STEP-LEVEL DENOISING-TIME DIFFUSION ALIGNMENT WITH MULTIPLE OBJECTIVES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has emerged as a powerful tool for aligning diffusion models with human preferences, typically by optimizing a single reward function under a KL regularization constraint. In practice, however, human preferences are inherently pluralistic, and aligned models must balance multiple downstream objectives, such as aesthetic quality and text-image consistency. Existing multi-objective approaches either rely on costly multi-objective RL fine-tuning or on fusing separately aligned models at denoising time, but they generally require access to reward values (or their gradients) and/or introduce approximation error in the resulting denoising objectives. In this paper, we revisit the problem of RL fine-tuning for diffusion models and address the intractability of identifying the optimal policy by introducing a *step-level* RL formulation. Building on this, we further propose **Multi-objective Step-level Denoising-time Diffusion Alignment** (MSDDA), a retraining-free framework for aligning diffusion models with multiple objectives, obtaining the optimal reverse denoising distribution in closed form, with mean and variance expressed directly in terms of single-objective base models. We prove that this denoising-time objective is exactly equivalent to the step-level RL fine-tuning, introducing no approximation error. Moreover, we provide numerical results, which indicate our method outperforms existing denoising-time approaches.

1 INTRODUCTION

Diffusion models (Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022) have gained increasing attention in text-to-image generation. However, these models are pre-trained on broad, large-scale datasets and are therefore not tailored to specific downstream tasks. To adapt them to particular application domains, fine-tuning approaches such as supervised fine-tuning (Lee et al., 2023) and reinforcement learning (RL) based methods (Black et al., 2024; Clark et al., 2024; Fan et al., 2023) have been proposed. In RL-based fine-tuning, the goal is to maximize a given reward function by updating a pre-trained model, typically with an additional Kullback–Leibler (KL) regularization term to keep the aligned model close to the pre-trained one.

Despite their success, most RL fine-tuning methods optimize with respect to a *single* reward function. In practice, human preferences are inherently pluralistic; thus, the alignment should balance multiple downstream objectives, such as aesthetic quality and text–image consistency. To address this problem, a multi-objective setting is investigated, in which we have a set of reward functions, and a task-specific reward is given by a preference-weighted combination of these rewards with respect to a weight vector w.

Existing work on multi-objective RL can be applied to diffusion fine-tuning. However, these methods require substantial additional computation, such as fine-tuning a large (often exponential in the size of the reward set) number of models to cover the space of preference weights (Rame et al., 2023; Zhou et al., 2022; Yang et al., 2019) or solving for conflict-avoiding update directions (Wang et al., 2025). To improve training efficiency, denoising-time diffusion alignment has been studied, which avoids training new models and instead fuses the denoising processes of existing aligned models to realize a target output distribution. Although these methods avoid extra training, most of them still require access to reward gradients or repeated estimation of value functions by generating many samples with associated rewards (Clark et al., 2024; Kim et al., 2025; Ye et al., 2024; Singh et al., 2025). To the best of our knowledge, only a few recent works (Cheng et al., 2026; Manujith et al., 2026) avoid

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

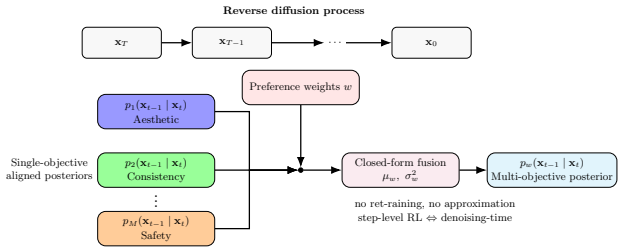


Figure 1: Overview of our proposed Multi-objective Step-level Denoising-time Diffusion Alignment (MSDDA) algorithm.

using reward information and obtain a target model by fusing aligned models corresponding to a set of base rewards; however, the derivation of their denoising-time objectives introduces approximation errors that are difficult to quantify.

In this paper, we aim to address the following question: *Can we design a retraining-free denoising-time alignment method that does not require access to individual reward functions and introduces no additional approximation error?*

We provide fundamental insights into denoising-time alignment and answer the above question firmly. Our contributions are summarized as follows:

- We begin by revisiting existing RL fine-tuning methods for diffusion models, which require sampling from the updated policy. This dependence makes the target policy difficult to track and forces analyses to rely on approximations whose errors are hard to quantify. To overcome this intractability, we propose a novel *step-level* RL fine-tuning formulation. Moreover, we derive the corresponding step-level DPO objective, which trains the model solely from preference pairs and does not require explicit access to the reward function.
- Building on our step-level RL formulation, we design a step-level denoising-time diffusion alignment method for multiple objectives. Without any additional training or access to reward functions, our algorithm can, for any preference weight vector w, compute the optimal reverse denoising distribution in closed form: both its mean and variance are explicit functions of those of the base reward models. We further show theoretically that the solution obtained from our algorithm is exactly equivalent to that of the step-level RL fine-tuning formulation, and therefore introduces *no* additional approximation error.
- We conduct extensive experiments using Stable Diffusion (Rombach et al., 2022) as the pre-trained model, considering multiple reward functions and a wide range of preference weights w. The results show that our method outperforms existing denoising-time approaches.

2 Related Work

Single-objective diffusion alignment: A large number of works study aligning diffusion models to human preferences under a *single* reward. For example, supervised fine-tuning (SFT) methods (Lee et al., 2023; Wu et al., 2023) improve output quality by minimizing a reward-weighted negative log-likelihood on a fixed dataset. These approaches are entirely offline, with training samples provided in advance rather than generated by the current model.

RL-based methods such as DDPO (Black et al., 2024) and DPOK (Fan et al., 2023) instead formulate the reverse diffusion process as a T-horizon Markov Decision Process (MDP), and optimize a terminal-state reward with a KL regularization term to keep the aligned model close to the pre-trained one. Compared with the above methods, DRaFT (Clark et al., 2024) backpropagates the reward gradient to update the diffusion model. Rather than modifying the diffusion model itself, Hao et al. (2023) uses RL to optimize prompts to improve model performance. Building on the RL formulation, diffusion DPO (Wallace et al., 2024) can be viewed as a special case: it does not require explicit reward modeling and on-policy sampling, and instead trains the model directly from human preference pairs.

Table 1: Comparison of existing diffusion denoising-time alignment methods, all of which are training-free. $\surd$ indicates the term is required and $\times$ indicates the term is not required.

METHOD	REWARD GRADIENT	REWARD FUNCTION	APPROXIMATION ERROR
MUDM (HAN ET AL., 2024)	$\surd$	$\surd$	NA
TFG (YE ET AL., 2024)	$\surd$	$\surd$	NA
DAS (KIM ET AL., 2025)	$\times$	$\surd$	NA
CODE (SINGH ET AL., 2025)	$\times$	$\surd$	NA
RS (RAME ET AL., 2023)	$\times$	$\times$	NA
DB-MPA (CHENG ET AL., 2026)	$\times$	$\times$	$\surd$
DERADIFF (MANUJITH ET AL., 2026)	$\times$	$\times$	$\surd$
THIS PAPER	$\times$	$\times$	$\times$

Generic multi-objective alignment: While the above works focus on a single objective, a straightforward strategy for multi-objective diffusion alignment is to use linear scalarizations (Roijers et al., 2013; Yang et al., 2019; Zhou et al., 2022; Rame et al., 2023): many models are trained, each corresponding to a distinct preference weight w, and for a given user, one selects the model whose weight is closest to the user's preference. To adequately cover the space of preference weights, however, the number of models must be exponential in the number of objectives, making this approach computationally prohibitive. In our framework, the number of aligned models is exactly the same as the number of reward functions, which is much smaller than the ones in these multi-objective RL schemes.

MGDA-based approaches (Désidéri, 2012) can also be directly applied to SFT (Chen et al., 2023; Xiao et al., 2023; Zhang et al., 2025) and RL fine-tuning (Wang et al., 2025). At each update, a conflict-avoiding descent direction that improves all objectives simultaneously is calculated. However, computing this direction introduces nontrivial overhead, and these methods treat diffusion models as general machine learning models, without exploiting the structure of the reverse diffusion process. In contrast, our method only requires single-objective alignment for each base reward, introduces no extra computations such as conflict-avoidant direction solving, and needs no training when adapting to new preference weights at denoising time.

LLM decoding-time alignment: There have been many recent studies in decoding-time alignment for LLMs (Shi et al., 2024; Zeng et al., 2024; Liu et al., 2024), where multiple alignment policies are fused at inference time to satisfy multi-objective preferences. These methods, however, are designed for auto-regressive token generation and do not directly transfer to the Gaussian-structured denoising dynamics of diffusion models.

Diffusion denoising-time alignment:

Similar to LLM decoding-time alignment, recent work proposes retraining-free approaches for multi-objective diffusion alignment by modifying the denoising process:

(1) Reward gradient-based methods. For single-objective alignment, several studies (Yu et al., 2023; Song et al., 2023; Bansal et al., 2024; Ye et al., 2024) use the gradient of a reward function to guide the denoising trajectory toward target regions of the output space. These methods extend naturally to multi-objective settings by using a weighted reward (Clark et al., 2024; Kim et al., 2025; Ye et al., 2024). However, they require differentiable reward models and additional computation to evaluate reward gradients.

(2) Reward value-based methods. To avoid differentiability requirements, other approaches (Kim et al., 2025; Li et al., 2024; Singh et al., 2025) rely on reward values only rather than gradients, adjusting the denoising process based on scalar feedback. While they remove the need for the reward gradient, both value-based and gradient-based methods still require access to the reward function and repeated estimation of reward expectations, typically via Tweedie-based estimators (Kadkhodaie & Simoncelli, 2021) or Monte Carlo sampling.

