TRAINING-FREE DIFFUSION MODEL ALIGNMENT WITH SAMPLING DEMONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Aligning diffusion models with user preferences has been a key challenge. Existing methods for aligning diffusion models either require retraining or are limited to differentiable reward functions. To address these limitations, we propose a stochastic optimization approach, dubbed *Demon*, to guide the denoising process at inference time without backpropagation through reward functions or model retraining. Our approach works by controlling noise distribution in denoising steps to concentrate density on regions corresponding to high rewards through stochastic optimization. We provide comprehensive theoretical and empirical evidence to support and validate our approach, including experiments that use non-differentiable sources of rewards such as Visual-Language Model (VLM) APIs and human judgements. To the best of our knowledge, the proposed approach is the first inference-time, backpropagation-free preference alignment method for diffusion models. Our method can be easily integrated with existing diffusion models without further training. Our experiments show that the proposed approach significantly improves the average aesthetics scores for text-to-image generation.

1 INTRODUCTION

Diffusion models have been the state-of-the-art for image generation (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Karras et al., 2022; Saharia et al., 2022; Rombach et al., 2022), but, commonly, the end users' preferences and intention diverge from the data distribution on which the model was trained. Aligning diffusion models with diverse user preferences is an ongoing and critical area of research.

One approach to aligning diffusion models with user preferences is to fine-tune using reinforcement learning (RL) to optimize the models based on rewards signals that reflect the user preferences (Black et al., 2023; Fan et al., 2023). However, retraining the model every time when the preference changes is computationally expensive and time-consuming.

An alternative approach is to guide the denoising process using a differentiable reward function. This can be done through classifier guidance at inference time (Dhariwal & Nichol, 2021; Wallace et al., 2023b; Bansal et al., 2024) or backpropagation at training time (Prabhudesai et al., 2024; Clark et al., 2024; Xu et al., 2023). These methods are generally less resource-demanding and more efficient. While these methods are generally more efficient, they require the reward function to be differentiable. This limits the types of reward sources that can be used, as it excludes the non-differentiable sources like third-party Visual-Language Model (VLM) APIs and human judgements.

To address these limitations, we propose *Demon*, a novel stochastic optimization approach for preference optimization of diffusion models at inference time. Demon is a metaphor from Maxwell's Demon, an imaginary manipulator of natural thermodynamic processes. The core ideas are: (1) Quality of noises that seed different possible backward steps in a discretized reverse-time Stochastic Differential Equation (SDE) can be evaluated given a reward source; (2) Such evaluation enables us to synthesize "optimal" noises that theoretically and empirically improve the final reward of the generated image through stochastic optimization. Specifically, we leverage Probability Flow Ordinary Differential Equation (PF-ODE) (Song et al., 2021) or Consistency Model (CM) (Song et al., 2023; Luo et al., 2023) to help us efficiently evaluate the possible backward steps, seeded with different Gaussian noises.

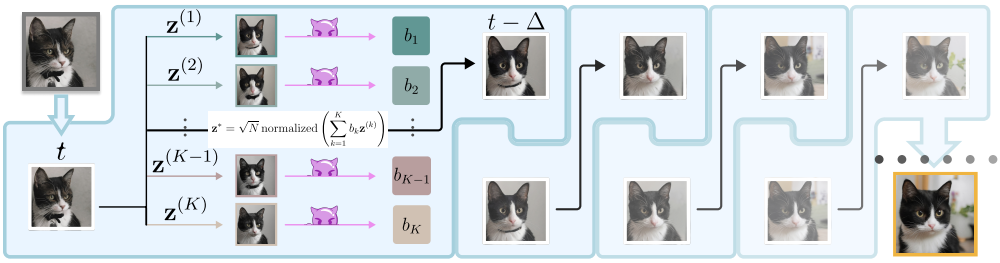


Figure 1: **Illustration of Demon.** Given a reverse-time SDE for denoising and an interval $[t_{\text{max}}, t_{\text{min}}]$, we first discretize it into T steps, $t_{\text{max}} > \cdots > t > t - \Delta > \cdots > t_{\text{min}}$. At every reverse-time denoising step, from t to $t - \Delta$, we synthesize an "optimal" noise $\mathbf{z}^*$ from K i.i.d. noises w.r.t a given reward source and use $\mathbf{z}^*$ to seed the step. This enables guiding the denoising process towards generating images that are more aligned with the reward source and the preference that the reward source represents. More details are presented in Section 4.

Our key contributions are summarized as follows:

- Our approach enables the use of reward signals in the denoising process regardless of whether the reward function is differentiable. This allows for the incorporation of previously inaccessible reward sources, such as VLM APIs. To the best of our knowledge, this is the first inference-time, backpropagation-free preference alignment method.
- Our method can be easily integrated with existing diffusion models in a plug-and-play fashion without retraining or fine-tuning.
- We provide a theoretical explanation for why our approach can improve the given reward function for image generation, which can be exploited for tuning hyperparameters.
- We demonstrate that our approach significantly improves the average aesthetics score (LAION, 2023) of Stable Diffusion models, achieving averages well above 8.0 compared to the Best-of-N random sampling upper bounds of 6.5 for SD v1.4 and 7 for SDXL. This improvement is achieved across various text-to-image generation tasks using prompts from prior work (Black et al., 2023), without relying on backpropagation-based preference alignment or model retraining.

2 Related Work

Diffusion Model. Diffusion models for data generation were first proposed by Sohl-Dickstein et al. (2015), further developed for high-fidelity image generation by Ho et al. (2020), and generalized by Song et al. (2021) through the lens of SDEs. Karras et al. (2022) comprehensively studied the design space of Diffusion SDEs. In this work, we base many of the derivations on theirs. Furthermore, we focus on evaluating our method in the text-to-image generation setting (Rombach et al., 2022; Ho & Salimans, 2021; Podell et al., 2024)

Human Preference Alignment. Aligning models with human preferences has been studied with several approaches:reinforcement learning-based policy optimization (Fan et al., 2023; Yang et al., 2024; Black et al., 2023); training with reward backpropagation (Clark et al., 2024; Xu et al., 2023); backpropagation through the reward model and the diffusion chain (Prabhudesai et al., 2024; Wallace et al., 2023b; Bansal et al., 2024). Many metrics and benchmarks for evaluating alignment has also been proposed, including those by Xu et al. (2023); Kirstain et al. (2023); LAION (2023); Wu et al. (2023), and we use these either as optimization objectives or evaluation of the generated image. In Table 1, we further provide detailed comparisons of the proposed Demon approach with relevant existing methods in the literature from different aspects.

Table 1: A detailed comparison of different methods along various dimensions, including the ability to generalize to an open vocabulary, the necessity of a backpropagation signal for optimization, the method's capacity to avoid mode collapse and ensure distributional guarantees (Divergence Control). Our proposed method stands out for its zero-shot learning capabilities.

Type	**Methods**	**Open Vocab**	**Non-Backprop Objective**	**Divergence Control**
Training	DPOK (Fan et al., 2023)	×	✓	✓
Training	DDPO (Black et al., 2023)	×	✓	×
Inference	DOODL (Wallace et al., 2023b)	✓	×	×
Training	DPO (Wallace et al., 2023a)	✓	✓	✓
Training	DRaFT (Clark et al., 2024)	✓	×	×
Inference	Demon	✓	✓	✓

3 Preliminary

Score-Based Diffusion Model. We base our derivation on EDM (Karras et al., 2022). With a sampling schedule $\sigma_t = t$, we can write the reverse-time SDE sampling towards the diffusion marginal distribution as follows.

$$\mathrm{d}\mathbf{x}_t = \underbrace{\left[-t\nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t, t\right) - \beta t^2 \nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t, t\right)\right]}_{\mathbf{f}_\beta(\mathbf{x}_t, t)} \mathrm{d}t + \underbrace{\sqrt{2\beta}t}_{g_\beta(t)} \mathrm{d}\omega_t, \tag{1}$$

where $p(\mathbf{x}_t, t) = p(\mathbf{x}_0, 0) \otimes \mathcal{N}\left(\mathbf{0}, t^2 \boldsymbol{I}_n\right)$ and $\otimes$ denotes the convolution operation. $\mathbf{x}_0$ is a clean sample, $\mathbf{x}_0 \sim p_{\mathrm{data}}$, and $\mathbf{x}_t$ is a noisy sample at time t. β expresses the relative rate at which existing noise is injected with new noise. In EDM, β is a function of t, but in our study, we set β to a constant for all t for simplicity. Essentially, $\mathbf{f}_\beta(\mathbf{x}, t)$ corresponds to drift and $g_\beta(t)$ corresponds to diffusion. As common in diffusion models, since $p(\mathbf{x}_t, t) \approx \mathcal{N}(\mathbf{0}, t^2 \mathbf{I}_N)$ for a large enough t, we sample $\mathbf{x}_{t_{\max}} \sim \mathcal{N}(\mathbf{0}, t_{\max}^2 \boldsymbol{I}_N)$ as the initial sample.

A comprehensive list of the notations and conventions used in this paper is provided at Appendix A.

4 Reward-Guided Denoising with Demons

In this section, we describe how Demon works in two steps: Section 4.1 explains the process of scoring Gaussian noises in reverse-time SDE with a reward function; Section 4.2 further explains how the noise scoring allows us to guide the denoising process to align with the reward function, which is what we refer to as *Demon*.

4.1 Scoring Noises in Reverse-Time SDE

Let $\mathbf{x}_0$ be the clean image corresponds to a $\mathbf{x}_t$ at time step t, say:

$$\mathbf{x}_0 = \mathbf{x}_t + \int_t^0 \mathbf{f}_\beta(\mathbf{x}_u, u)\, \mathrm{d}u + g_\beta(u)\, \mathrm{d}\omega_u\,, \tag{2}$$

where Equation (2) is denoted as $\mathbf{x}_0 \mid_\beta \mathbf{x}_t$, shorthanded as $\mathbf{x}_0 \mid \mathbf{x}_t$. For an arbitrary reward function r e.g. aesthetics score, we define the reward estimate of x_t at time step t as

$$r_\beta(\mathbf{x}_t, t) \coloneqq \mathbb{E}_{\mathbf{x}_0 \mid \mathbf{x}_t}\left[r(\mathbf{x}_0)\right]. \tag{3}$$

This can be estimated with a Monte Carlo estimator by averaging over the reward of several SDE samples, but it requires many sample evaluations for high accuracy. To address this weakness, we introduce an alternative estimator for $r_\beta(\mathbf{x}_t, t)$ based on PF-ODE (Song et al., 2021).

As shown in Song et al. (2021); Karras et al. (2022), the reversed-time SDE reduces to PF-ODE when $\beta \equiv 0$. For each t, a diffeomorphic relationship exists between a noisy sample $\mathbf{x}_t$ and a clean sample $\mathbf{x}_0$ generated by PF-ODE.

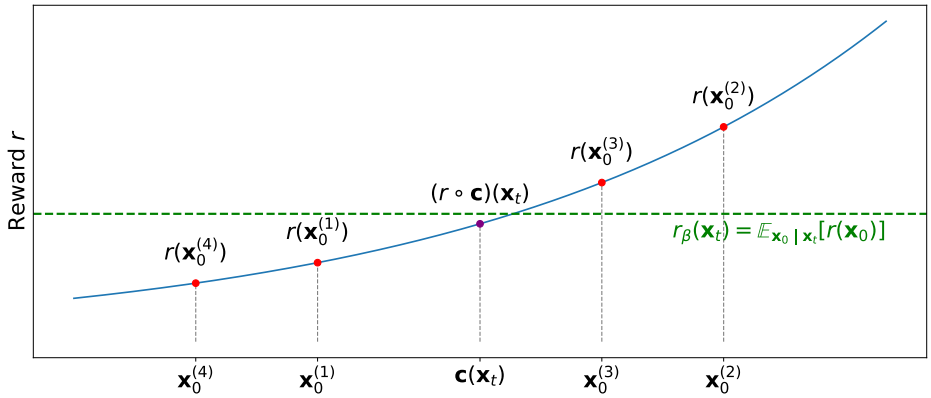


Figure 2: The illustration of the proximity between the r_β and $r \circ \mathbf{c}$. In this figure, the β is nonzero and r is near harmonic (i.e., $\nabla^2 r \approx 0$.). The red points indicate i.i.d. SDE samples and the purple ODE approximation of $\mathbf{x}_t$. The green line indicates the expectation of the rewards of the SDE samples (e.g., an approximate estimation, $\frac{1}{4}\sum_{i=1}^{4} r(\mathbf{x}_0^{(i)})$).

Similar to consistency models, with $\mathbf{x}'_{(t)}$ denoting an ODE trajectory instead of $\mathbf{x}_{(t)}$, we can denote this deterministic mapping from the domain of $\mathbf{x}_t$ to the domain of $\mathbf{x}_0$ as $\mathbf{c}(\mathbf{x}_t, t)$ as

$$\mathbf{c}(\mathbf{x}'_t, t) \coloneqq \mathbf{x}'_0 = \mathbf{x}'_t + \int_t^0 \mathrm{d}\mathbf{x}'_u, \quad \text{where} \quad \mathrm{d}\mathbf{x}'_u = -u\nabla_{\mathbf{x}'_u} \log p\left(\mathbf{x}'_u, u\right)\, \mathrm{d}u. \tag{4}$$

Then, we can write $(r \circ \mathbf{c})(\mathbf{x}_t, t) = r(\mathbf{c}(\mathbf{x}_t, t))$ as the reward of the generated clean sample. This approximates $r_\beta(\mathbf{x}_t, t)$ using only one evaluated sample. In fact, we can characterize the difference between the approximate reward using ODE $(r \circ \mathbf{c})(\mathbf{x}_t, t)$ and the exact reward estimate using SDE $r_\beta(\mathbf{x}_t, t)$ as in Lemma 1. The right hand side of Equation (5) shows that, as $\beta \to 0$, the approximation becomes exact: $\lim_{\beta \to 0^+} r_\beta(\mathbf{x}_t, t) = (r \circ \mathbf{c})(\mathbf{x}_t, t)$. Intuitively, this result aligns with SDEs reducing to ODEs when β approaches zero in image domains (Song et al., 2021).

