Composition and Alignment of Diffusion Models using Constrained Learning

Shervin Khalafi*
University of Pennsylvania
shervink@seas.upenn.edu

Ignacio Hounie
University of Pennsylvania
ihounie@seas.upenn.edu

Dongsheng Ding*
University of Tennessee, Knoxville
dongshed@utk.edu

Alejandro Ribeiro
University of Pennsylvania
aribeiro@seas.upenn.edu

Abstract

Diffusion models have become prevalent in generative modeling due to their ability to sample from complex distributions. To improve the quality of generated samples and their compliance with user requirements, two commonly used methods are: (i) Alignment, which involves finetuning a diffusion model to align it with a reward; and (ii) Composition, which combines several pretrained diffusion models together, each emphasizing a desirable attribute in the generated outputs. However, trade-offs often arise when optimizing for multiple rewards or combining multiple models, as they can often represent competing properties. Existing methods cannot guarantee that the resulting model faithfully generates samples with all the desired properties. To address this gap, we propose 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 pretrained model. We provide a theoretical characterization of the solutions to the constrained alignment and composition problems and develop a Lagrangian-based primal-dual training algorithm to approximate these solutions. Empirically, we demonstrate our proposed approach in image generation, applying it to alignment and composition, and show that our aligned or composed model satisfies constraints effectively. Our implementation can be found at: https://github.com/shervinkhalafi/constrained_comp_align

1 Introduction

Diffusion models have emerged as the tool of choice for generative modeling in a variety of settings [38, 3, 50, 9], image generation being most prominent among them [37]. Users of these diffusion models would like to adapt them to their specific preferences, but this aspiration is hindered by the often enormous cost and complexity of their training [48, 56]. For this reason, *alignment* and *composition* of what, in this context, become *pretrained* models, has become popular [29, 31].

Regardless of whether the goal is alignment or composition, we want to balance what are most likely conflicting requirements. In alignment, we want to stay close to the pretrained model while deviating sufficiently so as to affect some rewards of interest [17, 13]. In composition, given several pretrained models, our goal is to sample from the union or intersection of their distributions [14, 1]. The standard approach to balance these requirements involves the use of *weighted averages*. This can be a linear combination of score functions in composition [14, 1] or may involve a loss given by a linear combination of a Kullback-Leibler (KL) divergence and a reward [17] in the case of alignment.

*Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

In practice, weight-based methods are often designed in an ad hoc manner, with the weights treated as tunable hyperparameters, which makes the approach notoriously difficult to optimize and generalize.

In this work, we propose a unified view of alignment and composition via the lens of constrained learning [7, 6]. As their names indicate, constrained alignment and constrained composition problems balance conflicting requirements using *constraints* instead of *weights*. Learning with constraints and learning with weights are related problems – indeed, we will train constrained diffusion models in 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 summarize our key contributions in three aspects below.

(i) Problem Formulation

- For alignment, we formulate a reverse KL divergence-constrained distribution optimization problem that minimizes the reverse KL divergence to a pretrained model, subject to expected reward constraints with user-specified thresholds.
- For composition, we propose using KL divergence constraints to ensure the closeness to each pretrained model. It is important to distinguish composition with *reverse* KL and *forward* KL constraints as they lead to a weighted product or weighted mixture [22] of the individual distributions, respectively. In this work, we focus on composition with reverse KL constraints, and discuss forward KL constraints in Appendix E.

(ii) Theoretical Analysis

- In Section 3, we characterize the solution of the alignment problem as the pretrained model distribution scaled by an exponential function of a weighted sum of reward functions. 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. We establish strong duality for both problems, which enables us to use a dual-based approach to develop primal-dual training algorithms for solving them.
- We illustrate the distinction between the KL divergence between diffusion trajectories (path-wise), and the KL divergence between the final distributions (point-wise) in Section 2.2. We also propose a new method to evaluate the point-wise KL divergence.

(iii) Empirical Results

- For alignment, we demonstrate the difference between constrained and weighted alignment through experiments in Section 5.1. The constrained approach scales naturally to finetuning with multiple rewards, eliminating the need for extensive hyperparameter searches to determine suitable weights. Moreover, specifying reward thresholds is often more intuitive than choosing regularization weights. Without constraints, however, the model can easily overfit to one or several rewards and diverge substantially from the pretrained model. In contrast, our method identifies the model closest to the pretrained one that still satisfies the desired reward constraints (see Figure 4).
- For composition, we show the properties of constrained composition of diffusion models through experiments in Section 5.2. We see that when the composition weights are not chosen properly, the resulting model can become biased towards certain individual models while neglecting others. Constrained composition addresses this issue by finding optimal weights that preserve closeness to each individual model. Particularly, when composing multiple text-to-image models each finetuned on a different reward, imposing constraints yields weights that enable the composed model to achieve higher performance across all rewards, compared to composition with equal weights.

2 Composition and Alignment of Diffusion Models

We introduce constrained distribution problems for alignment and composition in Section 2.1, and characterize the reverse and forward KL divergences for diffusion models in Section 2.2.

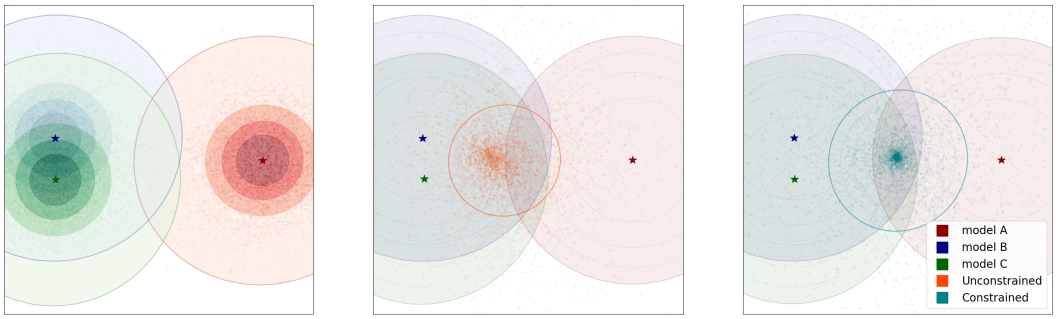

Figure 1: Product composition (AND). Three Gaussian distributions being composed (Left). Composition using equal weights (Middle), and with constraints (Right). The constrained model samples from the intersection of the three models.

2.1 Composition and alignment in distribution space

We formulate Unparameterized constrained distribution optimization problems using Reverse or Forward KL divergence, for Alignment and Composition as illustrated in (UR-A), (UR-C), and (UF-C).

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 $D_{\mathrm{KL}}(p \,\|\, q) := \int p(x) \log(p(x)/q(x))dx$ 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:

$$p^\star \;=\; \operatorname*{argmin}_{p}\; D_{\mathrm{KL}}\big(\,p \,\|\, q\,\big) \qquad \text{subject to } \mathbb{E}_{x\sim p}\big[\,r_i(x)\,\big] \;\geq\; b_i \;\text{ for } i = 1, \ldots, m. \tag{UR-A}$$

As per (UR-A), the constrained alignment problem is solved by the distribution $p^\star$ that is closest to the pretrained one q as measured by the reverse KL divergence $D_{\mathrm{KL}}(p \,\|\, q)$ 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 be $P^\star_{\mathrm{ALI}} := D_{\mathrm{KL}}(p^\star \,\|\, q)$.

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:

$$(p^\star, u^\star) \;=\; \operatorname*{argmin}_{p,\,u}\; u \qquad \text{subject to } D_{\mathrm{KL}}\big(\,p \,\|\, q^i\,\big) \leq u \;\text{ for } i = 1, \ldots, m. \tag{UR-C}$$

In (UR-C), the decision variable u serves as an upper bound on the 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 this common upper bound. Hence, the optimal solution $p^\star$ minimizes the maximum KL divergence among m terms, each computed between p and a pretrained model q_i. Let the primal value be $P^\star_{\mathrm{AND}} := u^\star$. The epigraph formulation (UR-C) is practical, as the constraint threshold u can be updated dynamically during training. In contrast, Figure 1 shows that the model composed with equal weights is biased toward the two most similar distributions.

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 $D_{\mathrm{KL}}(p \,\|\, q_i)$ in (UR-C) with the *forward* KL divergence $D_{\mathrm{KL}}(q_i \,\|\, p)$:

$$(p^\star, u^\star) \;=\; \operatorname*{argmin}_{p,\,u}\; u \qquad \text{subject to } D_{\mathrm{KL}}\big(\,q_i \,\|\, p\,\big) \leq u \;\text{ for } i = 1, \ldots, m. \tag{UF-C}$$

Mixture composition was studied in a related but slightly different constrained setting [22]. In fact, the solution of the constrained problem (UF-C) learns to sample from each distribution in proportion to its entropy; see [22, Theorem 2]. As shown in Figure 2, the constrained model samples more frequently from the higher-entropy distribution with two models, whereas the equally weighted composition samples equally from both distributions, leading to unbalanced sampling. Since the algorithmic design and analysis for (UF-C) follow those in [22], mixture composition is not the main focus of this work. For completeness, we compare it with product composition in Appendix E.

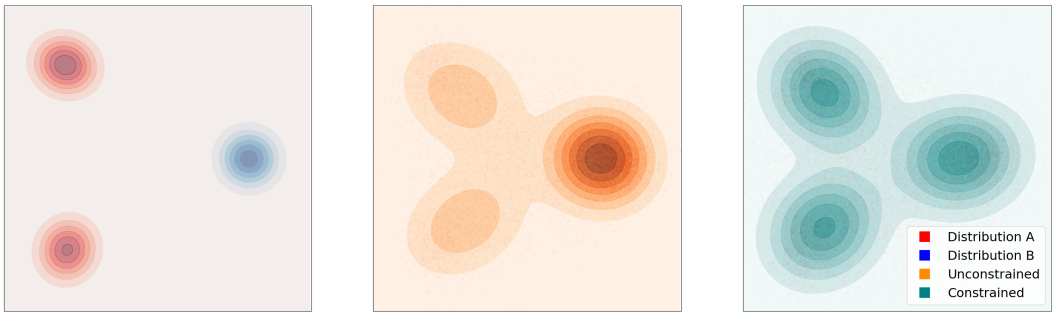

Figure 2: Mixture composition (OR). Two of Gaussian mixtures being composed (Left). One has two modes and the other has only a single mode. Composition using equal weights (Middle), and with constraints (Right).

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

2.2 KL divergence for diffusion models

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

$$\bar{X}_t \;=\; \frac{\alpha_t}{\alpha_{t-1}}\,\bar{X}_{t-1} \;+\; \sqrt{1-\frac{\alpha_t}{\alpha_{t-1}}}\,\epsilon_t, \quad \text{for } t = 1, \cdots, T \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ is the standard Gaussian noise, 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 $s(x,t)$: $\mathbb{R}^d \times \{1, \cdots, T\} \to \mathbb{R}^d$, we introduce a backward denoising diffusion implicit model (DDIM) process [42] as follows

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

where $\epsilon_t \sim \mathcal{N}(0, I)$ is the standard Gaussian noise, and $\{\sigma_t^2\}_{t=1}^T$ is the variance schedule that determines the level of randomness in the backward process (e.g., $\sigma_t = 0$ reduces to deterministic trajectories), and $\beta_t := \sqrt{\frac{\alpha_{t-1}}{\alpha_t}}\sqrt{(1-\alpha_t)(1-\bar{\alpha}_t)} - \sqrt{(1-\alpha_{t-1}-\sigma_t^2)(1-\bar{\alpha}_t)}$ is determined by the variance schedule σ_t and the noise schedule α_t. Here, we use the equivalence between the score-matching and denoising formulations of diffusion model to replace the denoising predictor in [42] by the score function. Given a score function $s(x,t)$, we denote the marginal density of X_t as $p_t(\cdot\,; s)$ and the joint distribution over the entire process as $p_{0:T}(x_{0:T}; s)$.

In the score-matching formulation [43], a denoising score-matching objective is minimized to obtain a function $s^\star$ that approximates the true score function of the forward process, i.e., $s^\star(x,t) \approx \nabla \log \bar{p}_t(x)$. Then, the marginal densities of the backward process (2) match those of the forward process (1), i.e., $p_t(\cdot\,; s^\star) = \bar{p}_t(\cdot)$ for all t. Thus we can run the backward process to generate samples $x_0 \sim p_0$ that resemble samples from the original data distribution $\bar{x}_0 \sim \bar{p}_0$.

We denote the KL divergence between two joint distributions p, q over the backward process by $D_{\mathrm{KL}}(p_{0:T}(\cdot) \,\|\, q_{0:T}(\cdot))$, which is known as path-wise KL [17, 19]. The path-wise KL divergence is often used in alignment to measure the difference between finetuned and pretrained models.

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

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

See Appendix C.1 for the proof. When the two backward processes differ in their variance and noise schedules, the path-wise KL divergence remains tractable, and we omit it for simplicity. While the path-wise KL divergence is a useful regularizer for alignment, when composing multiple models, the point-wise KL divergence $D_{\mathrm{KL}}(p_0(\cdot) \,\|\, q_0(\cdot))$ is a more natural measure of the closeness between two diffusion models. This is because we mainly care about the closeness of the final sampling distributions: $p_0(\cdot)$, $q_0(\cdot)$, and not the underlying processes: $p_{0:T}(\cdot)$, $q_{0:T}(\cdot)$. However, since our proposed approach to compute the point-wise KL is intractable for alignment, we adopt the path-wise KL for alignment and retain the point-wise KL for composition; see more discussion in Section 4.2.

However, it is not obvious how to compute the point-wise KL divergence, as evaluating the marginal densities is intractable. We next establish a similar formula as (3) by limiting the score function class.