(3) Reward-free fusion methods. To eliminate dependence on reward access and additional sample generation, some works require only fusing pre-aligned models. Reward Soup (RS) (Rame et al., 2023) linearly interpolates model parameters rather than their reverse conditional distributions. Although simple, RS requires all aligned models to share the same architecture and lacks theoretical guarantees, e.g., Shi et al. (2024) shows that RS fails to generate meaningful responses for LLM tasks.

Closer to our work, recent reward-free methods such as DB-MPA (Cheng et al., 2026) and DERAD-IFF (Manujith et al., 2026) combine the reverse distributions of multiple aligned models. These methods are typically derived from fine-tuning methods whose optimal policy is intractable, and to obtain a tractable denoising-time objective, they omit several terms in the diffusion dynamics, introducing approximation errors that are difficult to quantify. By contrast, our step-level RL fine-tuning formulation resolves this intractability, allowing us to derive an exact closed-form denoising-time objective without dropping any terms. To better illustrate the novelty and advantages of our approach, we provide a detailed comparison with existing denoising-time alignment methods in Table 1.

3 BACKGROUND

3.1 DIFFUSION MODEL

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are latent variable models that can be expressed by $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T}$, where θ is the model parameter, $\mathbf{x}_0$ follows the data distribution $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are the latents with the same dimensionality as $\mathbf{x}_0$. There are two processes in the diffusion model: the forward process and the reverse process. The goal is to learn a reverse model $p_\theta(\mathbf{x}_0)$ that approximates the data distribution $q(\mathbf{x}_0)$. During each step t of the forward process, Gaussian noise is added with a variance schedule β_t:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}),$$
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}),$$

where $\mathcal{N}$ denotes Gaussian distribution and $\mathbf{I}$ indicates the identity matrix. In the reverse process, a denoising neural network is trained to remove the noise. Specifically, the reverse starts at $p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ and follows that

$$p_\theta(\mathbf{x}_{0:T}) := p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t),$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)).$$

3.2 RL FINE-TUNING FOR DIFFUSION

The pre-training of the diffusion model focuses solely on matching the data distribution $q(\mathbf{x}_0)$ and does not explicitly incorporate human preferences. To better align the diffusion model with downstream applications such as aesthetics, text–image alignment, and safety, RL-based fine-tuning is widely leveraged. Concretely, the RL method aims to maximize a given reward function: $\mathbb{E}_{\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)}[r(\mathbf{x}_0)]$. However, as noted by Fan et al. (2023), this objective leads to potential over-optimization problems. Following RL fine-tuning methods in LLM tasks (Ouyang et al., 2022), a KL regularization term is added to the RL objective to keep the aligned model close to the pre-trained one:

$$\max_{p_\theta} \mathbb{E}_{\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0)}[r(\mathbf{x}_0)] - \lambda \mathbb{D}_{\mathrm{KL}}[p_\theta(\mathbf{x}_0)||p_{\mathrm{pre}}(\mathbf{x}_0)], \tag{1}$$

where $\lambda > 0$ is a pre-defined hyper-parameter, p_{pre} denotes the pre-training model and $\mathbb{D}_{\mathrm{KL}}$ is the KL divergence. In practice, typically it is hard to track the marginal distribution of $p_\theta(\mathbf{x}_0)$ and further evaluate the KL term, and then the marginal KL term is replaced by the step-level KL (Fan et al., 2023; Uehara et al., 2024):

$$\max_{p_\theta} \mathbb{E}_{\mathbf{x}_{0:T} \sim p_\theta} \left[r(\mathbf{x}_0) - \lambda \sum_{t=1}^{T} \mathbb{D}_{\mathrm{KL}}[p_\theta^t(\cdot|\mathbf{x}_t)||p_{\mathrm{pre}}^t(\cdot|\mathbf{x}_t)] \right], \tag{2}$$

where $p^t(\mathbf{x}_{t-1}|\mathbf{x}_t)$ denotes the reverse conditional distribution. Typically, the RL fine-tuning objective is implemented as a KL-regularized Markov Decision Process (MDP). We discuss this in Section 4.1.

3.3 MULTI-OBJECTIVE REWARD MODEL

In standard RL fine-tuning, the model is typically optimized with respect to one single reward function. As a result, the performance of this aligned model may degrade significantly when evaluated by

another reward function. To address this problem, we consider the following multi-objective setting (Yang et al., 2019; Zhou et al., 2022; Shi et al., 2024), which assumes that there exists a set of reward functions $\{r_i\}_{i=1}^M$, representing M distinct objectives. A normalized vector $w \in \Delta^{M-1}$ is used to represent the human preference among different objectives, where Δ^{M-1} denotes the probability simplex. For a user with preference vector w, the goal is to maximize the RL fine-tuning objective under the weighted reward function $r^w = \sum_{i=1}^M w_i r_i$. Although fine-tuning with this weighted reward can yield strong performance, it is impractical to retrain a separate model for each possible preference w.

To overcome this limitation, denoising-time alignment is studied, where the core idea is to maximize the weighted reward exclusively through denoising by combining the reverse conditional distributions of a set of existing single-objective aligned diffusion models. For any preference vector w, denoising-time alignment requires no additional training: it directly runs the diffusion reverse process using the combined reverse conditionals.

4 Main Results

4.1 Step-level RL fine-tuning

As noted in Section 3.2, $p_\theta(\mathbf{x}_0)$ is usually intractable to evaluate for diffusion models, and therefore in equation 2, the intractable marginal is replaced with step-level KL terms to keep each aligned reverse conditional near the pretrained model. However, optimizing either equation 1 or equation 2 requires drawing data from the updated policy π_θ; consequently, the optimal policy depends on advantages computed under that same (unknown) policy, which makes the target policy hard to track. Therefore, existing denoising-time methods based on equation 1 and equation 2 rely on approximations whose errors are difficult to quantify (Manujith et al., 2026; Cheng et al., 2026). Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) resolves this circular dependence between the policy and advantage estimates by optimizing a surrogate objective that can be evaluated using data from a fixed reference policy. This surrogate admits provable trust-region improvement and yields an optimal update that depends only on advantages under the reference policy, which can be estimated and tracked reliably. Inspired by TRPO (Schulman et al., 2015; Zeng et al., 2024), we therefore reformulate the problem as a step-level RL objective, applying the decomposition to both the reward (via step-wise advantages) and the KL terms.

Recall the T-horizon MDP formulation for single-objective diffusion models (Black et al., 2024; Fan et al., 2023). Let $s_t = \mathbf{x}_{T-t}$ denote the state and $a_t = \mathbf{x}_{T-t-1}$ the action. The initial state distribution is $P_0(s_0) = p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The dynamics are deterministic: $P(s_{t+1}|s_t, a_t) = \delta_{a_t}$, where δ_z denotes the Dirac distribution at z. The reward function is non-zero only at the final step:

$$R(s_t, a_t) = \begin{cases} r(s_T), & t = T-1, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, the policy can be expressed as $\pi_\theta(a_t|s_t) = p_\theta(\mathbf{x}_{T-t-1}|\mathbf{x}_{T-t})$. Define the following value functions:

$$\begin{aligned} Q_\pi(s_t, a_t) &= \mathbb{E}_\pi\big[r(s_T) \mid s_t, a_t\big], \\ V_\pi(s_t) &= \mathbb{E}_\pi\big[Q_\pi(s_t, a_t) \mid s_t\big], \\ A_\pi(s_t, a_t) &= Q_\pi(s_t, a_t) - V_\pi(s_t), \end{aligned}$$

where Q_π is the state–action value function under policy π, and V_π and A_π are the corresponding value and advantage functions.

We then introduce our step-level RL objective. Motivated by the TRPO approach (Schulman et al., 2015), we apply the decomposition to both the reward (via step-wise advantages) and the KL terms:

$$\max_{\pi_\theta} \quad \mathbb{E}_{s_t \sim \mathcal{D}, z \sim \pi_\theta(\cdot|s_t)}\big[A_{\pi_{\text{pre}}}(s_t, z) - \lambda \mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_{\text{pre}}(\cdot \mid s_t)\big)\big], \tag{3}$$

where $\mathcal{D}$ is the dataset and π_{pre} is the policy induced by the pre-trained model. Compared with equation 2, equation 3 replaces the terminal-state reward with the advantage function under the pre-trained policy π_{pre}. Furthermore, Lemma 4.1 in Zeng et al. (2024) shows that maximizing equation 3 improves the policy in expectation.

Wallace et al. (2024) studies an RL fine-tuning variant that replaces the marginal KL regularizer $\mathbb{D}_{\mathrm{KL}}[p_\theta(\mathbf{x}_0)||p_{\mathrm{pre}}(\mathbf{x}_0)]$ in equation 1 with the trajectory-level KL regularizer $\mathbb{D}_{\mathrm{KL}}[p_\theta(\mathbf{x}_{0:T})||p_{\mathrm{pre}}(\mathbf{x}_{0:T})]$. Based on this formulation, a DPO-style method for diffusion models has been proposed, which directly aligns the model to human preference data. For our step-level RL formulation, in the following lemma, we also provide a corresponding DPO objective. Let $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{i=1}^{t} \alpha_i$. Let $\mathbf{x}_0^w$ and $\mathbf{x}_0^l$ denote the winning and losing samples, respectively. For $* \in \{w, l\}$, define the forward diffusion process

$$\mathbf{x}_t^* = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0^* + \sqrt{1-\bar{\alpha}_t}\,\epsilon^*, \qquad \epsilon^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

so that $\mathbf{x}_t^* \sim q(\mathbf{x}_t^* \mid \mathbf{x}_0^*)$. The corresponding signal-to-noise ratio is $\lambda_t := \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}$.