Lemma 1 (Itô Integral Representation of Reward Proximity Error. Proof is in Appendix D.1)**.** *Let the reward estimate function, $h(\mathbf{x}_t, t) = (r \circ \mathbf{c})(\mathbf{x}_t, t)$, be shorthanded as h. We have:*

$$r_\beta(\mathbf{x}_t, t) - (r \circ \mathbf{c})(\mathbf{x}_t, t) = \mathbb{E}_{\mathbf{x}_0 | \mathbf{x}_t}\left[\int_t^0 \nabla_{\mathbf{x}_u} h \cdot \mathrm{d}\mathbf{J}_\beta(\mathbf{x}_u, u) - \beta u^2 \nabla^2 h \, \mathrm{d}u\right]. \tag{5}$$

where $\mathbf{x}_0$ is sampled from Equation (2) *and*

$$\mathrm{d}\mathbf{J}_\beta(\mathbf{x}_u, u) = -\beta u^2 \nabla_{\mathbf{x}_u} \log p\left(\mathbf{x}_u, u\right)\, \mathrm{d}u + \sqrt{2\beta} u \, \mathrm{d}\omega_u, \tag{6}$$

is the Langevin diffusion SDE term, and $\nabla^2 h$ is the Laplacian of h.

As demonstrated in Appendix D.1, Lemma 1 implies that when the Laplacian of the reward function is approximately zero ($\nabla^2 r \approx 0$), $r_\beta \approx r \circ \mathbf{c}$. We also illustrated the idea in Figure 2. For better presentation, we conveniently abbreviate $r_\beta(\mathbf{x}_t, t)$ as $r_\beta(\mathbf{x}_t)$, $\mathbf{c}(\mathbf{x}_t, t)$ as $\mathbf{c}(\mathbf{x}_t)$ and $(r \circ \mathbf{c})(\mathbf{x}_t, t)$ as $(r \circ \mathbf{c})(\mathbf{x}_t)$ in this paper.

4.2 Demons for Reward-Guided Denoising

Let's first revisit reverse-time SDE. Following Karras et al. (2022), an SDE numerical evaluation of $\hat{\mathbf{x}}_{t-\Delta}$ sampled from $\mathbf{x}_t$ can be seeded by noise $\mathbf{z}$ via a step of Heun's 2nd order method (Ascher & Petzold, 1998) as follows:

Algorithm 1 A Numerical Step with Demon

Input: $\mathbf{x}_t$, t, Δ, K
Output: $\hat{\mathbf{x}}_{t-\Delta}$
for $k = 1$ **to** K **do**
  Draw $\mathbf{z}^{(k)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$
  $\hat{\mathbf{x}}_{t-\Delta}^{(k)} \leftarrow \mathrm{heun}(\hat{\mathbf{x}}_t, \mathbf{z}^{(k)}, t, \Delta)$
  $R_k \leftarrow (r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$ {implementing $r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$}
end for
$[b_k] \leftarrow$ **Demon**$([R_k])$
$\mathbf{z}^* \leftarrow \sqrt{N}\,\mathrm{normalized}\left(\sum_{k=1}^{K} b_k \mathbf{z}^{(k)}\right)$
$\hat{\mathbf{x}}_{t-\Delta} \leftarrow \mathrm{heun}(\hat{\mathbf{x}}_t, \mathbf{z}^*, t, \Delta)$
Return $\hat{\mathbf{x}}_{t-\Delta}$

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n) \tag{7}$$

$$\hat{\mathbf{x}}_{t-\Delta} = \mathrm{heun}(\mathbf{x}_t, \mathbf{z}, t, \Delta) \tag{8}$$

$$:= \mathbf{x}_t - \frac{1}{2}\left[\mathbf{f}_\beta(\mathbf{x}_t, t) + \mathbf{f}_\beta(\tilde{\mathbf{x}}_{t-\Delta}, t-\Delta)\right]\Delta + \frac{1}{2}\left[g_\beta(t) + g_\beta(t-\Delta)\right]\mathbf{z}\sqrt{\Delta}\,, \tag{9}$$

where $\mathbf{z}$ is a Gaussian noise, and heun is the stochastic backward step from $\mathbf{x}_t$ to $\hat{\mathbf{x}}_{t-\Delta}$. The intermediate approximation $\tilde{\mathbf{x}}_{t-\Delta}$ is given by $\tilde{\mathbf{x}}_{t-\Delta} := \mathbf{x}_t - \mathbf{f}_\beta(\mathbf{x}_t, t)\Delta + g_\beta(t)\mathbf{z}\sqrt{\Delta}$. Note that, while we use Heun's method here, other solvers can work too.

For image generation, Gaussian noise $\mathbf{z}$ is usually high-dimensional. For a high-dimensional $\mathbf{z}$, we can assume that it's likely on a $\sqrt{N}$ sphere (Lemma 5, Appendix). This allows us to weighted-combine various noises into a new noise $\mathbf{z}^*$:

$$\mathbf{z}^* = \sqrt{N}\,\mathrm{normalized}\left(\sum_{k=1}^{K} b_k \mathbf{z}^{(k)}\right), \tag{10}$$

where $\mathbf{z}^{(k)}$ are i.i.d. unit Gaussian noises, and b_k are the search space. The pseudocode of a numerical step with our proposed method is outlined in Algorithm 1.

In the following, we introduce two stochastic optimization approaches, the *Tanh Demon* and the *Boltzmann Demon*, to determine the weights b_k and synthesize an optimal $\mathbf{z}^*$, with a goal of optimizing the final reward value. And we show that solving the reverse-time SDE with such optimal $\mathbf{z}^*$ theoretically and empirically improve r_β, essentially achieving alignment.

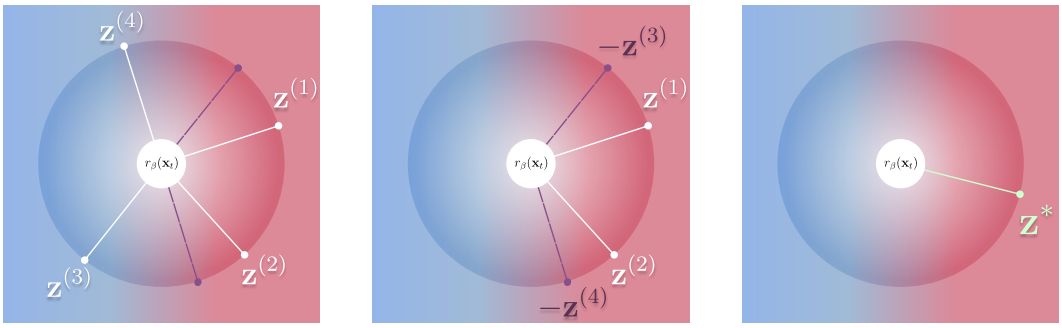
4.2.1 Tanh Demon

Intuitively, we may consider up-weighting the good noises that improve the reward and down-weighting the bad noises that harm the reward, compared to the average reward $\hat{\mu}$. Tanh Demon assigns positive weights to the good noises and negative weights to the bad noises with the $tanh$ function, based on the reward estimates of the noises (Section 4.1):

$$\mathbf{z}^* = \sqrt{N}\,\mathrm{normalized}\left(\sum_{k=1}^{K} b_k^{\tanh}\,\mathbf{z}^{(k)}\right), \quad \text{where} \quad b_k^{\tanh} \leftarrow \tanh\left(\frac{(r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - \hat{\mu}}{\tau}\right) \tag{11}$$

Here we can estimate $\hat{\mu}$ with $\hat{\mu} = \frac{1}{K}\sum_{k=1}^{K}(r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$. τ is the temperature parameter to tanh, which can be adaptively tuned (as shown in Table 8).

In the following, we demonstrate that synthesizing $\mathbf{z}^*$ with Equation (11) in every backward step, which nudges the sample towards the data distribution, leads to reward improvement of the final clean sample $\mathbf{x}_0$ with theoretical guarantee.



(a) Before (b) After (c) Final

Figure 3: An illustration of the Tanh Demon sampling method where $K = 4$. (a) A SDE step generates several samples, each determined by sampled noise $\mathbf{z}_k$. We use Tanh Demon to classify each noise sample as "low-reward" or "high-reward" w.r.t $r_\beta(\boldsymbol{x}_t)$ based on their respective reward estimates. (b) We penalize low-reward noise with $\tanh$, thereby transforming it into high-reward noise. (c) It shows how the post-processed noises are averaged and projected onto the high-dimensional sphere, resulting in a feasible noise representation $\mathbf{z}^*$ with high-reward estimate.

Lemma 2. *Assume the truncation error terms in Equation* (39) *is ignorable. Let* $\mathbf{z}^*$ *be derived from* $b_k^{\tanh} = \tanh\left(\frac{r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - r_\beta(\mathbf{x}_t)}{\tau}\right)$ *using Equation* (10) *for every numerical step from* t *to* $t - \Delta$ *and* $\tau < \infty$. *Then, almost surely,* $r(\hat{\mathbf{x}}_0^{\tanh}) > r_\beta(\mathbf{x}_t)$, *where* $\hat{\mathbf{x}}_0^{\tanh}$ *is derived by applying* $\mathbf{z}^*$ *on every step.*

The formal proof of Lemma 2 is in Appendix D.3. Lemma 2 states that the improvement of r_β in reversed-time order almost surely, assuming that $r_\beta \equiv r \circ \mathbf{c}$ and that, based on Fact 1 (Appendix), $\frac{1}{K}\sum_{k=1}^{K} r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$ is an unbiased estimator of $r_\beta(\mathbf{x}_t)$. Figure 3 provides an illustration of the Tanh Demon.

Since $(r \circ \mathbf{c})$ is not a perfect estimate of r_β, selecting an appropriate τ is crucial to manage errors. A small τ makes $\tanh(\cdot/\tau)$ nearly binary and can weight good noise negatively and vice versa. Conversely, a large τ diminishes contribution of noises.

4.2.2 Boltzmann Demon

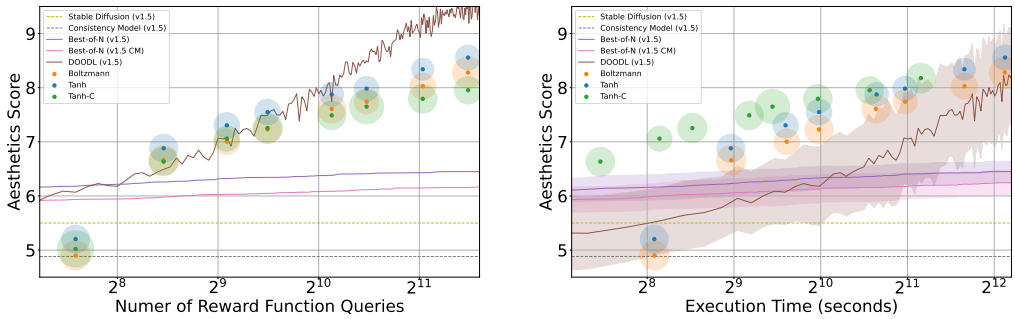
Another intuitive approach is to estimate the candidate with maximum reward. We propose the Boltzmann demon, which assign noise weights as follows.

$$b_k^{\text{boltz}} \leftarrow \frac{\exp\left(r \circ \mathbf{c}(\hat{\mathbf{x}}_{t-\Delta}^{(k)})/\tau\right)}{\sum_{k=1}^{K} \exp\left(r \circ \mathbf{c}(\hat{\mathbf{x}}_{t-\Delta}^{(k)})/\tau\right)}. \tag{12}$$

The theoretical guarantee of improvement in r_β in expectation is provided in Lemma 3 of Appendix, assuming $r_\beta \equiv r \circ \mathbf{c}$. This method is equivalent to the single-step cross entropy approach (De Boer et al., 2005). Although, empirically we find that Tanh Demon outperforms Boltzmann demon, adjusting τ in Boltzmann demon provides control over deviation from the original SDE distribution, as demonstrated in Lemma 4 (Appendix).

4.2.3 Computational Considerations

Let's first consider a Demon sampling trajectory $\mathbf{x}_{t_1} > \mathbf{x}_{t_2} > \cdots > \mathbf{x}_{t_T} \approx 0$ for a fixed number T. Each Demon's trajectory requires $\mathcal{O}(K \cdot T)$ evaluations of $\mathbf{c}$, and each evaluation comes with one reward estimation. The compute time is mainly influenced by the implementation of $r \circ \mathbf{c}$. We discuss two aspects of $r \circ \mathbf{c}$—the temporal cost and the fidelity—which are vital to the algorithm's time complexity and reward performance, respectively.



(a) Performance w.r.t Reward Query Number

(b) Performance w.r.t Execution Time

Figure 4: Performance comparison of the proposed algorithm and other baseline methods in terms of the number of reward queries and execution time; the dependent variable is T, which is suggested to be larger for SDE solver to reduce truncation error. Although DOODL can achieve similar results to ours, it relies on reward backpropagation, whereas our backpropagation-free methods do not require this. The shaded areas and the radii of solid circles represent the standard deviation of the evaluation results.

Note that Tanh or Boltzmann Demon itself does not strictly specify the implementation of $r \circ \mathbf{c}$; our default option uses Heun's ODE solver, but using a Consistency Model (CM) distilled from the original diffusion model significantly accelerates computation. An alternative, which we refer as Tanh-C, is to combine our Tanh Demon algorithm with an off-the-shelf CM to implement $r \circ \mathbf{c}$. While using Tanh-C may slightly degrade the results due to the fidelity loss from using a CM (see Table 2), this approach is particularly effective when faster results are required since the computation of $\mathbf{c}$ is much quicker. For a larger T, however, the default Tanh Demon using Heun's method outperforms Tanh-C in terms of reward performance.

As shown in Table 10, using the text-to-image generation task settings from Black et al. (2023), the Demon algorithm achieves an aesthetics score of 6.72 ± 0.26 on SD v1.4, requiring 5 minutes (i.e., $K = 16, T = 16$) on an NVIDIA RTX 3090 GPU. Within the same 5-minute computation window, the Tanh-C variant achieves an improved score of 7.27 ± 0.33 (i.e., $K = 16, T = 64$). Notably, the upper bound for randomly sampled SD v1.4 is approximately 6.5, obtained after more than 10 minutes and 800 reward function queries. See Appendix B for parameter guidelines and settings.

5 Experiments

In this section, we present both quantitative and qualitative evaluations of our methods. Due to the page limit, we include the details of the implementation and experimental settings in Appendix I and the subjective results in Appendix G.2.

Baseline Comparison. For the performance comparisons between our method and other baselines, we use the LAION (2023) aesthetics scores (Aes) as the evaluation metric, and the scores are evaluated on a set of various prompts for generating animal images, which were selected from the full set of 45 common animals in ImageNet-1K (Deng et al., 2009), created by Black et al. (2023). We use 20-step Heun's ODE for reward estimate for our methods and Best-of-N (SD v1.5).