Lemma 2 (Point-wise KL divergence)**.** *Assume two score functions $s_p(x,t) = \nabla \log \bar{p}_t(x)$, $s_q(x,t) = \nabla \log \bar{q}_t(x)$, where $\bar{p}_t$, $\bar{q}_t$ are two marginal densities induced by two forward diffusion processes, 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 DDIM with s_p and s_q is given by*

$$D_{\mathrm{KL}}(p_0(\cdot\,; s_p) \,\|\, q_0(\cdot\,; s_q)) \;=\; \sum_{t=0}^{T} \widetilde{\omega}_t \, \mathbb{E}_{x \sim p_t(\cdot\,; s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \;+\; \epsilon_T \tag{4}$$

where $\widetilde{\omega}_t$ is a time-dependent constant, and ϵ_T is a discretization error that depends on the total number of diffusion time steps T.

See Appendix C.2 for the proof. The key intuition behind Lemma 2 is that if two diffusion processes are close, and their starting distributions are the same (e.g., $\mathcal{N}(0, I)$ at time $t = T$), then the end points (i.e., the distributions at $t = 0$) must also be close. The sum on the right-hand side of (4) can be viewed as the difference of the two processes over time steps, up to a discretization error.

3 Aligning Pretrained Model with Multiple Reward Constraints

We provide a characterization of the solution to Problem (UR-A) in Section 3.1, and establish strong duality for diffusion models in Section 3.2, together with a dual-based training algorithm.

3.1 Reward alignment in distribution space

To apply Problem (UR-A) to diffusion models, we first employ Lagrangian duality to derive its solution in 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 [17]. They are related because the Lagrangian for Problem (UR-A) is precisely the weighted average:

$$L_{\mathrm{ALI}}(p, \lambda) \;=\; D_{\mathrm{KL}}\big(\, p \,\|\, q \,\big) \;-\; \lambda^\top \left(\mathbb{E}_{x \sim p}[\, r(x) \,] - b \right). \tag{5}$$

where we use shorthand $b := [\, b_1, \ldots, b_m \,]^\top$, $r := [\, r_1, \ldots, r_m \,]^\top$, and $\lambda := [\lambda_1, \ldots \lambda_m]^\top$ is the Lagrangian multiplier or dual variable. Let the dual function be $D_{\mathrm{ALI}}(\lambda) := \mathrm{minimize}_p \, L_{\mathrm{ALI}}(p, \lambda)$ and an optimal dual variable be $\lambda^\star \in \mathrm{argmax}_{\lambda \geq 0} \, D_{\mathrm{ALI}}(\lambda)$. Denote $D^\star_{\mathrm{ALI}} := D_{\mathrm{ALI}}(\lambda^\star)$. For any $\lambda > 0$, we define the reward weighted distribution $q_{\mathrm{rw}}^{(\lambda)}$ (subscript rw for *reward weighted*):

$$q_{\mathrm{rw}}^{(\lambda)}(\cdot) \;:=\; \frac{1}{Z_{\mathrm{rw}}(\lambda)} q(\cdot) \mathrm{e}^{\lambda^\top r(\cdot)} \tag{6}$$

where $Z_{\mathrm{rw}}(\lambda) = \int q(x) \mathrm{e}^{\lambda^\top r(x)} dx$ is the 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. Thus, we can 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 variable to a solution to the dual problem.

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

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

$$\underset{p}{\text{minimize}} \;\; D_{\text{KL}}\big(\, p \,\|\, q_{\text{rw}}^{(\lambda^\star)}\,\big) \tag{7}$$

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

$$p^\star \;=\; q_{\text{rw}}^{(\lambda^\star)}. \tag{8}$$

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

3.2 Reward alignment of diffusion models

We introduce diffusion models into Problem (UR-A) by representing p and q as two diffusion models: $p_{0:T}(\cdot; s_p)$ and $q_{0:T}(\cdot; s_q)$, with score functions s_p and s_q, respectively. The path-wise KL divergence has been widely used in diffusion model alignment to capture the difference between two diffusion models [44]. Hence, we instantiate Problem (UR-A) in a space of score functions as follows

$$\begin{array}{ll} \underset{s_p \,\in\, \mathcal{S}}{\text{minimize}} & D_{\text{KL}}\big(\, p_{0:T}(\cdot; s_p) \,\|\, q_{0:T}(\cdot; s_q)\,\big) \\ \text{subject to} & \mathbb{E}_{x_0 \sim\, p_0(\cdot; s_p)}\big[\, r_i(x_0)\,\big] \;\geq\; b_i \qquad \text{for } i = 1, \ldots, m. \end{array} \tag{SR-A}$$

We define the 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: $\bar{P}^\star_{\text{ALI}}$ and $\bar{D}^\star_{\text{ALI}}$. In general, Problem (SR-A) is not guaranteed to be convex, since the path-wise KL divergence (3) involves an expectation taken over the backward process $p_{0:T}(\cdot)$. Nevertheless, the path-wise KL divergence is convex in the entire path space $\{p_{0:T}(\cdot)\}$, and constraints are linear. Hence, when the score function class $\mathcal{S}$ is expressive enough to induce any path distribution, we establish strong duality for Problem (SR-A) in Theorem 2.

Theorem 2 (Strong duality)**.** *Let Assumption 1 hold for some $s \in \mathcal{S}$. If any path distribution $p_{0:T}(\cdot)$ can be induced by a score function $s_p \in \mathcal{S}$, then Problem* (SR-A) *is strongly dual, i.e., $\bar{P}^\star_{\text{ALI}} = \bar{D}^\star_{\text{ALI}}$.*

See Appendix C.4 for the proof. It is mild to assume the score function class is expressive, as diffusion models typically employ overparameterized networks (e.g., U-Nets or transformers) in practice. Motivated by strong duality, we propose a dual-based method for solving Problem (SR-A), alternating between minimizing the Lagrangian via gradient descent and maximizing the dual function via dual sub-gradient ascent below.

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

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

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:

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

4 Constrained Composition of Multiple Pretrained Models

We provide a characterization of the solution to Problem (UR-C) in Section 4.1, and establish strong duality for diffusion models in Section 4.2, together with a dual-based training algorithm.

4.1 Composition in distribution space

To apply Problem (UR-C) to diffusion models, we first employ Lagrangian duality to derive its solution in distribution space. Let the Lagrangian for Problem (UR-C) be

$$L_{\text{AND}}(p, u, \lambda) \;=\; u + \sum_{i=1}^{m} \lambda_i \left(D_{\text{KL}}(p \,\|\, q^i) - u \right), \tag{9}$$

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

$$D_{\text{AND}}(\lambda) \;:=\; \max_{u,\, p} \; L_{\text{AND}}(p, u, \lambda). \tag{10}$$

Let a solution to Problem (UR-C) be $(p^\star, u^\star)$, and let the optimal value of the objective function be $P^\star_{\text{AND}} = u^\star$. Let an optimal dual variable be $\lambda^\star \in \operatorname{argmax}_{\lambda \geq 0} D_{\text{AND}}(\lambda)$, and the optimal value of the dual function be $D^\star_{\text{AND}} := D_{\text{AND}}(\lambda^\star)$. For any $\lambda > 0$, we define the tilted product distribution $q^{(\lambda)}_{\text{AND}}$ as a product of m tilted distributions $\{q^i\}_{i=1}^m$:

$$q^{(\lambda)}_{\text{AND}}(\cdot) \;=\; \frac{1}{Z_{\text{AND}}(\lambda)} \prod_{i=1}^{m} \left(q^i(\cdot) \right)^{\frac{\lambda_i}{\mathbf{1}^\top \lambda}} \tag{11}$$

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

In the distribution space, Problem (UR-C) is a convex optimization problem, since the sub-level set of the KL divergence is convex. Again, we apply strong duality in convex optimization [4] to characterize the solution to Problem (UR-C) in Theorem 3. Moreover, it is ready to formulate the constrained composition problem (UR-C) as an unconstrained problem by specializing the dual variable to a solution to the dual problem.

Assumption 2 (Feasibility)**.** *There exists a pair (p, u) such that $D_{\text{KL}}(p \,\|\, q^i) < u$ for all $i = 1, \ldots, n$.*

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

$$\underset{p}{\text{minimize}} \;\; D_{\text{KL}}\left(p \,\|\, q^{(\lambda^\star)}_{\text{AND}} \right) \tag{12}$$

where $\lambda^\star$ is the optimal dual variable, and the dual function has the explicit form, $D(\lambda) = -\log Z_{\text{AND}}(\lambda)$. Furthermore, the optimal solution of (12) is given by

$$p^\star \;=\; q^{(\lambda^\star)}_{\text{AND}}. \tag{13}$$

See Appendix C.5 for proof. The distribution $q^{(\lambda)}_{\text{AND}} \propto \prod_{i=1}^m \left(q^i(\cdot) \right)^{\frac{\lambda_i}{\mathbf{1}^\top \lambda}}$ allows sampling from a weighted product of m distributions $\{q^i\}_{i=1}^m$, where the parameters $\{\lambda_i / \mathbf{1}^\top \lambda\}_{i=1}^m$ weight the importance of each distribution. The geometric mean is a special case when all λ_i are equal [1].

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

4.2 Product composition of diffusion models

We introduce diffusion models into Problem (UR-C) by representing p and q^i as two diffusion models: $p(x_0; s_p)$ and $q^i(x_0; s^i_q)$, with score functions s_p and s^i_q, respectively. The point-wise KL divergence naturally measures the closeness of the final sampling distributions we care about. Hence, we instantiate Problem (UR-C) in a space of score functions as follows

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

We define the 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, we introduce the primal and dual values $\bar{P}^\star_{\text{AND}}$ and $\bar{D}^\star_{\text{AND}}$. Although Problem (SR-C) is non-convex,

since the point-wise KL divergence (4) involves an expectation taken over the backward process $p_{0:T}(\cdot)$. Nevertheless, the point-wise KL divergence is convex in the final distribution space. Hence, when the score function class $\mathcal{S}$ is expressive enough to induce any path distribution (hence any final distribution), we establish strong duality for Problem (SR-C) in Theorem 4.

Theorem 4 (Strong duality)**.** *Let Assumption 2 hold for some $s \in \mathcal{S}$. If any path distribution $p_{0:T}(\cdot)$ can be induced by a score function $s_p \in \mathcal{S}$, then Problem (SR-C) is strongly dual, i.e., $\bar{P}^\star_{\text{AND}} = \bar{D}^\star_{\text{AND}}$.*

See Appendix C.6 for proof. It is mild to assume the score function class is expressive, as diffusion models typically employ overparameterized networks (e.g., U-Nets or transformers) in practice. To solve Problem (SR-C), similar to the one in Section 3.2, we apply a dual-based approach below.

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

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

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

$$\lambda_i^{(n+1)} = \left[\lambda_i^{(n)} + \eta \left(D_{\text{KL}}\big(p(x_0; s^{(n+1)}) \,\|\, q^i(x_0; s_q^i)\big) - u\right)\right]_+ \quad \text{for } i = 1, \ldots, m.$$

It is nontrivial to compute the point-wise KL divergence in the Lagrangian $\bar{L}_{\text{AND}}(s_p, \lambda^{(n)})$ and the constraint violations above. Recall that Lemma 2 gives us a way to compute the point-wise KL: $D_{\text{KL}}(p(x_0; s) \,\|\, q(x_0; s_q^i))$. However, it requires the functions s and s_q^i each to be a valid score function for some forward process. Indeed, this is the case for s_q^i, since it is a pretrained model where it would have been trained to approximate the true score of a forward process. Yet, regarding the function s that we are optimizing over, there is no guarantee that any given $s \in \mathcal{S}$ is a valid score function. To address this issue, we introduce Lemma 3 that allows us to minimize the Lagrangian.

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

$$L_{\text{AND}}(s, \lambda) = D_{\text{KL}}\Big(p(x_0; s) \,\|\, q^{(\lambda)}_{\text{AND}}(x_0)\Big) - \log Z_{\text{AND}}(\lambda). \tag{14}$$

Furthermore, a Lagrangian minimizer $s^{(\lambda)} \in \operatorname{argmin}_s L_{\text{AND}}(s, \lambda)$ is given by

$$s^{(\lambda)} \in \operatorname*{argmin}_{s \in \mathcal{S}} \sum_{t=0}^{T} \omega_t \, \mathbb{E}_{x_0 \sim q^{(\lambda)}_{\text{AND}}(\cdot)} \mathbb{E}_{x_t \sim q(x_t | x_0)} \left[\|s(x,t) - \nabla \log q(x_t \,|\, x_0)\|^2\right] \tag{15}$$

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

See Appendix C for proof. With Lemma 3, as long as we can obtain samples from the distribution $q^{(\lambda)}_{\text{AND}}$, we can approximate the expectation in (15) and use gradient-based optimization methods to find a Lagrangian minimizer $s^{(\lambda)}$. To do so, we use annealed Markov Chain Monte Carlo (MCMC) sampling [14], which requires having access to the scores of a sequence of distributions that interpolate smoothly between $q^{(\lambda)}_{\text{AND}}(x_T)$ and $q^{(\lambda)}_{\text{AND}}(x_0)$: $\nabla \log q^{(\lambda)}_{\text{AND}}(x_t) = \sum_{i=1}^m \lambda_i \nabla \log q^i(x_t)$. In alignment, since we don't have these 'intermediate' scores, we cannot employ the approach in Lemma 3. See Appendix B for sampling details.

For the dual update, we evaluate the KL divergence $D_{\text{KL}}(p_0(\cdot; s^{(\lambda)}) \,\|\, p_0(\cdot; s^i))$ between the marginal densities induced by the Lagrangian minimizer $s^{(\lambda)}$ and the individual score functions s^i using Lemma 2, since both are valid score functions.

