Composition and Alignment of Diffusion Models using Constrained Learning

Shervin Khalafi [1] **Ignacio Hounie** [1] **Dongsheng Ding** [1] **Alejandro Ribeiro** [1]

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 fine-tuning a diffusion model to align it with a reward; and (ii) Composition, which combines several pre-trained 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 pre-trained 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.

1. Introduction

Diffusion models have emerged as the tool of choice for generative models in a variety of settings (Saharia et al., 2022; Blattmann et al., 2023; Wang et al., 2025; Chi et al., 2023), image generation being most prominent among them (Rombach et al., 2022). 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 (Ulhaq & Akhtar, 2022; Yan et al., 2024). For this reason, *alignment* and *composition* of what, in this context, become *pretrained* models, has become popular (Liu et al., 2024; 2022).

Regardless of whether the goal is alignment or composition, we want to balance what are most likely conflicting requirements. In alignment tasks, we want to stay close to the pretrained model while deviating sufficiently so as to effect some rewards of interest (Fan et al., 2023; Domingo-Enrich et al., 2025). In composition tasks we are given several pretrained models and our goal is to sample from their union or intersection (Du et al., 2024; Biggs et al., 2024). The standard approach to balance these requirements involves the use of weighted averages. This can be a linear combination of score functions in composition problems (Du et al., 2024; Biggs et al., 2024) or may involve a loss given by a linear combination of a Kullback-Leibler (KL) divergence and a reward (Fan et al., 2023) in the case of alignment.

In this work we propose a unified view of alignment and composition via the lens of constrained learning (Chamon & Ribeiro, 2021; Chamon et al., 2022). 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 next outline our key contributions in alignment and composition.

Alignment. For alignment, we formulate a reverse KL divergence-constrained optimization problem that minimizes the reverse KL divergence to a pre-trained model, subject to expected reward constraints. The threshold for

[1] University of Pennsylvania, PA, USA. Correspondence to: Shervin Khalafi <shervink@seas.upenn.edu>, Ignacio Hounie <ihounie@seas.upenn.edu>, Dongsheng Ding <dongshed@seas.upenn.edu>.

ICML 2025 Workshop on Models of Human Feedback for AI Alignment. Copyright 2025 by the author(s).

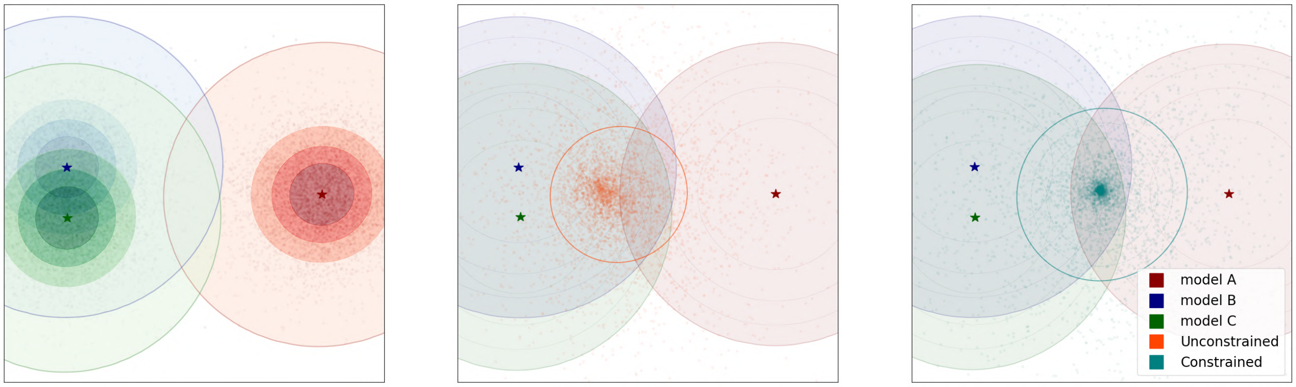

Figure 1. Product composition (AND): (left) Three Gaussian distributions being composed. (middle) Composition using equal weights and (right) with constraints. The constrained model samples from the intersection of the three models.

each reward constraint can be user-specified or automatically selected using a heuristic approach (see Section 5). In Section 3 we show that the solution of this alignment problem is the pretrained model distribution scaled by an exponential function of a weighted sum of reward functions. To solve this problem with diffusion models, we establish strong duality, and employ a Lagrangian dual-based approach to develop a primal-dual training algorithm.

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

Composition. For composition, we propose using KL divergence constraints to ensure the closeness to each individual model. It is important to distinguish composition with *reverse* KL and *forward* KL constraints. As previously shown in (Khalafi et al., 2024), using forward KL constraints results in the composed model sampling from a weighted mixture of the individual distributions. In the main paper we focus on composition with reverse KL constraints, while we discuss forward KL constraints in Appendix D. In Section 4, we characterize the solution of the constrained optimization problem with *reverse* KL divergence constraints as a tilted product of the individual distributions. To solve this problem with diffusion models, we similarly establish strong duality and develop a primal-dual training algorithm.

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

2. Composition and Alignment of Diffusion Models in Distribution Space

Reward alignment: Given a pretrained model q and a set of m rewards $\{r_i(x)\}_{i=1}^m$ that can be evaluated on a sample x, we consider the *reverse* KL divergence $D_{\text{KL}}(p \,\|\, q) := \int p(x) \log(p(x)/q(x))dx$ 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,

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

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 $D_{\text{KL}}(p \,\|\, q)$ 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 $P^\star_{\text{ALI}} := D_{\text{KL}}(p^\star \,\|\, q)$.

Product composition (AND): Given a set of m pretrained

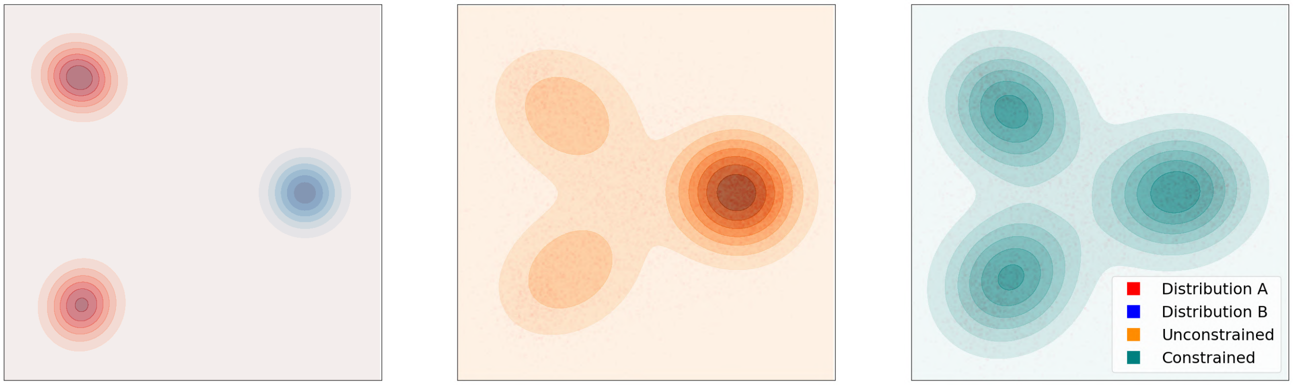

Figure 2. Mixture composition (OR): (left) Two of Gaussian mixtures being composed. One has two modes and the other has only a single mode. (middle) Composition using equal weights and (right) with constraints.

models $\{q_i\}_{i=1}^m$, we formulate a constrained composition problem that solves a reverse KL-constrained optimization problem,

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

In (UR-C), the decision variable u is an upper bound on m KL divergences between a distribution p and m pretrained models $\{q_i\}_{i=1}^m$. Partial minimization over u allows us to search for a distribution p that minimizes a common upper bound. Thus, the optimal solution $p^\star$ minimizes the maximum KL divergence among m terms, each computed between p and a pretrained model. The epigraph formulation (UR-C) is useful in practice, as the constraint threshold u is updated dynamically during training. Let the primal value be $P^\star_{\text{AND}} := u^\star$. See Figure 1 for an illustration. The unconstrained model composed with equal weights is biased towards the distributions that are closer to each other.

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

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

We note that mixture composition has been studied in a related but slightly different constrained setting in (Khalafi et al., 2024). It can be shown that the solution of the constrained problem (UF-C) learns to sample from each distribution proportional to its entropy. In Figure 2, we see that the constrained model learns to sample more often from the distribution with two modes that has a higher entropy in contrast to the equally weighted composition that samples equally from both distributions leading to unbalanced sampling from the modes. Since the algorithm design and analysis for (UF-C) closely resemble those in (Khalafi et al., 2024), mixture composition is not the focus of this work. For completeness, we discuss and compare it with product composition in Appendix D.

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

2.1. KL divergence for diffusion models

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

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

where $\epsilon_t \sim \mathcal{N}(0, I)$ and $\{\alpha_t\}_{t=1}^T$ is a decreasing sequence of coefficients called the noise schedule. We denote the marginal density of $\bar{X}_t$ at time t as $\bar{p}_t(\cdot)$. Given a d-dimensional score predictor function $s(x,t)$: $\mathbb{R}^d \times \{1, \cdots, T\} \to \mathbb{R}^d$, we define a backward denoising diffusion implicit model (DDIM) process (Song et al., 2022),

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

where $\epsilon_t \sim \mathcal{N}(0, I)$, and $\{\sigma_t^2\}_{t=1}^T$ is the variance schedule determining the level of randomness in the backward process (e.g., $\sigma_t = 0$ reduces to deterministic trajectories), and $\beta_t := \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} \sqrt{(1-\alpha_t)(1-\bar{\alpha}_t)} - \sqrt{(1-\alpha_{t-1}-\sigma_t^2)(1-\bar{\alpha}_t)}$ is determined by the variance schedule σ_t and the noise schedule α_t. Given a function s, 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 (Song et al., 2021), 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 entire backward process by $D_{\text{KL}}(p_{0:T}(\cdot) \,\|\, q_{0:T}(\cdot))$, which is known as the path-wise KL divergence (Fan et al., 2023; Han et al., 2024). This path-wise KL divergence is often used in alignment to quantify the gap 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*

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

See Appendix C.1 for proof. While the path-wise KL divergence is a useful regularizer in alignment, when composing multiple models, the point-wise KL divergence $D_{\text{KL}}(p_0(\cdot) \,\|\, p_0(\cdot))$ is a more natural measure of the closeness between two 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: $p_{0:T}(\cdot)$, $q_{0:T}(\cdot)$. In this work, we use path-wise KL for alignment and point-wise KL for composition. However, it is not obvious how to compute the point-wise KL, as evaluating the marginal densities is intractable. We next establish a similar formula as (3) by restricting 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*

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

where $\widetilde{\omega}_t$ is a time-dependent constant and ϵ_T is a discretization error depending on 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 (42) can be viewed as the difference of the processes over time steps, up to a discretization error.