In terms of reward queries, Tanh/Tanh-C outperforms other baseline methods in most cases, including our Boltzmann method and Best-of-N. Our methods are even comparable to the backpropagation-based DOODL (Wallace et al., 2023b), the state-of-the-art method optimized over the reward function. In terms of execution time, Tanh/Tanh-C consistently outperforms DOODL due to the exclusion of backpropagation; Tanh-C further benefits from its effective computational cost, given limited time. Moreover, our method's backpropagation-free nature makes it more resistant to reward hacking (Table 3). For further comparison on PickScore (Kirstain et al., 2023), please refer to Appendix E.1.

Comparison of Reward Estimation Approaches. Figure 4 demonstrates a comparison of the proposed methods with different $r \circ \mathbf{c}$ implementations, including 20-step Heun's ODE (Tanh) and

Table 2: Comparison of accuracy and time cost across different $r \circ \mathbf{c}$ implementations.

Implementation	Time (s)	Standard Deviation of $r_\beta - r \circ \mathbf{c}$				
		$t = 0.5$	$t = 1.0$	$t = 3.0$	$t = 7.0$	$t = 14.0$
20-step ODE	1.94	$\mathbf{9.3 \times 10^{-2}}$	$\mathbf{1.50 \times 10^{-1}}$	$\mathbf{3.04 \times 10^{-1}}$	$\mathbf{3.68 \times 10^{-1}}$	$\mathbf{3.80 \times 10^{-1}}$
6-step ODE	0.62	9.8×10^{-2}	1.58×10^{-1}	3.27×10^{-1}	3.72×10^{-1}	3.82×10^{-1}
1-step Consistency	**0.18**	1.73×10^{-1}	3.01×10^{-1}	4.10×10^{-1}	4.38×10^{-1}	5.04×10^{-1}

1-step CM (Tanh-C). **Tanh**, which uses a 20-step ODE for accurate $(r \circ \mathbf{c})$, consistently outperforms the Best-of-N baseline given an equivalent number of reward queries. **Tanh-C**, which employs a 1-step CM for fast reward evaluation, outperforms Tanh when considering limited execution time. These observations suggest that the quality of $r \circ \mathbf{c}$ indeed plays a significant role in the effectiveness of our method.

To further validate the importance of $r \circ \mathbf{c}$, we conduct a comparative analysis based on Lemma 1 ($r \circ \mathbf{c} \approx r_\beta$). In this analysis, we evaluate the accuracy and computational cost across three methods: 20-step Heun's ODE, 6-step Heun's ODE, and 1-step CM; both diffusion and consistency models are based on the SD v1.5 and distilled by Luo et al. (2023). Experiments were performed with $t = 0.5, 3.0, 1.0, 7.0, 14.0$ ranging from 0.002 to 14.648. Accuracy was quantified using the standard deviation of, $r_\beta(\mathbf{x}_t) - (r \circ \mathbf{c})(\mathbf{x}_t)$. Here, $\mathbf{x}_t$ is sampled from $\mathcal{N}(\mathbf{0}, t_{\max}^2 \mathbf{I}_n)$ and integrated from $t_{\max}$ to t using a 100-step diffusion model ODE, performed on 22 subset of Black et al. (2023)[1].

The results, presented in Table 2, support that the quality of $r \circ \mathbf{c}$ influences both the algorithm's speed and reward performance. For the ODE methods, the trend follows our expectation: As t approaches 0, the standard deviation decreases, which can be attributed to the diminishing noise as the posterior $p(\mathbf{x}_t \mid \mathbf{x}_0)$ becomes more sharply peaked; the number of ODE steps is crucial to the quality of the generated outputs; more steps generally lead to higher fidelity results, although this comes at the cost of increased computational time; using 1-step CM leads to inferior results compared to using ODE, supposedly as the distillation gap and the limited model capacity result in lower-fidelity reconstructions.

Table 3: Results using various reward functions and different generation methods. Each column represents a specific reward objective, with the best performance highlighted in bold.

Generation method	Aes ↑	IR ↑	Pick ↑	HPSv2 ↑	Time
SD v1.4	5.34 ± 0.56	-0.00 ± 0.95	0.202 ± 0.008	0.216 ± 0.036	5 s
DPO	5.36 ± 0.72	0.03 ± 0.84	0.203 ± 0.007	0.229 ± 0.027	5 s
Uni (CLIP-guided)	4.11 ± 0.74	-1.81 ± 0.50	0.191 ± 0.014	0.173 ± 0.022	55 min
Tanh + Aes	$\mathbf{7.35} \pm 0.40$	-0.03 ± 1.24	0.211 ± 0.010	0.257 ± 0.041	
Tanh + IR	5.96 ± 0.28	$\mathbf{1.95} \pm 0.07$	0.216 ± 0.012	0.286 ± 0.033	
Tanh + Pick	6.14 ± 0.48	1.39 ± 0.57	$\mathbf{0.245} \pm 0.010$	0.312 ± 0.033	18 min
Tanh + HPSv2	5.98 ± 0.45	1.51 ± 0.63	0.228 ± 0.011	$\mathbf{0.367} \pm 0.027$	
Tanh + Ensemble	6.53 ± 0.50	1.81 ± 0.15	0.236 ± 0.014	0.356 ± 0.030	
Best-of-N	$\mathbf{6.32} \pm 0.34$	$\mathbf{1.69} \pm 0.18$	$\mathbf{0.218} \pm 0.009$	$\mathbf{0.291} \pm 0.015$	18 min
DOODL + Aes	5.59 ± 0.29	-0.68 ± 1.06	0.197 ± 0.008	0.221 ± 0.028	18 min
DOODL + Pick	5.21 ± 0.46	-0.12 ± 0.84	0.204 ± 0.010	0.220 ± 0.035	1.1 hr

Image Generation with Various Reward Functions. While our method optimizes a given reward function, as shown in Figure 4, we also present qualitative results in Table 4 and cross-validation results in Table 3. These results demonstrate perceptual preferences by averaging rewards derived from prompts provided in Tables 12 and 15.

We employ our Tanh Demon with various reward functions, such as Aes (LAION, 2023), ImageReward (IR)(Xu et al., 2023), PickScore (Pick)(Kirstain et al., 2023), HPSv2 (Wu et al., 2023), and a scaled sum (Ensemble) of Aes, IR, Pick, and HPSv2. For comparison, we include the best reward during sampling under the same time condition (best-of-N) and DOODL (Wallace et al., 2023b), which is optimized on Aes and Pick to modify results generated by PF-ODE using their recommended settings. For reference, we also provide the performance of a training-based method, DPO Wallace

[1]Obtained from the official repository github.com/jannerm/ddpo/blob/main/assets/very_simple_animals.txt

Table 4: Using Tanh Demons with various reward functions. The baseline, Stable Diffusion v1.4, refers to the standard model without our proposed enhancements.

Baseline	Best-of-N	Uni	DOODL	Aes	Ensemble	DPO
A demon exiting through a portal						
A painting of a girl encountering a giant sunflower blocking her path in a hallway						

et al. (2023a), and a backpropagation-based method, Universal Guidance (Uni) by Bansal et al. (2024), guided by the CLIP condition.

Starting from the baseline SDv1.4 under similar computational conditions, Tanh exhibits improvement **across the four metrics**, demonstrating robustness even when acknowledging slight over-optimization of the objective—in contrast to DOODL. By comparing best-of-N and Tanh with the Ensemble reward, our method achieves superior performance on *each* objective using the Ensemble, demonstrating not only the ability to integrate a mixture of rewards but also generating a **superior samples** that outperforms all individual best samples selected by best-of-N methods.

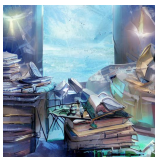
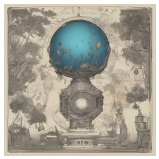
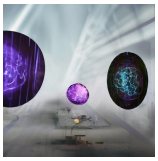
Alignment with preferences of VLMs (Non-differentiable). In Table 5, we present qualitative results of aligning diffusion models SDXL to preferences of VLMs from API, as a demonstration of using non-differentiable reward sources. In this experiment, we use Google Gemini Pro v1.0 (Gemini Team Google, 2024) and GPT4 Turbo (OpenAI, 2024). In each step, the VLM receives a fixed prompt, e.g. "You are a journalist who wants to add a visual teaser for your article to grab attention on social media or your news website", and is asked to select the best-matching intermediate sample from generated images. VLMs are presented with $\mathbf{c}(\mathbf{x}_t)$ and $\mathbf{c}(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$ produced by PF-ODE. The reward b_k^{VLM} is 0.5 if the VLM selects $\mathbf{c}(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$ and -0.5 otherwise. We also use PickScore (Kirstain et al., 2023) to evaluate the results and find that 14 out of 16 images generated with VLMs show improvements compared to directly generating with PF-ODE.

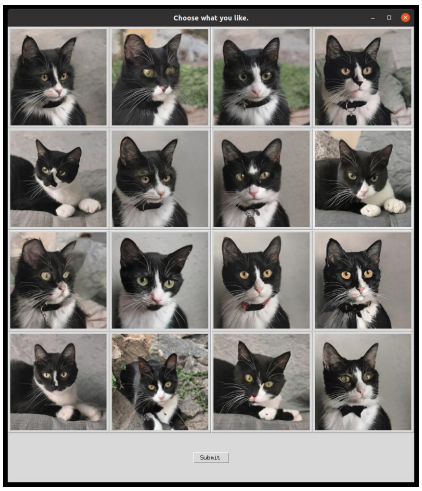
Manual Selection. We also explore using online interactive human judgements to guide diffusion. That means, the users themselves would be (non-differentiable) reward functions. We let users directly interact with our method to generate desired images. Figure 5a shows an example interface created by us for an image resembling a given reference cat image. At each iterative step from t to $t - \Delta$, we sample 16 i.i.d. copies of $\boldsymbol{x}_{t-\Delta}$ and compute $\mathbf{c}(\boldsymbol{x}_{t-\Delta})$ with PF-ODE. The user then manually select their preferred image, assigning a reward of $+1$ to it and -1 to the others. We continue this process until there is no obvious preferred ones among the generated images. As shown in Figure 5b, the image generated by our method more closely matches the target than the one produced by PF-ODE. We also measure the improvement with DINOv2 (Oquab et al., 2023) embedding cosine similarity between the reference image and the generated image, and observe that the similarity improves from 0.594 to 0.708 through online user interactions.

6 Conclusion

This work addresses the challenge of better aligning pre-trained diffusion models without training or backpropagation. We first demonstrate how to estimate noisy samples' rewards based on clean samples using PF-ODE. Additionally, we introduce a novel inference-time sampling method, based on stochastic optimization, to guide the denoising process with any reward sources, including non-differentiable reward sources that includes VLMs and interactive human judgements. Theoretical analysis and extensive experimental results validate the effectiveness of our proposed method for

Table 5: Using VLMs to generate images. PF-ODE (baseline) refers to a baseline without using our method for alignment. Columns 3-6 indicate the role that the agent plays in the given prompt.

Model	Baseline	Teacher	Artist	Researcher	Journalist
Gemini-SD v1.4					
Gemini-SDXL					
GPT-SD v1.4					
GPT-SDXL					



(a) Our user interface for interacting with our algorithm (0.594 cosine similarity).

(b) (Top Left) Image generated by PF-ODE (0.622 cosine similarity). (Bottom Left) Image generated by our method (0.708 cosine similarity). (Right) Reference image.

Figure 5: We design an application for manual interaction with our algorithm. Our author selects the images, and the criteria are based on the author's preference (non-preferred images are kept unselected), where the author tries to align the reference image. We evaluate performance by measuring the cosine similarity of DINOv2 features between the targeted and reference images.

improved image generation without requiring additional training. Through comprehensive empirical and theoretical analysis, we observe that the quality and efficiency of reward estimation $r \circ \mathbf{c}$ are essential for our algorithm, especially in balancing computational speed and reward performance.

REFERENCES

Uri M Ascher and Linda R Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM, 1998.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, 2024.

Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2017.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Machine Learning*, 2023.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, 2024.

Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition*, 2009.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

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

Gemini Team Google. Gemini: A family of highly capable multimodal models, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Charles AR Hoare. Quicksort. *The computer journal*, 1962.

Kiyosi Ito, Kiyosi Itô, Kiyosi Itô, Japon Mathématicien, Kiyosi Itô, and Japan Mathematician. *On stochastic differential equations*. American Mathematical Society New York, 1951.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

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

Andrei Kolmogoroff. Über die analytischen methoden in der wahrscheinlichkeitsrechnung. *Springer*, 1931.

LAION. Laion aesthetic score predictor. `https://laion.ai/blog/laion-aesthetics/`, 2023.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference, 2023.

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

OpenAI. Gpt-4 technical report, 2024.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2024. URL `https://openreview.net/forum?id=Vaf4sIrRUC`.

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

Alvin C Rencher. A review of "methods of multivariate analysis, ", 2005.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Computer Vision and Pattern Recognition*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Roman Vershynin. High-dimensional probability. *University of California, Irvine*, 2020.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization, 2023a.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *International Conference on Computer Vision*, 2023b.

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

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

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Qimai Li, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model, 2024.

A NOTATIONS AND CONVENTIONS

Although we keep the main paper self-consistent, we provide this section to establish a consistent notation and convention for this paper as an aid.

A.1 NOTATIONS

Table 6: Notations

Notation	Description
N	State dimension
K	Noise sample number
$t_{\min}, t_{\max}$	Upper bound and Lower bound of the noise level in numerical integration
T	Number of time steps to solve SDE/ODE
β	Noise parameter
$\mathbf{x}$	State variable
$\mathbf{z}$	Noise from Gaussian
Δ	Time step
b_k	Unnormalized weight of noise
$\mathbf{f}_\beta$	SDE policy drift
g_β	SDE policy diffusion coefficient
$\mathbf{f}_0$	PF-ODE policy drift
$\mathbf{J}_\beta$	Langevin diffusion SDE
ω_t	reversed time Brownian motion
r	Reward
r_β	Reward estimates of SDE policy
$\mathbf{c}$	Function to get expected ODE result
heun	Heuns's method, SDE solver for Karras SDE

A.2 CONVENTIONS

Table 7: Conventions

Convention	Details
$r \circ \mathbf{c}$	ODE reward estimate approximation, $r(\mathbf{c}(\mathbf{x}_t, t)) = (r \circ \mathbf{c})(\mathbf{x}_t, t)$
$f \equiv g$	For all x of our interest, $f(x) = g(x)$
$\hat{\mathbf{x}}$	Numerical approximation with SDE solver
$\tilde{\mathbf{x}}$	Intermediate value of Heun's method
$\mathbf{x}'$	An ODE trajectory
$\tilde{\mathbf{z}}$	Uniformly sampled from the sphere of radius $\sqrt{N}$
$\mathbf{z}^*$	Optimal noise generated by our algorithm
$\hat{\mu}$	Mean of next state ODE reward estimates, $\frac{1}{K}\sum_{k=1}^{K}(r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)})$
$r(\mathbf{x}_t)$	Shorthand for $r(\mathbf{x}_t, t)$ when the context is clear
$\mathbf{c}(\mathbf{x}_t)$	Shorthand for $\mathbf{c}(\mathbf{x}_t, t)$ when the context is clear
$(r \circ \mathbf{c})(\mathbf{x}_t)$	Shorthand for $(r \circ \mathbf{c})(\mathbf{x}_t, t)$ when the context is clear
$\mathbf{x}_0 \mid \mathbf{x}_t$	Shorthand for $\mathbf{x}_0 \mid_\beta \mathbf{x}_t$, where $\mathbf{x}_0 = \mathbf{x}_t + \int_t^0 \mathbf{f}_\beta(\mathbf{x}_u, u)\,\mathrm{d}u + g_\beta(u)\,\mathrm{d}\omega_u$
$\tilde{\omega}_t$	standard Brownian motion

Instead of just ODE, we use PF-ODE to highlight Song et al. (2021)'s contribution or when the context is unclear. They are equivalent here.

