Self-Play Fine-Tuning of Diffusion Models for Text-to-Image Generation

Huizhuo Yuan* **Zixiang Chen*** **Kaixuan Ji*** **Quanquan Gu**
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90095
{hzyuan,chenzx19,kaixuanji,qgu}@cs.ucla.edu

Abstract

Fine-tuning Diffusion Models remains an underexplored frontier in generative artificial intelligence (GenAI), especially when compared with the remarkable progress made in fine-tuning Large Language Models (LLMs). While cutting-edge diffusion models such as Stable Diffusion (SD) and SDXL rely on supervised fine-tuning, their performance inevitably plateaus after seeing a certain volume of data. Recently, reinforcement learning (RL) has been employed to fine-tune diffusion models with human preference data, but it requires at least two images ("winner" and "loser" images) for each text prompt. In this paper, we introduce an innovative technique called self-play fine-tuning for diffusion models (SPIN-Diffusion), where the diffusion model engages in competition with its earlier versions, facilitating an iterative self-improvement process. Our approach offers an alternative to conventional supervised fine-tuning and RL strategies, significantly improving both model performance and alignment. Our experiments on the Pick-a-Pic dataset reveal that SPIN-Diffusion outperforms the existing supervised fine-tuning method in aspects of human preference alignment and visual appeal right from its first iteration. By the second iteration, it exceeds the performance of RLHF-based methods across all metrics, achieving these results with less data. Codes are available at https://github.com/uclaml/SPIN-Diffusion/.

1 Introduction

Diffusion models (Ho et al., 2020; Peebles and Xie, 2023; Podell et al., 2023; Nichol et al., 2021; Rombach et al., 2022a; Song et al., 2020a) have rapidly emerged as critical entities within the realm of generative AIs (Creswell et al., 2018; Kingma and Welling, 2013), demonstrating exceptional capabilities in generating high-fidelity outputs. Their versatility spans a diverse area of applications, ranging from image generation (Rombach et al., 2022a; Podell et al., 2023; Ramesh et al., 2022) to more complex tasks like structure-based drug design (Corso et al., 2022; Guan et al., 2023), protein structure prediction (Watson et al., 2021), text generation (Austin et al., 2021; Zheng et al., 2023; Chen et al., 2023), and more. Prominent diffusion models in image generation, including DALL-E (Ramesh et al., 2022), Stable Diffusion (Rombach et al., 2022b), SDXL (Podell et al., 2023), and Dreamlike, etc., typically undergo a fine-tuning process following their initial pre-training phase.

Standard fine-tuning method for diffusion models suffers from low alignment with human preferences and low data efficiency due to two main reasons: (1) it does not directly optimize for alignment with human preferences, and (2) only one round of training can be performed. Recently, using Reinforcement Learning (RL) for fine-tuning diffusion models has received increasing attention. Lee et al. (2023) first studied the alignment of text-image diffusion models to human preferences using reward-weighted likelihood maximization with a reward function trained on human preference data.

*Equal contribution

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Black et al. (2023) formulated the fine-tuning of diffusion models as a RL problem solved by policy gradient optimization. In a concurrent work, Fan et al. (2023) studied a similar formulation but with a KL regularization. Very recently, Wallace et al. (2023) have bypassed the need for training reward functions by using Direct Preference Optimization (DPO) (Rafailov et al., 2023) for fine-tuning diffusion models. Similar approach was proposed in Yang et al. (2023) as well.

While RL fine-tuning of diffusion methods has been proven effective, its dependency on human preference data, often necessitating multiple images per prompt, poses a significant challenge. In many datasets including the community-sourced ones featuring custom content, it is often the case to have only one image associated with each prompt. This makes RL fine-tuning infeasible.

In this paper, drawing inspiration from the recently proposed self-play fine-tuning (SPIN) technique (Chen et al., 2024) for large language models (LLM), we introduce a new supervised fine-tuning (SFT) method for diffusion models, eliminating the necessity for human preference data in the fine-tuning process. Central to our method is a general-sum minimax game, where both the participating players, namely the main player and the opponent player, are diffusion models. The main player's goal is to discern between samples drawn from the target data distribution and those generated by the opponent player. The opponent player's goal is to garner the highest score possible, as assessed by the main player. A self-play mechanism can be made possible, if and only if the main player and the opponent player have the same structure, and therefore the opponent player can be designed to be previous copies of the main player (Chen et al., 2024). The proposed algorithm SPIN-Diffusion overcomes the drawbacks of both supervised fine-tuning (SFT) and RL fine-tuning. Compared with SFT, our method is more data-efficient, by repeatedly using the prompts from the SFT dataset to improve the model through self-play. Compared with RL fine-tuning methods, our method does not need external reward models or expensive human-annotated winner/loser pairs.

When applying the self-play fine-tuning technique (Chen et al., 2024) to diffusion models, there are two challenges: (a) an exponential or even infinite number of possible trajectories can lead to the same image. The generator in a diffusion model operates through a sequence of intermediate steps, but the performance of the generator is only determined by the quality of the image in the last step; and (b) diffusion models are parameterized by a sequence of score functions, which are the gradient of the probabilities rather than probabilities in LLMs. Our algorithm design effectively surmounts these challenges by (a) designing an objective function that considers all intermediate images generated during the reverse sampling process; and (b) decomposing and approximating the probability function step-by-step into products related to the score function. We also employ the Gaussian reparameterization technique in DDIM (Song et al., 2020a) to support the advanced sampling method. All these techniques together lead to an unbiased objective function that can be effectively calculated based on intermediate samples. For computational efficiency, we further propose an approximate objective function, which eliminates the need for intermediate images used in our model.

Contributions. Our contributions are summarized below:

- We propose a novel fine-tuning method for diffusion models based on the self-play mechanism, called SPIN-Diffusion. The proposed algorithm iteratively improves upon a diffusion model until converging to the target distribution. Theoretically, we prove that the model obtained by SPIN-Diffusion cannot be further improved via standard SFT. Moreover, the stationary point of our self-play mechanism is achieved when the diffusion model aligns with the target distribution.

- Empirically, we evaluate the performance of SPIN-Diffusion on text-to-image generation tasks (Ramesh et al., 2022; Rombach et al., 2022a; Saharia et al., 2022a). Our experiments on the Pick-a-Pic dataset (Kirstain et al., 2023), with base model being Stable Diffusion-1.5 (Rombach et al., 2022b), demonstrate that SPIN-Diffusion surpasses SFT from the very first iteration. Notably, by the second iteration, SPIN-Diffusion outperforms Diffusion-DPO (Wallace et al., 2023) that utilizes additional data from 'loser' samples. By the third iteration, the images produced by SPIN-Diffusion achieve a higher PickScore (Kirstain et al., 2023) than the base model SD-1.5 79.8% of the times, and a superior Aesthetic score 88.4% of the times.

SPIN-Diffusion exhibits a remarkable performance improvement over current state-of-the-art fine-tuning algorithms, retaining this advantage even against models trained with more extensive data usage. This highlights its exceptional efficiency in dataset utilization. It is beneficial for the general public, particularly those with restricted access to datasets containing multiple images per prompt.

Notation. We use lowercase letters and lowercase boldface letters to denote scalars and vectors, respectively. We use $0 : T$ to denote the index set $\{0, \ldots, T\}$. In the function space, let $\mathcal{F}$ be the function class. We use the symbol $\mathbf{q}$ to denote the real distribution in a diffusion process, while $\mathbf{p}_{\boldsymbol{\theta}}$ represents the distribution parameterized by a nueral network during sampling. The Gaussian distribution is represented as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix, respectively. Lastly, $\mathrm{Uniform}\{1, \ldots, T\}$ denotes the uniform distribution over the set $\{1, \ldots, T\}$.

2 Related Work

Diffusion Models. Diffusion-based generative models (Sohl-Dickstein et al., 2015) have recently gained prominence, attributed to their ability to produce high-quality and diverse samples. A popular diffusion model is denoising diffusion probabilistic modeling (DDPM) (Ho et al., 2020). Song et al. (2020a) proposed a denoising diffusion implicit model (DDIM), which extended DDPM to a non-Markov diffusion process, enabling a deterministic sampling process and the accelerated generation of high-quality samples. In addition to DDPM and DDIM, diffusion models have also been studied with a score-matching probabilistic model using Langevin dynamics (Song and Ermon, 2019; Song et al., 2020b). Diffusion models evolved to encompass guided diffusion models, which are designed to generate conditional distributions. When the conditioning input is text and the output is image, these models transform into text-to-image diffusion models (Rombach et al., 2022a; Ramesh et al., 2022; Ho et al., 2022; Saharia et al., 2022b). They bridge the gap between textual descriptions and image synthesis, offering exciting possibilities for content generation. A significant advancement in text-to-image generation is the introduction of Stable Diffusion (SD) (Rombach et al., 2022a). SD has expanded the potential of diffusion models by integrating latent variables into the generation process. This innovation in latent diffusion models enables the exploration of latent spaces and improves the diversity of generated content. Despite the introduction of latent spaces, generating images with desired content from text prompts remains a significant challenge (Gal et al., 2022; Ruiz et al., 2023). This is due to the difficulty in learning the semantic properties of text prompts with limited high-quality data.

Fine-Tuning Diffusion Models. Efforts to improve diffusion models have focused on aligning them more closely with human preferences. Rombach et al. (2022a) fine-tuned a pre-trained model using the COCO dataset (Caesar et al., 2018), demonstrating superior performance compared to a generative model directly trained on the same dataset. Podell et al. (2023) expanded the model size of Stable Diffusion (SD) to create the SDXL model, which was fine-tuned on a high-quality but private dataset, leading to a significant improvement in the aesthetics of the generated images. Dai et al. (2023) further demonstrated the effectiveness of fine-tuning and highlighted the importance of the supervised fine-tuning (SFT) dataset. In addition to using datasets with high-quality images, Betker et al. (2023); Segalis et al. (2023) found that SFT on a data set with high text fidelity can also improve the performance of the diffusion model. The aforementioned methods only requires a high-quality SFT dataset. Recently, preference datasets have been studied in finetuing diffusion models (Lee et al., 2023). Concurrently, DDPO (Black et al., 2023) and DPOK (Fan et al., 2023) proposed to use the preference dataset to train a reward model and then fine-tune diffusion models using reinforcement learning. Drawing inspiration from the recent Direct Preference Optimization (DPO) (Rafailov et al., 2023), Diffusion-DPO (Wallace et al., 2023) and D3PO (Yang et al., 2023) used the implicit reward to fine-tune diffusion models directly on the preference dataset. Furthermore, when a differentiable reward model is available, Clark et al. (2023); Prabhudesai et al. (2023) applied reward backpropagation for fine-tuning diffusion models. Our SPIN-Diffusion is most related to the SFT method, as it only assumes access to high-quality image-text pairs. However, the high-quality image-text dataset can be obtained from various sources, including selecting the winner from a preference dataset or identifying high-reward image-text pairs through a reward model.

3 Problem Setting and Preliminaries

In this section, we introduce basic settings for text-to-image generation by diffusion models and the self-play fine-tuning (SPIN) method.

3.1 Text-to-Image Diffusion Model

Denoising diffusion implicit models (DDIM) (Song et al., 2020a) is a generalized framework of denoising diffusion probabilistic models (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020). DDIM enables the fast generation of high-quality samples and has been widely used in text-to-image

diffusion models such as Stable Diffusion (Rombach et al., 2022a). We formulate our method building upon DDIM, which makes it more general.

Forwrd Process. Following Saharia et al. (2022b), the problem of text-to-image generation can be formulated as conditional diffusion models. We use $\mathbf{x}_0 \in \mathbb{R}^d$ to denote the value of image pixels where d is the dimension and use $\mathbf{c}$ to denote the text prompt. Given a prompt $\mathbf{c}$, image $\mathbf{x}_0$ is drawn from a target data distribution $p_{\text{data}}(\cdot|\mathbf{c})$. The diffusion process is characterized by the following dynamic parameterized by a positive decreasing sequence $\{\alpha_t\}_{t=1}^T$ with $\alpha_0 = 1$,

$$q(\mathbf{x}_{1:T}|\boldsymbol{x}_0) := q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0), \tag{3.1}$$

where $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ represents a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_t, \sigma_t^2\mathbf{I})$. Here, $\boldsymbol{\mu}_t$ is the mean of Gaussian defined as

$$\boldsymbol{\mu}_t := \sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}.$$

It can be derived from (3.1) that $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I})$ for all t (Song et al., 2020a). As a generalized diffusion process of DDPM, (3.1) reduces to DDPM (Ho et al., 2020) with a special choice of $\sigma_t = \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{(1-\alpha_t/\alpha_{t-1})}$.

Generative Process. Given the sequence of $\{\alpha_t\}_{t=1}^T$ and $\{\sigma_t\}_{t=1}^T$, examples from the generative model follows

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \cdot p_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c}), \qquad p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t), \sigma_t^2\mathbf{I}\big). \tag{3.2}$$

Here $\boldsymbol{\theta}$ belongs to the parameter space $\boldsymbol{\Theta}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)$ is the estimator of mean $\boldsymbol{\mu}_t$ that can be reparameterized (Ho et al., 2020; Song et al., 2020a) as the combination of $\mathbf{x}_t$ and a neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)$ named score function. Please see Appendix C for more details.

