CHANCE-CONSTRAINED FLOW OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting pre-trained diffusion and flow models to optimize downstream utilities is central to their real-world deployment. Existing steering methods typically optimize average rewards or impose constraints on average properties. However, downstream requirements often concern the probability of generating samples that meet particular criteria, which cannot be represented by average rewards or constraints. To address this gap, we propose **C**hance-**C**onstrained **F**low **O**ptimization (CCFO), a distributional steering framework that maximizes expected reward while explicitly controlling the probability of constraint satisfaction. For such a constrained optimization problem, we theoretically derive a closed-form expression for the optimal dual variable, which allows to directly characterize the optimal distribution. This characterization decomposes CCFO into a scalable two-stage procedure: a lightweight dual variable estimation and a single entropy-regularized fine-tuning step via a specific pseudo-reward. By decoupling dual estimation from model fine-tuning, CCFO achieves a computational cost comparable to that of standard expected-reward fine-tuning methods. Experiments on 2D illustrative examples, high-dimensional text-to-image generation, and protein sequence design establish that CCFO achieves better reward–feasibility trade-offs than state-of-the-art baseline methods.

1 INTRODUCTION

Diffusion and flow models have become powerful tools for learning complex data distributions and generating high-quality samples (Lipman et al., 2022; Liu et al., 2022; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2020). They have been widely adopted across domains such as vision (Song et al., 2020), biology (Chi et al., 2025), and chemistry (Corso et al., 2022; Hoogeboom et al., 2022). In many downstream applications, however, their practical value depends not only on reproducing the training distribution, but also on whether they can be steered toward the task-specific goals. A common approach is to fine-tune the model by maximizing the expected reward of generated samples while penalizing deviations from the pre-trained distribution through a Kullback–Leibler (KL) divergence. This KL-regularized formulation provides a principled foundation for reward-guided adaptation and underlies a range of fine-tuning methods for diffusion and flow models (Domingo-Enrich et al., 2024; Liu et al., 2026; Zheng et al., 2025; Choi et al., 2026). Extending this line, recent works (Khalafi et al., 2026; Gutjahr et al., 2025; Smith et al., 2025) also incorporates expected-reward constraints on the generated distribution.

These expected objectives and constraints primarily focus on the average performance of the generated distribution. Many downstream requirements, however, concern the probability of generating samples that meet a prescribed criterion. For example, in protein design, sequence novelty can be quantified by the mutation distance from a reference sequence (Yang et al., 2026), motivating requirements that a prescribed fraction of generated sequences exceed a desired novelty threshold. Such a requirement concerns how frequently the novelty criterion is satisfied, rather than the average mutation distance. More generally, a favorable average property value can coexist with a substantial probability of violating a desired threshold, since high scores on some samples can compensate for low scores on others. Consequently, optimizing expected rewards or constraining expected property values alone does not, in general, guarantee a prescribed probability of constraint satisfaction.

To address this limitation, we consider the distributional steering problem that maximizes expected reward under KL regularization while imposing a lower bound on the probability of constraint satisfaction. This chance-constrained formulation directly controls the allowable probability of

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

constraint violation of the generated distribution, and enables utility optimization subject to an explicit probabilistic feasibility requirement. We propose **C**hance-**C**onstrained **F**low **O**ptimization (CCFO), an efficient method to steer generative models under chance constraints. Our key theoretical result shows that the resulting distributional optimization problem admits an exact reduction to a one-dimensional convex optimization over a nonnegative dual multiplier. Moreover, the optimal multiplier has a closed-form expression that depends only on expectations under the pre-trained distribution and can therefore be estimated efficiently from pre-trained samples. Given this multiplier, the optimal primal distribution takes a Boltzmann form and coincides with the solution of a standard KL-regularized fine-tuning problem under a modified pseudo-reward. As a result, once a single scalar multiplier is estimated, the original chance-constrained problem can be solved with one call to an off-the-shelf entropy-regularized fine-tuning method. CCFO therefore combines explicit control of probabilistic feasibility with the computational simplicity of standard reward-guided adaptation.

Our main contribution are threefold:

- **Chance constrained distributional formulation and optimal characterization.** We formulate KL-regularized reward maximization under a prescribed minimum probability of constraint satisfaction. We characterize the optimal distribution and derive a closed-form expression for the optimal dual multiplier.
- **Efficient fine-tuning and sensitivity analysis.** We translate this characterization into a two-stage algorithm consisting of pre-trained-sample-based dual estimation and a single fine-tuning run. We further quantify how multiplier estimation errors and imperfect constraint evaluations perturb the induced target distribution.
- **Empirical evaluation across generative domains.** We evaluate CCFO on 2D illustrative examples, high-dimensional text-to-image generation, and protein sequence design. Across these settings, CCFO achieves improved reward–feasibility trade-offs compared with baseline methods.

2 Background

2.1 Generative Flow and Diffusion Models

Flow-based generative models aim to approximately sample data points from a data distribution p_{data} (Lipman et al., 2024). A convenient perspective is to construct a continuous-time transport that maps samples from a simple source distribution p_0 (e.g., a standard Gaussian) into samples from p_{data}.

A generative flow is characterized as a time-dependent map $\psi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d,\ (t,x) \mapsto \psi_t(x)$, which transports an initial random variable $X_0 \sim p_0$ along the trajectory $X_t = \psi_t(X_0),\ t \in [0,1]$, so that ideally $X_1 \sim p_{\text{data}}$. A common parametrization defines ψ_t through a time-dependent velocity field $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, via the flow ODE $\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)), \qquad \psi_0(x) = x$. If $X_0 \sim p_0$ and $X_t = \psi_t(X_0)$, then the model induces a path of marginal distributions $\{p_t\}_{t\in[0,1]}$, with terminal distribution p_1. The pair $(u, \{p_t\}_{t\in[0,1]})$ is consistent with the flow dynamics if the induced density path satisfies the continuity equation $\partial_t p_t(x) + \nabla \cdot \big(p_t(x)\, u_t(x)\big) = 0$, which expresses conservation of probability mass under the deterministic transport. We write $p^u := \{p_t^u\}_{t\in[0,1]}$ for the marginal density path induced by a velocity field u.

Standard flow-matching methods learn a parameterized velocity field u^π whose induced terminal distribution $p_1^{u^\pi}$ approximates p_{data}. After training, sampling proceeds by drawing $X_0 \sim p_0$ and numerically integrating the ODE to obtain $X_1 \sim p_1^{u^\pi}$. Diffusion models admit an equivalent ODE formulation with the same marginal densities, so the notation and results of this paper apply to both flow and diffusion models (Lipman et al., 2024).

2.2 Entropy-Regularized Fine-Tuning

Consider a generative flow model be parameterized by π. We denote its induced sampling distribution $p_1^{u^\pi}$ by p^π for simplicity of notation. Assume that the model is pre-trained, and can generate samples from the distribution p^{pre}. Given a reward function $r : \mathcal{X} \to \mathbb{R}$ that scores samples, the goal of fine-tuning is to learn a new sampling distribution that increases the reward while remaining close

to the pre-trained sampling distribution p^{pre}. The standard KL-regularized fine-tuning problem is

$$\max_{p_\pi \in \mathcal{P}(\mathcal{X})} \mathbb{E}_{X\sim p_\pi}[r(X)] - \alpha D_{\mathrm{KL}}(p^\pi \| p^{\mathrm{pre}}). \tag{1}$$

which optimizes the expectation of the reward distribution, penalized by a KL divergence term multiplied by α. The optimal distribution of (1) admits the closed-form Boltzmann distribution $p^\star(x) \propto p^{\mathrm{pre}}(x)\exp(r(x)/\alpha)$. This can be solved by stochastic optimal control methods (Domingo-Enrich et al., 2024; Uehara et al., 2024) and reinforcement learning methods (Zheng et al., 2025; Liu et al., 2026; Choi et al., 2026; Bergmeister et al., 2026).

3 Chance Constrained Flow Optimization

3.1 Problem Formulation

Starting from a pre-trained model with terminal distribution p^{pre}, we seek a fine-tuned distribution p^π that maximizes reward while ensuring that the generated sample is feasible with probability at least $1-\beta$. We consider the following generative optimization problem

$$\max_{p^\pi} \mathbb{E}_{p^\pi}[r(X)] - \alpha D_{\mathrm{KL}}(p^\pi || p^{\mathrm{pre}}), \qquad \text{s.t.} \quad \mathbb{P}\left[g(X) \le 0, X \sim p^\pi\right] \ge 1-\beta \tag{2}$$

Using the indicator $s(x) = \mathbf{1}\{g(x) \le 0\}$, we may equivalently rewrite (2) as

$$\max_{p^\pi} \mathbb{E}_{X\sim p^\pi}[r(X)] - \alpha D_{\mathrm{KL}}(p^\pi \| p^{\mathrm{pre}}) \qquad \text{s.t.} \quad \mathbb{E}_{X\sim p^\pi}[s(X)] \ge 1-\beta. \tag{3}$$

The constraint is thus an expectation of a binary indicator function. Unlike a constraint on an average property value, it directly lower-bounds the probability mass assigned to the feasible set. As we show next, this binary structure is what makes the problem tractable in closed form.

3.2 Reformulation

Since the constraint in (3) is affine in the optimizing distribution, we introduce a Lagrange multiplier $\lambda \ge 0$ and write the Lagrangian

$$\mathcal{L}(p,\lambda) = \mathbb{E}_p[r(X) + \lambda s(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) - \lambda(1-\beta).$$

For any fixed $\lambda \ge 0$, maximizing $\mathcal{L}(p,\lambda)$ over p is a standard KL-regularized fine-tuning problem with modified reward $r(x)+\lambda s(x)$. The following result shows that the resulting dual problem is a one-dimensional convex program whose solution is available in closed form.

Theorem 1. *Let $\alpha > 0$, $\beta \in (0,1)$, and $\mathbb{E}_{p^{\mathrm{pre}}}\left[e^{r(X)/\alpha}\right] < \infty$. Define $A := \mathbb{E}_{p^{\mathrm{pre}}}\left[e^{r(X)/\alpha}\mathbf{1}\{g(X)>0\}\right]$ and $B := \mathbb{E}_{p^{\mathrm{pre}}}\left[e^{r(X)/\alpha}\mathbf{1}\{g(X)\le 0\}\right] > 0$. Then, the problem (2) is equivalent to*

$$\min_{\lambda\ge 0} G(\lambda) := \alpha \log \mathbb{E}_{p^{\mathrm{pre}}}\left[\exp\left(\frac{r(X)+\lambda s(X)}{\alpha}\right)\right] - \lambda(1-\beta), \tag{4}$$

whose optimal solution is $\lambda^* = \alpha\left[\log\frac{(1-\beta)A}{\beta B}\right]_+$, where $[u]_+ := \max\{u,0\}$. The optimal distribution of (2) satisfies $p^*(x) \propto p^{\mathrm{pre}}(x)\exp\left(\frac{r(x)+\lambda^* s(x)}{\alpha}\right)$.

Theorem 1 admits a simple interpretation. The quantity A is the reward-weighted mass that the pre-trained model assigns to the infeasible set, and B is the corresponding mass on the feasible set. If the feasible region already carries sufficient reward-weighted mass relative to the target chance level, that is $B/(A+B) \ge 1-\beta$, then $\lambda^\star = 0$ and the chance constraint is inactive. Otherwise $\lambda^\star > 0$ adds an explicit bonus to feasible samples, tilting the Boltzmann distribution toward the feasible set. The practical consequence is that $\lambda^\star$ depends only on p^{pre}, r, g and α, and not on the fine-tuned model. It can therefore be computed *before* any fine-tuning, rather than through alternating primal–dual updates.

3.3 Algorithm

Theorem 1 naturally yields the two-stage procedure: estimate the multiplier from pre-trained samples, then run a single KL-regularized fine-tuning with a modified reward.

Stage 1: Multiplier estimation. Draw $x_1, \dots, x_N \overset{\text{i.i.d.}}{\sim} p^{\text{pre}}$ and evaluate $r(x_i)$ and $\hat{s}(x_i)$. Compute

$$\widehat{A} = \frac{1}{N}\sum_{i=1}^{N} e^{r(x_i)/\alpha}\big(1 - s(x_i)\big),$$
$$\widehat{B} = \frac{1}{N}\sum_{i=1}^{N} e^{r(x_i)/\alpha} s(x_i). \tag{5}$$

If $\widehat{A} = 0 < \widehat{B}$, set $\widehat{\lambda} = 0$. If $\widehat{B} = 0$, there are no feasible samples collected, and this calls for a larger batch size and more sampling. For $\widehat{A}, \widehat{B} > 0$, the estimated multiplier is $\widehat{\lambda} = \alpha\left[\log\frac{(1-\beta)\widehat{A}}{\beta\widehat{B}}\right]_+$. The multiplier is positive when $\widehat{B}/(\widehat{A}+\widehat{B}) < 1-\beta$, that is, when the estimated feasibility of the nominal reward-tilted distribution falls below the prescribed target. Its magnitude is precisely calculated based on the reward performance of feasible and infeasible samples.

