

---

# Composition and Alignment of Diffusion Models using Constrained Learning

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1        Diffusion models have become prevalent in generative modeling due to their ability  
2        to sample from complex distributions. To improve the quality of generated samples  
3        and their compliance with user requirements, two commonly used methods are:  
4        (i) Alignment, which involves fine-tuning a diffusion model to align it with a  
5        reward; and (ii) Composition, which combines several pre-trained diffusion models  
6        together, each emphasizing a desirable attribute in the generated outputs. However,  
7        trade-offs often arise when optimizing for multiple rewards or combining multiple  
8        models, as they can often represent competing properties. Existing methods cannot  
9        guarantee that the resulting model faithfully generates samples with all the desired  
10       properties. To address this gap, we propose a constrained optimization framework  
11       that unifies alignment and composition of diffusion models by enforcing that the  
12       aligned model satisfies reward constraints and/or remains close to each pre-trained  
13       model. We provide a theoretical characterization of the solutions to the constrained  
14       alignment and composition problems and develop a Lagrangian-based primal-dual  
15       training algorithm to approximate these solutions. Empirically, we demonstrate our  
16       proposed approach in image generation, applying it to alignment and composition,  
17       and show that our aligned or composed model satisfies constraints effectively.

## 18    1 Introduction

19    Diffusion models have emerged as the tool of choice for generative models in a variety of settings  
20    [36, 3, 48, 9], image generation being most prominent among them [35]. Users of these diffusion  
21    models would like to adapt them to their specific preferences, but this aspiration is hindered by  
22    the often enormous cost and complexity of their training [46, 54]. For this reason, *alignment* and  
23    *composition* of what, in this context, become *pretrained* models, has become popular [28, 29].

24    Regardless of whether the goal is alignment or composition, we want to balance what are most likely  
25    conflicting requirements. In alignment tasks, we want to stay close to the pretrained model while  
26    deviating sufficiently so as to effect some rewards of interest [16, 13]. In composition tasks we are  
27    given several pretrained models and our goal is to sample from their union or intersection [14, 1].  
28    The standard approach to balance these requirements involves the use of weighted averages. This  
29    can be a linear combination of score functions in composition problems [14, 1] or may involve a loss  
30    given by a linear combination of a Kullback-Leibler (KL) divergence and a reward [16] in the case of  
31    alignment.

32    In this work we propose a unified view of alignment and composition via the lens of constrained  
33    learning [7, 6]. As their names indicate, constrained alignment and constrained composition problems  
34    balance conflicting requirements using constraints instead of weights. Learning with constraints and  
35    learning with weights are related problems – indeed, we will train constrained diffusion models in  
36    their Lagrangian forms. Yet, they are also fundamentally different. In the constrained formulation,

the hyperparameter tuning spaces are more interpretable (see Section 3), and in some cases—such as the constrained composition formulation—hyperparameter tuning can even be avoided entirely (see Section 4). These advantages are particularly evident in constrained problems, as discussed in Sections 3 and 4. We next outline our key contributions in alignment and composition.

**Alignment.** For alignment, we formulate a reverse KL divergence-constrained optimization problem that minimizes the reverse KL divergence to a pre-trained model, subject to expected reward constraints. The threshold for each reward constraint can be user-specified or automatically selected using a heuristic approach (see Section 5). In Section 3 we show that the solution of this alignment problem is the pretrained model distribution scaled by an exponential function of a weighted sum of reward functions. To solve this problem with diffusion models, we establish strong duality, which enables us to employ a Lagrangian dual-based approach to develop a primal-dual training algorithm.

We demonstrate the differences between constrained and weighted alignment in numerical experiments in Section 5.1. The constrained approach easily scales to fine-tuning with multiple rewards, while avoiding the need for extensive hyperparameter search to find suitable weights. Moreover, specifying reward thresholds is more intuitive than selecting weights for each regularizer. Furthermore, without constraints, it is easy to overfit to one or multiple of the rewards and completely diverge from the pretrained model. In contrast, our method finds the closest model to the pretrained one that satisfies the reward constraints (see Figure 4).

**Composition.** For composition, we propose using KL divergence constraints to ensure the closeness to each individual model. It is important to distinguish composition with *reverse* KL and *forward* KL constraints. As previously shown in [21], using forward KL constraints results in the composed model sampling from a weighted mixture of the individual distributions. In the main paper we focus on composition with reverse KL constraints, while we discuss forward KL constraints in Appendix D. In Section 4, we characterize the solution of the constrained optimization problem with *reverse* KL divergence constraints as a tilted product of the individual distributions. To solve this problem with diffusion models, we similarly establish strong duality and develop a primal-dual training algorithm.

We demonstrate properties of constrained composition of models in numerical experiments in Section 5.2. We observe that if the composition weights are not chosen properly, it can lead to the composed model being biased towards some of the individual models while ignoring others. Constrained composition helps to avoid this by finding optimal weights that ensure closeness to each individual distribution. When composing multiple text-to-image models each finetuned on a different reward function, using constraints leads to optimal weights that result in the composed model having better performance on all of the rewards compared to just composing them with equal weights.

## 2 Composition and Alignment of Diffusion Models in Distribution Space

**Reward alignment:** Given a pretrained model  $q$  and a set of  $m$  rewards  $\{r_i(x)\}_{i=1}^m$  that can be evaluated on a sample  $x$ , we consider the *reverse* KL divergence  $D_{\text{KL}}(p \parallel q) := \int p(x) \log(p(x)/q(x)) dx$  that measures the difference between a distribution  $p$  and the pretrained model  $q$ . Additionally, for each reward  $r_i$ , we define a constant  $b_i$  standing for requirement for reward  $r_i$ . We formulate a constrained alignment problem that minimizes a reverse KL divergence subject to  $m$  constraints,

$$p^* = \underset{p}{\operatorname{argmin}} D_{\text{KL}}(p \parallel q) \quad \text{subject to } \mathbb{E}_{x \sim p}[r_i(x)] \geq b_i \text{ for } i = 1, \dots, m. \quad (\text{UR-A})$$

As per (UR-A), the constrained alignment problem is solved by the distribution  $p^*$  that is closest to the pretrained one  $q$  as measured by the reverse KL divergence  $D_{\text{KL}}(p \parallel q)$  among those whose expected rewards  $\mathbb{E}_{x \sim p}[r_i(x)]$  accumulate to at least  $b_i$ . By ‘pretrained model’ we refer to a sampling process that produces samples, not the underlying distribution. Let the primal value  $P_{\text{ALI}}^* := D_{\text{KL}}(p^* \parallel q)$ .

**Product composition (AND):** Given a set of  $m$  pretrained models  $\{q_i\}_{i=1}^m$ , we formulate a constrained composition problem that solves a reverse KL-constrained optimization problem,

$$(p^*, u^*) = \underset{p, u}{\operatorname{argmin}} u \quad \text{subject to } D_{\text{KL}}(p \parallel q_i) \leq u \text{ for } i = 1, \dots, m. \quad (\text{UR-C})$$

In (UR-C), the decision variable  $u$  is an upper bound on  $m$  KL divergences between a distribution  $p$  and  $m$  pretrained models  $\{q_i\}_{i=1}^m$ . Partial minimization over  $u$  allows us to search for a distribution  $p$  that minimizes a common upper bound. Thus, the optimal solution  $p^*$  minimizes the maximum KL divergence among  $m$  terms, each computed between  $p$  and a pretrained model. The epigraph



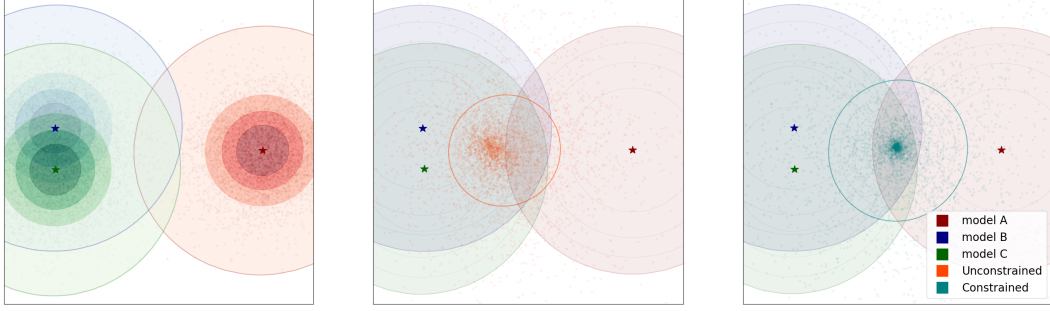


Figure 1: Product composition (AND): (left) Three Gaussian distributions being composed. (middle) Composition using equal weights and (right) with constraints. The constrained model samples from the intersection of the three models.

formulation (UR-C) is useful in practice, as the constraint threshold  $u$  is updated dynamically during training. Let the primal value be  $P_{\text{AND}}^* := u^*$ . See Figure 1 for an illustration. The unconstrained model composed with equal weights is biased towards the distributions that are closer to each other.

**Mixture composition (OR):** A different composition modality that also fits within our constrained framework is the *forward* KL-constrained composition problem. We obtain this formulation by replacing the *reverse* divergence  $D_{\text{KL}}(p \parallel q_i)$  in (UR-C) with the *forward* KL divergence  $D_{\text{KL}}(q_i \parallel p)$ ,

$$(p^*, u^*) = \underset{p, u}{\operatorname{argmin}} u \quad \text{subject to } D_{\text{KL}}(q_i \parallel p) \leq u \text{ for } i = 1, \dots, m. \quad (\text{UF-C})$$

We note that mixture composition has been studied in a related but slightly different constrained setting in [21]. It can be shown that the solution of the constrained problem (UF-C) learns to sample from each distribution proportional to its entropy [21]. In Figure 2, we see that the constrained model learns to sample more often from the distribution with two modes that has a higher entropy in contrast to the equally weighted composition that samples equally from both distributions leading to unbalanced sampling from the modes. Since the algorithm design and analysis for (UF-C) closely resemble those in [21], mixture composition is not the focus of this work. For completeness, we discuss and compare it with product composition in Appendix D.

The reverse KL-based composition (UR-C) tends to sample at the intersection of the pretrained models  $\{q_i\}_{i=1}^m$ , whereas the forward KL-based composition (UF-C) tends to sample at the union of the pretrained models  $\{q_i\}_{i=1}^m$ . Thus, product composition enforces a conjunction (logical AND) across pretrained models, while mixture composition corresponds to a disjunction (logical OR). We stress that Problems (UR-A), (UR-C), and (UF-C) should be viewed as canonical formulations; the methods proposed in this paper can be readily adapted to solve their variants e.g. mixture composition with reward constraints.

## 2.1 KL divergence for diffusion models

A generative diffusion model consists of forward and backward processes. In the forward process, we add Gaussian noise  $\epsilon_t$  to a clean sample  $\bar{X}_0 \sim \bar{p}_0$  over  $T$  time steps,

$$\bar{X}_t = \frac{\alpha_t}{\alpha_{t-1}} \bar{X}_{t-1} + \sqrt{1 - \frac{\alpha_t}{\alpha_{t-1}}} \epsilon_t, \text{ for } t \in \{1, \dots, T\} \quad (1)$$

where  $\epsilon_t \sim \mathcal{N}(0, I)$  and  $\{\alpha_t\}_{t=1}^T$  is a decreasing sequence of coefficients called the noise schedule. We denote the marginal density of  $\bar{X}_t$  at time  $t$  as  $\bar{p}_t(\cdot)$ . Given a  $d$ -dimensional score predictor function  $s(x, t): \mathbb{R}^d \times \{1, \dots, T\} \rightarrow \mathbb{R}^d$ , we define a backward denoising diffusion implicit model (DDIM) process [40],

$$X_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} X_t + \beta_t s(X_t, t) + \sigma_t \epsilon_t \quad (2)$$

where  $\epsilon_t \sim \mathcal{N}(0, I)$ , and  $\{\sigma_t^2\}_{t=1}^T$  is the variance schedule determining the level of randomness in the backward process (e.g.,  $\sigma_t = 0$  reduces to deterministic trajectories), and  $\beta_t :=$

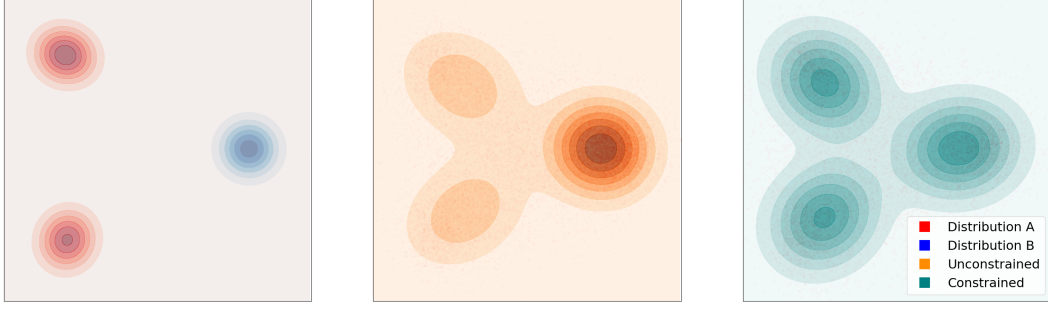


Figure 2: Mixture composition (OR): (left) Two of Gaussian mixtures being composed. One has two modes and the other has only a single mode. (middle) Composition using equal weights and (right) with constraints.

116  $\sqrt{\frac{\alpha_t-1}{\alpha_t}}\sqrt{(1-\alpha_t)(1-\bar{\alpha}_t)} - \sqrt{(1-\alpha_{t-1}-\sigma_t^2)(1-\bar{\alpha}_t)}$  is determined by the variance schedule  
 117  $\sigma_t$  and the noise schedule  $\alpha_t$ . Given a function  $s$ , we denote the marginal density of  $X_t$  as  $p_t(\cdot; s)$   
 118 and the joint distribution over the entire process as  $p_{0:T}(x_{0:T}; s)$ .

119 In the score-matching formulation [41], a denoising score-matching objective is minimized to obtain  
 120 a function  $s^*$  that approximates the true score function of the forward process, i.e.,  $s^*(x, t) \approx$   
 121  $\nabla \log \bar{p}_t(x)$ . Then, the marginal densities of the backward process (2) match those of the forward  
 122 process (1), i.e.,  $p_t(\cdot; s^*) = \bar{p}_t(\cdot)$  for all  $t$ . Thus we can run the backward process to generate samples  
 123  $x_0 \sim p_0$  that resemble samples from the original data distribution  $\bar{x}_0 \sim \bar{p}_0$ .

124 We denote the KL divergence between two joint distributions  $p, q$  over the entire backward process by  
 125  $D_{\text{KL}}(p_{0:T}(\cdot) \parallel q_{0:T}(\cdot))$ , which is known as the path-wise KL divergence [16, 18]. This path-wise KL  
 126 divergence is often used in alignment to quantify the gap between finetuned and pretrained models.

127 **Lemma 1** (Path-wise KL divergence). *If two backward processes  $p_{0:T}(\cdot)$  and  $q_{0:T}(\cdot)$  have the same*  
 128 *variance schedule  $\sigma_t$  and noise schedule  $\alpha_t$ , then the reverse KL divergence between them is given by*

$$D_{\text{KL}}(p_{0:T}(\cdot; s_p) \parallel p_{0:T}(\cdot; s_q)) = \sum_{t=1}^T \mathbb{E}_{x_t \sim p_t(\cdot; s_p)} \left[ \frac{1}{2\sigma_t^2} \|s_p(x_t, t) - s_q(x_t, t)\|^2 \right]. \quad (3)$$

129 See Appendix C.1 for proof. While the path-wise KL divergence is a useful regularizer for alignment,  
 130 when composing multiple models, the point-wise KL divergence  $D_{\text{KL}}(p_0(\cdot) \parallel p_0(\cdot))$  is a more natural  
 131 measure of the closeness between two diffusion models. This is because we mainly care about  
 132 the closeness of the final sampling distributions:  $p_0(\cdot)$ ,  $q_0(\cdot)$ , and not the underlying processes:  
 133  $p_{0:T}(\cdot)$ ,  $q_{0:T}(\cdot)$ . In this work, we use path-wise KL for alignment and point-wise KL for composition.  
 134 However, it is not obvious how to compute the point-wise KL, as evaluating the marginal densities is  
 135 intractable. We next establish a similar formula as (3) by restricting the score function class.

136 **Lemma 2** (Point-wise KL divergence). *Assume two score functions  $s_p(x, t) = \nabla \log \bar{p}_t(x)$ ,*  
 137  *$s_q(x, t) = \nabla \log \bar{q}_t(x)$ , where  $\bar{p}_t, \bar{q}_t$  are two marginal densities induced by two forward diffu-*  
 138 *sion processes, with the same noise schedule, starting from initial distributions  $\bar{p}_0$  and  $\bar{q}_0$ , respectively.*  
 139 *Then, the point-wise KL divergence between two distributions of the samples generated by running*  
 140 *DDIM with  $s_p$  and  $s_q$  is given by*

$$D_{\text{KL}}(p_0(\cdot; s_p) \parallel p_0(\cdot; s_q)) = \sum_{t=0}^T \tilde{\omega}_t \mathbb{E}_{x \sim p_t(\cdot; s_p)} \left[ \|s_p(x, t) - s_q(x, t)\|_2^2 \right] + \epsilon_T \quad (4)$$

141 where  $\tilde{\omega}_t$  is a time-dependent constant and  $\epsilon_T$  is a discretization error depending on the number of  
 142 diffusion time steps  $T$ .

143 See Appendix C.2 for the proof. The key intuition behind Lemma 2 is that if two diffusion processes  
 144 are close, and their starting distributions are the same (e.g.,  $\mathcal{N}(0, I)$  at time  $t = T$ ), then the end  
 145 points (i.e., the distributions at  $t = 0$ ) must also be close. The sum on the right hand side of (44) can  
 146 be viewed as the difference of the processes over time steps, up to a discretization error.