Training Objective. The score function $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)$ is trained by minimizing the evidence lower bound (ELBO) associated with the diffusion models in (3.1) and (3.2), which is equivalent to minimizing the following denoising score matching objective function L_{DSM}:

$$L_{\text{DSM}}(\boldsymbol{\theta}) = \mathbb{E}\big[\gamma_t \big\| \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t) - \boldsymbol{\epsilon}_t \big\|_2^2\big], \tag{3.3}$$

where $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t$ and the expectation is computed over the distribution $\mathbf{c} \sim q(\cdot), \mathbf{x}_0 \sim q_{\text{data}}(\cdot|\mathbf{c}), \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$, $t \sim \text{Uniform}\{1, \ldots, T\}$. In addition, $\{\gamma_t\}_{t=1}^T$ are pre-specified weights that depends on the sequences $\{\alpha_t\}_{t=1}^T$ and $\{\sigma_t\}_{t=1}^T$.

3.2 Self-Play Fine-Tuning

Self-Play mechanism, originating from TD-Gammon (Tesauro et al., 1995), has achieved great successes in various fields, particularly in strategic games (Silver et al., 2017b,a). Central to Self-Play is the idea of progressively improving a model by competing against its previous iteration. This approach has recently been adapted to fine-tuning Large Language Models (LLMs) (Chen et al., 2024), called self-play fine-tuning (SPIN). Considering an LLM where $\mathbf{c}$ is the input prompt and $\mathbf{x}_0$ is the response, the goal of SPIN is to fine-tune an LLM agent, denoted by $p_{\boldsymbol{\theta}}(\cdot|\mathbf{c})$, based on an SFT dataset. Chen et al. (2024) assumed access to a main player and an opponent player at each iteration and takes the following steps iteratively:

1. The main player maximizes the expected value gap between the target data distribution p_{data} and the opponent player's distribution $p_{\boldsymbol{\theta}_k}$:
2. The opponent player generates responses that are indistinguishable from p_{data} by the main player.

Instead of alternating optimization, SPIN directly utilizes a closed-form solution of the opponent player, which results in the opponent player at iteration $k+1$ to copy parameters $\boldsymbol{\theta}_{k+1}$, and forming an end-to-end training objective:

$$L_{\texttt{SPIN}} = \mathbb{E}\left[\ell\left(\lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{c})}{p_{\boldsymbol{\theta}_k}(\mathbf{x}_0|\mathbf{c})} - \lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_0'|\mathbf{c})}{p_{\boldsymbol{\theta}_k}(\mathbf{x}_0'|\mathbf{c})}\right)\right]. \tag{3.4}$$

Here the expectation is taken over the distribution $\mathbf{c} \sim q(\mathbf{c}), \mathbf{x} \sim p_{\text{data}}(\mathbf{x}|\mathbf{c}), \mathbf{x}' \sim p_{\boldsymbol{\theta}_k}(\mathbf{x}'|\mathbf{c})$, $\ell(\cdot)$ is a loss function that is both monotonically decreasing and convex, and $\lambda > 0$ is a hyperparameter. Notably, (3.4) only requires the knowledge of demonstration/SFT data, i.e., prompt-response pairs.

Algorithm 1 Self-Play Diffusion (SPIN-Diffusion)

Input: $\{(\mathbf{x}_0, \mathbf{c})\}_{i\in[N]}$: SFT Dataset, $p_{\boldsymbol{\theta}_0}$: Diffusion Model with parameter $\boldsymbol{\theta}_0$, K: Number of iterations.
for $k = 0, \ldots, K-1$ **do**
  for $i = 1, \ldots N$ **do**
    Generate real diffusion trajectories $\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)$.
    Generate synthetic diffusion trajectories $\mathbf{x}'_{0:T} \sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})$.
  end for
  Update $\boldsymbol{\theta}_{k+1} = \operatorname{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \widehat{L}_{\texttt{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$, which is the empirical version of (4.8) or (4.9) .
end for
Output: $\boldsymbol{\theta}_K$.

4 Method

In this section, we are going to present a method for fine-tuning diffusion models with self-play mechanism.

Consider a setting where we are training on a high-quality dataset containing image-text pairs $(\mathbf{c}, \mathbf{x}_0) \sim p_{\text{data}}(\mathbf{x}_0|\mathbf{c})q(\mathbf{c})$ where $\mathbf{c}$ is the text prompt and $\mathbf{x}_0$ is the image. Our goal is to fine-tune a pretrained diffusion model, denoted by $p_{\boldsymbol{\theta}}$, to align with the distribution $p_{\text{data}}(\mathbf{x}_0|\mathbf{c})$. Instead of directly minimizing the denoising score matching objective function L_{DSM} in (3.3), we adapt `SPIN` to diffusion models. However, applying SPIN to fine-tuning diffusion models presents unique challenges. Specifically, the objective of `SPIN` (3.4) necessitates access to the marginal probability $p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{c})$. While obtaining $p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{c})$ is straightforward in LLMs, this is not the case with diffusion models. Given the parameterization of the diffusion model as $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})$, computing the marginal probability $p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{c})$ requires integration over all potential trajectories $\int_{\mathbf{x}_{1:T}} p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})d\mathbf{x}_{1:T}$, which is computationally intractable.

In the following, we propose a novel SPIN-Diffusion method with a decomposed objective function that only requires the estimation of score function $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$. This is achieved by employing the DDIM formulation discussed in Section 3. The key technique is self-play mechanism with a focus on the joint distributions of the entire diffusion process, i.e., $p_{\text{data}}(\mathbf{x}_{0:T}|\mathbf{c}) = q(\mathbf{x}_{1:T}|\mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0|\mathbf{c})$ and $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})$, instead of marginal distributions.

4.1 Differentiating Diffusion Processes

In iteration $k+1$, we focus on training a function f_{k+1} to differentiate between the diffusion trajectory $\mathbf{x}_{0:T}$ generated by the diffusion model parameterized by $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})$, and the diffusion process $p_{\text{data}}(\mathbf{x}_{0:T}|\mathbf{c})$ from the data. Specifically, the training of f_{k+1} involves minimizing a generalized Integral Probability Metric (IPM) (Müller, 1997):

$$f_{k+1} = \operatorname*{argmin}_{f\in\mathcal{F}_k} \mathbb{E}\big[\ell\big(f(\mathbf{c}, \mathbf{x}_{0:T}) - f(\mathbf{c}, \mathbf{x}'_{0:T})\big)\big]. \tag{4.1}$$

Here, the expectation is taken over the distributions $\mathbf{c} \sim q(\cdot), \mathbf{x}_{0:T} \sim p_{\text{data}}(\cdot|\mathbf{c})$, and $\mathbf{x}'_{0:T} \sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})$. $\mathcal{F}_k$ denotes the class of functions under consideration and $\ell(\cdot)$ is a monotonically decreasing and convex function that helps stabilize training. The value of f reflects the degree of belief that the diffusion process $\mathbf{x}_{0:T}$ given context $\mathbf{c}$ originates from the target diffusion process $p_{\text{data}}(\mathbf{x}_{0:T}|\mathbf{c})$ rather than the diffusion model $p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})$. We name f the test function.

4.2 Deceiving the Test Function

The opponent player wants to maximize the expected value $\mathbb{E}_{\mathbf{c}\sim q(\cdot),\mathbf{x}_{0:T}\sim p(\cdot|\mathbf{c})}[f_{k+1}(\mathbf{c}, \mathbf{x})]$. In addition, to prevent excessive deviation of $p_{\boldsymbol{\theta}_{k+1}}$ from $p_{\boldsymbol{\theta}_k}$ and stabilize the self-play fine-tuning, we incorporate a Kullback-Leibler (KL) regularization term. Putting these together gives rise to the following optimization problem:

$$\operatorname*{argmax}_{p} \mathbb{E}_{\mathbf{c}\sim q(\cdot),\mathbf{x}_{0:T}\sim p(\cdot|\mathbf{c})}[f_{k+1}(\mathbf{c}, \mathbf{x}_{0:T})] - \lambda\mathbb{E}_{\mathbf{c}\sim q(\cdot)}\mathrm{KL}\big(p(\cdot|\mathbf{c})||p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})\big), \tag{4.2}$$

where $\lambda > 0$ is the regularization parameter. Notably, (4.2) has a closed-form solution $\widehat{p}(\cdot|\mathbf{c})$:

$$\widehat{p}(\mathbf{x}_{0:T}|\mathbf{c}) \propto p_{\boldsymbol{\theta}_k}(\mathbf{x}_{0:T}|\mathbf{c}) \exp\big(\lambda^{-1} f_{k+1}(\mathbf{c}, \mathbf{x}_{0:T})\big). \tag{4.3}$$

To ensure that $\widehat{p}$ lies in the diffusion process space $\{p_{\boldsymbol{\theta}}(\cdot|\mathbf{c})|\boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, we utilize the following test function class (Chen et al., 2024):

$$\mathcal{F}_k = \left\{ \lambda \cdot \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{1:T}|\mathbf{c})}{p_{\boldsymbol{\theta}_k}(\mathbf{x}_{1:T}|\mathbf{c})} \middle| \boldsymbol{\theta} \in \boldsymbol{\Theta} \right\}. \tag{4.4}$$

Given the choice of $\mathcal{F}_k$ in (4.4), optimizing (4.1) gives f_{k+1} parameterized by $\boldsymbol{\theta}_{k+1}$ in the following form:

$$f_{k+1}(\mathbf{c}, \mathbf{x}_{0:T}) = \lambda \cdot \log \frac{p_{\boldsymbol{\theta}_{k+1}}(\mathbf{x}_{0:T}|\mathbf{c})}{p_{\boldsymbol{\theta}_k}(\mathbf{x}_{0:T}|\mathbf{c})}. \tag{4.5}$$

Substituting (4.5) into (4.3) yields $\widehat{p}(\mathbf{x}_{0:T}|\mathbf{c}) = p_{\boldsymbol{\theta}_{k+1}}(\mathbf{x}_{0:T}|\mathbf{c})$. In other words, $\boldsymbol{\theta}_{k+1}$ learned from (4.1) is exactly the diffusion parameter for the ideal choice of opponent.

4.3 Decomposed Training Objective

The above two steps provide a training scheme depending on the full trajectory of $\mathbf{x}_{0:T}$. Specifically, substituting (4.4) into (4.1) yields the update rule $\boldsymbol{\theta}_{k+1} = \operatorname{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} L_{\texttt{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$, where $L_{\texttt{SPIN}}$ is defined as:

$$L_{\texttt{SPIN}} = \mathbb{E}\left[\ell\left(\lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})}{p_{\boldsymbol{\theta}_k}(\mathbf{x}_{0:T}|\mathbf{c})} - \lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}'_{0:T}|\mathbf{c})}{p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{0:T}|\mathbf{c})} \right) \right]. \tag{4.6}$$

Here the expectation is taken over the distributions $\mathbf{c} \sim q(\cdot), \mathbf{x}_{0:T} \sim p_{\text{data}}(\cdot|\mathbf{c}), \mathbf{x}'_{0:T} \sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})$. To formulate a computationally feasible objective, we decompose $\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c})$ using the backward process of diffusion models. Substituting (3.2) into (4.6), we have that

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c}) &= \log \left(\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \cdot p_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c}) \right) \\
&= \log p_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c}) + \sum_{t=1}^{T} \log \left(p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \right) \\
&= \text{Constant} - \sum_{t=1}^{T} \frac{1}{2\sigma_t^2} \left\| \mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t) \right\|_2^2.
\end{aligned} \tag{4.7}$$

where the last equality holds since $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c})$ is a Gaussian distribution $\mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t), \sigma_t^2 \mathbf{I}\big)$ according to (3.2), and $p_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c})$ is approximately a Gaussian independent of $\boldsymbol{\theta}$. By substituting (4.7) into (4.6) and introducing a reparameterization $\sigma_t^2 = \lambda T/(2\beta_t)$, where β_t is a fixed positive value, we obtain

$$\begin{aligned}
L_{\texttt{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \mathbb{E}\Bigg[\ell\Bigg(&- \sum_{t=1}^{T} \frac{\beta_t}{T} \left[\left\| \mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t) \right\|_2^2 - \left\| \mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t) \right\|_2^2 \right. \\
&\left. - \left\| \mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t) \right\|_2^2 + \left\| \mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t) \right\|_2^2 \right] \Bigg) \Bigg].
\end{aligned} \tag{4.8}$$

Here the expectation is taken over the distributions $\mathbf{c} \sim q(\cdot), \mathbf{x}_{0:T} \sim p_{\text{data}}(\cdot|\mathbf{c}), \mathbf{x}'_{0:T} \sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})$. Note that by considering the main player (reward function) across the full trajectory (3.2), rather than focusing solely on the final state as in Fan et al. (2023); Black et al. (2023); Wallace et al. (2023), we are able to formulate an exact objective function up to Equation (4.8). The detailed algorithm is presented in Algorithm 1. (4.8) naturally provides an objective function for DDIM with $\sigma_t > 0$, where σ_t controls the determinism of the reverse process (3.2). (4.8) remains valid for deterministic generation processes as $\sigma_t \to 0$.

