Training-free Aligning Diffusion Models with Probability-Regularized Noise Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

1 Introduction

- Introduction on Diffuison Models.
- Drawback of Diffusion Models -> not align with the goal -> Align with reward
- Related work: LLM Alignment, RL-based, Guidance-based, noise/latent optimization-based.
- Contribution in this work:
  - We focus noise optimization-based method. Rigorously study the limit distribution and per-step improvement.
  - A key issue in noise optimization is reward-hacking. We reveal that when the noise drives away from standard Gaussian, reward-hacking happens. Thus we propose a probability-regularized noise optimization method to force the noise stay within the high probability region of the standard Gaussian.
  - We also extend to optimizing non-differentiable reward functions by revealing the best way of gradient approximation.
  - We conduct experiment on several popular reward function, and show that noise optimization can achieve sota scores without reward hacking and any training, albeit can with the cost of more time at generation.
  - We remark that the optimized images can serve as dataset for finetuning in the future.

1.1 Diffusion Models

In recent years, diffusion models have been recognized as state-of-the-art for generating high-quality images, demonstrating exceptional resolution, fidelity, and diversity [Ho et al., 2020, Dhariwal and Nichol, 2021, Song et al., 2020b]. These models are also notably easy to train and can be effectively extended to conditional generation [Ho and Salimans, 2022]. Broadly speaking, diffusion models work by learning to reverse the diffusion of data into noise, a process that can be described by a stochastic differential equation (SDE) [Song et al., 2020b, Karras et al., 2022]:

$$dx_t = f(t)x_t dt + g(t)dw_t, \tag{1}$$

where dw_t is the standard Wiener process, and $f(t)$ and $g(t)$ are the drift and diffusion coefficients, respectively. The reverse process relies on the score function $\epsilon(x_t, t) \overset{\text{def.}}{=} \nabla_x \log p(x_t)$, and its closed form can be expressed either as an ordinary differential equation (ODE) [Song et al., 2020b]:

$$dx_t = \left(f(t)x_t - \frac{1}{2}g^2(t)\epsilon(x_t, t) \right) dt, \tag{2}$$

or as an SDE:

$$dx_t = \left(f(t)x_t - g^2(t)\epsilon(x_t, t)\right) dt + g(t)dw_t. \tag{3}$$

With the ability to evaluate $\epsilon(x_t, t)$, it becomes possible to generate samples from noise by numerically solving the ODE (2) or the SDE (3). The training process, therefore, involves learning a parameterized surrogate $\epsilon_\theta(x_t, t)$ for $\epsilon(x_t, t)$ following a denoising score matching framework described in [Song et al., 2020b, Karras et al., 2022].

1.1.1 Aligning Generative Models

Generative models, e.g., Large Language Models (LLM) and Diffusion models, are trained on large-scale unlabeled and mixed-quality data, with an aim to capture the distribution of the data. However, when deploying these generative models for specific tasks, it is not good to sample from the orginally learned distribution directly, as this distribution is not aligned with the task-specific goal. For example, in image generation, people wishes to generated images that is aesthetically pleasing, not just a random image. Therefore, before putting the generative models into use, it is necessary to align the generative models with the task-specific goal.

Aligning Generative models has become a heated topic since the sucesss of alignming LLMs with Reinforcement Learning from Human Feeback (RLHF) [Ouyang et al., 2022, Liu et al., 2023a, OpenAI, 2022, Bai et al., 2022], where human evaluators are asked to rank the outputs of large AI models according to their personal preferences, with an aim to improve the generation quality of these models.

For diffusion models, RLHF has also gained sufficient attention recently, as the very recent series of works like [Deng et al., 2024, Uehara et al., 2024, Yuan et al., 2024, Song et al., 2023a, Dong et al., 2023, Sun et al., 2023, Zhang et al., 2023, Prabhudesai et al., 2023, Black et al., 2023, Fan et al., 2023].