B GUIDELINE ON PARAMETER SETTING

We explore the optimal setting for parameter τ with respect to the Boltzmann Demon and the Tanh Demon. For the Tanh Demon, the most effective τ is neither ∞ nor 0. We recommend setting τ to the standard deviation of the estimations $\{(r \circ \mathbf{c})(\boldsymbol{x}_{t-\Delta}^{(k)})\}_{k=1}^{K}$, rendering it an adaptive parameter that is robust to scaling. For the Boltzmann Demon, optimal performance is achieved by setting τ to 0, as demonstrated in Table 8.

Table 8: Comparison of performance for different settings of τ in the setting of Figure 4.

	$\tau = 1$	$\tau = 0.01$	Adaptive τ
Tanh	7.40 ± 0.30	7.24 ± 0.31	$\mathbf{7.45} \pm 0.33$
Boltzmann	6.30 ± 0.35	$\mathbf{7.28} \pm 0.30$	6.85 ± 0.37

We also conduct an ablation study on the remaining parameters K and β. The base configuration is $K = 16, \beta = 0.1$, with an adaptive temperature τ for the Tanh Demon. We set $T = 32$ for the ablation study of β and $T = 64$ for K.

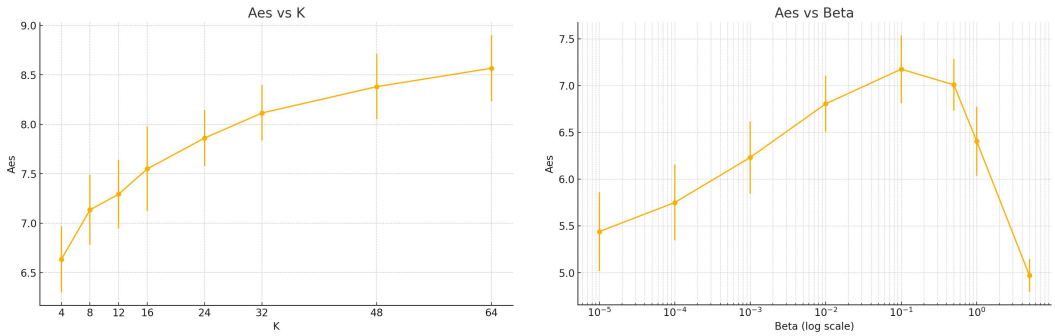


Figure 6: Comparison of our algorithm with respect to K and β

We found a large β makes the sampling unstable, given the number of steps T is fixed. Predictably, sampling with a β close to 0 is reduced to ODE. From our theoretical result Lemma 1, the design methodology, and empirical results, the guidelines Table 9 can assist users in setting parameters. We provide a sparse parameter search in Table 10.

Parameter	**Description**
K	Controls the noise distribution bias, positively affecting final quality and linearly increasing computational time.
β	Adjusts the distribution's proximity to the original PF-ODE. Set empirically based on r's characteristics. Lemma 1 suggests smaller β for reward functions with Laplacian deviations.
T	Inherit the properties of time steps T from diffusion models, scaling computational time linearly. Karras's EDM recommends $T > 17$.
τ	Recommended values vary for Boltzmann and Tanh Demons, as detailed in Table 8.
$r \circ \mathbf{c}$	Accurate reward estimates are critical for ensuring high final quality.

Table 9: Guidelines for Setting Hyperparameters

Table 10: This table presents the experimental configurations used to measure aesthetics score under various animal prompts, presenting a sparse search of parameters. The time column represents the duration required to generate each image. We alias adaptive temperature as Adaptive.

Demon	**Checkpoint**	β	K	T	τ	**Aes**	**Time** (min)
Boltzmann	SD v1.4	0.1	16	64	Adaptive	6.408 ± 0.36	17.6
					1e-10	7.111 ± 0.32	16.6
	SDXL	0.05	16	32	Adaptive	6.853 ± 0.37	45.8
					1e-02	7.276 ± 0.30	45.4
					1	6.300 ± 0.35	46.1
		0.1	16	64	Adaptive	6.990 ± 0.38	94.2
					1e-10	7.501 ± 0.31	93.1
Tanh	SD v1.4	0.05	16	16	Adaptive	6.723 ± 0.26	5.0
				32	Adaptive	7.073 ± 0.22	9.7
				64	Adaptive	7.394 ± 0.29	18.7
		0.1	16	64	Adaptive	7.549 ± 0.43	18.7
			64	64	Adaptive	8.566 ± 0.33	79.1
	Diffusion-DPO	0.1	16	64	Adaptive	7.564 ± 0.34	94.5
	SDXL	0.01	16	16	Adaptive	6.876 ± 0.40	22.0
		0.05	16	16	Adaptive	6.866 ± 0.35	21.9
				32	Adaptive	7.459 ± 0.33	46.0
					1e-02	7.244 ± 0.31	46.0
					1	7.398 ± 0.30	46.2
		0.1	8	64	Adaptive	7.446 ± 0.37	47.0
			16	64	Adaptive	7.841 ± 0.32	94.4
			32	64	Adaptive	8.179 ± 0.35	188.8
		0.5	16	32	Adaptive	6.370 ± 0.35	46.0
Tanh-C	SD v1.4	0.5	16	64	Adaptive	7.269 ± 0.33	5.0
		0.1	16	64	Adaptive	6.710 ± 0.34	5.0
	SDXL	0.5	16	64	Adaptive	7.301 ± 0.24	17.9

C PSEUDOCODES

As an aid, we provide pseudocodes for the design of Demons Algorithm 2, Algorithm 3:

Algorithm 2 Tanh Demon with Adaptive Temperature

Input: A list of ODE reward estimate $[R_k]$
Output: Noise Weights $[b_k]$
$K \leftarrow$ **length**$([R_k])$
$\hat{\mu} \leftarrow \frac{1}{K}\sum_{k=1}^{K} R_k$
$\tau \leftarrow \sqrt{\frac{1}{K}\sum_{k=1}^{K}(R_k - \hat{\mu})^2}$
for $k = 1$ **to** K **do**
    $b_k \leftarrow \tanh\left(\frac{R_k - \hat{\mu}}{\tau}\right)$
end for
Return $[b_k]$

Algorithm 3 Boltzmann Demon with Fixed Temperature τ

Input: A list of ODE reward estimate $[R_k]$
Output: Noise Weights $[b_k]$
$K \leftarrow$ **length**$([R_k])$
$Z \leftarrow \frac{1}{K}\sum_{k=1}^{K} \exp\left(\frac{R_k}{\tau}\right)$
for $k = 1$ **to** K **do**
    $b_k \leftarrow \frac{1}{Z}\exp\left(\frac{R_k}{\tau}\right)$
end for
Return $[b_k]$

D MATHEMATICS

D.1 ERROR COMPREHENSION FOR REWARD ESTIMATE APPROXIMATION

In this section, we present the theoretical analysis and proof better to understand the error in our reward estimate approximation.

D.1.1 ERROR TERM AS AN ITÔ INTEGRAL

Lemma 1. *Let the reward estimate function, $h(\mathbf{x}_t, t) = (r \circ \mathbf{c})(\mathbf{x}_t, t)$, be shorthanded as h. We have:*

$$r_\beta(\mathbf{x}_t, t) - (r \circ \mathbf{c})(\mathbf{x}_t, t) = \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}\left[\int_t^0 \nabla_{\mathbf{x}_u} h \cdot \, \mathrm{d}\mathbf{J}_\beta(\mathbf{x}_u, u) - \beta u^2 \nabla^2 h \, \mathrm{d}u\right]. \tag{13}$$

where $\mathbf{x}_0$ is sampled from Equation (2) *and*

$$\mathrm{d}\mathbf{J}_\beta(\mathbf{x}_u, u) = -\beta u^2 \nabla_{\mathbf{x}_u} \log p\left(\mathbf{x}_u, u\right) \, \mathrm{d}u + \sqrt{2\beta} u \, \mathrm{d}\omega_u, \tag{14}$$

is the Langevin diffusion SDE term, and $\nabla^2 h$ is the Laplacian of h.

Proof. We aim to prove:

$$r(\mathbf{x}_0) - (r \circ \mathbf{c})(\mathbf{x}_t, t) = \int_t^0 \nabla_{\mathbf{x}_u} h \cdot \, \mathrm{d}\mathbf{J}_\beta(\mathbf{x}_u, u) - \beta u^2 \nabla^2 h \, \mathrm{d}u, \tag{15}$$

Recall that

$$\mathbf{x}_0 = \mathbf{x}_t + \int_t^0 \mathbf{f}_\beta(\mathbf{x}_u, u) \, \mathrm{d}u + g_\beta(u) \, \mathrm{d}\omega_u, \tag{16}$$

$$\mathbf{c}\left(\mathbf{x}'_t, t\right) = \mathbf{x}'_t + \int_t^0 \mathbf{f}_0(\mathbf{x}'_u, u) \, \mathrm{d}u. \tag{17}$$

For an ODE trajectory $\mathbf{x}'(t)$, notice that:

$$0 = \frac{\mathrm{d}}{\mathrm{d}t} h(\mathbf{x}'(t), t) = \frac{\partial h}{\partial t} + \nabla_{\mathbf{x}} h \cdot \frac{\mathrm{d}\mathbf{x}'}{\mathrm{d}t} = \frac{\partial h}{\partial t} + \nabla_{\mathbf{x}} h \cdot \mathbf{f}_0. \tag{18}$$

We can write:

$$r(\mathbf{x}_0) - (r \circ \mathbf{c})(\mathbf{x}_t, t) = h(\mathbf{x}_0, 0) - h(\mathbf{x}_t, t) = \int_t^0 \mathrm{d}h, \tag{19}$$

where $\mathbf{x}_t$, which is not an ODE trajectory (noted by $\mathbf{x}'_t$), follows the SDE trajectory. Using Itô's lemma Ito et al. (1951), we find:

$$\mathrm{d}h = \left(\frac{\partial h}{\partial t} + \nabla_{\mathbf{x}} h \cdot \mathbf{f}_\beta - \frac{1}{2} \cdot g_\beta^2 \nabla^2 h\right) \mathrm{d}t + g_\beta \nabla_{\mathbf{x}} h \cdot \, \mathrm{d}\omega_t \tag{20}$$

$$= \left(\frac{\partial h}{\partial t} + \nabla_{\mathbf{x}} h \cdot \mathbf{f}_\beta - \left(\frac{\partial h}{\partial t} + \nabla_{\mathbf{x}} h \cdot \mathbf{f}_0\right) - \frac{1}{2} g_\beta^2 \nabla^2 h\right) \mathrm{d}t + g_\beta \nabla_{\mathbf{x}} h \cdot \, \mathrm{d}\omega_t \tag{21}$$

$$= \left(\nabla_{\mathbf{x}} h \cdot (\mathbf{f}_\beta - \mathbf{f}_0) - \frac{1}{2} g_\beta^2 \nabla^2 h\right) \mathrm{d}t + g_\beta \nabla_{\mathbf{x}} h \cdot \, \mathrm{d}\omega_t \tag{22}$$

$$= \nabla_{\mathbf{x}} h \cdot \left(-\beta t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t, t) \, \mathrm{d}t + \sqrt{2\beta} t \, \mathrm{d}\omega_t\right) - \beta t^2 \nabla^2 h \, \mathrm{d}t. \tag{23}$$

The sign of the Itô correction term is flipped due to reverse time Brownian Motion-—and the other is followed by expansion. We thus derived Equation (15). □

D.1.2 Discussion

We interpret the error terms of the reward estimates approximation as follows:

- The estimate becomes more accurate as β decreases, satisfying the intuition that SDE trajectories will reduce to the ODE trajectory as $\beta \to 0$.
- If $\nabla_{\mathbf{x}_u} h \perp \nabla_{\mathbf{x}_u} \log p\left(\mathbf{x}_u, u\right)$, the term $\nabla_{\mathbf{x}_u} h \cdot \mathrm{d}\mathbf{J}_\beta(\mathbf{x}_u, u)$ cancels out in expectation.
- If $\nabla^2 h \equiv 0$ and the previous condition holds, then $r \circ \mathbf{c} \equiv r_\beta$.

For estimation purposes, we make the following assumptions to facilitate understanding and derivation of Equation (5):

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t, t) \approx -\frac{\mathbf{x}_t}{t^2} \tag{24}$$

$$\mathbf{c}(\mathbf{x}_t, t) \approx C_t \mathbf{x}_t \tag{25}$$

$$\nabla_{\mathbf{x}} r \perp \mathbf{x} \tag{26}$$

where C_t is a time-dependent constant and r is scale-invariant.

- Equation (24) is derived from the assumption that $p(\mathbf{x}_t) \approx \mathcal{N}(\mathbf{0}, t^2\mathbf{I})$.
- Equation (25) stems from image preprocessing algorithms, such as those used in Stable Diffusion, which normalize the image distribution. This normalization implies that images in the dataset are often scaled to lie on a sphere. Therefore, we can reasonably assume that a randomly generated $\mathbf{x}_t$ is close to an image in the dataset in direction.
- Equation (26) is based on the intuition that minor changes in brightness do not significantly affect the semantic interpretation of an image. Besides, many training algorithms incorporate scaling as part of data augmentation, which aligns with the assumption that the gradient of $\nabla_{\mathbf{x}} r$ is orthogonal to $\mathbf{x}$.