4.4 Approximate Training Objective

While (4.8) is the exact ELBO, optimizing it requires storing all intermediate images during the reverse sampling. When the trajectory length T is large, it would require an impractical amount of GPU memory when the loss is summed over T. Additionally, the required samples from a reverse process are not readily accessible. To address these limitations, we propose an approximate objective function. By applying Jensen's inequality and the convexity of the loss function ℓ, we can give an upper bound of (4.8) and thus move the average over t outside the loss function ℓ:

$$L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \mathbb{E}\Bigg[\ell\Big(- \beta_t \Big[\left\| \mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t) \right\|_2^2 - \left\| \mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t) \right\|_2^2$$

$$- \left\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\right\|_2^2 + \left\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\right\|_2^2\Big]\Big)\Big], \tag{4.9}$$

where the expectation is taken over the distributions $\mathbf{c} \sim q(\mathbf{c}), (\mathbf{x}_{t-1}, \mathbf{x}_t) \sim p_{\text{data}}(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{c}), (\mathbf{x}'_{t-1}, \mathbf{x}'_t) \sim p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{c}),\ t \sim \text{Uniform}\{1, \ldots, T\}$. This approximation is directly motivated by the nature of diffusion models, which inherently decouple operations on a per-time-step basis. We provide theoretical justifications for our approximation method in the following sections.

The following lemma shows that $L_{\texttt{SPIN}}^{\text{approx}}$ is an upper bound of $L_{\texttt{SPIN}}$.

Lemma 4.1. Fix $\boldsymbol{\theta}_k \in \boldsymbol{\Theta}$ which serves as the starting point of Algorithm 1 for iteration $k+1$. It holds that $L_{\texttt{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \leq L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

$L_{\texttt{SPIN}}^{\text{approx}}$ eliminates the need to store all intermediate steps, as it only involves two consecutive sampling steps $t-1$ and t. Since the reverse process $p_{\boldsymbol{\theta}}(\mathbf{x}'_{1:T}|\mathbf{x}'_0, \mathbf{c})$ approximates the forward process $q(\mathbf{x}'_{1:T}|\mathbf{x}'_0)$, we use the per step forward process $q(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0)$ to approximate $p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0, \mathbf{c})$. We can further approximate $p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{c}) = \int p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0, \mathbf{c}) p_{\boldsymbol{\theta}_k}(\mathbf{x}'_0|\mathbf{c}) d\mathbf{x}'_0$ with $\int q(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0) p_{\boldsymbol{\theta}_k}(\mathbf{x}'_0|\mathbf{c}) d\mathbf{x}'_0$. Substituting the corresponding terms in (4.9) with the above approximation allows us to only compute the expectation of (4.9) over the distribution $\mathbf{c} \sim q(\mathbf{c}), (\mathbf{x}_{t-1}, \mathbf{x}_t) \sim p_{\text{data}}(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{c}), (\mathbf{x}'_{t-1}, \mathbf{x}'_t) \sim \int p_{\boldsymbol{\theta}_k}(\mathbf{x}'_0|\mathbf{c}) q(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0) d\mathbf{x}'_0, t \sim \text{Uniform}\{1, \ldots, T\}$. Furthermore, by incorporating the reparameterization of $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ into (4.8) and (4.9), we can express (4.8) and (4.9) in terms of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)$. Detailed derivations of (4.8) and (4.9) are provided in Appendix C.

5 Main Theory

In this section, we provide a theoretical analysis of Algorithm 1. Section 4 introduces two distinct objective functions, as defined in (4.8) and (4.9), both of which use the loss function ℓ. Since (4.8) is an exact objective function, its analysis closely follows the framework established by Chen et al. (2024). Consequently, we instead focus on the approximate objective function $L_{\texttt{SPIN}}^{\text{approx}}$ defined in (4.9), which is more efficient to optimize and is the algorithm we use in our experiments. However, its behavior is more difficult to analyze. We begin with a formal assumption regarding the loss function ℓ as follows.

Assumption 5.1. The function $\ell(t) : \mathbb{R} \to \mathbb{R}$ in (4.9) is monotonically decreasing, i.e., $\forall t, \ell'(t) \leq 0$ and satisfies $\ell'(0) < 0$. In addition, $\ell(t)$ is a convex function.

Assumption 5.1 can be satisfied by various commonly used loss functions in machine learning. This includes the correlation loss $\ell(t) = 1 - t$, the hinge loss $\ell(t) = \max(0, 1-t)$, and the logistic loss $\ell(t) = \log(1 + \exp(-t))$. In our experiments, we are using the logistic loss.

To understand the behavior of SPIN-Diffusion, let us first analyze the gradient of the objective function (4.9),

$$\nabla L_{\texttt{SPIN}}^{\text{approx}} = \mathbb{E}\Big[\underbrace{(-\beta_t \ell'_t)}_{\text{Reweighting}} \cdot \Big(\underbrace{\nabla_{\boldsymbol{\theta}} \left\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\right\|_2^2}_{\text{Matching}} - \underbrace{\nabla_{\boldsymbol{\theta}} \left\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\right\|_2^2}_{\text{Pushing}} \Big) \Big], \tag{5.1}$$

where the expectation is taken over the distributions $\mathbf{c} \sim q(\mathbf{c}), (\mathbf{x}_{t-1}, \mathbf{x}_t) \sim p_{\text{data}}(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{c}), (\mathbf{x}'_{t-1}, \mathbf{x}'_t) \sim p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{c})$. (D.3) can be divided into three parts:

- **Reweighting:** $\ell'(\cdot)$ in the "Reweighting" term is negative and increasing because $\ell()$ is monotonically decreasing and convex according to Assumption 5.1. Therefore, $-\beta_t \ell'_t = -\beta_t \ell'\big(-\beta_t \big[\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\|_2^2 - \ldots + \|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\|_2^2\big]\big)$ is always non-negative. Furthermore, $-\beta_t \ell'_t$ decreases as the argument inside $\ell()$ increases.
- **Matching:** The "Matching" term matches $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)$ to $\mathbf{x}_{t-1}$ coming from pairs $(\mathbf{x}_{t-1}, \mathbf{x}_t)$, that are sampled from the target distribution. This increases the likelihood of $(\mathbf{x}_{t-1}, \mathbf{x}_t) \sim p_{\text{data}}(\mathbf{x}_{t-1}, \mathbf{x}_t)$ following the generative process (3.2).
- **Pushing:** Contrary to the "Matching" term, the "Pushing" term pushes $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)$ away from $\mathbf{x}'_{t-1}$ coming from pairs $(\mathbf{x}'_{t-1}, \mathbf{x}'_t)$ drawn from the synthetic distribution $p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1}, \mathbf{x}'_t)$. Therefore, the "Pushing" term decreases the likelihood of these samples following the process in the generative process (3.2).

The "Matching" term aligns conceptually with the L_{DSM} in SFT, as both aim to maximize the likelihood that the target trajectory $\mathbf{x}_{0:T}$ follows the generative process described in (3.2). The following theorem shows a formal connection, which is pivotal for understanding the optimization dynamics of our method.

Theorem 5.2. Under Assumption 5.1, if $\boldsymbol{\theta}_k$ is not the global optima of L_{DM} in (3.3), there exists an appropriately chosen β_t, such that $\boldsymbol{\theta}_k$ is not the global minima of (4.9) and thus $\boldsymbol{\theta}_{k+1} \neq \boldsymbol{\theta}_k$.

Theorem 5.2 suggests that the optimization process stops only when $\boldsymbol{\theta}$ reaches global optimality of L_{DSM}. Consequently, the optimal diffusion model $\boldsymbol{\theta}^*$ found by Algorithm 1 cannot be further improved using L_{DSM}. This theoretically supports that SFT with (3.3) cannot improve over SPIN-Diffusion. It is also worth noting that Theorem 5.2 does not assert that every global minimum of L_{DSM} meets the convergence criterion (i.e., $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k$), particularly due to the influence of the "Pushing" term in (D.3). The following theorem provides additional insight into the conditions under which Algorithm 1 converges.

Theorem 5.3. Under Assumption 5.1, if $p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{x}) = p_{\mathrm{data}}(\cdot|\mathbf{x})$, then $\boldsymbol{\theta}_k$ is the global minimum of (4.9) for any $\lambda \geq 0$.

Theorem 5.3 shows that Algorithm 1 converges when $p_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = p_{\mathrm{data}}(\cdot|\mathbf{x})$, indicating the efficacy of SPIN-Diffusion in aligning with the target data distribution. In addition, while Theorems 5.2 and 5.3 are directly applicable to (4.9), the analogous conclusion can be drawn for (4.8) as well (see Appendix D for a detailed discussion).

6 Experiments

In this section, we conduct extensive experiments to demonstrate the effectiveness of SPIN-Diffusion. Our results show that SPIN-Diffusion outperforms other baseline fine-tuning methods including SFT and Diffusion-DPO.

6.1 Experiment Setup

Models, Datasets and Baselines. We use the stable diffusion v1.5 (SD-1.5) (Rombach et al., 2022a) as our base model. While adopting the original network structure, we use its Huggingface pretrained version[2], which is trained on LAION-5B (Schuhmann et al., 2022) dataset, a text-image pair dataset containing approximately 5.85 billion CLIP-filtered image-text pairs. We use the Pick-a-Pic dataset (Kirstain et al., 2023) for fine-tuning. Pick-a-Pic is a dataset with pairs of images generated by Dreamlike[3] (a fine-tuned version of SD-1.5) and SDXL-beta (Podell et al., 2023), where each pair corresponds to a human preference label. We also train SD-1.5 with SFT and Diffusion-DPO (Wallace et al., 2023) as the baselines. For SFT, we train the model to fit the winner images in the Pick-a-Pic (Kirstain et al., 2023) trainset. In addition to the Diffusion-DPO checkpoint provided by Wallace et al. (2023)[4] (denoted by Diffusion-DPO), we also fine-tune an SD-1.5 using Diffusion-DPO and denote it by "Diffusion-DPO (reproduced)".

Evaluation. We use the Pick-a-Pic test set, PartiPrompts (Yu et al., 2022) and HPSv2 (Wu et al., 2023) as our evaluation benchmarks. We defer the detailed introduction and results of PartiPrompts and HPSv2 to Appendix A.3. Our evaluation rubric contains two dimensions, human preference alignment and visual appeal. For visual appeal assessment, we follow Wallace et al. (2023); Lee et al. (2024) and use Aesthetic score. For human-preference alignment, we employ reward models including PickScore (Kirstain et al., 2023), ImageReward (Xu et al., 2023) and HPS (Wu et al., 2023). All these reward models are trained according to the Bradley-Terry-Luce (Bradley and Terry, 1952) model on different human-labeled preference datasets. For each prompt, we generate 5 images and choose the image with highest average score over those four metrics (best out of 5). We report the average of HPS, PickScore, ImageReward and Aesthetic scores over all the prompts. To investigate how the scores align with human preference, we further compare the accuracy of these reward models on a small portion of the Pick-a-Pic training set. It is worth noticing that PickScore is most aligned with human preference according to the experiments conducted by Kirstain et al. (2023).

[2]https://huggingface.co/runwayml/stable-diffusion-v1-5

[3]https://dreamlike.art/

[4]https://huggingface.co/mhdang/dpo-sd1.5-text2image-v1

Table 1: The results on the Pick-a-Pic test set. We report the mean of PickScore, HPS, ImageReward and Aesthetic over the whole test set. We also report the average score over the three evaluation metrics. SPIN-Diffusion outperforms all the baselines in terms of four metrics. For this and following tables, we use blue background to indicate our method, **bold** numbers to denote the best and underlined for the second best.