Lemma 1. *The step-level DPO objective derived from equation 3 can be written as*

$$-\mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D},\, t\sim\mathcal{U}(1,T),\, \mathbf{x}_t^w\sim q(\mathbf{x}_t^w|\mathbf{x}_0^w),\, \mathbf{x}_t^l\sim q(\mathbf{x}_t^l|\mathbf{x}_0^l)} \left[\log\sigma\left(-\lambda T\,\omega(\lambda_t)\left(\Delta_\theta^w - \Delta_\theta^l - \Delta_{\mathrm{diff}}\right)\right)\right], \tag{4}$$

where $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the sigmoid function, $\mathcal{U}(1,T)$ denotes a uniform distribution over $\{1, \cdots, T\}$ and

$$\Delta_\theta^* := \|\epsilon^* - \epsilon_\theta(\mathbf{x}_t^*, t)\|_2^2 - \|\epsilon^* - \epsilon_{\mathrm{pre}}(\mathbf{x}_t^*, t)\|_2^2,$$
$$\Delta_{\mathrm{diff}} := \|\epsilon_\theta(\mathbf{x}_t^w, t) - \epsilon_{\mathrm{pre}}(\mathbf{x}_t^w, t)\|_2^2 - \|\epsilon_\theta(\mathbf{x}_t^l, t) - \epsilon_{\mathrm{pre}}(\mathbf{x}_t^l, t)\|_2^2,$$

$\omega(\lambda_t)$ denotes a weighting function (often chosen to be constant in practice (Ho et al., 2020; Kingma et al., 2021)), and $\epsilon_*(\mathbf{x}_t, t)$ is a function approximator predicting the noise term such that the corresponding reverse-process mean is given by

$$\mu_*(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_*(\mathbf{x}_t, t)\right).$$

The full proof is shown in Appendix A.

Remark 1. *Unlike the diffusion DPO objective that is derived from a variant of equation 1 with a joint KL regularizer (Wallace et al., 2024), our step-level DPO loss contains an extra step-wise regularization term Δ_{diff} inside the $\log\sigma(\cdot)$. This term explicitly encourages the aligned model to remain closer to the pretrained model on less-preferred trajectories than on preferred ones at each denoising step. Intuitively, when the training signal is positive (preferred trajectory), the model is allowed to move away from the pretrained model, whereas when the signal is negative (less-preferred trajectory), the penalty keeps the aligned model close to the pretrained weights, making it safe to leave those behaviors unchanged.*

4.2 Multi-objective alignment

In Section 4.1, we developed a step-level RL fine-tuning framework for a single reward function. We now extend this framework to the multi-objective setting and design a retraining-free method that efficiently aligns diffusion models with diverse human preferences, without introducing any approximation error.

In the multi-objective setting, we assume a collection of reward functions $\{r_i\}_{i=1}^M$ corresponding to M distinct objectives. For each objective i, we assume access to the policy π_i and the corresponding reverse distribution p_i that maximizes the single-objective step-level RL objective in equation 3 with reward r_i. We then show that, for any weighted reward $r^w = \sum_{i=1}^M w_i r_i$ with preference weights $w \in \Delta^{M-1}$, we can construct the optimal reverse distribution p_w purely by combining the distributions $\{p_i\}_{i=1}^M$, without any additional training. The corresponding denoising-time sampling procedure is summarized in Algorithm 1.

Before presenting our main result, we state a lemma that characterizes the optimal policy for step-level RL objectives.

Lemma 2 (Lemma 4.2, (Zeng et al., 2024))**.** *For any reward function r, the optimal policy for the step-level RL objective in equation 3 has the closed-form expression*

$$\pi_\theta^\star(z \mid s_t) = \frac{\pi_{\mathrm{pre}}(z \mid s_t)\exp\big(Q_{\pi_{\mathrm{pre}}}(s_t, z)/\lambda\big)}{Z(s_t)}, \tag{5}$$

Algorithm 1 Multi-objective Step-level Denoising-time Diffusion Alignment (MSDDA)

Require: Aligned models $\{p_i(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\}_{i=1}^M$, preference weights $w \in \Delta^{M-1}$, inference steps T

Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $k = T, \ldots, 1$ **do**
  $t \leftarrow t_k$, $t_{\text{prev}} \leftarrow t_{k-1}$
  for $i = 1$ **to** M **do**
    $(\mu_i, \sigma_i^2) \leftarrow$ parameters of $p_i(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mu_i, \sigma_i^2 \mathbf{I})$
  end for
  $\sigma_{\text{new}}^{-2} \leftarrow \sum_{i=1}^M \frac{w_i}{\sigma_i^2}$, $\sigma_{\text{new}}^2 \leftarrow 1/\sigma_{\text{new}}^{-2}$, $\mu_{\text{new}} \leftarrow \sigma_{\text{new}}^2 \sum_{i=1}^M \frac{w_i}{\sigma_i^2} \mu_i$
  Sample $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_{t_{\text{prev}}} \leftarrow \mu_{\text{new}} + \sigma_{\text{new}} z$
end for
return $\mathbf{x}_0$

where $Z(s_t) = \int \pi_{\text{pre}}(z \mid s_t) \exp\big(Q_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)\, dz$ is the normalizing constant.

This lemma is from Lemma 4.2 of Zeng et al. (2024). For completeness, we include the proof in Appendix B.

A key advantage of our step-level RL formulation is that, once we have obtained the aligned models for each individual objective, we can derive closed-form solutions for the optimal policy corresponding to any weighted reward $r^w = \sum_{i=1}^M w_i r_i$.

Theorem 1. *Let π_i be the optimal policy of equation 3 with reward r_i, and p_i is the corresponding reverse distribution. Assume that, at each step t, $p_i(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is Gaussian with mean μ_i and variance σ_i^2 so that $p_i(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mu_i, \sigma_i^2 \mathbf{I})$. Then, for any $w \in \Delta^{M-1}$ and $r^w = \sum_{i=1}^M w_i r_i$, the step-level optimal posterior is*

$$p_w(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \frac{\prod_{i=1}^M p_i^{w_i}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{\displaystyle\int \prod_{i=1}^M p_i^{w_i}(\mathbf{x}'_{t-1} \mid \mathbf{x}_t)\, d\mathbf{x}'_{t-1}}, \tag{6}$$

which is Gaussian with closed-form parameters:

$$\sigma_w^2 = \left(\sum_{i=1}^M \frac{w_i}{\sigma_i^2}\right)^{-1}, \mu_w = \sigma_w^2 \sum_{i=1}^M \frac{w_i}{\sigma_i^2} \mu_i. \tag{7}$$

We provide a proof sketch below; the full proof is deferred to Appendix C.

Proof sketch. By Lemma 2, the optimal policy π_i for reward r_i satisfies

$$\pi_i \propto \exp\big(Q^i_{\pi_{\text{pre}}}(s_t, z)/\lambda\big), \tag{8}$$

where $Q^i_{\pi_{\text{pre}}}$ is the state–action value function under the reference policy and reward r_i. Since the combined reward is $r^w = \sum_{i=1}^M w_i r_i$, we have

$$Q^w_{\pi_{\text{pre}}}(s_t, z) = \sum_{i=1}^M w_i Q^i_{\pi_{\text{pre}}}(s_t, z). \tag{9}$$

Applying Lemma 2 with reward r^w and using Eqs. equation 8, equation 9 and the definition that $\pi_i(a_t|s_t) = p_i(\mathbf{x}_{T-t-1}|\mathbf{x}_{T-t})$ for $s_t = \mathbf{x}_{T-t}$ and $a_t = \mathbf{x}_{T-t-1}$, we obtain that

$$p_w \propto \exp\big(Q^w_{\pi_{\text{pre}}}(s_t, z)/\lambda\big) \propto \prod_{i=1}^M \exp\big(Q^i_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)^{w_i} \propto \prod_{i=1}^M p_i^{w_i}. \tag{10}$$

Since each p_i is Gaussian, their weighted product is also Gaussian, and we can calculate the mean and variance, which are in equation 7. □

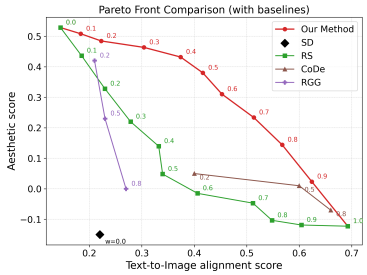


Figure 2: Pareto front of our proposed method and the baselines with different preference weight w. The results for CoDe and RGG are obtained from Cheng et al. (2026).



Figure 3: Image samples generated by our proposed method and RS under different preference weights w. The left image is generated by Stable Diffusion using the prompt shown in the lower-left corner. The top row shows images generated by our proposed MSDDA, while the bottom row shows images generated by RS.

Theorem 1 provides closed-form expressions for the mean and variance of the optimal policy for every preference vector $w \in \Delta^{M-1}$. Thus, our denoising-time alignment procedure requires no retraining: we only need to fuse the single-objective posteriors $\{p_i\}_{i=1}^{M}$.

Table 2: Quantitative comparison of our proposed MSDDA method and baseline methods. The results for CoDe and RGG are obtained from Cheng et al. (2026).

		SD		MSDDA		RS		CoDe		RGG	
		r_1	r_2	r_1	r_2	r_1	r_2	r_1	r_2	r_1	r_2
Reward (↑)	$w = 0.2$	0.22	-0.15	0.22	0.48	0.23	0.33	0.40	0.05	0.21	0.42
	$w = 0.5$	0.22	-0.15	0.42	0.38	0.34	0.05	0.60	0.01	0.23	0.23
	$w = 0.8$	0.22	-0.15	0.57	0.14	0.55	-0.10	0.66	-0.07	0.27	0.00
Inference Time (↓ sec/img)		5.08		10.14		5.08		185.26		121.58	

Finally, we highlight the novelty of our method compared with existing denoising-time alignment methods such as DB-MPA (Cheng et al., 2026) and DERADIFF (Manujith et al., 2026). These methods are derived from equation 1, whose optimal policy is hard to track due to the circular dependence between the policy and advantage estimates. To obtain a denoising-time objective, Cheng et al. (2026) and Manujith et al. (2026) discard several terms in the diffusion dynamics, resulting in approximation errors that are hard to quantify. In contrast, our step-level formulation resolves this intractability problem, which allows us to derive an exact closed-form denoising objective without any approximations. In Appendix D, we provide detailed discussions about how the approximated errors are introduced in DB-MPA and DERADIFF. From an algorithmic perspective, our sampler coincides with the procedure used in DERADIFF (Manujith et al., 2026), but our analysis provides explicit theoretical guarantees. Moreover, DB-MPA (Cheng et al., 2026) can be viewed as a special case of our method that **requires all objectives to share the same variance** σ_i^2. Because of this assumption, DB-MPA is not applicable when objectives have different variances. By allowing arbitrary σ_i^2, our framework strictly generalizes DB-MPA and remains valid in the heterogeneous-variance setting.