Remark 2. *In practice, the primal step only yields an approximate Lagrangian minimizer $s^{(\widetilde{\lambda})}(x,t) \approx \nabla \log q^{(\lambda)}_{\text{AND},\, t}(x)$. This results in two sources of error in evaluating the expectations on the RHS of (4):*

$$D_{\text{KL}}(p_0(\cdot\,; s^{(\lambda)}) \,\|\, p_0(\cdot\,; s^i)) = \sum_{t=0}^{T} \widetilde{\omega}_t \, \mathbb{E}_{x \sim p_t(\cdot\,; s^{(\lambda)})} \left[\left\| s^{(\lambda)}(x,t) - s^i(x,t) \right\|_2^2\right] + \epsilon_T \tag{16}$$

The first error caused by not using the exact $s^{(\lambda)}$ in $\left\|s^{(\lambda)}(x,t) - s^i(x,t)\right\|_2^2$. The second error introduced by not evaluating the expectation on correct trajectories given by $x \sim p_t(\cdot\,; s^{(\lambda)})$. However, the second error reduces, if we have a way of sampling from the true product $x_0 \sim q^{(\lambda)}_{\text{AND},\, 0}$, because we can get samples from $p_t(\cdot\,; s^{(\lambda)})$ just by adding Gaussian noise to x_0.

See Appendix F for the detailed algorithm of product composition.

5 Computational Experiments

We demonstrate the effectiveness and merits of our constrained alignment and composition in a series of computational experiments in Section 5.1 and Section 5.2, respectively.

5.1 Alignment of diffusion models with multiple rewards

We extend the AlignProp framework [35] to handle multiple rewards as constraints. We finetune Stable Diffusion v1.5[2] using several widely-used differentiable image quality and aesthetic rewards: aesthetic [39], hps [52], pickscore [23], imagereward [54] and MPS [60]. Since these rewards vary substantially in scale, making it difficult to set constraint levels, we normalize each by computing the average and standard deviation over a number of batches. In all experiments, models are finetuned using LoRA [21]. Experimental settings and hyperparameters are provided in Appendix G.

I. MPS, local contrast, and saturation constraints. A common shortcoming of several off-the-shelf aesthetics, image preference, and quality reward models is their tendency to overfit to specific image characteristics such as saturation and sharp, high-contrast textures; see, for example, images in the first column in Figure 3 (Right). To mitigate this issue, we add regularizers to the reward function to explicitly penalize these characteristics. However, if the regularization weight is not carefully tuned, models may overfit to the regularizers instead of optimizing for the intended reward. As shown in Figure 3, when using equal weights, the MPS reward *decreases* (Left). In contrast, our constrained approach effectively controls multiple undesired artifacts while ensuring none of the rewards are neglected, achieving a near feasible solution at the specified constraint level: a 50% improvement.

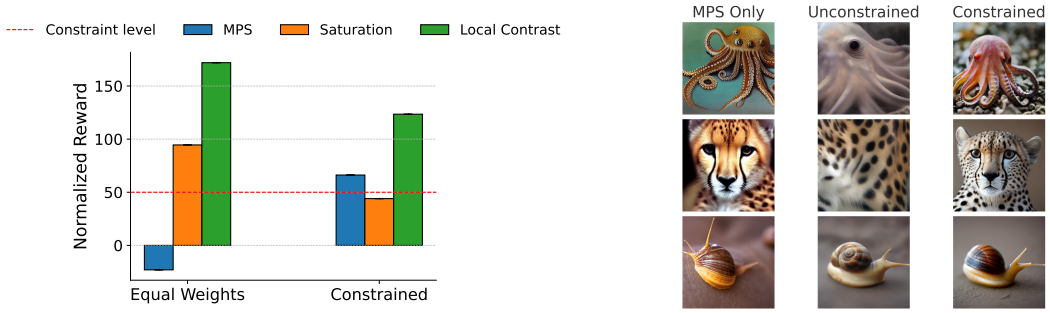

Figure 3: Reward alignment. Stable diffusion is finetuned using one reward that emphasizes aesthetic quality (MPS), and Saturation and Local Contrast as regularizers. Reward values for the equal weights method and our constrained alignment (Left). Images are sampled from the aligned models (Right), and the model trained solely with MPS reward is used for comparison.

II. Multiple aesthetic constraints. When finetuning with multiple rewards, arbitrarily assigning fixed weights can lead to uneven performance across rewards. As shown in Figure 4 (Left), the model tends to overfit one reward while neglecting more challenging ones (e.g., hps). In contrast, constraining all rewards enables the model to improve each reward up to its specified level, including the challenging ones. From Figure 4 (Middle), minimizing the KL divergence subject to these constraints also yields a smaller KL divergence to the pretrained model. Without constraints, overfitting to a subset of rewards causes the model to deviate excessively from the pretrained one, which is undesirable (Right).

5.2 Product composition of diffusion models

In high-dimensional settings such as image generation, obtaining samples from the true product distribution via MCMC and then minimizing the Lagrangian in (15) to estimate the true product score function is prohibitively expensive. To address this, we employ a surrogate for the true score both for sampling and for computing the KL divergence, as detailed in Appendix G.

I. Composing models finetuned on different rewards. We investigate the composition of several finetuned variants of the same base model, where each model is trained with LoRA a different reward

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

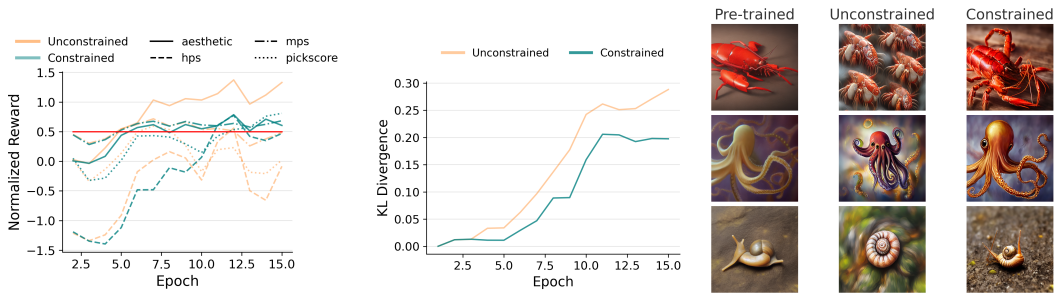

Figure 4: Reward alignment. Stable diffusion is finetuned using multiple image quality/aesthetic rewards. Reward trajectories for the regularization-based method and our constrained alignment during training (Left). KL divergences to the pretrained model (Middle). Images are sampled from the aligned models (Right), and the pretrained model is used for comparison.

function. A key challenge is determining appropriate combination weights: arbitrary choices can lead to undesirable trade-offs and underrepresentation of certain models in the mixture, as evidenced in Figure 5 by drops in up to 30% in some rewards. Our constrained composition provides a principled way to select weights that maintain proximity to each model, improving rewards across all models.

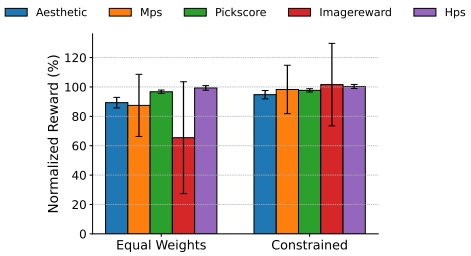

Figure 5: Product composition. Stable diffusion with LoRA is finetuned using different rewards, for equal weighted and product mixtures. 100% represents the reward levels attained by models aligned solely with the individual reward. Higher is better.

	Min. CLIP (↑)	**Min. BLIP** (↑)
Combined Prompting	22.1	0.204
Equal Weights	22.7	0.252
Constrained (Ours)	**22.9**	**0.268**

Table 1: Product composition. We compare our constrained composition with two baselines using minimum CLIP and BLIP scores. The score is averaged over 50 different prompt pairs that are sampled from a list of simple prompts.

II. Concept composition with stable diffusion. Following the setting in [40], we compose two text-to-image diffusion models, each conditioned on a different input prompt. We apply the constrained composition (SR-C) to determine the optimal weights for composing two models, and compare against the baseline that uses equal weights. Closeness to each model encourages faithful representation of both concepts in the images generated by the composed model, as reflected by improved text-to-image similarity metrics: CLIP [20] and BLIP [25], which are reported in Table 1. We compute similarity scores between the generated images and each of the two prompts and compare their minimum values. We also include a baseline where images are generated from a single combined prompt containing both inputs. Images from all approaches, along with implementation details and additional experimental results, are provided in Appendix G.

6 Conclusion

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 pretrained model. Theoretically, we characterize the solutions to the constrained alignment and composition problems and design dual-based training algorithms to approximate these solutions. Empirically, we demonstrate our constrained approach on image generation tasks, showing that the aligned or composed models effectively satisfy the specified constraints.

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. Drozdzal, 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] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang. Clap: Learning audio concepts from natural language supervision, 2022.

[16] Y. Fan and K. Lee. Optimizing DDPM sampling with shortcut fine-tuning. In *International Conference on Machine Learning*, pages 9623–9639. PMLR, 2023.

[17] 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.

[18] 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.

[19] Y. Han, M. Razaviyayn, and R. Xu. Stochastic control for fine-tuning diffusion models: Optimality, regularity, and convergence. *arXiv preprint arXiv:2412.18164*, 2024.

[20] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models, 2023.

[31] 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.

[32] S. Lyu. Interpretation and generalization of score matching, 2012.

[33] W. Mou, N. Flammarion, M. J. Wainwright, and P. L. Bartlett. Improved bounds for discretization of langevin diffusions: Near-optimal rates without convexity, 2019.

[34] 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.

[35] M. Prabhudesai, A. Goyal, D. Pathak, and K. Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation, 2024.

[36] M. Prabhudesai, R. Mendonca, Z. Qin, K. Fragkiadaki, and D. Pathak. Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.

[37] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models, 2022.

[43] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, 2021.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] A. Ulhaq and N. Akhtar. Efficient diffusion models for vision: A survey. *arXiv preprint arXiv:2210.09292*, 2022.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] S. Zampini, J. Christopher, L. Oneto, D. Anguita, and F. Fioretto. Training-free constrained generation with stable diffusion models. *arXiv preprint arXiv:2502.05625*, 2025.

[59] H. Zhang and T. Xu. Towards controllable diffusion models via reward-guided exploration, 2023.

[60] S. Zhang, B. Wang, J. Wu, Y. Li, T. Gao, D. Zhang, and Z. Wang. Learning multi-dimensional human preference for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8018–8027, 2024.

[61] Z. Zhang, L. Shen, S. Zhang, D. Ye, Y. Luo, M. Shi, B. Du, and D. Tao. Aligning few-step diffusion models with dense reward difference learning. *arXiv preprint arXiv:2411.11727*, 2024.

[62] H. Zhao, H. Chen, J. Zhang, D. D. Yao, and W. Tang. Scores as actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. *arXiv preprint 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 were we dicuss 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 G. 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 G. 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 G.

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 G.

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 G.

   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` 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.

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: -

    Guidelines:

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

Supplementary Materials for "Composition and Alignment of Diffusion Models using Constrained Learning"

A Limitations and Broader Impact

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

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

B Related Work

Alignment of diffusion models. Our constrained alignment is related to a line of work on finetuning diffusion models. Standard finetuning typically involves optimizing either a task-specific reward that encodes desired properties, or a weighted sum of this reward and a regularization term that encourages closeness to the pretrained model; see [16, 55, 24, 53, 59, 51, 2, 12, 61] for studies using the single reward objective and [45, 62, 47, 46, 36, 17, 19] for those using the weighted sum objective. The former class of single reward-based studies focus exclusively on generating samples with higher rewards, often at the cost of generalization beyond the training data. The latter class introduces a regularization term that regulates the model to be close to the pretrained one, while leaving the trade-off between reward and closeness unspecified; see [44] for their typical pros and cons in practice. There are three key drawbacks to using either the single reward or weighted sum objective: (i) the trade-off between reward maximization and leveraging the utility of the pretrained model is often chosen heuristically; (ii) it is unclear whether the reward satisfies the intended constraints; and (iii) multiple constraints are not naturally encoded within a single reward function. In contrast, we formulate alignment as a constrained learning problem: minimizing deviation from the pretrained model subject to reward constraints. This offers a more principled alternative to existing ad hoc approaches [8, 18]. Our new alignment formulation (i) offers a theoretical guarantee of an optimal trade-off between reward satisfaction and proximity to the pretrained model, and (ii) allows for the direct imposition of multiple reward constraints. We also remark that our constrained learning approach generalizes to finetuning of diffusion models with preference [49, 57, 26].

Composition of diffusion models. Our constrained composition approach is related to prior work on compositional generation with diffusion models. When composing pretrained diffusion models, two widely used approaches are (i) product composition (or conjunction) and (ii) mixture composition (or disjunction). In product composition, it has been observed that the diffusion process is not compositional, e.g., a weighted sum of diffusion models does not generate samples from the product of the individual target distributions [14, 5, 10]. To address this issue, the weighted sum approach has been shown to be effective when combined with additional assumptions or techniques, such as energy-based models [31, 14], MCMC sampling [14], diffusion soup [1], and superposition [40]. However, how to determine optimal weights for the individual models is not yet fully understood. In contrast, we propose a constrained optimization framework for composing diffusion models that explicitly determines the optimal composition weights. Hence, this formulation enables an optimal trade-off among the pretrained diffusion models. Moreover, our constrained composition approach also generalizes to mixture composition, offering advantages over prior work [31, 14, 1, 40].

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 [27, 11, 28], constrained posterior sampling [34], and proximal Langevin dynamics [58]. 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 [22]. 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:

$$D_{\mathrm{KL}}(p_{0:T}(x_{0:T}; s_p)) = D_{\mathrm{KL}}(p(x_{0:T}; s_p)) \tag{17}$$

C.1 Proof of Lemma 1

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

$$p(x_{t-1} \,|\, x_t; s) \;=\; \mathcal{N}\left(\sqrt{\frac{\alpha_{t-1}}{\alpha_t}} x_t \;+\; \beta_t \, s(x_t, t),\, \sigma_t^2 I\right). \tag{18}$$

Using (18) we expand the path-wise KL:

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

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 Guassians 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. □

C.2 Proof of Lemma 2

The proof for Lemma 2 is quite involved, thus we have divided it into multiple parts for readability. In Section C.2.1, we give a few definitions for continuous time diffusion processes. In Section C.2.2, we prove an analogue of Lemma 2 in continuous time. In Section C.2.3, we bound the discretization error ϵ_T incurred when going from continuous time processes to corresponding discretized processes and thus complete the proof. The proofs for all lemmas presented here can be found in Appendix D.

