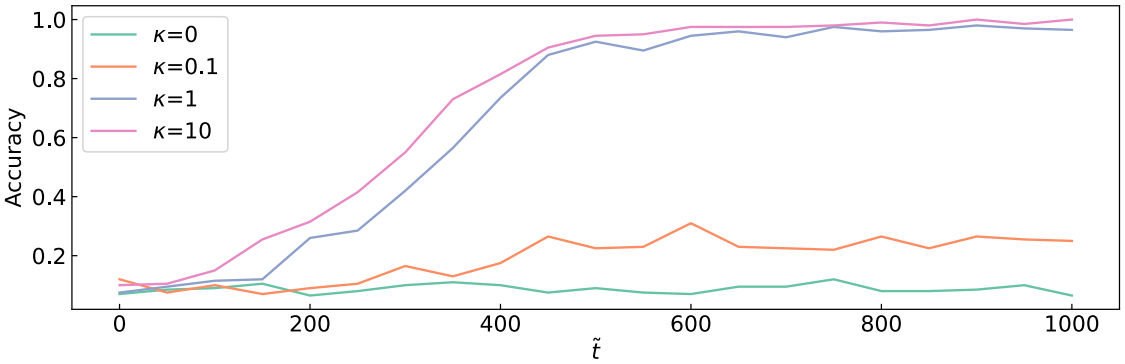

Figure 4: Gradient-Guided Results on class zero of MNIST [11] with different scales κ and gradient update steps $\tilde{t}$.

4.2 CLIP-Gradient-Guided Diffusion on CelebA

Description of loss function We combine diffusion with CLIP [4] to control the diffusion process via a text prompt. In order to do this, we run the generated image estimate $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ through CLIP with a pre-trained Vision Transformer [3] and our fixed prompt through an encoder at each time step. The loss is defined using cosine similarity:

$$\ell_{\text{clip}}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)) = -\frac{E_{\text{text}}(\text{prompt}) \cdot E_{\text{image}}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))}{||E_{\text{text}}(\text{prompt})|| \cdot ||E_{\text{image}}(\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))||}, \tag{7}$$

where E denotes encoder. Classifier-Gradient-Guided Diffusion is used with the CLIP loss ℓ_{clip}, as described in Algorithm 1. The image encoder we use is the ViT-B/32 [4] model and the text encoder was the default text encoder used in [4]. A visualization of the method and the generated results from different prompts is shown in Figure 3. As evidenced from the figure, the diffusion process reaches images that are somewhat semantically close to the prompt but with lots of artifacts. A discussion of why these artifacts occur is provided in the Appendix CLIP Artifacts.

5 Conclusion

In this work, we explore a method of conditioning diffusion models via a differentiable loss function. While we examine only a few losses in this paper, this method can be generalized to any differentiable loss such as perceptual losses[8][15], and even their linear combinations. This would allow for image synthesis diffusion model outputs to be highly customized toward user preferences.

One limitation of our method is that it requires the gradient of the loss function to be computed at every conditioning timestep (e.g., backpropagating through the Vision Transformer for CLIP), which can be very expensive. Another limitation is that the denoised estimate $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t)$ can be noisy near the start of the diffusion process, as shown in Figure 2, which can result in noisy gradients.

Contribution

Junru Lin mainly worked on the experiment for MNIST and helped with CLIP. Also worked on the visualizations, as well as most parts of the final report.

Aditya Mehrotra worked on literature review and formalizing the framework for doing conditioning. Helped with coding experiments for both MNIST and CLIP. Also worked on all parts of the final report.

Neil Gupta worked on finding correct CLIP model and combining its results with the diffusion model. Helped with running experiments with CLIP. Worked on the CLIP parts of the final report.

Acknowledgement

The authors are grateful to the teaching team of CSC413 for their guidance and support throughout the entire process. The authors also want to thanks reviewers in advance for their time and effort in providing feedback and insights on this report.

References

[1] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2023. doi: 10.1109/TPAMI.2023.3261988.

[2] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis, 2021.

[3] A. D. et. al. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[4] A. R. et. al. Learning transferable visual models from natural language supervision. 2021.

[5] S. gil Lee, H. Kim, C. Shin, X. Tan, C. Liu, Q. Meng, T. Qin, W. Chen, S. Yoon, and T.-Y. Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior, 2022.

[6] J. Ho and T. Salimans. Classifier-free diffusion guidance, 2022.

[7] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. URL `https://arxiv.org/abs/2006.11239`.

[8] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

[9] K. Kim and J. C. Ye. Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874, 2021.

[10] G. Kwon and J. C. Ye. Diffusion-based image translation using disentangled style and content representation, 2023.

[11] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[12] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[13] C. Luo. Understanding diffusion models: A unified perspective, 2022. URL `https://arxiv.org/abs/2208.11970`.

[14] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022.

[15] G. G. Pihlgren, K. Nikolaidou, P. C. Chhipa, N. Abid, R. Saini, F. Sandin, and M. Liwicki. A systematic performance analysis of deep perceptual loss networks breaks transfer learning conventions, 2023.

[16] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi. Palette: Image-to-image diffusion models, 2022.

[17] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

[18] I. Schwartz, V. Snæbjarnarson, S. Benaim, H. Chefer, R. Cotterell, L. Wolf, and S. Belongie. Discriminative class tokens for text-to-image diffusion models, 2023.

[19] Y. Tashiro, J. Song, Y. Song, and S. Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation, 2021.

[20] V. Voleti, A. Jolicoeur-Martineau, and C. Pal. Mcvd: Masked conditional video diffusion for prediction, generation, and interpolation, 2022.

[21] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen. Pretraining is all you need for image-to-image translation, 2022.

[22] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications, 2023.

[23] L. Zbinden, L. Doorenbos, T. Pissas, R. Sznitman, and P. Márquez-Neila. Stochastic segmentation with conditional categorical diffusion models, 2023.

Appendix

Details of Diffusion Models

Diffusion models are a class of probabilistic models that map noise sampled from a standard normal distribution $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$ to the data distribution $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x})$ over a series of timesteps $1, \ldots, T$.

The distribution of the final timestep conditioned on the original image, as well as the forward diffusion distribution, is given as

$$q(\boldsymbol{x}_T|\boldsymbol{x}_0) := \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \quad \text{where} \quad q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \tag{8}$$

where $\{\beta\}_{t=0}^{T}$ is a variance schedule.

For any timestep t, we can compute the latent variable x_t given an image x_0 in a differentiable manner using the reparameterization trick:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}). \tag{9}$$

We represent the backward diffusion process as a gaussian, which lets us compute previous timesteps conditioned on future timesteps, in order to reverse noise into a sample from the true data distribution. The variance of this Gaussian is fixed, and the mean is parameterized by a neural network.

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \sigma_t^2 \boldsymbol{I}). \tag{10}$$

CLIP Artifacts

As shown in figure 3, the images are somewhat similar to the prompt but have artifacts that ruin the perceptual quality of the image. We suspect that aligning the embeddings for the estimated images $\hat{\boldsymbol{x}}_0(\boldsymbol{x}_t))$ with the text embeddings does not enforce the generated image to be perceptually clean, but only preserves the semantic information.