Under these assumptions, we obtain:

$$\mathrm{d}h = \nabla_{\mathbf{x}} h \cdot \left(-\beta t^2 \nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t, t\right)\, \mathrm{d}t + \sqrt{2\beta} t\, \mathrm{d}\omega_t\right) - \beta t^2 \nabla^2 h\, \mathrm{d}t \tag{27}$$

$$\approx C_t \nabla_{\mathbf{x}} r \cdot \left(-\beta \mathbf{x}_t\, \mathrm{d}t + \sqrt{2\beta} t\, \mathrm{d}\omega_t\right) - \beta t^2 \nabla^2 h\, \mathrm{d}t \tag{28}$$

$$\approx \sqrt{2\beta} t C_t \nabla_{\mathbf{x}} r \cdot\, \mathrm{d}\omega_t - \beta t^2 C_t^2 \nabla^2 r\, \mathrm{d}t \tag{29}$$

If r is harmonic, i.e., $\nabla^2 r \equiv 0$, then $\mathrm{d}h$ becomes a martingale (Billingsley, 2017) and:

$$r_\beta(\mathbf{x}_t, t) - (r \circ \mathbf{c})(\mathbf{x}_t, t) \approx \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}\left[\int_t^0 \sqrt{2\beta} t C_t \nabla_{\mathbf{x}} r \cdot\, \mathrm{d}\omega_t\right] = 0. \tag{30}$$

The mean value property, an equivalent statement of a harmonic function, states that the value of a harmonic function at any point is the average of its values on any sphere centered at that point. This property provides an intuitive explanation of our method: if r is harmonic, the reward of the ODE-generated image is the mean value of the reward of SDE-generated ones, while empirically, we observe that the ODE generation resembles the SDE variants.

D.1.3 Illustration of Mismatch

For better understanding, we provide an example that r_β and $r \circ \mathbf{c}$ don't meet. We adopt assumptions in Appendix D.1.2 to illustrate the intuition, and suppose $\mathbf{x}_t$ is a noisy sample at time t such that $\mathbf{c}(\mathbf{x}_t)$ is a sharp local maxima of r, where $\nabla^2 r \ll 0$ near $\mathbf{c}(\mathbf{x}_t)$. Suppose further that β is small enough such that the generated $\mathbf{x}_0$ is near $\mathbf{c}(\mathbf{x}_t)$. In this case, $r_\beta(\mathbf{x}_t) - (r \circ \mathbf{c})(\mathbf{x}_t) < 0$ as $r_\beta(\mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}\left[r(\mathbf{x}_0)\right] < (r \circ \mathbf{c})(\mathbf{x}_t)$ by intuition.

We can also verify $r_\beta(\mathbf{x}_t) - (r \circ \mathbf{c})(\mathbf{x}_t) < 0$ using Equation (15). Under the assumptions in Appendix D.1.2, we can write:

$$r_\beta(\mathbf{x}_t) - (r \circ \mathbf{c})(\mathbf{x}_t) \approx \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}\left[\int_t^0 \sqrt{2\beta} t C_t \nabla_{\mathbf{x}} r \cdot \mathrm{d}\omega_t - \beta u^2 \nabla^2 h \,\mathrm{d}u\right] \tag{31}$$

$$= \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}\left[\int_t^0 -\beta u^2 C_t^2 \nabla^2 r \,\mathrm{d}u\right] \tag{32}$$

$$< 0. \tag{33}$$

Note that the value of $\nabla^2 r$ is taken at $\mathbf{c}(\mathbf{x}_t)$, fluctuating with SDE.

D.2 Martingale Property of Reward Estimates.

A martingale is a sequence of random variables that maintains a certain property over time Billingsley (2017): the expected future value, given all past values, is equal to the current value; for a fixed SDE, the current reward estimate is the expected value of the reward estimates at the next time step:

Fact 1. *For any time step $\Delta < 0$ such that $t > t - \Delta > 0$:*

$$r_\beta(\mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_{t-\Delta}|\mathbf{x}_t}\left[r_\beta(\mathbf{x}_{t-\Delta})\right]. \tag{34}$$

Intuitively speaking, this idea stems from the principles of conditional probability, which tell us that our current prediction of the final score should be the same as the average of all possible future predictions.

Proof. This result follows directly from the foundational theorem of expectation. Let $\mathcal{G}$ be the σ-algebra generated by $\mathbf{x}_{t-\Delta}$ and $\mathcal{F}$ be the σ-algebra generated by $\mathbf{x}_t$. Note that $\mathcal{G}$ is a refinement of $\mathcal{F}$.

For an integrable random variable $r(\mathbf{x}_0)$, we have:

$$\mathbb{E}\left[r(\mathbf{x}_0) \mid \mathcal{F}\right] = \mathbb{E}\left[\mathbb{E}\left[r(\mathbf{x}_0) \mid \mathcal{G}\right] \mid \mathcal{F}\right]$$

Applying this to our context:

$$\mathbb{E}_{\mathbf{x}_{t-\Delta}|\mathbf{x}_t}\left[r_\beta(\mathbf{x}_{t-\Delta})\right] = \mathbb{E}\left[\mathbb{E}\left[r(\mathbf{x}_0) \mid \mathcal{G}\right] \mid \mathcal{F}\right] \tag{35}$$

$$= \mathbb{E}\left[r(\mathbf{x}_0) \mid \mathcal{F}\right] \tag{36}$$

$$= \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}\left[r(\mathbf{x}_0)\right] \tag{37}$$

$$= r_\beta(\mathbf{x}_t) \tag{38}$$

Therefore, we conclude that:

$$r_\beta(\mathbf{x}_t) = \mathbb{E}_{\mathbf{x}_{t-\Delta}|\mathbf{x}_t}\left[r_\beta(\mathbf{x}_{t-\Delta})\right]$$

□

D.3 Tanh Demon

We provide the theoretical idea behind the development of the algorithm. To start with, there exists a linear relationship between the reward estimate increment from $\mathbf{x}_t$ to $\hat{\mathbf{x}}_{t-\Delta}^{(k)}$ and the injected noise $\mathbf{z}^{(k)}$, which can be derived from Itô's lemma Ito et al. (1951) and Kolmogorov backward equations Kolmogoroff (1931), as follows:

$$r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - r_\beta(\mathbf{x}_t) = g(t)\nabla_{\mathbf{x}_t} r_\beta \cdot \mathbf{z}^{(k)}\sqrt{\Delta} + o(\Delta), \quad \text{where} \quad \hat{\mathbf{x}}_{t-\Delta}^{(k)} = \mathrm{heun}(\mathbf{x}_t, \mathbf{z}^{(k)}, t, \Delta), \tag{39}$$

which can be interpreted from an SDE with the following Lemma.

Claim 1. *Let $r_\beta(\mathbf{x}_t, t) = \mathbb{E}_{\mathbf{x}_0|\mathbf{x}_t}[r(\mathbf{x}_0) \mid \mathbf{x}_t]$ be the expected future value of a function r at time 0, given the current state $\mathbf{x}_t$ at time t. Then, under the SDE:*

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}_\beta \,\mathrm{d}t + g_\beta \,\mathrm{d}\omega_t, \tag{40}$$

the differential of r_β is:

$$\mathrm{d}r_\beta = g_\beta \,\nabla_{\mathbf{x}_t} r_\beta \cdot \mathrm{d}\omega_t. \tag{41}$$

Proof. We begin by introducing a change of variables. Let $s = t_{\text{max}} - t$, so that as t decreases from t_{max} to 0, s increases from 0 to t_{max}. This allows us to consider a forward-time process with standard Brownian motion $\tilde{\omega}_s$.

Given the original SDE, we can write:

$$\mathrm{d}\mathbf{x}_s = -\mathbf{f}_\beta \,\mathrm{d}s + g_\beta \,\mathrm{d}\tilde{\omega}_s, \tag{42}$$

where $\tilde{\omega}_s$ is the standard Brownian motion.

Now, applying Itô's lemma to $r_\beta(\mathbf{x}_s, s)$:

$$\mathrm{d}r_\beta = \left(\frac{\partial r_\beta}{\partial s} - \mathbf{f}_\beta \cdot \nabla_{\mathbf{x}_s} r_\beta + \frac{1}{2} g_\beta^2 \nabla^2 r_\beta \right) \mathrm{d}s + g_\beta \nabla_{\mathbf{x}_s} r_\beta \cdot \mathrm{d}\tilde{\omega}_s. \tag{43}$$

We aim to prove the Kolmogorov backward equation:

$$\frac{\partial r_\beta}{\partial s} - \mathbf{f}_\beta \cdot \nabla_{\mathbf{x}_s} r_\beta + \frac{1}{2} g_\beta^2 \nabla^2 r_\beta = 0. \tag{44}$$

To do so, we integrate Itô's lemma from s to t_{max}:

$$\begin{aligned} r_\beta(\mathbf{x}_{t_{\text{max}}}) - r_\beta(\mathbf{x}_s) &= \int_s^{t_{\text{max}}} \mathrm{d}r_\beta \qquad (45) \\ &= \int_s^{t_{\text{max}}} \left(\frac{\partial r_\beta}{\partial s'} - \mathbf{f}_\beta \cdot \nabla_{\mathbf{x}_{s'}} r_\beta + \frac{1}{2} g_\beta^2 \nabla^2 r_\beta \right) \mathrm{d}s' \\ &\quad + \int_s^{t_{\text{max}}} g_\beta \nabla_{\mathbf{x}_{s'}} r_\beta \cdot \mathrm{d}\tilde{\omega}_{s'}. \qquad (46) \end{aligned}$$

Since $r_\beta(\mathbf{x}_{t_{\text{max}}})$ is a martingale, by taking the expectation (conditioned on $\mathbf{x}_s$) on both sides, we obtain:

$$\begin{aligned} 0 &= \mathbb{E}_{\mathbf{x}_{t_{\text{max}}}|\mathbf{x}_s} \left[r_\beta(\mathbf{x}_{t_{\text{max}}}) - r_\beta(\mathbf{x}_s) \right] \qquad (47) \\ &= \mathbb{E}_{\mathbf{x}_{t_{\text{max}}}|\mathbf{x}_s} \left[\int_s^{t_{\text{max}}} \left(\frac{\partial r_\beta}{\partial s'} - \mathbf{f}_\beta \cdot \nabla_{\mathbf{x}_{s'}} r_\beta + \frac{1}{2} g_\beta^2 \nabla^2 r_\beta \right) \mathrm{d}s' \right] \\ &\quad + \mathbb{E}_{\mathbf{x}_{t_{\text{max}}}|\mathbf{x}_s} \left[\int_s^{t_{\text{max}}} g_\beta \nabla_{\mathbf{x}_{s'}} r_\beta \cdot \mathrm{d}\tilde{\omega}_{s'} \right]. \qquad (48) \end{aligned}$$

The expectation of the stochastic integral is zero, as Itô integrals have a mean of zero:

$$\mathbb{E}_{\mathbf{x}_{t_{\text{max}}}|\mathbf{x}_s} \left[\int_s^{t_{\text{max}}} g_\beta \nabla_{\mathbf{x}_{s'}} r_\beta \cdot \mathrm{d}\tilde{\omega}_{s'} \right] = 0. \tag{49}$$

Thus, we are left with:

$$\mathbb{E}_{\mathbf{x}_{t_{\text{max}}}|\mathbf{x}_s} \left[\int_s^{t_{\text{max}}} \left(\frac{\partial r_\beta}{\partial s'} - \mathbf{f}_\beta \cdot \nabla_{\mathbf{x}_{s'}} r_\beta + \frac{1}{2} g_\beta^2 \nabla^2 r_\beta \right) \mathrm{d}s' \right] = 0. \tag{50}$$

Since the expectation is zero for any interval $[s, t_{\text{max}}]$, the integrand itself must be zero:

$$\frac{\partial r_\beta}{\partial s} - \mathbf{f}_\beta \cdot \nabla_{\mathbf{x}_s} r_\beta + \frac{1}{2} g_\beta^2 \nabla^2 r_\beta = 0. \tag{51}$$

Thus, the differential of r_β is given by:

$$\mathrm{d}r_\beta = g_\beta \nabla_{\mathbf{x}_s} r_\beta \cdot \mathrm{d}\tilde{\omega}_s, \tag{52}$$

Returning to the original time variable t, we substitute $s = t_{\text{max}} - t$ yielding:

$$\mathrm{d}r_\beta = g_\beta \nabla_{\mathbf{x}_t} r_\beta \cdot \mathrm{d}\omega_t, \tag{53}$$

completing the proof.

□

Although $g(t)\nabla_{\mathbf{x}_t} r_\beta$ is inaccessible without distillation and thus an intractable static vector, we can still leverage the linear relationship to derive applications. Using our standard approach of interpreting $r \circ \mathbf{c}$ as r_β and recognizing that $r_\beta(\mathbf{x}_{t-\Delta})$ is an unbiased estimator of $r_\beta(\mathbf{x}_t)$ (from Appendix D.2), we practically interpret Equation (39) as:

$$(r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - \hat{\mu} \approx g(t)\nabla_{\mathbf{x}_t} r_\beta \cdot \mathbf{z}^{(k)}\sqrt{\Delta}, \quad \text{where} \quad \hat{\mu} = \frac{1}{K}\sum_{k=1}^{K}(r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)}). \tag{54}$$

From Equation (39), flipping the sign of $\mathbf{z}^{(k)}$ reverses its contribution to r_β. Therefore, based on the observation $(r \circ \mathbf{c})(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - \hat{\mu}$, we flip $\mathbf{z}^{(k)}$ accordingly. We show the theoretical analysis and proof for the error of the reward estimate of our Tanh Demon in the following.