5 Experiments

We then present numerical results that validate the effectiveness of the proposed MSDDA algorithm, where $\lambda = 0.1$ and other experiment details can be found in Appendix E.

Prompt dataset: We adopt the same prompt dataset as in Cheng et al. (2026), which is a subset of DrawBench (Saharia et al., 2022) restricted to the "color" category. To obtain test prompts that do not appear in the training data, GPT-4 is used to synthesize novel color–object and object–object

combinations derived from the training set. To assess generalization beyond the "color" prompts, “color attribution” and “counting” categories of GenEval (Kirstain et al., 2023) are aslo used.

Diffusion and reward models: Our experiments use Stable Diffusion v1.5 (Rombach et al., 2022) as the pre-trained generative model. Generated images are evaluated using two reward models: (1) ImageReward (Xu et al., 2023) for measuring text–image alignment (higher is better), and (2) VILA (Ke et al., 2023) for assessing aesthetic quality. Following Cheng et al. (2026), we rescale VILA scores with the linear mapping $r \mapsto 4r - 2$ so they share a similar range as ImageReward. For each test prompt, we sample 32 images using seeds $0 - 31$, compute each reward for every sample, and report the average score.

Baselines: We compare MSDDA against several baselines: (1) Stable Diffusion (SD) v1.5 (Rombach et al., 2022), which serves as a pretrained baseline; (2) Reward Gradient-based Guidance (RGG) (Kim et al., 2025), which leverages reward values to guide denoising; (3) CoDe (Singh et al., 2025), a method that uses the gradient of the reward to steer sampling; and (4) Reward Soup (RS) (Rame et al., 2023). In our reward-free setup, we have two aligned Stable Diffusion models, each optimized with a different reward and with distinct denoising variance schedules; because RS combines model parameters linearly and the two models share the same architecture, it can be applied in this setting. Another reward-free method, DB-MPA (Cheng et al., 2026), requires identical denoising time variances across models and is therefore incompatible with our experimental setup. To address this limitation, we introduce a simple extension, termed DB-Extended, which constructs a new Gaussian distribution by taking a weighted linear combination of the means and variances, where the details can be found in Appendix E.

In Fig. 2 we plot the Pareto front for our method and the baselines across different preference weights w. The horizontal axis shows the ImageReward score, and the vertical axis shows the VILA score. For a given weight w, the scalarized target reward is $r^w = wr_1 + (1 - w)r_2$, where r_1 denotes ImageReward and r_2 denotes VILA. Table 2 lists the numeric reward values for each w and also reports the generation time required per image. From the numerical results, we can observe that

- Performance. MSDDA consistently outperforms the baselines. For most preference weights w, our method achieves the highest scores on both reward models. The pretrained SD model struggles on several test prompts (e.g., “A purple colored dog” or “A blue colored apple”), which are uncommon in real-world data; all baselines improve over the SD baseline, demonstrating their ability to enhance denoising and alignment.
- Latency. SD and RS are the fastest methods because they invoke only a single diffusion model. Our method has the second-shortest runtime, roughly twice the denoising time of SD/RS—consistent with MSDDA’s requirement to run two diffusion models. RGG and CoDe incur larger runtimes because they evaluate reward functions during denoising.

In Fig. 3, we show representative images produced by our MSDDA and by RS under different preference weights w to directly show the model performance. The base SD model often fails to follow the prompt (for example, producing a pink banana, a purple backpack with a red apple, or a blue apple). Both MSDDA and RS use the same aligned diffusion backbones, but MSDDA achieves noticeably better alignment with the prompts. For instance, at $w \in [0.2, 0.5, 0.8]$, MSDDA reliably generates the pink banana and the purple backpack with a red apple, while RS often does not. Additional examples and experiments with more objectives appear in Appendix E.

6 Conclusion

This paper addresses the multi-objective denoising-time alignment problem. To solve the intractability of the optimal policy in existing RL fine-tuning formulations, we model the task as a step-level RL problem and derive a corresponding DPO objective that encourages the aligned model to remain closer to the pretrained model on less-preferred trajectories than on preferred ones. Building on this formulation, we propose MSDDA, a retraining-free fusion method that leverages closed-form step-level solutions to combine existing aligned models and provably maximizes the multi-objective reward without approximation error or access to individual reward functions. Extensive numerical results are conducted to validate the effectiveness of our proposed MSDDA. As with other training-free fusion methods, MSDDA relies on the availability of well-aligned single-objective models, and extending it to settings with noisy or incomplete aligners is an important direction for future work.

AI USE STATEMENT

In this work, we used GPT-4 to generate a synthetic evaluation prompt set comprising novel color–object and object–object combinations derived from the training prompts. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, design or provide feedback on the research methodology or experiments; implement the proposed methods; assist with translation; clean or reformat datasets; support qualitative or thematic data analysis; or interpret the experimental results. Additionally, we used generative AI tools to edit portions of the manuscript for grammatical correctness, clarity, and readability. All AI-assisted texts were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 51304–51323, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Lisha Chen, Heshan Fernando, Yiming Ying, and Tianyi Chen. Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance. *Advances in Neural Information Processing Systems*, 36:70045–70093, 2023.

Min Cheng, Fatemeh Doudi, Dileep Kalathil, Mohammad Ghavamzadeh, and Panganamala Kumar. Diffusion blend: Inference-time multi-preference alignment for diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 51151–51181, 2026.

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus. Mathématique*, 350(5-6):313–318, 2012.

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. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Xu Han, Caihua Shan, Yifei Shen, Can Xu, Han Yang, Xiang Li, and Dongsheng Li. Training-free multi-objective diffusion model for 3d molecule generation. In *International Conference on Learning Representations*, volume 2024, pp. 45998–46013, 2024.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254, 2021.

Junjie Ke, Keren Ye, Jiahui Yu, Yonghui Wu, Peyman Milanfar, and Feng Yang. Vila: Learning image aesthetics from user comments with vision-language pretraining. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10041–10051. IEEE, 2023.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, volume 2025, pp. 87238–87272, 2025.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

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. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, et al. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. Decoding-time realignment of language models. *arXiv preprint arXiv:2402.02992*, 2024.

Ratnavibusena Don Shahain Manujith, Teoh Tze Tzun, Kenji Kawaguchi, and Yang Zhang. Deradiff: Denoising time realignment of diffusion models. *arXiv preprint arXiv:2601.20198*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36:71095–71134, 2023.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. Pmlr, 2015.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A Smith, and Simon S Du. Decoding-time language model alignment with multiple objectives. *Advances in Neural Information Processing Systems*, 37:48875–48920, 2024.

Anuj Singh, Sayak Mukherjee, Ahmad Beirami, and Hadi Jamali-Rad. Code: Blockwise control for denoising diffusion models. *arXiv preprint arXiv:2502.00968*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

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

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.

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. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Yudan Wang, Peiyao Xiao, Hao Ban, Kaiyi Ji, and Shaofeng Zou. Theoretical study of conflict-avoidant multi-objective reinforcement learning. *IEEE Transactions on Information Theory*, 71(9): 7254–7269, 2025.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2096–2105. IEEE, 2023.

Peiyao Xiao, Hao Ban, and Kaiyi Ji. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. *Advances in Neural Information Processing Systems*, 36:4509–4533, 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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32, 2019.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *Advances in Neural Information Processing Systems*, 37:22370–22417, 2024.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23117–23127. IEEE, 2023.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.

Qi Zhang, Peiyao Xiao, Shaofeng Zou, and Kaiyi Ji. Mgda converges under generalized smoothness, provably. In *International Conference on Learning Representations*, volume 2025, pp. 10425–10456, 2025.

Ruida Zhou, Tao Liu, Dileep Kalathil, PR Kumar, and Chao Tian. Anchor-changing regularized natural policy gradient for multi-objective reinforcement learning. *Advances in neural information processing systems*, 35:13584–13596, 2022.