Algorithm 1: Chance Constrained Flow Optimization

Input: Pre-trained model u^{pre}, reward $r(x)$.
// Stage 1: Multiplier estimation
Generate N samples $x_i \sim P^{\text{pre}}$
Evaluate the constraint satisfaction at x_i and obtain a noisy feedback $\mathbf{1}\{g(x_i) \le 0\}$
Compute $\hat{A}, \hat{B}$ by (5)
Compute $\hat{\lambda} = \alpha\left[\log\frac{(1-\beta)\hat{A}}{\beta\hat{B}}\right]_+$
// Stage 2: Fine-tuning with a feasibility bonus
Set $\widetilde{r}(x) = r(x) + \widehat{\lambda}s(x)$.
$u^* \leftarrow \text{FineTune}(u^{\text{pre}}, \widetilde{r})$, solving (6).
Output u^*

Stage 2: Fine-tuning with a feasibility bonus. Once the multiplier is obtained, CCFO constructs a pseudo-reward $r(X) + \widehat{\lambda}s(X)$, and uses a compatible fine-tuning solver to solve

$$\max_{p^\pi} \mathbb{E}_{p^\pi}\Big[r(X) + \widehat{\lambda}s(X)\Big] - \alpha D_{\text{KL}}(p^\pi \| p^{\text{pre}})\,. \tag{6}$$

The pseudo-reward retains the original reward while assigning a constant bonus to feasible samples. The cost of CCFO is that of one unconstrained reward fine-tuning run, plus the sampling cost of Stage 1.

Remarks. By exploiting the binary structure of the chance constraint, CCFO decouples multiplier estimation from model adaptation, and so avoids alternating multiplier–model updates. Two practical points are worth noting. First, the guarantee of Theorem 1 concerns the ideal target $p^\star$. In high-dimensional settings the solver may not realize this target, in which case $\widehat{\lambda}$ can be refined periodically using feasibility estimates from the current model. Second, the pseudo-reward is non-differentiable due to the indicator function, so the solver must accept non-differentiable rewards, as do reinforcement-learning methods such as DiffusionNFT (Zheng et al., 2025). Alternatively, the indicator function can be replaced by a smooth surrogate such as a sigmoid, which enables gradient-based solvers such as Adjoint Matching (Domingo-Enrich et al., 2024).

4 Theoretical Analysis

This section analyzes CCFO along three axes: the structure of the optimal distribution, the sensitivity of the ideal target to multiplier estimation errors, and to an approximate constraint oracle.

Throughout we assume $\alpha > 0$, $\beta \in (0,1)$, and $0 < A, B < \infty$, with A and B as defined in Theorem 1. Let $F = \{x : s(x) = 1\}$ and $F^c = \{x : s(x) = 0\}$ denote the feasible and infeasible sets, and define

$$\frac{d\mu}{dp^{\text{pre}}}(x) = \frac{e^{r(x)/\alpha}}{A+B}, \qquad h = \mu(F) = \frac{B}{A+B}. \tag{7}$$

Here $\mu = p_0$ is the nominal reward-tilted distribution, the ideal target of *unconstrained* reward fine-tuning, and h is its feasibility probability. For $\lambda \ge 0$, write

$$\frac{dp_\lambda}{dp^{\text{pre}}}(x) = \frac{\exp((r(x) + \lambda s(x))/\alpha)}{A + e^{\lambda/\alpha}B}. \tag{8}$$

Thus, the optimal distribution characterized in Section 3.2 is $p^\star = p_{\lambda^\star}$.

4.1 Optimal distribution structure

Because s takes only two values, the target family is a one-parameter mixture of two *fixed* conditional distributions.

Proposition 1 (Mixture structure). *For $\lambda \geq 0$, we have*

$$p_\lambda = (1 - h_\lambda)\mu(\cdot \mid F^c) + h_\lambda \mu(\cdot \mid F), \quad h_\lambda = \frac{e^{\lambda/\alpha} h}{1 - h + e^{\lambda/\alpha} h}. \tag{9}$$

Moreover, the constrained optimum $p^\star = p_{\lambda^\star}$ satisfies $p^\star(F) = \max\{h, 1 - \beta\}$.

The two mixture components do not depend on λ. Varying the feasibility bonus therefore changes only how often feasible samples are generated. In this sense CCFO enforces feasibility by reallocating probability mass between F and F^c, while leaving the reward-driven preferences on each side untouched. The second claim makes the role of β explicit. If the nominal target already meets the requirement, that is $h \geq 1 - \beta$, then $\lambda^\star = 0$ and $p^\star = \mu$. Otherwise the optimal bonus raises the feasible mass from h to exactly $1 - \beta$. This contrasts with an expected-value constraint $\mathbb{E}[g(X)] \leq c$, where the induced feasibility probability is an implicit function of c, r and p^{pre}. We compare CCFO with gated reward fine-tuning, another common heuristic, in Appendix B.

We note that Proposition 1 describes the *ideal* target $p^\star$. An approximate solver need not reproduce this structure, and the subsection below quantify two sources of errors.

4.2 Sensitivity analysis

Approximate dual multiplier. Stage 1 replaces A and B by empirical estimates, so $\widehat{\lambda} \neq \lambda^\star$ in general. The next result shows the ideal target is Lipschitz in the multiplier.

Proposition 2 (Sensitivity to the multiplier). *Let $\alpha > 0$, $\beta \in (0, 1)$, $\lambda^\star = \alpha \left[\log \frac{(1-\beta)A}{\beta B} \right]_+$, and $\widehat{\lambda} = \alpha \left[\log \frac{(1-\beta)\widehat{A}}{\beta \widehat{B}} \right]_+$. Then, we have $\|p_{\widehat{\lambda}} - p_{\lambda^\star}\|_{\mathrm{TV}} \leq \frac{|\widehat{\lambda} - \lambda^\star|}{4\alpha} \leq \frac{1}{4} \left| \log \frac{\widehat{A}}{A} - \log \frac{\widehat{B}}{B} \right|$.*

Two consequences are worth noting. First, only the relative errors of $\widehat{A}$ and $\widehat{B}$ matter, and a common multiplicative bias in both estimates cancels exactly. Second, since total variation bounds the difference of any event probability, taking the event to be F gives a direct guarantee on the realized feasibility of the ideal target, $p_{\widehat{\lambda}}(F) \geq (1 - \beta) - \left\| p_{\widehat{\lambda}} - p_{\lambda^\star} \right\|_{\mathrm{TV}}$. Under a boundedness condition on $e^{r(X)/\alpha}$, the right-hand side of Proposition 2 is $O(N^{-1/2})$ with high probability, where N is the Stage 1 sample size. Appendix A.4 states the assumptions, the explicit constants, and the derivation.

Approximate constraint evaluation. In practice feasibility is often assessed by a surrogate oracle: a learned classifier, a predicted structure, or the smooth relaxation of s discussed in Section 3.3. Let $\widehat{g}$ be a measurable approximation of g and $\widehat{s}(x) = \mathbf{1}\{\widehat{g}(x) \leq 0\}$. To isolate this effect from multiplier error, we fix λ and compare p_λ with the target induced by the surrogate $\frac{d\widehat{p}_\lambda}{d\mu}(x) = \frac{e^{\lambda \widehat{s}(x)/\alpha}}{\mathbb{E}_\mu \left[e^{\lambda \widehat{s}(X)/\alpha} \right]}$..

Proposition 3 (Sensitivity to approximate constraint functions). *Define the reward-weighted disagreement $\delta_g = \mathbb{E}_\mu[|s(X) - \hat{s}(X)|] = \frac{\mathbb{E}_{p^{\mathrm{pre}}}[e^{r(X)/\alpha} |s(X) - \hat{s}(X)|]}{A + B}$. Then, for every $\lambda \geq 0$, $\|\widehat{p}_\lambda - p_\lambda\|_{\mathrm{TV}} \leq \left(e^{\lambda/\alpha} - 1\right) \delta_g$.*

The error is controlled by the oracle's disagreement with the truth measured under μ, not under p^{pre}. Mistakes on high-reward samples are therefore weighted more heavily, which is intuitive: those are the samples the fine-tuned model is most likely to produce. The factor $e^{\lambda/\alpha} - 1$ vanishes at $\lambda = 0$, where both targets equal μ. A demanding chance level thus places a correspondingly higher demand on oracle accuracy.

5 Experiments

We evaluate CCFO on the task of adapting pre-trained generative models to maximize reward under a chance constraint. We present two types of experiments: (i) Illustrative and visually

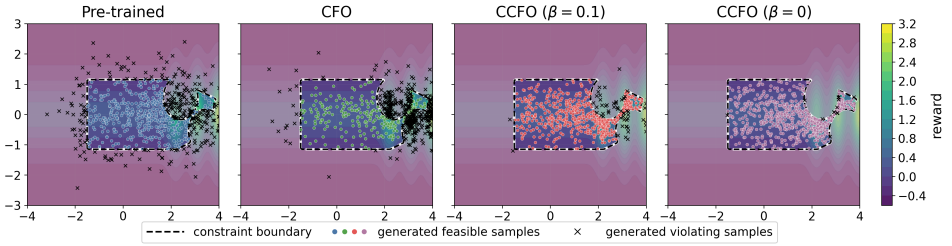


Figure 1: An illustrative 2D "striped tunnel"example. CFO satisfies an expected constraint but retains substantial infeasible mass; CCFO explicitly controls feasibility through β.

interpretable settings to provide insights about different methods, and (ii) High-dimensional real-world applications, including (a) text-to-image generation, and (b) biochemical design tasks over protein sequences. Additional experimental details are provided in Appendix C.

Baselines. **Reward FT** maximizes reward with no constraint. **Gated Reward FT** is a heuristic that replaces r with the gated reward $r(x)\mathbf{1}\{g(x) \leq 0\}$; Appendix B compares it with CCFO at the level of ideal targets. **CFO** (Gutjahr et al., 2025) maximizes expected reward under an expected-value constraint $\mathbb{E}[g(X)] \leq c$. **CGM** (Smith et al., 2025) handles distributional *equality* constraints without reward maximization and is therefore not directly applicable to our setting. For comparison, we augment its original algorithm with reward maximization and report the adapted **Augmented CGM-relax**.

5.1 2D Examples

We first use a two-dimensional illustrative example to illustrate how the constraint formulation affects the generated distribution. We take a four-component Gaussian mixture as the pre-trained distribution and optimize probability masses on a uniform grid, using the same KL regularization coefficient for all methods. The feasible set, defined by $g(x) \leq 0$, is a triangular corridor intersected with a horizontal band, with circular exclusion zones removed. The reward increases along vertical stripes toward the right, including regions that overlap the exclusions. This creates a trade-off between expected reward and feasibility.

Fig. 1 compares the pre-trained distribution, unconstrained reward fine-tuning, CFO with $\mathbb{E}[g(X)] \leq 0$, and CCFO with budgets $\beta = 0.1$ and $\beta = 0$. Although CFO satisfies its expected constraint exactly, only 38.3% of its probability mass is feasible: negative constraint values at feasible points offset positive values at infeasible ones, so frequent violations coexist with compliance in expectation. CCFO instead controls this frequency directly, attaining 90% feasibility at $\beta = 0.1$ and confining the distribution to the feasible set at $\beta = 0$.

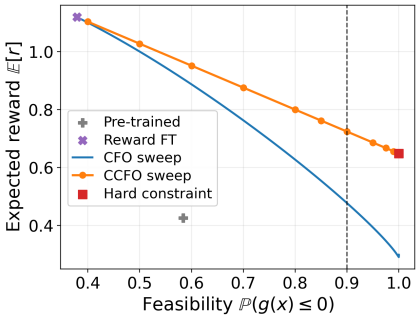


Figure 2: Reward–feasibility trade-offs in the 2D example.

Sweeping CFO's expected-constraint threshold and CCFO's feasibility target also reveals a better reward–feasibility trade-off for CCFO, as shown in Fig. 2. At matched 90% feasibility, CCFO achieves an expected reward of 0.733, compared with 0.480 for CFO. CCFO adjusts the mass assigned to the feasible set while preserving the reward-tilted conditional distributions; tightening CFO's threshold instead favors more negative constraint values, which can sacrifice reward within the feasible set.

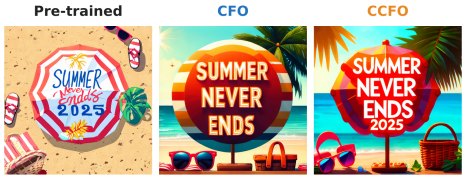
5.2 Text-to-Image Generation

To assess the efficacy of our CCFO in real-world applications, we apply it to high-dimensional text-to-image generation using Stable-Diffusion 3.5 (Esser et al., 2024). We optimize the optical character recognition (OCR) score, which measures the accuracy of visual text rendering, while imposing a chance constraint on the Aesthetic score (Schuhmann, 2022). Writing $a(x)$ for the

Table 1: Quantitative comparison on text-to-image generation. Each run is evaluated over 4096 generated samples averaged over 3 seeds.

Method	Mean OCR ↑ (all)	Feasibility (%) ↑ $\mathbb{P}$(aesthetic ≥ 5.3)	Mean OCR ↑ (given feasible)	ImageReward ↑	HPS ↑	ClipScore ↑
Pre-trained	$0.114_{\pm 0.010}$	$54.03_{\pm 0.34}$	$0.091_{\pm 0.004}$	$-0.342_{\pm 0.012}$	$0.203_{\pm 0.001}$	$0.265_{\pm 0.001}$
Gated REWARD FT	$0.617_{\pm 0.006}$	$\mathbf{93.20}_{\pm 0.69}$	$0.621_{\pm 0.007}$	$0.293_{\pm 0.008}$	$0.239_{\pm 0.001}$	$0.282_{\pm 0.001}$
AUGMENTED CGM-RELAX	$0.568_{\pm 0.005}$	$81.75_{\pm 0.61}$	$0.517_{\pm 0.007}$	$0.293_{\pm 0.019}$	$0.243_{\pm 0.001}$	$0.278_{\pm 0.001}$
CFO	$0.782_{\pm 0.009}$	$89.26_{\pm 1.34}$	$0.789_{\pm 0.007}$	$0.418_{\pm 0.015}$	$0.242_{\pm 0.006}$	$\mathbf{0.285}_{\pm 0.007}$
CCFO	$\mathbf{0.789}_{\pm 0.009}$	$90.67_{\pm 0.88}$	$\mathbf{0.795}_{\pm 0.003}$	$\mathbf{0.431}_{\pm 0.016}$	$\mathbf{0.244}_{\pm 0.001}$	$0.278_{\pm 0.001}$

Aesthetic score, we set $g(x) = \tau - a(x)$ with $\tau = 5.3$, so a sample is feasible when $a(x) \geq \tau$; the threshold is selected as the median Aesthetic score of pre-trained samples (5.3). This is a setting where the two objectives genuinely conflict, since sharpening text rendering tends to degrade visual quality, as shown in (Jin et al., 2025). We adopt DiffusionNFT (Zheng et al., 2025) as the solver. All fine-tuning methods use the same computational budget, and are evaluated based on 4096 generated images over three seeds. Beyond OCR and feasibility, we report ImageReward (Xu et al., 2023) for human preference, ClipScore (Hessel et al., 2021) for text-image consistency, and HPSv2 (Wu et al., 2023) for domain generalization.