3. Aligning Pretrained Model with Multiple Reward Constraints

To apply Problem (UR-A) to diffusion models, we first employ Lagrangian duality to derive its solution in the distribution space. Alignment with constraints is related but fundamentally different from the standard approach of minimizing a weighted average of the KL divergence and rewards (Fan et al., 2023). They are related because the Lagrangian for (UR-A) is precisely the weighted average,

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

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

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

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

In the distribution space, Problem (UR-A) is a convex optimization problem since the KL divergence is strongly convex and the reward constraints are linear in $p \in \mathcal{P}$. Thus, we apply strong duality in convex optimization (Boyd & Vandenberghe, 2004) to characterize the solution to Problem (UR-A) in Theorem 1. Moreover, it is ready to formulate the constrained alignment problem (UR-A) as an unconstrained problem by specializing the dual variables to a solution to the dual problem.

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

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

$$\underset{p\,\in\,\mathcal{P}}{\text{minimize}}\;\; D_{\text{KL}}\left(p \,\|\, q_{\text{rw}}^{(\lambda^\star)}\right) \tag{7}$$

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

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

See Appendix C.3 for proof. Theorem 1 characterizes the solution to the constrained alignment problem (UR-A), i.e., $q_{\text{rw}}^{(\lambda^\star)}$. This solution generalizes the reward-tilted distribution (Domingo-Enrich et al., 2025), which is the solution of finetuning a model with an expected reward regularizer. In Problem (UR-A), the optimal dual variable $\lambda^\star$ weights each reward so that all the constraints are satisfied optimally, while staying as close as possible to the pretrained model.

3.1. Reward alignment of diffusion models

We now introduce diffusion models to Problem (UR-A) by representing p and q as two diffusion models $p_{0:T}(\cdot\,; s_p)$ and $q_{0:T}(\cdot\,; s_q)$, respectively. The path-wise KL divergence has been widely used in diffusion model alignment to capture the difference between two diffusion models (see (Fan et al., 2023)). Hence, we can instantiate Problem (UR-A) in the function space below,

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

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

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

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

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

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

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,

$$\lambda^{(n+1)} \;=\; \Big[\lambda^{(n)} \,-\, \eta\,\Big(r(s^{(n+1)}) - b\Big)\Big]_+. \tag{10}$$

4. Constrained Composition of Multiple Pretrained Models

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

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

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

$$D_{\text{AND}}(\lambda) \;:=\; \underset{u\in\mathbb{R},\; p\,\in\,\mathcal{P}}{\max}\;\; L_{\text{AND}}(p, u, \lambda). \tag{12}$$

Let a solution to Problem (UR-A) 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 pair be $\lambda^\star \in \text{argmax}_{\lambda\,\geq\,0}\, D_{\text{AND}}(\lambda)$, and the optimal value of the dual function be $D^\star_{\text{AND}} := D_{\text{AND}}(\lambda^\star)$.

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

$$q_{\text{AND}}^{(\lambda)}(\cdot) \;:=\; \frac{1}{Z_{\text{AND}}(\lambda)} \prod_{i\,=\,1}^{m} \left(q^i(\cdot)\right)^{\frac{\lambda_i}{\mathbf{1}^\top \lambda}} \tag{13}$$

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

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

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

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*

$$\underset{p\,\in\,\mathcal{P}}{\text{minimize}}\;\; D_{\text{KL}}(p \,\|\, q_{\text{AND}}^{(\lambda^\star)}) \tag{14}$$

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 (14) is given by

$$p^\star \;=\; q_{\text{AND}}^{(\lambda^\star)}. \tag{15}$$

We defer the proof of Theorem 3 to Appendix C.5. 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, where the parameters $\{\lambda_i/\mathbf{1}^\top\lambda\}_{i=1}^m$ weight the importance of each distribution. The geometric mean (Biggs et al., 2024) is a special case when all λ_i are equal.

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 (Biggs et al., 2024; Du et al., 2024). 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).*

4.1. Product composition of diffusion models

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

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

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

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

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

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

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

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

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

We note that computing the point-wise KL that shows up in both the Lagrangian and the constraint violations is not trivial. Recall that Lemma 2 gives us a way to compute the point-wise KL $D_{\text{KL}}(p(x_0; s)\,\|\,p(x_0; s^i))$. However, it requires that the functions s and s^i each be a valid score function for some process. It is reasonable to assume this is the case for s^i since it represents a pretrained model where it would have been trained to approximate the true score of a forward diffusion 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. Lemma 3 lets us minimize the Lagrangian in spite of this:

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

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

Furthermore, a Lagrangian minimizer $s^{(\lambda)} :=$ $\operatorname{argmin} L_{\text{AND}}(s, \lambda)$ can be obtained through minimizing the following objective:

$$\sum_{t=0}^{T} \omega_t\, \mathbb{E}_{x_0 \sim q^{(\lambda)}_{\text{AND}}(\cdot)} \mathbb{E}_{x_t \sim q(x_t|x_0)} \left[\|s(x, t) - \nabla \log q(x_t|x_0)\|^2\right] \tag{18}$$

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

See Appendix C for proof. With Lemma 3, as long as we have access to samples from the distribution $q^{(\lambda)}_{\text{AND}}$, we can approximate the expectation in (18) and use gradient-based optimization methods to find the minimizer $s^{(\lambda)}$. To obtain these samples, we use annealed Markov Chain Monte Carlo (MCMC) sampling as proposed by (Du et al., 2024); see Appendix B for sampling details. For the dual update, we can 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 steps will yield 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 (42):*

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

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

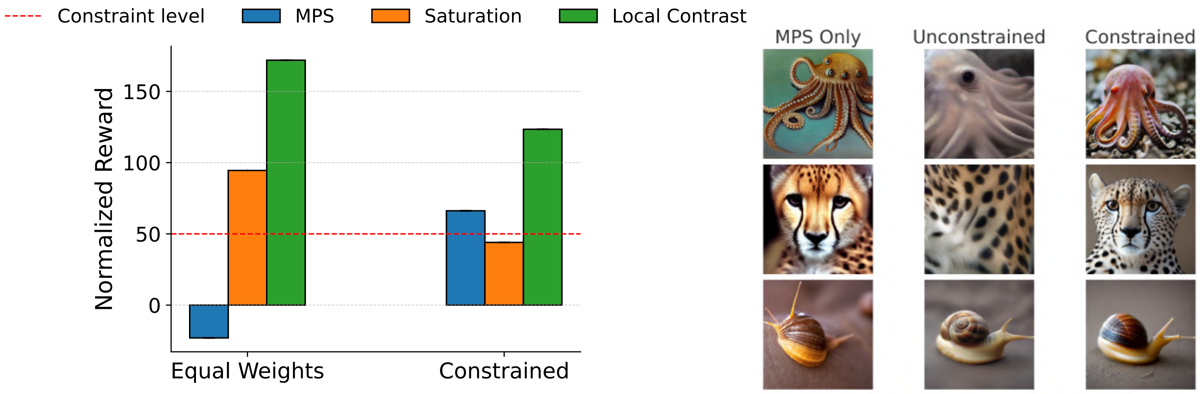

Figure 3. We finetune stable diffusion with one reward emphasizing aesthetic quality (MPS), and Saturation and Local Contrast regularizers. Left: Value of the rewards after finetuning. Right: Images sampled from the aligned models, and the model trained solely with MPS (Zhang et al., 2024a) reward for comparison.

5. Computational experiments

5.1. Finetuning to align with multiple rewards

We extend the AlignProp framework (Prabhudesai et al., 2024a) to accommodate multiple reward constraints and dual updates. We finetune Stable Diffusion v1.5 [1] on widely used differentiable image quality and aesthetic rewards, namely aesthetic (Schuhmann et al., 2022), hps (Wu et al., 2023a), pickscore (Kirstain et al., 2023), imagereward (Xu et al., 2023a) and mps (Zhang et al., 2024a). Since these rewards have widely varying scales, which can make setting the constraint levels challenging, we normalize them by computing the average and standard deviation over a number of batches. In all experiments, models are finetuned using LoRA (Hu et al., 2022). Experimental settings and hyperparameters are detailed in Appendix F.

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

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

5.2. Product composition of diffusion models

In high-dimensional settings like image generation, using MCMC to get samples from the true product distribution and then minimizing the Lagrangian via (18) to find the true product score function is prohibitively costly. To circumvent this, we use a surrogate for the true score both for sampling and computing the KL, as detailed in Appendix F.

I. Composing models fine-tuned on different rewards. We investigate the composition of multiple finetuned versions of the same base model, where each one fit LoRA adapters a different reward function. A key challenge lies in selecting appropriate weights for this combination. Arbitrary choices may 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 approach gives us a way to find the weights that keep us close to each individual model, leading to higher rewards for all models.

II. Concept composition with stable diffusion. Following the setting in (Skreta et al., 2025), we compose two text-to-image diffusion models conditioned on different inputs. We

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

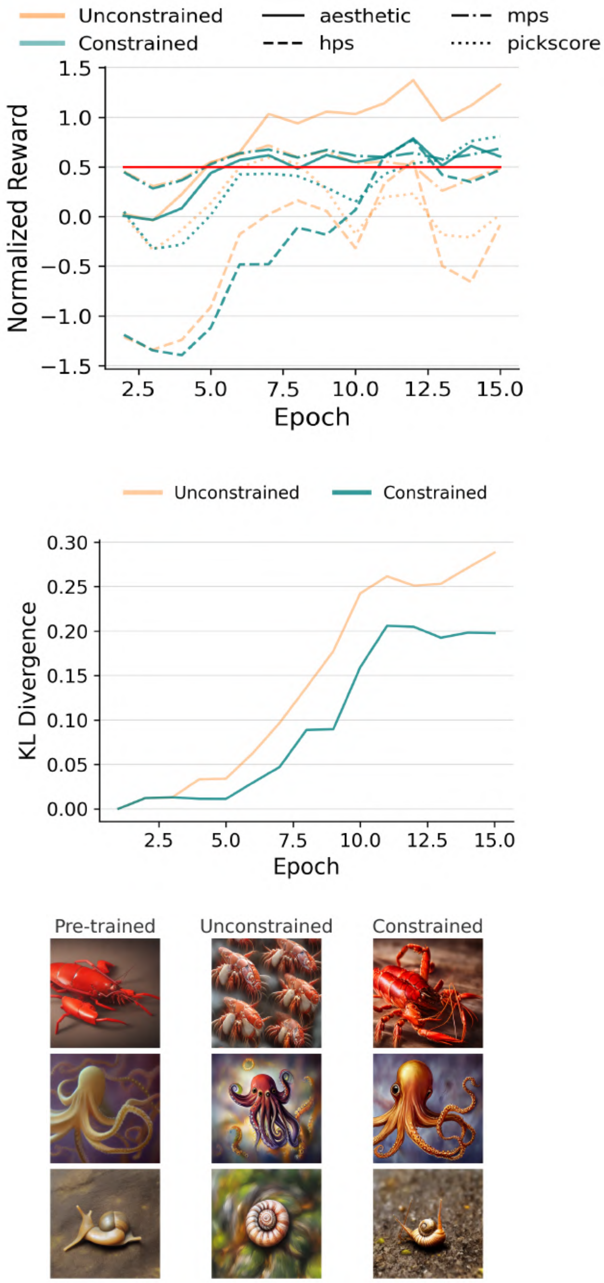

Figure 4. Finetuning with multiple image quality/aesthetic rewards. Top: Reward trajectories in training. Middle: KL to pre-trained model constrained. Bottom: Images sampled from the aligned models and the pre-trained model for reference.