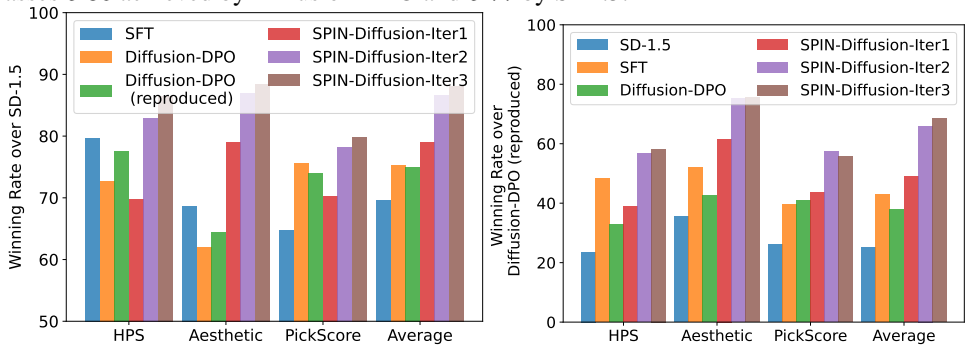
Model	HPS ↑	Aesthetic ↑	ImageReward ↑	PickScore ↑	Average ↑
SD-1.5	0.2699	5.7691	0.8159	21.1983	7.0133
SFT (reproduced)	0.2749	5.9451	1.1051	21.4542	7.1948
Diffusion-DPO	0.2724	5.8635	0.9625	21.5919	7.1726
Diffusion-DPO (reproduced)	0.2753	5.8918	1.0495	21.8866	7.2758
SPIN-Diffusion-Iter1	0.2728	6.1206	1.0131	21.6651	7.2679
SPIN-Diffusion-Iter2	0.2751	6.2399	1.1086	21.9567	7.3951
SPIN-Diffusion-Iter3	**0.2759**	**6.2481**	**1.1239**	**22.0024**	**7.4126**

6.2 Main Results

In this subsection, we provide empirical evidence demonstrating the superiority of our SPIN-Diffusion model over previous fine-tuning baselines based on the network structure of SD1.5.

Comparison in Terms of Average Score. The results are presented in Table 1. While all fine-tuning algorithms yield improvements over the SD1.5 baseline, at iteration 1, our SPIN-Diffusion not only exceeds the original DPO checkpoint but also surpasses SFT in both Aesthetic score and PickScore.

At iteration 2, the superiority of our model becomes even more pronounced, particularly in terms of Aesthetic score, where it consistently outperforms other fine-tuning methods, indicating a dominant performance in visual quality. Furthermore, at iteration 3, our model's HPSv2 score surpasses all competing models, highlighting the effectiveness and robustness of the SPIN-Diffusion approach. Specifically, on the Pick-a-Pic dataset, while SFT achieves a PickScore of 21.45, and Diffusion-DPO has a slightly higher score of 21.45, SPIN-Diffusion achieves 22.00 at iteration 3, showing a total improvement of 0.80 over the original SD1.5 checkpoint. Furthermore, SPIN-Diffusion demonstrates exceptional performance in terms of Aesthetic score, achieving 6.25 at iteration 3, which significantly surpasses 5.86 achieved by Diffusion-DPO and 5.77 by SD1.5.



(a) Compared to SD-1.5 (b) Compared to Diffusion-DPO (reproduced)

Figure 1: Left: winning rate in percentage of SFT, Diffusion-DPO, Diffusion-DPO (reproduced) and SPIN-Diffusion over SD1.5 checkpoint. Right: winning rate in percentage of SFT, Diffusion-DPO, Diffusion-DPO (reproduced) and SPIN-Diffusion over SD1.5 checkpoint. SPIN-Diffusion shows a much higher winning rate than SFT and Diffusion-DPO tuned models.

Comparison in Terms of Winning Rate. We further validate our claim by a comparative analysis of the winning rate for our trained model. The winning rate is defined as the proportion of prompts for which a model's generated images exceed the quality of those produced by another model. This experiment is conducted on the Pick-a-Pic test set. We show both the winning rate over SD-1.5, as well as the winning rate over Diffusion-DPO (reproduced) in Figure 1. The complete results are detailed in Tables 3 and 4 in Appendix A.2. We observe that throughout fine-tuning, our SPIN-Diffusion tremendously beats the baselines. When competing with SD-1.5, SPIN-Diffusion achieves an impressive winning rate of 90.0% at iteration 2, which further increases to 91.6% at iteration 3. This winning rate surpasses 73.2% achieved by SFT and 84.8% achieved by Diffusion-DPO (reproduced). When competing with Diffusion-DPO (reproduced), at iteration 3, SPIN-Diffusion

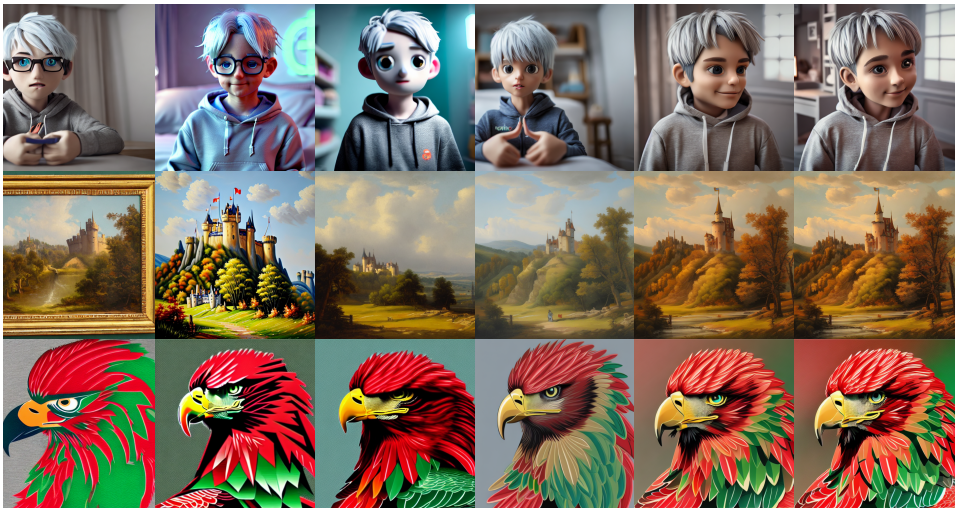


Figure 2: We show the images generated by different models. The prompts are "*a very cute boy, looking at audience, silver hair, in his room, wearing hoodie, at daytime, ai language model, 3d art, c4d, blender, pop mart, blind box, clay material, pixar trend, animation lighting, depth of field, ultra detailed*", "*painting of a castle in the distance*" and "*red and green eagle*". The models are: SD-1.5, SFT, Diffusion-DPO (reproduced), SPIN-Diffusion-Iter1, SPIN-Diffusion-Iter2, SPIN-Diffusion-Iter3 from left to right. SPIN-Diffusion demonstrates a notable improvement in image quality. The quantitative evaluation of the aesthetic score of the above images is in Table 5.

achieves a winning rate of 56.2% on HPS, 86.8% on Aesthetic, 62.4% on PickScore, 55.8% on Image Reward, and has an overall winning rate of 70.2%.

6.3 Qualitative Analysis

We illustrate the qualitative performance of our model on three prompts coming from the Pick-a-Pic test dataset. We prompt SD-1.5, SFT, Diffusion-DPO (reproduced), and SPIN-Diffusion at iteration 1 to 3 and present the generated images in Figure 2. Compared to the baseline methods, SPIN-Diffusion demonstrates a notable improvement in image quality, even more apparent than the improvements in scores. This is especially evident in aspects such as aligning, shading, visual appeal, and the intricacy of details within each image. This qualitative assessment underscores the effectiveness of SPIN-Diffusion in producing images that are not only contextually accurate but also visually superior to those generated by other existing models.

7 Conclusion

This paper presents SPIN-Diffusion, an innovative fine-tuning approach tailored for diffusion models, particularly effective in scenarios where only a single image is available per text prompt. By employing a self-play mechanism, SPIN-Diffusion iteratively refines the model's performance, converging towards the target data distribution. Theoretical evidence underpins the superiority of SPIN-Diffusion, demonstrating that traditional supervised fine-tuning cannot surpass its stationary point, achievable at the target data distribution. Empirical evaluations highlight SPIN-Diffusion's remarkable success in text-to-image generation tasks, surpassing the state-of-the-art fine-tuning methods even without the need for additional data. This underscores SPIN-Diffusion's potential to revolutionize the practice of diffusion model fine-tuning, leveraging solely demonstration data to achieve unprecedented performance levels.

Limitations While our theoretical analysis ensures that $\boldsymbol{\theta}_k$ is the only global optimum of our objective function, it relies on the assumption that the data distribution can be adequately represented by the parameterized family. Additionally, as our methodology is fundamentally a distribution matching algorithm, it cannot, in principle, exceed the performance of the underlying data distribution. Finally, although SPIN-Diffusion is data-efficient, it requires additional sampling overhead. The high sampling cost can be alleviated by software-level upgrades such as larger batch size, and memory-efficient attention backends. On the algorithm level, advanced sampling acceleration techniques also offer promising improvements. These techniques are orthogonal to our efforts in improving the performance of fine-tuning diffusion models, and therefore we decide to explore them as a future work.

Acknowledgement

We thank the anonymous reviewers and area chair for their helpful comments. HY, ZC, KJ, and QG are supported in part by the National Science Foundation CAREER Award 1906169, IIS-2008981, and the Sloan Research Fellowship. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

AUSTIN, J., JOHNSON, D. D., HO, J., TARLOW, D. and VAN DEN BERG, R. (2021). Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems* **34** 17981–17993.

BETKER, J., GOH, G., JING, L., BROOKS, T., WANG, J., LI, L., OUYANG, L., ZHUANG, J., LEE, J., GUO, Y. ET AL. (2023). Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf* **2** 3.

BLACK, K., JANNER, M., DU, Y., KOSTRIKOV, I. and LEVINE, S. (2023). Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301* .

BRADLEY, R. A. and TERRY, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* **39** 324–345.

CAESAR, H., UIJLINGS, J. and FERRARI, V. (2018). Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

CHEN, Z., DENG, Y., YUAN, H., JI, K. and GU, Q. (2024). Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335* .

CHEN, Z., YUAN, H., LI, Y., KOU, Y., ZHANG, J. and GU, Q. (2023). Fast sampling via de-randomization for discrete diffusion models. *arXiv preprint arXiv:2312.09193* .

CLARK, K., VICOL, P., SWERSKY, K. and FLEET, D. J. (2023). Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400* .

CORSO, G., STÄRK, H., JING, B., BARZILAY, R. and JAAKKOLA, T. (2022). Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776* .

CRESWELL, A., WHITE, T., DUMOULIN, V., ARULKUMARAN, K., SENGUPTA, B. and BHARATH, A. A. (2018). Generative adversarial networks: An overview. *IEEE signal processing magazine* **35** 53–65.

DAI, X., HOU, J., MA, C.-Y., TSAI, S., WANG, J., WANG, R., ZHANG, P., VANDENHENDE, S., WANG, X., DUBEY, A. ET AL. (2023). Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807* .

FAN, Y., WATKINS, O., DU, Y., LIU, H., RYU, M., BOUTILIER, C., ABBEEL, P., GHAVAMZADEH, M., LEE, K. and LEE, K. (2023). Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *arXiv preprint arXiv:2305.16381* .

GAL, R., ALALUF, Y., ATZMON, Y., PATASHNIK, O., BERMANO, A. H., CHECHIK, G. and COHEN-OR, D. (2022). An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* .

GUAN, J., ZHOU, X., YANG, Y., BAO, Y., PENG, J., MA, J., LIU, Q., WANG, L. and GU, Q. (2023). Decompdiff: Diffusion models with decomposed priors for structure-based drug design .

HO, J., JAIN, A. and ABBEEL, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33** 6840–6851.

HO, J., SAHARIA, C., CHAN, W., FLEET, D. J., NOROUZI, M. and SALIMANS, T. (2022). Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research* **23** 2249–2281.

KINGMA, D. P. and WELLING, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

KIRSTAIN, Y., POLYAK, A., SINGER, U., MATIANA, S., PENNA, J. and LEVY, O. (2023). Pick-a-pic: An open dataset of user preferences for text-to-image generation. *arXiv preprint arXiv:2305.01569* .

LEE, K., LIU, H., RYU, M., WATKINS, O., DU, Y., BOUTILIER, C., ABBEEL, P., GHAVAMZADEH, M. and GU, S. S. (2023). Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192* .

LEE, S. H., LI, Y., KE, J., YOO, I., ZHANG, H., YU, J., WANG, Q., DENG, F., ENTIS, G., HE, J., LI, G., KIM, S., ESSA, I. and YANG, F. (2024). Parrot: Pareto-optimal multi-reward reinforcement learning framework for text-to-image generation.

LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P. and ZITNICK, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer.

MÜLLER, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in applied probability* **29** 429–443.

NICHOL, A., DHARIWAL, P., RAMESH, A., SHYAM, P., MISHKIN, P., MCGREW, B., SUTSKEVER, I. and CHEN, M. (2021). Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* .

PEEBLES, W. and XIE, S. (2023). Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

PODELL, D., ENGLISH, Z., LACEY, K., BLATTMANN, A., DOCKHORN, T., MÜLLER, J., PENNA, J. and ROMBACH, R. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* .

PRABHUDESAI, M., GOYAL, A., PATHAK, D. and FRAGKIADAKI, K. (2023). Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739* .

RAFAILOV, R., SHARMA, A., MITCHELL, E., ERMON, S., MANNING, C. D. and FINN, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290* .

RAMESH, A., DHARIWAL, P., NICHOL, A., CHU, C. and CHEN, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* **1** 3.