A Derivation of Lemma 1

In this section, we provide the proof to derive the step-level DPO objective. For each $\mathbf{x}_{0:T}$, we have the following relationships between the reward and value functions:

$$Q_{\pi_{\text{pre}}}(s_t, a_t) = R(s_t, a_t) + V_{\pi_{\text{pre}}}(s_{t+1}),$$
$$A_{\pi_{\text{pre}}}(s_t, a_t) = Q_{\pi_{\text{pre}}}(s_t, a_t) - V_{\pi_{\text{pre}}}(s_t).$$

It then follows that

$$\begin{aligned} r(\mathbf{x}_0) &= \sum_{t=0}^{T-1} R(s_t, a_t) \\ &= \sum_{t=0}^{T-1} Q_{\pi_{\text{pre}}}(s_t, a_t) - V_{\pi_{\text{pre}}}(s_{t+1}) \\ &= V_{\pi_{\text{pre}}}(s_0) + \sum_{t=0}^{T-1} \big(Q_{\pi_{\text{pre}}}(s_t, a_t) - V_{\pi_{\text{pre}}}(s_t)\big) - V_{\pi_{\text{pre}}}(s_T) \\ &= V_{\pi_{\text{pre}}}(s_0) + \sum_{t=0}^{T-1} A_{\pi_{\text{pre}}}(s_t, a_t), \end{aligned} \tag{11}$$

where the first equality is due to the definition of reward function R and the last equality is due to $V_{\pi_{\text{pre}}}(s_T) = 0$. According to the BT model, we have that

$$p_{BT}(\mathbf{x}_0^w \succ \mathbf{x}_0^l) = \sigma(r(\mathbf{x}_0^w) - r(\mathbf{x}_0^l)).$$

Based on the BT model and equation 11, we can write the loss as

$$\begin{aligned} &-\mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}} \log \sigma\left(r(\mathbf{x}_0^w) - r(\mathbf{x}_0^l)\right) \\ &= -\mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}} \log \sigma\left(\mathbb{E}_{\mathbf{x}_{0:T}^w \sim q(\mathbf{x}_{0:T}^w|\mathbf{x}_0^w), \mathbf{x}_{0:T}^l \sim q(\mathbf{x}_{0:T}^l|\mathbf{x}_0^l)}[r(\mathbf{x}_0^w) - r(\mathbf{x}_0^l)]\right) \\ &= -\mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}} \log \sigma\Bigg(\mathbb{E}_{\mathbf{x}_{0:T}^w \sim q(\mathbf{x}_{0:T}^w|\mathbf{x}_0^w), \mathbf{x}_{0:T}^l \sim q(\mathbf{x}_{0:T}^l|\mathbf{x}_0^l)}\Bigg[V_{\pi_{\text{pre}}}(s_0^w) + \sum_{t=0}^{T-1} A_{\pi_{\text{pre}}}(s_t^w, a_t^w) \\ &\quad - V_{\pi_{\text{pre}}}(s_0^l) - \sum_{t=0}^{T-1} A_{\pi_{\text{pre}}}(s_t^l, a_t^l)\Bigg]\Bigg). \end{aligned} \tag{12}$$

Similar to Wallace et al. (2024), we remove the $V_{\pi_{\text{pre}}}(s_0^w), V_{\pi_{\text{pre}}}(s_0^l)$ terms since they are constant and obtain the following objective

$$\begin{aligned} &-\mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}} \log \sigma\left(\mathbb{E}_{\mathbf{x}_{0:T}^w \sim q(\mathbf{x}_{0:T}^w|\mathbf{x}_0^w), \mathbf{x}_{0:T}^l \sim q(\mathbf{x}_{0:T}^l|\mathbf{x}_0^l)}\left[\sum_{t=0}^{T-1} A_{\pi_{\text{pre}}}(s_t^w, a_t^w) - \sum_{t=0}^{T-1} A_{\pi_{\text{pre}}}(s_t^l, a_t^l)\right]\right) \\ &= -\mathbb{E}_{(\mathbf{x}_0^w, \mathbf{x}_0^l)\sim\mathcal{D}} \log \sigma\Bigg(T \\ &\quad \times \mathbb{E}_{t\sim\mathcal{U}(0,T-1), \mathbf{x}_{0:T}^w \sim q(\mathbf{x}_{0:T}^w|\mathbf{x}_0^w), \mathbf{x}_{0:T}^l \sim q(\mathbf{x}_{0:T}^l|\mathbf{x}_0^l)}\left[A_{\pi_{\text{pre}}}(s_t^w, a_t^w) - A_{\pi_{\text{pre}}}(s_t^l, a_t^l)\right]\Bigg). \end{aligned} \tag{13}$$

Moreover, based on the definition of value functions and Lemma 2, we can show that

$$\begin{aligned} A_{\pi_{\text{pre}}}(s_t^w, a_t^w) &= Q_{\pi_{\text{pre}}}(s_t^w, a_t^w) - V_{\pi_{\text{pre}}}(s_t^w) \\ &= Q_{\pi_{\text{pre}}}(s_t^w, a_t^w) - \mathbb{E}_{a\sim\pi_{\text{pre}}}[Q_{\pi_{\text{pre}}}(s_t^w, a)] \\ &= \lambda \log\left(\frac{p_\theta(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}{p_{\text{pre}}(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}\right) + \lambda \log(Z(\mathbf{x}_{T-t}^w)) \\ &\quad - \mathbb{E}_{\mathbf{x}_{T-t-1}^w \sim \pi_{\text{pre}}}\left[\lambda \log\left(\frac{p_\theta(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}{p_{\text{pre}}(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}\right)\right] - \lambda \log(Z(\mathbf{x}_{T-t}^w)) \\ &= \lambda \log\left(\frac{p_\theta(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}{p_{\text{pre}}(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}\right) + \lambda \mathbb{D}_{\text{KL}}(p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^w)||p_\theta(\cdot|\mathbf{x}_{T-t}^w)). \end{aligned} \tag{14}$$

By Jensen's inequality, the objective in equation 13 can be upper bounded by

$$
\begin{aligned}
&-\mathbb{E}_{(\mathbf{x}_0^w,\mathbf{x}_0^l)\sim\mathcal{D}}\log\sigma\Bigg(T \\
&\quad\times\mathbb{E}_{t\sim\mathcal{U}(0,T-1),\mathbf{x}_{0:T}^w\sim q(\mathbf{x}_{0:T}^w|\mathbf{x}_0^w),\mathbf{x}_{0:T}^l\sim q(\mathbf{x}_{0:T}^l|\mathbf{x}_0^l)}\left[A_{\pi_{\text{pre}}}(s_t^w,a_t^w)-A_{\pi_{\text{pre}}}(s_t^l,a_t^l)\right]\Bigg) \\
&\leq-\mathbb{E}_{t\sim\mathcal{U}(0,T-1),(\mathbf{x}_0^w,\mathbf{x}_0^l)\sim\mathcal{D},\mathbf{x}_{T-t}^w\sim q(\mathbf{x}_{T-t}^w|\mathbf{x}_0^w),\mathbf{x}_{T-t}^l\sim q(\mathbf{x}_{T-t}^l|\mathbf{x}_0^l)}\log\sigma\Bigg(T \\
&\quad\times\mathbb{E}_{\mathbf{x}_{T-t-1}^w\sim q(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w),\mathbf{x}_{T-t-1}^l\sim q(\mathbf{x}_{T-t-1}^l|\mathbf{x}_{T-t}^l,\mathbf{x}_0^l)}\left[A_{\pi_{\text{pre}}}(s_t^w,a_t^w)-A_{\pi_{\text{pre}}}(s_t^l,a_t^l)\right]\Bigg).
\end{aligned}
\tag{15}
$$

It can be further shown that

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{x}_{T-t-1}^w\sim q(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w),\mathbf{x}_{T-t-1}^l\sim q(\mathbf{x}_{T-t-1}^l|\mathbf{x}_{T-t}^l,\mathbf{x}_0^l)}\left[A_{\pi_{\text{pre}}}(s_t^w,a_t^w)\right] \\
&=\lambda\mathbb{D}_{\text{KL}}(p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^w)||p_\theta(\cdot|\mathbf{x}_{T-t}^w)) \\
&\quad+\lambda\mathbb{E}_{\mathbf{x}_{T-t-1}^w\sim q(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w),\mathbf{x}_{T-t-1}^l\sim q(\mathbf{x}_{T-t-1}^l|\mathbf{x}_{T-t}^l,\mathbf{x}_0^l)}\left[\log\left(\frac{p_\theta(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}{p_{\text{pre}}(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w)}\right)\right] \\
&=\lambda\mathbb{D}_{\text{KL}}(p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^w)||p_\theta(\cdot|\mathbf{x}_{T-t}^w)) \\
&\quad-\lambda\mathbb{D}_{\text{KL}}(q(\cdot|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w)||\pi_\theta(\cdot|\mathbf{x}_{T-t}^w)) \\
&\quad+\lambda\mathbb{D}_{\text{KL}}(q(\cdot|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w)||p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^w)).
\end{aligned}
\tag{16}
$$

Applying the same analysis to $A_{\pi_{\text{pre}}}(s_t^l,a_t^l)$ and plugging the results to equation 15, we then can get the following objective:

$$
\begin{aligned}
&-\mathbb{E}_{t\sim\mathcal{U}(0,T-1),(\mathbf{x}_0^w,\mathbf{x}_0^l)\sim\mathcal{D},\mathbf{x}_{T-t}^w\sim q(\mathbf{x}_{T-t}^w|\mathbf{x}_0^w),\mathbf{x}_{T-t}^l\sim q(\mathbf{x}_{T-t}^l|\mathbf{x}_0^l)} \\
&\quad\log\sigma\left(T\mathbb{E}_{\mathbf{x}_{T-t-1}^w\sim q(\mathbf{x}_{T-t-1}^w|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w),\mathbf{x}_{T-t-1}^l\sim q(\mathbf{x}_{T-t-1}^l|\mathbf{x}_{T-t}^l,\mathbf{x}_0^l)}\left[A_{\pi_{\text{pre}}}(s_t^w,a_t^w)-A_{\pi_{\text{pre}}}(s_t^l,a_t^l)\right]\right) \\
&=\mathbb{E}_{t\sim\mathcal{U}(0,T-1),(\mathbf{x}_0^w,\mathbf{x}_0^l)\sim\mathcal{D},\mathbf{x}_{T-t}^w\sim q(\mathbf{x}_{T-t}^w|\mathbf{x}_0^w),\mathbf{x}_{T-t}^l\sim q(\mathbf{x}_{T-t}^l|\mathbf{x}_0^l)} \\
&\quad\log\sigma\Bigg(T\lambda\Bigg(-\mathbb{D}_{\text{KL}}(q(\cdot|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w)||\pi_\theta(\cdot|\mathbf{x}_{T-t}^w))+\mathbb{D}_{\text{KL}}(q(\cdot|\mathbf{x}_{T-t}^l,\mathbf{x}_0^l)||\pi_\theta(\cdot|\mathbf{x}_{T-t}^l)) \\
&\quad+\mathbb{D}_{\text{KL}}(q(\cdot|\mathbf{x}_{T-t}^w,\mathbf{x}_0^w)||p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^w))-\mathbb{D}_{\text{KL}}(q(\cdot|\mathbf{x}_{T-t}^l,\mathbf{x}_0^l)||p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^l)) \\
&\quad+\mathbb{D}_{\text{KL}}(p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^w)||p_\theta(\cdot|\mathbf{x}_{T-t}^w))-\mathbb{D}_{\text{KL}}(p_{\text{pre}}(\cdot|\mathbf{x}_{T-t}^l)||p_\theta(\cdot|\mathbf{x}_{T-t}^l))\Bigg)\Bigg)
\end{aligned}
\tag{17}
$$