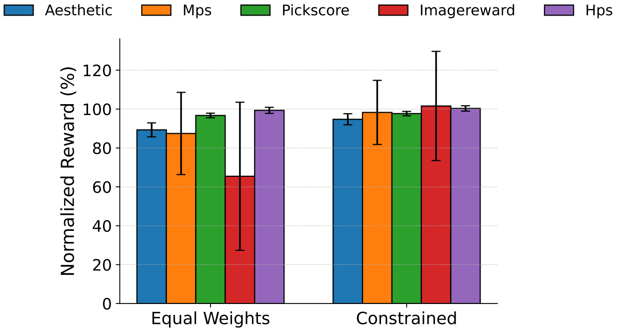

Figure 5. Composition of adapters finetuned for different rewards, for an equal weighted and product mixture. 100 represents the reward attained by the model trained with each individual reward. Higher is better.

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

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

use constrained learning (SR-C) to find the optimal weights to compose these two models. We compare to the baseline of using equal weights for the composition. The closeness to each model also encourages the representation of the concept in the images generated by the composed model as reflected by the improved text-to-image similarity metrics CLIP (Hessel et al., 2022) and BLIP (Li et al., 2022) scores in Table 1. We compute the similarity score between the generated images and each of the two prompts and compare the minimums. We also compare to the baseline of generating images from a combined prompt. Images generated with each approach along with implementation details and more experimental results can be found in Appendix F.

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 pre-trained model. We provide a theoretical characterization of the solutions to the constrained alignment and composition problems, and develop Lagrangian-based primal-dual training algorithms to approximate these solutions. Empirically, we demonstrate our constrained approach in image generation, applying it to alignment and composition, and show that our aligned or composed model satisfies constraints, effectively.

References

Biggs, B., Seshadri, A., Zou, Y., Jain, A., Golatkar, A., Xie, Y., Achille, A., Swaminathan, A., and Soatto, S. Diffusion soup: Model merging for text-to-image diffusion models. *arXiv preprint arXiv:2406.08431*, 2024.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22563–22575, 2023.

Boyd, S. P. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Bradley, A. and Nakkiran, P. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

Chamon, L. F., Paternain, S., Calvo-Fullana, M., and Ribeiro, A. Constrained learning with non-convex losses. *IEEE Transactions on Information Theory*, 69(3):1739–1760, 2022.

Chamon, L. F. O. and Ribeiro, A. Probably approximately correct constrained learning, 2021. URL https://arxiv.org/abs/2006.05487.

Chen, J., Zhang, R., Zhou, Y., and Chen, C. Towards aligned layout generation via diffusion model with aesthetic constraints. In *The Twelfth International Conference on Learning Representations*, 2024.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, pp. 02783649241273668, 2023.

Chidambaram, M., Gatmiry, K., Chen, S., Lee, H., and Lu, J. What does guidance do? a fine-grained analysis in a simple setting. *arXiv preprint arXiv:2409.13074*, 2024.

Christopher, J. K., Baek, S., and Fioretto, N. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024.

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Q. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control, 2025. URL https://arxiv.org/abs/2409.08861.

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc, 2024. URL https://arxiv.org/abs/2302.11552.

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

Fan, Y., Watkins, O., Du, Y., Liu, H., Ryu, M., Boutilier, C., Abbeel, P., Ghavamzadeh, M., Lee, K., and Lee, K. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2305.16381.

Giannone, G., Srivastava, A., Winther, O., and Ahmed, F. Aligning optimization trajectories with diffusion models for constrained design generation. *Advances in Neural Information Processing Systems*, 36:51830–51861, 2023.

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

Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. Clipscore: A reference-free evaluation metric for image captioning, 2022. URL https://arxiv.org/abs/2104.08718.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Khalafi, S., Ding, D., and Ribeiro, A. Constrained diffusion models via dual training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., and Levy, O. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:36652–36663, 2023.

Lee, K., Liu, H., Ryu, M., Watkins, O., Du, Y., Boutilier, C., Abbeel, P., Ghavamzadeh, M., and Gu, S. S. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Li, S., Kallidromitis, K., Gokul, A., Kato, Y., and Kozuka, K. Aligning diffusion models by optimizing human utility. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Liang, J., Christopher, J. K., Koenig, S., and Fioretto, F. Multi-agent path finding in continuous spaces with projected diffusion models. *arXiv preprint arXiv:2412.17993*, 2024.

Liang, J., Christopher, J. K., Koenig, S., and Fioretto, F. Simultaneous multi-robot motion planning with projected diffusion models. *arXiv preprint arXiv:2502.03607*, 2025.

Liu, B., Shao, S., Li, B., Bai, L., Xu, Z., Xiong, H., Kwok, J., Helal, S., and Xie, Z. Alignment of diffusion models: Fundamentals, challenges, and future. *arXiv preprint arXiv:2409.07253*, 2024.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Lyu, S. Interpretation and generalization of score matching, 2012. URL https://arxiv.org/abs/1205.2629.

Mou, W., Flammarion, N., Wainwright, M. J., and Bartlett, P. L. Improved bounds for discretization of langevin diffusions: Near-optimal rates without convexity, 2019. URL https://arxiv.org/abs/1907.11331.

Narasimhan, S. S., Agarwal, S., Rout, L., Shakkottai, S., and Chinchali, S. P. Constrained posterior sampling: Time series generation with hard constraints. *arXiv preprint arXiv:2410.12652*, 2024.

Prabhudesai, M., Goyal, A., Pathak, D., and Fragkiadaki, K. Aligning text-to-image diffusion models with reward backpropagation, 2024a. URL https://arxiv.org/abs/2310.03739.

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

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35: 36479–36494, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., 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.

Skreta, M., Atanackovic, L., Bose, J., Tong, A., and Neklyudov, K. The superposition of diffusion models using the itô density estimator. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=2o58Mbqkd2.

Sohrabi, M., Ramirez, J., Zhang, T. H., Lacoste-Julien, S., and Gallego-Posada, J. On pi controllers for updating lagrange multipliers in constrained optimization. In *International Conference on Machine Learning*, pp. 45922–45954. PMLR, 2024.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2022. URL https://arxiv.org/abs/2010.02502.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Uehara, M., Zhao, Y., Biancalani, T., and Levine, S. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024a.

Uehara, M., Zhao, Y., Black, K., Hajiramezanali, E., Scalia, G., Diamant, N. L., Tseng, A. M., Biancalani, T., and Levine, S. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024b.

Uehara, M., Zhao, Y., Black, K., Hajiramezanali, E., Scalia, G., Diamant, N. L., Tseng, A. M., Levine, S., and Biancalani, T. Feedback efficient online fine-tuning of diffusion models. In *Forty-first International Conference on Machine Learning*, 2024c.

Uehara, M., Zhao, Y., Hajiramezanali, E., Scalia, G., Eraslan, G., Lal, A., Levine, S., and Biancalani, T. Bridging model-based optimization and generative modeling via conservative fine-tuning of diffusion models. *Advances in Neural Information Processing Systems*, 37: 127511–127535, 2024d.

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

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Wang, L., Song, C., Liu, Z., Rong, Y., Liu, Q., and Wu, S. Diffusion models for molecules: A survey of methods and tasks. *arXiv preprint arXiv:2502.09511*, 2025.

Wu, X., Sun, K., Zhu, F., Zhao, R., and Li, H. Better aligning text-to-image models with human preference. *arXiv preprint arXiv:2303.14420*, 1(3), 2023a.

Wu, X., Sun, K., Zhu, F., Zhao, R., and Li, H. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2096–2105, 2023b.

Wu, X., Hao, Y., Zhang, M., Sun, K., Huang, Z., Song, G., Liu, Y., and Li, H. Deep reward supervisions for tuning text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 108–124, 2024.

Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 15903–15935, 2023a.

Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36, 2023b.

Yan, J. N., Gu, J., and Rush, A. M. Diffusion models without attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8239–8249, 2024.

Yang, K., Tao, J., Lyu, J., Ge, C., Chen, J., Shen, W., Zhu, X., and Li, X. 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*, pp. 8941–8951, 2024.

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

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

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

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

Zhao, H., Chen, H., Zhang, J., Yao, D. D., and Tang, W. Scores as actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. *arXiv preprint arXiv:2409.08400*, 2024.

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 fine-tuning diffusion models. Standard fine-tuning 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 pre-trained model; see (Fan & Lee, 2023; Xu et al., 2023b; Lee et al., 2023; Wu et al., 2023b; Zhang & Xu, 2023; Wu et al., 2024; Black et al., 2024; Clark et al., 2024; Zhang et al., 2024b) for studies using the single reward objective and (Uehara et al., 2024b; Zhao et al., 2024; Uehara et al., 2024d;c; Prabhudesai et al., 2024b; Fan et al., 2023; Han et al., 2024) 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 pre-trained one, while leaving the trade-off between reward and closeness unspecified; see (Uehara et al., 2024a) 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 pre-trained 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 pre-trained model subject to reward constraints. This offers a more principled alternative to existing ad hoc approaches (Chen et al., 2024; Giannone et al., 2023). Our new alignment formulation (i) offers a theoretical guarantee of an optimal trade-off between reward satisfaction and proximity to the pre-trained model, and (ii) allows for the direct imposition of multiple reward constraints. We also remark that our constrained learning approach generalizes to fine-tuning of diffusion models with preference (Wallace et al., 2024; Yang et al., 2024; Li et al., 2024).

Composition of diffusion models. Our constrained composition approach is related to prior work on compositional generation with diffusion models. When composing pre-trained 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 (Du et al., 2024; Bradley & Nakkiran, 2024; Chidambaram et al., 2024). 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 (Liu et al., 2022; Du et al., 2024), MCMC sampling (Du et al., 2024), diffusion soup (Biggs et al., 2024), and superposition (Skreta et al., 2025). 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 pre-trained diffusion models. Moreover, our constrained composition approach also generalizes to mixture composition, offering advantages over prior work (Liu et al., 2022; Du et al., 2024; Biggs et al., 2024; Skreta et al., 2025).