ROMBACH, R., BLATTMANN, A., LORENZ, D., ESSER, P. and OMMER, B. (2022a). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

ROMBACH, R., BLATTMANN, A., LORENZ, D., ESSER, P. and OMMER, B. (2022b). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

RUIZ, N., LI, Y., JAMPANI, V., PRITCH, Y., RUBINSTEIN, M. and ABERMAN, K. (2023). Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

SAHARIA, C., CHAN, W., SAXENA, S., LI, L., WHANG, J., DENTON, E. L., GHASEMIPOUR, K., GONTIJO LOPES, R., KARAGOL AYAN, B., SALIMANS, T. ET AL. (2022a). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* **35** 36479–36494.

SAHARIA, C., HO, J., CHAN, W., SALIMANS, T., FLEET, D. J. and NOROUZI, M. (2022b). Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45** 4713–4726.

SCHUHMANN, C., BEAUMONT, R., VENCU, R., GORDON, C., WIGHTMAN, R., CHERTI, M., COOMBES, T., KATTA, A., MULLIS, C., WORTSMAN, M. ET AL. (2022). Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems* **35** 25278–25294.

SEGALIS, E., VALEVSKI, D., LUMEN, D., MATIAS, Y. and LEVIATHAN, Y. (2023). A picture is worth a thousand words: Principled recaptioning improves image generation. *arXiv preprint arXiv:2310.16656* .

SILVER, D., HUBERT, T., SCHRITTWIESER, J., ANTONOGLOU, I., LAI, M., GUEZ, A., LANCTOT, M., SIFRE, L., KUMARAN, D., GRAEPEL, T. ET AL. (2017a). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* .

SILVER, D., SCHRITTWIESER, J., SIMONYAN, K., ANTONOGLOU, I., HUANG, A., GUEZ, A., HUBERT, T., BAKER, L., LAI, M., BOLTON, A. ET AL. (2017b). Mastering the game of go without human knowledge. *nature* **550** 354–359.

SOHL-DICKSTEIN, J., WEISS, E., MAHESWARANATHAN, N. and GANGULI, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR.

SONG, J., MENG, C. and ERMON, S. (2020a). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* .

SONG, Y. and ERMON, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* **32**.

SONG, Y., SOHL-DICKSTEIN, J., KINGMA, D. P., KUMAR, A., ERMON, S. and POOLE, B. (2020b). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* .

TESAURO, G. ET AL. (1995). Temporal difference learning and td-gammon. *Communications of the ACM* **38** 58–68.

WALLACE, B., DANG, M., RAFAILOV, R., ZHOU, L., LOU, A., PURUSHWALKAM, S., ERMON, S., XIONG, C., JOTY, S. and NAIK, N. (2023). Diffusion model alignment using direct preference optimization. *arXiv preprint arXiv:2311.12908* .

WATSON, D., HO, J., NOROUZI, M. and CHAN, W. (2021). Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802* .

WU, X., HAO, Y., SUN, K., CHEN, Y., ZHU, F., ZHAO, R. and LI, H. (2023). Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341* .

XU, J., LIU, X., WU, Y., TONG, Y., LI, Q., DING, M., TANG, J. and DONG, Y. (2023). Imagereward: Learning and evaluating human preferences for text-to-image generation. *arXiv preprint arXiv:2304.05977* .

YANG, K., TAO, J., LYU, J., GE, C., CHEN, J., LI, Q., SHEN, W., ZHU, X. and LI, X. (2023). Using human feedback to fine-tune diffusion models without any reward model. *arXiv preprint arXiv:2311.13231* .

YU, J., XU, Y., KOH, J. Y., LUONG, T., BAID, G., WANG, Z., VASUDEVAN, V., KU, A., YANG, Y., AYAN, B. K. ET AL. (2022). Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789* **2** 5.

ZHENG, L., YUAN, J., YU, L. and KONG, L. (2023). A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737* .

Broader Impact

This approach enhances model performance across diverse benchmarks without the need for supervision from more advanced models, facilitating better alignment of AI with human preferences. This improvement bolsters the reliability and safety of AI systems in the field of text-to-image generation. It provides a more effective and scalable method for model fine-tuning, resulting in cost reductions and the expedited deployment of models that more accurately reflect human aesthetic and content preferences.

However, there exists the potential for overfitting, which may not lead to genuine improvements in real-world applications. Adhering too closely in creative fields such as text-to-image generation could inadvertently perpetuate existing societal biases in the generated imagery. Furthermore, the capability to finely tune alignment with human preferences could be exploited for unethical ends, such as crafting tailored and manipulative content or disseminating false information.

A Additional Details for Experiments

A.1 Hyperparameters

We train the SPIN-Diffusion on 8 NVIDIA A100 GPUs with 80G memory. In training the SPIN-Diffusion, we use the AdamW optimizer with a weight decay factor of $1e-2$. The images are processed at a 512×512 resolution. The batch size is set to 8 locally, alongside a gradient accumulation of 32. For the learning rate, we use a schedule starting with 200 warm-up steps, followed by linear decay. We conduct a grid search on the learning rate, coefficient β_t, and number of training steps, and choose the hyperparameters that perform the best on the validation set. We set the learning rate at $2.0e-5$ for the initial two iterations, reducing it to $5.0e-8$ for the third iteration. The coefficient β_t is chosen as 2000 for the first iteration, increasing to 5000 for the subsequent second and third iterations. The trend in different learning rate and β_t choices reveals that later iterations typically benefit from more conservative updates. Training steps are 50 for the first iteration, 500 for the second, and 200 for the third. In training the DPO model, we employ the same AdamW optimizer and maintain a batch size of 8 and a gradient accumulation of 32. The learning rate is set to $2.0e-5$, and β_t is set to 2000. The total number of training steps for DPO is 350. In SFT training, we use 4 NVIDIA A6000 GPUs. We use the AdamW optimizer with a weight decay of 0.01. The local batch size is set to 32 and the global batch size is set to 512. Our learning rate is 1e-5, with linear warmup for 500 steps with no learning rate decay. We save checkpoints every 500 steps and evaluate the checkpoints on Pick-a-Pic validation. We select the best checkpoint, trained after 2000 steps as our SFT checkpoint.

During generation, we use a guidance scale of 7.5, and fixed the random seed as 5775709.

A.2 Additional Results

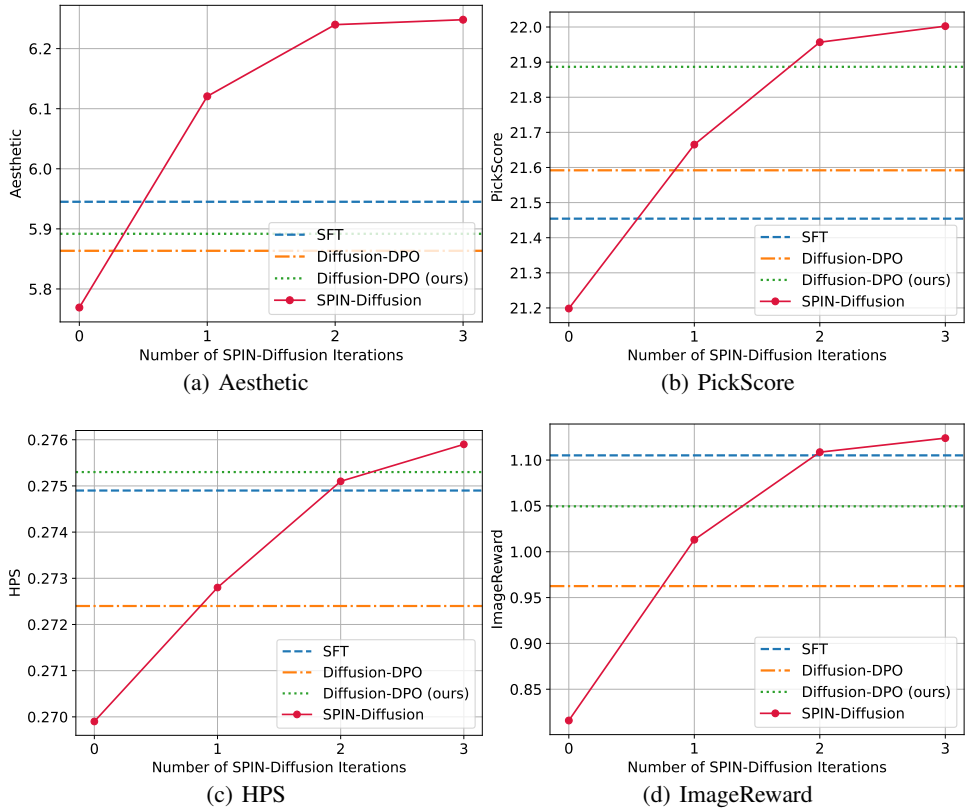
In this section, we first illustrate the main results shown in Table 1 by Figure 3 and a radar plot Figure 4.

We present the median scores of baselines and SPIN-Diffusion on Pick-a-Pic testset in Table 2. The results are consistent to the results in Table 1. We present the detailed winning rate of baselines and SPIN-Diffusion over SD-1.5 in Table 3 and the winning rate over Diffusion-DPO in Table 4. We present the aesthetic scores of the images in Figure 2 in Table 5.

A.3 Additional Ablation Study

We conduct ablation study to investigate several aspects in the performance of SPIN-Diffusion.

Continual Training for More Epochs. We further study the training behavior of SPIN-Diffusion by continual training within iteration 1. Both iteration 1 and iteration 2 commence training from the same checkpoint. However, for subsequent epochs in iteration 1, images generated by SD-1.5 are used, with SD-1.5 also serving as the opponent player. In contrast, during iteration 2, both the generated images and the opponent player originate from the iteration 1 checkpoint. The results shown in Figure 5 are reported on the 500 prompts validation set of Pick-a-Pic. We observe that in terms of PickScore, HPS, and average score, continual training on iteration 1 even results in a performance decay. Even in terms of Aesthetic score, continual training cannot guarantee a consistent improvement. Compared to training for more epochs in iteration 1, iteration 2 has a much more ideal performance. These results show the key role in updating the opponent.



(a) Aesthetic (b) PickScore

(c) HPS (d) ImageReward

Figure 3: Comparison between SPIN-Diffusion at different iterations with SD-1.5, SFT and Diffusion-DPO. SPIN-Diffusion outperforms SFT at iteration 1, and outperforms all the baselines after iteration 2. In the legend, Diffusion-DPO (ours) denotes our reproduced version of Diffusion-DPO.

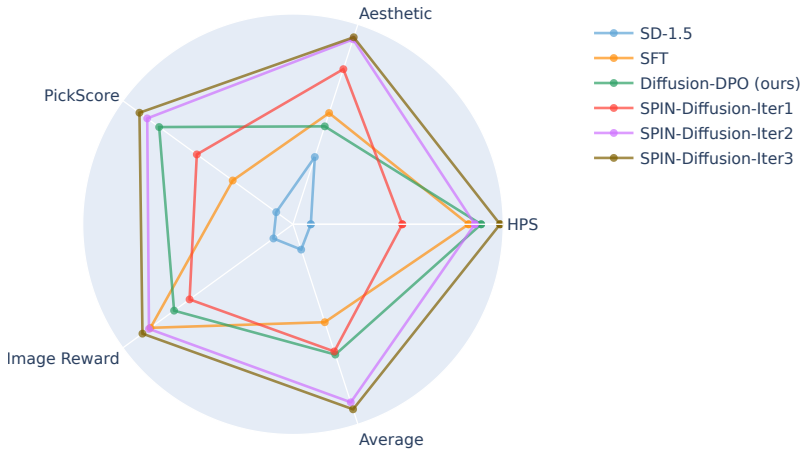


Figure 4: The main result is presented in radar chart. The scores are adjusted to be shown on the same scale. Compared with the baselines, SPIN achieves higher scores in all the four metrics and the average score by a large margin. In the legend, Diffusion-DPO (ours) denotes our reproduced version of Diffusion-DPO.

Evaluation on Other Benchmarks We also conduct experiment on PartiPrompts (Yu et al., 2022) and HPSv2 (Wu et al., 2023). PartiPrompts consist of 1632 prompts that contains a wide range of categories and difficulties that beyond daily scenarios and natural objects. HPSv2 is a text-image prefence dataset, where the prompts come from DiffusionDB and MSCOCO (Lin et al., 2014) dataset.

Table 2: The results of median scores on Pick-a-Pic test set. We report the median of PickScore, HPSv2, ImageReward and Aesthetic over the whole test set. We also report the average score over the four evaluation metric. SPIN-Diffusion outperforms all the baselines regarding HPS, Aesthetic, PickScore and the average score, which agrees with the results of mean scores.

Model	HPS ↑	Aesthetic ↑	ImageReward ↑	PickScore ↑	Average ↑
SD-1.5	0.2705	5.7726	0.9184	21.1813	7.0357
SFT (reproduced)	0.2750	5.9331	**1.3161**	21.4159	7.2350
Diffusion-DPO	0.2729	5.8837	1.1361	21.6064	7.2248
Diffusion-DPO (reproduced)	0.2756	5.8895	1.2219	21.8995	7.3216
SPIN-Diffusion-Iter1	0.2739	6.1297	1.1366	21.6464	7.2967
SPIN-Diffusion-Iter2	0.2751	6.2385	1.3059	22.0101	7.4574
SPIN-Diffusion-Iter3	**0.2761**	**6.2769**	1.3073	**22.0703**	**7.4827**

Table 3: The winning rate over SD-1.5 Pick-a-Pic testset. SPIN-Diffusion shows the highest winning rate over SD-1.5 among all the baselines.