Lemma 2. *Assume the truncation error terms in Equation* (39) *is ignorable. Let* $\mathbf{z}^*$ *be derived from* $b_k^{\tanh} = \tanh\left(\frac{r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - r_\beta(\mathbf{x}_t)}{\tau}\right)$ *using Equation* (10) *for every numerical step from* t *to* $t-\Delta$ *and* $\tau < \infty$. *Then, almost surely,* $r(\hat{\mathbf{x}}_0^{\tanh}) > r_\beta(\mathbf{x}_t)$, *where* $\hat{\mathbf{x}}_0^{\tanh}$ *is derived by applying* $\mathbf{z}^*$ *on every step.*

Let $\boldsymbol{\ell} = g(t)\nabla_{\mathbf{x}_t} r_\beta$. Recall that we assume

$$r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - r_\beta(\mathbf{x}_t) = \boldsymbol{\ell} \cdot \mathbf{z}^{(k)}\sqrt{\Delta} \tag{55}$$

$$r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{\tanh}) - r_\beta(\mathbf{x}_t) = \boldsymbol{\ell} \cdot \mathbf{z}^{*}\sqrt{\Delta} \tag{56}$$

$$\hat{\mathbf{x}}_{t-\Delta}^{(k)} = \text{heun}(\mathbf{x}_t, \mathbf{z}^{(k)}, t, \Delta) \tag{57}$$

$$\hat{\mathbf{x}}_{t-\Delta}^{\tanh} = \text{heun}(\mathbf{x}_t, \mathbf{z}^{*}, t, \Delta) \tag{58}$$

$$\mathbf{z}^* = \sqrt{N}\,\text{normalized}\left(\sum_{k=1}^{K} b_k^{\tanh}\mathbf{z}^{(k)}\right) \tag{59}$$

$$b_k^{\tanh} = \tanh\left(\frac{r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - r_\beta(\mathbf{x}_t)}{\tau}\right). \tag{60}$$

We aim to prove the sufficient condition: $r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{\tanh}) > r_\beta(\mathbf{x}_t)$ for each numerical step. Under a rotation of basis, without loss of generality, we assume $\boldsymbol{\ell}$ only has value in the first component, i.e., $\boldsymbol{\ell} = (\ell, 0, \ldots, 0)$ and $\ell > 0$. We have:

$$r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{\tanh}) > r_\beta(\mathbf{x}_t) \iff \ell z_1^*\sqrt{\Delta} > 0 \tag{61}$$

Claim 2. *Almost surely, the first component* z_1^* *of* $\mathbf{z}^*$ *is positive.*

Proof. Since

$$b_k^{\tanh} = \tanh\left(\frac{r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(k)}) - r_\beta(\mathbf{x}_t)}{\tau}\right) \tag{62}$$

$$= \tanh\left(\frac{\boldsymbol{\ell} \cdot \mathbf{z}^{(k)}\sqrt{\Delta}}{\tau}\right) \tag{63}$$

$$= \tanh\left(\frac{\ell z_1^{(k)}\sqrt{\Delta}}{\tau}\right), \tag{64}$$

where $z_1^{(k)}$ is the first component of $\mathbf{z}^{(k)}$.

Almost surely, $z_1^{(k)} \neq 0$, so $b_k^{\tanh}$ will have the same sign as $z_1^{(k)}$. This implies $b_k^{\tanh} z_1^{(k)} > 0$.

Since the first component of $\mathbf{z}^*$ will have the same sign as the first component of $\sum_{k=1}^{K} b_k^{\tanh}\mathbf{z}^{(k)}$ i.e. $\sum_{k=1}^{K} b_k^{\tanh} z_1^{(k)} > 0$. We conclude that $z_1^* > 0$.

□

In addition, we provide proof of the linear relationship presented in Equation (39).

D.4 Boltzmann Demon

Recall that

$$\mathbf{x}_{t-\Delta} \coloneqq \mathbf{x}_t + \int_t^{t-\Delta} \mathbf{f}_\beta(\mathbf{x}_u, u)\,\mathrm{d}u + g_\beta(u)\,\mathrm{d}\omega_u \tag{65}$$

$$\tilde{\mathbf{x}}_{t-\Delta} \coloneqq \mathbf{x}_t - \mathbf{f}_\beta(\mathbf{x}_t, t)\Delta + g_\beta(t)\mathbf{z}\sqrt{\Delta} \tag{66}$$

$$\hat{\mathbf{x}}_{t-\Delta} \coloneqq \mathbf{x}_t - \frac{1}{2}\left[\mathbf{f}_\beta(\mathbf{x}_t, t) + \mathbf{f}_\beta(\tilde{\mathbf{x}}_{t-\Delta}, t-\Delta)\right]\Delta + \frac{1}{2}\left[g_\beta(t) + g_\beta(t-\Delta)\right]\mathbf{z}\sqrt{\Delta} \tag{67}$$

We first present the theoretical analysis and proof for the reward estimate error of the proposed Boltzmann Demon as follows.

Lemma 3. *Assume t is bounded by $t_{\max}$, $|r_\beta|$ is bounded by L. Given $\boldsymbol{x}_t$, if the truncation error per Heun's SDE step in Equation* (67) *is $\mathbf{x}_{t-\Delta} = \hat{\mathbf{x}}_{t-\Delta} + o(\Delta)$ as $\Delta \to 0^+$, then we have:*

$$\mathbb{E}\left[r(\hat{\boldsymbol{x}}_0^{\text{boltz}})\right] \geq r_\beta(\boldsymbol{x}_t) - o(L \cdot t_{\max}), \tag{68}$$

where the expectation denotes that each step of the numerical approximation from every t to $t+\Delta$ is taken with the maximum value of $r_\beta(\cdot)$ among i.i.d. SDE samples $\hat{\boldsymbol{x}}_{t+\Delta}^{(k)}$, representing the Boltzmann Demon with $\tau = 0$.

Lemma 3 establishes a lower bound based on the sample maximum and reward estimate accuracy, providing an improvement guarantee of expected reward in expectation.

We first claim the following statement.

Claim 3.

$$\mathbb{E}\left[r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{\text{boltz}})\right] \geq r_\beta(\mathbf{x}_t) - o(L \cdot \Delta). \tag{69}$$

The rest is the induction of SDE time steps $t_0 = t > \cdots > t_{T-2} > t_{T-1} > t_T = 0$, i.e.,

$$\begin{aligned}
\mathbb{E}\left[r(\hat{\mathbf{x}}_0^{\text{boltz}})\right] &= \mathbb{E}\left[r_\beta(\hat{\mathbf{x}}_0^{\text{boltz}})\right] && (70)\\
&\geq \mathbb{E}\left[r_\beta(\hat{\mathbf{x}}_{t_{T-1}}^{\text{boltz}})\right] - o(L \cdot t_{T-1}) && (71)\\
&\geq \mathbb{E}\left[r_\beta(\hat{\mathbf{x}}_{t_{T-2}}^{\text{boltz}})\right] - o(L \cdot (t_{T-1} + (t_{T-2} - t_{T-1}))) && (72)\\
&\vdots && (73)\\
&\geq \mathbb{E}\left[r_\beta(\hat{\mathbf{x}}_t^{\text{boltz}})\right] - o(L \cdot t) && (74)\\
&\geq r_\beta(\hat{\mathbf{x}}_t^{\text{boltz}}) - o(L \cdot t_{\max}) && (75)
\end{aligned}$$

Proof. We list the premise as the following:

$$\mathbf{z}^{(k)} = \omega_{t-\Delta}^{(k)} - \omega_t^{(k)} \tag{76}$$

$$\hat{\mathbf{x}}_{t-\Delta}^{(k)} = \text{heun}(\mathbf{x}_t, \mathbf{z}^{(k)}, t, \Delta) \tag{77}$$

$$\hat{\mathbf{x}}_{t-\Delta}^{(k)} = \mathbf{x}_{t-\Delta}^{(k)} - o(\Delta) \tag{78}$$

$$r_\beta(\mathbf{z}^{\text{boltz}}) = \max\{r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(1)}), \cdots, r_\beta(\hat{\mathbf{x}}_{t-\Delta}^{(K)})\}. \tag{79}$$

We can deduce that:

$$\mathbb{E}\left[r_\beta(\hat{\mathbf{x}}^{\text{boltz}}_{t-\Delta})\right] = \mathbb{E}\left[\max\{r_\beta(\hat{\mathbf{x}}^{(1)}_{t-\Delta}), \cdots, r_\beta(\hat{\mathbf{x}}^{(K)}_{t-\Delta})\}\right] \tag{80}$$

$$\geq \mathbb{E}\left[r_\beta(\hat{\mathbf{x}}^{(1)}_{t-\Delta})\right] \tag{81}$$

$$= \mathbb{E}\left[r_\beta(\mathbf{x}^{(1)}_{t-\Delta}) - L \cdot o(\Delta)\right] \tag{82}$$

$$= r_\beta(\mathbf{x}_t) - o(L \cdot \Delta) \tag{83}$$

The last equation is followed by Equation (34). Here, $r_\beta(\hat{\mathbf{x}}_{t-\Delta})$ is the numerical estimation of the underlying SDE value $r_\beta(\mathbf{x}_{t-\Delta})$. □

Lemma 4. *When $\tau = \infty$ and the time step is small enough, the Boltzmann Demon sampling is identically distributed as the SDE sampling.*

By adjusting τ, we can smoothly transition from prioritizing high-reward noise samples to the standard SDE sampling method, balancing Demon and SDE strategies; note that when $\tau = \infty$, the weights are $b_k = \exp(0) = 1$. Thus, $\sum_{k=1}^{K} b_k \mathbf{z}_k$ results in a Gaussian distribution $\mathcal{N}(\mathbf{0}, K\mathbf{I}_N)$. This distribution is identical distributed to drawing a Gaussian after both are projected onto a sphere of radius $\sqrt{N}$.

We justify replacing Gaussian sampling with uniform sampling from a sphere of radius $\sqrt{N}$ could result in the same effect of SDE during the Euler-Maruyama discretization of SDEs. Assuming constant drift $\mathbf{f}$ and diffusion g for Euler-Maruyama step, the SDE is $d\mathbf{x} = \mathbf{f}\, dt + g\, d\mathbf{W}$. We aim to demonstrate that this replacement yields an identical distribution under small step sizes. Define:

$$\mathbf{Y}_n = -\mathbf{f}\Delta + \sum_{k=1}^{n} g\sqrt{\frac{\Delta}{n}}\tilde{\mathbf{z}}_k = -\mathbf{f}\Delta + g\sqrt{\Delta}\frac{1}{\sqrt{n}}\sum_{k=1}^{n}\tilde{\mathbf{z}}_k \tag{84}$$

where $\tilde{\mathbf{z}}_k$ are i.i.d. vectors uniformly sampled from the surface of a sphere with radius $\sqrt{N}$. Also, define:

$$\mathbf{Y} = -\mathbf{f}\Delta + g\sqrt{\Delta}\mathbf{z} \tag{85}$$

Claim 4. *$\mathbf{Y}_n$ converges to $\mathbf{Y}$ in distribution as $n \to \infty$.*

Proof. It's sufficient to determine whether the normalized sum $\frac{1}{\sqrt{n}}\sum_{k=1}^{n}\tilde{\mathbf{z}}_k$ converges to $\mathbf{z}$ in distribution. Consider each $\tilde{\mathbf{z}}_k$ as an i.i.d. vector uniformly distributed on the surface of a sphere with radius $\sqrt{N}$. The expectation of each vector is $\mathbb{E}[\tilde{\mathbf{z}}_k] = \mathbf{0}$ by symmetry of the uniform distribution on the sphere.

We need to establish the covariance matrix of $\tilde{\mathbf{z}}_k$. Let $\tilde{\mathbf{z}}$ represent any of these i.i.d vectors without the subscript to generalize the analysis. We investigate the covariance between any two components $\tilde{\mathbf{z}}^{(i)}$ and $\tilde{\mathbf{z}}^{(j)}$ of the vector $\tilde{\mathbf{z}}$:

If $i \neq j$, by the symmetric condition of the sphere, we have:

$$\mathbb{E}\left[\tilde{\mathbf{z}}^{(i)}\tilde{\mathbf{z}}^{(j)}\right] = \mathbb{E}\left[\tilde{\mathbf{z}}^{(i)}\mathbb{E}\left[\tilde{\mathbf{z}}^{(j)} \mid \tilde{\mathbf{z}}^{(i)}\right]\right] = \mathbb{E}\left[\tilde{\mathbf{z}}^{(i)} \cdot 0\right] = 0, \tag{86}$$

reflecting the orthogonality of different components.

For the diagonal entries, where $i = j$, we know that the sum of the squares of the components of $\tilde{\mathbf{z}}$ equals the square of the radius of the sphere:

$$\sum_{i=1}^{N}\mathbb{E}\left[(\tilde{\mathbf{z}}^{(i)})^2\right] = \mathbb{E}\left[\sum_{i=1}^{N}(\tilde{\mathbf{z}}^{(i)})^2\right] = N, \tag{87}$$

since $\|\tilde{\mathbf{z}}\|^2 = N$. This implies that each component $\tilde{\mathbf{z}}^{(i)}$ has an expected squared value of 1, because all components contribute equally due to the symmetry of the sphere. Therefore, $\mathrm{Var}[\tilde{\mathbf{z}}^{(i)}] = \mathbb{E}\left[(\tilde{\mathbf{z}}^{(i)})^2\right] = 1$ for all i.

Combining these results, the covariance matrix $\mathrm{Var}(\tilde{\mathbf{z}}) = \mathbf{I}_N$. The Central Limit Theorem for vector-valued random variables (as described in Rencher (2005)) then asserts that $\frac{1}{\sqrt{n}} \sum_{k=1}^{n} \tilde{\mathbf{z}}_k$ converges in distribution to a Gaussian vector $\mathbf{z}$ with mean $\mathbf{0}$ and covariance matrix $\mathbf{I}_N$ as $n \to \infty$. Hence, the normalized sum approximates the Gaussian vector $\mathbf{z}$ used in the diffusion term of the original SDE, which justifies the replacement of Gaussian sampling with uniform sampling from the sphere. □

D.5 High Dimensional Gaussian on Sphere

The original statement is more general in the textbook, but we provide specific proof for Gaussian.

Lemma 5. *(Vershynin, 2020, Chap. 3) Let $\mathbf{z}$ be independent and identically distributed (i.i.d.) instances of a standard isotropic Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ in a high-dimensional space N. With a high probability (e.g., 0.9999), it holds that*

$$\|\mathbf{z}\| = \sqrt{N} + \mathcal{O}(1) \tag{88}$$

Proof. Consider the norm $\|\mathbf{z}\|^2$, where $\mathbf{z}$ is an instance of a standard isotropic Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}_N)$ in N dimensions. The distribution of $\|\mathbf{z}\|^2$ follows a Chi-squared distribution with N degrees of freedom. The mean and variance of this distribution are N and $2N$, respectively.

Applying a central limit theorem argument, we approximate the distribution of $\|\mathbf{z}\|^2$ by a normal distribution when N is large, giving:

$$\|\mathbf{z}\|^2 = N + C\sqrt{N} \tag{89}$$

for some constant C, where $C \in \mathcal{O}(1)$ represents fluctuations around the mean which are typically on the order of the standard deviation of $\|\mathbf{z}\|^2$, which is $\sqrt{2N}$.