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 (Liang et al., 2024; Christopher et al., 2024; Liang et al., 2025), constrained posterior sampling (Narasimhan et al., 2024), and proximal Langevin dynamics (Zampini et al., 2025). 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 (Khalafi et al., 2024). 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:

$$D_{\mathrm{KL}}(p_{0:T}(x_{0:T}; s_p)) = D_{\mathrm{KL}}(p(x_{0:T}; s_p)) \tag{19}$$

C.1. Proof of Lemma 1

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

$$p(x_{t-1}|x_t; s) = \mathcal{N}\left(\sqrt{\frac{\alpha_{t-1}}{\alpha_t}}x_t \,+\, \beta_t\, s(x_t, t),\, \sigma_t^2 I\right) \tag{20}$$

Using (20) we expand the path-wise KL:

$$\begin{aligned}
&D_{\mathrm{KL}}(p_{0:T}(\cdot; s_p) \,\|\, p_{0:T}(\cdot; s_q)) \\
&= \quad \mathbb{E}_{x_{0:T} \sim p}\left[\log p(x_{0:T}; s_p) - \log p(x_{0:T}; s_q)\right] \\
&\overset{(a)}{=} \quad \mathbb{E}_{x_T \sim p_{T+1}(\cdot), x_{T-1} \sim p_T(\cdot \,|\, x_T), \dots, x_0 \sim \mathfrak{p}_1(\cdot \,|\, x_1)}\left[\sum_{t=T}^{1} \log \frac{p(x_{t-1} \,|\, x_t; s_p)}{p(x_{t-1} \,|\, x_t; s_q)}\right] \\
&\overset{(b)}{=} \quad \sum_{t=T}^{1} \mathbb{E}_{x_T \sim p_{T+1}(\cdot), x_{T-1} \sim p_T(\cdot \,|\, x_T), \dots, x_0 \sim \mathfrak{p}_1(\cdot \,|\, x_1)}\left[\log \frac{p(x_{t-1} \,|\, x_t; s_p)}{p(x_{t-1} \,|\, x_t; s_q)}\right] \\
&\overset{(c)}{=} \quad \sum_{t=T}^{1} \mathbb{E}_{x_{0:T} \sim p}\left[D_{\mathrm{KL}}(p(x_{t-1} \,|\, x_t; s_p) \,\|\, p(x_{t-1} \,|\, x_t; s_q))\right] \\
&\overset{(d)}{=} \quad \sum_{t=T}^{1} \mathbb{E}_{x_t \sim p_{t+1}}\left[\frac{\beta_t^2}{2\sigma_t^2}\left\|s_p(x_t, t) - s_q(x_t, t)\right\|^2\right] \\
&\overset{(e)}{=} \quad \sum_{t=T}^{1} \mathbb{E}_{\{p_t\}}\left[\frac{\beta_t^2}{2\sigma_t^2}\left\|s_p(x_t, t) - s_q(x_t, t)\right\|^2\right]
\end{aligned}$$

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

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

- We begin by giving a few definitions for continuous time diffusion processes.
- Then in Lemma 4 we characterize how the KL between the marginals of two processes changes over time.
- Using Lemma 4 we prove Lemma 5 which is the analogue of Lemma 2 in continuous time.
- Next, Lemmas 6, 7, 8, allow us to bound the discretization error ϵ_T incurred when going from continuous time processes to corresponding discretized processes and complete the proof.

Notation Guide: In this Section(C.2) 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 $\tau = 0$. [2]
- 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: $\mathfrak{p}_\tau(x, s)$

Continuous time Preliminaries. 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:

$$d\mathfrak{X}_\tau = (\frac{\dot{\bar{\alpha}}_\tau}{2\bar{\alpha}_\tau}\mathfrak{X}_\tau + (\frac{\dot{\bar{\alpha}}_\tau}{2\bar{\alpha}_\tau} + \frac{\sigma_\tau^2}{2})s(\mathfrak{X}_\tau, \tau))dt + \sigma_\tau d\mathfrak{B}_\tau, \quad \mathfrak{X}_0 \sim \mathcal{N}(0, I) \tag{21}$$

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 (21) 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:

$$X_{t-1} \; = \; \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} X_t \; + \; \beta_t \, s(X_t, t) \; + \; \sigma_t \epsilon_t \tag{22}$$

Up to first order approximation, the discrete time process (22) is the Euler-Maruyama discretization of the continuous time process (21). A uniform discretization of time is assumed i.e. $\tau = 1 - \frac{t}{T}$ (See (Domingo-Enrich et al., 2025) 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:

$$\bar{\mathfrak{X}}_\tau = \alpha_\tau \bar{\mathfrak{X}}_0 + \zeta_\tau \bar{\mathfrak{X}}_1 \tag{23}$$

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 (21) has the same marginals as the reference flow (23) 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. $\alpha_\tau = \sqrt{1 - \bar{\alpha}_\tau}$, $\zeta_\tau = \sqrt{\bar{\alpha}_\tau}$.

The following Lemma which generalizes Theorem 1 from (Lyu, 2012), characterizes how the KL between 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:*

$$\frac{d}{d\tau} D_{\mathrm{KL}}(\bar{\mathfrak{p}}_\tau(\cdot) || \bar{\mathfrak{q}}_\tau(\cdot)) = -\gamma_\tau \dot{\gamma}_\tau D_{\mathrm{F}}(\bar{\mathfrak{p}}_\tau(\cdot) || \bar{\mathfrak{q}}_\tau(\cdot)) \tag{24}$$

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

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

$$\bar{\mathfrak{Y}}_\tau := \frac{1}{\alpha_\tau} \bar{\mathfrak{X}}_\tau = \bar{\mathfrak{X}}_1 + \gamma_\tau \bar{\mathfrak{X}}_0 \tag{25}$$

[2] For consistency with other works from whom we will utilize some results in our proofs, namely (Domingo-Enrich et al., 2025; Lyu, 2012), the direction of time we consider in continuous time is reversed compared to discrete time. This does not affect any of our derivations and results beyond a small change of notation.

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 (Lyu, 2012) to show that (24) holds for $\widetilde{\mathfrak{p}}_\tau, \widetilde{\mathfrak{q}}_\tau$. Their Theorem is for the specific case of $\gamma_\tau = \sqrt{1-t}$.[3]

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

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

$$\frac{d}{dt}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) = \gamma_\tau \dot{\gamma}_\tau \Delta_{\bar{\mathfrak{Y}}_\tau} \widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau). \tag{26}$$

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

$$\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) = \int_{\bar{\mathfrak{X}}_1} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1), \tag{27}$$

Taking the derivative we have:

$$\begin{aligned} \frac{d}{dt}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) &= \int_{\bar{\mathfrak{X}}_1} \frac{\dot{\gamma}_\tau \left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{\gamma_\tau^3} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1) \\ &- \int_{\bar{\mathfrak{X}}_1} \frac{d}{\gamma_\tau} \frac{\dot{\gamma}_\tau}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1). \end{aligned} \tag{28}$$

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:

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

Taking the divergence of the gradient, we have:

$$\begin{aligned} \Delta_{\bar{\mathfrak{Y}}_\tau}\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau) &= \int_{\bar{\mathfrak{X}}_1} \frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{\gamma_\tau^4} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1), \\ &- \int_{\bar{\mathfrak{X}}_1} \frac{d}{\gamma_\tau^2} \frac{1}{(2\pi\gamma_\tau^2)^{d/2}} \exp\left(-\frac{\left\|\bar{\mathfrak{Y}}_\tau - \bar{\mathfrak{X}}_1\right\|^2}{2\gamma_\tau^2}\right) \mathfrak{p}_1(\bar{\mathfrak{X}}_1). \end{aligned} \tag{30}$$

Comparing Equations (28) and (30) proves the result.

□

Lemma 4.2. *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*

$$\frac{\Delta_x f(x)}{f(x)} = \Delta_x \log f(x) + \left\|\nabla_x \log f(x)\right\|^2. \tag{31}$$

[3] Just to avoid any confusion, in (Lyu, 2012), 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:

$$\begin{aligned} D_F(p\,\|\,q) &= \int_x p(x)\left\|\nabla \log p(x) - \nabla \log q(x)\right\|^2 dx \\ &= \int_x p(x)\left\|\frac{\nabla p(x)}{p(x)} - \frac{\nabla q(x)}{q(x)}\right\|^2 dx \\ &= \int_x p(x)\left(\left\|\frac{\nabla p(x)}{p(x)}\right\|^2 + \left\|\frac{\nabla q(x)}{q(x)}\right\|^2 - 2\frac{\nabla p(x)^\top \nabla q(x)}{p(x)q(x)}\right) dx \end{aligned} \tag{32}$$

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

$$\begin{aligned} \int_{x\in\Omega} \nabla p(x)^\top \frac{\nabla q(x)}{q(x)} dx &= \int_{x\in\Omega} \nabla p(x)^\top (\nabla \log q(x)) dx \\ &= -\int_{x\in\Omega} p(x)\Delta \log q(x) dx + \int_\Gamma p(x)(\nabla \log q(x)^\top \widehat{n}) d\Gamma \end{aligned} \tag{33}$$

Assuming that both $p(x)$ and $q(x)$ are smooth and fast-decaying, the boundary term in (33) vanishes.

Then we can combine (32) and (33) to write:

$$D_F(p\,\|\,q) = \int_x p(x)\left(\|\nabla \log p(x)\|^2 + \|\nabla \log q(x)\|^2 + 2\Delta_x \log q(x)\right) dx \tag{34}$$

Returning to our distributions $\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)$ and $\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)$ we can rewrite (34) as:

$$D_F(\widetilde{\mathfrak{p}}_\tau(\cdot)\,\|\,\widetilde{\mathfrak{q}}_\tau(\cdot)) = \int_{\mathfrak{Y}_\tau} \widetilde{\mathfrak{p}}_\tau(\mathfrak{Y}_\tau)\left(\left\|\nabla \log \widetilde{\mathfrak{p}}_\tau(\mathfrak{Y}_\tau)\right\|^2 + \|\nabla \log \widetilde{\mathfrak{q}}_\tau(\mathfrak{Y}_\tau)\|^2 + 2\Delta_{\mathfrak{Y}_\tau} \log \widetilde{\mathfrak{q}}_\tau(\mathfrak{Y}_\tau)\right) d\mathfrak{Y}_\tau \tag{35}$$

For conciseness in notation, we drop references to variables $\bar{\mathfrak{Y}}_\tau$ and $\bar{\mathfrak{X}}_1$ in the integration, the density functions, and the operators whenever this does not lead to ambiguity. We start by applying Lemma 4.2 to Equation (34):

$$\begin{aligned} D_F(\widetilde{p}\|\widetilde{q}) &= \int \widetilde{p}\left(|\nabla \log \widetilde{p}|^2 + |\nabla \log \widetilde{q}|^2 + 2\Delta \log \widetilde{q}\right), \\ &= \int \widetilde{p}\left(|\nabla \log \widetilde{p}|^2 + \frac{\Delta \widetilde{q}}{\widetilde{q}} + \Delta \log \widetilde{q}\right). \end{aligned} \tag{36}$$

Next, we expand the derivative of the KL divergence:

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

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

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

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

$$\begin{aligned} \frac{d}{d\tau} D_{KL}(\widetilde{p}\|\widetilde{q}) &= \int \frac{d}{d\tau}\widetilde{p} \log \widetilde{p} - \int \frac{d}{d\tau}\widetilde{p} \log \widetilde{q} - \int \widetilde{p} \frac{d}{d\tau} \log \widetilde{q}, \\ &= \gamma_\tau \dot{\gamma}_\tau \left(\int \Delta \widetilde{p} \log \widetilde{p} - \int \Delta \widetilde{p} \log \widetilde{q} - \int \widetilde{p} \frac{\Delta \widetilde{q}}{\widetilde{q}}\right). \end{aligned} \tag{37}$$

Using integration by parts, the first term in (37) is changed to:

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

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

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

The second term in (37) can be manipulated similarly, by first using integration by parts to get:

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

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

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

The limits at the boundary values are all zero due to the smoothness and fast decay properties of $\widetilde{p}(\vec{y})$. Now collecting all terms, we have $\int \widetilde{p}\log\widetilde{p} = -\int \widetilde{p}|\nabla\log\widetilde{p}|^2$ and $\int \widetilde{p}\log\widetilde{q} = \int \widetilde{p}\Delta\log\widetilde{q}$. Thus (37) becomes:

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

Combining with (36), this leads to the following:

$$\frac{d}{d\tau}D_{\text{KL}}(\widetilde{p}_\tau||\widetilde{q}_\tau) = -\gamma_\tau\dot{\gamma}_\tau D_{\text{F}}(\widetilde{p}_\tau||\widetilde{q}_\tau). \tag{38}$$

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

$$\frac{d}{d\tau}D_{\text{KL}}(\widetilde{\mathfrak{p}}_\tau(\cdot)\,\|\,\widetilde{\mathfrak{q}}_\tau(\cdot)) = -\gamma_\tau\dot{\gamma}_\tau D_{\text{F}}(\widetilde{\mathfrak{p}}_\tau(\cdot)\,\|\,\widetilde{\mathfrak{q}}_\tau(\cdot)). \tag{39}$$

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.:

$$D_{\text{KL}}(\widetilde{\mathfrak{p}}_\tau(\cdot)||\widetilde{\mathfrak{q}}_\tau(\cdot)) = \int \widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)\log\frac{\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)}{\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)}d\bar{\mathfrak{Y}}_\tau = \int \mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)\log\frac{\mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)}{\mathfrak{q}_\tau(\bar{\mathfrak{X}}_\tau)}d\bar{\mathfrak{X}}_\tau = D_{\text{KL}}(\mathfrak{p}_\tau(\cdot)||\mathfrak{q}_\tau(\cdot)) \tag{40}$$

$$\begin{aligned} D_{\text{F}}(\widetilde{\mathfrak{p}}_\tau(\cdot)||\widetilde{\mathfrak{q}}_\tau(\cdot)) &= \int \widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)\left\|\frac{\nabla\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)}{\widetilde{\mathfrak{p}}_\tau(\bar{\mathfrak{Y}}_\tau)} - \frac{\nabla\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)}{\widetilde{\mathfrak{q}}_\tau(\bar{\mathfrak{Y}}_\tau)}\right\|^2 d\bar{\mathfrak{Y}}_\tau \\ &= \int \mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)\left\|\frac{\nabla\mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)}{\mathfrak{p}_\tau(\bar{\mathfrak{X}}_\tau)} - \frac{\nabla\mathfrak{q}_\tau(\bar{\mathfrak{X}}_\tau)}{\mathfrak{q}_\tau(\bar{\mathfrak{X}}_\tau)}\right\|^2 d\bar{\mathfrak{X}}_\tau = D_{\text{F}}(\mathfrak{p}_\tau(\cdot)||\mathfrak{q}_\tau(\cdot)) \end{aligned} \tag{41}$$

Thus we can replace the divergences in (38) with those of the non-scaled distribution, which concludes the proof. □

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

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* (21) *with $s_{\mathfrak{p}}$ and $s_{\mathfrak{q}}$ is given by*

$$D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot\,; s_{\mathfrak{p}}) \,\|\, \mathfrak{p}_0(\cdot\,; s_{\mathfrak{q}})) \;=\; \int_{\tau=0}^{1} \widetilde{\omega}_\tau \,\mathbb{E}_{x\,\sim\,\mathfrak{p}_\tau(\,\cdot\,; s_p)} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] \tag{42}$$

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

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

$$\begin{aligned}
D_{\mathrm{KL}}(\mathfrak{p}_1(\cdot)||\mathfrak{q}_1(\cdot)) &= D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot)||\mathfrak{q}_0(\cdot)) - \int_{\tau=0}^{1} \dot{\gamma_\tau}\gamma_\tau D_{\mathrm{F}}(\widetilde{\mathfrak{p}}_\tau(\cdot)||\widetilde{\mathfrak{q}}_\tau(\cdot))d\tau \\
&= -\int_{\tau=0}^{1} \dot{\gamma_\tau}\gamma_\tau \mathbb{E}_{x\sim\widetilde{\mathfrak{p}}_\tau} \left[\left\|\nabla \log \widetilde{\mathfrak{p}}_\tau(x) - \nabla \log \widetilde{\mathfrak{q}}_\tau(x)\right\|_2^2 \right] d\tau \\
&= -\int_{\tau=0}^{1} \dot{\gamma_\tau}\gamma_\tau \mathbb{E}_{x\sim\widetilde{\mathfrak{p}}_\tau} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] d\tau \\
&= \int_{\tau=0}^{1} \frac{\dot{\alpha_\tau}}{\alpha_\tau^3} \mathbb{E}_{x\sim\widetilde{\mathfrak{p}}_\tau} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] d\tau
\end{aligned} \tag{43}$$

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 $\alpha_\tau^2 + \zeta_\tau^2 = 1$:

$$\begin{aligned}
\dot{\gamma_\tau}\gamma_\tau &= \frac{d}{d\tau}\left(\frac{\zeta_\tau}{\alpha_\tau}\right)\frac{\zeta_\tau}{\alpha_\tau} \\
&= \frac{\dot{\zeta_\tau}\zeta_\tau\alpha_\tau - \dot{\alpha}_\tau\zeta_\tau^2}{\alpha_\tau^3} \\
&= \frac{-\dot{\alpha}_\tau\alpha_\tau^2 - \dot{\alpha}_\tau(1-\alpha_\tau^2)}{\alpha_\tau^3} \\
&= -\frac{\dot{\alpha_\tau}}{\alpha_\tau^3}
\end{aligned} \tag{44}$$

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

We now start bridging the gap between continuous and discrete time. First we present a result from (Mou et al., 2019):

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

$$D_{\mathrm{KL}}(p_t(\cdot\,; s_p)||\mathfrak{p}_{t/T}(\cdot\,; s_p)) \leq \frac{c}{T^2} \tag{45}$$

where c is a constant depending on assumptions.

See (Mou et al., 2019) for the proof (Theorem 1). Next we need to characterize the sensitivity of the KL divergence to perturbations in the first and second arguments.

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

$$|D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot\,; s_p)||\mathfrak{p}_0(\cdot\,; s_q)) - D_{\mathrm{KL}}(p_0(\cdot\,; s_p)||p_0(\cdot\,; s_q))| \leq \epsilon_1(T), \tag{46}$$

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

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:

$$\begin{aligned} |D_{\mathrm{KL}}(\widehat{\pi} \parallel \rho) - D_{\mathrm{KL}}(\pi \parallel \rho)| &= \int \widehat{\pi} \log\left(\frac{\widehat{\pi}}{\rho}\right) - \int \pi \log\left(\frac{\pi}{\rho}\right) + \int (\widehat{\pi} \log \pi - \widehat{\pi} \log \pi) \\ &= D_{\mathrm{KL}}(\widehat{\pi} \parallel \pi) + \int (\widehat{\pi} - \pi) \log\left(\frac{\pi}{\rho}\right) \\ &\leq D_{\mathrm{KL}}(\widehat{\pi} \parallel \pi) + \max\left(\left|\log\left(\frac{\pi}{\rho}\right)\right|\right) \int |\widehat{\pi} - \pi| \\ &= D_{\mathrm{KL}}(\widehat{\pi} \parallel \pi) + 2 \log M \, d_{\mathrm{TV}}(\widehat{\pi}, \pi) \end{aligned} \tag{47}$$

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:

$$\begin{aligned} |D_{\mathrm{KL}}(\widehat{\pi} \parallel \widehat{\rho}) - D_{\mathrm{KL}}(\widehat{\pi} \parallel \rho)| &= \left|\int \widehat{\pi} \log\left(\frac{\widehat{\pi}}{\widehat{\rho}}\right) - \int \widehat{\pi} \log\left(\frac{\widehat{\pi}}{\rho}\right)\right| \\ &= -\int \widehat{\pi} \log\left(\frac{\widehat{\rho}}{\rho}\right) = -\int \widehat{\pi} \log\left(1 + \frac{\widehat{\rho} - \rho}{\rho}\right) \\ &\leq \int \widehat{\pi} \frac{\widehat{\rho} - \rho}{\rho} = \int \frac{\widehat{\pi}}{\pi} \frac{\pi}{\rho} (\widehat{\rho} - \rho) \\ &\leq \max\left(\frac{\pi}{\rho}\right) \int |\widehat{\rho} - \rho| = 2M \, d_{\mathrm{TV}}(\widehat{\rho}, \rho). \end{aligned} \tag{48}$$

Using (47), (48) we get:

$$\begin{aligned} |D_{\mathrm{KL}}(\widehat{\pi} \parallel \widehat{\rho}) - D_{\mathrm{KL}}(\pi \parallel \rho)| &\leq |D_{\mathrm{KL}}(\widehat{\pi} \parallel \widehat{\rho}) - D_{\mathrm{KL}}(\widehat{\pi} \parallel \rho)| + |D_{\mathrm{KL}}(\widehat{\pi} \parallel \rho) - D_{\mathrm{KL}}(\pi \parallel \rho)| \\ &\leq D_{\mathrm{KL}}(\widehat{\pi} \parallel \pi) + 2M \, d_{\mathrm{TV}}(\widehat{\rho}, \rho) + 2 \log M \, d_{\mathrm{TV}}(\widehat{\pi}, \pi) \\ &\leq D_{\mathrm{KL}}(\widehat{\pi} \parallel \pi) + 2M \sqrt{\frac{1}{2} D_{\mathrm{KL}}(\widehat{\rho} \parallel \rho)} + 2 \log M \sqrt{\frac{1}{2} D_{\mathrm{KL}}(\widehat{\pi} \parallel \pi)} \end{aligned} \tag{49}$$

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 (49) to diffusion models:

$$\begin{aligned} |D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot; s_p) || \mathfrak{p}_0(\cdot; s_q)) - D_{\mathrm{KL}}(p_0(\cdot; s_p) || p_0(\cdot; s_q))| \leq{}& D_{\mathrm{KL}}(p_0(\cdot; s_p) \parallel \mathfrak{p}_0(\cdot; s_p)) \\ &+ 2M \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_0(\cdot; s_q) \parallel \mathfrak{p}_0(\cdot; s_q))} \\ &+ 2 \log M \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_0(\cdot; s_p) \parallel \mathfrak{p}_0(\cdot; s_p))} \end{aligned} \tag{50}$$

Furthermore from Lemma 6 we know:

$$D_{\mathrm{KL}}(p_0(\cdot; s_p) \parallel \mathfrak{p}_0(\cdot; s_p)) \leq c/T^2, \quad D_{\mathrm{KL}}(p_0(\cdot; s_q) \parallel \mathfrak{p}_0(\cdot; s_q)) \leq c/T^2 \tag{51}$$

Putting together (50) and (51) we get:

$$|D_{\mathrm{KL}}(\mathfrak{p}_0(\cdot; s_p) || \mathfrak{p}_0(\cdot; s_q)) - D_{\mathrm{KL}}(p_0(\cdot; s_p) || p_0(\cdot; s_q))| \leq \epsilon_1(T) \tag{52}$$

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.