Broadly speaking, given a generative models θ and its corresponding distribution $p_\theta(x)$. The most common setting in the alignment problem is that there exists a reward model $r(x)$, which can evaluate the generated samples with a real-valued score. Such reward models can either be obtained by training a neural network with human feedback, or using some pre-defined metrics. The goal of alignment is to improve the distribution $p_\theta(x)$, namely, the generated sample should have as high reward as possible. A common mathematical formulation in RLHF is to maximize the expected reward, while the obtained distribution should not deviate too much from the original distribution. This can be formulated as the following KL-regularized optimization problem:

$$\max_p \mathbb{E}_{x\sim p(x)} r(x) - \lambda KL(p||p_\theta). \tag{4}$$

existing models require training for every new reward model

1.1.2 Aligning Diffusion Models with Noise Optimization

In this work, we specifically focus on aligning diffusion models, and propose to align Diffusion Models by directly optimizing the latent noise.

Spefically, as we have introduced above, sampling from diffusion boils down to solving the ODE (2) or SDE (3). When solving for the ODE (2), the solving process can be viewed a mapping that maps the initial noise x_T into the generated sample x_0. Similarly, when solving for the SDE (3), the solving process can be viewed as a mapping that maps both the noise x_T and the Wiener process w_t into the generated sample x_0. With this observation, we can regard the solving process as a mapping $M(z)$ that maps the noise vectors z drawn from the standard Gaussian distribution to the generated samples.

As one can see, since the generated samples are determined by the noise vectors, it is to treat the noise vectors as the optimization variable, and optimize the noise vectors directly to maximize the reward. Mathematically, the optimization problem can be formulated as:

$$\max_z r((M(z))). \tag{5}$$

By solving this optimization problem, we can obtain the optimized noise vectors, and then use the optimized noise vectors to generate the optimized samples. Consider that $r(\cdot)$ is differentiable, we can use gradient-based optimization methods to solve this optimization problem.

Algorithm 1 Direct Noise Optimization with Differentiable Reward

Require: Initial noise z_0, stepsize η, number of iterations T.
for $t = 1$ to T **do**
  $z_t = z_{t-1} + \eta \nabla r(M(z_{t-1}))$.
end for
return Optimized sample $M(z_T)$.

1.1.3 Aligning Generative Models via prompt optimization

There are also some works that align the generative models by optimizing the prompt. For example, in the work of , the authors propose to optimize the prompt of LLMs to achieve the goal of the task. The prompt optimization can be viewed as a special case of the noise optimization, where the noise is the prompt. As we will show some example in Appendix.

2 Understanding Noise Optimization

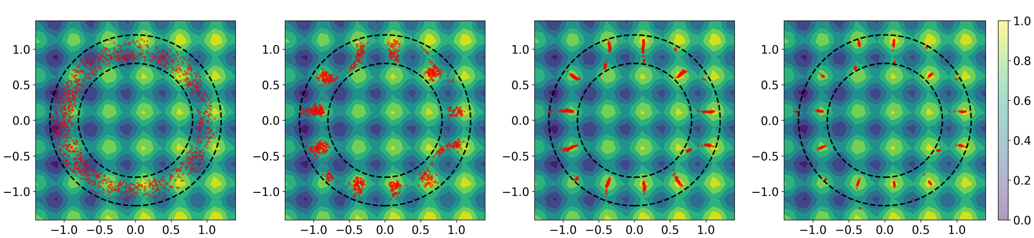
- Use toy example to show the evolving of distribution.
- Show that the noise-to-image mapping is continuous and smooth, using DDIM and DDPM as example.
- Theoretical results, limit distribution, per-step improvement.

a

We remark that directly optimizing the noise vectors can be viewed as sampling from an improved distribution. Spefically, let us denote that $g_t(z)$ as an mapping that define the procedure of optimizing z with t gradient steps. For example, with the z_0 and z_T in Algorithm 1, we have $g_T(z_0) = z_T$. We denote $p_t(x)$ as the distribution of the generated samples after t steps of optimization, namely, $x \sim p_t(x)$ if $x = M(g_t(z))$ and $z \sim \mathcal{N}(0, I)$.

Figure: Toy example of the evolving of distribution. Showing that the point are moving towards the closest local minimum.