C.2.1 Continuous time preliminaries

Notation Guide: Throughout the proof, we will be dealing with continuous time forward and reverse diffusion processes and their discretized counterparts.

- We denote the continuous time variable $\tau \in [0,1]$ to differentiate it from the discrete time indices $t \in \{0, \cdots, T\}$. $t = 0$ corresponds to $\tau = 1$ and $t = T$ corresponds to $\tau = 0$. [3]
- We denote as $\mathfrak{X}_\tau$ the continuous time reverse DDIM process and X_t as the corresponding discrete time process.
- The forward processes we denote with an additional bar e.g. $\bar{\mathfrak{X}}_\tau, \bar{X}_t$ denote the continuous time and discrete time forward processes respectively.
- Marginal density of continuos time DDIM process with score predictor $s(x,\tau)$ at time τ we denote as: $\mathfrak{p}_\tau(x,s)$.

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

$$d\mathfrak{X}_\tau \;=\; (\frac{\dot{\bar{\alpha}}_\tau}{2\bar{\alpha}_\tau}\mathfrak{X}_\tau + (\frac{\dot{\bar{\alpha}}_\tau}{2\bar{\alpha}_\tau} + \frac{\sigma_\tau^2}{2})s(\mathfrak{X}_\tau,\tau))dt + \sigma_\tau d\mathfrak{B}_\tau, \quad \mathfrak{X}_0 \sim \mathcal{N}(0,I) \tag{19}$$

The variance schedule σ_τ is arbitrary and determines the randomness of the trajectories (e.g. if $\sigma_\tau = 0$ for all τ, then the trajectories will be deterministic). The DDIM generative process (19) induces marginal densities $\mathfrak{p}_\tau(x,s)$ for $\tau \in [0,1]$.

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

$$X_{t-1} \;=\; \sqrt{\frac{\alpha_{t-1}}{\alpha_t}}X_t \;+\; \beta_t\, s(X_t,t) \;+\; \sigma_t\epsilon_t. \tag{20}$$

Up to first order approximation, the discrete time process (20) is the Euler-Maruyama discretization of the continuous time process (19). A uniform discretization of time is assumed, i.e., $\tau = 1 - \frac{t}{T}$ (See [13, Appendix B.1] for the full derivation).

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

$$\bar{\mathfrak{X}}_\tau \;=\; \alpha_\tau\bar{\mathfrak{X}}_0 + \zeta_\tau\bar{\mathfrak{X}}_1. \tag{21}$$

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

If the score predictor $s(x,\tau) = \nabla_x \log \bar{\mathfrak{p}}_\tau(x)$, then the DDIM process (19) has the same marginals as the reference flow (21), i.e., $\mathfrak{p}_\tau(x,s) = \bar{\mathfrak{p}}_\tau(x)$ for $\tau \in [0,1]$. This is assuming proper choice of α_τ, ζ_τ, i.e., $\alpha_\tau = \sqrt{1-\bar{\alpha}_\tau}$, $\zeta_\tau = \sqrt{\bar{\alpha}_\tau}$.

C.2.2 Proof for continuous time

We generalize [32, Theorem 1] to characterize how the KL divergence between the marginals of two continuous time forward processes changes with time.

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

$$\frac{d}{d\tau}D_{\mathrm{KL}}(\bar{\mathfrak{p}}_\tau(\cdot)\,\|\,\bar{\mathfrak{q}}_\tau(\cdot)) \;=\; -\gamma_\tau\dot{\gamma}_\tau D_{\mathrm{F}}(\bar{\mathfrak{p}}_\tau(\cdot)\,\|\,\bar{\mathfrak{q}}_\tau(\cdot)) \tag{22}$$

where $\gamma_\tau = \zeta_\tau/\alpha_\tau$, and $D_{\mathrm{F}}(p\,\|\,q)$ denotes the Fisher divergence.

By integrating the derivative of the KL divergence as given by Lemma 4, we obtain the following continuous-time analogue of Lemma 2, which characterizes the point-wise KL divergence of two continuous time diffusion processes.

[3] For consistency with other works from whom we will utilize some results in our proofs, namely [13, 32], the direction of time we consider in continuous time is reversed compared to discrete time. This does not affect our derivations and results beyond a notation change.

Lemma 5. *Consider two score predictors $s_{\mathfrak{p}}(x,\tau) = \nabla_x \log \bar{\mathfrak{p}}_\tau(x)$, $s_{\mathfrak{q}}(x,\tau) = \nabla_x \log \bar{\mathfrak{q}}_\tau(x)$, where $\bar{\mathfrak{p}}_\tau$, $\bar{\mathfrak{q}}_\tau$ are marginal densities of two reference flows, with the same noise schedule, starting from initial distributions $\bar{\mathfrak{p}}_0$ and $\bar{\mathfrak{q}}_0$, respectively. Then, the point-wise KL divergence between two distributions of the samples generated by running continuous time DDIM (19) with $s_{\mathfrak{p}}$ and $s_{\mathfrak{q}}$ is given by*

$$D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot; s_{\mathfrak{p}}) \,\|\, \mathfrak{q}_0(\cdot; s_{\mathfrak{q}})) \;=\; \int_{\tau=0}^{1} \widetilde{\omega}_\tau \, \mathbb{E}_{x\sim \mathfrak{p}_\tau(\cdot\,; s_p)} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] \tag{23}$$

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

C.2.3 Bounding the discretization error

We now turn to bridging the gap between continuous and discrete times. In [33], they bound this gap which arises from the discretization of the continuous time diffusion process. We will utilize the main result from this paper with a minor modification in that we consider a time-dependent drift term. This is formalized in Lemma 6 which allows us bound the KL divergence between the marginals $p_t(\cdot)$ of the discrete time backward DDIM process and the corresponding marginal $\mathfrak{p}_{t/T}(\cdot)$ of the continuous time backward process.

Lemma 6. *(Modification of Theorem 1 from [33].) Under mild assumptions on the score function (outlined in the proof), the KL divergence between the marginals of the discrete time backward process $p_t(\cdot)$ and continuous time backward process $\mathfrak{p}_{t/T}(\cdot)$ can be bounded as follows:*

$$D_{\mathrm{KL}}(p_t(\cdot; s_p) \,\|\, \mathfrak{p}_{1-t/T}(\cdot; s_p)) \;\le\; \frac{c}{T^2} \tag{24}$$

where c is a constant depending on the assumptions.

Next we need to characterize the sensitivity of the KL divergence to perturbations in the first and second arguments so that we can apply Lemma 6.

Lemma 7. *Assume $M := \max_x \left|\log(\frac{\mathfrak{p}_0(\cdot;s_p)}{\mathfrak{p}_0(\cdot;s_q)})\right|$ 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 $\epsilon_1(T)$:*

$$|D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot; s_p) \,\|\, \mathfrak{q}_0(\cdot; s_q)) - D_{\mathrm{KL}}(p_0(\cdot; s_p) \,\|\, q_0(\cdot; s_q))| \;\le\; \epsilon_1(T), \tag{25}$$

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

And lastly, we need to characterize the discretization error in going from a integral over continuous time to a sum over discrete time steps.

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

$$B_1 \;:=\; \sup_{x,\tau} \|s_p(x,\tau) - s_q(x,\tau)\|_2 \tag{26}$$

$$B_2 \;:=\; \sup_{x,\tau} \left\| \frac{d}{d\tau}(s_p(x,\tau) - s_q(x,\tau)) \right\|_2 \tag{27}$$

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

$$\left| \int_{\tau=0}^{1} \widetilde{\omega}_\tau \, \mathbb{E}_{x\sim \mathfrak{p}_\tau(\cdot\,; s_p)} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] - \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x\sim p_t(\cdot\,; s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \right| \;\le\; \epsilon_2(T) \tag{28}$$

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

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

$$D_{\mathrm{KL}}(p_0(\cdot\,; s_p) \,\|\, q_0(\cdot\,; s_q)) \;=\; \sum_{t=0}^{T} \widetilde{\omega}_t \, \mathbb{E}_{x\sim p_t(\cdot\,; s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \;+\; \epsilon_T \tag{29}$$

where $|\epsilon_T| \le \epsilon_1(T) + \epsilon_2(T) = O(1/T)$. (We abuse notation to denote $\frac{1}{T}\widetilde{\omega}_{t/T}$ as $\widetilde{\omega}_t$ in (29) and in the main paper.)

C.3 Proof of Theorem 1

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

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

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

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

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

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

C.4 Proof of Theorem 2

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

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

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

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

Since the score function class $\mathcal{S}$ is expressive enough, any path distribution $p_{0:T}(\cdot)$ can be represented as $p_{0:T}(\cdot; s_p)$ with some $s_p \in \mathcal{S}$; and vice versa. Thus, we can express $p^\star_{0:T}(\cdot)$ as $p_{0:T}(\cdot; s^\star_p)$ with some $s^\star_p \in \mathcal{S}$. We also note that the dual functions $\bar{D}_{\text{ALI}}(\lambda)$ in the path and score function spaces are the same. Hence, the dual value for Problem (SR-A) remains to be $\bar{D}_{\text{ALI}}(\lambda^\star)$. Thus, $(s^\star_p, \lambda^\star)$ is a saddle point of the Lagrangian $\bar{L}_{\text{ALI}}(s_p, \lambda) := L_{\text{ALI}}(p_{0:T}(\cdot; s_p), \lambda)$,

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

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

C.5 Proof of Theorem 3

Proof. By the definition,

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

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

$$p^\star(\cdot;\lambda^\star) \;\propto\; q^{(\lambda^\star)}_{\text{AND}}(\cdot)$$

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

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

$$p^\star(\cdot; \lambda, \mu) \;\propto\; q^{(\lambda)}_{\text{AND}}(\cdot)$$

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

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

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

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. □

C.7 Proof of Lemma 3

Proof. From section C.5, we recall:

$$L_{\text{AND}}(p, u; \lambda) \;=\; u + \sum_{i=1}^m \lambda_i \left(D_{\text{KL}}(p \,\|\, q^{(\lambda)}_{\text{AND}}) - u \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \tag{30}$$

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 (30) that:

$$L_{\text{AND}}(p_0(\cdot; s), u; \lambda) \;=\; u + \sum_{i=1}^m \lambda_i \left(D_{\text{KL}}(p_0(\cdot; s) \,\|\, q^{(\lambda)}_{\text{AND},0}(\cdot)) - u \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \tag{31}$$

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

$$L_{\text{AND}}(p(\cdot; s); \lambda) \;=\; L_{\text{AND}}(s, \lambda) \;=\; D_{\text{KL}}(p_0(\cdot; s) \,\|\, q^{(\lambda)}_{\text{AND},0}) - \log Z_{\text{AND}}(\lambda). \tag{32}$$

The second term $\log Z_{\text{AND}}(\lambda)$ does not depend on s, thus it suffices to minimize $D_{\text{KL}}(p_0(\cdot; s) \,\|\, q^{(\lambda)}_{\text{AND},0})$ to find the Lagrangian minimizer which we call $s^{(\lambda)}$. The KL is minimized when $p_0(\cdot; s^{(\lambda)}) = q^{(\lambda)}_{\text{AND},0}$. If we have access to samples from $q^{(\lambda)}_{\text{AND},0}$, we can fit s to $q^{(\lambda)}_{\text{AND},0}$ by optimizing the Denoising score matching objective similar to Equation (1) in [43]:

$$L_{\text{sm}}(s, \lambda) \;=\; \sum_{t=0}^T \omega_t \, \mathbb{E}_{x_0 \sim q^{(\lambda)}_{\text{AND}}(\cdot)} \mathbb{E}_{x_t \sim q(x_t \,|\, x_0)} \left[\| s(x, t) - \nabla \log q(x_t \,|\, x_0) \|^2 \right] \tag{33}$$

From [43] we know that given sufficient data and predictor capacity of s we have $\operatorname{argmin}_s L_{\text{sm}}(s, \lambda) \simeq q^{(\lambda)}_{\text{AND},0}$ which concludes the proof. □

D Additional Proofs

We provide detailed proofs for all lemmas in Section C.2.

D.1 Proof of Lemma 4

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

$$\bar{\mathfrak{Y}}_\tau \;:=\; \frac{1}{\alpha_\tau}\bar{\mathfrak{X}}_\tau \;=\; \bar{\mathfrak{X}}_1 + \gamma_\tau \bar{\mathfrak{X}}_0 \tag{34}$$

where $\gamma_\tau := \zeta_\tau/\alpha_\tau$. Denote by $\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$, the marginal density of $\bar{\mathfrak{Y}}_\tau$ when $\bar{\mathfrak{X}}_1 \sim \mathfrak{p}_1$ and similarly $\widetilde{q}_t(\bar{\mathfrak{Y}}_\tau)$, the marginal density of $\bar{\mathfrak{Y}}_\tau$ when $\mathfrak{X}_1 \sim \mathfrak{q}_1$. Now we generalize Theorem 1 from [32] to show that (22) holds for $\widetilde{\mathfrak{p}}_\tau, \widetilde{\mathfrak{q}}_\tau$. Their Theorem is for the specific case of $\gamma_\tau = \sqrt{1-t}$.[4]

We now present Lemmas 9 and 10 which we will need in the remainder of the proof.