□

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

$$B_1 := \sup_{x,\tau} \|s_p(x,\tau) - s_q(x,\tau)\|_2 \tag{53}$$

$$B_2 := \sup_{x,\tau} \left\| \frac{d}{d\tau}(s_p(x,\tau) - s_q(x,\tau)) \right\|_2 \tag{54}$$

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

$$\left| \int_{\tau=0}^{1} \widetilde{\omega}_\tau \, \mathbb{E}_{x\sim \mathfrak{p}_\tau(\cdot\,;s_p)} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] - \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x\sim p_t(\cdot\,;s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \right| \leq \epsilon_2(T) \tag{55}$$

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

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

$$\left| \int_{\tau=0}^{1} \widetilde{\omega}_\tau \, \mathbb{E}_{x\sim \mathfrak{p}_\tau(\cdot\,;s_p)} \left[\|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \right] - \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x\sim \mathfrak{p}_{t/T}(\cdot\,;s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \right| \tag{56}$$

$$= \left| \int_{\tau=0}^{1} f(\tau) d\tau - \sum_{t=0}^{T} f(t/T) \cdot \frac{1}{T} \right| \leq \frac{1}{T} \sup_{\tau\in[0,1]} \left| \frac{df}{d\tau} \right| \tag{57}$$

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:

$$\frac{df}{d\tau} = \frac{d}{d\tau} \left(\int \mathfrak{p}_\tau(x\,;s_p) \, \|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \, dx \right) \tag{58}$$

$$= \int \frac{d}{d\tau}(\mathfrak{p}_\tau(x\,;s_p)) \, \|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \, dx + \int \mathfrak{p}_\tau(x\,;s_p) \frac{d}{d\tau}(\|s_p(x,\tau) - s_q(x,\tau)\|_2^2) dx \tag{59}$$

We bound each term in (59) separately. Then the first term in (59) is bounded because $\frac{d}{d\tau}(\mathfrak{p}_\tau(x\,;s_p))$ is finite as characterized in Lemma 4.1. The second term in (59) we expand further:

$$\int \mathfrak{p}_\tau(x\,;s_p) \frac{d}{d\tau}(\|s_p(x,\tau) - s_q(x,\tau)\|_2^2) dx = \int 2\mathfrak{p}_\tau(x\,;s_p) \langle s_p(x,\tau) - s_q(x,\tau), \, \frac{ds_p(x,\tau)}{d\tau} - \frac{ds_q(x,\tau)}{d\tau} \rangle dx \tag{60}$$

$$\leq 2 \sup_{x,\tau} \|s_p(x,\tau) - s_q(x,\tau)\|_2 \left\| \frac{d}{d\tau}(s_p(x,\tau) - s_q(x,\tau)) \right\|_2 \leq 2B_1 B_2 \tag{61}$$

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.

$$\left| \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x\sim \mathfrak{p}_{t/T}(\cdot\,;s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] - \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} \, \mathbb{E}_{x\sim p_t(\cdot\,;s_p)} \left[\|s_p(x,t) - s_q(x,t)\|_2^2 \right] \right| \tag{62}$$

$$\leq \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/T} d_{TV}(p_t(\cdot\,;s_p), \mathfrak{p}_{t/T}(\cdot\,;s_p)) \cdot \sup_{x} \|s_p(x,\tau) - s_q(x,\tau)\|_2^2 \tag{63}$$

$$\leq \sum_{t=0}^{T} \frac{1}{T} \widetilde{\omega}_{t/\tau} \sqrt{\frac{c}{T^2}} \cdot B_1^2 \leq T \cdot \frac{1}{T} \cdot \sqrt{\frac{c}{T^2}} \cdot B_1^2 = O(\frac{1}{T}) \tag{64}$$

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

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

$$D_{\mathrm{KL}}(p_0(\cdot\,; s_p)\,\|\,p_0(\cdot\,; s_q)) \;=\; \sum_{t\,=\,0}^{T} \widetilde{\omega}_t\, \mathbb{E}_{x\,\sim\, p_t(\cdot\,;s_p)}\left[\,\|s_p(x,t) - s_q(x,t)\|_2^2\,\right] \,+\, \epsilon_T \tag{65}$$

where $|\epsilon_T| \leq \epsilon_1(T) + \epsilon_2(T) = O(1/T)$. (We abuse notation to denote $\frac{1}{T}\widetilde{\omega}_{t/T}$ as $\widetilde{\omega}_t$ in (65) 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,

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

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 path space $\{p_{0:T}(\cdot)\}$. Since the 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

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

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

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

Since the function class $\mathcal{S}$ is expressive enough, any path $p_{0:T}(\cdot)$ can be represented as $p_{0:T}(\cdot; s_p)$ 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 $\bar{D}_{\mathrm{ALI}}(\lambda)$ in the path and function spaces are the same. Hence, the dual value for (SR-A) remains to be $\bar{D}_{\mathrm{ALI}}(\lambda^\star)$. Thus, $(s^\star_p, \lambda^\star)$ is a saddle point of the Lagrangian $\bar{L}_{\mathrm{ALI}}(s_p, \lambda) := L_{\mathrm{ALI}}(p_{0:T}(\cdot; s_p), \lambda)$,

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

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

C.5. Proof of Theorem 3

Proof. By the definition,

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

By taking $\lambda = \lambda^\star$, we obtain a primal problem: $\operatorname{maximize}_{p \,\in\, \mathcal{P}, u \,\geq\, 0} L_{\text{AND}}(p, u; \lambda^\star)$, 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,

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

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,

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

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

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

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:

$$L_{\text{AND}}(p, u; \lambda) \;=\; u + \sum_{i\,=\,1}^{m} \lambda_i \left(D_{\text{KL}}(p \,\|\, q_{\text{AND}}^{(\lambda)}) - u \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \tag{66}$$

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 (66) that:

$$L_{\text{AND}}(p_0(\cdot\,; s), u; \lambda) \;=\; u + \sum_{i\,=\,1}^{m} \lambda_i \left(D_{\text{KL}}(p_0(\cdot\,; s) \,\|\, q_{\text{AND},0}^{(\lambda)}(\cdot)) - u \right) - \mathbf{1}^\top \lambda \log Z_{\text{AND}}(\lambda). \tag{67}$$

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:

$$L_{\text{AND}}(p(\cdot\,; s); \lambda) \;=\; L_{\text{AND}}(s, \lambda) \;=\; D_{\text{KL}}(p_0(\cdot\,; s) \,\|\, q_{\text{AND},0}^{(\lambda)}) - \log Z_{\text{AND}}(\lambda). \tag{68}$$

The second term $\log Z_{\text{AND}}(\lambda)$ does not depend on s, thus it suffices to minimize $D_{\text{KL}}(p_0(\cdot\,; s) \,\|\, q_{\text{AND},0}^{(\lambda)})$ to find the Lagrangian minimizer which we call $s^{(\lambda)}$. The KL is minimized when $p_0(\cdot\,; s^{(\lambda)}) = q_{\text{AND},0}^{(\lambda)}$. If we have access to samples from $q_{\text{AND},0}^{(\lambda)}$, we can fit s to $q_{\text{AND},0}^{(\lambda)}$ by optimizing the Denoising score matching objective similar to Equation (1) in (Song et al., 2021):

$$L_{\text{sm}}(s, \lambda) \;=\; \sum_{t\,=\,0}^{T} \omega_t \, \mathbb{E}_{x_0 \sim q_{\text{AND}}^{(\lambda)}(\cdot)} \mathbb{E}_{x_t \sim q(x_t|x_0)} \left[\left\| s(x, t) - \nabla \log q(x_t|x_0) \right\|^2 \right] \tag{69}$$

From (Song et al., 2021) we know that given sufficient data and predictor capacity of s we have $\operatorname{argmin}_s L_{\text{sm}}(s, \lambda) \simeq q_{\text{AND},0}^{(\lambda)}$. □

D. Composition with Forward KL Divergences

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

$$\begin{array}{ll} \underset{u\in\mathbb{R},\, p\,\in\,\mathcal{P}}{\text{minimize}} & u \\ \text{subject to} & D_{\text{KL}}(q^i \,\|\, p) \;\leq\; u \quad \text{for } i = 1, \ldots, m. \end{array} \tag{70}$$

In the case of diffusion models, the KL divergence in (70) becomes the forward path-wise KL between the processes:

$$\begin{array}{ll} \underset{u\in\mathbb{R},\, p\,\in\,\mathcal{P}}{\text{minimize}} & u \\ \text{subject to} & D_{\text{KL}}(q^i_{0:T}(\cdot) \,\|\, p_{0:T}(\cdot\,; s)) \;\leq\; u \quad \text{for } i = 1, \ldots, m. \end{array} \tag{71}$$

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

In summary, the two key differences of Problem (71) to Problem (UR-A) are: (i) The closeness of a model p to a pretrained model q^i is measured by the forward KL divergence $D_{\text{KL}}(q^i \,\|\, p)$, instead of the reverse KL divergence $D_{\text{KL}}(p \,\|\, q^i)$; (ii) The distributions $\{q^i\}_{i=1}^m$ can be the distributions underlying m datasets, not necessarily m pretrained models.

Regardless of whether the q^i represent pre-trained models or datasets, evaluating $D_{\text{KL}}(q^i_{0:T}(\cdot) \,\|\, p_{0:T}(\cdot\,; s))$ 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 (71) by replacing the KL with the Evidence Lower Bound (Elbo):

$$\begin{array}{ll} \underset{u\in\mathbb{R},\, p\,\in\,\mathcal{P}}{\text{minimize}} & u \\ \text{subject to} & \text{Elbo}(q^i_{0:T}; p_{0:T}) \;\leq\; u \quad \text{for } i = 1, \ldots, m \end{array} \tag{72}$$

where the Elbo is defined as

$$\text{Elbo}(q_{0:T}; p_{0:T}) \;:=\; \mathbb{E}_{x_0\sim q_0}\mathbb{E}_{q(x_{1:T}|x_0)} \log \frac{p_{0:T}(x_{0:T})}{q(x_{1:T}|x_0)}. \tag{73}$$

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 (Khalafi et al., 2024) for more details on this)

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

$$q_{\text{mix}}^{(\lambda)}(\cdot) \;=\; \sum_{i=1}^{m} \frac{\lambda_i}{\lambda^\top 1} q^i(\cdot), \tag{74}$$

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

$$H(q) \;:=\; -\mathbb{E}_{x\sim q}[\log q(x)] \tag{75}$$

Theorem 5. *Problem (72) is equivalent to the following unconstrained problem:*

$$\underset{p\,\in\,\mathcal{P}}{\text{minimize}}\; D_{\text{KL}}(q_{\text{mix}}^{(\lambda^\star)} \,\|\, p) \tag{76a}$$

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

$$p^\star = q_{\text{mix}}^{(\lambda^\star)}. \tag{76b}$$

Unlike the reverse KL case, here we can characterize the optimal dual multipliers, and the optimal solution further; Note that the optimal dual multiplier $\lambda^\star = \mathrm{argmax}_{\lambda \geq 0}\, D(\lambda) = \mathrm{argmax}_{\lambda \geq 0}\, H(q_{\text{mix}}(\cdot; \lambda^\star))$ 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 (72) can be characterized explicitly as*

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

E. Algorithm Details

E.1. Alignment

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

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

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

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,

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

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

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

```
Input: total diffusion steps $T$, diffusion parameter $\alpha_t$, total dual iterations $H$, number of primal steps per dual update $N$, dual step size $\eta_d$, primal step size $\eta_p$, initial model parameters $\theta(0)$.
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
            $\theta_{n+1} = \theta_n - \eta_p \cdot \nabla_\theta \bar{L}_{\text{ALI}}(\theta, \lambda^{(n)})$   (77)
    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 $i = 1, \cdots, m$:
        $\lambda_i(h+1) = \left[\lambda_i(h) + \eta_d(\mathbb{E}_{x_0 \sim p_0(\cdot; s_\theta)}\left[r_i(x_0)\right] - b_i)\right]_+$   (78)
end for
```

We now discuss the practicality of the primal gradient descent step (77) regarding the Lagrangian function,

$$\bar{L}_{\text{ALI}}(\theta, \lambda) = D_{\text{KL}}\left(p_{0:T}(\cdot; s_\theta) \,\|\, q_{0:T}(\cdot; s_q)\right) - \sum_i \lambda_i(\mathbb{E}_{x_0 \sim p_0(\cdot; s_\theta)}\left[r_i(x_0)\right] - b_i) \tag{79}$$

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 (Fan et al., 2023)) to take the gradient inside the expectation.