Let us first present a simple example while the distribution is the uniform distribution on the ring whose radius is between 0.8 and 1.2. We run the optimization with the reward function $r(x) = \sin(4\pi x[1]) + \sin(4\pi x[2]) - \left((x[1]-1)^2 - x[2]^2\right)/5$, which is a highly nonconvex function with many local minima, maxima and saddle points. We can see that the distribution of the samples are moving towards the local maxima of the reward function.



(a) Initial (b) 10 steps (c) 50 steps (d) 100 steps

Figure 1: evolving of distribution

Now let us states some assumptions before analyzing the noise optimization.

Assumption 1. *Throughout this paper, we have these assumptions:*

1. *For some constant B, $r(x) \geq B$ for all x.*

2. *$r \circ M$ is L-smooth, meaning that $\|\nabla^2 r \circ M(z)\| \leq L$.*

shows that the noise-to-image mapping is continuous and smooth, especially for continuous distribution. DDIM and DDPM, can just cite [Tang et al., 2024a]

Suppose that we choose the stepsize η in Algorithm 1 such that $\eta \leq 1/L$. We can analytically quantified the improvement of the distribution after each step of optimization. Spefically, we can use the expected reward as the metric to measure the improvement.

$$\mathbb{E}_{x \sim p_{t+1}(x)} r(x) \geq \mathbb{E}_{x \sim p_t(x)} r(x) + \frac{1}{2L} \mathbb{E}_{z_0 \sim N(0,I)} \left\| \nabla_z r \circ M(z)_{|z=g^i(z_0)} \right\|_2^2 > \mathbb{E}_{x \sim p_t(x)} r(x).$$

When does the distribution stop improving? As we can see, the distribution will stop improving when the expected gradient norm is zero. some discussion here

Different initialization leads to different samples. As we can see that the optimization process is sensitive to the initialization. For example, in the toy example, we can see that different initialization will leads to different images. This is also true for image diffusion models.

3 Understaning Reward-Hacking in Noise Optimization

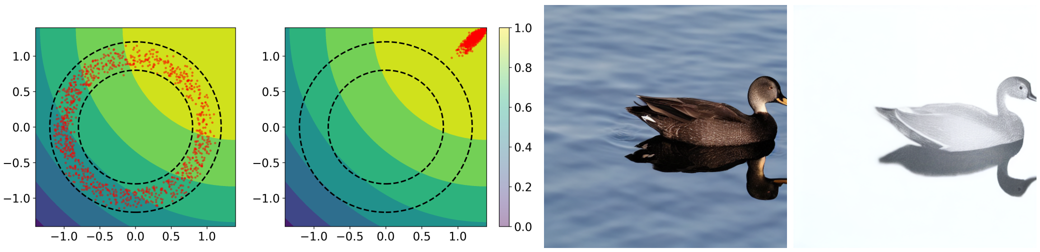
- Definition of reward-hacking: The reward is high but the generated images are out-of-distribution.
- Show examples of reward-hacking: Toy example, SD v1.5 with white reward and black prompt.
- Understand the reason of reward-hacking - Claim: Diffuison models generated out-of-manifold samples if and if only the initial noise is rare sample. The noise drives away from standard Gaussian during optimization.
- Use two concentration inequalities to measure the probability of sampled noise. Claim: The probability can be used to measure the degree of out-of-distribution.
- Four empricial evidence to support the claim.
  - Toy example: Show the noise-to-sample mapping. Use color to indicate the probability of sampled noise.
  - Set the noise to rare sample, e.g., all zero, all equal, etc. Bad images happen
  - Show some empirical investigation: Toy example, SD v1.5 with white reward. Compute the probability of sampled noise. When the probability of sampled noise is low, the reward-hacking happens.
  - For manifold-preserved reward, reward-hacking is less likely to happen. This is reflected in the probability of sampled noise.
- A plausible explination of why rare sample lead to out-of-distribution: The NN have high approximation error on rare samples.
- Formulate the probability-regularized noise optimization problem.

While noise optimization is an effective way to align diffusion models with the reward, sometimes it may have the issue of reward-hacking. Spefically, reward-hacking means that the optimized samples are no longer in the support of pretrained distribution $p_\theta(x)$. In the example of image diffusion models, reward-hacking may lead to the generation of unnatrual images.