Model	PickScore ↑	HPS ↑	ImageReward ↑	Aesthetic ↑	Average ↑
SFT (reproduced)	62.4	82.0	75.0	70.8	73.2
Diffusion-DPO	78.4	75.8	65.0	65.4	79.8
Diffusion-DPO (reproduced)	83.8	81.2	71.2	69.0	84.8
SPIN-Diffusion-Iter1	75.4	70.0	65.8	86.0	80.8
SPIN-Diffusion-Iter2	86.6	82.6	72.6	92.2	90.0
SPIN-Diffusion-Iter3	**87.0**	**86.2**	**77.0**	**93.8**	**91.6**

Table 4: The winning rate over Diffusion DPO (reproduced) on Pick-a-Pic testset. SPIN-Diffusion shows the highest winning rate over Diffusion DPO (reproduced) among all the baselines.

Model	PickScore ↑	HPS ↑	ImageReward ↑	Aesthetic ↑	Average ↑
SD-1.5	16.2	20.8	28.8	31.0	15.2
SFT (reproduced)	26.8	48.2	51.4	52.8	35.2
Diffusion-DPO	30.6	29.4	36.8	45.2	30.4
SPIN-Diffusion-Iter1	37.2	35.6	40.6	74.8	47.4
SPIN-Diffusion-Iter2	56.8	49.0	52.6	86.6	68.2
SPIN-Diffusion-Iter3	**62.4**	**56.2**	**55.8**	**86.8**	**70.2**

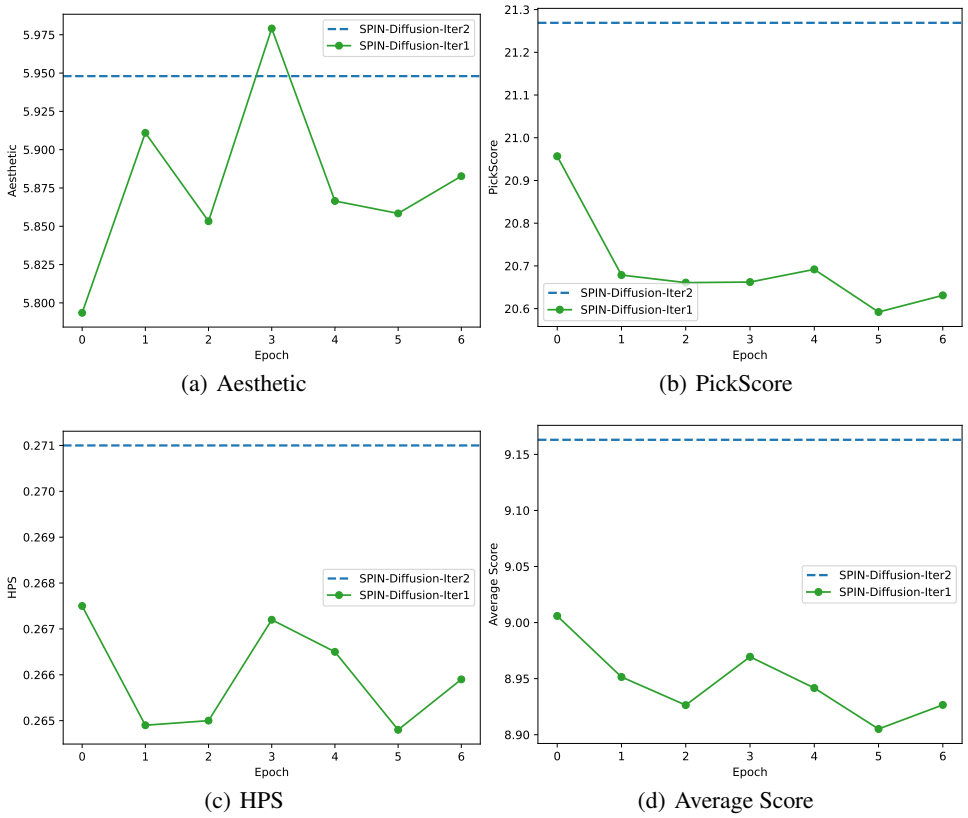
Table 5: Aesthetic scores of pictures in Figure 2

	SD-1.5	SFT	Diffusion-DPO (reproduced)	SPIN-Diffusion Iter1	Iter2	Iter3
Boy	6.171	6.096	6.072	6.158	6.407	6.831
Castle	6.180	6.346	5.995	6.886	6.993	6.940
Eagle	4.927	5.428	5.289	5.601	6.103	6.189

Table 6: The size of benchmark datasets in our evaluation

Benchmarks	Pick-a-Pic	PartiPrompts	HPSv2
# Prompts	500	1630	3200

In our experiment, we use the prompts from its test set, which contains 3200 prompts. We use the same evaluation metrics as before and the results are shown in Table 7 and 8. The results show that, on both PartiPrompts and HPSv2, SPIN-Diffusion achieves a comparable performance with Diffusion DPO (reproduced) and surpasses other baseline models at the first iteration. SPIN-Diffusion further



(a) Aesthetic (b) PickScore

(c) HPS (d) Average Score

Figure 5: The evaluation results on Pick-a-Pic validation set of continual training within SPIN-Diffusion iteration 1, and SPIN-Diffusion iteration 2. The x-axis is the number of epochs. Consecutive epochs in iteration 1 reach their limit quickly while switching to iteration 2 boosts the performance.

Table 7: The results of mean scores on PartiPrompts. We report the mean and median of PickScore, HPS, ImageReward and Aesthetic score over the whole dataset. We also report the average score over the four evaluation metrics. SPIN-Diffusion outperforms all the baselines in terms of four metrics.

Model	HPS ↑	Aesthetic ↑	ImageReward ↑	PickScore ↑	Average ↑
SD-1.5	0.2769	5.6721	0.9196	21.8926	7.1903
SFT (reproduced)	<u>0.2814</u>	5.8568	**1.1559**	21.9719	7.3165
Diffusion-DPO	**0.2815**	5.7758	<u>1.1495</u>	22.2723	7.3698
SPIN-Diffusion-Iter1	0.2783	5.9073	0.9952	22.1221	7.3257
SPIN-Diffusion-Iter2	0.2804	<u>6.0533</u>	1.0845	<u>22.3122</u>	<u>7.4326</u>
SPIN-Diffusion-Iter3	0.2813	**6.0534**	1.0893	**22.3435**	**7.4419**

reaches an average score of 7.4326 and 7.5244 on PartiPrompts and HPSv2 dataset respectively at second iteration, which outpuerforms all other baselines by a large margin. These results consolidate our statement that `SPIN` shows a superior performance over both SFT and DPO. We also conduct qualitative result on PartiPrompts and the results are shown in Figure 6.

Remarks on LoRA fine-tuning While LoRA is a parameter-efficient fine-tuning method that focuses on reducing trainable parameters under resource constraints, it is orthogonal to SPIN-Diffusion, which utilizes a self-play mechanism for fine-tuning. We also provide SFT (LoRA) fine-tuning results in Figure 11. We can see that full fine-tuning generally surpasses the performance of LoRA fine-tuning. Therefore, we leave the exploration of LoRA version of SPIN-Diffusion to future work.

Table 8: The results of median scores on PartiPrompts. We report the mean and median of PickScore, HPS, ImageReward and Aesthetic score over the whole dataset. We also report the average score over the four evaluation metrics. SPIN-Diffusion outperforms all the baselines in terms of four metrics.

Model	HPS ↑	Aesthetic ↑	ImageReward ↑	PickScore ↑	Average ↑
SD-1.5	0.2781	5.6823	1.1247	21.9339	7.2548
SFT (reproduced)	0.2781	5.6823	1.1247	21.9339	7.2548
Diffusion-DPO	0.2822	5.7820	**1.3823**	22.3251	7.4429
SPIN-Diffusion-Iter1	0.2793	5.8926	1.1906	22.1632	7.3814
SPIN-Diffusion-Iter2	0.2810	6.0400	1.2857	22.2998	7.4766
SPIN-Diffusion-Iter3	**0.2825**	**6.0480**	1.3095	**22.3361**	**7.4940**

Table 9: The results of mean scores on HPSv2. We report the mean and median of PickScore, HPS, ImageReward and Aesthetic score over the whole dataset. We also report the average score over the four evaluation metrics. SPIN-Diffusion outperforms all the baselines in terms of four metrics.

Model	HPS ↑	Aesthetic ↑	ImageReward ↑	PickScore ↑	Average ↑
SD-1.5	0.2783	5.9017	0.8548	21.4978	7.1332
SFT (reproduced)	0.2846	6.0378	**1.1547**	21.8549	7.333
Diffusion-DPO	0.2843	6.0306	1.1391	22.3012	7.4388
SPIN-Diffusion-Iter1	0.2804	6.1943	1.0133	21.8778	7.3415
SPIN-Diffusion-Iter2	0.2838	6.3403	1.1145	22.2994	7.5095
SPIN-Diffusion-Iter3	**0.2849**	**6.342**	1.1292	**22.3415**	**7.5244**

Table 10: The results of median scores on HPSv2. We report the mean and median of PickScore, HPS, ImageReward and Aesthetic score over the whole dataset. We also report the average score over the four evaluation metrics. SPIN-Diffusion outperforms all the baselines in terms of four metrics.

Model	HPS ↑	Aesthetic ↑	ImageReward ↑	PickScore ↑	Average ↑
SD-1.5	0.2781	5.8529	0.9324	21.4825	7.1365
SFT (reproduced)	0.2847	6.0057	1.308	21.8211	7.3549
Diffusion-DPO	0.2847	5.9878	**1.3085**	22.2854	7.4666
SPIN-Diffusion-Iter1	0.2803	6.1519	1.1331	21.858	7.3558
SPIN-Diffusion-Iter2	0.2839	**6.3401**	1.2711	22.2577	7.5382
SPIN-Diffusion-Iter3	**0.2849**	6.3296	1.2853	**22.3029**	**7.5507**

Table 11: The results of LoRA fine-tuning vs. full fine-tuning.

Method	HPS	Aesthetic	ImageReward	PickScore	Average
SFT (full)	0.2749	5.9451	1.1051	21.4542	7.1948
SFT (LoRA)	0.2745	5.8573	1.1393	21.4121	7.1708

A.4 Training Dynamics of SFT and DPO

We first study the training dynamic of SPIN-Diffusion in comparison with SFT and Diffusion-DPO, and we plot the results in Figure 7. We observe that after training with about 50k data, the performance of SFT stop improving and maintains at about 20.8 in PickScore, 0.270 in HPS, 5.6 in Aesthetic and 8.9 in average score. These results is significantly inferior to those achieved by SPIN-Diffusion, which achieves 21.2 in PickScore, 0.272 in HPS, 5.9 in Aesthetic and 9.1 in average score. Compared to Diffusion-DPO, SPIN-Diffusion achieves a superior performance without the loser image. These results demonstrate that self-play fine-tuning plays a key role in SPIN-Diffusion's performance.

Figure 6: We show the images generated by different models based on prompts from PartiPrompts. The prompts are "*a photo of san francisco's golden gate bridge*", "*an aerial view of the Great Wall*" and "*Face of an orange frog in cartoon style*". The models are: SD-1.5, SFT, Diffusion-DPO, Diffusion-DPO (reproduced), SPIN-Diffusion-Iter2 from left to right. SPIN-Diffusion demonstrates a notable improvement in image quality

B Additional Qualitative Results

In this section, we provide extensive qualitative results to further support our findings. We first demonstrate the impact of different random seeds on model comparisons and include a wider range of visual examples to support the qualitative results. We also provide a image gallary in Figure 12.

Effect of different random seeds Random seeds sometimes influence the results produced by image generation models. Figures 8 and 9 demonstrate this effect, showcasing outputs of multiple models for the same prompt across four different random seeds. SPIN-Diffusion consistently generates higher-quality images across these variations.

More examples on Partiprompts To further showcase SPIN-Diffusion's capabilities to handle a wide range of styles and subjects, we present results on 10 additional prompts from the PartiPrompts dataset (totaling 1630 prompts). These examples highlight the model's ability to handle a wide range of styles and subjects. Figure 10 showcases results for 5 of these prompts, while Figure 11 highlights SPIN-Diffusion's ability in generating cartoon-style images with 5 additional prompts specifically containing the word 'cartoon'.

C Additional Details for SPIN-Diffusion

C.1 Additional Details of DDIM.