Prompt: A sunny tropical beach scene with a striped umbrella displaying the phrase "Summer Never Ends 2025", surrounded by sandals, sunglasses, and a picnic basket.

Figure 3: Qualitative comparison of generated samples. Qualitative comparison of generated samples. CCFO renders visual text more accurately than CFO.

Fig. 4 compares the empirical reward–feasibility trade-offs. Unconstrained reward fine-tuning (REWARD FT) achieves a high OCR score but drives aesthetic feasibility to be around 0, confirming these two metrics conflict with each other. Among the constrained methods, CCFO dominates CFO: at comparable probabilities of satisfying the Aesthetic constraint, CCFO renders text more accurately. Fig. 3 illustrates this for a prompt requesting the phrase "Summer Never Ends 2025" on a beach umbrella, where CCFO renders the full phrase and CFO drops "2025". Table 1 reports the quantitative comparison. Gated REWARD FT attains the highest feasibility. The price is that it discards the information carried by high-reward samples that violate the constraint, and its OCR score stalls at 0.617. CCFO exploits those samples instead, reaching the highest OCR score of 0.789 while still holding feasibility at 90.67%. The same ordering holds for OCR restricted to feasible samples. Besides, CCFO also attains the best ImageReward and HPSv2, indicating that the feasibility bonus does not come at the cost of overall perceptual quality.

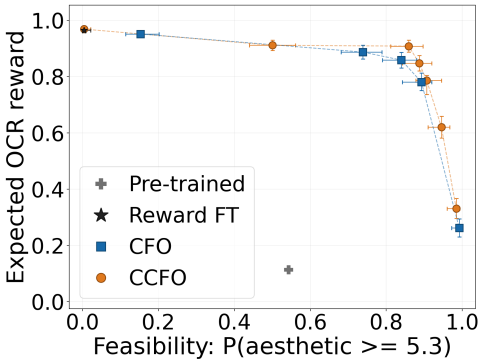


Figure 4: Reward–feasibility trade-offs in text-to-image generation. CCFO achieves a better Pareto frontier than CFO.

5.3 PROTEIN DESIGN

We further evaluate CCFO on protein sequence generation using the continuous ESM diffusion model from SGPO (Yang et al., 2026). We consider the TrpB enzyme dataset (Johnston et al., 2024), in which each protein has a length of 389 and 15 residues are designable. The task is to maximize the protein fitness, which measures the catalytic activity of TrpB for its native reaction. We require the generated sequences to be sufficiently novel relative to the wild type (WT), and declare a sample feasible when its mutation distance from WT satisfies $d \geq 8$. Our goal is therefore to improve fitness while maintaining a high probability of generating sufficiently mutated sequences.

We evaluate each method using 4096 generated samples over three independent runs. Besides mean fitness (over all and over feasible samples) and feasibility, we report sequence diversity as the per-site

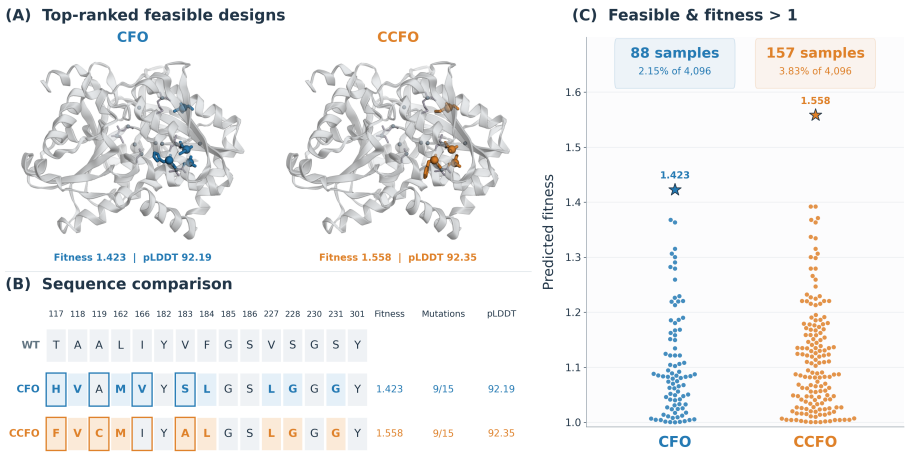


Figure 5: Qualitative comparison of CFO and CCFO on TrpB protein generation. **A**: ESMFold structures of the top-ranked feasible designs, with mutated residues highlighted. **B**: Sequence comparison at the 15 design positions relative to the wild-type sequence. **C**: Generated feasible sequences with fitness above that of WT ($f > 1$). CCFO generates more feasible and high-fitness candidates. CCFO discovers a top protein sequence that has higher fitness and pLDDT than CFO.

entropy over the 15 design positions, and the number of *distinct* designs that are both feasible and high-fitness ($f > 1$), which is the quantity a design campaign ultimately cares about. We also compute pLDDT with ESMFold (Lin et al., 2023) as a proxy for structural confidence. As anticipated by the remark in Section 3.3, the indicator provides a sparse training signal in this discrete setting, so we use the optional refinement there: $\widehat{\lambda}$ is initialized from the closed-form estimate and then adjusted periodically from the current model's feasibility, and the indicator is partially smoothed at the highest feasibility targets. Appendix C.3 gives the details.

Fig. 6 compares the empirical fitness–feasibility frontiers. As the requirement is tightened, both methods shift probability mass onto sequences with $d \geq 8$ at the cost of mean fitness, but CCFO remains above CFO throughout the evaluated range. Table 2 provides the quantitative comparison. CCFO achieves the highest mean fitness 0.392 and simultaneously the highest feasibility 82.30%. It is also best on mean fitness among feasible samples. The diversity of CCFO is also comparable to that of the pre-trained model. The practically decisive column is the last one, where CCFO yields 142 distinct feasible and high-fitness designs, against 94 for the strongest baseline. Fig. 5 makes this concrete for one run: 157 CCFO samples are both feasible and above WT fitness, compared with 88 for CFO, and CCFO's top-ranked feasible design has the higher fitness and higher ESMFold confidence. Together, these results demonstrate CCFO's ability to discover feasible and high-fitness protein sequences.

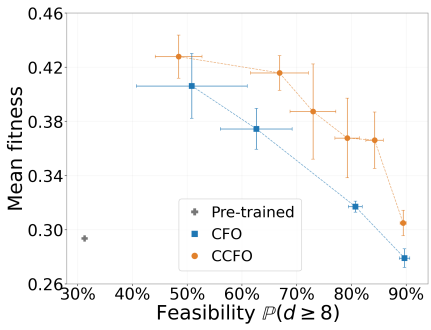


Figure 6: Fitness–feasibility trade-offs on TrpB protein generation. CCFO maintains higher mean fitness than CFO at comparable feasibility.

6 Related Works

Fine-tuning of diffusion and flow models. Reward-guided adaptation of pre-trained diffusion and flow models can be solved by stochastic optimal control methods (Tang & Zhou, 2024; Uehara et al., 2024; Berner et al., 2022) or reinforcement learning approaches (Liu et al., 2026; Zheng et al., 2025). These solvers have been adopted as subroutines to tackle fine-tuning problems beyond expected-reward maximization, including manifold exploration (De Santi et al., 2025b), model merging (De Santi et al., 2026) and optimization of distributional objectives (Wang et al., 2026; De Santi et al., 2025a). However, most existing methods optimize a scalar reward or a fixed weighted

Table 2: Quantitative comparison on TrpB protein generation. Feasibility requires a mutation distance of at least 8 from the wild-type protein. Here, "hit" denotes generated samples with predicted fitness greater than that of the wild type ($f > 1$). The last column reports the number of distinct feasible and high-fitness designs among 4096 samples.

Method	Mean fitness ↑ (all)	Feasibility ↑ (%)	Mean fitness ↑ (given feasible)	pLDDT ↑	Diversity ↑ (all; bits/site)	Diversity ↑ (given feasible; bits/site)	Distinct designs ↑ (feasible & hit)
Pre-trained	$0.318_{\pm 0.005}$	$34.64_{\pm 0.75}$	$0.175_{\pm 0.006}$	$\mathbf{92.46}_{\pm 0.04}$	$1.656_{\pm 0.010}$	$\mathbf{2.024}_{\pm 0.013}$	$12_{\pm 2.3}$
Gated REWARD FT	$0.369_{\pm 0.005}$	$44.15_{\pm 0.61}$	$0.345_{\pm 0.011}$	$92.37_{\pm 0.04}$	$1.390_{\pm 0.008}$	$1.727_{\pm 0.015}$	$46_{\pm 4.4}$
AUGMENTED CGM-RELAX	$0.358_{\pm 0.007}$	$74.61_{\pm 0.69}$	$0.348_{\pm 0.006}$	$92.33_{\pm 0.05}$	$1.688_{\pm 0.008}$	$1.752_{\pm 0.008}$	$94_{\pm 10.2}$
CFO	$0.331_{\pm 0.005}$	$79.57_{\pm 0.65}$	$0.312_{\pm 0.005}$	$92.41_{\pm 0.04}$	$\mathbf{1.763}_{\pm 0.007}$	$1.802_{\pm 0.007}$	$84_{\pm 7.8}$
CCFO	$\mathbf{0.392}_{\pm 0.005}$	$\mathbf{82.30}_{\pm 0.60}$	$\mathbf{0.384}_{\pm 0.006}$	$92.34_{\pm 0.05}$	$1.657_{\pm 0.007}$	$1.703_{\pm 0.007}$	$\mathbf{142}_{\pm 16.3}$

combination of rewards, and therefore do not explicitly guarantee that the fine-tuned distribution satisfies feasibility constraints. This limitation is especially pronounced when the desired properties conflict with one another. CCFO addresses this gap by maximizing rewards while explicitly controlling the feasibility of generated samples. Moreover, our reformulation of the chance constrained problem is agnostic to the underlying data modality (continuous, discrete, or mixed), so CCFO can be applied to any data type.

Constrained sampling. Our work is related to the constrained sampling methods (Zhang et al., 2026; Xiao et al., 2025; Liang et al., 2025), which enforce constraints at inference time while keeping the pre-trained model fixed. For example, Liang et al. (2025) use chance constraints to construct a training-free projection rule for inference-time sampling. Despite the relevant terminology, CCFO addresses a different problem: rather than modifying the sampling trajectory, it fine-tunes the generative model so that the resulting distribution directly satisfies a prescribed constraint-satisfaction probability. Consequently, constrained sampling methods impose feasibility through inference-time corrections, whereas CCFO internalizes the constraint through model adaptation and retains standard sampling at inference.

Fine-tuning under distribution-level constraints. Our work is closely related to fine-tuning generative models under distribution-level constraints Khalafi et al. (2026); Gutjahr et al. (2025); Smith et al. (2025). Khalafi et al. (2026) study fine-tuning with expected inequality constraints, whereas Smith et al. (2025) seek the KL-closest distribution that matches prescribed statistics through equality constraints. In addition to constrained generation considered in these two works, CCFO additionally performs reward-driven optimization. Perhaps the closest problem is constrained generative optimization (CGO) considered in (Gutjahr et al., 2025), which maximizes expected reward subject to an expected inequality constraint. Since a chance constraint can be written as an expectation of the binary indicator, CCFO falls within this broader CGO framework. However, Gutjahr et al. (2025) handle expected constraints through sequential fine-tuning and updates of the associated Lagrange multiplier, whereas CCFO exploits the binary indicator structure to derive the closed-form expression for the optimal multiplier and only requires a single fine-tuning run. Moreover, although CGO represents chance constraints through an indicator constraint function, they do not consider the applications of probabilistic feasibility control, which is the central focus of this paper.

7 CONCLUSION

We introduced CCFO, a framework for fine-tuning diffusion and flow models under probabilistic feasibility requirements. By exploiting the binary structure of chance constraints, we derived a closed-form optimal dual multiplier and reduced the constrained optimization problem to a lightweight estimation followed by one KL-regularized fine-tuning run with a modified reward. We further characterized the optimal target distribution and established sensitivity guarantees with respect to multiplier estimation and approximate constraint evaluations. Experiments on 2D illustrative examples, text-to-image generation, and protein sequence design demonstrate improved reward–feasibility tradeoffs over existing baselines. These results provide a simple and scalable route toward incorporating probabilistic requirements into generative-model adaptation. A current limitation is that our closed-form characterization exploits the binary structure of a single chance constraint. Extending CCFO to multiple simultaneous chance constraints, where the dual variables become coupled, is an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, designing or providing feedback on research methodology or experiments, and interpreting results, and generating synthetic data sets, proposing or refining hypotheses, assisting with translation, cleaning and reformat dataset, supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating or editing code and editing a research paper to improve readability. We verified the LLM-generated code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure that our results can be reproduced. All theoretical claims are stated with their assumptions in the main paper, and their detailed proofs can be found in Appendix A and B. Appendix C provides the experimental details for all the experiments, including the 2D example, text-to-image generation, and protein sequence design. For each experiment, we provide detailed information required to reproduce the results, including the employed dataset, reward model, pre-trained model, fine-tuning method, evaluation metrics, and hyperparameters.