### 3 Aligning Pretrained Model with Multiple Reward Constraints

To apply Problem (UR-A) to diffusion models, we first employ Lagrangian duality to derive its solution in the distribution space. Alignment with constraints is related but fundamentally different from the standard approach of minimizing a weighted average of the KL divergence and rewards [16]. They are related because the Lagrangian for (UR-A) is precisely the weighted average,

$$L_{\text{ALI}}(p, \lambda) = D_{\text{KL}}(p \| q) - \lambda^\top (\mathbb{E}_{x \sim p}[r(x)] - b). \quad (5)$$

where we use shorthands  $b := [b_1, \dots, b_m]^\top$ ,  $r := [r_1, \dots, r_m]^\top$ , and  $\lambda := [\lambda_1, \dots, \lambda_m]^\top$  is the Lagrangian multiplier or dual variable. Let the dual function be  $D_{\text{ALI}}(\lambda) := \min_{p \in \mathcal{P}} L_{\text{ALI}}(p, \lambda)$  and an optimal dual variable be  $\lambda^* \in \arg\max_{\lambda \geq 0} D_{\text{ALI}}(\lambda)$ . Denote  $D_{\text{ALI}}^* := D_{\text{ALI}}(\lambda^*)$ . For  $\lambda > 0$ , we define the reward weighted distribution  $q_{\text{rw}}^{(\lambda)}$  as

$$q_{\text{rw}}^{(\lambda)}(\cdot) = \frac{1}{Z_{\text{rw}}(\lambda)} q(\cdot) e^{\lambda^\top r(\cdot)} \quad (6)$$

where  $Z_{\text{rw}}(\lambda) := \int q(x) e^{\lambda^\top r(x)} dx$  is a normalizing constant.

In the distribution space, Problem (UR-A) is a convex optimization problem since the KL divergence is strongly convex and the reward constraints are linear in  $p \in \mathcal{P}$ . Thus, we apply strong duality in convex optimization [4] to characterize the solution to Problem (UR-A) in Theorem 1. Moreover, it is ready to formulate the constrained alignment problem (UR-A) as an unconstrained problem by specializing the dual variables to a solution to the dual problem.

**Assumption 1** (Feasibility). *There exist a model  $p$  such that  $\mathbb{E}_{x \sim p}[r_i(x)] > b_i$  for all  $i = 1, \dots, n$ .*

**Theorem 1** (Reward alignment). *Let Assumption 1 hold. Then, Problem (UR-A) is strongly dual, i.e.,  $P_{\text{ALI}}^* = D_{\text{ALI}}^*$ . Moreover, Problem (UR-A) is equivalent to*

$$\min_{p \in \mathcal{P}} D_{\text{KL}}(p \| q_{\text{rw}}^{(\lambda^*)}) \quad (7)$$

where  $\lambda^*$  is the optimal dual variable, and the dual function has the explicit form,  $D_{\text{ALI}}(\lambda) = -\log Z_{\text{rw}}(\lambda)$ . Furthermore, the optimal solution of (UR-A) is given by

$$p^* = q_{\text{rw}}^{(\lambda^*)}. \quad (8)$$

See Appendix C.3 for proof. Theorem 1 characterizes the solution to the constrained alignment problem (UR-A), i.e.,  $q_{\text{rw}}^{(\lambda^*)}$ . This solution generalizes the reward-tilted distribution [13], which is the solution of finetuning a model with an expected reward regularizer. In Problem (UR-A), the optimal dual variable  $\lambda^*$  weights each reward so that all the constraints are satisfied optimally, while staying as close as possible to the pretrained model.

#### 3.1 Reward alignment of diffusion models

We now introduce diffusion models to Problem (UR-A) by representing  $p$  and  $q$  as two diffusion models  $p_{0:T}(\cdot; s_p)$  and  $q_{0:T}(\cdot; s_q)$ , respectively. The path-wise KL divergence has been widely used in diffusion model alignment to capture the difference between two diffusion models (see [16]). Hence, we can instantiate Problem (UR-A) in the function space below,

$$\begin{aligned} & \min_{s_p \in \mathcal{S}} D_{\text{KL}}(p_{0:T}(\cdot; s_p) \| q_{0:T}(\cdot; s_q)) \\ & \text{subject to } \mathbb{E}_{x_0 \sim p_{0:T}(\cdot; s_p)}[r_i(x_0)] \geq b_i \quad \text{for } i = 1, \dots, m. \end{aligned} \quad (\text{SR-A})$$

We define a Lagrangian for Problem (SR-A) as  $\bar{L}_{\text{ALI}}(s_p, \lambda) := L_{\text{ALI}}(p_{0:T}(\cdot; s_p), \lambda)$ . Similarly, we introduce the primal and dual values  $\bar{P}_{\text{ALI}}^*$  and  $\bar{D}_{\text{ALI}}^*$ . Although Problem (SR-A) is a non-convex optimization problem, the strong duality still holds.

**Theorem 2** (Strong duality). *Let Assumption 1 hold for some  $s \in \mathcal{S}$ . Then, Problem (SR-A) is strongly dual, i.e.,  $\bar{P}_{\text{ALI}}^* = \bar{D}_{\text{ALI}}^*$ .*

See the proof of Theorem 2 in Appendix C.4. Motivated by strong duality, we present a dual-based method for solving Problem (SR-A) in which we alternate between minimizing the Lagrangian via gradient descent steps and maximizing the dual function via dual sub-gradient ascent steps below.

185 **Primal minimization:** At iteration  $n$ , we obtain a new model  $s^{(n+1)}$  via a Lagrangian maximization,

$$s^{(n+1)} \in \operatorname{argmin}_{s \in \mathcal{S}} \bar{L}_{\text{ALI}}(s_p, \lambda^{(n)}). \quad (9)$$

186 **Dual maximization:** Then, we use the model  $s^{(n+1)}$  to estimate the constraint violation  $\mathbb{E}_{x_0}[r(x_0)] -$   
 187  $b$ , denoted as  $r(s^{(n+1)}) - b$ , and perform a dual sub-gradient ascent step,

$$\lambda^{(n+1)} = \left[ \lambda^{(n)} + \eta \left( r(s^{(n+1)}) - b \right) \right]_+. \quad (10)$$

## 188 4 Constrained Composition of Multiple Pretrained Models

189 To apply Problem (UR-C) to diffusion models, we first employ Lagrangian duality to derive its  
 190 solution in the distribution space  $\mathcal{P}$ . Let the Lagrangian for Problem (UR-C) be

$$L_{\text{AND}}(p, u, \lambda) = u + \sum_{i=1}^m \lambda_i (D_{\text{KL}}(p \| q^i) - u), \quad (11)$$

191 and the associated dual function  $D_{\text{AND}}(\lambda)$ , which is always concave, is defined as

$$D_{\text{AND}}(\lambda) := \max_{u \in \mathbb{R}, p \in \mathcal{P}} L_{\text{AND}}(p, u, \lambda). \quad (12)$$

192 Let a solution to Problem (UR-A) be  $(p^*, u^*)$ , and let the optimal value of the objective function be  
 193  $P_{\text{AND}}^* = u^*$ . Let an optimal dual variable pair be  $\lambda^* \in \operatorname{argmax}_{\lambda \geq 0} D_{\text{AND}}(\lambda)$ , and the optimal value  
 194 of the dual function be  $D_{\text{AND}}^* := D_{\text{AND}}(\lambda^*)$ .

195 For  $\lambda > 0$ , we define the tilted product distribution  $q_{\text{AND}}^{(\lambda)}$  as a product of  $m$  tilted distributions  $q^i$ ,

$$q_{\text{AND}}^{(\lambda)}(\cdot) = \frac{1}{Z_{\text{AND}}(\lambda)} \prod_{i=1}^m (q^i(\cdot))^{\frac{\lambda_i}{1^\top \lambda}} \quad (13)$$

196 where  $Z_{\text{AND}}(\lambda) := \int \prod_{i=1}^m (q^i(x))^{\frac{\lambda_i}{1^\top \lambda}} dx$  is a normalizing constant.

197 **Assumption 2** (Feasibility). *There exist a model  $p$  such that  $D_{\text{KL}}(p \| q_i) < u$  for all  $i = 1, \dots, n$ .*

198 Note that Assumption 2 only requires that the supports of the distributions  $q^i$  have non-empty  
 199 intersection.

200 **Theorem 3** (Product composition). *Let Assumption 2 hold. Then, Problem (UR-C) is strongly dual,*  
 201 *i.e.,  $P_{\text{AND}}^* = D_{\text{AND}}^*$ . Moreover, Problem (UR-C) is equivalent to*

$$\underset{p \in \mathcal{P}}{\text{minimize}} \quad D_{\text{KL}}(p \| q_{\text{AND}}^{(\lambda^*)}) \quad (14)$$

202 where  $\lambda^*$  is the optimal dual variable, and the dual function has the explicit form,  $D(\lambda) =$   
 203  $-\log Z_{\text{AND}}(\lambda)$ . Furthermore, the optimal solution of (14) is given by

$$p^* = q_{\text{AND}}^{(\lambda^*)}. \quad (15)$$

204 We defer the proof of Theorem 3 to Appendix C.5. The distribution  $q_{\text{AND}}^{(\lambda)} \propto \prod_{i=1}^m (q^i(\cdot))^{\frac{\lambda_i}{1^\top \lambda}}$  allows  
 205 sampling from a weighted product of  $m$  distributions, where the parameters  $\{\lambda_i / 1^\top \lambda\}_{i=1}^m$  weight  
 206 the importance of each distribution. The geometric mean [1] is a special case when all  $\lambda_i$  are equal.

207 **Remark 1.** *Theorem 3 connects our proposed constrained optimization problem (UR-C) to the*  
 208 *well-known problem of sampling from a product of multiple distributions [1, 14]. Furthermore, our*  
 209 *constraints enforce that the resulting product is properly weighted to ensure the solution diverges as*  
 210 *little as possible from each of the individual distributions (see Figure 1).*

#### 211 4.1 Product composition of diffusion models

212 We introduce diffusion models to Problem (UR-A) via an optimization problem in the function space,

$$\begin{aligned} & \underset{u \geq 0, s \in \mathcal{S}}{\text{minimize}} && u \\ & \text{subject to} && D_{\text{KL}}(p(x_0; s) \| p(x_0; s^i)) \leq u \quad \text{for } i = 1, \dots, m. \end{aligned} \quad (\text{SR-C})$$

213 We define a Lagrangian for Problem (SR-C) as  $\bar{L}_{\text{AND}}(s_p, u, \lambda) := L_{\text{AND}}(p(x_0; s_p), u, \lambda)$ . Similarly,  
214 we introduce the primal and dual values  $\bar{P}_{\text{AND}}^*$  and  $\bar{D}_{\text{AND}}^*$ . Although Problem (SR-C) is a non-convex  
215 optimization problem, the strong duality still holds.

216 **Theorem 4** (Strong duality). *Let Assumption 2 hold for some  $p(\cdot; s)$  with  $s \in \mathcal{S}$ . Then, Problem (SR-C) is strongly dual, i.e.,  $\bar{P}_{\text{AND}}^* = \bar{D}_{\text{AND}}^*$ .*  
217

218 See the proof of Theorem 4 in Appendix C.6. For solving the constrained optimization problem (SR-C) we can use a primal-dual approach similar to the one discussed in Section 3.1.

220 **Primal minimization:** At iteration  $n$ , we obtain a new model  $s^{(n+1)}$  via a Lagrangian maximization,

$$s^{(n+1)} \in \underset{s \in \mathcal{S}}{\text{argmin}} \bar{L}_{\text{AND}}(s_p, \lambda^{(n)}). \quad (16)$$

221 **Dual maximization:** Then, we use the model  $s^{(n+1)}$  to estimate the constraint violation and perform  
222 a dual sub-gradient ascent step,

$$\lambda^{(n+1)} = \left[ \lambda^{(n)} + \eta \left( D_{\text{KL}}(p(x_0; s^{(n+1)}) \| p(x_0; s^i)) - u \right) \right]_+. \quad (17)$$

223 We note that computing the point-wise KL that shows up in both the Lagrangian and the constraint violations is not trivial. Recall that Lemma 2 gives us a way to compute the point-wise KL  
224  $D_{\text{KL}}(p(x_0; s) \| p(x_0; s^i))$ . However, it requires that the functions  $s$  and  $s^i$  each be a valid score function for some process. It is reasonable to assume this is the case for  $s^i$  since it represents  
225 a pretrained model where it would have been trained to approximate the true score of a forward  
226 diffusion process. Yet regarding the function  $s$  that we are optimizing over, there is no guarantee that  
227 any given  $s \in \mathcal{S}$  is a valid score function. Lemma 3 lets us minimize the Lagrangian in spite of this:

230 **Lemma 3.** *The Lagrangian for Problem (SR-C) is equivalently written as*

$$L_{\text{AND}}(s, \lambda) = D_{\text{KL}}(p(x_0; s) \| q_{\text{AND}}^{(\lambda)}(x_0)) - \log Z_{\text{AND}}(\lambda). \quad (18)$$

231 *Furthermore, a Lagrangian minimizer  $s^{(\lambda)} := \underset{s \in \mathcal{S}}{\text{argmin}} L_{\text{AND}}(s, \lambda)$  is given by*

$$s^{(\lambda)} = \underset{s \in \mathcal{S}}{\text{argmin}} \sum_{t=0}^T \omega_t \mathbb{E}_{x_0 \sim q_{\text{AND}}^{(\lambda)}(\cdot)} \mathbb{E}_{x_t \sim q(x_t | x_0)} \left[ \|s(x, t) - \nabla \log q(x_t | x_0)\|^2 \right] \quad (19)$$

232 *where  $q(x_t | x_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)I)$ , and  $s^{(\lambda)} = \nabla \log q_{\text{AND}, t}^{(\lambda)}$ .*

233 See Appendix C for proof. With Lemma 3, as long as we have access to samples from the distribution  
234  $q_{\text{AND}}^{(\lambda)}$ , we can approximate the expectation in (19) and use gradient-based optimization methods  
235 to find the minimizer  $s^{(\lambda)}$ . To obtain these samples, we use annealed Markov Chain Monte Carlo  
236 (MCMC) sampling as proposed by [14]; see Appendix B for sampling details. For the dual update, we  
237 can evaluate the KL divergence  $D_{\text{KL}}(p_0(\cdot; s^{(\lambda)}) \| p_0(\cdot; s^i))$  between the marginal densities induced  
238 by the Lagrangian minimizer  $s^{(\lambda)}$  and the individual score functions  $s^i$  using Lemma 2 since both  
239 are valid score functions.

240 **Remark 2.** *In practice the primal steps will yield an approximate Lagrangian minimizer  $s^{(\tilde{\lambda})}(x, t) \approx$   
241  $\nabla \log q_{\text{AND}, t}^{(\lambda)}(x)$ . This results in two sources of error in evaluating the expectations on the RHS  
242 of (44):*

$$D_{\text{KL}}(p_0(\cdot; s^{(\lambda)}) \| p_0(\cdot; s^i)) = \sum_{t=0}^T \tilde{\omega}_t \mathbb{E}_{x \sim p_t(\cdot; s^{(\lambda)})} \left[ \|s^{(\lambda)}(x, t) - s^i(x, t)\|_2^2 \right] + \epsilon_T \quad (20)$$

243 *First, the error induced by not using the exact  $s^{(\lambda)}$  in  $\|s^{(\lambda)}(x, t) - s^i(x, t)\|_2^2$ . Second, the error*  
244 *induced by not evaluating the expectation on correct trajectories given by  $x \sim p_t(\cdot; s^{(\lambda)})$ . However*  
245 *the second error can be reduced since if we have a way of sampling from the true product  $x_0 \sim q_{\text{AND}, 0}^{(\lambda)}$ ,*  
246 *we can get samples from  $p_t(\cdot; s^{(\lambda)})$  just by adding Gaussian noise to  $x_0$ .*



See Appendix E for the detailed algorithm for product composition.

## 5 Computational experiments

### 5.1 Finetuning to align with multiple rewards

We extend the AlignProp framework [33] to accommodate multiple reward constraints and dual updates. We finetune Stable Diffusion v1.5<sup>1</sup> on widely used differentiable image quality and aesthetic rewards, namely aesthetic [37], hps [50], pickscore [22], imagereward [52] and mps [58]. Since these rewards have widely varying scales, which can make setting the constraint levels challenging, we normalize them by computing the average and standard deviation over a number of batches. In all experiments, models are finetuned using LoRA [20]. Experimental settings and hyperparameters are detailed in Appendix F.

**I. MPS + contrast, saturation, sharpness constraints.** A common shortcoming of several off-the-shelf aesthetics, image preference and quality reward models is their tendency to overfit to certain image characteristics such as saturation, and sharp high-contrast textures. See, for example, images in the first column in Figure 3 (right). In order to mitigate this issue, we add regularizers to the reward to explicitly penalize these characteristics. However, if the regularization weight is not carefully set, models fit these regularizers rather than the reward. As shown in Figure 3, when using equal weights the MPS reward *decreases* (left plot). In contrast, our constrained approach can effectively control multiple undesired artifacts while ensuring none of the rewards are neglected by obtaining a near feasible solution for the specified constraint level, which represents a 50% improvement.

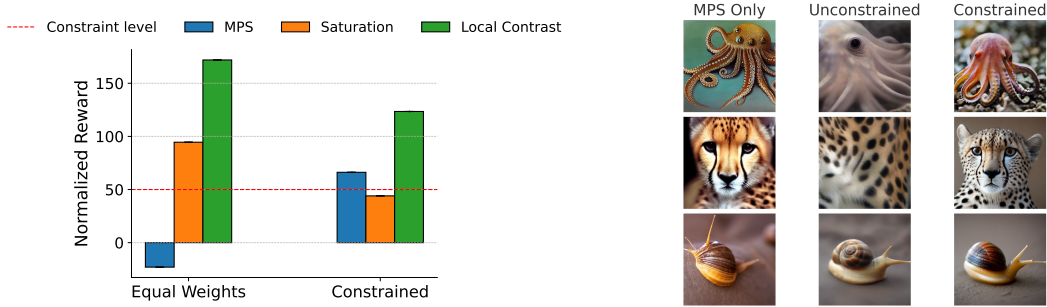


Figure 3: We finetune stable diffusion with one reward emphasizing aesthetic quality (MPS), and Saturation and Local Contrast regularizers. Left: Value of the rewards after finetuning. Right: Images sampled from the aligned models, and the model trained solely with MPS [58] reward for comparison.