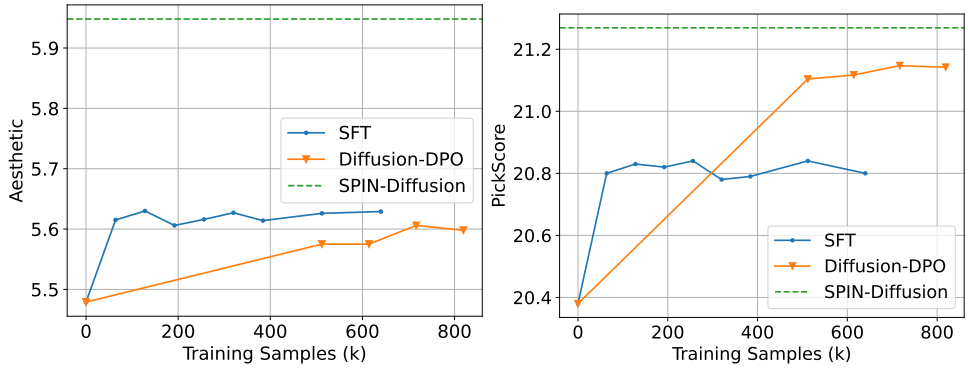
Given a prompt $\mathbf{c}$, image $\mathbf{x}_0$, sequence $\{\alpha_t\}_{t=1}^T \subseteq (0,1]$ and $\{\sigma_t\}_{t=1}^T \subseteq [0,+\infty)$, the forward diffusion process defined in (3.1) is

$$q(\mathbf{x}_{1:T}|\boldsymbol{x}_0) := q(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0),$$

where $q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_T}\mathbf{x}_0,(1-\alpha_T)\mathbf{I})$ and $q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$ admits the following distribution,

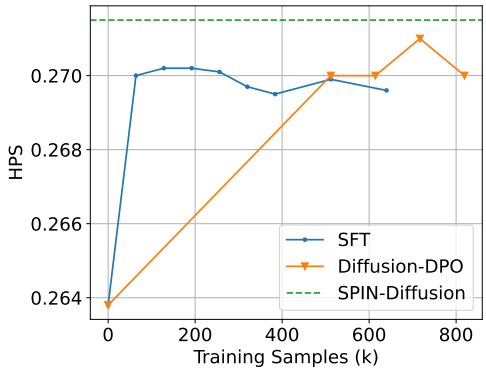
$$\mathcal{N}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}},\sigma_t^2\mathbf{I}\right). \tag{C.1}$$

Here $\{\alpha_t\}_{t=1}^T$ is a decreasing sequence with $\alpha_0 = 1$ and α_T approximately zero. By Bayesian rule, we can show that this diffusion process ensures that $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0,(1-$

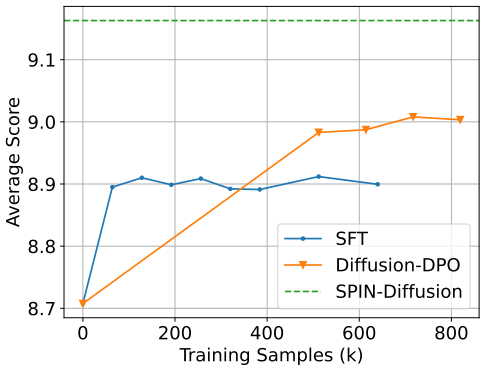


(a) Aesthetic

(b) PickScore



(c) HPS



(d) Average Score

Figure 7: The evaluation results on the Pick-a-Pic validation set of SFT, Diffusion-DPO and SPIN-Diffusion. The x-axis is the number of training data. SFT reaches its limit quickly, while Diffusion-DPO and SPIN-Diffusion continue to improve after training with over 800k data.

$\alpha_t)\mathbf{I})$ for all t and reduces to DDPM (Ho et al., 2020) with a special choice of $\sigma_t = \sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{(1-\alpha_t/\alpha_{t-1})}$.

Given noise schedule α_t and σ_t, examples from the generative model follows

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}|\mathbf{c}) = \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) \cdot p_{\boldsymbol{\theta}}(\mathbf{x}_T|\mathbf{c}),$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{c}) = \mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t), \sigma_t^2 \mathbf{I}\big).$$

Here $\boldsymbol{\theta}$ belongs to the parameter space $\boldsymbol{\Theta}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)$ is the mean of the Gaussian that can be parameterized (Ho et al., 2020; Song et al., 2020a) as

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t) = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t), \tag{C.2}$$

where $\{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\}_{t=1}^T$ are score functions that approximate noise. Compare (C.2) and (C.1), we can see that $\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)}{\sqrt{\alpha_t}}\right)$ approximates $\mathbf{x}_0$, and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ approximates the noise $\boldsymbol{\epsilon}_t := \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}} \sim \mathcal{N}(0, \mathbf{I})$.

C.2 Decoupling Technique

In Section 4, we demonstrate that the objective function defined in (4.8) can be simplified to the form in (4.9). This reformulation only requires considering two consecutive sampling steps, $t-1$ and t, rather than involving all intermediate steps. Now, we provide a detailed derivation.

Proof of Lemma 4.1.

$L_{\text{SPIN}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k)$

Figure 8: We show the figures generated by different models based on a prompt from Pick-A-Pic test set. The prompt used is "*a picture of the sea on which a boat sails in a storm and sways in the sea*". The models are SD-1.5, SFT, Diffusion-DPO (reproduced), SPIN-Diffusion-Iter1, SPIN-Diffusion-Iter2, and SPIN-Diffusion-Iter3, displayed from left to right. Each row shows the results for a different random seed. SPIN-Diffusion demonstrates a notable improvement in image quality

$$
\begin{aligned}
&= \mathbb{E}_{\mathbf{c}\sim q(\cdot),\mathbf{x}_{0:T}\sim p_{\text{data}}(\cdot|\mathbf{c}),\mathbf{x}'_{0:T}\sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})}\Bigg[\ell\Bigg(-\sum_{t=1}^{T}\frac{\beta_t}{T}\Big[\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2-\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2\\
&\quad-\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2+\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2\Big]\Bigg)\Bigg]\\
&\le \mathbb{E}_{\mathbf{c}\sim q(\cdot),\mathbf{x}_{0:T}\sim p_{\text{data}}(\cdot|\mathbf{c}),\mathbf{x}'_{0:T}\sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})}\Bigg[\frac{1}{T}\sum_{t=1}^{T}\ell\Bigg(-\beta_t\Big[\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2-\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2\\
&\quad-\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2+\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2\Big]\Bigg)\Bigg]\\
&= \mathbb{E}_{\mathbf{c}\sim q(\cdot),\mathbf{x}_{0:T}\sim p_{\text{data}}(\cdot|\mathbf{c}),\mathbf{x}'_{0:T}\sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c}),t\sim\text{Uniform}\{1,\dots,T\}}\Bigg[\ell\Bigg(-\beta_t\Big[\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2\\
&\quad-\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2-\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2+\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2\Big]\Bigg)\Bigg]\\
&= \mathbb{E}_{\mathbf{c}\sim q(\mathbf{c}),(\mathbf{x}_{t-1},\mathbf{x}_t)\sim p_{\text{data}}(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{c}),(\mathbf{x}'_{t-1},\mathbf{x}'_t)\sim p_{\boldsymbol{\theta}_k}(\mathbf{x}'_{t-1},\mathbf{x}'_t|\mathbf{c}),t\sim\text{Uniform}\{1,\dots,T\}.}\Bigg[\ell\Bigg(-\beta_t\Big[\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2\\
&\quad-\big\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t,\mathbf{c},t)\big\|_2^2-\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2+\big\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)\big\|_2^2\Big]\Bigg)\Bigg]\\
&= L_{\text{SPIN}}^{\text{approx}}(\boldsymbol{\theta},\boldsymbol{\theta}_k),
\end{aligned}
$$

where the first inequality is by Jensen's inequality and the convexity of the function ℓ, the second equality is by integrating the average $\frac{1}{T}\sum_{t=1}^{T}$ into the expectation via $t\sim\text{Uniform}\{1,\dots,T\}$, and the third inequality holds because the argument inside the expectation is only depend of sampling step $t-1$ and t. □

Figure 9: We show the figures generated by different models based on a prompt from Pick-A-Pic test set. The prompt used is "*A cute hedgehog holding flowers*". The models are SD-1.5, SFT, Diffusion-DPO (reproduced), SPIN-Diffusion-Iter1, SPIN-Diffusion-Iter2, and SPIN-Diffusion-Iter3, displayed from left to right. Each row shows the results for a different random seed. SPIN-Diffusion demonstrates a notable improvement in image quality

C.3 Objective Function of SPIN-Diffusion

We look deep into the term $\left\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\right\|_2^2$ and $\left\|\mathbf{x}'_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t,\mathbf{c},t)\right\|_2^2$ of (4.8) and (4.9) in this section.

When $\mathbf{x}_{0:T}$ Follows Forward Process. We have that $\mathbf{x}_{0:T}\sim p_{\text{data}}(\cdot|\mathbf{c})$ and by (C.1) and (C.2) we have that

$$\mathbf{x}_{t-1}=\sqrt{\alpha_{t-1}}\mathbf{x}_0+\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}+\sigma_t\widehat{\boldsymbol{\epsilon}}_t$$

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)=\sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t-\sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)}{\sqrt{\alpha_t}}\right)+\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t),$$

where $\widehat{\boldsymbol{\epsilon}}_t\sim\mathcal{N}(0,\mathbf{I})$. Therefore, $\left\|\mathbf{x}_{t-1}-\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\right\|_2^2$ can be simplified to

$$h_t^2\left\|\frac{\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)+(\sigma_t/h_t)\cdot\widehat{\boldsymbol{\epsilon}}_t\right\|_2^2, \tag{C.3}$$

where $h_t=\left[\sqrt{1-\alpha_{t-1}-\sigma_t^2}-\sqrt{\alpha_{t-1}/\alpha_t}\sqrt{1-\alpha_{t-1}}\right]$ and $\frac{\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}\sim\mathcal{N}(0,\mathbf{I})$ following a Gaussian distribution. When $\sigma_t\to 0$, (C.3) becomes $h_t^2\left\|\boldsymbol{\epsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,\mathbf{c},t)\right\|_2^2$ with $h_t=\left[\sqrt{1-\alpha_{t-1}}-\sqrt{\alpha_{t-1}/\alpha_t}\sqrt{1-\alpha_{t-1}}\right]$ and $\boldsymbol{\epsilon}_t:=\frac{\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1-\alpha_t}}\sim\mathcal{N}(0,\mathbf{I})$.

When $\mathbf{x}'_{0:T}$ Follows the Backward Process. We have that $\mathbf{x}'_{0:T}\sim p_{\boldsymbol{\theta}_k}(\cdot|\mathbf{c})$ and

$$\begin{aligned}\mathbf{x}'_{t-1}&=\boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)+\sigma_t\widehat{\boldsymbol{\epsilon}}'_t\\&=\sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}'_t-\sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)}{\sqrt{\alpha_t}}\right)+\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t,\mathbf{c},t)+\sigma_t\widehat{\boldsymbol{\epsilon}}'_t\end{aligned}$$

Figure 10: We show the figures generated by different models based on prompts from PartiPrompts. The prompts are "*a old-time car with a large front grille*", "*a full moon rising above a mountain at night*", "*a young badger delicately sniffing a yellow rose, richly textured oil painting*", "*a cartoon of a man standing under a tree*" and "*a prop plane flying low over the Great Wall*". The models are: SD-1.5, SFT, Diffusion-DPO, Diffusion-DPO (reproduced), SPIN-Diffusion-Iter2 from left to right, all utilizing the same random seed for fair comparison

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t) = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}'_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t),$$

where $\boldsymbol{\epsilon}'_t \sim \mathcal{N}(0, \mathbf{I})$. Therefore, $\left\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\right\|_2^2$ can be simplified to

$$h_t^2\left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t) + (\sigma_t/h_t)\cdot\widehat{\boldsymbol{\epsilon}}'_t\right\|_2^2, \tag{C.4}$$

where $h_t = \left[\sqrt{1-\alpha_{t-1}-\sigma_t^2} - \sqrt{\alpha_{t-1}/\alpha_t}\sqrt{1-\alpha_{t-1}}\right]$. When $\sigma_t \to 0$, (C.4) becomes $h_t^2\left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\right\|_2^2$ with $h_t = \left[\sqrt{1-\alpha_{t-1}} - \sqrt{\alpha_{t-1}/\alpha_t}\sqrt{1-\alpha_{t-1}}\right]$.