REFERENCES

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling rl post-training of diffusion and flow-matching models. *arXiv preprint arXiv:2605.10759*, 2026.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2022.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Jaemoo Choi, Wei Guo, Yuchen Zhu, Arash Vahdat, Molei Tao, Julius Berner, and Yongxin Chen. Scaling reinforcement learning for diffusion models via velocity matching. *arXiv preprint arXiv:2608.23664*, 2026.

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Flow density control: Generative optimization beyond entropy-regularized fine-tuning. In *Advances in neural information processing systems*, 2025a.

Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Provable maximum entropy manifold exploration via diffusion models. *arXiv preprint arXiv:2506.15385*, 2025b.

Riccardo De Santi, Malte Franke, Ya-Ping Hsieh, and Andreas Krause. A unified density operator view of flow control and merging. *arXiv preprint arXiv:2602.08012*, 2026.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. doi: 10.1017/CBO9780511581274.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Sven Gutjahr, Riccardo De Santi, Luca Schaufelberger, Kjell Jorner, and Andreas Krause. Constrained flow optimization via sequential fine-tuning for molecular design. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2025.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

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

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Luozhijie Jin, Zijie Qiu, Jie Liu, Zijie Diao, Lifeng Qiao, Ning Ding, Alex Lamb, and Xipeng Qiu. Inference-time alignment control for diffusion models with reinforcement learning guidance. *arXiv preprint arXiv:2508.21016*, 2025.

Kadina E Johnston, Patrick J Almhjell, Ella J Watkins-Dulaney, Grace Liu, Nicholas J Porter, Jason Yang, and Frances H Arnold. A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121(32):e2400439121, 2024.

Shervin Khalafi, Ignacio Hounie, Dongsheng Ding, and Alejandro Ribeiro. Composition and alignment of diffusion models using constrained learning. *Advances in Neural Information Processing Systems*, 38:18629–18676, 2026.

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

Jinhao Liang, Yixuan Sun, Anirban Samaddar, Sandeep Madireddy, and Ferdinando Fioretto. Chance-constrained flow matching for high-fidelity constraint-aware generation. *arXiv preprint arXiv:2509.25157*, 2025.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Christoph Schuhmann. Laion-aesthetics, 2022. URL https://laion.ai/blog/laion-aesthetics/.

Henry D Smith, Nathaniel L Diamant, and Brian L Trippe. Calibrating generative models to distributional constraints. *arXiv preprint arXiv:2510.10020*, 2025.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Wenpin Tang and Fuzhong Zhou. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. *arXiv preprint arXiv:2403.06279*, 2024.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

Zifan Wang, Riccardo De Santi, Xiaoyu Mo, Michael M Zavlanos, Andreas Krause, and Karl H Johansson. Efficient tail-aware generative optimization via flow model fine-tuning. *arXiv preprint arXiv:2602.16796*, 2026.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Wei Xiao, Johnson Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations*, volume 2025, pp. 100391–100412, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Jason Yang, Wenda Chu, Daniel Khalil, Raul Astudillo, Bruce Wittmann, Frances Arnold, and Yisong Yue. Steering generative models with experimental data for protein fitness optimization. *Advances in Neural Information Processing Systems*, 38:145383–145421, 2026.

Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained diffusers for safe planning and control. *Advances in Neural Information Processing Systems*, 38:34965–34998, 2026.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. *arXiv preprint arXiv:2509.16117*, 2025.

A Proofs for Chance Constrained Flow Optimization

A.1 Proof of Theorem 1

Recall that $s(x) := \mathbf{1}\{g(x) \le 0\}$. Then s takes values in $\{0, 1\}$. By definition of s,

$$\mathbb{E}_p[s(X)] = \mathbb{E}_p[\mathbf{1}\{g(X) \le 0\}] = \mathbb{P}_{X\sim p}(g(X) \le 0).$$

Therefore the primal chance-constrained problem

$$V_{\mathrm{cc}} := \sup_{p \ll p^{\mathrm{pre}}} \Big\{ \mathbb{E}_p[r(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) \Big\} \quad \text{s.t.} \quad \mathbb{P}_{X\sim p}(g(X) \le 0) \ge 1 - \beta \tag{10}$$

is equivalent to

$$V_{\mathrm{cc}} = \sup_{p \ll p^{\mathrm{pre}}} \Big\{ \mathbb{E}_p[r(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) \Big\} \quad \text{s.t.} \quad \mathbb{E}_p[s(X)] \ge 1 - \beta. \tag{11}$$

For $\lambda \ge 0$, define the Lagrangian

$$\mathcal{L}(p, \lambda) = \mathbb{E}_p[r(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) + \lambda\big(\mathbb{E}_p[s(X)] - (1 - \beta)\big).$$

Equivalently,

$$\mathcal{L}(p, \lambda) = \mathbb{E}_p[r(X) + \lambda s(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) - \lambda(1 - \beta).$$

Hence, for each fixed $\lambda \ge 0$,

$$\sup_{p \ll p^{\mathrm{pre}}} \mathcal{L}(p, \lambda) = \sup_{p \ll p^{\mathrm{pre}}} \Big\{ \mathbb{E}_p[r(X) + \lambda s(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) \Big\} - \lambda(1 - \beta). \tag{12}$$

Fix $\lambda \ge 0$, and define

$$f_\lambda(x) := r(x) + \lambda s(x).$$

Since $0 \le s(x) \le 1$, we have

$$e^{f_\lambda(x)/\alpha} = \exp\left(\frac{r(x) + \lambda s(x)}{\alpha}\right) \le e^{\lambda/\alpha} e^{r(x)/\alpha}.$$

By the assumption $\mathbb{E}_{p^{\mathrm{pre}}}[e^{r(X)/\alpha}] < \infty$, it follows that

$$Z(\lambda) := \mathbb{E}_{p^{\mathrm{pre}}}\left[e^{f_\lambda(X)/\alpha}\right] < \infty.$$

Since $\sup_{p \ll p^{\mathrm{pre}}} \Big\{ \mathbb{E}_p[f_\lambda(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) \Big\}$ is a linear generative optimization problem, its optimal value is $\alpha \log Z(\lambda)$, and the optimal distribution satisfies

$$\frac{dp^*_\lambda}{dp^{\mathrm{pre}}}(x) = \frac{e^{f_\lambda(x)/\alpha}}{Z(\lambda)} = \frac{\exp\Big(\frac{r(x)+\lambda s(x)}{\alpha}\Big)}{\mathbb{E}_{p^{\mathrm{pre}}}\Big[\exp\Big(\frac{r(X)+\lambda s(X)}{\alpha}\Big)\Big]}. \tag{13}$$

Then, we have

$$\sup_{p \ll p^{\mathrm{pre}}} \mathcal{L}(p, \lambda) = \alpha \log \mathbb{E}_{p^{\mathrm{pre}}}\left[\exp\left(\frac{r(X) + \lambda s(X)}{\alpha}\right)\right] - \lambda(1 - \beta) = G(\lambda).$$

By weak duality, for every $\lambda \ge 0$ and every feasible p,

$$\mathbb{E}_p[r(X)] - \alpha D_{\mathrm{KL}}(p\|p^{\mathrm{pre}}) \le \mathcal{L}(p, \lambda) \le \sup_{Q \ll p^{\mathrm{pre}}} \mathcal{L}(Q, \lambda) = G(\lambda).$$

Taking the supremum over feasible P yields

$$V_{\mathrm{cc}} \le G(\lambda), \qquad \forall \lambda \ge 0.$$

Therefore

$$V_{\mathrm{cc}} \le \inf_{\lambda \ge 0} G(\lambda).$$

Under the standard Slater-type feasibility condition for (11), strong duality holds for this concave maximization problem with affine constraint; hence

$$V_{\mathrm{cc}} = \inf_{\lambda \geq 0} G(\lambda).$$

Thus the original chance-constrained problem is equivalent to the one-dimensional convex optimization problem

$$\min_{\lambda \geq 0} G(\lambda).$$

In what follows, we characterize the optimal solution of $\min_{\lambda \geq 0} G(\lambda)$. Since $s(x) = \mathbf{1}\{g(x) \leq 0\}$, we have

$$\exp\left(\frac{r(x) + \lambda s(x)}{\alpha}\right) = e^{r(x)/\alpha}\mathbf{1}\{g(x) > 0\} + e^{\lambda/\alpha}e^{r(x)/\alpha}\mathbf{1}\{g(x) \leq 0\}.$$

Taking expectation under p^{pre}, and recalling the definitions

$$A = \mathbb{E}_{p^{\mathrm{pre}}}\left[e^{r(X)/\alpha}\mathbf{1}\{g(X) > 0\}\right], \qquad B = \mathbb{E}_{p^{\mathrm{pre}}}\left[e^{r(X)/\alpha}\mathbf{1}\{g(X) \leq 0\}\right],$$

we obtain

$$\mathbb{E}_{p^{\mathrm{pre}}}\left[\exp\left(\frac{r(X) + \lambda s(X)}{\alpha}\right)\right] = A + e^{\lambda/\alpha}B.$$

Hence

$$G(\lambda) = \alpha \log\big(A + e^{\lambda/\alpha}B\big) - \lambda(1 - \beta). \tag{14}$$

Differentiate (14):

$$G'(\lambda) = \alpha \cdot \frac{\frac{1}{\alpha}e^{\lambda/\alpha}B}{A + e^{\lambda/\alpha}B} - (1 - \beta) = \frac{e^{\lambda/\alpha}B}{A + e^{\lambda/\alpha}B} - (1 - \beta).$$

Differentiate once more:

$$G''(\lambda) = \frac{1}{\alpha} \cdot \frac{Ae^{\lambda/\alpha}B}{(A + e^{\lambda/\alpha}B)^2} \geq 0.$$

Thus G is convex on $[0, \infty)$. Therefore any minimizer is characterized by either the boundary case $\lambda^* = 0$, or the first-order condition $G'(\lambda^*) = 0$ when $\lambda^* > 0$.

First consider the boundary. Since

$$G'(0) = \frac{B}{A + B} - (1 - \beta),$$

if $G'(0) \geq 0$, then by convexity G is nondecreasing on $[0, \infty)$, and therefore the minimizer is

$$\lambda^* = 0.$$

Now consider the case $G'(0) < 0$. Then the minimizer lies in the interior and satisfies

$$G'(\lambda^*) = 0.$$

Thus

$$\frac{e^{\lambda^*/\alpha}B}{A + e^{\lambda^*/\alpha}B} = 1 - \beta.$$

Multiplying both sides by $A + e^{\lambda^*/\alpha}B$, we get

$$e^{\lambda^*/\alpha}B = (1 - \beta)A + (1 - \beta)e^{\lambda^*/\alpha}B.$$

Rearranging,

$$\beta e^{\lambda^*/\alpha}B = (1 - \beta)A.$$

Since $B > 0$, this yields

$$e^{\lambda^*/\alpha} = \frac{(1 - \beta)A}{\beta B},$$

and therefore

$$\lambda^* = \alpha \log \frac{(1-\beta)A}{\beta B}.$$

Combining the boundary and interior cases, we conclude that

$$\lambda^* = \alpha \left[\log \frac{(1-\beta)A}{\beta B} \right]_+ .$$

Since the primal problem is equivalent to $\min_{\lambda \geq 0} G(\lambda)$, and the minimizing dual variable is λ^*, the corresponding optimal primal distribution is the Gibbs optimizer $P^*_{\lambda^*}$ from (13). That is,

$$\frac{dp^*}{dp^{\mathrm{pre}}}(x) = \frac{\exp\left(\frac{r(x)+\lambda^* s(x)}{\alpha}\right)}{\mathbb{E}_{p^{\mathrm{pre}}}\left[\exp\left(\frac{r(X)+\lambda^* s(X)}{\alpha}\right)\right]}.$$

Equivalently,

$$p^*(x) \propto p^{\mathrm{pre}}(x) \exp\left(\frac{r(x)+\lambda^* s(x)}{\alpha}\right).$$

This completes the proof.

A.2 Proof of Proposition 1

Since s is zero on F^c and one on F, for every $\lambda \geq 0$,

$$\int \exp\left(\frac{r(x)+\lambda s(x)}{\alpha}\right) dp^{\mathrm{pre}}(x) = A + e^{\lambda/\alpha} B.$$

This normalizing constant is positive and finite. Dividing the density of p_λ by that of μ gives

$$\frac{dp_\lambda}{d\mu}(x) = \frac{e^{\lambda s(x)/\alpha}}{1-h+e^{\lambda/\alpha}h}. \tag{15}$$

Both conditional measures are well-defined because $0 < h < 1$: for $E \in \mathcal{F}$,

$$\mu(E \mid F) = \frac{\mu(E \cap F)}{h}, \qquad \mu(E \mid F^c) = \frac{\mu(E \cap F^c)}{1-h}.$$

Integrating (15) over E therefore yields

$$\begin{aligned} p_\lambda(E) &= \frac{\mu(E \cap F^c) + e^{\lambda/\alpha}\mu(E \cap F)}{1-h+e^{\lambda/\alpha}h} \\ &= \frac{1-h}{1-h+e^{\lambda/\alpha}h}\,\mu(E \mid F^c) + \frac{e^{\lambda/\alpha}h}{1-h+e^{\lambda/\alpha}h}\,\mu(E \mid F). \end{aligned}$$

The second coefficient is h_λ, and the first is $1-h_\lambda$. Since the equality holds for every measurable E, it proves the mixture identity. Setting $E = F$ also shows that $p_\lambda(F) = h_\lambda$.