To connect this with the norm of $\mathbf{z}$, we consider:

$$\lim_{N\to\infty} \sqrt{N + C\sqrt{N}} - \sqrt{N} = \lim_{N\to\infty} \sqrt{N}\left(\sqrt{1 + \frac{C}{\sqrt{N}}} - 1\right) \tag{90}$$

$$= \lim_{N\to\infty} \sqrt{N}\left(\frac{C}{2\sqrt{N}}\right) \tag{91}$$

$$= \frac{C}{2} \tag{92}$$

Here, we use the Taylor series expansion for $\sqrt{1+x}$, approximated as $1 + \frac{x}{2}$ for small x, to find the limit. This expansion leads to the conclusion that $\|\mathbf{z}\| = \sqrt{N} + \mathcal{O}(1)$. □

E Comparison on PickScore

E.1 PickScore Comparisons.

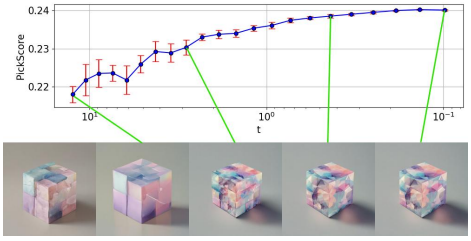
Since PickScore Kirstain et al. (2023) is trained specifically on generated images, we believe it is a more reliable measure and objective than the aesthetics score. To emphasize the strength of our method, we show how the median PickScore reward function improves across 20 different prompts using our Tanh Demon, as shown in Figure 7a.

Our approach utilizes 1440 reward queries per sample and achieves a PickScore of 0.253, outperforming other methods alongside reduced computation time (180 minutes for our method vs. 240 minutes for resampling methods due to shortened ODE trajectories). Specifically, we compare our method to:

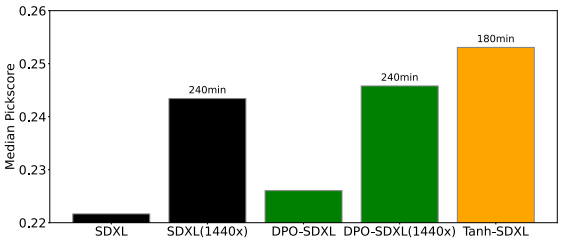
- **SDXL/SDXL-DPO** Wallace et al. (2023a): A state-of-the-art method for direct preference optimization in diffusion models, which achieves a PickScore of 0.226, while the baseline SDXL reaches 0.222.
- **Diffusion-DPO(1440x)**: A variant that selects the highest quality median PickScore from 1440 samples among 20 prompts, achieving a PickScore of 0.246.

- **SDXL(1440x)**: Similar to the above, but without preference optimization, achieving a PickScore of 0.243.

Additionally, resampling an ODE from $\mathbf{x}_{t_{\max}}$ is crucial in applications where the distribution $\mathbf{x}_{t_{\max}} \mid \mathbf{x}_0$ plays a key role, such as in SDEdit Meng et al. (2022). Resampling methods fail to address such applications, highlighting the advantage of our approach.



(a) A Trajectory of Tanh Demon. We plot $(r \circ \mathbf{c})(\mathbf{x}_t)$ for different t.



(b) The performance of each method on PickScore.

Figure 7: Quantitative results for Tanh Demon.

E.2 Qualitative Results

In this section, we demonstrate the quantitative and qualitative results of PickScore in SDXL with our Tanh Demon.

F Comparison on HPSv2

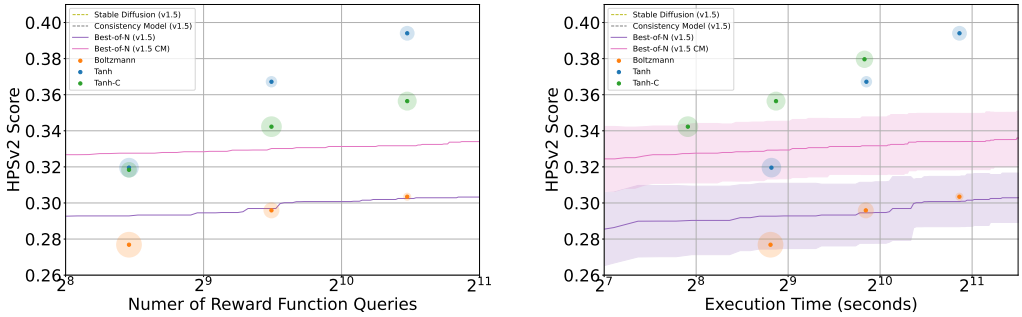
Table 11: We present qualitative results for various methods. For our method, we set $T = 128$, $\beta = 0.5$, and $K = 16$. The Best-of-N samples are generated using 2336 (5440 for CM) reward queries and 3.8k seconds, which is significantly more than our method's 1,424 reward queries and 1.8k seconds. Moreover, the presented image from Best-of-N possesses an inferior HPSv2 score compared to ours.

Best-of-N	**Best-of-N (CM)**	**Tanh-C**	**Tanh**	**Boltzmann**
a castle is in the middle of a eurpean city				
A motorcycle that is sitting in the dirt.				

We present quantitative and qualitative results in Figure 9 and table 11, using 10 prompts sampled from HPDv2 (Wu et al., 2023). Both the diffusion model and CM are implemented and distilled with SD v1.5.

We observe similar results in Figure 4. Regarding reward queries, the Tanh Demon method outperforms Tanh-C, followed by the Boltzmann Demon method. Regarding execution time, however, Tanh-C is recommended over the Tanh Demon if computational time is limited.

Figure 8: Each row in the figure presents two pairs of images where the image of each pair on the left illustrates results generated using the original PF-ODE method. The image on the right in each pair showcases enhancements achieved by applying our Tanh Demon based on the PickScore metric and SDXL. This figure demonstrates the improvements in visual fidelity and adherence to targeted characteristics achieved through our proposed method.



(a) Performance w.r.t Reward Query Number

(b) Performance w.r.t Execution Time

Figure 9: The performance comparison of the proposed algorithm and the best-of-N baseline methods is presented in terms of the number of reward queries and execution time, with the dependent variable being T. The shaded areas and the solid circle radii represent the evaluation results' standard deviations. If the computational bottleneck is the number of reward queries, we recommend Tanh; if it is computational time, we recommend Tanh-C.

G Additional Results with Various Reward Functions.

G.1 Image Generation Results with Different Reward Functions

We show more image generation results in SDv1.4/SDXL with our Tanh Demon and other reward functions in Tables 12 to 15, using the four reward as objective.

G.2 Subjective Test Overview

We surveyed with 101 participants via Google Forms, as shown in Figure 10. Participants evaluated different image generation methods based on:

- **Subjective Preference**: Visual aesthetics and image quality.
- **Semantic Alignment**: Correspondence between generated images and text prompts.

Each participant ranked images across four sections, with rankings aggregated using the following formula:

$$\frac{1}{ML}\sum_{i=1}^{M}\sum_{j=1}^{L}\exp(-(\text{rank}_{ij}-1)) \tag{93}$$

where:

- $M = 4$ (number of sections),
- $L = 101$ (participants),
- rank_{ij} is the ranking by participant j for method i.

G.2.1 Survey Structure

The subjective test comprised four sections: two comparing methods (DOODL, Baseline (SD or SDXL), Ensemble) based on subjective preference and prompt alignment, each with 3 sets containing one image per method; and two comparing methods applied to different objectives (Baseline, Ensemble, IR, Pick, HPSv2, Aes) also based on preference and prompt alignment, each with 3 sets containing six images per set.

Table 12: Generative Results using SDXL

SDXL	Aes	IR	Pick	HPSv2	Ensemble
An Octopus Playing Chess with a Robot Underwater					
A Samurai Gardening on a Floating Island in the Sky					
Insanely detailed portrait, wise man					
A painting of a girl encountering a giant sunflower blocking her path in a hallway					
A demon exiting through a portal					
A butterfly flying above an ocean					

Table 13: Generative Results using SDXL (Cont)

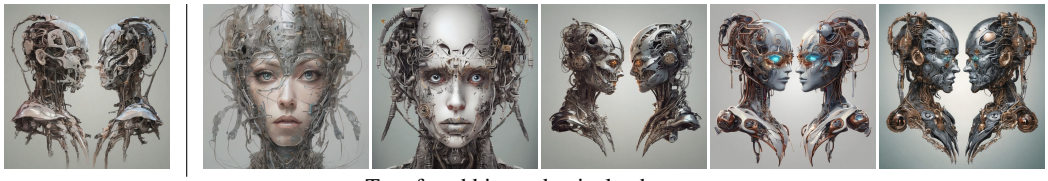
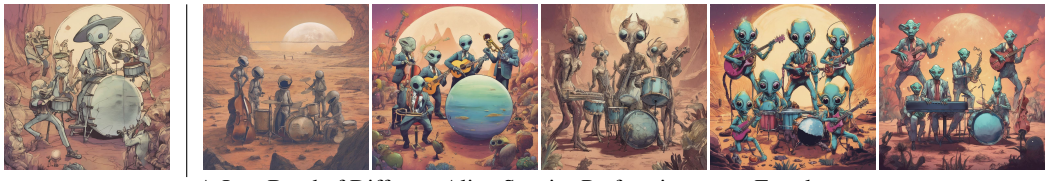
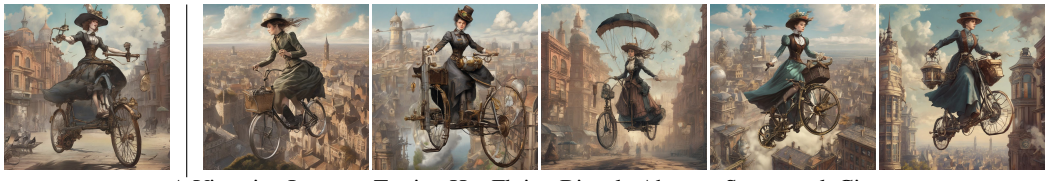
SDXL	Aes	IR	Pick	HPSv2	Ensemble
					
Two-faced biomechanical cyborg					
Highway to hell					
					
A Jazz Band of Different Alien Species Performing on an Exoplanet					
					
A Victorian Inventor Testing Her Flying Bicycle Above a Steampunk City					
A Time Traveler's Picnic at the Edge of a Volcano During the Mesozoic Era					
jedi duck holding a lightsaber					

Table 14: More Qualitative Results of SD1.4

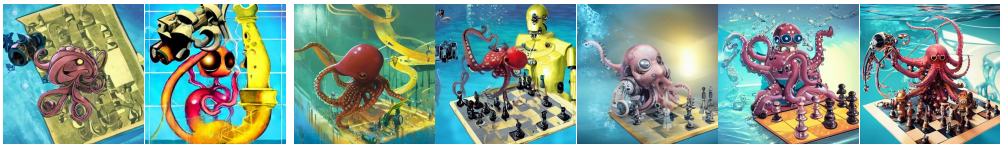
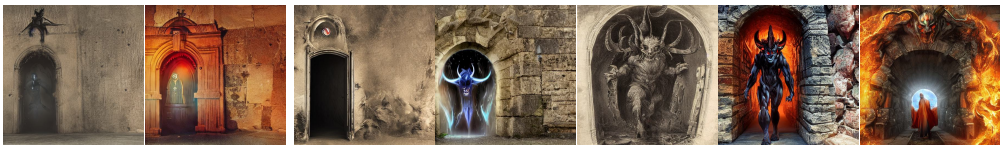
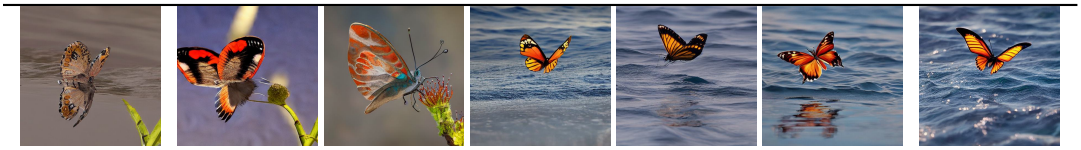
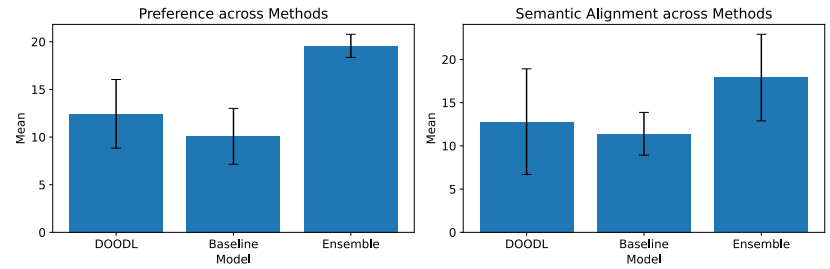
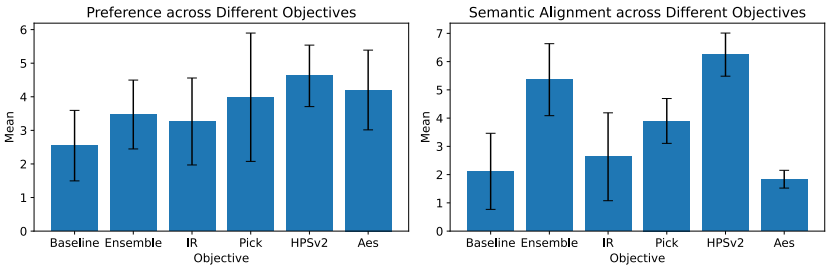
Baseline	DOODL	Aes	IR	Pick	HPSv2	Ensemble
						
An Octopus Playing Chess with a Robot Underwater						
Two-faced biomechanical cyborg						
Highway to hell						
jedi duck holding a lightsaber						
						
A demon exiting through a portal						
A Samurai Gardening on a Floating Island in the Sky						

Table 15: More Qualitative Results of SD1.4 (cont)

Baseline	Aes	IR	Pick	HPSv2	Ensemble	DOODL
A Victorian Inventor Testing Her Flying Bicycle Above a Steampunk City						
A Time Traveler's Picnic at the Edge of a Volcano During the Mesozoic Era						
Insanely detailed portrait, wise man						
						
A butterfly flying above an ocean						
A Jazz Band of Different Alien Species Performing on an Exoplanet						
A painting of a girl encountering a giant sunflower blocking her path in a hallway						



(a) Comparison across methods.



(b) Comparison across objectives.

Figure 10: Subjective test results: Preferences and prompt alignment across methods and objectives.

G.2.2 Results Overview

Methods Comparison Figure 10a shows that DOODL slightly outperforms the Baseline in aesthetic preference and prompt alignment. The Ensemble method significantly surpasses both, indicating superior visual quality and semantic accuracy.

Objectives Comparison As seen in Figure 10b, all objectives outperform the Baseline in prompt alignment, with the HPSv2 method leading. In subjective preference, methods applied to different objectives show varied improvements, with some achieving substantial gains over the Baseline.