Definition 1 (Reward-Hacking)**.** *We say that the noise optimization has the issue of reward-hacking if the expected reward is high, but the generated samples are out-of-distribution.*

Figure: Toy example of reward-hacking. SD v1.5 with white reward and black duck prompt. Before optimization and after optimization.

In this example, we present two examples using both the simple diffusion models in the toy example and the open-sourced image diffusion model SD v1.5. For the toy example, we optimize the reward fucntion $r(x) = -(x[1] - 1.4)^2 - (x[2] - 1.4)^2$ with 1000 gradient steps. For the SD v1.5, we optimize the whiteness reward, i.e., for an image $M \in [0, 255]^{3\times H\times W}$, we define the reward as $r(M) = -\sum_{i=1}^{3}\sum_{j=1}^{H}\sum_{k=1}^{W} M[i, j, k]$, and do the optimization with 30 steps. For both example, we use Adam optimizer with learning rate $1e-2$.



(a) Initial (b) 1000 steps (c) Black Duck (d) 50 steps

Figure 2: examples of reward-hacking

We identify that the main reason of reward-hacking is that optimized noise vectors becomes rare samples of standard Gaussian distribution. By saying rare samples, we mean that it is of extremely low probability to sample such noise vectors from the standard Gaussian distribution.

Since the diffusion models are trained on the standard Gaussian noise, when the noise vectors are rare samples, the neural network of diffusion models could have large approximation error on these rare samples. This leads to the generation of out-of-manifold samples.

3.1 Quantifying OOD via Concentration Inequalities

Therefore, to prevent reward-hacking, we propose to regularize the noise vectors to stay within the high probability region of the standard Gaussian distribution. We can achieve this by adding a regularization term to the optimization problem.

The regularization terms are designed with the inspiration from the classical concentration inequality for Gaussian Distribution. Specifically, for a sample $z \sim \mathcal{N}(0, I_n)$, we group it into m subvectors: $z = [z_1^1, ..., z_m^k], n = m \cdot k,\ z_i = [z_i^1, ..., z_i^k] \sim \mathcal{N}(0, I_k)$. $z = [z_1^1, ..., z_m^k], n = m \cdot k,\ z_i = [z_i^1, ..., z_i^k] \sim \mathcal{N}(0, I_k)$. We use the following two concentration inequalities for the mean and covariance matrix:

$$\Pr\left[\left\|\frac{1}{m}\sum_{i=1}^{m} z_i\right\| > M\right] < p_1(M) \overset{\text{def.}}{=} \max\left\{2e^{-\frac{mM^2}{2k}}, 1\right\} \tag{6}$$

$$\Pr\left[\left\|\frac{1}{m}\sum_{i=1}^{m} z_i z_i^\top - I_k\right\| > M\right] < p_2(M) \overset{\text{def.}}{=} \max\left\{2e^{-\frac{m\left(\max\left\{\sqrt{1+M}-1-\sqrt{\frac{k}{m}}, 0\right\}\right)^2}{2}}, 1\right\} \tag{7}$$

Remark 1. *According to [Wainwright, 2019], the two inequalities are tight when m/k is large. On the other hand, using a large k allows use to examine the covariance of different subvectors in the noise vector. For example, using $k = 1$ is a bad choice, as it only examines the mean and variance the noise vector, but not the covariance of different subvectors. In practice, we found that choosing k to be around $m/100$ to be an empirical good choice. In Appendix, we also provide a grid search for guiding the choice of k.*

A important thing to note is that the two inequalities are invariant to the permutation of the subvectors, i.e., for any permutation matrix Π, if z follows standard Gaussian probability, the permuted vector Πz will have the same probability behavior. This is due to the permutation-invariant property of the standard Gaussian distribution. With this observation, a natural idea is to examine the probability of many permuted vectors, i.e., given b permutation matrices $\Pi_1, ..., \Pi_b$, we wish to examine whether the all the vectors $\Pi_1 z, ..., \Pi_b z$ lies in the high probability region of the standard Gaussian distribution.