It remains to compute the feasible mass at $\lambda^\star$. Recall that

$$\lambda^\star = \alpha \left[\log \frac{(1-\beta)A}{\beta B} \right]_+ .$$

If $h \geq 1-\beta$, then

$$\frac{B}{A+B} \geq 1-\beta \quad \Longleftrightarrow \quad \beta B \geq (1-\beta)A.$$

Thus, the logarithm is nonpositive, $\lambda^\star = 0$, and $p_{\lambda^\star}(F) = h$. If $h < 1-\beta$, the inequalities reverse, so $\lambda^\star > 0$ and

$$e^{\lambda^\star/\alpha} = \frac{(1-\beta)A}{\beta B}.$$

Substituting this identity into the feasible mass gives

$$p_{\lambda^\star}(F) = \frac{e^{\lambda^\star/\alpha}B}{A+e^{\lambda^\star/\alpha}B} = \frac{(1-\beta)A/\beta}{A+(1-\beta)A/\beta} = 1-\beta.$$

These two cases establish $p^\star(F) = \max\{h, 1-\beta\}$. The proof is complete.

A.3 Proof of Proposition 2

Before the proof of Proposition 2, we provide a useful lemma.

Lemma 1 (Multiplier stability)**.** *For any* $\lambda_1, \lambda_2 \geq 0$*,*

$$\|p_{\lambda_1} - p_{\lambda_2}\|_{\mathrm{TV}} = |h_{\lambda_1} - h_{\lambda_2}| \leq \frac{|\lambda_1 - \lambda_2|}{4\alpha}. \tag{16}$$

Proof: Subtracting the two mixture representations in Proposition 1, we obtain, for every $E \in \mathcal{F}$,

$$p_{\lambda_1}(E) - p_{\lambda_2}(E) = (h_{\lambda_1} - h_{\lambda_2})\big(\mu(E \mid F) - \mu(E \mid F^c)\big).$$

The absolute value of the difference of the two conditional probabilities is at most one for every E. For $E = F$, it is exactly one, since $\mu(F \mid F) = 1$ and $\mu(F \mid F^c) = 0$. Taking the supremum over measurable E thus gives the exact identity

$$\|p_{\lambda_1} - p_{\lambda_2}\|_{\mathrm{TV}} = |h_{\lambda_1} - h_{\lambda_2}|.$$

Next, differentiating the explicit expression for h_λ gives

$$\begin{aligned}\frac{dh_\lambda}{d\lambda} &= \frac{\alpha^{-1}e^{\lambda/\alpha}h(1-h+e^{\lambda/\alpha}h) - \alpha^{-1}e^{2\lambda/\alpha}h^2}{(1-h+e^{\lambda/\alpha}h)^2} \\ &= \frac{e^{\lambda/\alpha}h(1-h)}{\alpha(1-h+e^{\lambda/\alpha}h)^2} = \frac{h_\lambda(1-h_\lambda)}{\alpha}.\end{aligned}$$

Because $h_\lambda(1-h_\lambda) = \frac{1}{4} - (h_\lambda - \frac{1}{2})^2 \leq \frac{1}{4}$, we have $0 \leq dh_\lambda/d\lambda \leq 1/(4\alpha)$. Integrating this derivative between $\min\{\lambda_1, \lambda_2\}$ and $\max\{\lambda_1, \lambda_2\}$ yields

$$|h_{\lambda_1} - h_{\lambda_2}| \leq \frac{|\lambda_1 - \lambda_2|}{4\alpha}.$$

Combining the last two displays proves the lemma. The proof is complete. □

Now we are ready to prove Proposition 2. The first inequality follows directly from Lemma 1 with $\lambda_1 = \widehat{\lambda}$ and $\lambda_2 = \lambda^\star$. For the second inequality, define

$$u = \log\frac{(1-\beta)A}{\beta B}, \qquad \widehat{u} = \log\frac{(1-\beta)\widehat{A}}{\beta\widehat{B}}.$$

These quantities are finite because all four masses are positive. The map $z \mapsto [z]_+$ is 1-Lipschitz. Indeed, if both arguments are nonnegative, its difference equals their difference; if both are nonpositive, its difference is zero. If one argument is nonnegative and the other nonpositive, for example $a \geq 0 \geq b$, then $|[a]_+ - [b]_+| = a \leq a - b = |a - b|$. It follows that

$$\begin{aligned}|\widehat{\lambda} - \lambda^\star| &= \alpha|[\widehat{u}]_+ - [u]_+| \\ &\leq \alpha|\widehat{u} - u| \\ &= \alpha\left|(\log\widehat{A} - \log A) - (\log\widehat{B} - \log B)\right| \\ &= \alpha\left|\log\frac{\widehat{A}}{A} - \log\frac{\widehat{B}}{B}\right|.\end{aligned}$$

Combining this estimate with Lemma 1 completes the proof.

A.4 Finite-sample analysis of the estimated multiplier

We quantify the sampling error in the log ratio appearing in Proposition 2. Let $x_1, \ldots, x_N \overset{\text{i.i.d.}}{\sim} p^{\mathrm{pre}}$, and let $\widehat{A}, \widehat{B}$ be the empirical means in (5). The weighted observations defining $\widehat{A}$ and $\widehat{B}$ are nonnegative and integrable, with expectations A and B, respectively. When these estimates are formed from the first N observations of an infinite i.i.d. sequence, the strong law of large numbers yields

$$\widehat{A} \longrightarrow A, \qquad \widehat{B} \longrightarrow B \qquad \text{almost surely as } N \to \infty.$$

Since $A, B > 0$, both empirical masses are positive for all sufficiently large N, almost surely. Continuity of the logarithm therefore gives

$$\left|\log \frac{\widehat{A}}{A} - \log \frac{\widehat{B}}{B}\right| \longrightarrow 0 \qquad \text{almost surely.}$$

The definition of $\widehat{\lambda}$ and continuity of the positive-part map imply $\widehat{\lambda} \to \lambda^\star$ almost surely. Proposition 2 further gives $\|p_{\widehat{\lambda}} - p_{\lambda^\star}\|_{\mathrm{TV}} \to 0$ almost surely. These conclusions require only integrability; the following high-probability rate uses an additional boundedness assumption.

Proposition 4. *Assume that $e^{r(X)/\alpha} \le M$ almost surely under p^{pre} for some $M \in (0, \infty)$. For $\delta \in (0, 1)$, define*

$$\rho_N = \sqrt{\frac{3M \log(4/\delta)}{N \min\{A, B\}}}. \tag{17}$$

If $\rho_N < 1$, then, with probability at least $1 - \delta$, $\widehat{A}, \widehat{B} > 0$ and

$$\left|\log \frac{\widehat{A}}{A} - \log \frac{\widehat{B}}{B}\right| \le \log \frac{1 + \rho_N}{1 - \rho_N}. \tag{18}$$

On the same event, the multiplier $\widehat{\lambda} = \alpha \left[\log \frac{(1-\beta)\widehat{A}}{\beta \widehat{B}}\right]_+$ is well-defined and satisfies

$$\|p_{\widehat{\lambda}} - p_{\lambda^\star}\|_{\mathrm{TV}} \le \frac{1}{4} \log \frac{1 + \rho_N}{1 - \rho_N}. \tag{19}$$

In particular, if

$$N \ge \frac{12M}{\min\{A, B\}} \log \frac{4}{\delta}, \tag{20}$$

then, with probability at least $1 - \delta$,

$$\left|\log \frac{\widehat{A}}{A} - \log \frac{\widehat{B}}{B}\right| \le 4\rho_N, \qquad \|p_{\widehat{\lambda}} - p_{\lambda^\star}\|_{\mathrm{TV}} \le \rho_N. \tag{21}$$

Proof. Define the weighted observations

$$Y_i^A = e^{r(x_i)/\alpha}(1 - s(x_i)), \qquad Y_i^B = e^{r(x_i)/\alpha} s(x_i).$$

Each sequence is i.i.d. across i, and

$$0 \le Y_i^A, Y_i^B \le M, \qquad \mathbb{E}[Y_i^A] = A, \qquad \mathbb{E}[Y_i^B] = B.$$

Also, $\widehat{A} = N^{-1} \sum_i Y_i^A$ and $\widehat{B} = N^{-1} \sum_i Y_i^B$.

For independent random variables $Y_1, \ldots, Y_N \in [0, M]$ with common mean $m > 0$, the multiplicative Chernoff–Hoeffding bound, applied to Y_i/M, gives, for every $\rho \in (0, 1)$,

$$\Pr\left(\left|\frac{N^{-1} \sum_{i=1}^N Y_i}{m} - 1\right| > \rho\right) \le 2 \exp\left(-\frac{Nm\rho^2}{3M}\right); \tag{22}$$

see Dubhashi & Panconesi (2009, Chapter 1). Applying (22) separately to Y_i^A and Y_i^B, followed by a union bound, yields

$$\begin{aligned}
&\Pr\left(\left|\frac{\widehat{A}}{A} - 1\right| > \rho \text{ or } \left|\frac{\widehat{B}}{B} - 1\right| > \rho\right) \\
&\qquad \le 2 \exp\left(-\frac{NA\rho^2}{3M}\right) + 2 \exp\left(-\frac{NB\rho^2}{3M}\right) \\
&\qquad \le 4 \exp\left(-\frac{N \min\{A, B\}\rho^2}{3M}\right).
\end{aligned}$$

Although $\widehat{A}$ and $\widehat{B}$ use the same samples, independence between the two estimates is not required for this union bound. Substituting $\rho = \rho_N < 1$ makes the final expression equal to δ. Consequently, with probability at least $1 - \delta$,

$$1 - \rho_N \le \frac{\widehat{A}}{A} \le 1 + \rho_N, \qquad 1 - \rho_N \le \frac{\widehat{B}}{B} \le 1 + \rho_N. \tag{23}$$

On this event, $1 - \rho_N > 0$ implies $\widehat{A}, \widehat{B} > 0$. Dividing the bounds in (23) gives

$$\frac{1 - \rho_N}{1 + \rho_N} \le \frac{\widehat{A}/A}{\widehat{B}/B} \le \frac{1 + \rho_N}{1 - \rho_N}.$$

Taking logarithms, using their monotonicity on $(0, \infty)$, we obtain

$$-\log \frac{1 + \rho_N}{1 - \rho_N} \le \log \frac{\widehat{A}}{A} - \log \frac{\widehat{B}}{B} \le \log \frac{1 + \rho_N}{1 - \rho_N}.$$

This proves (18). Applying Proposition 2 on the same event gives (19).

Finally, (20) implies $\rho_N \le 1/2$. For $0 \le \rho \le 1/2$,

$$\log \frac{1 + \rho}{1 - \rho} = \int_{1-\rho}^{1+\rho} \frac{dt}{t} \le \frac{2\rho}{1 - \rho} \le 4\rho.$$

Substituting $\rho = \rho_N$ into (18) and (19) proves (21). □

For fixed problem parameters and confidence level δ, both the empirical log-ratio error and the resulting total-variation error admit $O(N^{-1/2})$ bounds. The constant depends on $M/\min\{A, B\}$, so the bound can require a large calibration sample when either region has little reward-weighted mass relative to the maximum weight.

The same event also controls the feasibility shortfall. Under (20), Proposition 1 and the definition of total variation give

$$p_{\widehat{\lambda}}(F) \ge p_{\lambda^\star}(F) - \|p_{\widehat{\lambda}} - p_{\lambda^\star}\|_{\mathrm{TV}} \ge 1 - \beta - \rho_N.$$

These statements concern the ideal target $p_{\widehat{\lambda}}$ and do not include approximation error from the fine-tuning solver.

A.5 Proof of Proposition 3

Let $\hat{g}$ be measurable, so that $\hat{s} = \mathbf{1}\{\hat{g} \le 0\}$ is a measurable binary function. Fix $\lambda \ge 0$ and define, only within this proof,

$$c = e^{\lambda/\alpha} - 1, \qquad a(x) = 1 + c\, s(x), \qquad \widehat{a}(x) = 1 + c\, \hat{s}(x).$$

Since $s, \hat{s} \in \{0, 1\}$, we have

$$a(x) = e^{\lambda s(x)/\alpha}, \qquad \widehat{a}(x) = e^{\lambda \hat{s}(x)/\alpha}.$$

Both functions take values in $[1, e^{\lambda/\alpha}]$. Their normalizing constants $Z = \int a\, d\mu, , \widehat{Z} = \int \widehat{a}\, d\mu$ therefore satisfy $1 \le Z, \widehat{Z} \le e^{\lambda/\alpha}$, and

$$\frac{dp_\lambda}{d\mu} = \frac{a}{Z}, \qquad \frac{d\widehat{p}_\lambda}{d\mu} = \frac{\widehat{a}}{\widehat{Z}}.$$

Moreover,

$$\int |\widehat{a} - a|\, d\mu = c \int |\hat{s} - s|\, d\mu = c\delta_g, \qquad |\widehat{Z} - Z| \le \int |\widehat{a} - a|\, d\mu = c\delta_g. \tag{24}$$

We next recall why total variation equals half the L^1 distance between probability densities. If f and $\hat{f}$ are densities with respect to μ, then $H = \hat{f} - f$ satisfies $\int H\, d\mu = 0$. Its positive and negative

parts therefore have equal integrals: $\int H_+ \, d\mu = \int H_- \, d\mu = \frac{1}{2} \int |H| \, d\mu$. For every measurable E, $\int_E H \, d\mu$ lies between $-\int H_- \, d\mu$ and $\int H_+ \, d\mu$; the upper endpoint is attained at $E = \{H \geq 0\}$. Taking the supremum of the absolute value proves

$$\sup_E \left| \int_E (\hat{f} - f) \, d\mu \right| = \frac{1}{2} \int |\hat{f} - f| \, d\mu.$$

Applying this identity to the two densities above, and using

$$\frac{\widehat{a}}{\widehat{Z}} - \frac{a}{Z} = \frac{\widehat{a} - a}{\widehat{Z}} + a \left(\frac{1}{\widehat{Z}} - \frac{1}{Z} \right),$$

we obtain

$$\begin{aligned} 2\|\widehat{p}_\lambda - p_\lambda\|_{\mathrm{TV}} &= \int \left| \frac{\widehat{a}}{\widehat{Z}} - \frac{a}{Z} \right| d\mu \\ &\leq \frac{1}{\widehat{Z}} \int |\widehat{a} - a| \, d\mu + \left| \frac{1}{\widehat{Z}} - \frac{1}{Z} \right| \int a \, d\mu \\ &= \frac{\int |\widehat{a} - a| \, d\mu + |Z - \widehat{Z}|}{\widehat{Z}} \\ &\leq \frac{2c\delta_g}{\widehat{Z}} \leq 2c\delta_g. \end{aligned}$$

The penultimate inequality uses (24); the last uses $\widehat{Z} \geq 1$. Dividing by two and substituting $c = e^{\lambda/\alpha} - 1$ proves $\|\widehat{p}_\lambda - p_\lambda\|_{\mathrm{TV}} \leq \big(e^{\lambda/\alpha} - 1\big)\delta_g$.