G.2.3 Analysis

We compared DOODL, Baseline, and Ensemble based on aesthetics and prompt alignment. DOODL marginally improves over the Baseline in both criteria, while the Ensemble method consistently outperforms both DOODL and Baseline, excelling in image quality and semantic accuracy. The Ensemble method demonstrates significant enhancements, particularly in tasks requiring visual refinement.

Evaluating different objectives (IR, Pick, HPSv2, Aes) against Baseline and Ensemble revealed that almost all objectives surpass the Baseline in both preference and prompt alignment. However, Aes, an objective without explicit text guidance, shows weaker prompt alignment. Among the objectives, HPSv2 achieves the best performance on both criteria.

The Ensemble method provides the most substantial improvements in visual aesthetics and semantic alignment among method comparisons. Among the factors of the Ensemble method, HPSv2 outperforms other objectives, even the Ensemble method, highlighting its effectiveness in aligning preference for a real human.

H More Details of VLM as Demon

In this section, we provide more details of experiments and quantitative results of utilizing VLM during generation.

Table 16: Using VLMs to generate images. PF-ODE (baseline) refers to a baseline without using our method for alignment. The top row of each set indicates the agent's role in the given prompt. The hyperparameters are set to $\beta = 0.1$, $\tau = 0.0001$, $K = 16$, and $T = 128$ using Tanh Demon. We apply ODE after applying 15 Demon steps.

	Teacher			Artist	
PF-ODE	**GPT**	**Gemini**	**PF-ODE**	**GPT**	**Gemini**
13.7%	17.7%	68.6%	24.3%	27.2%	48.5%

	Researcher			Journalist	
PF-ODE	**GPT**	**Gemini**	**PF-ODE**	**GPT**	**Gemini**
9.3%	69.4%	21.3%	32.7%	33.1%	34.2%

Role	Prompts
Teacher	*A colorful, labeled educational illustration with simple, engaging visuals.*
Artist	*A detailed, cinematic concept art of unique characters or scenes for games or movies.*
Researcher	*A realistic, vivid visualization of scientific data with a professional style.*
Journalist	*A bold, vibrant illustration with dynamic design for digital platforms.*

H.1 Generation with latest model

In Table 16, we present an experiment aligning diffusion models to preferences of VLMs from APIs—Google Gemini Flash v1.5 (Gemini Team Google, 2024) and GPT-4o (OpenAI, 2024)—as a demonstration of using non-differentiable reward sources. In each setting, the VLM receives a given scenario, e.g., "You are a journalist who wants to add a visual teaser for your article to grab attention on social media or your news website," and is asked to select the best-matching state vector $\mathbf{x}_{t-\Delta}^{(k)}$ based on generated images. Each scenario is assigned a prompt. In a generation, we apply Tanh Demon with $K = 16$ on SDXL. For scenarios and quantitative results, please refer to Appendix H.

Empirically, VLMs achieve better accuracy when selecting 1 out of 2 options rather than 8 out of 16. We, hence, utilize binary comparisons by applying the *Quicksort partition* (Hoare, 1962) twice on the array of $\mathbf{c}(\mathbf{x}_{t-\Delta}^{(k)})$ derived from $\mathbf{x}_t$. The first application partitions the entire array, and the second partitions the larger subset resulting from the first step. This process allows us to identify roughly the top 8 images within $2K$ comparison.

We assign a $+1$ reward on the roughly top 8 images and -1 on the rest for each Demon step. Using PickScore (Kirstain et al., 2023), trained based on image comparisons, we convert logits to probabilities and assess the effectiveness of our method. The results indicate improvements with this metric on every image in any degree.

H.2 Experiments Settings

We provide the prompt template we used in Table 5 to VLMs. The following are the full system prompts for the scenarios:

Listing 1: Prompt template used in the Gemini image selection task. This template guides the decision-making process for choosing between two images generated from a fixed prompt. For the prompt used in GPT selection, we replace the output format by asking it to return JSON.

Table 17: Roles and System Prompts

Role	System Prompt
Teacher	*You are a teacher looking to create custom illustrations for your educational materials to make learning more engaging for your students.*
Artist	*You are a game or movie concept artist tasked with creating concept art for characters, settings, and scenes to speed up the pre-production process.*
Researcher	*You are a researcher needing to visualize complex data, such as molecular structures in chemistry or weather patterns in meteorology, for better understanding or presentation.*
Journalist	*You are a journalist who wants to add a visual teaser for your article to grab attention on social media or your news website.*

```
Scenario: {scenario}

You are presented with two images generated from the prompt "{prompt}." Examine both images carefully to decide which one best matches the given scenario. Consider how each image relates to the scenario, including its relevance and how well it captures the intended theme and concept.

Choose the image that you believe is the most appropriate for the scenario. Each image has an id: 0 or 1.

Please share your thought process or any observations you made while making your decision. This reflection helps in understanding your choice.

Respond with the id of the image you've chosen in a JSON format. For example, if you choose the first image, your response should look like this:

```json
[0]
```

Or, if you choose the second image:

```json
[1]
```


Please response in the following format:

< Here write down your argument >

< Here write down your decision, either [0] or [1], the [] is necessary >

Take a deep breath and work on this problem step-by-step. Ensure your choice truly aligns with the intended scenario.
```

Listing 2: An example of GPT generated output

```
{
    "justification": "Considering the scenario described, the
       first image is a better choice. This image effectively
       showcases a mysterious, illuminated object that instantly
       attracts the viewer's curiosity. The background hints at
       an academic or historical setting, with architectural
       elements and various artifacts that seem like elements
       from different times and places. This diverse and
       enigmatic setup aligns well with the theme of
       transformation and curiosity across different realms like
       education, history, literature, and science. The glowing
       object in a seemingly ancient, cluttered environment truly
        sparks wonder, making it ideal for grabbing attention on
       social media or a news website.",
    "chosen_image": [
        0
    ]
}
```

H.3 Quantitative Measurement of Effectiveness

For VLMs as reward functions, we use Pickscore Kirstain et al. (2023), which is trained from CLIP Radford et al. (2021), to evaluate the effectiveness of VLM in aligning designed scenarios during image generation. For each scenario, we create a corresponding prompt that partially describes the scenario: "For education" for Teacher, "For entertainment" for Artist, "For research" for Researcher, and "For Journalism" for Journalist. Then, we assess the PickScore between the prompt and the scenario. The results are presented in Table 18, Table 19, Table 20 and Table 21, where the highest score for each prompt is highlighted in bold. Our observations indicate that 14 out of our VLM-generated 16 images demonstrate better PickScore alignment with the corresponding prompt than PF-ODE. Given that all images are generated using the same prompt and initial noisy sample in the same table, these results demonstrate the effectiveness of our approach employing VLM in aligning the scenarios.

Table 18: GPT-SDXL generation, validated by PickScore on related prompts

Prompt	Teacher	Artist	Researcher	Journalist	PF-ODE
For education	0.2050	0.2069	**0.2080**	0.2071	0.2073
For entertainment	0.2042	**0.2073**	0.2061	0.2058	0.2032
For research	0.1980	0.1989	**0.1996**	0.1985	0.1971
For journalism	**0.1994**	0.1957	0.1978	0.1946	0.1970

Table 19: Gemini-SDXL generation, validated by PickScore on related prompts

Prompt	Teacher	Artist	Researcher	Journalist	PF-ODE
For education	**0.2111**	0.2042	0.2102	0.2072	0.2073
For entertainment	0.2057	0.2058	**0.2062**	0.2013	0.2032
For research	0.2018	0.1979	**0.2035**	0.1986	0.1971
For journalism	0.2011	0.1991	0.1978	**0.2049**	0.1970

Table 20: GPT-SD v1.4 generation, validated by PickScore on related prompts

Prompt	Teacher	Artist	Researcher	Journalist	PF-ODE
For education	0.1978	**0.2008**	0.1996	0.1988	0.1941
For entertainment	**0.2026**	0.2018	0.1991	0.2004	0.1966
For research	0.1896	**0.1936**	0.1912	0.1935	0.1878
For journalism	0.1930	**0.1951**	0.1918	0.1942	0.1901

Table 21: Gemini-SD v1.4 generation, validated by PickScore on related prompts

Prompt	Teacher	Artist	Researcher	Journalist	PF-ODE
For education	**0.1997**	0.1945	0.1961	0.1961	0.1941
For entertainment	0.1954	0.1973	0.1982	**0.1989**	0.1966
For research	**0.1910**	**0.1910**	0.1897	0.1895	0.1878
For journalism	0.1935	**0.1936**	0.1927	0.1914	0.1901

I General Implementation Details

In this section, we show the details of the implementation and experimental settings of the proposed approach as follows.

I.1 Adapting Stable Diffusion to EDM Framework

In this paper, we tailor the existing text-to-image Stable Diffusion v1.4/v1.5/XL v1.0 (SDXL) (i.e., we use fp16 SD v1.4/SDXL v1.0 for generation.) to the SDE formulation proposed in EDM Karras et al. (2022) by Karras et al. for image generation since its reparameterized time domain, $t \in [t_{\min}, t_{\max}]$, improves numerical stability and sample quality during image generation. We realize the modification through the equation, $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}, t) = (D(\boldsymbol{x}, t) - \boldsymbol{x}) / t^2$, where the function $D(\boldsymbol{x}, t) = \boldsymbol{x} - t\mathbf{F}(s(t)\boldsymbol{x}, u(t))$ derived from the original model $\mathbf{F}$. In addition, $s(t)$ and $u(t)$ represent the scaling schedule and the original temporal domain of the reparameterized temporal domain t, respectively.

I.2 Numerical Methods for Image Generation

Moreover, for image sampling with ODE/SDE, our approach follows Karras et al. (2022), adopting Heun's method and time intervals determined by $t_i = \left(t_{\max}^{1/\rho} + \frac{i-1}{T-1}(t_{\min}^{1/\rho} - t_{\max}^{1/\rho})\right)^{\rho}$, setting $\rho = 7, T \geq 20$ and $\ln t_{\max} \approx 2.7, \ln t_{\min} \approx -6.2$. The classifier-free guidance parameter is set to 2 throughout this paper. Across all temporal steps t of image generation, we keep K and β constant. We have found that when t is less than 0.11, i.i.d. samples from SDE all appear similar to human perception. For the remaining evaluations, we will directly use ODE. As a result, the actual number of samples will be slightly smaller than $K \cdot T$.

I.3 Simplifications in Diffusion Process Modification

It is worth noting that in our work, since our main focus is on the modification of the diffusion process, without loss of generality, we omit the VAEs (Kingma & Welling (2014)) of Stable Diffusion models, the prompt c, and η of classifier-free guidance (CFG) Ho & Salimans (2021) in our formulation for simplicity (i.e., using $p(\boldsymbol{x})$ to denote the unnormalized $p(\boldsymbol{x})p(c \mid \boldsymbol{x})^{\eta}$ for conciseness).

I.4 Batch Size and Memory Constraints

When we generate many SDE samples, the batch size for solving ODE/SDE is 8 for both Stable Diffusion v1.4, v1.5, and SDXL models. However, due to memory limitations on the RTX 3090, the batch size for evaluating the VAE in SDXL is restricted to 1. This memory bottleneck prevents any further acceleration from using larger batch sizes, as it limits the parallelization during VAE evaluation.

Due to memory limitations, DOODL was run on an Nvidia RTX A6000, which is slightly slower (0.92x) than the RTX 3090 used for the other experiments.

I.5 Experimental Setup and Hyperparameters

Among all the discussed methods, DOODL is the only one that utilizes the differentiation of the reward function. Due to memory limitations, DOODL was run on an Nvidia RTX A6000, which is slightly slower (0.92x) than the RTX 3090 used for the other experiments. We present the detailed hyperparameter settings of different experiments as follows:

Baseline Comparison. The hyperparameters for generation are set to $\beta = 0.5$, $K = 16$, $\eta = 2$ and τ adaptive for Tanh, 10^{-5} for Boltzmann.

Reward Estimate Approximation Comparison. We set $\beta = 0.5$ and use SD v1.5 and its distilled CM. The CFG parameter is ignored in CM(set to 1). The reward estimate r_β is obtained by averaging over 200 Monte Carlo i.i.d. SDE samples—each with 200 SDE steps.

Generation with Various Reward Functions. We use Tanh Demon for sampling with adaptive temperature. The hyperparameters for generation are set to $\beta = 0.05$, $K = 16$, $T = 64$ as shown in Tables 4, 12 to 14 and 15 on SD v1.4/SDXL.

For reward scaling in the ensemble setting, the PickScore was multiplied by 98.86, and HPSv2 was multiplied by 40.

The interaction step of DOODL is used as suggested by their implementation, 50 iteration for Aes and 100 iteration for Pick.

Non-differentiable Reward. In Table 5, the hyperparameters are set to $\beta = 0.05$, $\tau = 0.0001$, $K = 16$, and $T = 32$ using Tanh Demon.

Manual Selection. In Figure 5, the parameters are $\beta = 0.1, K = 16, T = 128$ but terminate manually, using Tanh Demon with adaptive temperature. We terminate the iteration after ten rounds of operating the UI.

J LIMITATIONS

We present the theoretical result in Equation (5), which demonstrates that $r \circ \mathbf{c} \approx r_\beta$. This result relies on the assumption that the reward function r is near harmonic near the ODE sample ourput, as detailed in Appendix D.1.2.

In practice, implementing $r \circ \mathbf{c}$ faces challenges related to time complexity and accuracy bottlenecks, thoroughly discussed in Section 4.

K FUTURE WORKS

The only difference between Tanh-C and Tanh Demon lies in how $r \circ \mathbf{c}$ is implemented. Analysis of the data in Table 2 and Figure 4 indicates that Tanh-C's reward performance can be enhanced by mitigating the RMSE in $r \circ \mathbf{c}$ without compromising Tanh-C's speed performance. Potential strategies for improvement include increasing the fidelity of CM distillation or training a dedicated distilled model for $r \circ \mathbf{c}$. We propose these enhancements as future work.

L CODE OF ETHICS

The experiments involving human judgment are fully compliant with established ethical standards. Approval is obtained from the Institutional Review Board (IRB) to ensure that the research meets all necessary guidelines for the ethical treatment of human subjects. For anonymity purposes, the IRB approval number will be concealed under double-blind review.

M SOCIETAL IMPACT

Our method has the potential to both discourage and encourage harmful content. Users can generate images through manual selections with malicious intentions (Figure 5). This increases accessibility but also raises concerns about misuse. We implement safeguards provided by Stable Diffusion; end-users are responsible for employing them, as recommended in prior works OpenAI (2024); Gemini Team Google (2024); Rombach et al. (2022); Podell et al. (2024), to mitigate potential risks.