With this two concentration inequalities, we can measure the plausiblity of the sampled noise vectors. Specifically, given b permutation matrices $\Pi_1, ..., \Pi_b$, we can define the probability of the sampled noise vectors as: We denote $M_1(z) = \left\|\frac{1}{m}\sum_{i=1}^{m} z_i\right\|$ and $M_2(z) = \left\|\frac{1}{m}\sum_{i=1}^{m} z_i z_i^\top - I_k\right\|$

$$P(z) \stackrel{\text{def.}}{=} \min\left\{\min_{i\in\{1,\ldots,b\}} p_1\left(M_1(\Pi_i z)\right), \min_{i\in\{1,\ldots,b\}} p_2\left(M_2(\Pi_i z)\right)\right\}. \tag{8}$$

If the probability $P(z)$ is low, it means that there is a permutation matrix Π_i such that the noise vector $\Pi_i z$ lies in the low probability region of the standard Gaussian distribution, therefore, the noise vector z is less likely to be sampled from the standard Gaussian distribution.

Remark 2. *For the numbers of permutation matrices b, we can set it to be $b = 100$ throughout this work, and the used permutation matrices are randomly generated. Using a small b may make it less capable in locating the out-of-distribution case. But we will show that the probability $P(z)$ is robust to the choice of b as long as it is sufficiently large. More details can be found in Appendix.*

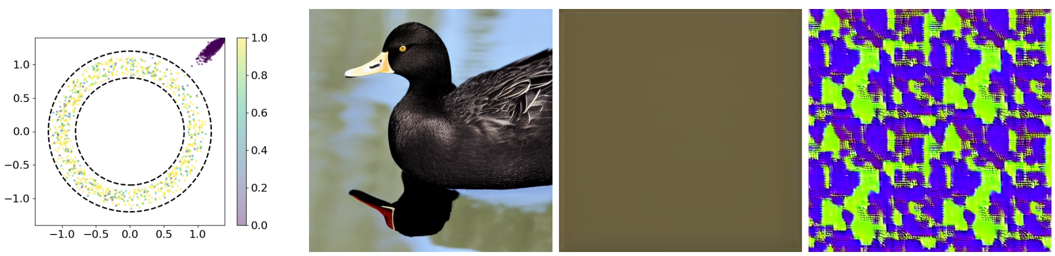
Figure: Four empricial evidence to support the claim.

Figure 1: One image, with color indicating the probability of sampled noise.

Figure 2: SD v1.5, set the noise to all zero. set part of the noise to be repetitive

Figure 3: Show the optimization trajectory of optimizing white reward, y-axis is the prob. prob is decreasiing

Figure 4: Show the optimization trajectory of optimizing aesthetic reward, y-axis is the prob. prob is not decreasiing



(a) Toy example (b) $P(z) = 0.36$ (c) Zero: $P(z) = 0.00$ (d) Repeat: $P(z) = 0.02$

Figure 3: examples of reward-hacking

Remark 3. *A possible reason that rare samples lead to out-of-distribution is that the neural network have high approximation error on rare samples.*

3.2 Probability-Regularized Noise Optimization

With these two inequalities, we propose the following probability-regularized noise optimization problem:

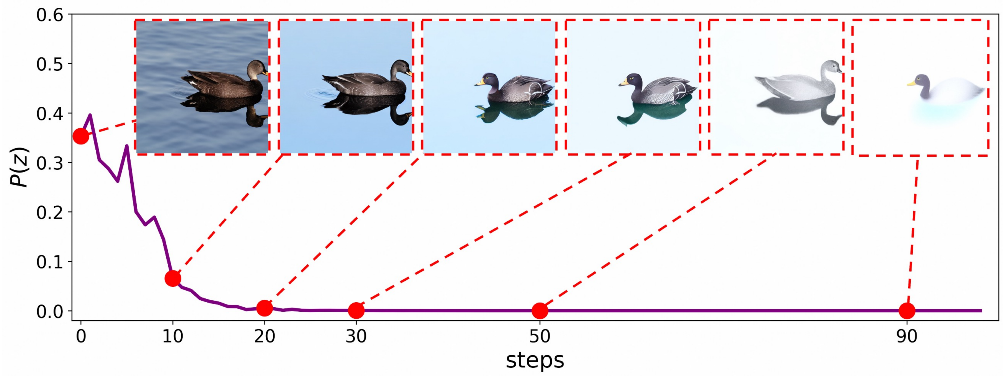
$$\max_z r((M(z))) + \mu \mathbb{E}_\Pi \left[\log p_1\left(M_1(\Pi z)\right) + \log p_2\left(M_2(\Pi z)\right)\right]. \tag{9}$$