Finally, F is measurable, so the definition of total variation gives

$$|\widehat{p}_\lambda(F) - p_\lambda(F)| \leq \|\widehat{p}_\lambda - p_\lambda\|_{\mathrm{TV}}.$$

Combining the two inequalities proves $|\widehat{p}_\lambda(F) - p_\lambda(F)| \leq \big(e^{\lambda/\alpha} - 1\big)\delta_g$. The proof is complete.

B RELATION TO GATED REWARD FINE-TUNING

A simple heuristic for encouraging constraint satisfaction is the so-called *gated-reward fine-tuning*, which replaces $r(x)$ with $r(x)s(x)$ and thus assigns zero reward to infeasible samples. We compare this approach with CCFO at the level of their ideal target distributions, using the same reference distribution p^{pre}, reward r, feasibility indicator s, and KL regularization coefficient α. It can be shown that the gated-reward target is

$$\frac{dp^{\mathrm{gate}}}{dp^{\mathrm{pre}}}(x) = \frac{\exp(r(x)s(x)/\alpha)}{B + p^{\mathrm{pre}}(F^c)}. \tag{25}$$

The following proposition characterizes its relationship with the CCFO target family p_λ.

Proposition 5. *Assume $0 < A, B < \infty$. For $\lambda \geq 0$, CCFO and gated-reward fine-tuning satisfy*

$$p_\lambda(\cdot \mid F) = p^{\mathrm{gate}}(\cdot \mid F) = \mu(\cdot \mid F). \tag{26}$$

On the infeasible region,

$$p_\lambda(\cdot \mid F^c) = \mu(\cdot \mid F^c), \qquad p^{\mathrm{gate}}(\cdot \mid F^c) = p^{\mathrm{pre}}(\cdot \mid F^c). \tag{27}$$

Their feasibility probabilities are $p_\lambda(F) = \frac{e^{\lambda/\alpha}B}{A+e^{\lambda/\alpha}B}$ and $p^{\mathrm{gate}}(F) = \frac{B}{B+p^{\mathrm{pre}}(F^c)}$, respectively.

Proposition 5 shows that CCFO and gated-reward fine-tuning share the same conditional distribution of feasible samples. Their ideal targets therefore differ in two respects: the probability mass assigned to the feasible set and the reward-based preferences retained within the infeasible region.

Explicit control of feasibility. CCFO selects $\lambda^\star$ to ensure $p_{\lambda^\star}(F) = \max\{h, 1-\beta\}$, where $h = \mu(F)$. In contrast, the feasibility probability of the uncalibrated gated target is determined implicitly by r, p^{pre}, F, and α, and need not satisfy a prescribed requirement. CCFO thus provides explicit control over how frequently feasible samples are generated without changing their conditional distribution.

Preserving the original preference within infeasible region. Within F^c, CCFO retains the reward-tilted conditional distribution $\mu(\cdot \mid F^c)$, whereas gating preserves $p^{\mathrm{pre}}(\cdot \mid F^c)$ and removes reward-based preferences among infeasible samples. At matched feasibility, the CCFO target achieves both an expected original reward and a KL-regularized original objective at least as large as those of the gated target; see Proposition 6 for more details. Since their feasible conditional distributions coincide, the improvement arises entirely from the treatment of infeasible samples.

Preserving the original reward is meaningful when the constraint oracle is imperfect. For example, in a scientific discovery task with an inaccurate feasibility oracle, a high-reward candidate classified as infeasible may still warrant further evaluation. CCFO retains reward-based preferences among such candidates, whereas gated reward fine-tuning assigns them all zero reward. Conversely, when infeasible samples have no downstream utility, the gated reward rs may better represent the intended objective. This choice is compatible with explicit feasibility control: applying CCFO with rs as the base reward yields the pseudo-reward $(r+\lambda)s$, with λ calibrated for this gated objective to meet the prescribed feasibility requirement.

Comparison at matched feasibility. The next result quantifies the difference for the original, ungated utility. It includes the unshifted gated baseline by setting $c = 0$, and also covers the matched shifted gate in (31).

Proposition 6 (Original utility at matched feasibility)**.** *In addition to the standing assumptions, suppose $\mathbb{E}_\mu[|r(X)|] < \infty$ and $\mathbb{E}_{p^{\mathrm{pre}}}[|r(X)|] < \infty$. Let $\lambda \geq 0$ and $c \in \mathbb{R}$ be finite and satisfy $p_\lambda(F) = p_c^{\mathrm{gate}}(F) = q \in (0,1)$. Define, for this comparison,*

$$J(p) = \mathbb{E}_p[r(X)] - \alpha D_{\mathrm{KL}}(p \| p^{\mathrm{pre}}).$$

Then

$$\begin{aligned} J(p_\lambda) - J(p_c^{\mathrm{gate}}) &= \alpha(1-q) D_{\mathrm{KL}}(p^{\mathrm{pre}}(\cdot \mid F^c) \| \mu(\cdot \mid F^c)) \\ &\geq 0. \end{aligned} \tag{28}$$

Moreover,

$$\begin{aligned}&\mathbb{E}_{p_\lambda}[r(X)] - \mathbb{E}_{p_c^{\mathrm{gate}}}[r(X)] \\ &\quad = \alpha(1-q)\Big[D_{\mathrm{KL}}(\mu(\cdot \mid F^c)\|p^{\mathrm{pre}}(\cdot \mid F^c)) \\ &\qquad + D_{\mathrm{KL}}(p^{\mathrm{pre}}(\cdot \mid F^c)\|\mu(\cdot \mid F^c))\Big] \geq 0.\end{aligned} \tag{29}$$

Both gaps vanish if and only if r is constant $p^{\mathrm{pre}}(\cdot \mid F^c)$-almost surely.

Additive reward shifts and calibration of a gated target. For a constant $c \in \mathbb{R}$, replace r by $r + c$. The corresponding CCFO masses are

$$A_c = e^{c/\alpha}A, \qquad B_c = e^{c/\alpha}B.$$

The multiplier is unchanged because

$$\alpha\left[\log\frac{(1-\beta)A_c}{\beta B_c}\right]_+ = \alpha\left[\log\frac{(1-\beta)A}{\beta B}\right]_+ = \lambda^\star.$$

For any fixed finite $\lambda \geq 0$, the common factor $e^{c/\alpha}$ also cancels from the target density:

$$\frac{e^{(r(x)+c+\lambda s(x))/\alpha}}{A_c + e^{\lambda/\alpha}B_c} = \frac{e^{(r(x)+\lambda s(x))/\alpha}}{A + e^{\lambda/\alpha}B}.$$

Hence both the multiplier and the target distribution are invariant to the shift. The same cancellation applies to the empirical multiplier when the same calibration samples are used and both empirical masses are positive.

For gated-reward fine-tuning, the shifted target instead satisfies

$$\frac{dp_c^{\mathrm{gate}}}{dp^{\mathrm{pre}}}(x) = \frac{e^{(r(x)+c)s(x)/\alpha}}{p^{\mathrm{pre}}(F^c) + e^{c/\alpha}B}.$$

The conditional distributions remain unchanged:

$$p_c^{\mathrm{gate}}(\cdot \mid F) = \mu(\cdot \mid F), \qquad p_c^{\mathrm{gate}}(\cdot \mid F^c) = p^{\mathrm{pre}}(\cdot \mid F^c).$$

The feasible odds are

$$\frac{p_c^{\mathrm{gate}}(F)}{1 - p_c^{\mathrm{gate}}(F)} = \frac{e^{c/\alpha}B}{p^{\mathrm{pre}}(F^c)}.$$

They are strictly increasing in c and range from zero to infinity. For any desired probability $q \in (0,1)$, the unique offset attaining $p_c^{\mathrm{gate}}(F) = q$ is consequently

$$c = \alpha \log\frac{q\,p^{\mathrm{pre}}(F^c)}{(1-q)B}. \tag{30}$$

In particular, to match the feasibility of a given p_λ, use

$$c = \lambda - \alpha\log\frac{A}{p^{\mathrm{pre}}(F^c)}. \tag{31}$$

This follows by substituting $q/(1-q) = e^{\lambda/\alpha}B/A$ into (30). Thus, gating itself does not prevent probability calibration; an uncalibrated gated reward simply fixes this offset at zero.

B.1 Proof of Proposition 5.

Since s is binary,

$$e^{r(x)s(x)/\alpha} = e^{r(x)/\alpha}s(x) + (1 - s(x)).$$

Its integral under p^{pre} is therefore $B + p^{\mathrm{pre}}(F^c)$, which is finite and strictly positive. The exponential-tilt characterization of KL-regularized fine-tuning gives (25). Integrating this density over F gives

$$p^{\mathrm{gate}}(F) = \frac{B}{B + p^{\mathrm{pre}}(F^c)}.$$

For any measurable set E, conditioning on F then yields

$$p^{\text{gate}}(E \mid F) = \frac{p^{\text{gate}}(E \cap F)}{p^{\text{gate}}(F)} = \frac{\int_{E\cap F} e^{r(x)/\alpha}\, dp^{\text{pre}}(x)}{B} = \mu(E \mid F).$$

Similarly, the density of p_λ gives

$$p_\lambda(F) = \frac{e^{\lambda/\alpha} B}{A + e^{\lambda/\alpha} B}.$$

Since λ is finite, this probability lies strictly between zero and one, and

$$p_\lambda(E \mid F) = \frac{e^{\lambda/\alpha} \int_{E\cap F} e^{r(x)/\alpha}\, dp^{\text{pre}}(x)}{e^{\lambda/\alpha} B} = \frac{\int_{E\cap F} e^{r(x)/\alpha}\, dp^{\text{pre}}(x)}{B} = \mu(E \mid F).$$

These identities prove (26).

On F^c, the gated exponential weight equals one, whereas the CCFO weight equals $e^{r(x)/\alpha}$. Thus,

$$p^{\text{gate}}(E \mid F^c) = \frac{p^{\text{pre}}(E \cap F^c)}{p^{\text{pre}}(F^c)}, \qquad p_\lambda(E \mid F^c) = \frac{\int_{E\cap F^c} e^{r(x)/\alpha}\, dp^{\text{pre}}(x)}{A}.$$

The right-hand sides are $p^{\text{pre}}(E \mid F^c)$ and $\mu(E \mid F^c)$, respectively. This proves (27). The two feasibility probabilities were established above, completing the proof.

B.2 Proof of Proposition 6

For the duration of the proof, write

$$Q = \mu(\cdot \mid F), \qquad P = \mu(\cdot \mid F^c), \qquad P_0 = p^{\text{pre}}(\cdot \mid F^c).$$

The conditional-distribution identities and the matching masses give

$$p_\lambda = qQ + (1-q)P, \qquad p_c^{\text{gate}} = qQ + (1-q)P_0.$$

The integrability assumptions imply finite absolute reward expectations under Q, P, P_0 and hence under both mixtures. The logarithms of the target densities relative to p^{pre} are affine functions of r on each of F, F^c, with finite constants. Their absolute expectations are therefore finite as well, so the objective values used below are well-defined.

Set $K = A/p^{\text{pre}}(F^c) \in (0, \infty)$. On the infeasible region, the conditional density ratio is

$$\frac{dP}{dP_0}(x) = \frac{e^{r(x)/\alpha}}{K}.$$

This ratio is strictly positive and finite almost surely, since r is real-valued. The conditional measures are thus mutually absolutely continuous. Taking its logarithm and integrating under P and P_0, respectively, gives

$$D_{\text{KL}}(P\|P_0) = \frac{\mathbb{E}_P[r(X)]}{\alpha} - \log K, \tag{32}$$

$$D_{\text{KL}}(P_0\|P) = \log K - \frac{\mathbb{E}_{P_0}[r(X)]}{\alpha}. \tag{33}$$

All terms are finite by the moment assumptions. Adding these identities yields

$$\mathbb{E}_P[r(X)] - \mathbb{E}_{P_0}[r(X)] = \alpha\big(D_{\text{KL}}(P\|P_0) + D_{\text{KL}}(P_0\|P)\big).$$

The common feasible component of the two mixtures cancels, so

$$\mathbb{E}_{p_\lambda}[r(X)] - \mathbb{E}_{p_c^{\text{gate}}}[r(X)] = (1-q)\big(\mathbb{E}_P[r(X)] - \mathbb{E}_{P_0}[r(X)]\big).$$

Substitution proves (29).