Lemma 9. *If $p_\theta(x_{0:T})r(x_0)$ and $\nabla_\theta p_\theta(x_{0:T})r(x_0)$ are continuous functions of 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

$$\nabla_\theta D_{\text{KL}}\big(\, p_{0:T}(\cdot; s_\theta) \,\|\, q_{0:T}(\cdot; s_q) \,\big) \;=\; \nabla_\theta \left(\sum_{t\,=\,1}^{T} \mathbb{E}_{x_t \,\sim\, p_t(\cdot; s_\theta)} \left[\frac{1}{2\sigma_t^2} \| s_\theta(x_t, t) - s_q(x_t, t) \|^2 \right] \right)$$

$$= \nabla_\theta \left(\sum_{t\,=\,1}^{T} \mathbb{E}_{x_t \,\sim\, p_t(\cdot; s_\theta)} \left[D_{\text{KL}}(p(x_{t-1} \,|\, x_t; s_\theta) \,\|\, p(x_{t-1} \,|\, x_t; s_q)) \right] \right)$$

$$= \sum_{t\,=\,1}^{T} \mathbb{E}_{x_t \,\sim\, p_t(\cdot; s_\theta)} \left[\nabla_\theta D_{\text{KL}}(p(x_{t-1} \,|\, x_t; s_\theta) \,\|\, p(x_{t-1} \,|\, x_t; s_q)) \right]$$

$$+ \sum_{t\,=\,1}^{T} \mathbb{E}_{x_t \,\sim\, p_t(\cdot; s_\theta)} \left[\sum_{t' > t}^{T} \nabla_\theta \log p(x_{t'-1} \mid x_{t'};\, s_\theta) D_{\text{KL}}(p(x_{t-1} \,|\, x_t; s_\theta) \,\|\, p(x_{t-1} \,|\, x_t; s_q)) \right].$$

The second term we ignore in practice for simplicity without hurting performance. See (**?**)Appendix A.3]fan2023dpokreinforcementlearningfinetuning for the derivation.

E.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:

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

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

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

Note that implicit in Algorithm 2 is the fact that for minimizing the Lagrangian $\widehat{L}_{\text{AND}}(\theta, \lambda)$ we need samples from the weighted product distribution $q_{\text{AND}}^{(\lambda)}(\cdot)$. We do this using the Annealed MCMC sampling algorithm proposed in (Du et al., 2024).

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

We achieve this by using the surrogate product score (rather than the true score) for computing the point-wise KL needed for the dual updates. The difference between the two is also discussed in (Du et al., 2024).

$$\text{true score:} \quad \nabla \log q_{\text{AND},\, t}^{(\lambda)}(x_t) = \nabla \log \left(\int \sum_i (q_0(x_0))^{\lambda_i} q(x_t|x_0) dx_0 \right) \tag{82}$$

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

```
Input: total diffusion steps $T$, diffusion parameter $\alpha_t$, total dual iterations $H$, number of primal steps per dual update $N$, dual step size $\eta_d$, primal step size $\eta_p$, initial model parameters $\theta(0)$.
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
            $$\theta_{n+1} = \theta_n - \eta_p \cdot \nabla_\theta \widehat{L}_{\text{AND}}(\theta, \lambda^{(n)}) \tag{80}$$
    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 $i = 1, \cdots, m$:
        $$\widetilde{\lambda}_i(h+1) \; = \; \lambda_i(h) \; + \; \eta_d D_{\text{KL}}(p_0(\cdot\,; s_{\theta(h)}) \,\|\, p_0(\cdot\,; s^i)) \tag{81}$$
    $\lambda(h+1) = \text{proj}\left(\widetilde{\lambda}(h+1)\right)$, where proj$(y)$ projects its input onto the simplex $\lambda^T 1 = 1$.
end for
```

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

```
Input: total diffusion steps $T$, diffusion parameter $\alpha_t$, total dual iterations $H$, dual step size $\eta_d$.
Initialize: $\lambda(1) = 1/m$.
for $h = 1, \cdots, H$ do
    Update dual multipliers for $i = 1, \cdots, m$:
        $$\widetilde{\lambda}_i(h+1) \; = \; \lambda_i(h) \; + \; \eta_d D_{\text{KL}}(\widehat{q}_{\text{AND},0}^{(\lambda(h))}(\cdot) \,\|\, p_0(\cdot\,; s^i)) \tag{84}$$
    $\lambda(h+1) = \text{proj}\left(\widetilde{\lambda}(h+1)\right)$, where proj$(y)$ projects its input onto the simplex $\lambda^T 1 = 1$.
end for
```

$$\text{surrogate score:} \quad \nabla \log \widehat{q}^{(\lambda)}_{\text{AND},\, t}(x_t) = \sum_i \lambda_i \nabla \log \left(\int q_0(x_0) q(x_t|x_0) dx_0 \right) \tag{83}$$

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

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

$$= \sum_i \lambda_i \nabla \log p_t(x_t; s^i) \tag{86}$$

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.

F. More Experiments and Experimental Details

F.1. Low-dimensional synthetic experiments

For illustrating the difference between the constrained and unconstrained approach visually, we set up experiments where the generated samples are in $\mathbb{R}^2$. For the score predictor we used the same ResNet architecture as used in (Du et al., 2024).

Product composition (AND). Unlike the image experiments, in this low-dimensional setting we used 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 (Khalafi et al., 2024) 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.

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

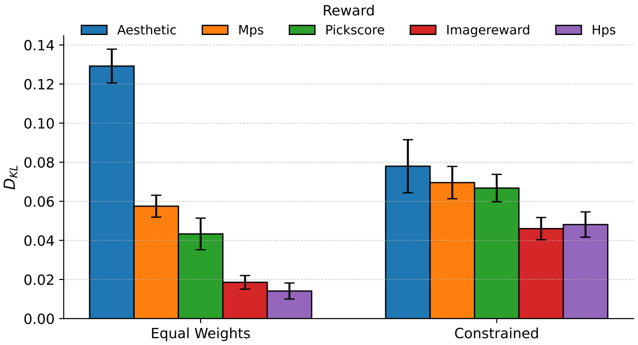

Figure 6. KL divergence for the product composition of 5 adapters pre-trained with different rewards. Error bars denote the standard deviation computed across 8 text prompts each with four samples.

Implementation details and hyperparameters. We finetuned the model using the Alignprop (Prabhudesai et al., 2024a) official implementation [4] 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 E.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% corresponds to the reward obtained by the pre-trained model, and 100% the reward obtained by the model fine-tuned solely on the corresponding reward.

Additional results. As shown in Figure 6, equal weighting leads to disparate KL's 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. Table 4 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.

F.3. 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 (Skreta et al., 2025)) As seen in table 5, our approach retains a clear advantage in both CLIP and BLIP scores. See Table 6 for examples of images generated using each method. Images with the constrained method typically do a better job of representing all concepts.

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

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

F.4. 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 pre-trained 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 pre-trained 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

$$\widetilde{r} \;=\; \frac{r - \widehat{\mu}_{\text{pre}}}{\widehat{\sigma}_{\text{pre}}}, \tag{87}$$

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

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 8.

Additional results. We include images sampled from the constrained model in Figure 9 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) (Sohrabi et al., 2024). Hyperparameters are detailed in Table 10.

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

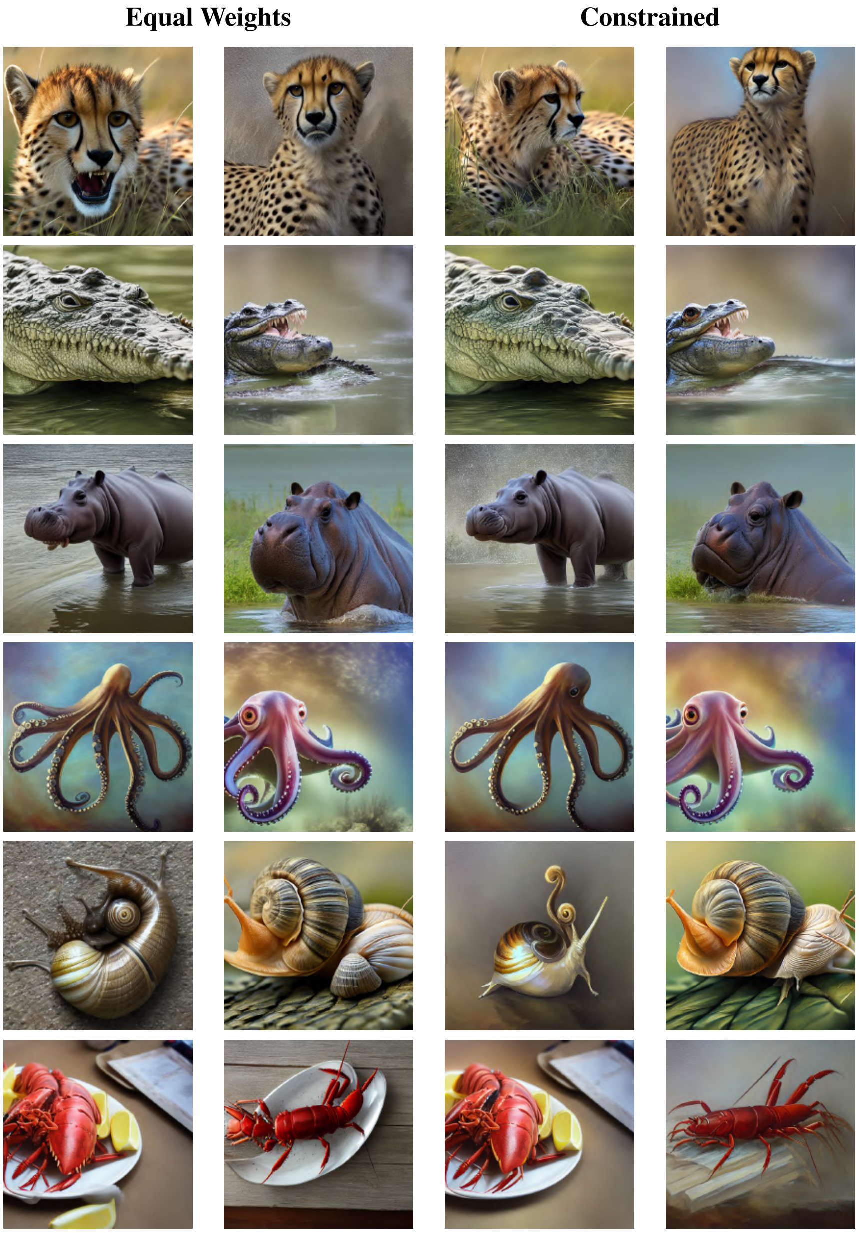

Table 4. Images sampled from the same latents for the product of adapters using the equal weights and when using the proposed KL-constrained reweighting scheme using 5 dual steps.