Lemma 9. *For density $\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$ as defined in Theorem 1, the following identity holds:*

$$\frac{d}{dt}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) = \gamma_\tau\dot{\gamma}_\tau\Delta_{\bar{\mathfrak{Y}}_\tau}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau). \tag{35}$$

Proof. Proof of Lemma 9. We start with $\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$ which is the convolution of a Gaussian distribution with $\mathfrak{p}_1(\bar{\mathfrak{X}}_1)$:

$$\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) \;=\; \int_{\bar{\mathfrak{X}}_1} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right)\mathfrak{p}_1(\bar{\mathfrak{X}}_1), \tag{36}$$

Taking the derivative we have:

$$\begin{aligned}\frac{d}{dt}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) \;=\; &\int_{\bar{\mathfrak{X}}_1} \frac{\dot{\gamma}_\tau\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{\gamma_\tau^3}\frac{1}{(2\pi\gamma_\tau^2)^{d/2}}\exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right)\mathfrak{p}_1(\bar{\mathfrak{X}}_1)\\ &-\int_{\bar{\mathfrak{X}}_1}\frac{d}{\gamma_\tau}\frac{\dot{\gamma}_\tau}{(2\pi\gamma_\tau^2)^{d/2}}\exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right)\mathfrak{p}_1(\bar{\mathfrak{X}}_1).\end{aligned} \tag{37}$$

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

$$\nabla_{\bar{\mathfrak{Y}}_\tau}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) \;=\; -\int_{\bar{\mathfrak{X}}_1}\frac{\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1}{\gamma_\tau^2}\frac{1}{(2\pi\gamma_\tau^2)^{d/2}}\exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right)\mathfrak{p}_1(\bar{\mathfrak{X}}_1). \tag{38}$$

Taking the divergence of the gradient, we have:

$$\begin{aligned}\Delta_{\bar{\mathfrak{Y}}_\tau}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) \;=\; &\int_{\bar{\mathfrak{X}}_1}\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{\gamma_\tau^4}\frac{1}{(2\pi\gamma_\tau^2)^{d/2}}\exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right)\mathfrak{p}_1(\bar{\mathfrak{X}}_1),\\ &-\int_{\bar{\mathfrak{X}}_1}\frac{d}{\gamma_\tau^2}\frac{1}{(2\pi\gamma_\tau^2)^{d/2}}\exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right)\mathfrak{p}_1(\bar{\mathfrak{X}}_1).\end{aligned} \tag{39}$$

Comparing Equations (37) and (39) proves the result.

□

Lemma 10. *For any positive valued function $f(x) : \mathbb{R}^d \to \mathbb{R}$ whose gradient $\nabla_x f$ and Laplacian $\Delta_x f$ are well defined, we have the identity*

$$\frac{\Delta_x f(x)}{f(x)} \;=\; \Delta_x \log f(x) + \left\|\nabla_x \log f(x)\right\|^2. \tag{40}$$

[4] Just to avoid any confusion, in [32], 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.

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

$$\begin{aligned} D_{\mathrm{F}}(p \,\|\, q) &= \int_x p(x) \left\|\nabla \log p(x) - \nabla \log q(x)\right\|^2 dx \\ &= \int_x p(x) \left\| \frac{\nabla p(x)}{p(x)} - \frac{\nabla q(x)}{q(x)} \right\|^2 dx \\ &= \int_x p(x) \left(\left\| \frac{\nabla p(x)}{p(x)} \right\|^2 + \left\| \frac{\nabla q(x)}{q(x)} \right\|^2 - 2\frac{\nabla p(x)^\top \nabla q(x)}{p(x) q(x)} \right) dx \end{aligned} \tag{41}$$

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

$$\begin{aligned} \int_{x\in\Omega} \nabla p(x)^\top \frac{\nabla q(x)}{q(x)} dx &= \int_{x\in\Omega} \nabla p(x)^\top (\nabla \log q(x)) dx \\ &= -\int_{x\in\Omega} p(x) \Delta \log q(x) dx + \int_\Gamma p(x) (\nabla \log q(x)^\top \widehat{n}) d\Gamma \end{aligned} \tag{42}$$

Assuming that both $p(x)$ and $q(x)$ are smooth and fast-decaying, the boundary term in (42) vanishes. Then we can combine (41) and (42) to write:

$$D_{\mathrm{F}}(p \,\|\, q) = \int_x p(x) \left(\|\nabla \log p(x)\|^2 + \|\nabla \log q(x)\|^2 + 2\Delta_x \log q(x) \right) dx \tag{43}$$

Returning to our distributions $\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$ and $\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)$ we can rewrite (43) as:

$$D_{\mathrm{F}}(\widetilde{\mathfrak{p}}_\tau(\cdot) \,\|\, \widetilde{\mathfrak{q}}_\tau(\cdot)) = \int_{\mathfrak{Y}_\tau} \widetilde{\mathfrak{p}}_\tau(\mathfrak{Y}_\tau) \left(\left\|\nabla \log \widetilde{\mathfrak{p}}_\tau(\mathfrak{Y}_\tau)\right\|^2 + \|\nabla \log \widetilde{\mathfrak{q}}_\tau(\mathfrak{Y}_\tau)\|^2 + 2\Delta_{\mathfrak{Y}_\tau} \log \widetilde{\mathfrak{q}}_\tau(\mathfrak{Y}_\tau) \right) d\mathfrak{Y}_\tau \tag{44}$$

For conciseness in notation, we drop references to variables $\bar{\mathfrak{Y}}_\tau$ and $\bar{\mathfrak{X}}_\tau$ in the integration, the density functions, and the operators whenever this does not lead to ambiguity. We start by applying Lemma 10 to Equation (43):

$$\begin{aligned} D_{\mathrm{F}}(\widetilde{\mathfrak{p}}_\tau \,\|\, \widetilde{\mathfrak{q}}_\tau) &= \int \widetilde{\mathfrak{p}}_\tau \left(|\nabla \log \widetilde{\mathfrak{p}}_\tau|^2 + |\nabla \log \widetilde{\mathfrak{q}}_\tau|^2 + 2\Delta \log \widetilde{\mathfrak{q}}_\tau \right), \\ &= \int \widetilde{\mathfrak{p}}_\tau \left(|\nabla \log \widetilde{\mathfrak{p}}_\tau|^2 + \frac{\Delta \widetilde{\mathfrak{q}}_\tau}{\widetilde{\mathfrak{q}}_\tau} + \Delta \log \widetilde{\mathfrak{q}}_\tau \right). \end{aligned} \tag{45}$$

Next, we expand the derivative of the KL divergence:

$$\frac{d}{d\tau} D_{\mathrm{KL}}(\widetilde{\mathfrak{p}}_\tau \,\|\, \widetilde{\mathfrak{q}}_\tau) = \int \frac{d}{d\tau} \widetilde{\mathfrak{p}}_\tau \log \frac{\widetilde{\mathfrak{p}}_\tau}{\widetilde{\mathfrak{q}}_\tau} + \int \widetilde{\mathfrak{p}}_\tau \frac{d}{d\tau} \log \widetilde{\mathfrak{p}}_\tau - \int \widetilde{\mathfrak{p}}_\tau \frac{d}{d\tau} \log \widetilde{\mathfrak{q}}_\tau.$$

We can eliminate the second term by exchanging integration and differentiation of τ:

$$\int \widetilde{\mathfrak{p}}_\tau \frac{d}{d\tau} \log \widetilde{\mathfrak{p}}_\tau = \int \frac{d\widetilde{\mathfrak{p}}_\tau}{d\tau} = \frac{d}{d\tau} \int \widetilde{\mathfrak{p}}_\tau = 0.$$

As a result, there are three remaining terms in computing $\frac{d}{d\tau} D_{KL}(\widetilde{\mathfrak{p}}_\tau \| \widetilde{\mathfrak{q}}_\tau)$, which we can further substitute using Lemma 9, as:

$$\begin{aligned} \frac{d}{d\tau} D_{\mathrm{KL}}(\widetilde{\mathfrak{p}}_\tau \,\|\, \widetilde{\mathfrak{q}}_\tau) &= \int \frac{d}{d\tau} \widetilde{\mathfrak{p}}_\tau \log \widetilde{\mathfrak{p}}_\tau - \int \frac{d}{d\tau} \widetilde{\mathfrak{p}}_\tau \log \widetilde{\mathfrak{q}}_\tau - \int \widetilde{\mathfrak{p}}_\tau \frac{d}{d\tau} \log \widetilde{\mathfrak{q}}_\tau, \\ &= \gamma_\tau \dot{\gamma}_\tau \left(\int \Delta \widetilde{\mathfrak{p}}_\tau \log \widetilde{\mathfrak{p}}_\tau - \int \Delta \widetilde{\mathfrak{p}}_\tau \log \widetilde{\mathfrak{q}}_\tau - \int \widetilde{\mathfrak{p}}_\tau \frac{\Delta \widetilde{\mathfrak{q}}_\tau}{\widetilde{\mathfrak{q}}_\tau} \right). \end{aligned} \tag{46}$$

Using integration by parts, the first term in (46) is changed to:

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

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

$$\int \nabla\widetilde{\mathfrak{p}}_\tau^T \nabla \log \widetilde{\mathfrak{p}}_\tau \;=\; \int \widetilde{\mathfrak{p}}_\tau (\nabla \log \widetilde{\mathfrak{p}}_\tau)^T \nabla \log \widetilde{\mathfrak{p}}_\tau \;=\; \int \widetilde{\mathfrak{p}}_\tau |\nabla \log \widetilde{\mathfrak{p}}_\tau|^2.$$

The second term in (46) can be manipulated similarly, by first using integration by parts to get:

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

Applying integration by parts again to $\nabla\widetilde{\mathfrak{p}}_\tau^T \nabla \log \widetilde{\mathfrak{q}}_\tau$, we have:

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

The limits at the boundary values are all zero due to the smoothness and fast decay properties of $\widetilde{\mathfrak{p}}_\tau(\vec{y})$. Now collecting all terms, we have $\int \widetilde{\mathfrak{p}}_\tau \log \widetilde{\mathfrak{p}}_\tau \;=\; -\int \widetilde{\mathfrak{p}}_\tau |\nabla \log \widetilde{\mathfrak{p}}_\tau|^2$ and $\int \widetilde{\mathfrak{p}}_\tau \log \widetilde{\mathfrak{q}}_\tau = \int \widetilde{\mathfrak{p}}_\tau \Delta \log \widetilde{\mathfrak{q}}_\tau$. Thus (46) becomes:

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

Combining with (45), this leads to the following:

$$\frac{d}{d\tau} D_{\mathrm{KL}}(\widetilde{\mathfrak{p}}_\tau \,\|\, \widetilde{\mathfrak{q}}_\tau) \;=\; -\gamma_\tau \dot{\gamma}_\tau D_{\mathrm{F}}(\widetilde{\mathfrak{p}}_\tau \,\|\, \widetilde{\mathfrak{q}}_\tau). \tag{47}$$

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

$$D_{\mathrm{KL}}(\widetilde{\mathfrak{p}}_\tau(\cdot) \,\|\, \widetilde{\mathfrak{q}}_\tau(\cdot)) = \int \widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) \log \frac{\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)}{\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)} d\bar{\mathfrak{Y}}_\tau = \int \mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau) \log \frac{\mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)}{\mathfrak{q}_\tau(\bar{\mathfrak{X}}_\tau)} d\bar{\mathfrak{X}}_\tau = D_{\mathrm{KL}}(\mathfrak{p}_\tau(\cdot) \,\|\, \mathfrak{q}_\tau(\cdot)) \tag{48}$$

And for Fisher Divergence we can write

$$\begin{aligned} D_{\mathrm{F}}(\widetilde{\mathfrak{p}}_\tau(\cdot) \,\|\, \widetilde{\mathfrak{q}}_\tau(\cdot)) \;&=\; \int \widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) \left\| \frac{\nabla \widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)}{\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)} - \frac{\nabla \widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)}{\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)} \right\|^2 d\bar{\mathfrak{Y}}_\tau \\ &=\; \int \mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau) \left\| \frac{\nabla \mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)}{\mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)} - \frac{\nabla \mathfrak{q}_\tau(\bar{\mathfrak{X}}_\tau)}{\mathfrak{q}_\tau(\bar{\mathfrak{X}}_\tau)} \right\|^2 d\bar{\mathfrak{X}}_\tau \\ &=\; D_{\mathrm{F}}(\mathfrak{p}_\tau(\cdot) \,\|\, \mathfrak{q}_\tau(\cdot)) \end{aligned} \tag{49}$$

Thus we can replace the divergences in (47) with those of the non-scaled distribution, which concludes the proof. □

D.2 Proof of Lemma 5

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

$$\begin{aligned} D_{\mathrm{KL}}(\mathfrak{p}_1(\cdot)\,\|\,\mathfrak{q}_1(\cdot)) &= D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot)\,\|\,\mathfrak{q}_0(\cdot)) - \int_{\tau=0}^{1} \dot{\gamma_\tau}\gamma_\tau D_{\mathrm{F}}(\widetilde{\mathfrak{p}}_\tau(\cdot)\,\|\,\widetilde{\mathfrak{q}}_\tau(\cdot))d\tau \\ &= -\int_{\tau=0}^{1} \dot{\gamma_\tau}\gamma_\tau \mathbb{E}_{x\sim\widetilde{\mathfrak{p}}_\tau}\left[\left\|\nabla \log \widetilde{\mathfrak{p}}_\tau(x) - \nabla \log \widetilde{\mathfrak{q}}_\tau(x)\right\|_2^2\right] d\tau \\ &= -\int_{\tau=0}^{1} \dot{\gamma_\tau}\gamma_\tau \mathbb{E}_{x\sim\widetilde{\mathfrak{p}}_\tau}\left[\left\|s_p(x,\tau) - s_q(x,\tau)\right\|_2^2\right] d\tau \\ &= \int_{\tau=0}^{1} \frac{\dot{\alpha_\tau}}{\alpha_\tau^3} \mathbb{E}_{x\sim\widetilde{\mathfrak{p}}_\tau}\left[\left\|s_p(x,\tau) - s_q(x,\tau)\right\|_2^2\right] d\tau \end{aligned} \tag{50}$$