**II. Multiple aesthetic constraints.** When finetuning with multiple rewards, arbitrarily setting fixed weights can lead to disparate performance among them. This can be observed in Figure 4 (left plot), where the model overfits to one reward while neglecting the more challenging reward (hps). In contrast, constraining all rewards allows the model to improve all rewards by the desired constraint level, including challenging ones (hps). As pictured in Figure 4, minimizing the KL subject to constraints also results in lower KL to the pre-trained model (middle plot). Without constraint, due to reward overfitting the finetuned model diverges too far from the pretrained model which is undesirable (right plot).

### 5.2 Product composition of diffusion models

In high-dimensional settings like image generation, using MCMC to get samples from the true product distribution and then minimizing the Lagrangian via (19) to find the true product score function is prohibitively costly. To circumvent this, we use a surrogate for the true score both for sampling and computing the KL, as detailed in Appendix F.

**I. Composing models fine-tuned on different rewards.** We investigate the composition of multiple finetuned versions of the same base model, where each one fit LoRA adapters a different reward

<sup>1</sup><https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>



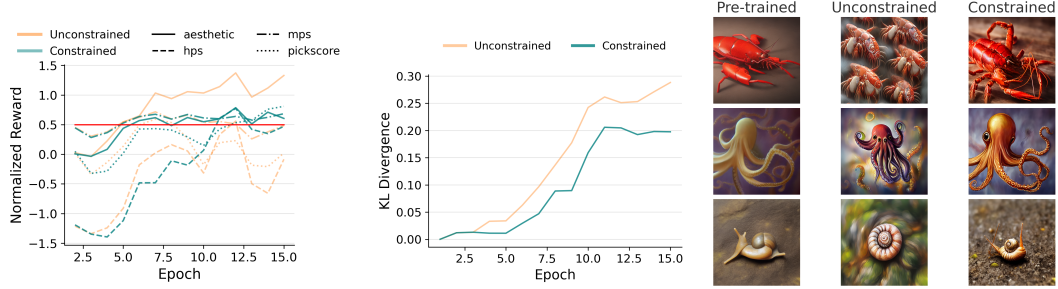


Figure 4: Finetuning with multiple image quality/aesthetic rewards. Left: Reward trajectories in training. Middle: KL to pre-trained model constrained. Right: Images sampled from the aligned models and the pre-trained model for reference.

function. A key challenge lies in selecting appropriate weights for this combination. Arbitrary choices may lead to undesirable trade-offs and under-representation of certain models in the mixture as evidenced in Figure 5 by drops in up to 30% in some rewards. Our constrained approach gives us a way to find the weights that keep us close to each individual model, leading to higher rewards for all models.

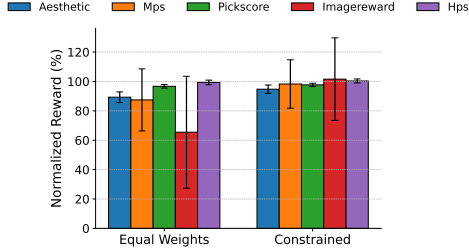


Figure 5: Composition of adapters finetuned for different rewards, for an equal weighted and product mixture. 100 represents the reward attained by the model trained with each individual reward. Higher is better.

	Min. CLIP ( $\uparrow$ )	Min. BLIP ( $\uparrow$ )
Combined Prompting	22.1	0.204
Equal Weights	22.7	0.252
Constrained (Ours)	<b>22.9</b>	<b>0.268</b>

Table 1: Comparing constrained approach to baselines on minimum CLIP and BLIP scores. The scores are averaged over 50 different prompt pairs sampled from a list of simple prompts.

**II. Concept composition with stable diffusion.** Following the setting in [38], we compose two text-to-image diffusion models conditioned on different inputs. We use constrained learning (SR-C) to find the optimal weights to compose these two models. We compare to the baseline of using equal weights for the composition. We can see that closeness to each model also encourages the representation of the concept in the images generated by the composed model as reflected by the improved text-to-image similarity metrics CLIP [19] and BLIP [24] scores in Table 1. We compute the similarity score between the generated images and each of the two prompts and compare the minimums. We also compare to the baseline of generating images from a combined prompt that includes both prompts. Images generated with each approach along with implementation details and more experimental results can be found in Appendix F.

## 6 Conclusions

We have developed a constrained optimization framework that unifies alignment and composition of diffusion models by enforcing that the aligned model satisfies reward constraints and/or remains close to each pre-trained model. We provide a theoretical characterization of the solutions to the constrained alignment and composition problems, and develop Lagrangian-based primal-dual training algorithms to approximate these solutions. Empirically, we demonstrate our constrained approach in image generation, applying it to alignment and composition, and show that our aligned or composed model satisfies constraints, effectively.

## References

- [1] B. Biggs, A. Seshadri, Y. Zou, A. Jain, A. Golatkar, Y. Xie, A. Achille, A. Swaminathan, and S. Soatto. Diffusion soup: Model merging for text-to-image diffusion models. *arXiv preprint arXiv:2406.08431*, 2024.
- [2] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.
- [3] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.
- [4] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [5] A. Bradley and P. Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.
- [6] L. F. Chamon, S. Paternain, M. Calvo-Fullana, and A. Ribeiro. Constrained learning with non-convex losses. *IEEE Transactions on Information Theory*, 69(3):1739–1760, 2022.
- [7] L. F. O. Chamon and A. Ribeiro. Probably approximately correct constrained learning, 2021.
- [8] J. Chen, R. Zhang, Y. Zhou, and C. Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *The Twelfth International Conference on Learning Representations*, 2024.
- [9] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.
- [10] M. Chidambaram, K. Gatmiry, S. Chen, H. Lee, and J. Lu. What does guidance do? a fine-grained analysis in a simple setting. *arXiv preprint arXiv:2409.13074*, 2024.
- [11] J. K. Christopher, S. Baek, and N. Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.
- [12] K. Clark, P. Vicol, K. Swersky, and D. J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024.
- [13] C. Domingo-Enrich, M. Drozdal, B. Karrer, and R. T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control, 2025.
- [14] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc, 2024.
- [15] Y. Fan and K. Lee. Optimizing DDPM sampling with shortcut fine-tuning. In *International Conference on Machine Learning*, pages 9623–9639. PMLR, 2023.
- [16] Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023.
- [17] G. Giannone, A. Srivastava, O. Winther, and F. Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. *Advances in Neural Information Processing Systems*, 36:51830–51861, 2023.
- [18] Y. Han, M. Razaviyayn, and R. Xu. Stochastic control for fine-tuning diffusion models: Optimality, regularity, and convergence. *arXiv preprint arXiv:2412.18164*, 2024.

- [19] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.
- [20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- [21] S. Khalafi, D. Ding, and A. Ribeiro. Constrained diffusion models via dual training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [22] Y. Kirstain, A. Polyak, U. Singer, S. Matiana, J. Penna, and O. 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.
- [23] K. Lee, H. Liu, M. Ryu, O. Watkins, Y. Du, C. Boutilier, P. Abbeel, M. Ghavamzadeh, and S. S. Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.
- [24] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.
- [25] S. Li, K. Kallidromitis, A. Gokul, Y. Kato, and K. Kozuka. Aligning diffusion models by optimizing human utility. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [26] J. Liang, J. K. Christopher, S. Koenig, and F. Fioretto. Multi-agent path finding in continuous spaces with projected diffusion models. *arXiv preprint arXiv:2412.17993*, 2024.
- [27] J. Liang, J. K. Christopher, S. Koenig, and F. Fioretto. Simultaneous multi-robot motion planning with projected diffusion models. *arXiv preprint arXiv:2502.03607*, 2025.
- [28] B. Liu, S. Shao, B. Li, L. Bai, Z. Xu, H. Xiong, J. Kwok, S. Helal, and Z. Xie. Alignment of diffusion models: Fundamentals, challenges, and future. *arXiv preprint arXiv:2409.07253*, 2024.
- [29] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022.
- [30] S. Lyu. Interpretation and generalization of score matching, 2012.
- [31] W. Mou, N. Flammarion, M. J. Wainwright, and P. L. Bartlett. Improved bounds for discretization of langevin diffusions: Near-optimal rates without convexity, 2019.
- [32] S. S. Narasimhan, S. Agarwal, L. Rout, S. Shakkottai, and S. P. Chinchali. Constrained posterior sampling: Time series generation with hard constraints. *arXiv preprint arXiv:2410.12652*, 2024.
- [33] M. Prabhudesai, A. Goyal, D. Pathak, and K. Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2024.
- [34] M. Prabhudesai, R. Mendonca, Z. Qin, K. Fragkiadaki, and D. Pathak. Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.
- [35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022.
- [36] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
- [37] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

- [38] M. Skreta, L. Atanackovic, J. Bose, A. Tong, and K. Neklyudov. The superposition of diffusion models using the itô density estimator. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [39] M. Sohrabi, J. Ramirez, T. H. Zhang, S. Lacoste-Julien, and J. Gallego-Posada. On pi controllers for updating lagrange multipliers in constrained optimization. In *International Conference on Machine Learning*, pages 45922–45954. PMLR, 2024.
- [40] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models, 2022.
- [41] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, 2021.
- [42] M. Uehara, Y. Zhao, T. Biancalani, and S. Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024.
- [43] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, T. Biancalani, and S. Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.
- [44] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, S. Levine, and T. Biancalani. Feedback efficient online fine-tuning of diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.
- [45] M. Uehara, Y. Zhao, E. Hajiramezanali, G. Scalia, G. Eraslan, A. Lal, S. Levine, and T. Biancalani. Bridging model-based optimization and generative modeling via conservative fine-tuning of diffusion models. *Advances in Neural Information Processing Systems*, 37:127511–127535, 2024.
- [46] A. Ulhaq and N. Akhtar. Efficient diffusion models for vision: A survey. *arXiv preprint arXiv:2210.09292*, 2022.
- [47] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.
- [48] L. Wang, C. Song, Z. Liu, Y. Rong, Q. Liu, and S. Wu. Diffusion models for molecules: A survey of methods and tasks. *arXiv preprint arXiv:2502.09511*, 2025.
- [49] X. Wu, Y. Hao, M. Zhang, K. Sun, Z. Huang, G. Song, Y. Liu, and H. Li. Deep reward supervisions for tuning text-to-image diffusion models. In *European Conference on Computer Vision*, pages 108–124, 2024.
- [50] X. Wu, K. Sun, F. Zhu, R. Zhao, and H. Li. Better aligning text-to-image models with human preference. *arXiv preprint arXiv:2303.14420*, 1(3), 2023.
- [51] X. Wu, K. Sun, F. Zhu, R. Zhao, and H. Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2096–2105, 2023.
- [52] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.
- [53] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2023.
- [54] J. N. Yan, J. Gu, and A. M. Rush. Diffusion models without attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8239–8249, 2024.
- [55] K. Yang, J. Tao, J. Lyu, C. Ge, J. Chen, W. Shen, X. Zhu, and X. Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8941–8951, 2024.

- 443 [56] S. Zampini, J. Christopher, L. Oneto, D. Anguita, and F. Fioretto. Training-free constrained  
444 generation with stable diffusion models. *arXiv preprint arXiv:2502.05625*, 2025.
- 445 [57] H. Zhang and T. Xu. Towards controllable diffusion models via reward-guided exploration,  
446 2023.
- 447 [58] S. Zhang, B. Wang, J. Wu, Y. Li, T. Gao, D. Zhang, and Z. Wang. Learning multi-dimensional  
448 human preference for text-to-image generation. In *Proceedings of the IEEE/CVF Conference*  
449 *on Computer Vision and Pattern Recognition*, pages 8018–8027, 2024.
- 450 [59] Z. Zhang, L. Shen, S. Zhang, D. Ye, Y. Luo, M. Shi, B. Du, and D. Tao. Aligning few-step  
451 diffusion models with dense reward difference learning. *arXiv preprint arXiv:2411.11727*,  
452 2024.
- 453 [60] H. Zhao, H. Chen, J. Zhang, D. D. Yao, and W. Tang. Scores as actions: a framework  
454 of fine-tuning diffusion models by continuous-time reinforcement learning. *arXiv preprint*  
455 *arXiv:2409.08400*, 2024.

## NeurIPS Paper Checklist

### 1. Claims

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

Answer: [\[Yes\]](#)

Justification: There are references in the introduction to sections of the paper where we discuss in depth the claims made.

Guidelines:

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

### 2. Limitations

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

Answer: [\[Yes\]](#)

Justification: We discuss the limitations briefly in the conclusion, and more thoroughly in Appendix A.

Guidelines:

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

### 3. Theory assumptions and proofs

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



Answer: [Yes]

Justification: Yes, the full proofs for the theoretical results are provided in Appendix C.

Guidelines:

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

#### 4. Experimental result reproducibility

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

Answer: [Yes]

Justification: We provide details for reproducing the experiments in the paper in Appendix F. We will also provide the code used for all of the experiments upon the paper's publication.

Guidelines:

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

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

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

Answer: [Yes]

Justification: We will make public the repository with our code and implementations used for all of the experiments upon the paper's publication and include a link to it in the paper. Instructions and implementation details are provided in Appendix F. Anonymized code for implementing some of the experiments will also be provided with the supplementary material.

Guidelines:

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

## 6. Experimental setting/details

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

Answer: [Yes]

Justification: These details are provided in Appendix F.

Guidelines:

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

## 7. Experiment statistical significance

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

Answer: [Yes]

Justification: Most of the plots in the main paper include error bars. Some don't for visual clarity. More details on statistical significance of the results are provided in Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

## 8. Experiments compute resources

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

Answer: [Yes]

Justification: We ran the experiments on a system with 2 NVIDIA RTX A6000 GPUs with 48 GB of GPU memory each. More details can be found in Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

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

Answer: [Yes]

Justification: Yes.

Guidelines:

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

## 10. Broader impacts

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

Answer: [Yes]

Justification: These impacts are discussed in Appendix A.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

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

## 11. Safeguards

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

Answer: [NA]

Justification: We use publicly available pretrained models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

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

Answer: [Yes]

Justification: The models and code used have been properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

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

Answer: [NA]

Justification: -

Guidelines:

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

### 14. Crowdsourcing and research with human subjects

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

Answer: [NA]

Justification: -

Guidelines:

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

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

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

Answer: [NA]

Justification: -

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- 766           • We recognize that the procedures for this may vary significantly between institutions  
767           and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the  
768           guidelines for their institution.  
769           • For initial submissions, do not include any information that would break anonymity (if  
770           applicable), such as the institution conducting the review.

771 **16. Declaration of LLM usage**

772       Question: Does the paper describe the usage of LLMs if it is an important, original, or  
773       non-standard component of the core methods in this research? Note that if the LLM is used  
774       only for writing, editing, or formatting purposes and does not impact the core methodology,  
775       scientific rigorousness, or originality of the research, declaration is not required.

776       Answer: [NA]

777       Justification: -

778       Guidelines:

- 779           • The answer NA means that the core method development in this research does not  
780           involve LLMs as any important, original, or non-standard components.  
781           • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)  
782           for what should or should not be described.



783  
784

# Supplementary Materials for “Composition and Alignment of Diffusion Models using Constrained Learning”

785

## A Limitations and Broader Impact

786 **Limitations:** Despite offering a unified constrained learning framework and demonstrating strong  
787 empirical results, further experiments are needed to assess our method’s effectiveness on alignment  
788 and composition tasks beyond image generation, under mixed alignment and composition constraints,  
789 and in combination with inference-time techniques. Additionally, further theoretical work is needed  
790 to understand optimality of non-convex constrained optimization, convergence and sample complexity  
791 of primal-dual training algorithms.

792 **Broader impact:** Our method can enhance diffusion models’ compliance with diverse requirements,  
793 such as realism, safety, fairness, and transparency. By introducing a unified constrained learning  
794 framework, our work offers practical guidance for developing more reliable and responsible diffusion  
795 model training algorithms, with potential impact across applications such as content generation,  
796 robotic control, and scientific discovery.

797

## B Related Work

798 **Alignment of diffusion models.** Our constrained alignment is related to a line of work on fine-tuning  
799 diffusion models. Standard fine-tuning typically involves optimizing either a task-specific reward  
800 that encodes desired properties, or a weighted sum of this reward and a regularization term that  
801 encourages closeness to the pre-trained model; see [15, 53, 23, 51, 57, 49, 2, 12, 59] for studies using  
802 the single reward objective and [43, 60, 45, 44, 34, 16, 18] for those using the weighted sum objective.  
803 The former class of single reward-based studies focus exclusively on generating samples with higher  
804 rewards, often at the cost of generalization beyond the training data. The latter class introduces a  
805 regularization term that regulates the model to be close to the pre-trained one, while leaving the  
806 trade-off between reward and closeness unspecified; see [42] for their typical pros and cons in  
807 practice. There are three key drawbacks to using either the single reward or weighted sum objective:  
808 (i) the trade-off between reward maximization and leveraging the utility of the pre-trained model is  
809 often chosen heuristically; (ii) it is unclear whether the reward satisfies the intended constraints; and  
810 (iii) multiple constraints are not naturally encoded within a single reward function. In contrast, we  
811 formulate alignment as a constrained learning problem: minimizing deviation from the pre-trained  
812 model subject to reward constraints. This offers a more principled alternative to existing ad hoc  
813 approaches [8, 17]. Our new alignment formulation (i) offers a theoretical guarantee of an optimal  
814 trade-off between reward satisfaction and proximity to the pre-trained model, and (ii) allows for  
815 the direct imposition of multiple reward constraints. We also remark that our constrained learning  
816 approach generalizes to fine-tuning of diffusion models with preference [47, 55, 25].