To compare the regularized objectives, let $Z_\lambda = A + e^{\lambda/\alpha} B$. The target density satisfies

$$\log \frac{dp_\lambda}{dp^{\mathrm{pre}}}(x) = \frac{r(x) + \lambda s(x)}{\alpha} - \log Z_\lambda.$$

Using this identity to expand $D_{\mathrm{KL}}(p_c^{\mathrm{gate}} \| p_\lambda)$ gives

$$\begin{aligned} D_{\mathrm{KL}}(p_c^{\mathrm{gate}} \| p_\lambda) = D_{\mathrm{KL}}(p_c^{\mathrm{gate}} \| p^{\mathrm{pre}}) \\ - \frac{\mathbb{E}_{p_c^{\mathrm{gate}}}[r(X)] + \lambda q}{\alpha} + \log Z_\lambda. \end{aligned}$$

Rearranging,

$$J(p_c^{\mathrm{gate}}) = \alpha \log Z_\lambda - \lambda q - \alpha D_{\mathrm{KL}}(p_c^{\mathrm{gate}} \| p_\lambda).$$

Integrating the same log-density identity under p_λ gives $J(p_\lambda) = \alpha \log Z_\lambda - \lambda q$. Consequently,

$$J(p_\lambda) - J(p_c^{\mathrm{gate}}) = \alpha D_{\mathrm{KL}}(p_c^{\mathrm{gate}} \| p_\lambda).$$

Because the two distributions have identical restrictions to F and equal mass on F^c, their density ratio is one on F and dP_0/dP on F^c. Integration over the two regions therefore yields

$$D_{\mathrm{KL}}(p_c^{\mathrm{gate}} \| p_\lambda) = (1 - q) D_{\mathrm{KL}}(P_0 \| P).$$

This proves (28). Nonnegativity follows from nonnegativity of KL divergence. Finally, because $q < 1$, either gap can be zero only when $P = P_0$. The conditional density formula shows that this equality holds if and only if $e^{r/\alpha} = K$ P_0-almost surely, equivalently $r = \alpha \log K$ P_0-almost surely. This proves the equality characterization.

C EXPERIMENTAL RESULTS

C.1 2D EXAMPLES

C.1.1 IMPLEMENTATION

Distribution and optimization. We discretize $[-4, 6] \times [-4, 4]$ using a uniform 161×129 grid $\mathcal{G}$ with spacing 0.0625 along both coordinates. Each method produces a categorical distribution over the resulting 20769 grid points. The reference distribution is obtained by evaluating an isotropic Gaussian mixture on the grid and renormalizing:

$$p_{\mathrm{pre}}(x_i) \propto \sum_{k=1}^{4} w_k \mathcal{N}(x_i; \mu_k, \sigma_k^2 I_2), \qquad x_i \in \mathcal{G}. \tag{34}$$

The mixture parameters are listed in Table 3. We optimize the distribution directly on this finite support, without neural approximation, so that these examples isolate the effect of the constraint formulation. All optimized distributions maximize

$$J(q) = \mathbb{E}_{X\sim q}[r(X)] - \alpha D_{\mathrm{KL}}(q\|p_{\mathrm{pre}}), \qquad \alpha = 0.7. \tag{35}$$

Let $\mathcal{F} = \{x \in \mathcal{G} : g(x) \le 0\}$. CFO imposes $\mathbb{E}_q[g(X)] \le c$, whereas CCFO imposes $q(\mathcal{F}) \ge 1 - \beta$. The visual comparisons use $c = 0$ and $\beta \in \{0.1, 0\}$.

Table 3: Parameters of the pre-trained Gaussian mixtures. Each component has covariance $\sigma_k^2 I_2$.

Example	μ_k	σ_k	w_k
Triangles	$(-2.3, -0.5)$	0.75	0.24
	$(1.4, 0.1)$	0.78	0.24
	$(2.5, 0.35)$	0.60	0.24
	$(0, 0)$	1.10	0.28
Striped Tunnel	$(-0.8, 0)$	0.80	0.28
	$(1.0, 0)$	0.70	0.30
	$(2.2, 0.2)$	0.65	0.24
	$(3.2, -0.1)$	0.65	0.18

Triangles. For a triangle T, define $h_T(x) = \max_{j=1,2,3} n_j^\top (x - v_j)$, where v_j lies on its jth edge and n_j is the corresponding outward unit normal. Thus, $h_T(x) \le 0$ exactly inside or on the boundary of T. The two triangles have vertices

$$\begin{aligned} T_1 &: \quad (-3.2, -2.2),\ (-1.2, -2.0),\ (-2.4, 1.2), \\ T_2 &: \quad (0.3, -2.2),\ (3.7, 0.15),\ (1.0, 2.2), \end{aligned}$$

and the constraint is

$$g_{\mathrm{tri}}(x) = \min\{h_{T_1}(x), h_{T_2}(x)\}. \tag{36}$$

The reward is a sum of three Gaussian peaks:

$$r_{\mathrm{tri}}(x) = -1 + 3.8 \sum_{j=1}^{3} a_j \exp\left(-\frac{\|x - m_j\|_2^2}{2s_j^2}\right), \tag{37}$$

with

$$\begin{aligned} (m_1, m_2, m_3) &= ((3.15, 0.70), (1.55, 0.35), (-2.15, -0.15)), \\ (s_1, s_2, s_3) &= (0.34, 0.55, 0.70), \\ (a_1, a_2, a_3) &= (1.90, 0.55, 0.40). \end{aligned}$$

The strongest peak lies just outside T_2, making concentration near this peak rewarding but frequently infeasible.

Striped Tunnel. Let T have vertices $(-1.5, -2.6)$, $(-1.5, 2.6)$, and $(5.2, 0)$. The feasible region is its intersection with the band $|x_2| \le 1.15$, excluding the interiors of three circular regions. We use

$$g_{\mathrm{str}}(x) = \max\left\{ h_T(x),\ |x_2| - 1.15,\quad \max_{j=1,2,3}\left(\rho_j^2 - \|x - o_j\|_2^2\right) \right\}, \tag{38}$$

where

$$\begin{aligned}(o_1, o_2, o_3) &= ((2.4, 0.55), (3.45, -0.55), (4.45, 0.15)),\\ (\rho_1, \rho_2, \rho_3) &= (0.72, 0.72, 0.68).\end{aligned}$$

The reward consists of vertical stripes with increasing amplitudes:

$$\begin{aligned} r_{\mathrm{str}}(x) = &-0.5 + 1.8 \exp\left(-\frac{x_2^2}{2(0.95)^2}\right) \sum_{j=1}^{4} a_j \exp\left(-\frac{(x_1 - b_j)^2}{2s_j^2}\right) \\ &+ 0.8 \exp\left(-\frac{x_2^2}{2(0.80)^2}\right), \end{aligned} \tag{39}$$

where $(b_1, b_2, b_3, b_4) = (2.2, 3.1, 4.0, 4.9)$, $(s_1, s_2, s_3, s_4) = (0.22, 0.22, 0.24, 0.24)$, and $(a_1, a_2, a_3, a_4) = (0.70, 1.00, 1.25, 1.55)$. The circular exclusions overlap the rewarding part of the corridor.

Computing the optimized distributions. Without a constraint, the solution is $q_0(x) \propto p_{\mathrm{pre}}(x) \exp(r(x)/\alpha)$. The CFO solution has the form

$$q_{\mathrm{CFO},c}(x) \propto p_{\mathrm{pre}}(x) \exp\left(\frac{r(x) - \nu_c g(x)}{\alpha}\right). \tag{40}$$

If $\mathbb{E}_{q_0}[g(X)] \le c$, we set $\nu_c = 0$. Otherwise, we bracket a feasible multiplier by repeated doubling from 1 and perform 80 bisection iterations to solve $\mathbb{E}_{q_{\mathrm{CFO},c}}[g(X)] = c$.

For CCFO, let $s_0 = q_0(\mathcal{F})$ and $f = 1 - \beta$. If $f \le s_0$, the constraint is inactive and the solution is q_0. If $f > s_0$, the solution is

$$q_{\mathrm{CCFO},\beta}(x) = \begin{cases} \dfrac{f}{s_0} q_0(x), & x \in \mathcal{F}, \\ \dfrac{1 - f}{1 - s_0} q_0(x), & x \notin \mathcal{F}. \end{cases} \tag{41}$$

In particular, $\beta = 0$ gives q_0 conditioned on $\mathcal{F}$. These solutions are computed to numerical precision on the discrete grid; they do not require stochastic training.

Evaluation and visualization. We compute expected reward, feasibility probability, and expected constraint value by summing over the entire grid. For each visual panel, we draw 700 samples with random seed 0. Independent Gaussian jitter with standard deviation $0.24 \times 0.0625 = 0.015$ per coordinate is added for display, and marker types are determined by evaluating g at the displayed positions. This display jitter can move a point across a nearby boundary; the reported metrics refer to the original grid distributions. The visualizations show $[-4, 4] \times [-3, 3]$, whereas all numerical results use the full grid domain.

C.1.2 Additional results

An additional example: Triangles. Figure 7 considers a disconnected feasible set formed by the union of two triangles, with the strongest reward peak located just outside the right triangle. CFO with $c = 0$ satisfies its expected constraint at equality, yet only 37.3% of its probability mass is feasible. In contrast, CCFO attains 90% feasibility with $\beta = 0.1$ and restricts its support to the feasible set with $\beta = 0$. Allowing a 10% violation budget increases expected reward from 4.573 to 4.675. This additional example illustrates that direct control of feasibility also applies when the feasible region has multiple disconnected components.

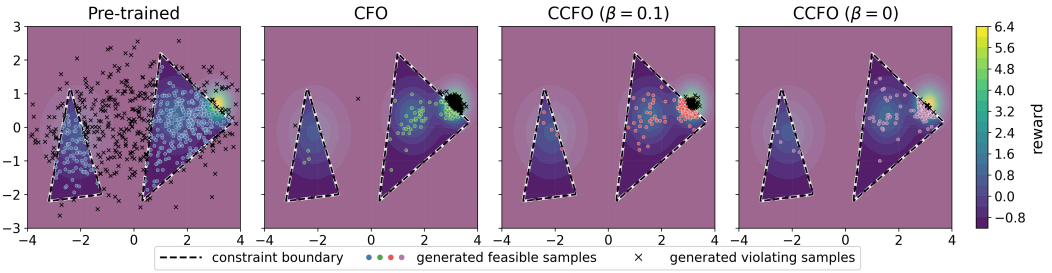


Figure 7: Generated samples in the supplementary **Triangles** example. Background colors indicate reward, with a translucent pink overlay outside the feasible set. Dashed curves mark the constraint boundary; colored circles denote feasible samples and black crosses denote violations. CFO satisfies its expected constraint while assigning substantial mass outside the two triangles. CCFO controls feasibility through the violation budget β.

Table 4: Metrics computed over the full grid. Unconstrained FT denotes the reward-tilted distribution q_0. CFO uses $c = 0$.

Example	Method	$\mathbb{E}[r(X)]$	$\Pr(g(X) \le 0)$	$\mathbb{E}[g(X)]$
Triangles	Pre-trained	-0.215	0.550	0.00156
	Unconstrained FT	5.327	0.262	0.06652
	CFO	5.041	0.373	0.00000
	CCFO ($\beta = 0.1$)	4.675	0.900	-0.08499
	CCFO ($\beta = 0$)	4.573	1.000	-0.10875
Striped Tunnel	Pre-trained	0.426	0.587	-0.16646
	Unconstrained FT	1.120	0.383	-0.00038
	CFO	1.120	0.383	-0.00038
	CCFO ($\beta = 0.1$)	0.733	0.900	-0.35925
	CCFO ($\beta = 0$)	0.658	1.000	-0.42863

Quantitative comparison. Table 4 reports the distribution-level metrics for both examples. In Striped Tunnel, the unconstrained reward-tilted distribution already has $\mathbb{E}[g(X)] \approx -3.80 \times 10^{-4}$, so CFO with $c = 0$ leaves it unchanged despite a 61.7% violation probability. These examples demonstrate that an expected constraint on a signed function does not directly specify the probability of satisfying that constraint.CCFO attains the prescribed 90% feasibility in both examples. In Stripe Tunnel, allowing a 10% violation budget improves expected reward from 0.658 to 0.733 relative to $\beta = 0$, consistent with the behavior in Triangles.

Reward–feasibility trade-offs. We keep the reference distribution, reward, constraint function, grid, and α fixed, and vary only the constraint threshold. For CFO, we sweep 201 equally spaced expected-constraint thresholds from $\mathbb{E}_{q_0}[g(X)]$ down to the threshold attaining 99.9% feasibility. The resulting threshold intervals are approximately $[-0.962827, 0.066518]$ for Triangles and $[-1.001064, -0.000380]$ for Striped Tunnel. For CCFO, we sweep 201 equally spaced feasibility targets from s_0 to 1, equivalently varying β from $1 - s_0$ to 0. Each curve is plotted against its *achieved* feasibility probability, and the vertical axis reports expected reward without subtracting the KL penalty. We additionally solve for CFO thresholds attaining exactly 90% feasibility; these are approximately -0.696564 and -0.690502, respectively.