In the second line we used the fact that $\mathfrak{p}_0(\cdot) = \mathfrak{q}_0(\cdot) = \mathcal{N}(0, I)$, therefore $D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot)||\mathfrak{q}_0(\cdot)) = 0$. 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 $\alpha_\tau^2 + \zeta_\tau^2 = 1$:

$$\begin{aligned} \dot{\gamma_\tau}\gamma_\tau &= \frac{d}{d\tau}\left(\frac{\zeta_\tau}{\alpha_\tau}\right)\frac{\zeta_\tau}{\alpha_\tau} \\ &= \frac{\dot{\zeta_\tau}\zeta_\tau\alpha_\tau - \dot{\alpha}_\tau\zeta_\tau^2}{\alpha_\tau^3} \\ &= \frac{-\dot{\alpha}_\tau\alpha_\tau^2 - \dot{\alpha}_\tau(1-\alpha_\tau^2)}{\alpha_\tau^3} \\ &= -\frac{\dot{\alpha_\tau}}{\alpha_\tau^3} \end{aligned} \tag{51}$$

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

D.3 Proof of Lemma 6

Proof. In [33] they prove this result assuming a drift term that only depends on x. For our modification, we begin by defining the time-dependent drift $b_\tau : \mathbb{R}^d \to \mathbb{R}^d$ of the diffusion process (19) as

$$b_\tau(x) := \left(\frac{\dot{\alpha}_\tau}{2\alpha_\tau}x + \left(\frac{\dot{\alpha}_\tau}{2\alpha_\tau} + \frac{\sigma_\tau^2}{2}\right)s(x,\tau)\right).$$

Assumption 3. *The drift $b_\tau(\cdot)$ satisfies the following properties for all times $\tau \in [0,1]$,*

1. **Lipschitz drift term.** *There is a finite constant L_1 such that*

$$\|b_\tau(x) - b_\tau(y)\|_2 \;\le\; L_1\|x-y\|_2 \quad \textit{for all } x,y \in \mathbb{R}^d. \tag{52}$$

2. **Smooth drift term.** *There is a finite constant L_2 such that*

$$\|\nabla b_\tau(x) - \nabla b_\tau(y)\|_\infty \;\le\; L_2\|x-y\|_2 \quad \textit{for all } x,y \in \mathbb{R}^d. \tag{53}$$

3. **Distant dissipativity.** *There exist strictly positive constants μ, β such that*

$$\langle b_\tau(x), x\rangle \;\le\; -\mu\|x\|_2^2 + \beta \quad \textit{for all } x \in \mathbb{R}^d. \tag{54}$$

4. **Time-continuous drift term.** *There is a finite constant L_3 such that*

$$\left\|\frac{\partial b_\tau(x)}{\partial \tau}\right\|_2 \;\le\; L_3 \quad \textit{for all } x \in \mathbb{R}^d. \tag{55}$$

There is an additional assumption in [33] on the smoothness of the initial densities of the continuous and discrete processes. In our case both are the standard Gaussian which satisfies the assumption. We do not provide the whole proof here as it would consist of almost the entirety of [33]. We focus on a

small part of the proof, that is the only part that changes when we use a time-dependent drift $b_\tau(x)$ as opposed to [33] where they assume a time-independent drift $b(x)$.

Consistent with their notation, we define a continuous time diffusion process:

$$dX_\tau \;=\; b_\tau(X_\tau)d\tau \;+\; dB_\tau, \tag{56}$$

and its Euler-Maruyama discretization parameterized by step size $\eta > 0$ (In our case $\eta = \frac{1}{T}$):

$$\widehat{X}_{(k+1)\eta} \;=\; \widehat{X}_{k\eta} \;+\; \eta b_{k\eta}(\widehat{X}_{k\eta}) \;+\; \sqrt{\eta}\xi_k, \qquad \xi_k \sim \mathcal{N}(0, I) \tag{57}$$

Furthermore they construct a continuous time stochastic process over the interval $\tau \in [\eta, (k+1)\eta]$ that interpolates (57):

$$\widehat{X}_\tau \;:=\; \widehat{X}_{k\eta} \;+\; \int_0^{\tau - k\eta} b_{k\eta}(\widehat{X}_{k\eta})ds \;+\; \int_{k\eta}^{\tau} d\widehat{B}_s \tag{58}$$

Then they prove that the densities of the two continuous time processes given by (56),(58) denoted as π_τ and $\widehat{\pi}_\tau$ respectively satisfy the following (Lemma 2 from [33]):

$$\frac{d}{d\tau} D_{\mathrm{KL}}(\widehat{\pi}_\tau \,\|\, \pi_\tau) \;\le\; \frac{1}{2} \int_{\mathbb{R}^d} \widehat{\pi}_\tau(x) \left\| \widehat{b}_\tau(x) - b_\tau(x) \right\|_2^2 dx. \tag{59}$$

where $\widehat{b}_\tau(x) := \mathbb{E}\left[b_{k\eta}(\widehat{X}_{k\eta}) | \widehat{X}_\tau = x \right]$ where the expectation is over the process (58). Then they proceed to bound the norm inside the integral. The next equation based on equation (18) from [33] is where the time dependence of the drift term in our case enters the picture.

$$\begin{aligned}
\widehat{b}_\tau(x) - b_\tau(x) \;&=\; \mathbb{E}\left[b_{k\eta}(\widehat{X}_{k\eta}) \,|\, \widehat{X}_\tau = x \right] - b_\tau(x) \\
&=\; \mathbb{E}\left[b_{k\eta}(\widehat{X}_{k\eta}) | \widehat{X}_\tau = x \right] - \left(b_{k\eta}(x) + \frac{\partial b_\tau(x)}{\partial \tau}(\tau - k\eta) + O((\tau - k\eta)^2) \right) \\
&=\; \mathbb{E}\left[b_{k\eta}(\widehat{X}_{k\eta}) - b_{k\eta}(\widehat{X}_\tau) | \widehat{X}_\tau = x \right] - \left. \frac{\partial b_\tau(x)}{\partial \tau} \right|_{\tau = k\eta} (\tau - k\eta) + O((\tau - k\eta)^2)
\end{aligned} \tag{60}$$

They prove that the first term in (60) is $O(\eta)$ and from our additional time-continuity requirement for the drift in Assumption 3, the second term is also $O(\eta)$. (Note that $\tau \in [k\eta, (k+1)\eta]$ thus $(\tau - k\eta)$ can be at most η). With this, the rest of the proof from [33] goes through. □

D.4 Proof of Lemma 7

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

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

$$\begin{aligned}
|D_{\mathrm{KL}}(\widehat{\pi} \,\|\, \rho) - D_{\mathrm{KL}}(\pi \,\|\, \rho)| \;&=\; \int \widehat{\pi} \log\left(\frac{\widehat{\pi}}{\rho}\right) - \int \pi \log\left(\frac{\pi}{\rho}\right) + \int (\widehat{\pi} \log \pi - \widehat{\pi} \log \pi) \\
&=\; D_{\mathrm{KL}}(\widehat{\pi} \,\|\, \pi) + \int (\widehat{\pi} - \pi) \log\left(\frac{\pi}{\rho}\right) \\
&\le\; D_{\mathrm{KL}}(\widehat{\pi} \,\|\, \pi) + \max\left(\left| \log\left(\frac{\pi}{\rho}\right) \right| \right) \int |\widehat{\pi} - \pi| \\
&=\; D_{\mathrm{KL}}(\widehat{\pi} \,\|\, \pi) + 2 \log M \, d_{\mathrm{TV}}(\widehat{\pi}, \pi)
\end{aligned} \tag{61}$$

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

$$\begin{aligned}
|D_{\text{KL}}(\widehat{\pi} \,\|\, \widehat{\rho}) - D_{\text{KL}}(\widehat{\pi} \,\|\, \rho)| \;&=\; \left| \int \widehat{\pi} \log\left(\frac{\widehat{\pi}}{\widehat{\rho}}\right) - \int \widehat{\pi} \log\left(\frac{\widehat{\pi}}{\rho}\right) \right| \\
&=\; -\int \widehat{\pi} \log\left(\frac{\widehat{\rho}}{\rho}\right) = -\int \widehat{\pi} \log\left(1 + \frac{\widehat{\rho} - \rho}{\rho}\right) \\
&\leq\; \int \widehat{\pi} \frac{\widehat{\rho} - \rho}{\rho} = \int \frac{\widehat{\pi}}{\pi}\frac{\pi}{\rho}(\widehat{\rho} - \rho) \\
&\leq\; \max\left(\frac{\pi}{\rho}\right) \int |\widehat{\rho} - \rho| \\
&=\; 2M\, d_{\text{TV}}(\widehat{\rho}, \rho).
\end{aligned} \tag{62}$$

Using (61), (62) we get:

$$\begin{aligned}
|D_{\text{KL}}(\widehat{\pi} \,\|\, \widehat{\rho}) - D_{\text{KL}}(\pi \,\|\, \rho)| \;&\leq\; |D_{\text{KL}}(\widehat{\pi} \,\|\, \widehat{\rho}) - D_{\text{KL}}(\widehat{\pi} \,\|\, \rho)| + |D_{\text{KL}}(\widehat{\pi} \,\|\, \rho) - D_{\text{KL}}(\pi \,\|\, \rho)| \\
&\leq\; D_{\text{KL}}(\widehat{\pi} \,\|\, \pi) + 2M\, d_{\text{TV}}(\widehat{\rho}, \rho) + 2 \log M\, d_{\text{TV}}(\widehat{\pi}, \pi) \\
&\leq\; D_{\text{KL}}(\widehat{\pi} \,\|\, \pi) + 2M \sqrt{\frac{1}{2} D_{\text{KL}}(\widehat{\rho} \,\|\, \rho)} + 2 \log M \sqrt{\frac{1}{2} D_{\text{KL}}(\widehat{\pi} \,\|\, \pi)}
\end{aligned} \tag{63}$$

where in the last line we utilized Pinsker's inequality to bound the TV distance with the square root of the KL divergence. Now we apply (63) to diffusion models:

$$\begin{aligned}
|D_{\text{KL}}(\mathfrak{p}_0(\cdot; s_p) \,\|\, \mathfrak{q}_0(\cdot; s_q)) - D_{\text{KL}}(p_0(\cdot; s_p) \,\|\, q_0(\cdot; s_q))| \;&\leq D_{\text{KL}}(p_0(\cdot; s_p) \,\|\, \mathfrak{q}_0(\cdot; s_p)) \\
&\quad + 2M \sqrt{\frac{1}{2} D_{\text{KL}}(p_0(\cdot; s_q) \,\|\, \mathfrak{q}_0(\cdot; s_q))} \\
&\quad + 2 \log M \sqrt{\frac{1}{2} D_{\text{KL}}(p_0(\cdot; s_p) \,\|\, \mathfrak{q}_0(\cdot; s_p))}
\end{aligned} \tag{64}$$

Furthermore from Lemma 6 we know:

$$D_{\text{KL}}(p_0(\cdot; s_p) \,\|\, \mathfrak{q}_0(\cdot; s_p)) \;\leq\; c/T^2, \quad D_{\text{KL}}(p_0(\cdot; s_q) \,\|\, \mathfrak{q}_0(\cdot; s_q)) \;\leq\; c/T^2 \tag{65}$$

Putting together (64) and (65) we get:

$$|D_{\text{KL}}(\mathfrak{p}_0(\cdot; s_p) \,\|\, \mathfrak{q}_0(\cdot; s_q)) - D_{\text{KL}}(p_0(\cdot; s_p) \,\|\, q_0(\cdot; s_q))| \;\leq\; \epsilon_1(T) \tag{66}$$

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

□

D.5 Proof of Lemma 8

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

$$\begin{aligned}
&\left| \int_{\tau=0}^{1} \widetilde{\omega}_\tau\, \mathbb{E}_{x \sim \mathfrak{p}_\tau(\cdot\,; s_p)} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] - \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T}\, \mathbb{E}_{x \sim \mathfrak{p}_{t/T}(\cdot\,; s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \right| \\
=\;& \left| \int_{\tau=0}^{1} f(\tau) d\tau - \sum_{t=0}^{T} f(t/T) \cdot \frac{1}{T} \right| \\
=\;& \frac{1}{T} \sup_{\tau \in [0,1]} \left| \frac{df}{d\tau} \right|
\end{aligned}$$

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

$$\begin{aligned}
\frac{df}{d\tau} &= \frac{d}{d\tau}\left(\int \mathfrak{p}_\tau(x\,; s_p) \, \|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \, dx \right) \\
&= \int \frac{d}{d\tau}(\mathfrak{p}_\tau(x\,; s_p)) \, \|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \, dx \\
&\quad + \int \mathfrak{p}_\tau(x\,; s_p) \frac{d}{d\tau}(\|s_p(x,\tau) - s_q(x,\tau)\|_2^2) dx.
\end{aligned} \tag{67}$$

We bound each term in (67) separately. Then the first term in (67) is bounded because $\frac{d}{d\tau}(\mathfrak{p}_\tau(x\,; s_p))$ is finite as characterized in Lemma 9. The second term in (67) we expand further:

$$\begin{aligned}
&\int \mathfrak{p}_\tau(x\,; s_p) \frac{d}{d\tau}(\|s_p(x,\tau) - s_q(x,\tau)\|_2^2) dx \\
&= \int 2\mathfrak{p}_\tau(x\,; s_p) \left\langle s_p(x,\tau) - s_q(x,\tau), \, \frac{ds_p(x,\tau)}{d\tau} - \frac{ds_q(x,\tau)}{d\tau} \right\rangle dx \\
&\leq 2 \sup_{x,\tau} \|s_p(x,\tau) - s_q(x,\tau)\|_2 \left\| \frac{d}{d\tau}(s_p(x,\tau) - s_q(x,\tau)) \right\|_2 \\
&\leq 2B_1 B_2.
\end{aligned}$$

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

$$\begin{aligned}
&\left| \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x \sim \mathfrak{p}_{t/T}(\cdot\,; s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] - \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x \sim p_t(\cdot\,; s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \right| \\
&\leq \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} d_{TV}(p_t(\cdot\,; s_p), \mathfrak{p}_{t/T}(\cdot\,; s_p)) \cdot \sup_x \|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \\
&\leq \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/\tau} \sqrt{\frac{c}{T^2}} \cdot B_1^2 \\
&\leq T \cdot \frac{1}{T} \cdot \sqrt{\frac{c}{T^2}} \cdot B_1^2 \\
&= O(\frac{1}{T})
\end{aligned}$$

where we used Lemma 6 to get the last line which concludes the proof. □

E Composition with Forward KL Divergences

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

$$\begin{array}{ll} \underset{u,\, p}{\text{minimize}} & u \\ \text{subject to} & D_{\text{KL}}(q^i \,\|\, p) \;\leq\; u \quad \text{for } i = 1, \ldots, m. \end{array} \tag{68}$$

In the case of diffusion models, the KL divergence in (68) becomes the forward path-wise KL between the processes:

$$\begin{array}{ll} \underset{u,\, p}{\text{minimize}} & u \\ \text{subject to} & D_{\text{KL}}(q^i_{0:T}(\cdot) \,\|\, p_{0:T}(\cdot; s)) \;\leq\; u \quad \text{for } i = 1, \ldots, m. \end{array} \tag{69}$$

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 $q^i_0(x_0)$.