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

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

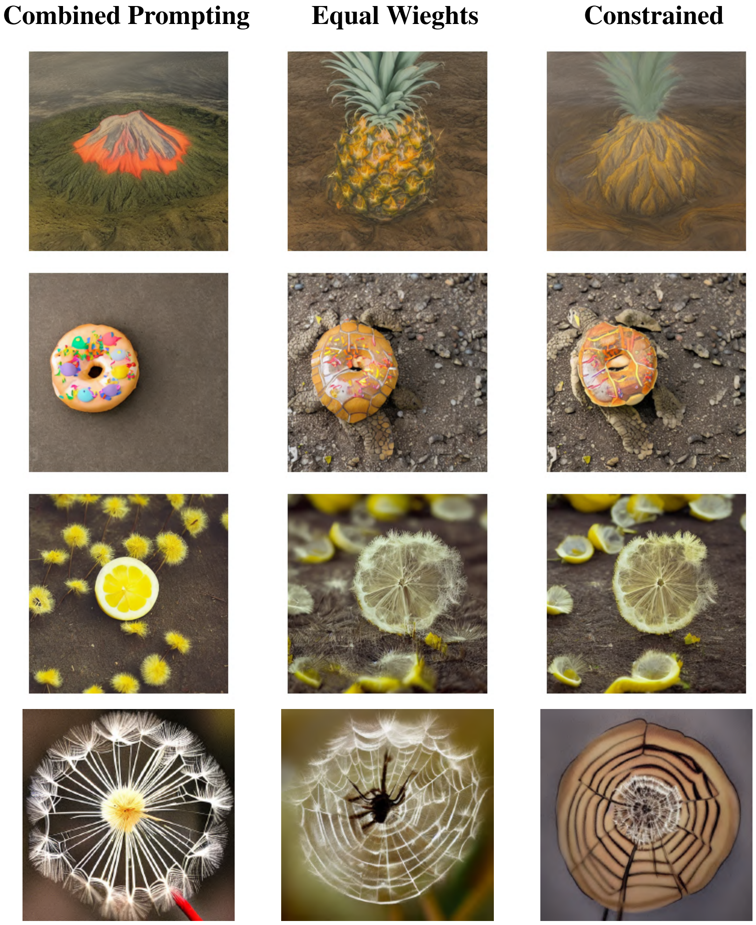

Table 6. Concept composition examples for each method. Prompts used for each row:
Row 1: "a pineapple", "a volcano". **Row 2:** "a donut", "a turtle". **Row 3:** "a lemon", "a dandelion". **Row 4:** "a dandelion", "a spider web", "a cinammon roll".

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

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

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

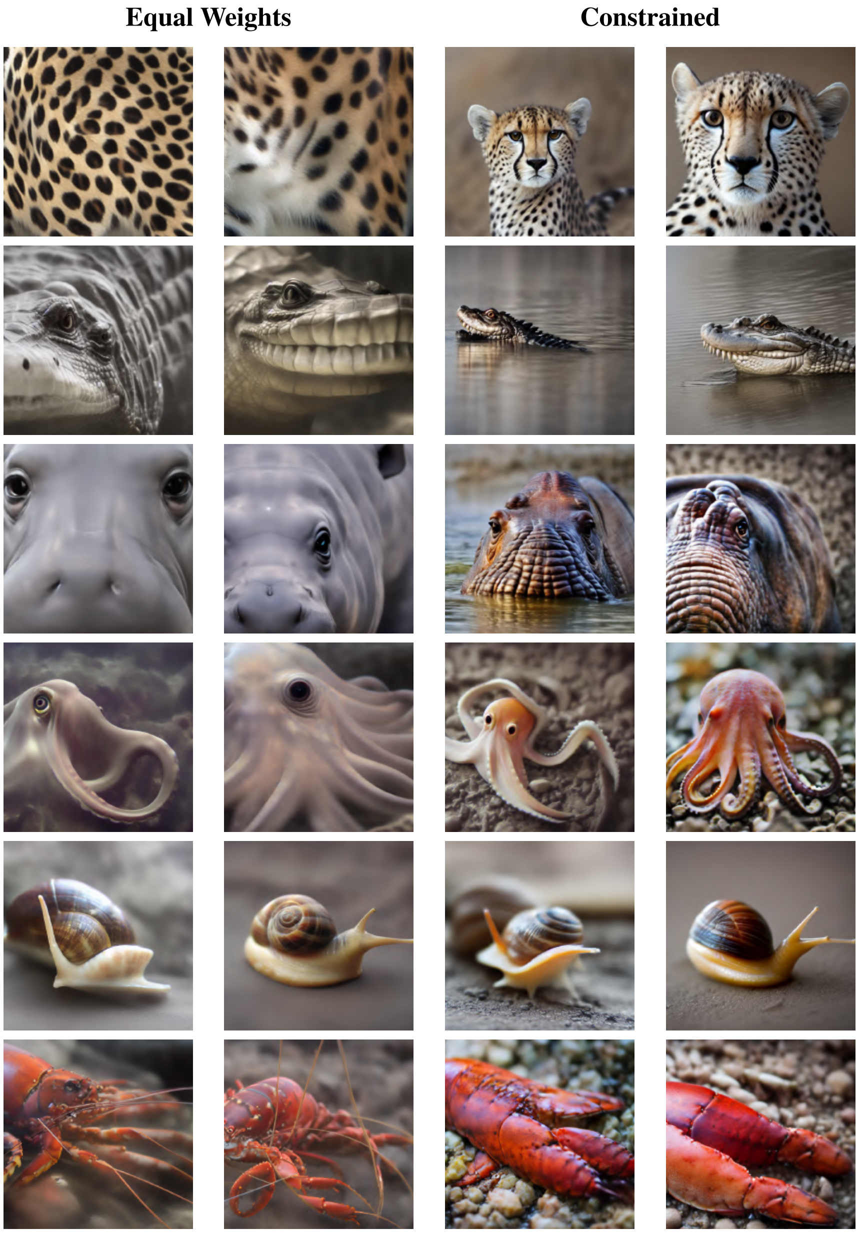

Table 9. Images sampled from models finetuned to maximize MPS (Zhang et al., 2024a), along with sharpness and saturation penalizations. We compare optimizing an equally weighted objective against our constrained approach.

Hyperparameter	**Value**
Base model	`runwayml/stable-diffusion-v1-5`
Sampling steps	15
Dual learning rate	0.05
Batch size (effective)	$4 \times 16 = 64$
Samples per epoch	128
Epochs	25
KL penalty	0.1
LoRA rank	4
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 10. Hyperparameters for reward alignment with multiple rewards. Constraint levels correspond to normalised rewards.

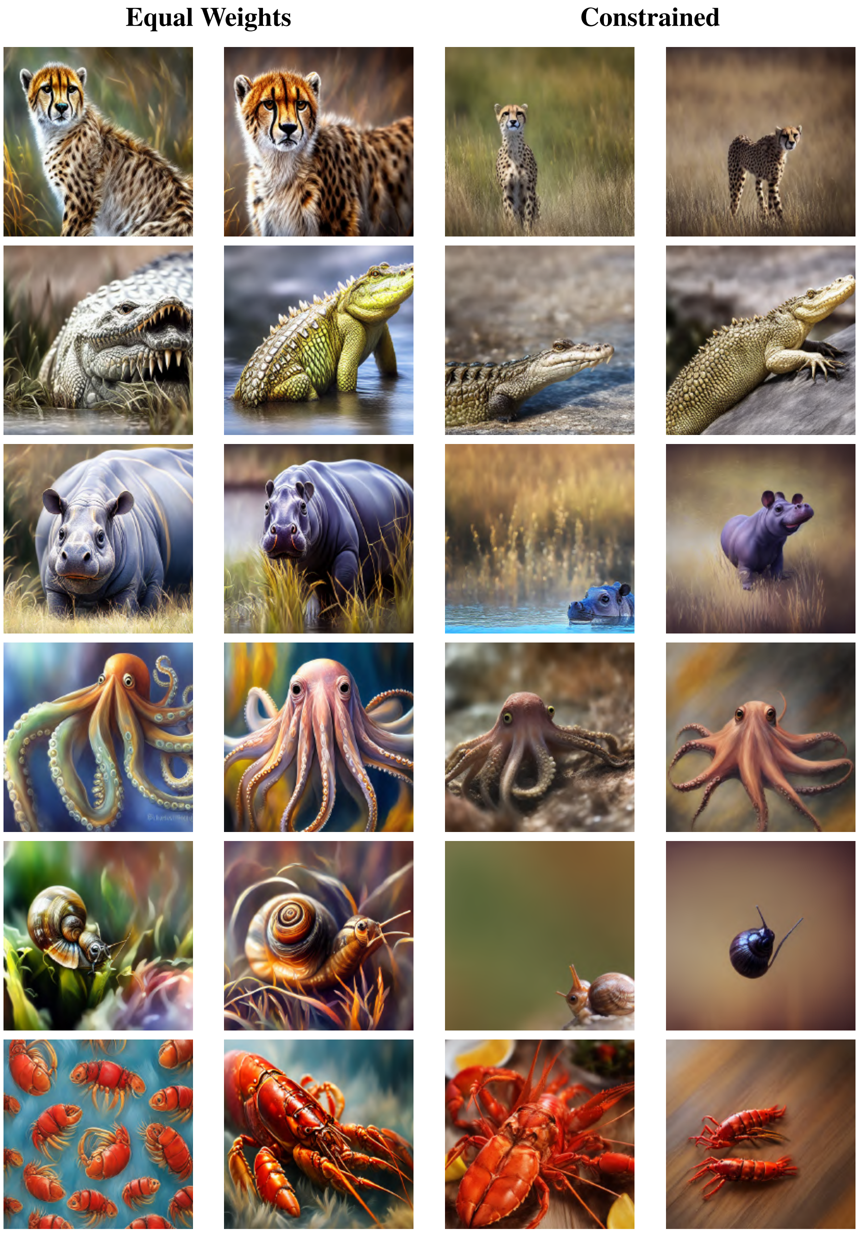

Table 11. Samples from models fine-tuned using multiple rewards with equal weights and with our constrained alignment method.