By the Gaussian parameterization of the reverse process (Wallace et al., 2024), this objective can be simplified to

$$
\begin{aligned}
L_{SDPO}=&-\mathbb{E}_{(\mathbf{x}_0^w,\mathbf{x}_0^l)\sim\mathcal{D},\,t\sim\mathcal{U}(1,T),\,\mathbf{x}_t^w\sim q(\mathbf{x}_t^w|\mathbf{x}_0^w),\,\mathbf{x}_t^l\sim q(\mathbf{x}_t^l|\mathbf{x}_0^l)} \\
&\Big[\log\sigma\Big(-\lambda T\omega(\lambda_t)\Big(\|\epsilon^w-\epsilon_\theta(\mathbf{x}_t^w,t)\|_2^2-\|\epsilon^w-\epsilon_{\text{pre}}(\mathbf{x}_t^w,t)\|_2^2 \\
&-\big(\|\epsilon^l-\epsilon_\theta(\mathbf{x}_t^l,t)\|_2^2-\|\epsilon^l-\epsilon_{\text{pre}}(\mathbf{x}_t^l,t)\|_2^2\big) \\
&-\big(\|\epsilon_\theta(\mathbf{x}_t^w,t)-\epsilon_{\text{pre}}(\mathbf{x}_t^w,t)\|_2^2-\|\epsilon_\theta(\mathbf{x}_t^l,t)-\epsilon_{\text{pre}}(\mathbf{x}_t^l,t)\|_2^2\big)\Big)\Big)\Big],
\end{aligned}
\tag{18}
$$

where $\mathbf{x}_0^w$ and $\mathbf{x}_0^l$ denote the winning and losing samples, respectively and we have that

$$
\mathbf{x}_t^*=\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0^*+\sqrt{1-\bar{\alpha}_t}\,\epsilon^*,\qquad\epsilon^*\sim\mathcal{N}(\mathbf{0},\mathbf{I}),
$$

which is drawn from $q(\mathbf{x}_t^* \mid \mathbf{x}_0^*)$, and the quantity $\lambda_t = \bar{\alpha}_t/(1-\bar{\alpha}_t)$ is the signal-to-noise ratio. The function $\omega(\lambda_t)$ is a weighting function (often taken to be constant in practice (Ho et al., 2020; Kingma et al., 2021)). $\epsilon_*(\mathbf{x}_t, t)$ is a function approximator that predicts ϵ from $\mathbf{x}_t$ such that

$$\mu_*(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_*(\mathbf{x}_t, t)\right). \tag{19}$$

B Proof of Lemma 2

Proof. We can first define that

$$\begin{aligned}\mathcal{J}(\pi_\theta; s_t) &= \mathbb{E}_{z\sim\pi_\theta(\cdot|s_t)}\left[Q_{\pi_{\text{pre}}}(s_t, z) - V_{\pi_{\text{pre}}}(s_t) - \lambda \log \frac{\pi_\theta(z \mid s_t)}{\pi_{\text{pre}}(z \mid s_t)}\right] \\ &= \lambda \mathbb{E}_{z\sim\pi_\theta(\cdot|s_t)}\left[\log \frac{\pi_{\text{pre}}(z \mid s_t)\exp\big(Q_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)}{\pi_\theta(z \mid s_t)}\right] - V_{\pi_{\text{pre}}}(s_t),\end{aligned}$$

where we have that $Z(s_t) = \int \pi_{\text{ref}}(z \mid s_t)\exp\big(Q_{\pi_{\text{ref}}}(s_t, z)/\lambda\big)\,dz$. The optimal solution of equation 3 can be obtained by optimizing $\mathcal{J}(\pi_\theta; s_t)$ for each s_t. We can further rewrite $\mathcal{J}(\pi_\theta; s_t)$ as:

$$\mathcal{J}(\pi_\theta; s_t) = -\lambda \mathbb{D}_{\text{KL}}\left(\pi_\theta(\cdot \mid s_t) \,\middle\|\, \frac{\pi_{\text{pre}}(\cdot \mid s_t)\exp\big(Q_{\pi_{\text{pre}}}(s_t, \cdot)/\lambda\big)}{Z(s_t)}\right) - V_{\pi_{\text{pre}}}(s_t) + \lambda \log Z(s_t). \tag{20}$$

Thus, we can find the optimal solution:

$$\pi_\theta^\star(z \mid s_t) = \frac{\pi_{\text{ref}}(z \mid s_t)\exp\big(Q_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)}{Z(s_t)}. \tag{21}$$

This completes the proof. □

C Proof of Theorem 1

Proof. Based on Lemma 2, we can show that for each $1 \le i \le M$, we have that

$$\pi_i(z \mid s_t) = \frac{\pi_{\text{pre}}(z \mid s_t)\exp\big(Q^i_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)}{Z_i(s_t)},$$

where $Q^i_{\pi_{\text{pre}}}$ is the state-action value function with reward r_i and $Z_i(s_t) = \int \pi_{\text{ref}}(z \mid s_t)\exp\big(Q^i_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)\,dz$. Moreover, we can further show that

$$\pi_w(z \mid s_t) = \frac{\pi_{\text{pre}}(z \mid s_t)\exp\big(Q^w_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)}{Z_w(s_t)}, \tag{22}$$

where $Q^w_{\pi_{\text{pre}}}$ is the state-action value function with reward r_w and $Z_w(s_t) = \int \pi_{\text{ref}}(z \mid s_t)\exp\big(Q^w_{\pi_{\text{pre}}}(s_t, z)/\lambda\big)\,dz$. For the state-action function, we can show that

$$\begin{aligned}Q^w_{\pi_{\text{pre}}}(s_t, z) &= \mathbb{E}_{\pi_{\text{pre}}}\big[r^w(s_T) \mid s_t, a_t\big] \\ &= \mathbb{E}_{\pi_{\text{pre}}}\left[\sum_{i=1}^M w_i r_i(s_T) \mid s_t, a_t\right] \\ &= \sum_{i=1}^M w_i \mathbb{E}_{\pi_{\text{pre}}}\big[r_i(s_T) \mid s_t, a_t\big] \\ &= \sum_{i=1}^M w_i Q^i_{\pi_{\text{pre}}}(s_t, z).\end{aligned} \tag{23}$$

As a result, we can show that

$$\begin{aligned}\exp\big(Q^w_{\pi_{\text{pre}}}(s_t,z)/\lambda\big) &= \exp\left(\sum_{i=1}^{M} w_i Q^i_{\pi_{\text{pre}}}(s_t,z)/\lambda\right)\\ &= \prod_{i=1}^{M}\exp\big(w_i Q^i_{\pi_{\text{pre}}}(s_t,z)/\lambda\big)\\ &= \prod_{i=1}^{M}\exp\big(Q^i_{\pi_{\text{pre}}}(s_t,z)/\lambda\big)^{w_i}\\ &= \frac{\prod_{i=1}^{M}\pi_i^{w_i}(z\mid s_t)Z_i^{w_i}(s_t)}{\pi_{\text{pre}}}.\end{aligned}$$

Plugging the above equation into equation 22, it can be shown that

$$\pi_w(z\mid s_t) = \frac{\prod_{i=1}^{M}\pi_i^{w_i}(z\mid s_t)}{\int\prod_{i=1}^{M}\pi_i^{w_i}(z'\mid s_t)dz'}.$$

Based on the definition that $\pi_i(a_t|s_t) = p_i(\mathbf{x}_{T-t-1}|\mathbf{x}_{T-t})$ for $s_t = \mathbf{x}_{T-t}$ and $a_t = \mathbf{x}_{T-t-1}$, we know that

$$p_w(\mathbf{x}_{t-1}\mid\mathbf{x}_t) = \frac{\prod_{i=1}^{M}p_i^{w_i}(\mathbf{x}_{t-1}\mid\mathbf{x}_t)}{\displaystyle\int\prod_{i=1}^{M}p_i^{w_i}(\mathbf{x}'_{t-1}\mid\mathbf{x}_t)\,d\mathbf{x}'_{t-1}},$$

where each $p_i(\mathbf{x}_{t-1}\mid\mathbf{x}_t)$ follows the Gaussian distribution with mean and variance (μ_i,σ_i^2). Thus, $p_w(\mathbf{x}_{t-1}\mid\mathbf{x}_t)$ is also a Gaussian distribution, and $p_w(\mathbf{x}_{t-1}\mid\mathbf{x}_t)\propto\exp\left(-\sum_{i=1}^{M}\frac{w_i}{2\sigma_i^2}\|\mathbf{x}_{t-1}-\mu_i\|^2\right)$. We can then derive its mean and variance as:

$$\sigma_w^2 = \left(\sum_{i=1}^{M}\frac{w_i}{\sigma_i^2}\right)^{-1},$$

$$\mu_w = \sigma_w^2\sum_{i=1}^{M}\frac{w_i}{\sigma_i^2}\,\mu_i.$$

This completes the proof. □

D Approximated Errors in Existing Denoising-Time Works

DB-MPA (Cheng et al., 2026) considers a stochastic differential equation (SDE) formulation:

$$\mathrm{d}\mathbf{x}_t = f^{\text{pre}}(\mathbf{x}_t,t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}w_t,\qquad \forall\, t\in[T,0],$$

where $\sigma(t),\beta(t)$ represents two fixed scalar functions, w_t is a standard Wiener process, p_t is a marginal distribution and $f^{\text{pre}}(\mathbf{x}_t,t)$ denotes the term

$$f^{\text{pre}}(\mathbf{x}_t,t) = -\tfrac{1}{2}\beta(t)\,\mathbf{x}_t\;-\;\beta(t)\,\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t).$$