In particular, for the regularization term, we use the expectation of the probability over the permutation matrices, rather than the minimum probability $P(z)$, this is because the expectation is more smooth for optimization.

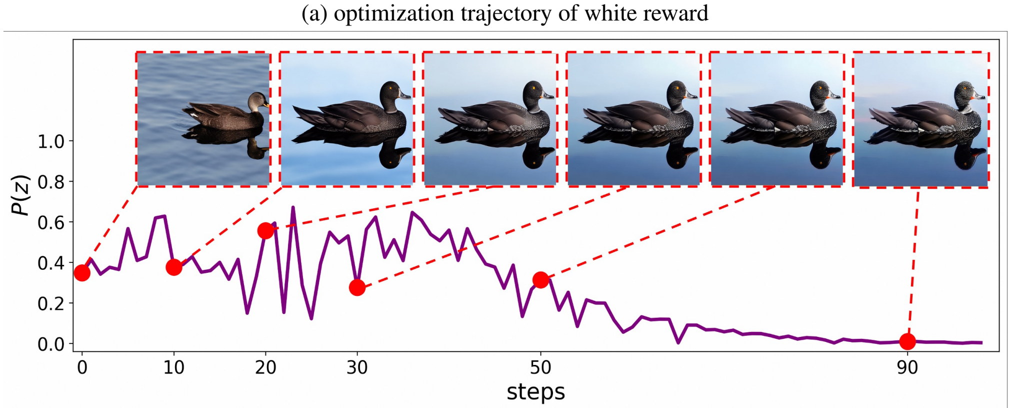
This can be similarly tackled by the gradient-based optimization algorithm.

4 Tackling Non-Differentiable Reward Functions

- In many case, we may face non-differential reward: 1. The metric, by design, is non-differentiable. 2. The reward is a third-party black-box. 3. The reward is represented by a huge neural network, thus the gradient is hard to compute.
- We study three way to extend noise optimization to non-differentiable reward functions, and identified the optimal one.



(a) optimization trajectory of white reward



(b) optimization trajectory of aesthetic reward

Figure 4: optimization trajectory

We remark that the noise optimization can also be extended to non-differentiable reward. Specifically, for the problem $\max_z r(M(x))$ with r is non-differentiable, we can see that the problem is partially differentiable as we can compute the gradient of M. In this case, we can estimate the gradient $\nabla_z r(M(z))$ via

$$\hat{g}(z) = \hat{h}(y) \cdot \nabla_z M(z),$$

where $\hat{h}(y) = (r(y_1) - r(y))(y_1 - y)$ and $y_1 = M(z + \xi), y = M(z)$, ξ is a small and random perturbation vector.

Theorem: provably efficient for the gradient estimation.

5 Experiments

- Major experiment 1: Black animal with whiteness reward, white animal with blackness reward. The main purpose is to prove the effectiveness of probability regularization.
- Major experiment 2: Benchmarking three NN-based reward functions with SD 1.5: aesthetic, pick score, hps. For these reward, it is less likely to hack them. But the regularization can also be helpful. When optimize one of them, use the other two as test metrics. Show that the score is superior to the one obtained by directly optimizing the image.
- Comparing three ways of non-differentiable optimization. Use Jpeg and aesthetic score. The purpose is to show that the optimal way is the one we proposed.
- Demonstrative experiments: 1. Use multi-modal LLM as reward function. show that it is less likely to hack it 2. use SDXL as base model for optimization. describe the computing cost. using complex prompt
- Experiments in Appendix:

- Tuning the regularization parameter.
- Visualization of optimization with images.
- finetuning on optimized images.
- Demonstrate the faliure of some common metrics. CLIP score, mlm-filter and some other metrics on the black and white example.
- more demonstartive examples

5.1 Examine the Effectiveness of Probability Regularization

Discuss the metrics used, the top-k mean of pixels value, because the classical metrics fails. detiails in appendix.