817 **Composition of diffusion models.** Our constrained composition approach is related to prior work on  
818 compositional generation with diffusion models. When composing pre-trained diffusion models, two  
819 widely used approaches are (i) product composition (or conjunction) and (ii) mixture composition  
820 (or disjunction). In product composition, it has been observed that the diffusion process is not  
821 compositional, e.g., a weighted sum of diffusion models does not generate samples from the product  
822 of the individual target distributions [14, 5, 10]. To address this issue, the weighted sum approach  
823 has been shown to be effective when combined with additional assumptions or techniques, such as  
824 energy-based models [29, 14], MCMC sampling [14], diffusion soup [1], and superposition [38].  
825 However, how to determine optimal weights for the individual models is not yet fully understood.  
826 In contrast, we propose a constrained optimization framework for composing diffusion models that  
827 explicitly determines the optimal composition weights. Hence, this formulation enables an optimal  
828 trade-off among the pre-trained diffusion models. Moreover, our constrained composition approach  
829 also generalizes to mixture composition, offering advantages over prior work [29, 14, 1, 38].

**Diffusion models under constraints.** Our work is pertinent to a line of research that incorporates constraints into diffusion models. To ensure that generated samples satisfy given constraints, several ad hoc approaches have proposed that train diffusion models under hard constraints, e.g., projected diffusion models [26, 11, 27], constrained posterior sampling [32], and proximal Langevin dynamics [56]. In contrast, our constrained alignment approach focuses on expected constraints defined via reward functions and provides optimality guarantees through duality theory. A more closely related work considers constrained diffusion models with expected constraints, focusing on mixture composition [21]. In comparison, we develop new constrained diffusion models for reward alignment and product composition.

## C Proofs

For conciseness, wherever it is clear from the context we omit the time subscript:

$$D_{\text{KL}}(p_{0:T}(x_{0:T}; s_p)) = D_{\text{KL}}(p(x_{0:T}; s_p)) \quad (21)$$

### C.1 Proof of Lemma 1

*Proof.* The DDIM process is Markovian in reverse time with the conditional likelihoods given by

$$p(x_{t-1}|x_t; s) = \mathcal{N}\left(\sqrt{\frac{\alpha_{t-1}}{\alpha_t}}x_t + \beta_t s(x_t, t), \sigma_t^2 I\right) \quad (22)$$

Using (22) we expand the path-wise KL:

$$\begin{aligned} & D_{\text{KL}}(p_{0:T}(\cdot; s_p) \parallel p_{0:T}(\cdot; s_q)) \\ &= \mathbb{E}_{x_{0:T} \sim p} [\log p(x_{0:T}; s_p) - \log p(x_{0:T}; s_q)] \\ &\stackrel{(a)}{=} \mathbb{E}_{x_T \sim p_{T+1}(\cdot), x_{T-1} \sim p_T(\cdot | x_T), \dots, x_0 \sim p_1(\cdot | x_1)} \left[ \sum_{t=T}^1 \log \frac{p(x_{t-1} | x_t; s_p)}{p(x_{t-1} | x_t; s_q)} \right] \\ &\stackrel{(b)}{=} \sum_{t=T}^1 \mathbb{E}_{x_T \sim p_{T+1}(\cdot), x_{T-1} \sim p_T(\cdot | x_T), \dots, x_0 \sim p_1(\cdot | x_1)} \left[ \log \frac{p(x_{t-1} | x_t; s_p)}{p(x_{t-1} | x_t; s_q)} \right] \\ &\stackrel{(c)}{=} \sum_{t=T}^1 \mathbb{E}_{x_{0:T} \sim p} [D_{\text{KL}}(p(x_{t-1} | x_t; s_p) \parallel p(x_{t-1} | x_t; s_q))] \\ &\stackrel{(d)}{=} \sum_{t=T}^1 \mathbb{E}_{x_t \sim p_{t+1}} \left[ \frac{\beta_t^2}{2\sigma_t^2} \|s_p(x_t, t) - s_q(x_t, t)\|^2 \right] \\ &\stackrel{(e)}{=} \sum_{t=T}^1 \mathbb{E}_{\{p_t\}} \left[ \frac{\beta_t^2}{2\sigma_t^2} \|s_p(x_t, t) - s_q(x_t, t)\|^2 \right] \end{aligned}$$

where (a) is due to the diffusion process, (b) is due to the exchangeable sum and integration, (c) is the definition of reverse KL divergence at time  $t$ , (d) is due to the reverse KL divergence between two Gaussians with the same covariance and means differing by  $\beta_t(s_p(x_t, t) - s_q(x_t, t))$ , and in (e) we abbreviate  $\mathbb{E}_{x_t \sim p_{t+1}}$  as  $\mathbb{E}_{\{p_t\}}$  that is taken over the randomness of Markov process.  $\square$

### C.2 Proof of Lemma 2

**Proof Roadmap:** The proof for Lemma 2 is quite involved, thus we have divided the proof into multiple parts for readability.

- We begin by giving a few definitions for continuous time diffusion processes.
- Then in Lemma 4 we characterize how the KL between the marginals of two processes changes over time.
- Using Lemma 4 we prove Lemma 5 which is the analogue of Lemma 2 in continuous time.

855 • Next, Lemmas 6, 7, 8, allow us to bound the discretization error  $\epsilon_T$  incurred when going  
 856 from continuous time processes to corresponding discretized processes and complete the  
 857 proof.

858 **Notation Guide:** In this Section(C.2) we will be dealing with continuous time forward and reverse  
 859 diffusion processes and their discretized counterparts.

- 860 • We denote the continuous time variable  $\tau \in [0, 1]$  to differentiate it from the discrete time  
 861 indices  $t \in \{0, \dots, T\}$ .  $t = 0$  corresponds to  $\tau = 1$  and  $t = T$  corresponds to  $\tau = 0$ .<sup>2</sup>
- 862 • We denote as  $\mathfrak{X}_\tau$  the continuous time reverse DDIM process and  $X_t$  as the corresponding  
 863 discrete time process.
- 864 • The forward processes we denote with an additional bar e.g.  $\bar{\mathfrak{X}}_\tau, \bar{X}_t$  denote the continuous  
 865 time and discrete time forward processes respectively.
- 866 • Marginal density of continuous time DDIM process with score predictor  $s(x, \tau)$  at time  $\tau$  we  
 867 denote as:  $\mathbf{p}_\tau(x, s)$

868 **Continuous time Preliminaries.** Given a function  $s(x, \tau) : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$ , and a noise schedule  
 869  $\bar{\alpha}_\tau$  increasing from  $\bar{\alpha}_0 = 0$  to  $\bar{\alpha}_1 = 1$ , we define a continuous time reverse DDIM process as:

$$d\mathfrak{X}_\tau = \left( \frac{\dot{\bar{\alpha}}_\tau}{2\bar{\alpha}_\tau} \mathfrak{X}_\tau + \left( \frac{\dot{\bar{\alpha}}_\tau}{2\bar{\alpha}_\tau} + \frac{\sigma_\tau^2}{2} \right) s(\mathfrak{X}_\tau, \tau) \right) dt + \sigma_\tau d\mathfrak{B}_\tau, \quad \mathfrak{X}_0 \sim \mathcal{N}(0, I) \quad (23)$$

870 The variance schedule  $\sigma_\tau$  is arbitrary and determines the randomness of the trajectories (e.g. if  
 871  $\sigma_\tau = 0$  for all  $\tau$ , then the trajectories will be deterministic). The DDIM generative process (23)  
 872 induces marginal densities  $\mathbf{p}_\tau(x, s)$  for  $\tau \in [0, 1]$

873 For reference the Discrete time DDIM process defined in the main paper is:

$$X_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} X_t + \beta_t s(X_t, t) + \sigma_t \epsilon_t \quad (24)$$

874 Up to first order approximation, the discrete time process (24) is the Euler-Maruyama discretization  
 875 of the continuous time process (23). A uniform discretization of time is assumed i.e.  $\tau = 1 - \frac{t}{T}$   
 876 (See [13] Appendix B.1 for the full derivation).

877 Given random variables  $\tilde{\mathfrak{X}}_0 \sim \bar{\mathbf{p}}_0 = \mathcal{N}(0, I)$  and  $\tilde{\mathfrak{X}}_1 \sim \bar{\mathbf{p}}_1$ , where  $\bar{\mathbf{p}}_1$  is some probability distribution  
 878 (e.g. the data distribution), we define a reference flow  $\tilde{\mathfrak{X}}_\tau$  for  $\tau \in [0, 1]$  as:

$$\tilde{\mathfrak{X}}_\tau = \alpha_\tau \tilde{\mathfrak{X}}_0 + \zeta_\tau \tilde{\mathfrak{X}}_1 \quad (25)$$

879 Note that there is no specific process implied by the definition above, since different processes can  
 880 have the same marginal densities as the reference flow at all times  $\tau$ . We denote by  $\bar{\mathbf{p}}_\tau(\cdot)$  the density  
 881 of  $\tilde{\mathfrak{X}}_\tau$ . As  $\alpha_\tau$  decreases from  $\alpha_0 = 1$  to  $\alpha_1 = 0$ , and  $\zeta_\tau$  increases from  $\zeta_0 = 0$  to  $\zeta_1 = 1$  the reference  
 882 flow gives an interpolation between  $\bar{\mathbf{p}}_0 = \mathcal{N}(0, I)$  and  $\bar{\mathbf{p}}_1$ .

883 If the score predictor  $s(x, \tau) = \nabla_x \log \bar{\mathbf{p}}_\tau(x)$ , then the DDIM process (23) has the same marginals as  
 884 the reference flow (25) i.e.  $\mathbf{p}_\tau(x, s) = \bar{\mathbf{p}}_\tau(x)$  for  $\tau \in [0, 1]$ . This is assuming proper choice of  $\alpha_\tau, \zeta_\tau$   
 885 i.e.  $\alpha_\tau = \sqrt{1 - \bar{\alpha}_\tau}, \zeta_\tau = \sqrt{\bar{\alpha}_\tau}$ .

886 The following Lemma which generalizes Theorem 1 from [30], characterizes how the KL between  
 887 marginals of two continuous time forward processes changes with time.

888 **Lemma 4.** Consider reference flows defined as  $\tilde{\mathfrak{X}}_\tau = \alpha_\tau \tilde{\mathfrak{X}}_0 + \zeta_\tau \tilde{\mathfrak{X}}_1$ , for  $\tau \in [0, 1]$  where  $\tilde{\mathfrak{X}}_0 \sim$   
 889  $\mathcal{N}(0, I)$ . Denote by  $\bar{\mathbf{p}}_\tau(\cdot)$ , the marginal density of  $\tilde{\mathfrak{X}}_\tau$  when  $\tilde{\mathfrak{X}}_1 \sim \bar{\mathbf{p}}_1$  and similarly  $\bar{\mathbf{q}}_\tau(\cdot)$ , the marginal  
 890 density of  $\tilde{\mathfrak{X}}_\tau$  when  $\tilde{\mathfrak{X}}_1 \sim \bar{\mathbf{q}}_1$ . The following then holds:

$$\frac{d}{d\tau} D_{\text{KL}}(\bar{\mathbf{p}}_\tau(\cdot) || \bar{\mathbf{q}}_\tau(\cdot)) = -\gamma_\tau \dot{\gamma}_\tau D_F(\bar{\mathbf{p}}_\tau(\cdot) || \bar{\mathbf{q}}_\tau(\cdot)) \quad (26)$$

891 where  $\gamma_\tau = \zeta_\tau / \alpha_\tau$ , and  $D_F(p || q)$  denotes the Fisher divergence.

<sup>2</sup>For consistency with other works from whom we will utilize some results in our proofs, namely [13, 30], the direction of time we consider in continuous time is reversed compared to discrete time. This does not affect any of our derivations and results beyond a small change of notation.

892 *Proof.* We start by defining  $\bar{\mathfrak{Y}}_\tau$  as a time-dependent scaling of  $\bar{\mathfrak{X}}_\tau$ :

$$\bar{\mathfrak{Y}}_\tau := \frac{1}{\alpha_\tau} \bar{\mathfrak{X}}_\tau = \bar{\mathfrak{X}}_1 + \gamma_\tau \bar{\mathfrak{X}}_0 \quad (27)$$

893 where  $\gamma_\tau := \zeta_\tau / \alpha_\tau$ . Denote by  $\tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$ , the marginal density of  $\bar{\mathfrak{Y}}_\tau$  when  $\bar{\mathfrak{X}}_1 \sim \mathfrak{p}_1$  and similarly  
 894  $\tilde{q}_t(\bar{\mathfrak{Y}}_\tau)$ , the marginal density of  $\bar{\mathfrak{Y}}_\tau$  when  $\bar{\mathfrak{X}}_1 \sim \mathfrak{q}_1$ . Now we generalize Theorem 1 from [30] to  
 895 show that (26) holds for  $\tilde{\mathfrak{p}}_\tau, \tilde{\mathfrak{q}}_\tau$ . Their Theorem is for the specific case of  $\gamma_\tau = \sqrt{1-t}$ .<sup>3</sup>

896 We now present Lemmas 4.1 and 4.2 which we will need in the remainder of the proof.

897 **Lemma 4.1.** *For density  $\tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$  as defined in Theorem 1, the following identity holds:*

$$\frac{d}{dt} \tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) = \gamma_\tau \dot{\gamma}_\tau \Delta_{\bar{\mathfrak{Y}}_\tau} \tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau). \quad (28)$$

898 *Proof.* Proof of Lemma 4.1. We start with  $\tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$  which is the convolution of a Gaussian distribution  
 899 with  $\mathfrak{p}_1(\bar{\mathfrak{X}}_1)$ :

$$\tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) = \int_{\bar{\mathfrak{X}}_1} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1), \quad (29)$$

900 Taking the derivative we have:

$$\begin{aligned} \frac{d}{dt} \tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) &= \int_{\bar{\mathfrak{X}}_1} \frac{\dot{\gamma}_\tau \|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{\gamma_\tau^3} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1) \\ &\quad - \int_{\bar{\mathfrak{X}}_1} \frac{d}{\gamma_\tau} \frac{\dot{\gamma}_\tau}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1). \end{aligned} \quad (30)$$

901 On the other hand, taking the gradient of  $\tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$  with respect to  $\bar{\mathfrak{Y}}_\tau$  we get:

$$\nabla_{\bar{\mathfrak{Y}}_\tau} \tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) = - \int_{\bar{\mathfrak{X}}_1} \frac{\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1}{\gamma_\tau^2} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1). \quad (31)$$

902 Taking the divergence of the gradient, we have:

$$\begin{aligned} \Delta_{\bar{\mathfrak{Y}}_\tau} \tilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) &= \int_{\bar{\mathfrak{X}}_1} \frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{\gamma_\tau^4} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1), \\ &\quad - \int_{\bar{\mathfrak{X}}_1} \frac{d}{\gamma_\tau^2} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1). \end{aligned} \quad (32)$$

903 Comparing Equations (30) and (32) proves the result.

904 □

905 **Lemma 4.2.** *For any positive valued function  $f(x) : \mathbb{R}^d \rightarrow \mathbb{R}$  whose gradient  $\nabla_x f$  and Laplacian*  
 906  *$\Delta_x f$  are well defined, we have the identity*

$$\frac{\Delta_x f(x)}{f(x)} = \Delta_x \log f(x) + \|\nabla_x \log f(x)\|^2. \quad (33)$$

---

<sup>3</sup>Just to avoid any confusion, in [30], at  $t = 0$  we have the data distribution and as  $t$  increases the distributions converge to Gaussians. However in the current paper, the direction of time is the opposite, meaning  $t = 0$  corresponds to the pure Gaussians and at  $t = 1$  we have the data distributions.

907 We now continue with the proof of Lemma 4. We start with the definition of Fisher divergence for  
 908 generic distributions  $p, q$ :

$$\begin{aligned}
 D_F(p \| q) &= \int_{\mathbb{R}^d} p(x) \|\nabla \log p(x) - \nabla \log q(x)\|^2 dx \\
 &= \int_{\mathbb{R}^d} p(x) \left\| \frac{\nabla p(x)}{p(x)} - \frac{\nabla q(x)}{q(x)} \right\|^2 dx \\
 &= \int_{\mathbb{R}^d} p(x) \left( \left\| \frac{\nabla p(x)}{p(x)} \right\|^2 + \left\| \frac{\nabla q(x)}{q(x)} \right\|^2 - 2 \frac{\nabla p(x)^\top \nabla q(x)}{p(x)q(x)} \right) dx
 \end{aligned} \tag{34}$$

909 We apply integration by parts to the third term. For any open bounded subset  $\Omega$  of  $\mathbb{R}^d$  with a piecewise  
 910 smooth boundary  $\Gamma = \partial\Omega$ :

$$\begin{aligned}
 \int_{x \in \Omega} \nabla p(x)^\top \frac{\nabla q(x)}{q(x)} dx &= \int_{x \in \Omega} \nabla p(x)^\top (\nabla \log q(x)) dx \\
 &= - \int_{x \in \Omega} p(x) \Delta \log q(x) dx + \int_{\Gamma} p(x) (\nabla \log q(x)^\top \hat{n}) d\Gamma
 \end{aligned} \tag{35}$$

911 Assuming that both  $p(x)$  and  $q(x)$  are smooth and fast-decaying, the boundary term in (35) vanishes.

912 Then we can combine (34) and (35) to write:

$$D_F(p \| q) = \int_{\mathbb{R}^d} p(x) \left( \|\nabla \log p(x)\|^2 + \|\nabla \log q(x)\|^2 + 2\Delta_x \log q(x) \right) dx \tag{36}$$

913 Returning to our distributions  $\tilde{p}_\tau(\mathfrak{Y}_\tau)$  and  $\tilde{q}_\tau(\mathfrak{Y}_\tau)$  we can rewrite (36) as:

$$D_F(\tilde{p}_\tau(\cdot) \| \tilde{q}_\tau(\cdot)) = \int_{\mathfrak{Y}_\tau} \tilde{p}_\tau(\mathfrak{Y}_\tau) \left( \|\nabla \log \tilde{p}_\tau(\mathfrak{Y}_\tau)\|^2 + \|\nabla \log \tilde{q}_\tau(\mathfrak{Y}_\tau)\|^2 + 2\Delta_{\mathfrak{Y}_\tau} \log \tilde{q}_\tau(\mathfrak{Y}_\tau) \right) d\mathfrak{Y}_\tau \tag{37}$$