To maximize equation 1 with reward r, the SDE becomes

$$\mathrm{d}\mathbf{x}_t = f^{(r,\lambda)}(\mathbf{x}_t,t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}w_t,\qquad \forall\, t\in[T,0],$$

where

$$f^{(r,\lambda)}(\mathbf{x}_t,t) = f^{\text{pre}}(\mathbf{x}_t,t) + u^{(r,\lambda)}(\mathbf{x}_t,t),$$

and

$$u^{(r,\lambda)}(\mathbf{x}_t,t) = \nabla_{\mathbf{x}_t}\log\mathbb{E}_{\mathbf{x}_0\sim p^{\text{pre}}_{0|t}(\cdot|\mathbf{x}_t)}\left[\exp\left(\frac{r(\mathbf{x}_0)}{\lambda}\right)\right].$$

Let $\Delta^{(r_i,\lambda)}(\mathbf{x},t) = u^{(r,\lambda)}(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\mathbb{E}_{\mathbf{x}_0\sim p^{\text{pre}}_{0|t}(\cdot|\mathbf{x}_t)}\left[\frac{r(\mathbf{x}_0)}{\lambda}\right]$, for each $r^w = \sum_{i=1}^M w_i r_i$, it can be shown that with the identical variance,

$$f^{(r^w,\lambda)}(\mathbf{x}_t,t) = \sum_{i=1}^M w_i\, f^{(r_i,\lambda)}(\mathbf{x}_t,t) + \Big(\Delta^{(r^w,\lambda)}(\mathbf{x}_t,t) - \sum_{i=1}^M w_i\, \Delta^{(r_i,\lambda)}(\mathbf{x}_t,t)\Big). \tag{24}$$

To get the denoising-time solution, DB-MPA ignores the last two terms in equation 24 and gets $f^{(r^w,\lambda)}(\mathbf{x}_t,t) \approx \sum_{i=1}^M w_i\, f^{(r_i,\lambda)}(\mathbf{x}_t,t)$.

DERADIFF (Manujith et al., 2026) studies the RL fine-tuning problem in equation 1. Similar to our Lemma 2 and Theorem 1, the optimal solution should satisfy that

$$p_w(\mathbf{x}_0) = \frac{\prod_{i=1}^M p_i^{w_i}(\mathbf{x}_0)}{\int \prod_{i=1}^M p_i^{w_i}(\mathbf{x}_0')d\mathbf{x}_0'}. \tag{25}$$

However, it is hard to track the marginal distribution of $\mathbf{x}_0$, thus DERADIFF conducts a stepwise approximation for any t without any theoretical justification:

$$p_w(\mathbf{x}_{t-1}|\mathbf{x}_t) = \frac{\prod_{i=1}^M p_i^{w_i}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\int \prod_{i=1}^M p_i^{w_i}(\mathbf{x}_{t-1}'|\mathbf{x}_t)d\mathbf{x}_{t-1}'}. \tag{26}$$

E Experiment details

E.1 Prompts

For completeness, we provide the prompt dataset (Cheng et al., 2026) in Table 3, which is a subset of DrawBench (Saharia et al., 2022) restricted to the "color" category. To obtain test prompts that do not appear in the training data, GPT-4 is used to synthesize novel color–object and object–object combinations derived from the training set.

Training Prompts	**Test Prompts**
A red colored car.	A white colored dog.
A black colored car.	A purple colored dog.
A pink colored car.	A yellow colored dog.
A black colored dog.	A green colored apple.
A red colored dog.	A black colored apple.
A blue colored dog.	A blue colored apple.
A green colored banana.	A purple colored apple.
A red colored banana.	A pink colored banana.
A black colored banana.	A pink colored cup.
A white colored sandwich.	A purple colored sandwich.
A black colored sandwich.	A green colored giraffe.
An orange colored sandwich.	A blue colored backpack.
A pink colored giraffe.	A blue car and a pink sheep.
A yellow colored giraffe.	A red apple and a purple backpack.
A brown colored giraffe.	A pink car and a yellow sheep.
A red car and a white sheep.	A black cup and a yellow cell phone.
A blue bird and a brown bear.	A blue car and a red giraffe.
A green apple and a black backpack.	A yellow bird and a purple sheep.
A green cup and a blue cell phone.	A pink car and a green bear.
A yellow book and a red vase.	A purple hat and a black clock.
A white car and a red sheep.	A black chair and a red table.
A brown bird and a blue bear.	A red car and a blue bird.
A black apple and a green backpack.	A green car and a yellow banana.
A blue cup and a green cell phone.	A pink vase and a red apple.
A red book and a yellow vase.	A blue book and an orange colored sandwich.

Table 3: Training and test prompt datasets used in the experiments.

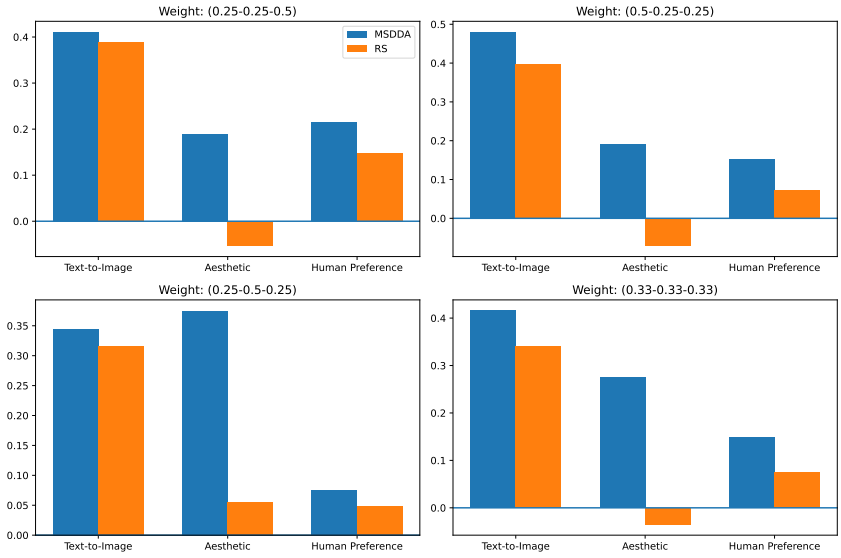


Figure 4: Performance comparison of our proposed method and RS via three rewards.

	MSDDA			DB-Extended		
w_1	r_1	r_2	$\sum w_i r_i$	r_1	r_2	$\sum w_i r_i$
0	0.15± 0.09	0.53 ± 0.02	0.53	0.15± 0.09	0.53± 0.02	0.53
0.1	0.18± 0.09	0.51± 0.02	**0.48**	0.23± 0.09	0.48± 0.02	0.46
0.2	0.22± 0.09	0.48± 0.02	**0.43**	0.37± 0.09	0.42 ± 0.02	0.41
0.3	0.30± 0.09	0.46± 0.02	**0.42**	0.42± 0.09	0.35 ± 0.02	0.37
0.4	0.37± 0.09	0.43± 0.02	**0.41**	0.49± 0.09	0.28± 0.02	0.40
0.5	0.42± 0.09	0.38± 0.02	**0.40**	0.53± 0.09	0.20± 0.02	0.37
0.6	0.45± 0.09	0.31± 0.02	**0.40**	0.56± 0.09	0.13± 0.02	0.39
0.7	0.51± 0.09	0.23± 0.02	0.43	0.63± 0.09	0.06± 0.02	**0.46**
0.8	0.57± 0.09	0.14± 0.02	0.48	0.65± 0.09	0.02± 0.03	**0.53**
0.9	0.62± 0.08	0.02± 0.02	0.56	0.67± 0.08	-0.06± 0.02	**0.60**
1	0.69± 0.08	-0.12 ± 0.02	0.69	0.69± 0.08	-0.12± 0.02	0.69

Table 4: Comparison between MSDDA and DB-Extended across different w_1 values with 95% confidence interval.

E.2 Training and Evaluation Details

To get the base model via RL, we optimize the step-level objective in equation 3. In contrast to DPOK (Fan et al., 2023), which updates the reference policy at every epoch, we only update π_{ref} periodically (every 1000 epochs) as a practical implementation detail to improve empirical performance. For each diffusion model, we used one H100 GPU for fine-tuning, where the batch size was set to 2 and gradient accumulation was 12. We used the same setup (Cheng et al., 2026) that the learning rate of AdamW was set to 1×10^{-5} and the LoRA rank was set to 4. For the policy update, we trained the model for 8000 epochs, and other settings were the same as Fan et al. (2023): we set the clipping ratio to 1×10^{-4} and ran 5 policy gradient steps and value function updates each iteration. The η of the denoising scheduler was set to 1.0 for the VILA model and 0.8 for the ImageReward model. All methods sample using classifier-free guidance (CFG) with a guidance scale of 7.5. CFG is applied independently to each objective-specific model before posterior fusion. For CoDe, we set the number of particles for the search to 20 and the lookahead steps to 5.

E.3 Additional Experiments

As we mentioned before, DB-MPA (Cheng et al., 2026), requires identical denoising time variances across models and is therefore incompatible with our experimental setup. To address this limitation, we introduce a simple extension, termed DB-Extended, which constructs a new Gaussian distribution

by taking a weighted linear combination of the means and variances. The corresponding results are reported in Table. 4. The results indicate that our method achieves a higher weighted reward in most cases, which is consistent with our theoretical results.