In summary, the two key differences of Problem (69) 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 pretrained models or datasets, evaluating $D_{\text{KL}}(q^i_{0:T}(\cdot) \,\|\, p_{0:T}(\cdot; s))$ is intractable since it requires knowing $q^i_{0:T}(\cdot)$ which in turn requires knowing $q^i_0(\cdot)$ exactly. To get around this issue we formulate a closely related problem to (69) by replacing the KL with the Evidence Lower Bound (Elbo):

$$\begin{array}{ll} \underset{u,\, p}{\text{minimize}} & u \\ \text{subject to} & \text{Elbo}(q^i_{0:T}; p_{0:T}) \;\leq\; u \quad \text{for } i = 1, \ldots, m \end{array} \tag{70}$$

where the Elbo is defined as

$$\text{Elbo}(q_{0:T}; p_{0:T}) \;:=\; \mathbb{E}_{x_0 \sim q_0} \mathbb{E}_{q(x_{1:T}|x_0)} \log \frac{p_{0:T}(x_{0:T})}{q(x_{1:T}|x_0)}. \tag{71}$$

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 $D_{\text{KL}}(q^i_{0:T}(\cdot) \,\|\, p_{0:T}(\cdot; s))$ since they only differ by a constant that does not depend on p. (see [22] for more details on this)

For a given λ, we define a weighted mixture of distributions as

$$q^{(\lambda)}_{\text{mix}}(\cdot) \;=\; \sum_{i=1}^{m} \frac{\lambda_i}{\lambda^\top 1} q^i(\cdot) \tag{72}$$

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

$$H(q) \;:=\; -\mathbb{E}_{x \sim q}[\log q(x)]. \tag{73}$$

Theorem 5. *Problem (70) is equivalent to the following unconstrained problem:*

$$\underset{p}{\text{minimize}} \;\; D_{\text{KL}}(q^{(\lambda^\star)}_{\text{mix}} \,\|\, p) \tag{74a}$$

where $\lambda^\star$ is the optimal dual variable given by $\lambda^\star = \operatorname{argmax}_{\lambda \geq 0} D(\lambda)$. The dual function has the explicit form, $D(\lambda) = H(q^{(\lambda)}_{\text{mix}})$. Furthermore, the optimal solution of (7) is given by

$$p^\star = q^{(\lambda^\star)}_{\text{mix}}. \tag{74b}$$

Unlike the reverse KL case, here we can characterize the optimal dual multipliers, and the optimal solution further; Note that the optimal dual multiplier $\lambda^\star = \operatorname{argmax}_{\lambda \geq 0} D(\lambda) = \operatorname{argmax}_{\lambda \geq 0} H(q_{\text{mix}}(\cdot; \lambda^\star))$ 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 $\lambda^\star$ of Problem (70) can be characterized explicitly as*

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

F Algorithm Details

F.1 Alignment

Recall from Section 3.2 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:

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

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

Input: total diffusion steps T, diffusion parameter α_t, total dual iterations H, number of primal steps per dual update N, dual step size η_d, primal step size η_p, initial model parameters $\theta(0)$.
Initialize: $\lambda(1) = 1/m$.
for $h = 1, \cdots, H$ **do**
  Initialize $\theta_1 = \theta(h-1)$
  for $n = 1, \cdots, N$ **do**
    Take a primal gradient descent step:

$$\theta_{n+1} \;=\; \theta_n - \eta_p \cdot \nabla_\theta \bar{L}_{\text{ALI}}(\theta, \lambda^{(n)}). \tag{75}$$

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

$$\lambda_i(h+1) \;=\; \big[\lambda_i(h) \,+\, \eta_d(\mathbb{E}_{x_0\,\sim\,p_0(\cdot;s_\theta)}\big[\,r_i(x_0)\,\big] - b_i)\big]_+. \tag{76}$$

end for

We now discuss the practicality of the primal gradient descent step (75) regarding the Lagrangian function,

$$\bar{L}_{\text{ALI}}(\theta, \lambda) \;=\; D_{\text{KL}}\big(\,p_{0:T}(\cdot; s_\theta)\,\|\,q_{0:T}(\cdot; s_q)\,\big) - \sum_i \lambda_i(\mathbb{E}_{x_0\,\sim\,p_0(\cdot;s_\theta)}\big[\,r_i(x_0)\,\big] - b_i). \tag{77}$$

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 θ. We can use the following result (Lemma 4.1 from [17]) to take the gradient inside the expectation.

Lemma 11. *If $p_\theta(x_{0:T})r(x_0)$ and $\nabla_\theta p_\theta(x_{0:T})r(x_0)$ are continuous functions of θ, then we can write the gradient of the reward function as*

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

For the gradient of the KL divergence, we have

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

For simplicity, we omit the second term in practice, as it has negligible effect on performance. See [17, Appendix A.3] for the derivation.

F.2 Composition

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

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

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 α_t, total dual iterations H, number of primal steps per dual update N, dual step size η_d, primal step size η_p, initial model parameters $\theta(0)$.
2: **Initialize**: $\lambda(1) = 1/m$.
3: **for** $h = 1, \cdots, H$ **do**
4: Initialize $\theta_1 = \theta(h-1)$
5: **for** $n = 1, \cdots, N$ **do**
6: Take a primal gradient descent step:

$$\theta_{n+1} \;=\; \theta_n - \eta_p \cdot \nabla_\theta \widehat{L}_{\mathrm{AND}}(\theta, \lambda^{(n)}). \tag{78}$$

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 $i = 1, \cdots, m$:

$$\widetilde{\lambda}_i(h+1) \;=\; \lambda_i(h) \;+\; \eta_d D_{\mathrm{KL}}(p_0(\cdot\,; s_{\theta(h)}) \,\|\, q_0^i(\cdot\,; s^i)). \tag{79}$$

10: $\lambda(h+1) = \mathrm{proj}\left(\widetilde{\lambda}(h+1)\right)$, where $\mathrm{proj}(y)$ projects its input onto the simplex $\lambda^T 1 = 1$.
11: **end for**

The projection of the dual multiplier vector (line 10) ensures that $\lambda^\top \mathbf{1} \;=\; 1$, as required when maximizing the dual function (see the proof of Theorem 3).

Note that Algorithm 2 implicitly requires samples from the weighted product distribution $q^{(\lambda)}_{\text{AND}}(\cdot)$ in order to minimize the Lagrangian $\widehat{L}_{\text{AND}}(\theta, \lambda)$. We obtain these samples using the Annealed MCMC sampling algorithm proposed in [14].

Skipping the primal. As discussed in Section 5, both Annealed MCMC sampling and the minimization of the Lagrangian $\widehat{L}_{\text{AND}}(\theta, \lambda)$ at each primal step—to match the true score $\nabla \log q^{(\lambda)}_{\text{AND}}$—are challenging and computationally expensive. Therefore, for all settings except the low-dimensional case described in Appendix G.2, we employ Algorithm 3, which skips the primal step entirely.

In Algorithm 3 we bypass the primal steps by using the surrogate product score, rather than the true score, to compute the point-wise KL used in the dual updates. The distinction between the true and surrogate scores is discussed in detail in [14].

$$\text{true score:} \quad \nabla \log q^{(\lambda)}_{\text{AND},\,t}(x_t) = \nabla \log \left(\int \sum_i (q_0(x_0))^{\lambda_i} q(x_t|x_0) dx_0 \right) \tag{80}$$

$$\text{surrogate score:} \quad \nabla \log \widehat{q}^{(\lambda)}_{\text{AND},\,t}(x_t) = \sum_i \lambda_i \nabla \log \left(\int q_0(x_0) q(x_t|x_0) dx_0 \right) \tag{81}$$

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

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

$$\widetilde{\lambda}_i(h+1) \;=\; \lambda_i(h) \;+\; \eta_d D_{\text{KL}}(\widehat{q}^{(\lambda(h))}_{\text{AND},0}(\cdot) \,\|\, p_0(\cdot\,; s^i)). \tag{82}$$

5: $\lambda(h+1) \;=\; \text{proj}\left(\widetilde{\lambda}(h+1)\right)$, where $\text{proj}(y)$ projects its input onto the simplex $\lambda^T 1 = 1$.
6: **end for**

For a given λ, the surrogate score can be easily computed:

$$\begin{aligned} \nabla \log \widehat{q}^{(\lambda)}_{\text{AND},\,t}(x_t) \;&=\; \sum_i \lambda_i \nabla \log \left(\int q_0(x_0) q(x_t|x_0) dx_0 \right) \\ &=\; \sum_i \lambda_i \nabla \log p_t(x_t; s^i) \end{aligned}$$

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

G Additional Experiments and Experimental Details

G.1 Related work

Here we review related work and explain why these approaches are not directly applicable as baselines for our experiments.

In [40] they propose a superposition method to sample from the mixture of diffusion models with arbitrary weights. However, they only use equal weight mixtures and don't discuss different weights. They also devise a method to sample points that have equal likelihood under different models which is fundamentally different to the product composition that we discuss in this work.

Existing works including [11, 27, 34, 58] discuss constrained sampling from diffusion models, but the nature of their constraints is completely different from our work as it mainly involves sampling from a constrained set and they propose to do this through projection onto a feasible set at each diffusion time step. It is not clear how to apply these methods to reward constraints or how to use them to preserve distance to a model.

Other works like [18, 8] enforce very specific constraints by adding additional losses with fixed weights to the objective which implicitly enforces the constraint. These methods are very specific to the constraints they are designed for and do not generalize to arbitrary reward functions and don't give us a way to constrain closeness to a model.

G.2 Low-dimensional synthetic experiments

To visually illustrate the difference between the constrained and unconstrained approaches, we conduct experiments where the generated samples lie in $\mathbb{R}^2$. For the score predictor we used the same ResNet architecture as used in [14].

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

Mixture composition (OR). For this experiment we used the same Algorithm as the one used in [22] for mixture of distributions. The only modification is doing an additional dual multiplier projection step similar to the last step of the product composition Algorithm 2. See Figure 2 for visualization of the resulting distributions.

G.3 Reward product composition (Section 5.2 (I)

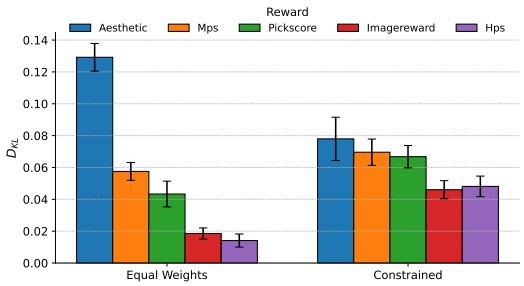

Figure 6: KL divergence for the product composition of 5 adapters pretrained with different rewards. Error bars denote the standard deviation computed across 8 text prompts each with four samples.

Implementation details and hyperparameters. We finetuned the model using the Alignprop [35] official implementation [5] for each individual reward using the hyperparameters reported in Table 2. We then composed the trained adapters running dual ascent using the surrogate score as described in section F.2. We use the average of scores (denoted as "Equal weights") as a baseline. Hyperparameters are described in Table 3. The reward values reported in Figure 4 were normalised so that 0%

[5] https://github.com/mihirp1998/AlignProp

corresponds to the reward obtained by the pretrained model, and 100% the reward obtained by the model finetuned solely on the corresponding reward.

Additional results. As shown in Figure 6, equal weighting leads to disparate KL divergences across adapters – in particular high KL with respect to the adapter trained with the "aesthetic" reward – while our constrained approach effectively reduces the worst case KL, equalizing divergences across adapters. Figure 13 shows images sampled from these two compositions exhibit different characteristics, with our constrained approach producing smoother backgrounds, shallower depth of field and more painting-like images.

Hyperparameter	**Value**
Batch size	64
Samples per epoch	128
Epochs	10
Sampling steps	50
Backpropagation sampling	Gaussian
KL penalty	0.1
Learning rate	1×10^{-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.

G.4 Concept composition (Section 5.2 (II))

We present additional results for concept composition using three different concepts (as opposed to just 2 in the main paper and in [40]) As seen in table 4, our approach retains a clear advantage in both CLIP and BLIP scores. See Table 5 for examples of images generated using each method. Images with the constrained method typically do a better job of representing all concepts.