914 For conciseness in notation, we drop references to variables  $\mathfrak{Y}_\tau$  and  $\mathfrak{X}_1$  in the integration, the  
 915 density functions, and the operators whenever this does not lead to ambiguity. We start by applying  
 916 Lemma 4.2 to Equation (36):

$$\begin{aligned}
 D_F(\tilde{p} \| \tilde{q}) &= \int \tilde{p} (|\nabla \log \tilde{p}|^2 + |\nabla \log \tilde{q}|^2 + 2\Delta \log \tilde{q}), \\
 &= \int \tilde{p} \left( |\nabla \log \tilde{p}|^2 + \frac{\Delta \tilde{q}}{\tilde{q}} + \Delta \log \tilde{q} \right).
 \end{aligned} \tag{38}$$

917 Next, we expand the derivative of the KL divergence:

$$\frac{d}{d\tau} D_{KL}(\tilde{p} \| \tilde{q}) = \int \frac{d}{d\tau} \tilde{p} \log \frac{\tilde{p}}{\tilde{q}} + \int \tilde{p} \frac{d}{d\tau} \log \tilde{p} - \int \tilde{p} \frac{d}{d\tau} \log \tilde{q}.$$

918 We can eliminate the second term by exchanging integration and differentiation of  $\tau$ :

$$\int \tilde{p} \frac{d}{d\tau} \log \tilde{p} = \int \frac{d\tilde{p}}{d\tau} = \frac{d}{d\tau} \int \tilde{p} = 0.$$

919 As a result, there are three remaining terms in computing  $\frac{d}{d\tau} D_{KL}(\tilde{p} \| \tilde{q})$ , which we can further  
 920 substitute using Lemma 4.1, as:

$$\begin{aligned}
 \frac{d}{d\tau} D_{KL}(\tilde{p} \| \tilde{q}) &= \int \frac{d}{d\tau} \tilde{p} \log \tilde{p} - \int \frac{d}{d\tau} \tilde{p} \log \tilde{q} - \int \tilde{p} \frac{d}{d\tau} \log \tilde{q}, \\
 &= \gamma_\tau \dot{\gamma}_\tau \left( \int \Delta \tilde{p} \log \tilde{p} - \int \Delta \tilde{p} \log \tilde{q} - \int \tilde{p} \frac{\Delta \tilde{q}}{\tilde{q}} \right).
 \end{aligned} \tag{39}$$

921 Using integration by parts, the first term in (39) is changed to:

$$\int \Delta \tilde{p} \log \tilde{p} = \sum_{i=1}^d \frac{\partial \tilde{p}}{\partial y_i} \log \tilde{p}(\vec{y}) \Big|_{y_i=-\infty}^{y_i=\infty} - \int \nabla \tilde{p}^T \nabla \log \tilde{p}.$$

922 The limits in the first term become zero given the smoothness and fast decay properties of  $\tilde{p}(\vec{y})$ . The  
923 remaining term can be further simplified as:

$$\int \nabla \tilde{p}^T \nabla \log \tilde{p} = \int \tilde{p} (\nabla \log \tilde{p})^T \nabla \log \tilde{p} = \int \tilde{p} |\nabla \log \tilde{p}|^2.$$

924 The second term in (39) can be manipulated similarly, by first using integration by parts to get:

$$\int \Delta \tilde{p} \log \tilde{q} = \sum_{i=1}^d \frac{\partial \tilde{p}}{\partial y_i} \log \tilde{q} \Big|_{y_i=-\infty}^{y_i=\infty} - \int \nabla \tilde{p}^T \nabla \log \tilde{q}.$$

925 Applying integration by parts again to  $\nabla \tilde{p}^T \nabla \log \tilde{q}$ , we have:

$$\int \nabla \tilde{p}^T \nabla \log \tilde{q} = \sum_{i=1}^d \tilde{p} \frac{\partial \log \tilde{q}}{\partial y_i} \Big|_{y_i=-\infty}^{y_i=\infty} - \int \tilde{p} \Delta \log \tilde{q}.$$

926 The limits at the boundary values are all zero due to the smoothness and fast decay properties of  
927  $\tilde{p}(\vec{y})$ . Now collecting all terms, we have  $\int \tilde{p} \log \tilde{p} = -\int \tilde{p} |\nabla \log \tilde{p}|^2$  and  $\int \tilde{p} \log \tilde{q} = \int \tilde{p} \Delta \log \tilde{q}$ .  
928 Thus (39) becomes:

$$\frac{d}{d\tau} D_{KL}(\tilde{p}||\tilde{q}) = -\gamma_\tau \dot{\gamma}_\tau \int \tilde{p} \left( |\nabla \log \tilde{p}|^2 + \Delta \log \tilde{q} + \frac{\Delta \tilde{q}}{\tilde{q}} \right).$$

929 Combining with (38), this leads to the following:

$$\frac{d}{d\tau} D_{KL}(\tilde{p}_\tau||\tilde{q}_\tau) = -\gamma_\tau \dot{\gamma}_\tau D_F(\tilde{p}_\tau||\tilde{q}_\tau). \quad (40)$$

930 Again replacing  $\tilde{p}_\tau, \tilde{q}_\tau$  with the marginals of diffusion processes we get:

$$\frac{d}{d\tau} D_{KL}(\tilde{p}_\tau(\cdot)||\tilde{q}_\tau(\cdot)) = -\gamma_\tau \dot{\gamma}_\tau D_F(\tilde{p}_\tau(\cdot)||\tilde{q}_\tau(\cdot)). \quad (41)$$

931 Recall that  $\tilde{p}_\tau(\cdot)$  and  $\tilde{q}_\tau(\cdot)$  were the densities of the scaled random variable  $\tilde{\mathfrak{Y}}_\tau = \frac{1}{\alpha_\tau} \tilde{\mathfrak{X}}_\tau$ . This leads  
932 to  $\mathbf{p}_\tau(\tilde{\mathfrak{X}}_\tau) d\tilde{\mathfrak{X}}_\tau = \tilde{\mathbf{p}}_\tau(\tilde{\mathfrak{Y}}_\tau) d\tilde{\mathfrak{Y}}_\tau$ . Thus, it is straightforward to show that both KL divergence and  
933 Fisher divergence are invariant to the scaling of the underlying random variables.:

$$D_{KL}(\tilde{p}_\tau(\cdot)||\tilde{q}_\tau(\cdot)) = \int \tilde{p}_\tau(\tilde{\mathfrak{Y}}_\tau) \log \frac{\tilde{p}_\tau(\tilde{\mathfrak{Y}}_\tau)}{\tilde{q}_\tau(\tilde{\mathfrak{Y}}_\tau)} d\tilde{\mathfrak{Y}}_\tau = \int \mathbf{p}_\tau(\tilde{\mathfrak{X}}_\tau) \log \frac{\mathbf{p}_\tau(\tilde{\mathfrak{X}}_\tau)}{\mathbf{q}_\tau(\tilde{\mathfrak{X}}_\tau)} d\tilde{\mathfrak{X}}_\tau = D_{KL}(\mathbf{p}_\tau(\cdot)||\mathbf{q}_\tau(\cdot)) \quad (42)$$

$$\begin{aligned} D_F(\tilde{p}_\tau(\cdot)||\tilde{q}_\tau(\cdot)) &= \int \tilde{p}_\tau(\tilde{\mathfrak{Y}}_\tau) \left\| \frac{\nabla \tilde{p}_\tau(\tilde{\mathfrak{Y}}_\tau)}{\tilde{p}_\tau(\tilde{\mathfrak{Y}}_\tau)} - \frac{\nabla \tilde{q}_\tau(\tilde{\mathfrak{Y}}_\tau)}{\tilde{q}_\tau(\tilde{\mathfrak{Y}}_\tau)} \right\|^2 d\tilde{\mathfrak{Y}}_\tau \\ &= \int \mathbf{p}_\tau(\tilde{\mathfrak{X}}_\tau) \left\| \frac{\nabla \mathbf{p}_\tau(\tilde{\mathfrak{X}}_\tau)}{\mathbf{p}_\tau(\tilde{\mathfrak{X}}_\tau)} - \frac{\nabla \mathbf{q}_\tau(\tilde{\mathfrak{X}}_\tau)}{\mathbf{q}_\tau(\tilde{\mathfrak{X}}_\tau)} \right\|^2 d\tilde{\mathfrak{X}}_\tau = D_F(\mathbf{p}_\tau(\cdot)||\mathbf{q}_\tau(\cdot)) \end{aligned} \quad (43)$$

934 Thus we can replace the divergences in (40) with those of the non-scaled distribution, which concludes  
935 the proof.  $\square$

936 We now present the continuous-time analogue of Lemma 2 which characterizes the point-wise KL  
937 divergence of two continuous time diffusion processes:



**Lemma 5.** Consider two score predictors  $s_p(x, \tau) = \nabla_x \log \bar{p}_\tau(x)$ ,  $s_q(x, \tau) = \nabla_x \log \bar{q}_\tau(x)$ , where  $\bar{p}_\tau, \bar{q}_\tau$  are marginal densities of two reference flows, with the same noise schedule, starting from initial distributions  $\bar{p}_0$  and  $\bar{q}_0$ , respectively. Then, the point-wise KL divergence between two distributions of the samples generated by running continuous time DDIM (23) with  $s_p$  and  $s_q$  is given by

$$D_{\text{KL}}(\mathbf{p}_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_q)) = \int_{\tau=0}^1 \tilde{\omega}_\tau \mathbb{E}_{x \sim \mathbf{p}_\tau(\cdot; s_p)} \left[ \|s_p(x, \tau) - s_q(x, \tau)\|_2^2 \right] \quad (44)$$

where  $\tilde{\omega}_\tau$  is a time-dependent constant

*Proof.* We start with a direct application of Lemma 4:

$$\begin{aligned} D_{\text{KL}}(\mathbf{p}_1(\cdot) \parallel \mathbf{q}_1(\cdot)) &= D_{\text{KL}}(\mathbf{p}_0(\cdot) \parallel \mathbf{q}_0(\cdot)) - \int_{\tau=0}^1 \dot{\gamma}_\tau \gamma_\tau D_F(\tilde{\mathbf{p}}_\tau(\cdot) \parallel \tilde{\mathbf{q}}_\tau(\cdot)) d\tau \\ &= - \int_{\tau=0}^1 \dot{\gamma}_\tau \gamma_\tau \mathbb{E}_{x \sim \tilde{\mathbf{p}}_\tau} \left[ \|\nabla \log \tilde{\mathbf{p}}_\tau(x) - \nabla \log \tilde{\mathbf{q}}_\tau(x)\|_2^2 \right] d\tau \\ &= - \int_{\tau=0}^1 \dot{\gamma}_\tau \gamma_\tau \mathbb{E}_{x \sim \tilde{\mathbf{p}}_\tau} \left[ \|s_p(x, \tau) - s_q(x, \tau)\|_2^2 \right] d\tau \\ &= \int_{\tau=0}^1 \frac{\dot{\alpha}_\tau}{\alpha_\tau^3} \mathbb{E}_{x \sim \tilde{\mathbf{p}}_\tau} \left[ \|s_p(x, \tau) - s_q(x, \tau)\|_2^2 \right] d\tau \end{aligned} \quad (45)$$

In the second line we used the fact that  $\mathbf{p}_0(\cdot) = \mathbf{q}_0(\cdot) = \mathcal{N}(0, I)$ , therefore  $D_{\text{KL}}(\mathbf{p}_0(\cdot) \parallel \mathbf{q}_0(\cdot)) = 0$ . The third line follows from our definition of the score functions. Finally, in the last line we used the fact that  $\dot{\gamma}_\tau \gamma_\tau = -\frac{\dot{\alpha}_\tau}{\alpha_\tau^3}$  which follows from  $\gamma_\tau = \zeta_\tau / \alpha_\tau$  and  $\alpha_\tau^2 + \zeta_\tau^2 = 1$ :

$$\begin{aligned} \dot{\gamma}_\tau \gamma_\tau &= \frac{d}{d\tau} \left( \frac{\zeta_\tau}{\alpha_\tau} \right) \frac{\zeta_\tau}{\alpha_\tau} \\ &= \frac{\dot{\zeta}_\tau \zeta_\tau \alpha_\tau - \dot{\alpha}_\tau \zeta_\tau^2}{\alpha_\tau^3} \\ &= \frac{-\dot{\alpha}_\tau \alpha_\tau^2 - \dot{\alpha}_\tau (1 - \alpha_\tau^2)}{\alpha_\tau^3} \\ &= -\frac{\dot{\alpha}_\tau}{\alpha_\tau^3} \end{aligned} \quad (46)$$

by denoting  $\tilde{\omega}_\tau := -\frac{\dot{\alpha}_\tau}{\alpha_\tau^3}$  we conclude the proof.  $\square$

We now start bridging the gap between continuous and discrete time. First we present a result from [31]:

**Lemma 6.** The KL divergence between the marginals of the discrete time  $p_t(\cdot)$  and continuous time  $\mathbf{p}_{t/T}(\cdot)$  is bounded as:

$$D_{\text{KL}}(p_t(\cdot; s_p) \parallel \mathbf{p}_{t/T}(\cdot; s_p)) \leq \frac{c}{T^2} \quad (47)$$

where  $c$  is a constant depending on assumptions.

See [31] for the proof (Theorem 1). Next we need to characterize the sensitivity of the KL divergence to perturbations in the first and second arguments.

**Lemma 7.** Assume  $M := \max_x \left| \log \left( \frac{\mathbf{p}_0(\cdot; s_p)}{\mathbf{p}_0(\cdot; s_q)} \right) \right|$  is bounded. Then, the point-wise KL between the continuous time processes approximates the point-wise KL between the discrete time processes up to a discretization error  $\epsilon_1(T)$ :

$$|D_{\text{KL}}(\mathbf{p}_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_q)) - D_{\text{KL}}(p_0(\cdot; s_p) \parallel p_0(\cdot; s_q))| \leq \epsilon_1(T), \quad (48)$$

where  $\epsilon_1(T) = O(1/T)$ .

*Proof.* We first prove a similar relation for generic distributions  $\pi(x), \rho(x)$  and their perturbations  $\hat{\pi}(x), \hat{\rho}(x)$ ;

961 Where it is clear from the context, we omit the integration variables. Perturbing the first argument  
 962 gives us:

$$\begin{aligned}
 |D_{\text{KL}}(\hat{\pi} \parallel \rho) - D_{\text{KL}}(\pi \parallel \rho)| &= \int \hat{\pi} \log \left( \frac{\hat{\pi}}{\rho} \right) - \int \pi \log \left( \frac{\pi}{\rho} \right) + \int (\hat{\pi} \log \pi - \pi \log \pi) \\
 &= D_{\text{KL}}(\hat{\pi} \parallel \pi) + \int (\hat{\pi} - \pi) \log \left( \frac{\pi}{\rho} \right) \\
 &\leq D_{\text{KL}}(\hat{\pi} \parallel \pi) + \max \left( \left| \log \left( \frac{\pi}{\rho} \right) \right| \right) \int |\hat{\pi} - \pi| \\
 &= D_{\text{KL}}(\hat{\pi} \parallel \pi) + 2 \log M d_{\text{TV}}(\hat{\pi}, \pi)
 \end{aligned} \tag{49}$$

963 where  $\log M := \max_x \left| \log \left( \frac{\pi(x)}{\rho(x)} \right) \right|$  and  $d_{\text{TV}}$  denotes the total variation distance between distributions.  
 964 Next, perturbing the second argument we get:

$$\begin{aligned}
 |D_{\text{KL}}(\hat{\pi} \parallel \hat{\rho}) - D_{\text{KL}}(\hat{\pi} \parallel \rho)| &= \left| \int \hat{\pi} \log \left( \frac{\hat{\pi}}{\hat{\rho}} \right) - \int \hat{\pi} \log \left( \frac{\hat{\pi}}{\rho} \right) \right| \\
 &= - \int \hat{\pi} \log \left( \frac{\hat{\rho}}{\rho} \right) = - \int \hat{\pi} \log \left( 1 + \frac{\hat{\rho} - \rho}{\rho} \right) \\
 &\leq \int \hat{\pi} \frac{\hat{\rho} - \rho}{\rho} = \int \frac{\hat{\pi}}{\pi} \frac{\pi}{\rho} (\hat{\rho} - \rho) \\
 &\leq \max \left( \frac{\pi}{\rho} \right) \int |\hat{\rho} - \rho| = 2M d_{\text{TV}}(\hat{\rho}, \rho).
 \end{aligned} \tag{50}$$

965 Using (49), (50) we get:

$$\begin{aligned}
 |D_{\text{KL}}(\hat{\pi} \parallel \hat{\rho}) - D_{\text{KL}}(\pi \parallel \rho)| &\leq |D_{\text{KL}}(\hat{\pi} \parallel \hat{\rho}) - D_{\text{KL}}(\hat{\pi} \parallel \rho)| + |D_{\text{KL}}(\hat{\pi} \parallel \rho) - D_{\text{KL}}(\pi \parallel \rho)| \\
 &\leq D_{\text{KL}}(\hat{\pi} \parallel \pi) + 2M d_{\text{TV}}(\hat{\rho}, \rho) + 2 \log M d_{\text{TV}}(\hat{\pi}, \pi) \\
 &\leq D_{\text{KL}}(\hat{\pi} \parallel \pi) + 2M \sqrt{\frac{1}{2} D_{\text{KL}}(\hat{\rho} \parallel \rho)} + 2 \log M \sqrt{\frac{1}{2} D_{\text{KL}}(\hat{\pi} \parallel \pi)}
 \end{aligned} \tag{51}$$

966 where in the last line we utilized Pinsker's inequality to bound the TV distance with the square root  
 967 of the KL divergence. Now we apply (51) to diffusion models:

$$\begin{aligned}
 |D_{\text{KL}}(\mathbf{p}_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_q)) - D_{\text{KL}}(p_0(\cdot; s_p) \parallel p_0(\cdot; s_q))| &\leq D_{\text{KL}}(p_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_p)) \\
 &\quad + 2M \sqrt{\frac{1}{2} D_{\text{KL}}(p_0(\cdot; s_q) \parallel \mathbf{p}_0(\cdot; s_q))} \\
 &\quad + 2 \log M \sqrt{\frac{1}{2} D_{\text{KL}}(p_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_p))}
 \end{aligned} \tag{52}$$

968 Furthermore from Lemma 6 we know:

$$D_{\text{KL}}(p_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_p)) \leq c/T^2, \quad D_{\text{KL}}(p_0(\cdot; s_q) \parallel \mathbf{p}_0(\cdot; s_q)) \leq c/T^2 \tag{53}$$

969 Putting together (52) and (53) we get:

$$|D_{\text{KL}}(\mathbf{p}_0(\cdot; s_p) \parallel \mathbf{p}_0(\cdot; s_q)) - D_{\text{KL}}(p_0(\cdot; s_p) \parallel p_0(\cdot; s_q))| \leq \epsilon_1(T) \tag{54}$$

970 where  $\epsilon_1(T) := c/T^2 + (2M + 2 \log M) \sqrt{c/T^2}$ . The second term dominates therefore  $\epsilon_1(T) =$   
 971  $O(1/T)$  which concludes the proof.