To evaluate our method in a more challenging setting, we consider three reward functions. In addition to ImageReward and VILA used in the main text, we incorporate an additional reward , PickScore (Kirstain et al., 2023), to capture human preference, using a linear transformation $r \mapsto r - 22.5$. The results are shown in Figure 4. We observe that MSDDA consistently outperforms the RS baselines.

E.4 Evaluation With More General Prompts

Our main experiments use prompts from the "color" category of DrawBench. To evaluate whether MSDDA generalizes beyond this setting, we additionally consider prompts from the "color attribution" and "counting" categories of GenEval. Tables 5 and 6 report the corresponding results.

Across both categories, MSDDA consistently improves the weighted reward over RS and is competitive with or better than DB-Extended. These results suggest that the benefits of MSDDA extend beyond the DrawBench color prompts to more general compositional tasks.

Table 5: Results on the "counting" category of the held-out GenEval prompts (140 prompts $\times$ 7 seeds). Here, r_1 denotes ImageReward and r_2 denotes the VILA reward. We report the mean and 95% confidence interval.

Method	w	r_1	r_2	$\sum w_i r_i$
SD	–	0.180 ± 0.105	-0.025 ± 0.042	–
RS	0.2	0.144 ± 0.059	0.302 ± 0.024	0.270 ± 0.023
	0.5	0.263 ± 0.055	0.110 ± 0.024	0.186 ± 0.029
	0.8	0.336 ± 0.055	0.051 ± 0.024	0.279 ± 0.044
DB-Extended	0.2	0.215 ± 0.058	0.391 ± 0.023	0.356 ± 0.023
	0.5	0.307 ± 0.056	0.262 ± 0.025	0.284 ± 0.031
	0.8	0.374 ± 0.054	0.165 ± 0.024	$\mathbf{0.332} \pm 0.044$
MSDDA	0.2	0.151 ± 0.058	0.438 ± 0.023	$\mathbf{0.381} \pm 0.022$
	0.5	0.226 ± 0.057	0.355 ± 0.024	$\mathbf{0.290} \pm 0.032$
	0.8	0.328 ± 0.056	0.235 ± 0.025	0.310 ± 0.045

Table 6: Results on the "color attribution" category of the held-out GenEval prompts (140 prompts $\times$ 7 seeds). Here, r_1 denotes ImageReward and r_2 denotes the VILA reward. We report the mean and 95% confidence interval.

Method	w	r_1	r_2	$\sum w_i r_i$
SD†	–	-0.501 ± 0.157	-0.059 ± 0.040	–
RS	0.2	-0.414 ± 0.082	0.287 ± 0.021	0.147 ± 0.024
	0.5	-0.277 ± 0.084	0.083 ± 0.022	-0.097 ± 0.045
	0.8	-0.093 ± 0.086	0.018 ± 0.022	-0.071 ± 0.069
DB-Extended	0.2	-0.243 ± 0.084	0.362 ± 0.022	0.241 ± 0.025
	0.5	-0.085 ± 0.084	0.225 ± 0.022	0.070 ± 0.044
	0.8	-0.070 ± 0.084	0.111 ± 0.022	-0.034 ± 0.068
MSDDA	0.2	-0.375 ± 0.083	0.416 ± 0.021	$\mathbf{0.258} \pm 0.025$
	0.5	-0.176 ± 0.085	0.324 ± 0.022	$\mathbf{0.074} \pm 0.045$
	0.8	-0.087 ± 0.085	0.186 ± 0.022	$-\mathbf{0.033} \pm 0.069$

E.5 Generated Images

In Figs. 5-8, we provide some generated image samples from SD, RS and our method with different w. The left image is generated by Stable Diffusion using the prompt shown in the lower-left corner.

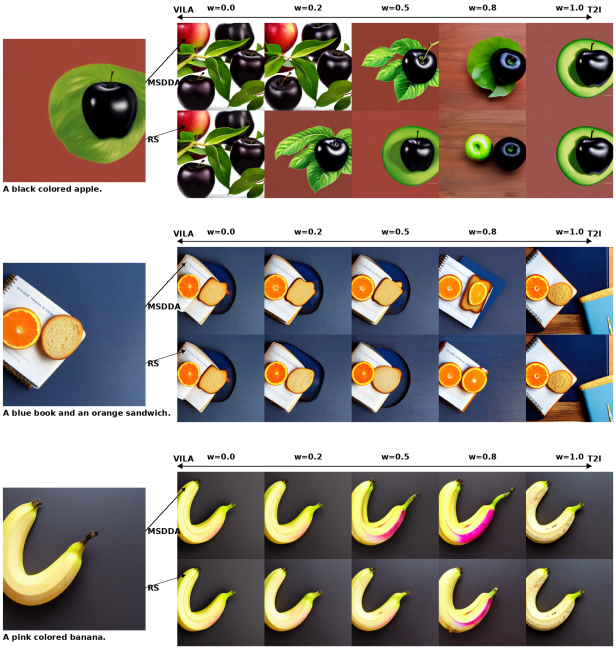


Figure 5: Image samples of our proposed method and RS.

The top row shows images generated by our proposed MSDDA, while the bottom row shows images generated by RS.

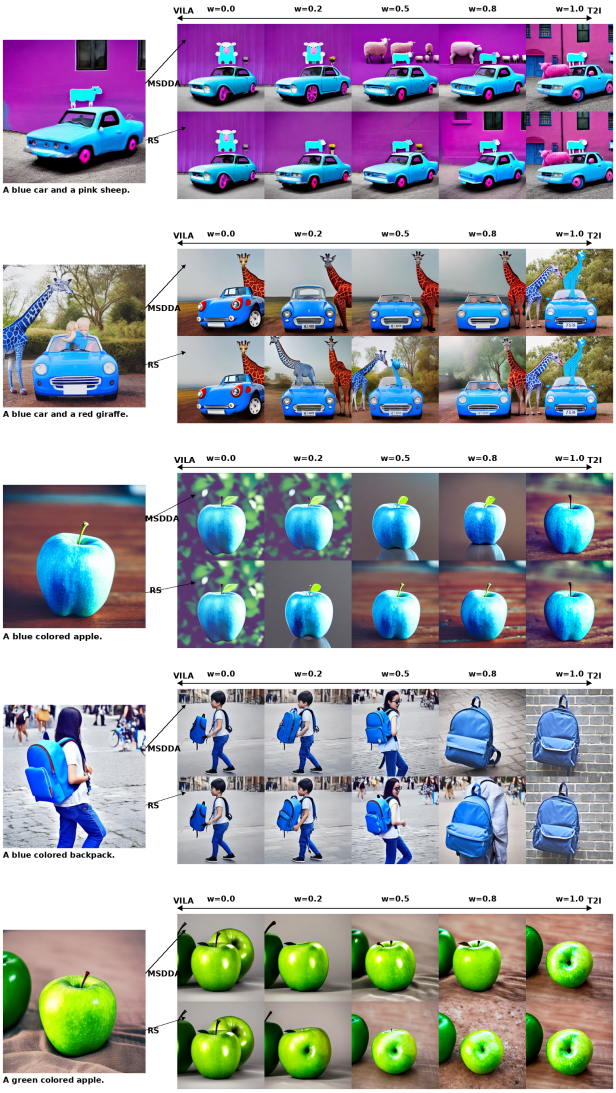


Figure 6: Image samples of our proposed method and RS.

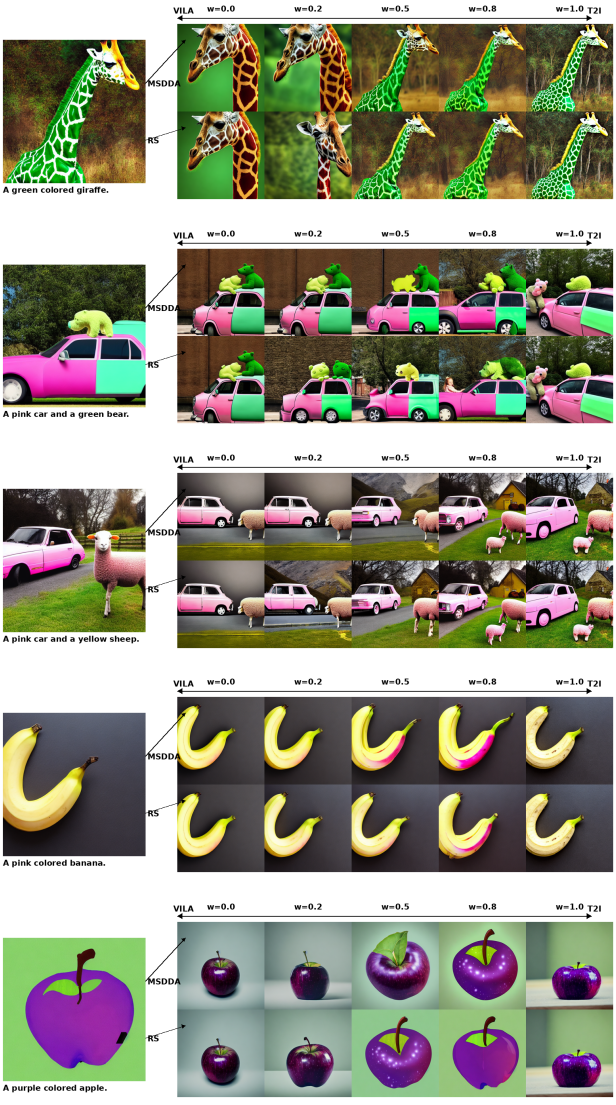


Figure 7: Image samples of our proposed method and RS.



Figure 8: Image samples of our proposed method and RS.