	Min. CLIP (↑)	**Min. BLIP (↑)**
Combined Prompting	21.52	0.206
Equal Weights	22.18	0.203
Constrained (Ours)	**22.45**	**0.221**

Table 4: 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 5: 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 cinammon roll".

G.5 Concept composition for text-to-audio diffusion models

We note that our proposed framework and theoretical analysis do not depend on any specific modality or task types. From our theoretical guarantees, we would expect experiments in other modalities to provide results similar to those presented for images. To validate this, we conduct concept composition experiments with a text-to-audio diffusion model as an example of another modality. We treat a text-to-audio diffusion model (in this case AudioLDM [30]) conditioned on different inputs, each representing a concept, as the models to be composed. We apply our constrained learning to find the optimal weights to compose these two models, and use the CLAP score [15] to measure the similarity between the generated audio samples and the text prompts representing each model.

	Min. CLAP Score(↑)
Combined Prompting	0.816
Equal Weights	1.57
Constrained (Ours)	**1.92**

Table 6: Minimum CLAP scores across prompts for each method

Similar to concept composition for images, we observe in Table 6 that using our constrained approach, the minimum CLAP score across prompts increases compared to the two baselines. The constraints ensure closeness to each model, which in turn results in a more equal representation of the concepts.

G.6 Alignment experiments

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

In order to tackle this, we propose normalizing rewards using the pretrained model statistics as a simple yet effective heuristic. This normalization facilitates the setting of constraint levels, enables direct comparisons across rewards and enhances interpretability. In all of our experiments, we apply this normalization before enforcing constraints. Explicitly, we set

$$\widetilde{r} \;=\; \frac{r - \widehat{\mu}_{\text{pre}}}{\widehat{\sigma}_{\text{pre}}} \tag{83}$$

where r denotes the original reward and $\widehat{\mu}_{pre}, \widehat{\sigma}_{pre}$ the sample mean and standard deviation of the reward for the pretrained model. We find that, with this simple transformation, setting equal 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 pretrained model.

The effects of varying the constraint thresholds.

What we observed by varying the reward constraint thresholds in our experiments was that for thresholds up to 1.0 (i.e. $\widehat{\mu}_{\text{pre}} + 1.0 \times \widehat{\sigma}_{\text{pre}}$ for each reward) the model was typically able to satisfy the constraints with minimal violation. Another trend that we observed was that increasing thresholds usually leads to constraints that are harder to satisfy leading to higher Lagrange multipliers and resulting in higher KL to the pretrained model. See Tables 8, 9 below.

An advantage of our constrained approach is that Lagrange multipliers give information about the sensitivity of the objective with respect to relaxing the constraints i.e. if the multiplier for a certain reward ends up being much higher than the rest it means that constraint is particularly harder to satisfy. Consequently, even slightly relaxing the threshold for the corresponding reward can lead to much smaller KL objective.

Constraint	Threshold	Slack	Dual Variable	D_{KL}
contrast	0.250	-0.245	0.282	0.177
contrast	0.500	-0.985	0.000	0.296
contrast	1.000	-0.381	0.000	0.332
saturation	0.250	-0.126	0.081	0.177
saturation	0.500	0.060	0.006	0.296
saturation	1.000	0.052	1.195	0.332

Table 8: MPS reward alignment with saturation and contrast constraints, for varying thresholds.

Constraint	Threshold	Slack	Dual Variable	D_{KL}
contrast	0.250	-0.684	0.000	0.136
contrast	0.500	-1.011	0.000	0.109
contrast	1.000	0.661	0.192	0.293
saturation	0.250	-0.025	0.014	0.136
saturation	0.500	-0.060	0.000	0.109
saturation	1.000	0.062	1.020	0.293

Table 9: Pickscore reward alignment with saturation and contrast constraints, for varying thresholds.

I. MPS + local contrast, saturation.

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

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

where H, W denote image dimensions.

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

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

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

Implementation details and hyperparameters. We implemented our primal-dual alignment approach (Algorithm 1) in the Alignprop framework. Following their experimental setting, we use different animal prompts for training and evaluation. Hyperparameters are detailed in Table 10.

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
Constraint level	MPS: 0.5 Saturation: 0.5 Local contrast: 0.25
Equal weights	0.2

Table 10: Hyperparameters for reward alignment with contrast and saturation constraints. Constraint levels correspond to normalized rewards.

Additional results. We include images sampled from the constrained model in Figure 14 for hps and aesthetic reward functions. Samples from a model trained with an equally weighted model are included for comparison. Constraints prevent overfitting to the saturation and smoothness penalties.

II. Multiple aesthetic constraints

Implementation details and hyperparameters. We modified the Alignprop framework to accomodate Algorithm 1. Following their setup, we use text conditioning on prompts of simple animals, using separate sets for training and evaluation. In this setting, due to the high variability of rewards throughout training, utilized an exponential moving average to reduce the variance in slack estimates (and hence dual subgradients) [41]. Hyperparameters are detailed in Table 11.

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
Constraint level	MPS: 0.5 HPS: 0.5 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 11: Hyperparameters for reward alignment with multiple rewards. Constraint levels correspond to normalised rewards.

Additional results. We include two images per method and prompt in Figure 15. These are sampled from the same latents for both models.

G.7 Combining constrained alignment and composition

As mentioned in Section 2, The constrained alignment and composition problem formulations can be combined. An example of this is composing reward-specialized models while enforcing a minimum aggregate reward level. To demonstrate the viability of this approach, we conducted a simple experiment in which we finetune a pretrained model using two KL constraints: one for pretrained stable diffusion and one for another model finetuned on the aesthetics reward, respectively, along with a constraint on the saturation reward. The finetuned model achieves less than 10% reward constraint violation and similar KL divergences with respect to both pretrained models, as seen in Table 12. We leave in-depth exploration of the combined Alignment and Composition problem to future work.

Constraint	Dual Variable	Initial Value	Final Value	Slack
Saturation	1.080	0.100	0.533	0.033
KL (pretrained)	0.493	0.000	0.101	0.004
KL (aesthetics)	0.507	0.260	0.097	0.000

Table 12: Results for finetuning a model with both expected reward and KL divergence constraints.

G.8 Computational details

All experiments were run on a single Nvidia A6000 GPU.

For alignment, there is little additional time overhead compared to baselines like AlignProp. For example, for the experiment in Figure 3, runtime is 33 minutes for both constrained and unconstrained methods, and for the experiments in Figure 4, constrained runtime is 64 minutes, unconstrained is 60 minutes. Existing approaches already estimate the KL and sample batches to evaluate and back-propagate through the reward. The only additional computation for our method is the dual updates which is negligible in terms of added time.

For composition, there is no meaningful comparison to the equal weights baseline since the weights are not learned in the equal weights baseline. For constrained composition, it takes around 5–10 dual updates for dual variables to converge which for composing the 5 finetuned stable diffusion models takes 9 minutes total and for concept composition it takes 2 minutes.

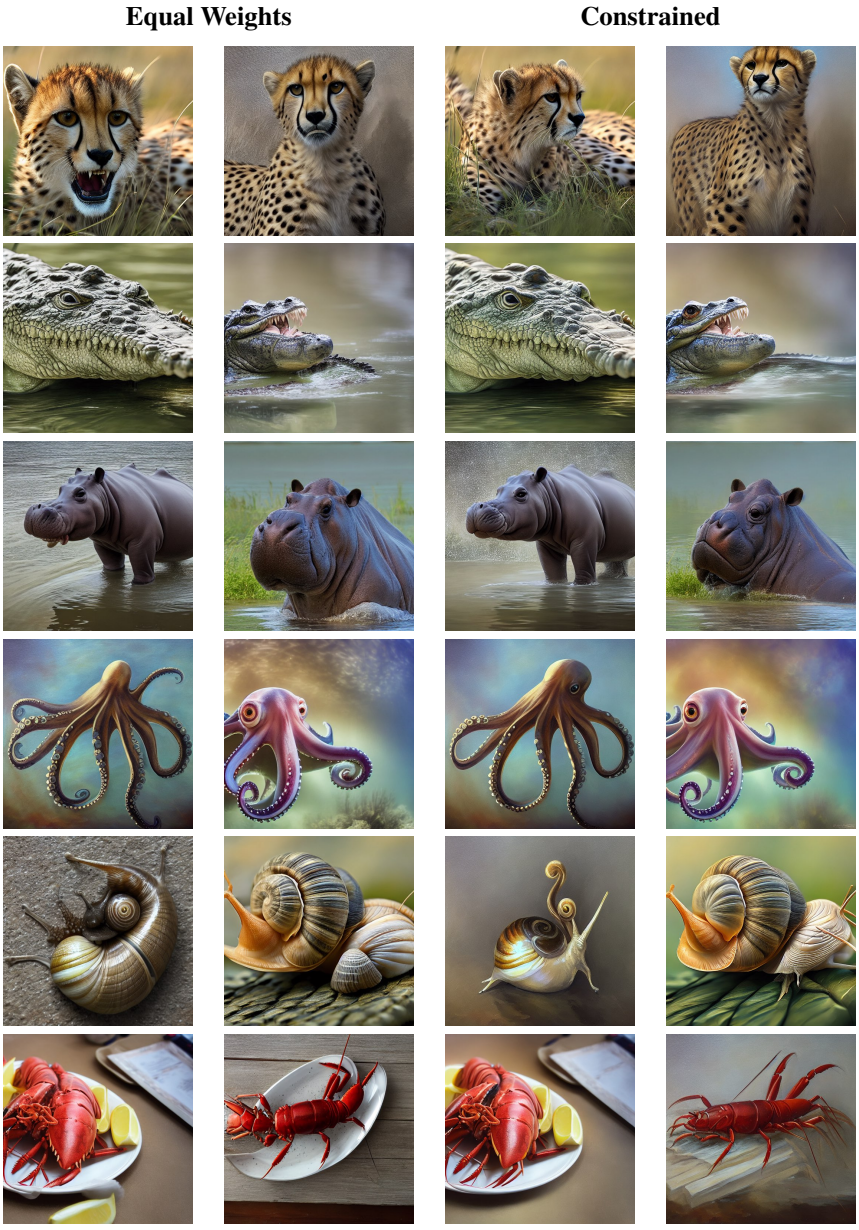

Table 13: Images sampled from the same latents for the product of adapters using the equal weights and when using the proposed KL-constrained reweighting scheme using 5 dual steps.

Table 14: Images sampled from models finetuned to maximize MPS [60], along with sharpness and saturation penalizations. We compare optimizing an equally weighted objective against our constrained approach.

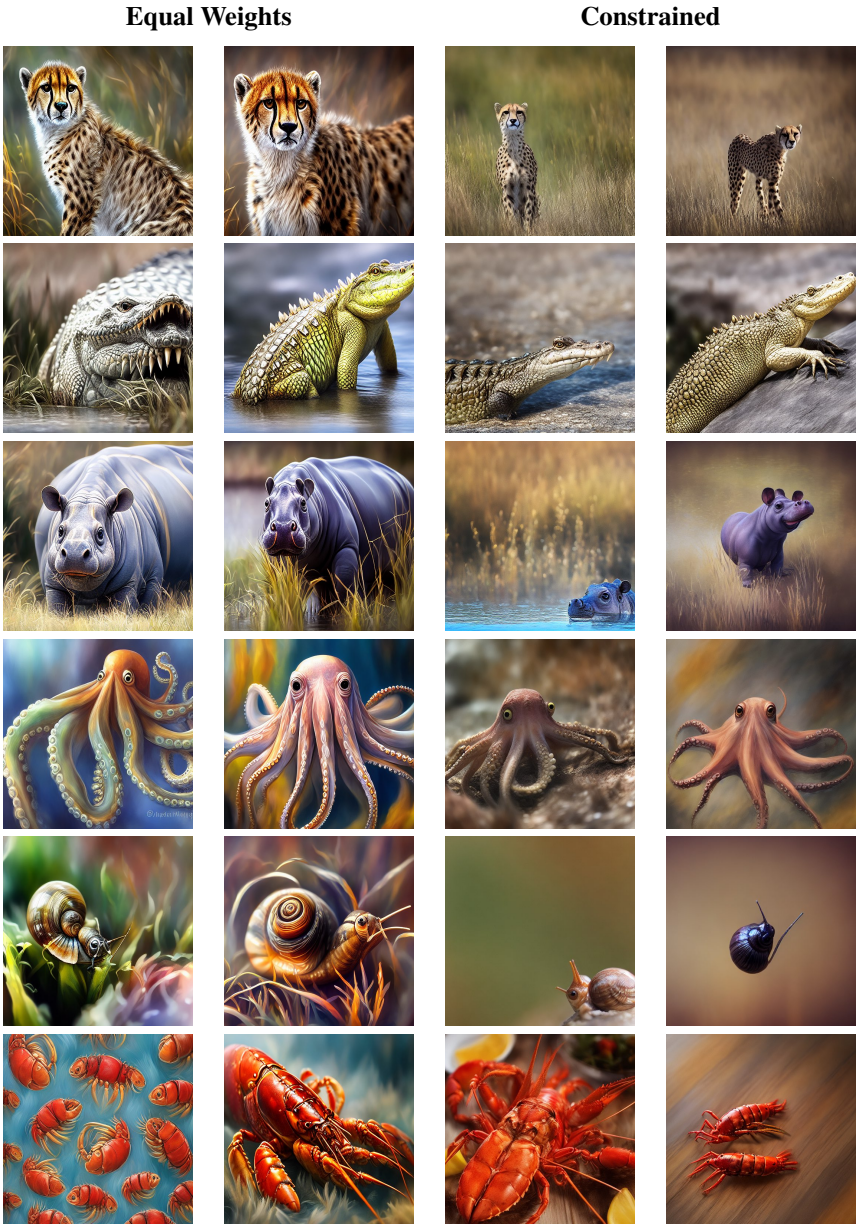

Table 15: Samples from models finetuned using multiple rewards with equal weights and with our constrained alignment method.