Figure: y-axis: value of the metric, probability, over-optimize metric. two legends: with and without regularization.

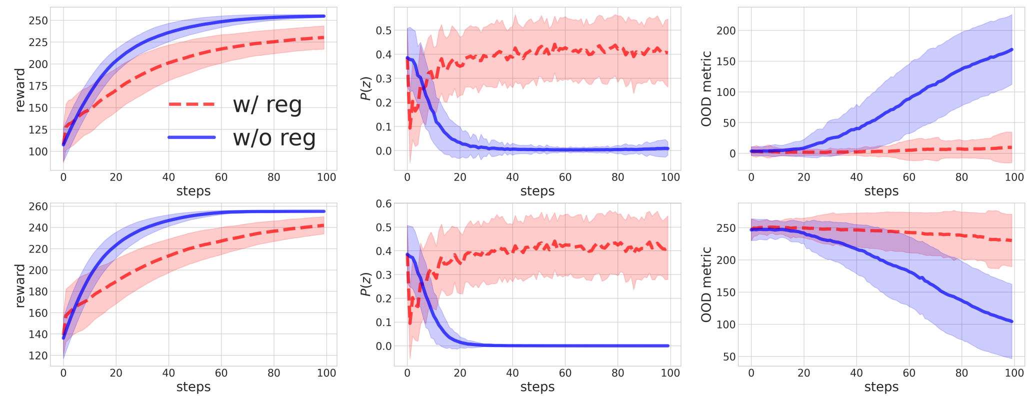


Figure 5: The effectiveness of probability regularization.

example images: white and black, with and without regularization.

(a) White w/o reg (b) White w/ reg (c) Black w/o reg (d) Black w/ reg

Figure 6: examples of optimized image with and without regularization

main conclusion: similar to the four empirical evidence mentioned above. the probability can reflect the level of out-of-distribution. penalizing the probability can prevent reward-hacking.

for the algorithm without regularization, we use the leanring rate 0.001, while for with regularization. we use the learning rate 0.01.

Remark 4. *We provide hyperparameters analysis for the regularization coefficients in Appendix.*

we run the experiment with 1000 samples, with the randomly generated animial prompt.

5.2 Benchmarking Three NN-based Reward Functions

Figure: y-axis: three metrics, probability. six legends: optimize three metrics * with/without regularization.

Table: comparing the score of the three metrics to existing works. for our methods, report the scores under time/steps.

we run the experiment with 1000 samples, with the randomly generated animial prompt.

main conclusion: optimizing these reward functions trained on human feedback is less likely to lead to reward-hacking, but not impossible. The regularization can also be helpful. This can be shown by using the other two as test metrics. Noise optimization can easily beat sota RL methods without any training on the reward function. Albeit the cost of more time at generation.

Remark 5. *we provide the Visualization of optimization in appendix.*

Remark 6. *To explore a possible way to alleviate the sampling cost, we also show that it works well by finetuning the model on optimized images in appendix.*

5.3 Comparing Three Ways of Non-Differentiable Optimization

main conclusion: the optimal way is the one we proposed. note that existing RL like DDPO consider only non-differentiable reward functions, we show that this case can also be handled by our method. Using estimated gradient is slower than using the true gradient, but it is still faster than RL methods.

Figure: Jpeg and aesthetic score. comparing the three ways.

5.4 Demonstrative Experiments

Figure: 1. Optimizing SDXL w.r.t aesthetic score, trajectory. 2. Optimizing SD v1.5 w.r.t multi-modal LLM, trajectory.

We believe our method have the potential in many real-world applications, such as the ID-preserving tasks in [Chen et al., 2024, Guo et al., 2024].