Simple Decoupled SPIN-Diffusion Objective Function. Substituting (C.3) and (C.4) into (4.9) and applying $\sigma_t \to 0$ yields,

$$L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \mathbb{E}\Bigg[\ell\Bigg(-\beta_t h_t^2\Big[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\right\|_2^2 - \left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\right\|_2^2$$

Figure 11: We show the figures generated by different models based on prompts from PartiPrompts. The prompts are "*A cartoon house with red roof*", "*a cartoon of an angry shark*", "*a cartoon of a bear birthday party*", "*a cartoon of a house on a mountain*" and "*a cartoon of a boy playing with a tiger*". The models are: SD-1.5, SFT, Diffusion-DPO, Diffusion-DPO (reproduced), SPIN-Diffusion-Iter2 from left to right, all utilizing the same random seed for fair comparison

$$- \left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t', \mathbf{c}, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t', \mathbf{c}, t)\right\|_2^2\Big]\Big)\Big], \tag{C.5}$$

where $h_t = \sqrt{1-\alpha_{t-1}} - \sqrt{\alpha_{t-1}/\alpha_t}\sqrt{1-\alpha_{t-1}}$, $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + (1-\alpha_t)\boldsymbol{\epsilon}_t$, and the expectation is computed over the distribution,$\mathbf{c} \sim q(\mathbf{c}), \mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0|\mathbf{c}), \mathbf{x}_t' \sim p_{\boldsymbol{\theta}_k}(\mathbf{x}_t'|\mathbf{c})$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim \text{Uniform}\{1, \ldots, T\}$. (C.5) still need the intermediate steps $\mathbf{x}_t'$, as discussed below (4.9) in Section 4, we can approximate the backward process with the forward process and obtain

$$L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \mathbb{E}\Big[\ell\Big(-\beta_t h_t^2\Big[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\right\|_2^2 - \left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\right\|_2^2 - \left\|\boldsymbol{\epsilon}_t' - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t', \mathbf{c}, t)\right\|_2^2$$
$$+ \left\|\boldsymbol{\epsilon}_t' - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t', \mathbf{c}, t)\right\|_2^2\Big]\Big)\Big],$$

where $h_t = \sqrt{1-\alpha_{t-1}} - \sqrt{\alpha_{t-1}/\alpha_t}\sqrt{1-\alpha_{t-1}}$, $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + (1-\alpha_t)\boldsymbol{\epsilon}_t$, $\mathbf{x}_t' = \alpha_t \mathbf{x}_0' + (1-\alpha_t)\boldsymbol{\epsilon}_t'$, and the expectation is computed over the distribution,$\mathbf{c} \sim q(\mathbf{c}), \mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0|\mathbf{c}), \mathbf{x}_0' \sim p_{\boldsymbol{\theta}_k}(\mathbf{x}_0'|\mathbf{c})$, $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$, $\boldsymbol{\epsilon}_t' \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim \text{Uniform}\{1, \ldots, T\}$.

Figure 12: Image galary generated by SPIN-Diffusion, a self-play fine-tuning algorithm for diffusion models. The results are fine-tuned from Stable Diffusion v1.5 on the winner images of the Pick-a-Pic dataset. The prompts used for generating the above images are chosen from the Pick-a-Pic test set. The generated images demonstrate superior performance in terms of overall visual attractiveness and coherence with the prompts. SPIN-Diffusion is featured by its independence from paired human preference data, offering a useful tool for fine-tuning on custom datasets with only single image per text prompt provided.

D Proof of Theorems in Section 5

Proof of Theorem 5.2. Plugging (C.3) and (C.4) into (4.9) yields the following loss parameterized with $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$,

$$L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \mathbb{E}\Bigg[\ell\Bigg(- \beta_t h_t^2\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 - \big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 + \big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t'\big\|_2^2 - \big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t', c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t', c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t'\big\|_2^2\Big]\Bigg)\Bigg], \quad \text{(D.1)}$$

where $\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t' \sim \mathcal{N}(0, \mathbf{I})$. When $\sigma_t \to 0$, (D.1) can be simplified to (C.5). In this proof, we will stick to the formula (D.1) to provide the proof for all $\sigma_t \geq 0$.

Since $\boldsymbol{\theta}_k$ is not the global optimum of L_{DM} by condition, there exists $\boldsymbol{\theta}^*$ such that $L_{\text{DM}}(\boldsymbol{\theta}^*) \leq L_{\text{DM}}(\boldsymbol{\theta}_k)$, which gives that

$$\mathbb{E}\Big[\gamma_t\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t, c) - \boldsymbol{\epsilon}\big\|_2^2\Big] \leq \mathbb{E}\Big[\gamma_t\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, t, c) - \boldsymbol{\epsilon}\big\|_2^2\Big], \quad \text{(D.2)}$$

where the expectation is computed over the distribution $c \sim q(\cdot), \mathbf{x}_0 \sim q_{\text{data}}(\cdot|c), \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $t \sim \text{Uniform}\{1, \ldots, T\}$. Define $g(s) = L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}^*, \boldsymbol{\theta}_k)$ with $\beta_t = s\gamma_t/h_t^2$ as follows,

$$\begin{aligned} g(s) &= \mathbb{E}\Bigg[\ell\Bigg(- \beta_t h_t^2\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 - \big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \\ &\qquad + \big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t'\big\|_2^2 - \big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t', c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t', c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t'\big\|_2^2\Big]\Bigg)\Bigg] \\ &= \mathbb{E}\Bigg[\ell\Bigg(- s\gamma_t\Big[\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 - \big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \\ &\qquad + \big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t'\big\|_2^2 - \big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t', c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t', c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t'\big\|_2^2\Big]\Bigg)\Bigg]. \end{aligned}$$

Then we have that $g(0) = 0$ and

$$\begin{aligned}\frac{dg}{ds}(0) = \mathbb{E}\Bigg[&- \ell'(0)\gamma_t\Bigg(\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 - \big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \\ &+ \big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}'_t\big\|_2^2 - \big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}'_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}'_t\big\|_2^2\Bigg)\Bigg] \\ = -\ell'(0)\Bigg(&\mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 - \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \\ &+ \mathbb{E}\gamma_t\big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}'_t\big\|_2^2 - \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}'_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}'_t\big\|_2^2\Bigg), \end{aligned} \tag{D.3}$$

where $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$ and the expectation is computed over the distribution $c \sim q(\cdot), \mathbf{x}_0 \sim q_{\mathrm{data}}(\cdot|c), \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, $t \sim \mathrm{Uniform}\{1, \ldots, T\}$ and $\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}'_t \sim \mathcal{N}(0, \mathbf{I})$. Since $\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}'_t$ follows standard Multivariate normal distribution and independent with $\mathbf{x}_t, \mathbf{x}'_t, \boldsymbol{\epsilon}, \mathbf{x}_0$, we can simplify the the terms in (D.3) as follows,

$$\begin{aligned}&\mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \\ &= \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t)\big\|_2^2 + \mathbb{E}\gamma_t\big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \end{aligned} \tag{D.4}$$

$$\begin{aligned}&\mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \\ &= \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t)\big\|_2^2 + \mathbb{E}\gamma_t\big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}_t\big\|_2^2 \end{aligned} \tag{D.5}$$

$$\begin{aligned}&\mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}'_t, c, t) + (\sigma_t/h_t)\cdot\boldsymbol{\epsilon}'_t\big\|_2^2 \\ &= \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}'_t, c, t)\big\|_2^2 + \mathbb{E}\gamma_t\big\|(\sigma_t/h_t)\cdot\boldsymbol{\epsilon}'_t\big\|_2^2 \end{aligned} \tag{D.6}$$

where we apply the property of standard normal distribution that $\mathbb{E}[\boldsymbol{\epsilon}_t] = \mathbb{E}[\boldsymbol{\epsilon}'_t] = \mathbf{0}$. Plugging (D.4), (D.5), (D.6) into (D.3) gives that

$$\begin{aligned}\frac{dg}{ds}(0) &= -\ell'(0)\Bigg(\mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, c, t)\big\|_2^2 - \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, c, t)\big\|_2^2 \\ &\qquad - \mathbb{E}\gamma_t\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, c, t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}(\mathbf{x}'_t, c, t)\big\|_2^2\Bigg) \\ &< 0, \end{aligned}$$

where the last inequality is by (D.2). Here $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$ and the expectation is computed over the distribution $c \sim q(\cdot), \mathbf{x}_0 \sim q_{\mathrm{data}}(\cdot|c), \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), t \sim \mathrm{Uniform}\{1, \ldots, T\}$.

Therefore, there exist a λ_0 such that for all $0 < \lambda < \lambda_0$, we have $g(\lambda) < \ell(0)$. So for those $\beta_t = s\gamma_t/h_t^2$ with $0 < \lambda < \lambda_0$, we have that

$$L_{\texttt{SPIN}}^{\mathrm{approx}}(\boldsymbol{\theta}^*, \boldsymbol{\theta}_k) = g(\lambda) < g(0) = L_{\texttt{SPIN}}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k),$$

where the inequality holds due to the choice of λ. Therefore, we conclude that $\boldsymbol{\theta}_k$ is not the global optimum of (4.9).

□

Proof of Theorem 5.3. By (4.9) we have that,

$$\begin{aligned}L_{\texttt{SPIN}}^{\mathrm{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \mathbb{E}\Bigg[\ell\Bigg(&- \beta_t\Big[\big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 - \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 \\ &- \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 + \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2\Big]^2\Bigg)\Bigg], \end{aligned}$$

where the expectationis computed over the distribution $\mathbf{c} \sim q(\mathbf{c})$, $(\mathbf{x}_{t-1}, \mathbf{x}_t) \sim \int p_{\mathrm{data}}(\mathbf{x}_0|\mathbf{c})q(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{x}_0)d\mathbf{x}_0$, $(\mathbf{x}'_{t-1}, \mathbf{x}'_t) \sim \int p_{\boldsymbol{\theta}_k}(\mathbf{x}'_0|\mathbf{c})q(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0)d\mathbf{x}'_0$, $t \sim \mathrm{Uniform}\{1, \ldots, T\}$. Since $p_{\mathrm{data}}(\cdot|\mathbf{c}) = p_{\boldsymbol{\theta}_t}(\cdot|\mathbf{c})$, we can conclude that $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ and $(\mathbf{x}'_{t-1}, \mathbf{x}'_t)$ are independent and identically distributed random variable. Therefore, by symmetry property of $(\mathbf{x}_{t-1}, \mathbf{x}_t)$ and $(\mathbf{x}'_{t-1}, \mathbf{x}'_t)$, we have for any $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ that

$$
\begin{aligned}
L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) = \frac{1}{2}\mathbb{E}\Bigg[\ell\Bigg(& -\beta_t\Big[\big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 - \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 \\
& - \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 + \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2\Big]\Bigg) \\
& + \ell\Bigg(-\beta_t\Big[\big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 - \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 \\
& - \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 + \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2\Big]\Bigg)\Bigg] \\
\geq \mathbb{E}\Bigg[\ell\Bigg(& -\frac{\beta_t}{2}\Big[\big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 - \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 \\
& - \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 + \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2\Big] \\
& - \frac{\beta_t}{2}\Big[\big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 - \big\|\mathbf{x}'_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}'_t, \mathbf{c}, t)\big\|_2^2 \\
& - \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2 + \big\|\mathbf{x}_{t-1} - \boldsymbol{\mu}_{\boldsymbol{\theta}_k}(\mathbf{x}_t, \mathbf{c}, t)\big\|_2^2\Big]\Bigg)\Bigg] \\
= \ell(0),
\end{aligned}
$$

where the inequality is due to Jensen's inequality (recalling that ℓ is convex in Assumption 5.1), and the expectation is computed over the distribution $\mathbf{c} \sim q(\mathbf{c})$, $(\mathbf{x}_{t-1}, \mathbf{x}_t) \sim \int p_{\text{data}}(\mathbf{x}_0|\mathbf{c})q(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{x}_0)d\mathbf{x}_0$, $(\mathbf{x}'_{t-1}, \mathbf{x}'_t) \sim \int p_{\boldsymbol{\theta}_k}(\mathbf{x}'_0|\mathbf{c})q(\mathbf{x}'_{t-1}, \mathbf{x}'_t|\mathbf{x}'_0)d\mathbf{x}'_0$, $t \sim \text{Uniform}\{1, \ldots, T\}$. Therefore, we have that

$$
L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}, \boldsymbol{\theta}_k) \geq \ell(0) = L_{\texttt{SPIN}}^{\text{approx}}(\boldsymbol{\theta}_k, \boldsymbol{\theta}_k),
$$

which means that $\boldsymbol{\theta}_k$ is the global optimum of (4.9). As a consequence, $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k$. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We validate our claim through theoretical analysis and experimental results.

   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: [Yes]

   Justification: We discuss the limitations after our conclusion.

   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: [Yes]

   Justification: The assumptions and proof are fully presented.

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: [Yes]

   Justification: We provide a detailed descriptions on model, data, pipeline and parameters.

   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: [No]

Justification: We are not able to reorganize the code at the time of submission.

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: [Yes]

   Justification: We provide the training methods in detail.

   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: [No]

   Justification: Error bars are not involved in our paper. However, we provide the qualitative results of different random seed.

   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: [Yes]

   Justification: We describe the computing resource with experimental settings.

   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: [Yes]

   Justification: The research conducted in this paper conform with the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: We place our broader impact section at the start of the Appendix.

    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: [NA]

    Justification: This paper proposes fine-tuning methodology that is generally applicable to any pretrained language model and preference model, and poses no particular such risks.

    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: [Yes]

    Justification: We cite the original papers that produced the dataset and base models.

    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: [NA]

    Justification: The paper does not release new assets

    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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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.