972 □

973 **Lemma 8.** Assume  $B_1, B_2$  as defined below are finite:

$$B_1 := \sup_{x, \tau} \|s_p(x, \tau) - s_q(x, \tau)\|_2 \quad (55)$$

974

$$B_2 := \sup_{x, \tau} \left\| \frac{d}{d\tau} (s_p(x, \tau) - s_q(x, \tau)) \right\|_2 \quad (56)$$

975 Then the integral from Lemma 5 giving the point-wise KL in continuous time can be approximated  
976 with a discrete time sum as follows:

$$\left| \int_{\tau=0}^1 \tilde{\omega}_\tau \mathbb{E}_{x \sim \mathbf{p}_\tau(\cdot; s_p)} [\|s_p(x, \tau) - s_q(x, \tau)\|_2^2] - \sum_{t=0}^T \frac{1}{T} \tilde{\omega}_{t/T} \mathbb{E}_{x \sim p_t(\cdot; s_p)} [\|s_p(x, t) - s_q(x, t)\|_2^2] \right| \leq \epsilon_2(T) \quad (57)$$

977 where the discretization error is  $\epsilon_2(T) = O(1/T)$ .

978 *Proof.* There are two sources of error we need to consider. First we bound the error in approximating  
979 an integral with a sum:

$$\left| \int_{\tau=0}^1 \tilde{\omega}_\tau \mathbb{E}_{x \sim \mathbf{p}_\tau(\cdot; s_p)} [\|s_p(x, \tau) - s_q(x, \tau)\|_2^2] - \sum_{t=0}^T \frac{1}{T} \tilde{\omega}_{t/T} \mathbb{E}_{x \sim \mathbf{p}_{t/T}(\cdot; s_p)} [\|s_p(x, t) - s_q(x, t)\|_2^2] \right| \quad (58)$$

980

$$= \left| \int_{\tau=0}^1 f(\tau) d\tau - \sum_{t=0}^T f(t/T) \cdot \frac{1}{T} \right| \leq \frac{1}{T} \sup_{\tau \in [0,1]} \left| \frac{df}{d\tau} \right| \quad (59)$$

981 where we have defined  $f(\tau) := \tilde{\omega}_\tau \mathbb{E}_{x \sim \mathbf{p}_\tau(\cdot; s_p)} [\|s_p(x, \tau) - s_q(x, \tau)\|_2^2]$ . We now upper bound the  
982 supremum to show that it is finite:

$$\frac{df}{d\tau} = \frac{d}{d\tau} \left( \int \mathbf{p}_\tau(x; s_p) \|s_p(x, \tau) - s_q(x, \tau)\|_2^2 dx \right) \quad (60)$$

983

$$= \int \frac{d}{d\tau} (\mathbf{p}_\tau(x; s_p)) \|s_p(x, \tau) - s_q(x, \tau)\|_2^2 dx + \int \mathbf{p}_\tau(x; s_p) \frac{d}{d\tau} (\|s_p(x, \tau) - s_q(x, \tau)\|_2^2) dx \quad (61)$$

984 We bound each term in (61) separately. Then the first term in (61) is bounded because  $\frac{d}{d\tau} (\mathbf{p}_\tau(x; s_p))$   
985 is finite as characterized in Lemma 4.1. The second term in (61) we expand further:

$$\int \mathbf{p}_\tau(x; s_p) \frac{d}{d\tau} (\|s_p(x, \tau) - s_q(x, \tau)\|_2^2) dx = \int 2\mathbf{p}_\tau(x; s_p) \langle s_p(x, \tau) - s_q(x, \tau), \frac{ds_p(x, \tau)}{d\tau} - \frac{ds_q(x, \tau)}{d\tau} \rangle dx \quad (62)$$

986

$$\leq 2 \sup_{x, \tau} \|s_p(x, \tau) - s_q(x, \tau)\|_2 \left\| \frac{d}{d\tau} (s_p(x, \tau) - s_q(x, \tau)) \right\|_2 \leq 2B_1 B_2 \quad (63)$$

987 The second source of error is replacing expectation over the continuous time marginal  $\mathbf{p}_{t/T}(\cdot; s_p)$   
988 with expectation over the discrete time marginal  $p_t(\cdot; s_p)$  which we can bound by using the fact that  
989 the two aforementioned marginals are close to each other.

$$\left| \sum_{t=0}^T \frac{1}{T} \tilde{\omega}_{t/T} \mathbb{E}_{x \sim \mathbf{p}_{t/T}(\cdot; s_p)} [\|s_p(x, t) - s_q(x, t)\|_2^2] - \sum_{t=0}^T \frac{1}{T} \tilde{\omega}_{t/T} \mathbb{E}_{x \sim p_t(\cdot; s_p)} [\|s_p(x, t) - s_q(x, t)\|_2^2] \right| \quad (64)$$

990

$$\leq \sum_{t=0}^T \frac{1}{T} \tilde{\omega}_{t/T} d_{TV}(p_t(\cdot; s_p), \mathbf{p}_{t/T}(\cdot; s_p)) \cdot \sup_x \|s_p(x, \tau) - s_q(x, \tau)\|_2^2 \quad (65)$$

991

$$\leq \sum_{t=0}^T \frac{1}{T} \tilde{\omega}_{t/\tau} \sqrt{\frac{c}{T^2}} \cdot B_1^2 \leq T \cdot \frac{1}{T} \cdot \sqrt{\frac{c}{T^2}} \cdot B_1^2 = O\left(\frac{1}{T}\right) \quad (66)$$

992 where we used Lemma 6 to get the last line which concludes the proof.  $\square$

993 It remains to combine Lemmas 7, 8 to complete the proof of Lemma 2:

$$D_{\text{KL}}(p_0(\cdot; s_p) \| p_0(\cdot; s_q)) = \sum_{t=0}^T \tilde{\omega}_t \mathbb{E}_{x \sim p_t(\cdot; s_p)} \left[ \|s_p(x, t) - s_q(x, t)\|_2^2 \right] + \epsilon_T \quad (67)$$

994 where  $|\epsilon_T| \leq \epsilon_1(T) + \epsilon_2(T) = O(1/T)$ . (We abuse notation to denote  $\frac{1}{T}\tilde{\omega}_t$  as  $\tilde{\omega}_t$  in (67) and in  
995 the main paper.)

### 996 C.3 Proof of Theorem 1

997 *Proof.* For any  $\lambda \geq 0$ , the optimal solution  $p^*(\cdot; \lambda)$  is uniquely determined by solving a partial  
998 minimization problem,

$$\underset{p \in \mathcal{P}}{\text{minimize}} \quad L_{\text{ALI}}(p, \lambda).$$

999 Application of Donsker and Varadhan’s variational formula yields the optimal solution

$$p^*(\cdot; \lambda) \propto q(\cdot) e^{\lambda^\top r(\cdot)}.$$

1000 Since the strong duality holds for Problem (UR-A), its optimal solution is given by  $p^*(\cdot; \lambda)$  evaluated  
1001 at  $\lambda = \lambda^*$ .

1002 It is straightforward to evaluate the dual function by the definition  $D(\lambda) = L(p^*(\cdot; \lambda), \lambda)$ .  $\square$

### 1003 C.4 Proof of Theorem 2

1004 *Proof.* We first consider the constrained alignment (SR-A) in the path space  $\{p_{0:T}(\cdot)\}$ . Since the KL  
1005 divergence is convex in the path space and the constraints are linear, the strong duality holds in the  
1006 path space, i.e., there exists a pair  $(p_{0:T}^*(\cdot), \lambda^*)$  such that

$$\bar{P}_{\text{ALI}}^* := D_{\text{KL}}(p_{0:T}^*(\cdot) \| q_{0:T}(\cdot; s_q)) = \bar{D}_{\text{ALI}}(\lambda^*) := \bar{D}_{\text{ALI}}^*.$$

1007 Equivalently,  $(p_{0:T}^*(\cdot), \lambda^*)$  is a saddle point of the Lagrangian  $L_{\text{ALI}}(p_{0:T}(\cdot), \lambda)$ ,

$$L_{\text{ALI}}(p_{0:T}^*(\cdot), \lambda) \leq L_{\text{ALI}}(p_{0:T}^*(\cdot), \lambda^*) \leq L_{\text{ALI}}(p_{0:T}(\cdot), \lambda^*) \text{ for all } p_{0:T}(\cdot) \text{ and } \lambda \geq 0.$$

1008 Since the function class  $\mathcal{S}$  is expressive enough, any path  $p_{0:T}(\cdot)$  can be represented as  $p_{0:T}(\cdot; s_p)$   
1009 with some  $s_p \in \mathcal{S}$ ; and vice versa. Thus, we can express  $p_{0:T}^*(\cdot)$  as  $p_{0:T}(\cdot; s_p^*)$  with some  $s_p^* \in \mathcal{S}$ .

1010 We also note that the dual functions  $\bar{D}_{\text{ALI}}(\lambda)$  in the path and function spaces are the same. Hence,  
1011 the dual value for (SR-A) remains to be  $\bar{D}_{\text{ALI}}(\lambda^*)$ . Thus,  $(s_p^*, \lambda^*)$  is a saddle point of the Lagrangian

$$\bar{L}_{\text{ALI}}(s_p, \lambda) := L_{\text{ALI}}(p_{0:T}(\cdot; s_p), \lambda),$$

$$\bar{L}_{\text{ALI}}(s_p^*, \lambda) \leq \bar{L}_{\text{ALI}}(s_p^*, \lambda^*) \leq \bar{L}_{\text{ALI}}(s_p, \lambda^*) \text{ for all } s_p \in \mathcal{S} \text{ and } \lambda \geq 0.$$

1013 Therefore, the strong duality holds for (SR-A) in the function space  $\mathcal{S}$ .  $\square$

### 1014 C.5 Proof of Theorem 3

1015 *Proof.* By the definition,

$$\begin{aligned} L_{\text{AND}}(p, u; \lambda) &= u + \sum_{i=1}^m \lambda_i (D_{\text{KL}}(p \| q^i) - u) \\ &= u - u\lambda^\top \mathbf{1} + \sum_{i=1}^m (\lambda_i \mathbb{E}_{x \sim p} [\log p(x)] - \lambda_i \mathbb{E}_{x \sim p} [\log q^i(x)]) \\ &= u - u\lambda^\top \mathbf{1} + \sum_{i=1}^m \lambda_i \mathbb{E}_{x \sim p} [\log p(x)] - \mathbb{E}_{x \sim p} \left[ \log \prod_{i=1}^m (q^i(x))^{\lambda_i} \right] \\ &= u - u\lambda^\top \mathbf{1} \\ &\quad + \sum_{i=1}^m \lambda_i \left( \mathbb{E}_{x \sim p} [\log p(x)] - \mathbb{E}_{x \sim p} \left[ \log \prod_{i=1}^m (q^i(x))^{\frac{\lambda_i}{1^\top \lambda}} \right] \right) \\ &= u + \sum_{i=1}^m \lambda_i (D_{\text{KL}}(p \| q_{\text{AND}}^{(\lambda)}) - u) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \end{aligned}$$

By taking  $\lambda = \lambda^*$ , we obtain a primal problem: maximize $_{p \in \mathcal{P}, u \geq 0}$   $L_{\text{AND}}(p, u; \lambda^*)$ , which solves the constrained alignment problem (UR-A) because of the strong duality. By the variational optimality, maximization of  $L_{\text{AND}}(p, u; \lambda^*)$  over  $p$  and  $u$  is at a unique maximizer,

$$p^*(\cdot; \lambda^*) \propto q_{\text{AND}}^{(\lambda^*)}(\cdot)$$

and  $u^* = 0$  if  $1 - \mathbf{1}^\top \lambda^* \geq 0$  and  $u^* = \infty$  otherwise. This gives the optimal model  $p^*(\cdot) = p^*(\cdot; \lambda^*)$ .

Meanwhile, for any  $\lambda \geq 0$ , the primal problem: maximize $_{p \in \mathcal{P}, u \geq 0}$   $L_{\text{AND}}(p, u; \lambda)$  defines the dual function  $D_{\text{AND}}(\lambda)$ . By the variational optimality, maximization of  $L_{\text{AND}}(p, u; \lambda)$  over  $p$  and  $u$  is at a unique maximizer,

$$p^*(\cdot; \lambda, \mu) \propto q_{\text{AND}}^{(\lambda)}(\cdot)$$

and  $u^*(\lambda) = 0$  if  $1 - \mathbf{1}^\top \lambda \geq 0$  and  $u^*(\lambda) = \infty$  otherwise. This defines the dual function,

$$\begin{aligned} D_{\text{AND}}(\lambda) &= L_{\text{AND}}(p^*(\cdot; \lambda), u^*(\lambda); \lambda) \\ &= u^*(\lambda) + \sum_{i=1}^m \lambda_i \left( D_{\text{KL}}(p^*(\cdot; \lambda) \| q_{\text{AND}}^{(\lambda)}(\cdot)) - u^*(\lambda) \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda) \\ &= (1 - \mathbf{1}^\top \lambda) u^*(\lambda) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda) \end{aligned}$$

which completes the proof by following the definition of the dual problem and the dual constraint  $\mathbf{1}^\top \lambda \leq 1$ .  $\square$

## C.6 Proof of Theorem 4

*Proof.* Similar to the proof of Theorem 2, we can establish a saddle point condition for the Lagrangian  $\bar{L}_{\text{AND}}(s_p, u, \lambda)$  by leveraging the expressiveness of the function class  $\mathcal{S}$  which represents the path space  $\{p_{0:T}(\cdot)\}$ . As the proof follows similar steps, we omit the detail.  $\square$

## C.7 Proof of Lemma 3

*Proof.* From section C.5, we recall:

$$L_{\text{AND}}(p, u; \lambda) = u + \sum_{i=1}^m \lambda_i \left( D_{\text{KL}}(p \| q_{\text{AND}}^{(\lambda)}) - u \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \quad (68)$$

Since in the diffusion formulation of the problem (SR-A) we have  $p = p_0(x_0; s)$ ,  $q^i = p_0(x_0; s^i)$ , we can derive similarly to (68) that:

$$L_{\text{AND}}(p_0(\cdot; s), u; \lambda) = u + \sum_{i=1}^m \lambda_i \left( D_{\text{KL}}(p_0(\cdot; s) \| q_{\text{AND},0}^{(\lambda)}(\cdot)) - u \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \quad (69)$$

Since minimizing over  $u$  would trivially give  $\min_u L_{\text{AND}}(p, u; \lambda) = -\infty$  unless  $\mathbf{1}^\top \lambda = 1$ , we consider the Lagrangian in the non-trivial case where  $\mathbf{1}^\top \lambda = 1$ . Then we have:

$$L_{\text{AND}}(p(\cdot; s); \lambda) = L_{\text{AND}}(s, \lambda) = D_{\text{KL}}(p_0(\cdot; s) \| q_{\text{AND},0}^{(\lambda)}) - \log Z_{\text{AND}}(\lambda). \quad (70)$$

The second term  $\log Z_{\text{AND}}(\lambda)$  does not depend on  $s$ , thus it suffices to minimize  $D_{\text{KL}}(p_0(\cdot; s) \| q_{\text{AND},0}^{(\lambda)})$  to find the Lagrangian minimizer which we call  $s^{(\lambda)}$ . The KL is minimized when  $p_0(\cdot; s^{(\lambda)}) = q_{\text{AND},0}^{(\lambda)}$ . If we have access to samples from  $q_{\text{AND},0}^{(\lambda)}$ , we can fit  $s$  to  $q_{\text{AND},0}^{(\lambda)}$  by optimizing the Denoising score matching objective similar to Equation (1) in [41]:

$$L_{\text{sm}}(s, \lambda) = \sum_{t=0}^T \omega_t \mathbb{E}_{x_0 \sim q_{\text{AND}}^{(\lambda)}(\cdot)} \mathbb{E}_{x_t \sim q(x_t|x_0)} \left[ \|s(x, t) - \nabla \log q(x_t|x_0)\|^2 \right] \quad (71)$$

From [41] we know that given sufficient data and predictor capacity of  $s$  we have  $\arg\min_s L_{\text{sm}}(s, \lambda) \simeq q_{\text{AND},0}^{(\lambda)}$ .  $\square$

## D Composition with Forward KL Divergences

We start with the constrained problem formulation using forward KL divergence (UF-C) which we rewrite here:

$$\begin{aligned} & \underset{u \in \mathbb{R}, p \in \mathcal{P}}{\text{minimize}} && u \\ & \text{subject to} && D_{\text{KL}}(q^i \| p) \leq u \quad \text{for } i = 1, \dots, m. \end{aligned} \quad (72)$$

In the case of diffusion models, the KL divergence in (72) becomes the forward path-wise KL between the processes:

$$\begin{aligned} & \underset{u \in \mathbb{R}, p \in \mathcal{P}}{\text{minimize}} && u \\ & \text{subject to} && D_{\text{KL}}(q_{0:T}^i(\cdot) \| p_{0:T}(\cdot; s)) \leq u \quad \text{for } i = 1, \dots, m. \end{aligned} \quad (73)$$

It is important to note here that using the forward KL as a constraint makes sense when  $q^i$  represent forward diffusion processes obtained by adding noise to samples from some dataset. We can also solve this forward KL constrained problem to compose multiple models; In that case we treat samples generated by each model as a separate dataset with underlying distribution  $q_0^i(x_0)$ .