6 Conclusions

- Noise optimization is an effective way to align diffusion models, but be cautious about reward-hacking. By preventing the noise from being rare, we can alleviate the reward-hacking.
- It is training-free, but requires more time to generate images.
- A possible way to alleviate the time cost is to use the generated images as dataset for finetuning.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. *arXiv preprint arXiv:2402.14017*, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Weifeng Chen, Jiacheng Zhang, Jie Wu, Hefeng Wu, Xuefeng Xiao, and Liang Lin. Id-aligner: Enhancing identity-preserving text-to-image generation with reward feedback learning, 2024.

Fei Deng, Qifei Wang, Wei Wei, Matthias Grundmann, and Tingbo Hou. Prdp: Proximal reward difference prediction for large-scale reward finetuning of diffusion models. *arXiv preprint arXiv:2402.08714*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023.

Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. *arXiv preprint arXiv:2401.08639*, 2023.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling, 2023.

Zinan Guo, Yanze Wu, Zhuowei Chen, Lang Chen, and Qian He. Pulid: Pure and lightning id customization via contrastive alignment, 2024.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *arXiv preprint arXiv:2212.09611*, 2022.

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. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Jian Hu, Li Tao, June Yang, and Chandler Zhou. Aligning language models with offline learning from human feedback, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. *arXiv preprint arXiv:2312.11994*, 2023.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation, 2023.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences, 2023.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. Languages are rewards: Hindsight finetuning using human feedback, 2023a. URL `https://arxiv.org/abs/2302.02676`.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. *arXiv preprint arXiv:2309.06380*, 2023b.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models, 2023.

Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J Bryan. Ditto: Diffusion inference-time t-optimization for music generation. *arXiv preprint arXiv:2401.12179*, 2024.

OpenAI. Chatgpt,https://openai.com/ blog/chatgpt/, 2022. URL `https://openai.com/blog/chatgpt/`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pages 32483–32498. PMLR, 2023a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023b.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Jiao Sun, Deqing Fu, Yushi Hu, Su Wang, Royi Rassin, Da-Cheng Juan, Dana Alon, Charles Herrmann, Sjoerd van Steenkiste, Ranjay Krishna, and Cyrus Rashtchian. Dreamsync: Aligning text-to-image generation with image understanding feedback, 2023.

Zhiwei Tang, Dmitry Rybin, and Tsung-Hui Chang. Zeroth-order optimization meets human feedback: Provable learning via ranking oracles. In *The Twelfth International Conference on Learning Representations*, 2024a. URL `https://openreview.net/forum?id=TVDUVpgu9s`.

Zhiwei Tang, Jiasheng Tang, Hao Luo, Fan Wang, and Tsung-Hui Chang. Accelerating parallel sampling of diffusion models. *arXiv preprint arXiv:2402.09970*, 2024b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Sergey Levine, and Tommaso Biancalani. Feedback efficient online fine-tuning of diffusion models. *arXiv preprint arXiv:2402.16359*, 2024.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7280–7290, 2023.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. Openchat: Advancing open-source language models with mixed-quality data, 2023a.

Weizhi Wang, Khalil Mrini, Linjie Yang, Sateesh Kumar, Yu Tian, Xifeng Yan, and Heng Wang. Finetuned multimodal language models are high-quality image-text data filters. *arXiv preprint arXiv:2403.02677*, 2024.

Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models, 2023b.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis, 2023.

Yazhou Xing, Yingqing He, Zeyue Tian, Xintao Wang, and Qifeng Chen. Seeing and hearing: Open-domain visual-audio generation with diffusion latent aligners. *arXiv preprint arXiv:2402.17723*, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement, 2023.

Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *arXiv preprint arXiv:2402.10210*, 2024.

Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. *arXiv preprint arXiv:2303.09618*, 2023.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023.

Appendix

A Proofs

B hyperparameters

B.1 The η in DDIM

Comparing optimizing full noise and only initial noise. optimizing full noise tends to be faster.

B.2 Effect of learning rate on reward-hacking

B.3 m and k

B.4 Numbers of permutation matrices

B.5 Regularization coefficients

hyperparameters of regularization coefficients on white, aesthetic, hps, pick score

C Faliure of in existing metrics in capturing the reward-hacking phenomenon

D Visualization of optimization with images

E More Demonstrative Examples

F Comparing to Prompt tuning

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: **[TODO]**

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.