In summary, the two key differences of Problem (73) to Problem (UR-A) are: (i) The closeness of a model  $p$  to a pretrained model  $q^i$  is measured by the forward KL divergence  $D_{\text{KL}}(q^i \| p)$ , instead of the reverse KL divergence  $D_{\text{KL}}(p \| q^i)$ ; (ii) The distributions  $\{q^i\}_{i=1}^m$  can be the distributions underlying  $m$  datasets, not necessarily  $m$  pretrained models.

Regardless of whether the  $q^i$  represent pre-trained models or datasets, evaluating  $D_{\text{KL}}(q_{0:T}^i(\cdot) \| p_{0:T}(\cdot; s))$  is intractable since it requires knowing  $q_{0:T}^i(\cdot)$  which in turn requires knowing  $q_0^i(\cdot)$  exactly. To get around this issue we formulate a closely related problem to (73) by replacing the KL with the Evidence Lower Bound (Elbo):

$$\begin{aligned} & \underset{u \in \mathbb{R}, p \in \mathcal{P}}{\text{minimize}} && u \\ & \text{subject to} && \text{Elbo}(q_{0:T}^i; p_{0:T}) \leq u \quad \text{for } i = 1, \dots, m \end{aligned} \quad (74)$$

where the Elbo is defined as

$$\text{Elbo}(q_{0:T}; p_{0:T}) := \mathbb{E}_{x_0 \sim q_0} \mathbb{E}_{q(x_{1:T}|x_0)} \log \frac{p_{0:T}(x_{0:T})}{q(x_{1:T}|x_0)}. \quad (75)$$

We note that the typical approach to train a diffusion model is minimizing the Elbo. Furthermore, minimizing  $\text{Elbo}(q_{0:T}; p_{0:T})$  over  $p$  is equivalent to minimizing the KL divergence  $D_{\text{KL}}(q_{0:T}^i(\cdot) \| p_{0:T}(\cdot; s))$  since they only differ by a constant that does not depend on  $p$ . (see [21] for more details on this)

For a given  $\lambda$ , we define a weighted mixture of distributions as

$$q_{\text{mix}}^{(\lambda)}(\cdot) = \sum_{i=1}^m \frac{\lambda_i}{\lambda + 1} q^i(\cdot), \quad (76)$$

and we denote by  $H(q)$  the differential entropy of a given distribution  $q$ ,

$$H(q) := -\mathbb{E}_{x \sim q} [\log q(x)] \quad (77)$$

**Theorem 5.** Problem (74) is equivalent to the following unconstrained problem:

$$\underset{p \in \mathcal{P}}{\text{minimize}} \quad D_{\text{KL}}(q_{\text{mix}}^{(\lambda^*)} \| p) \quad (78a)$$

where  $\lambda^*$  is the optimal dual variable given by  $\lambda^* = \arg\max_{\lambda \geq 0} D(\lambda)$ . The dual function has the explicit form,  $D(\lambda) = H(q_{\text{mix}}^{(\lambda)})$ . Furthermore, the optimal solution of (7) is given by

$$p^* = q_{\text{mix}}^{(\lambda^*)}. \quad (78b)$$

Unlike the reverse KL case, here we can characterize the optimal dual multipliers, and the optimal solution further; Note that the optimal dual multiplier  $\lambda^* = \arg\max_{\lambda \geq 0} D(\lambda) = \arg\max_{\lambda \geq 0} H(q_{\text{mix}}(\cdot; \lambda^*))$  is one that maximizes the differential entropy  $H(\cdot)$  of the distribution of

the corresponding mixture. This implies that the optimal solution is the most diverse mixture of the individual distributions.

There are many potential use cases where we may want to compose distributions that don't overlap in their supports; For example when combining distributions of multiple dissimilar classes of a dataset. The following characterizes the optimal solution in such settings.

**Corollary 1.** *For the special case where the distributions  $q^i$  all have disjoint supports, the optimal dual multiplier  $\lambda^*$  of Problem (74) can be characterized explicitly as*

$$\lambda_i^* = \frac{e^{H(q^i)}}{\sum_{j=1}^m e^{H(q^j)}}.$$

## E Algorithm Details

### E.1 Alignment

Recall from Section 3.1 that the algorithm consists of two alternating steps:

**Primal minimization:** At iteration  $n$ , we obtain a new model  $s^{(n+1)}$  via a Lagrangian maximization,

$$s^{(n+1)} \in \operatorname{argmin}_{s \in \mathcal{S}} \bar{L}_{\text{ALI}}(s_p, \lambda^{(n)}).$$

**Dual maximization:** Then, we use the model  $s^{(n+1)}$  to estimate the constraint violation  $\mathbb{E}_{x_0}[r(x_0)] - b$ , denoted as  $r(s^{(n+1)}) - b$ , and perform a dual sub-gradient ascent step,

$$\lambda^{(n+1)} = \left[ \lambda^{(n)} + \eta \left( r(s^{(n+1)}) - b \right) \right]_+.$$

In practice we replace minimization over  $\mathcal{S}$  with minimization over a parametrized family of functions  $\mathcal{S}_\theta$ . The full algorithm is detailed in Algorithm 1.

---

#### Algorithm 1 Primal-Dual Algorithm for Reward Alignment of Diffusion Models

---

- 1: **Input:** total diffusion steps  $T$ , diffusion parameter  $\alpha_t$ , total dual iterations  $H$ , number of primal steps per dual update  $N$ , dual step size  $\eta_d$ , primal step size  $\eta_p$ , initial model parameters  $\theta(0)$ .
- 2: **Initialize:**  $\lambda(1) = 1/m$ .
- 3: **for**  $h = 1, \dots, H$  **do**
- 4:   Initialize  $\theta_1 = \theta(h - 1)$
- 5:   **for**  $n = 1, \dots, N$  **do**
- 6:     Take a primal gradient descent step

$$\theta_{n+1} = \theta_n - \eta_p \cdot \nabla_\theta \bar{L}_{\text{ALI}}(\theta, \lambda^{(n)}) \quad (79)$$

- 7:   **end for**
- 8:   Set the value of the parameters to be used for the next dual update:  $\theta(h) = \theta_{N+1}$ .
- 9:   Update dual multipliers for  $i = 1, \dots, m$ :

$$\lambda_i(h+1) = \left[ \lambda_i(h) + \eta_d (\mathbb{E}_{x_0 \sim p_0(\cdot; s_\theta)} [r_i(x_0)] - b_i) \right]_+ \quad (80)$$

10: **end for**

---

We now discuss the practicality of the primal gradient descent step (79) regarding the Lagrangian function,

$$\bar{L}_{\text{ALI}}(\theta, \lambda) = D_{\text{KL}}(p_{0:T}(\cdot; s_\theta) \| q_{0:T}(\cdot; s_q)) - \sum_i \lambda_i (\mathbb{E}_{x_0 \sim p_0(\cdot; s_\theta)} [r_i(x_0)] - b_i) \quad (81)$$

To derive the gradient of  $\bar{L}_{\text{ALI}}(\theta, \lambda)$ , we first take the derivative of the expected reward terms by noting that the expectation is taken over a distribution that depends on the optimization variable  $\theta$ . We can use the following result (Lemma 4.1 from [16]) to take the gradient inside the expectation.



1092 **Lemma 9.** *If  $p_\theta(x_{0:T})r(x_0)$  and  $\nabla_\theta p_\theta(x_{0:T})r(x_0)$  are continuous functions of  $\theta$ , then we can write*  
 1093 *the gradient of the reward function as*

$$\nabla_\theta \mathbb{E}_{x_0 \sim p_0(\cdot; s_\theta)} [r(x_0)] = \mathbb{E}_{x_{0:T} \sim p_{0:T}(\cdot; s_\theta)} \left[ r(x_0) \sum_{t=1}^T \nabla_\theta \log p(x_{t-1} | x_t; s_\theta) \right].$$

1094 For the gradient of the KL divergence, we have

$$\begin{aligned} \nabla_\theta D_{\text{KL}}(p_{0:T}(\cdot; s_\theta) \| q_{0:T}(\cdot; s_q)) &= \nabla_\theta \left( \sum_{t=1}^T \mathbb{E}_{x_t \sim p_t(\cdot; s_\theta)} \left[ \frac{1}{2\sigma_t^2} \|s_\theta(x_t, t) - s_q(x_t, t)\|^2 \right] \right) \\ &= \nabla_\theta \left( \sum_{t=1}^T \mathbb{E}_{x_t \sim p_t(\cdot; s_\theta)} [D_{\text{KL}}(p(x_{t-1} | x_t; s_\theta) \| p(x_{t-1} | x_t; s_q))] \right) \\ &= \sum_{t=1}^T \mathbb{E}_{x_t \sim p_t(\cdot; s_\theta)} [\nabla_\theta D_{\text{KL}}(p(x_{t-1} | x_t; s_\theta) \| p(x_{t-1} | x_t; s_q))] \\ &\quad + \sum_{t=1}^T \mathbb{E}_{x_t \sim p_t(\cdot; s_\theta)} \left[ \sum_{t' > t}^T \nabla_\theta \log p(x_{t'-1} | x_{t'}; s_\theta) D_{\text{KL}}(p(x_{t-1} | x_t; s_\theta) \| p(x_{t-1} | x_t; s_q)) \right]. \end{aligned}$$

1098 The second term we ignore in practice for simplicity without hurting performance. See [16, Appendix  
 1099 A.3] for the derivation.

## 1100 E.2 Composition

1101 For composition, we take a similar approach to Algorithm 1. Recall from Lemma 3 that the Lagrangian  
 1102 minimizer for the constrained composition problem can be found by minimizing:

$$\widehat{L}_{\text{AND}}(\theta, \lambda) := \sum_{t=0}^T \omega_t \mathbb{E}_{x_0 \sim q_{\text{AND}}^{(\lambda)}(\cdot)} \mathbb{E}_{x_t \sim q(x_t | x_0)} \left[ \|s_\theta(x, t) - \nabla \log q(x_t | x_0)\|^2 \right]$$

1103 Thus, we detail the algorithm for composition in Algorithm 2.

---

### Algorithm 2 Primal-Dual Algorithm for Product Composition (AND) of Diffusion Models

---

- 1: **Input:** total diffusion steps  $T$ , diffusion parameter  $\alpha_t$ , total dual iterations  $H$ , number of primal steps per dual update  $N$ , dual step size  $\eta_d$ , primal step size  $\eta_p$ , initial model parameters  $\theta(0)$ .
- 2: **Initialize:**  $\lambda(1) = 1/m$ .
- 3: **for**  $h = 1, \dots, H$  **do**
- 4:   Initialize  $\theta_1 = \theta(h-1)$
- 5:   **for**  $n = 1, \dots, N$  **do**
- 6:     Take a primal gradient descent step

$$\theta_{n+1} = \theta_n - \eta_p \cdot \nabla_\theta \widehat{L}_{\text{AND}}(\theta, \lambda^{(n)}) \quad (82)$$

- 7:   **end for**
- 8:   Set the value of the parameters to be used for the next dual update:  $\theta(h) = \theta_{N+1}$ .
- 9:   Update dual multipliers for  $i = 1, \dots, m$ :

$$\widetilde{\lambda}_i(h+1) = \lambda_i(h) + \eta_d D_{\text{KL}}(p_0(\cdot; s_{\theta(h)}) \| p_0(\cdot; s^i)) \quad (83)$$

- 10:    $\lambda(h+1) = \text{proj}(\widetilde{\lambda}(h+1))$ , where  $\text{proj}(y)$  projects its input onto the simplex  $\lambda^\top \mathbf{1} = 1$ .
  - 11: **end for**
- 

1104 The projection of the dual multipliers vector at the end is because we are maximizing the Dual  
 1105 function and as seen in the proof of Theorem 3 this requires that  $\lambda^\top \mathbf{1} = 1$ .

1106 Note that implicit in Algorithm 2 is the fact that for minimizing the Lagrangian  $\widehat{L}_{\text{AND}}(\theta, \lambda)$  we need  
 1107 samples from the weighted product distribution  $q_{\text{AND}}^{(\lambda)}(\cdot)$ . We do this using the Annealed MCMC  
 1108 sampling algorithm proposed in [14].

1109 **Skipping the Primal.** As mentioned in Section 5, Annealed MCMC sampling and the minimization  
 1110 of the Lagrangian  $\widehat{L}_{\text{AND}}(\theta, \lambda)$  at each primal step to match the true score  $\nabla \log q_{\text{AND}}^\lambda$  are both difficult  
 1111 and computationally costly. This is why for the settings other than the Low-Dimensional setting  
 1112 discussed in Appendix F.1 we propose Algorithm 3 that skips the primal step entirely.

1113 We achieve this by using the surrogate product score (rather than the true score) for computing the  
 1114 point-wise KL needed for the dual updates. The difference between the two is also discussed in [14].

$$\text{true score: } \nabla \log q_{\text{AND},t}^{(\lambda)}(x_t) = \nabla \log \left( \int \sum_i (q_0(x_0))^{\lambda_i} q(x_t|x_0) dx_0 \right) \quad (84)$$

$$\text{surrogate score: } \nabla \log \widehat{q}_{\text{AND},t}^{(\lambda)}(x_t) = \sum_i \lambda_i \nabla \log \left( \int q_0(x_0) q(x_t|x_0) dx_0 \right) \quad (85)$$

---

**Algorithm 3** Dual-Only Algorithm for Product Composition (AND) of Diffusion Models

---

- 1: **Input:** total diffusion steps  $T$ , diffusion parameter  $\alpha_t$ , total dual iterations  $H$ , dual step size  $\eta_d$ .
- 2: **Initialize:**  $\lambda(1) = 1/m$ .
- 3: **for**  $h = 1, \dots, H$  **do**
- 4:   Update dual multipliers for  $i = 1, \dots, m$ :

$$\widetilde{\lambda}_i(h+1) = \lambda_i(h) + \eta_d D_{\text{KL}}(\widehat{q}_{\text{AND},0}^{(\lambda(h))}(\cdot) \| p_0(\cdot; s^i)) \quad (86)$$

- 5:    $\lambda(h+1) = \text{proj}(\widetilde{\lambda}(h+1))$ , where  $\text{proj}(y)$  projects its input onto the simplex  $\lambda^T 1 = 1$ .

6: **end for**

---

1115 For a given  $\lambda$ , the surrogate score can be easily computed:

$$\nabla \log \widehat{q}_{\text{AND},t}^{(\lambda)}(x_t) = \sum_i \lambda_i \nabla \log \left( \int q_0(x_0) q(x_t|x_0) dx_0 \right) \quad (87)$$

$$= \sum_i \lambda_i \nabla \log p_t(x_t; s^i) \quad (88)$$

1117 and thus we can use Lemma 2 to compute the point-wise KLs needed for the dual update. As for the  
 1118 samples needed from the true product distribution, we also replace them with samples obtained by  
 1119 running DDIM using the surrogate score.

## 1120 F More Experiments and Experimental Details

### 1121 F.1 Low-dimensional synthetic experiments

1122 For illustrating the difference between the constrained and unconstrained approach visually, we set  
 1123 up experiments where the generated samples are in  $\mathbb{R}^2$ . For the score predictor we used the same  
 1124 ResNet architecture as used in [14].

1125 **Product composition (AND).** Unlike the image experiments, in this low-dimensional setting we used  
 1126 Algorithm 2 for product composition. See Figure 1 for visualization of the resulting distributions.

1127 **Mixture composition (OR).** For this experiment we used the same Algorithm as the one used in [21]  
 1128 for mixture of distributions. The only modification is doing an additional dual multiplier projection  
 1129 step similar to the last step of the product composition Algorithm 2. See Figure 2 for visualization of  
 1130 the resulting distributions.

### 1131 F.2 Reward product composition (section 5.2 (I))

1132

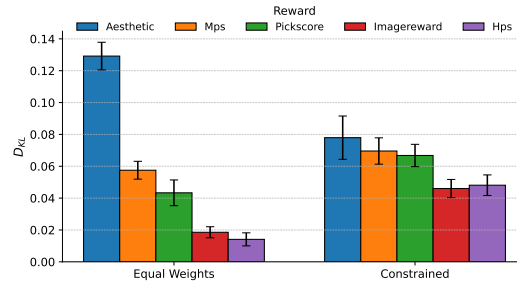


Figure 6: KL divergence for the product composition of 5 adapters pre-trained with different rewards. Error bars denote the standard deviation computed across 8 text prompts each with four samples.

1133 **Implementation details and hyperparameters.** We finetuned the model using the Alignprop [33]  
 1134 official implementation <sup>4</sup> for each individual reward using the hyperparameters reported in Table 2.  
 1135 We then composed the trained adapters running dual ascent using the surrogate score as described in  
 1136 section E.2. We use the average of scores (denoted as “Equal weights”) as a baseline. Hyperparameters  
 1137 are described in Table 3. The reward values reported in Figure 4 were normalised so that 0%  
 1138 corresponds to the reward obtained by the pre-trained model, and 100% the reward obtained by the  
 1139 model fine-tuned solely on the corresponding reward.

1140 **Additional results.** As shown in Figure 6, equal weighting leads to disparate KL’s across adapters  
 1141 – in particular high KL with respect to the adapter trained with the “aesthetic” reward – while our  
 1142 constrained approach effectively reduces the worst case KL, equalizing divergences across adapters.  
 1143 Figure 4 shows images sampled from these two compositions exhibit different characteristics, with  
 1144 our constrained approach producing smoother backgrounds, shallower depth of field and more  
 1145 painting-like images.

<sup>4</sup><https://github.com/mihirp1998/AlignProp>

Hyperparameter	Value
Batch size	64
Samples per epoch	128
Epochs	10
Sampling steps	50
Backpropagation sampling	Gaussian
KL penalty	0.1
Learning rate	$1 \times 10^{-3}$
LoRA rank	4

Table 2: Hyperparameters used to finetune models using individual rewards.

Hyperparameter	Value
Base model	runwayml/stable-diffusion-v1-5
Prompts	{"cheetah", "snail", "hippopotamus", "crocodile", "lobster", "octopus"}
Resolution	512
Batch size	4
Dual steps	5
Dual learning rate	1.0
Sampling steps	25
Guidance scale	5.0
Rewards	aesthetic, hps, pickscore, imagereward, mps

Table 3: Hyperparameters for product composition of models finetuned with different rewards.

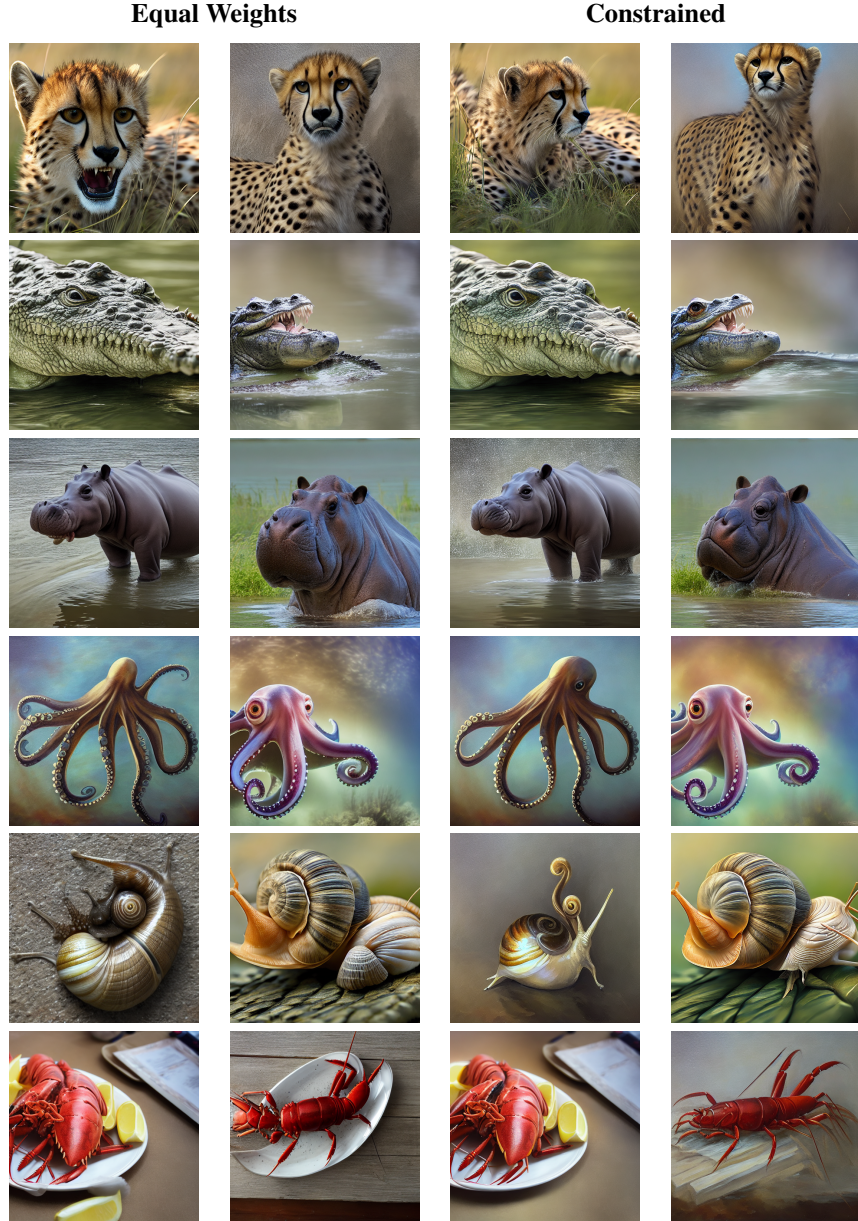


Table 4: Images sampled from the same latents for the product of adapters using the equal weights and when using the proposed KL-constrained reweighting scheme using 5 dual steps.

### 1146 F.3 Concept composition (section 5.2 (II))

1147 We present additional results for concept composition using three different concepts (as opposed to  
 1148 just 2 in the main paper and in [38]) As seen in table 5, our approach retains a clear advantage in both  
 1149 CLIP and BLIP scores. See Table 6 for examples of images generated using each method. Images  
 1150 with the constrained method typically do a better job of representing all concepts.



	Min. CLIP ( $\uparrow$ )	Min. BLIP ( $\uparrow$ )
Combined Prompting	21.52	0.206
Equal Weights	22.18	0.203
Constrained (Ours)	<b>22.45</b>	<b>0.221</b>

Table 5: Comparing constrained approach to baselines on minimum CLIP and BLIP scores. The scores are averaged over 50 different prompt triplets sampled from a list of simple prompts.



Table 6: Concept composition examples for each method. Prompts used for each row:  
**Row 1:** "a pineapple", "a volcano". **Row 2:** "a donut", "a turtle". **Row 3:** "a lemon", "a dandelion".  
**Row 4:** "a dandelion", "a spider web", "a cinnamon roll".

#### 1151 F.4 Alignment experiments

1152 **Reward normalization.** In practice, setting constraint levels for multiple rewards that are both  
1153 feasible and sufficiently strict to enforce the desired behavior is challenging. Different rewards  
1154 exhibit widely varying scales. This is illustrated in Table 7, which shows the mean and standard  
1155 deviation of reward values for the pre-trained model. This issue can be exacerbated by the unknown  
1156 interdependencies among constraints and the lack of prior knowledge about their relative difficulty or  
1157 sensitivity.

1158 In order to tackle this, we propose normalizing rewards using the pre-trained model statistics as a  
1159 simple yet effective heuristic. This normalization facilitates the setting of constraint levels, enables  
1160 direct comparisons across rewards and enhances interpretability. In all of our experiments, we apply

1161 this normalization before enforcing constraints. Explicitly, we set

$$\tilde{r} = \frac{r - \hat{\mu}_{\text{pre}}}{\hat{\sigma}_{\text{pre}}}, \quad (89)$$

1162 where  $r$  denotes the original reward and  $\hat{\mu}_{\text{pre}}, \hat{\sigma}_{\text{pre}}$  the sample mean and standard deviation of  
 1163 the reward for the pre-trained model. We find that, with this simple transformation, setting equal  
 1164 constraint levels can yield satisfactory results while forgoing extensive hyperparameter tuning.

Reward	Mean	Std
Aesthetic	5.1488	0.4390
HPS	0.2669	0.0057
MPS	5.2365	3.5449
PickScore	21.1547	0.6551
Local Contrast	0.0086	0.0032
Saturation	0.1060	0.0706

Table 7: Mean and standard deviation of reward values for the pre-trained model.

## 1165 I. MPS + local contrast, saturation.

1166 In this experiment, we augment a standard alignment loss—trained on user preferences—with two  
 1167 differentiable rewards that control specific image characteristics: local contrast and saturation. These  
 1168 rewards are computationally inexpensive to evaluate and offer direct interpretability in terms of their  
 1169 visual effect on the generated images. In addition, the unconstrained maximization of these features  
 1170 would lead to undesirable generations. other potentially useful rewards not explored in this work are  
 1171 brightness, chroma energy, edge strength, white balancing and histogram matching.

**Local contrast reward.** In order to prevent images with excessive sharpness, we minimize the  
 “local contrast”, which we define as the mean absolute difference between the luminance of the  
 image and a low-pass filtered version. Explicitly, let  $Y$  denote the luminance, computed as  $Y =$   
 $0.2126R + 0.7152G + 0.0722B$ , and  $G_{\sigma} * Y$  the luminance blurred with a Gaussian kernel of standard  
 deviation  $\sigma = 1.0$ . We minimize the average per pixel difference by maximizing the reward

$$r_C = -\frac{1}{HW} \sum_{i,j} |Y_{ij} - (G_{\sigma} * Y)_{ij}|,$$

1172 where  $H, W$  denote image dimensions.

1173 **Saturation reward.** To discourage overly saturated images, we simply penalize saturation, which we  
 1174 compute from  $R, G, B$  pixel values as

$$r_S = -\frac{1}{HW} \sum_{i,j} \frac{\max_{c \in \{R, G, B\}} x_{i,j}^{(c)} - \min_{c \in \{R, G, B\}} x_{i,j}^{(c)}}{\max_{c \in \{R, G, B\}} x_{i,j}^{(c)} + \varepsilon},$$

1175 where  $\varepsilon = 1 \times 10^{-8}$  is a small constant added for numerical stability.

1176 **Implementation details and hyperparameters.** We implemented our primal-dual alignment ap-  
 1177 proach (Algorithm 1) in the Alignprop framework. Following their experimental setting, we use  
 1178 different animal prompts for training and evaluation. Hyperparameters are detailed in Table 8.

1179 **Additional results.** We include images sampled from the constrained model in Figure 9 for hps  
 1180 and aesthetic reward functions. Samples from a model trained with an equally weighted model are  
 1181 included for comparison. Constraints prevent overfitting to the saturation and smoothness penalties.

## 1182 II. Multiple aesthetic constraints

1183 **Implementation details and hyperparameters.** We modified the Alignprop framework to accom-  
 1184 modate Algorithm 1. Following their setup, we use text conditioning on prompts of simple animals,  
 1185 using separate sets for training and evaluation. In this setting, due to the high variability of rewards  
 1186 throughout training, utilized an exponential moving average to reduce the variance in slack estimates  
 1187 (and hence dual subgradients) [39]. Hyperparameters are detailed in Table 10.

1188 **Additional results.** We include two images per method and prompt in Figure 11. These are sampled  
 1189 from the same latents for both models.



Hyperparameter	Value
Base model	runwayml/stable-diffusion-v1-5
Sampling steps	15
Dual learning rate	0.05
Batch size (effective)	$4 \times 16 = 64$
Samples per epoch	128
Epochs	20
KL penalty	0.1
LoRA rank	4
	MPS: 0.5
Constraint level	Saturation: 0.5
	Local contrast: 0.25
Equal weights	0.2

Table 8: Hyperparameters for reward alignment with contrast and saturation constraints. Constraint levels correspond to normalised rewards.

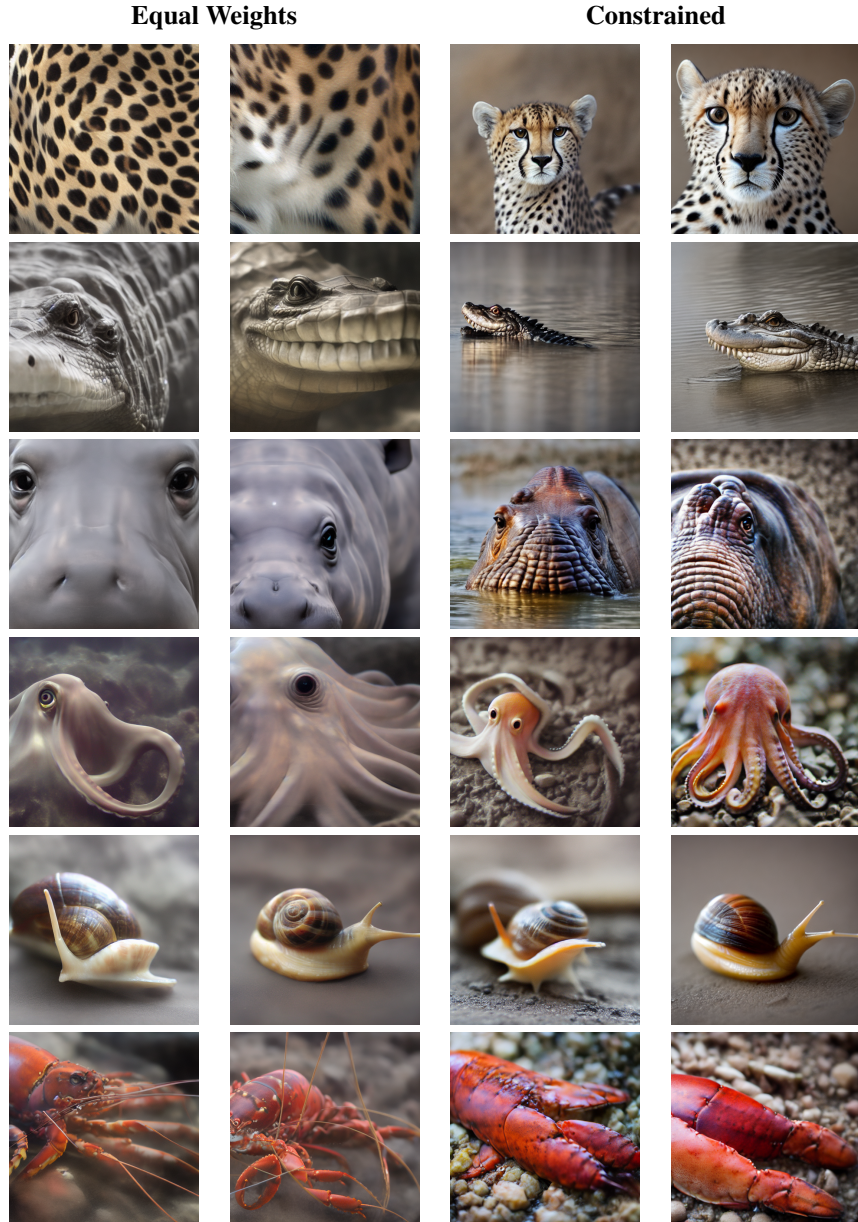


Table 9: Images sampled from models finetuned to maximize MPS [58], along with sharpness and saturation penalizations. We compare optimizing an equally weighted objective against our constrained approach.

Hyperparameter	Value
Base model	runwayml/stable-diffusion-v1-5
Sampling steps	15
Dual learning rate	0.05
Batch size (effective)	$4 \times 16 = 64$
Samples per epoch	128
Epochs	25
KL penalty	0.1
LoRA rank	4
	MPS: 0.5 HPS: 0.5
Constraint level	Aesthetic: 0.5 Pickscore : 0.5
Equal weights	0.2
Training Prompts	{"cat", "dog", "horse", "monkey", "rabbit", "zebra" "spider", "bird", "sheep", "deer", "cow", "goat" "lion", "tiger", "bear", "raccoon", "fox", "wolf" "lizard", "beetle", "ant", "butterfly", "fish", "shark" "whale", "dolphin", "squirrel", "mouse", "rat", "snake" "turtle", "frog", "chicken", "duck", "goose", "bee" "pig", "turkey", "fly", "llama", "camel", "bat" "gorilla", "hedgehog", "kangaroo"}
Evaluation Prompts	{"cheetah", "snail", "hippopotamus", "crocodile", "lobster", "octopus"}

Table 10: Hyperparameters for reward alignment with multiple rewards. Constraint levels correspond to normalised rewards.

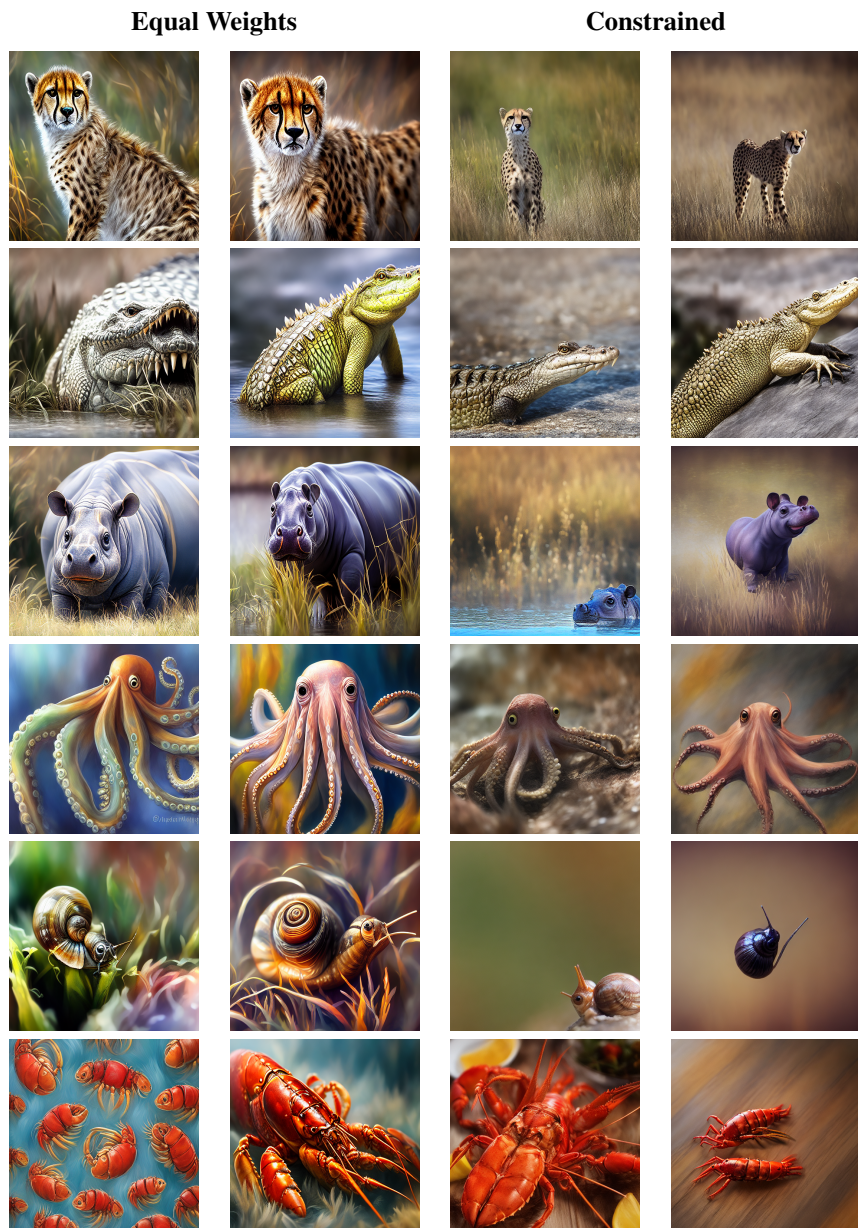


Table 11: Samples from models fine-tuned using multiple rewards with equal weights and with our constrained alignment method.