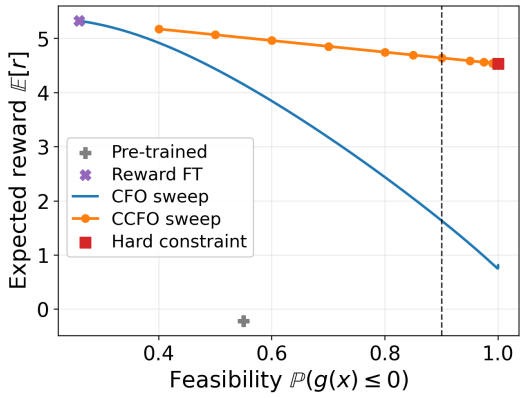


Figure 8: Reward–feasibility trade-offs in the "triangle" example.

Figure 8 shows that CCFO achieves a higher expected reward at matched feasibility throughout the evaluated CFO sweep, except at the shared unconstrained endpoint where the two methods coincide. At 90% feasibility, the rewards are 4.675 versus 1.645 in Triangles and 0.733 versus 0.480 in Striped Tunnel. Thus, CCFO improves the empirical reward–feasibility frontier over th CFO family in these examples.

C.2 TEXT-TO-IMAGE GENERATION

C.2.1 IMPLEMENTATION DETAILS

Experimental setup. We study constrained text-to-image generation, where the objective is to improve text-rendering accuracy while maintaining visual aesthetics. We use Stable Diffusion 3.5 Medium as the pretrained model and build our fine-tuning pipeline on DiffusionNFT (Zheng et al., 2025). We use the OCR prompt dataset provided with its implementation. After removing duplicate training prompts and train–test overlap, the dataset contains 19,648 training prompts and 1,018 held-out test prompts.

Reward and constraint. The reward $r(x, c) \in [0, 1]$ measures how accurately image x renders the target text specified in prompt c. We use the English PaddleOCR recognizer and follow the OCR scoring convention of the underlying implementation. The target is the first double-quoted text span in the prompt. Both the target and recognized text are lowercased and have spaces removed. The reward is one if the recognized text contains the target; otherwise, it is one minus the Levenshtein distance normalized by the target length, clipped to $[0, 1]$. Thus, the reward provides partial credit rather than measuring binary exact-match accuracy.

Let $a(x)$ denote the aesthetic score, computed using the frozen CLIP ViT-L/14-based aesthetic predictor. We define

$$g(x) = \tau - a(x), \qquad h(x) = \mathbf{1}\{a(x) \geq \tau\}, \qquad \tau = 5.3.$$

The threshold is selected using pretrained samples from training prompts only: we round the median aesthetic score of the calibration pool to one decimal place. The chance constraint is $\mathbb{E}[h(x)] \geq 1-\beta$, where the expectation includes both the prompt distribution and conditional image generation.

Fine-tuning protocol. All fine-tuning methods use DiffusionNFT as the backbone. Each run uses two NVIDIA GH200 GPUs and 50 optimizer updates. Each update collects 1,152 images, comprising 48 prompt groups with 24 images per group. The per-GPU microbatch size is 12, with 48 gradient-accumulation steps and one inner training epoch per rollout. Training uses 512×512 images, the deterministic DPM2 sampler with 10 sampling steps, and classifier-free guidance (CFG) scale 1.0. We use AdamW with a constant learning rate of 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 10^{-4}, $\epsilon = 10^{-8}$, and gradient-norm clipping at 1.0. We set the reference-regularization coefficient to $\kappa = 10^{-2}$. Specifically, this coefficient weights the mean-squared discrepancy between the trainable and frozen pretrained velocity predictions. We evaluate exponential-moving-average adapter weights, with EMA decay capped at 0.9. All fine-tuning methods use the same pre-trained model, the same solver, and the same computational budget.

Constraint handling and calibration. CFO and CCFO use the respective modified rewards

$$R_{\mathrm{CFO}}(x, c) = r(x, c) + \lambda\big(a(x) - \tau\big), \qquad R_{\mathrm{CCFO}}(x, c) = r(x, c) + \lambda_\beta h(x).$$

For CCFO, we compute λ_β before fine-tuning using a fixed pool of 64 training prompts and 16 pretrained samples per prompt. The main quantitative table reports CCFO with $\beta = 0.18$ and CFO with $\lambda = 0.20$. The former corresponds to the calibrated multiplier $\lambda_\beta \approx 0.46067$.

Additional baselines. Pre-trained denotes the unmodified pre-trained model. Reward FT optimizes the OCR reward without any constraint, whereas Gated reward FT uses $R(x, c) = r(x, c)\mathbf{1}\{a(x) \geq 5.3\}$. We also implement CGM (Smith et al., 2025), a framework of calibrating generative models to distributional equality constraints. Note that the original CGM methods, including CGM-reward and CGM-relax, do not incorporate reward maximization. Nevertheless, we implement the reward-augmented adaptations of both approaches, targeting the equality $\mathbb{E}[h(x)] = 0.90$. Specifically, CGM-reward uses an offline-calibrated indicator bonus with multiplier approximately 0.70142, while CGM-relax uses the squared moment penalty $50(\mathbb{E}[h(x)]-0.90)^2$, in addition to OCR reward maximization.

Evaluation. For the main quantitative table, each method is evaluated on all 1,018 held-out prompts, generating one image per prompt in each of four sampling rounds. Sampling seeds are

paired across methods, using seed bases 2,700,000, 2,800,000, and 2,900,000, plus the prompt index. Evaluation uses the same resolution, sampler, sampling-step count, and CFG scale as training.

We report mean OCR reward over all samples, feasibility, mean OCR reward conditional on feasibility, ImageReward, HPS v2.1, and CLIPScore. All metrics are computed from the same generated images. OCR is always evaluated using the original, unmodified reward, including for Gated reward FT. Conditional OCR is computed on each method's own feasible subset. For each metric, we report the mean and standard deviation over 3 seeds.

C.2.2 Additional results

Ablation on different β. Table 5 shows how β affects the empirical reward–feasibility trade-off. We observe that most CCFO settings achieve the desired target feasibility, i.e., feasibility is larger than $1-\beta$. Reducing β from 0.30 to 0.07 increases aesthetic feasibility from 83.79% to 98.23%, while decreasing mean OCR from 0.8984 to 0.3473. An intermediate choice, $\beta = 0.18$, achieves an OCR score of 0.7891 with 90.67% feasibility, together with the highest ImageReward and HPS among the reported CCFO settings. Thus, small values of β favor constraint satisfaction, but can reduce text-rendering performance.

Table 5: Comprehensive comparison on text-to-image generation. CFO and CCFO are evaluated with different values of λ and β, respectively.

Method	Mean OCR ↑ (all)	Feasibility (%) ↑ $\mathbb{P}$(aesthetic ≥ 5.3)	Mean OCR ↑ (given feasible)	ImageReward ↑	HPS ↑	ClipScore ↑
Pre-trained	0.1140	54.03	0.0905	-0.3422	0.2032	0.2646
Reward FT	0.9528	0.39	0.9818	-1.5696	0.1532	0.2713
Gated Reward FT	0.6174	93.20	0.6214	0.2934	0.2385	0.2818
Augmented CGM-relax	0.5676	81.75	0.5168	0.2932	0.2427	0.2776
CFO ($\lambda = 0.02$)	0.9427	11.20	0.9629	-0.8581	0.1848	0.2795
CFO ($\lambda = 0.12$)	0.8804	74.46	0.8972	0.1133	0.2351	0.2831
CFO ($\lambda = 0.17$)	0.8479	83.30	0.8606	0.2468	0.2389	0.2851
CFO ($\lambda = 0.20$)	0.7817	89.26	0.7897	0.4178	0.2417	0.2846
CFO ($\lambda = 0.35$)	0.2446	99.71	0.2445	-0.3088	0.2185	0.2214
CCFO ($\beta = 0.07$)	0.3473	98.23	0.3467	0.1992	0.2356	0.2569
CCFO ($\beta = 0.15$)	0.6129	93.91	0.6144	0.1657	0.2270	0.2670
CCFO ($\beta = 0.18$)	0.7891	90.67	0.7947	0.4306	0.2437	0.2780
CCFO ($\beta = 0.25$)	0.8379	88.02	0.8535	0.2209	0.2354	0.2776
CCFO ($\beta = 0.30$)	0.8984	83.79	0.9081	0.0093	0.2264	0.2785
CCFO ($\beta = 0.42$)	0.9066	52.16	0.9171	-0.2136	0.2139	0.2831

Ablation on different regularization coefficient κ. We compare the reference-regularization coefficients $\kappa \in \{0.01, 0.04, 0.1\}$ at $\beta \in \{0.15, 0.25, 0.30\}$. For this ablation, we evaluate checkpoints every ten optimizer updates up to 60 updates. Fig. 9 shows their evolution during training. We observe that the OCR increases in all the settings. Larger values of κ lead to slower reward improvement. Large κ (0.1) slows OCR improvement without yielding higher final feasibility. In both $\kappa = 0.01, 0.04$ settings, the performance is more stable, and the target feasibility is first achieved at around 10 optimizer updates.

Fig. 10 compares the resulting reward–feasibility trade-offs. We observe again that large values of β lead to large OCR scores and small feasibility in all three settings, and the setting of $\kappa = 0.01$ achieves the best trade-offs.

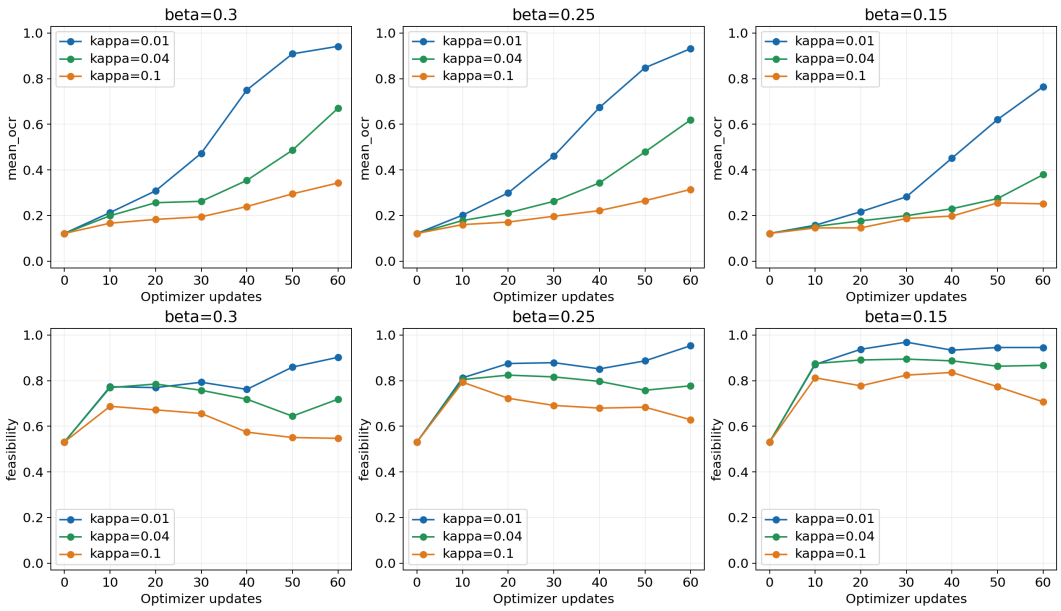


Figure 9: Training trajectories under different regularization coefficients κ. The top and bottom rows show mean OCR and aesthetic feasibility, respectively. Columns correspond to $\beta = 0.30$, 0.25, and 0.15.

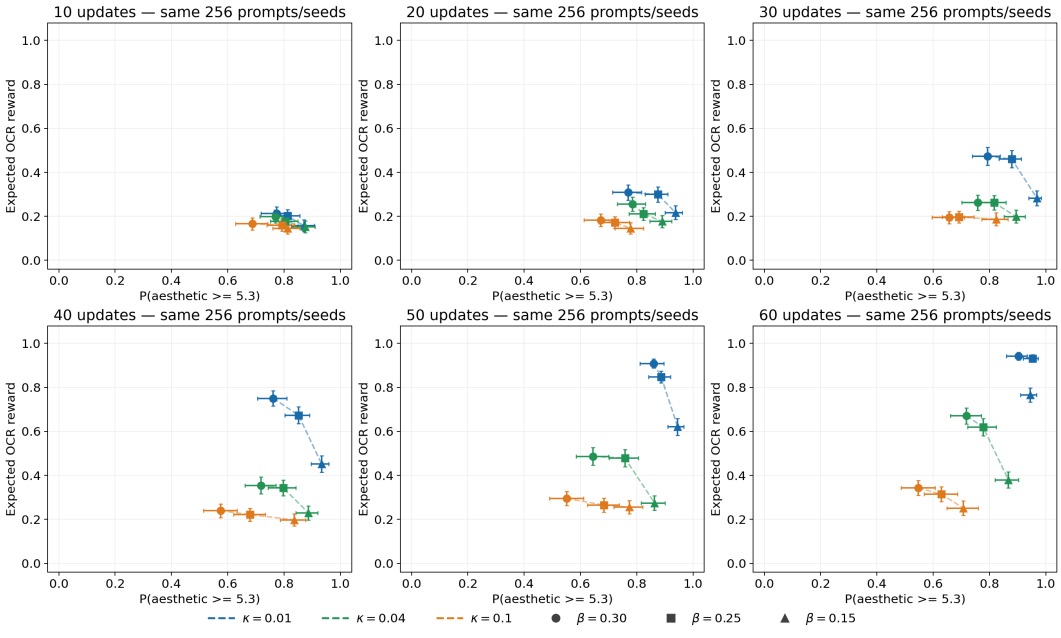


Figure 10: Reward–feasibility trade-offs across training checkpoints for different β and κ.

C.3 PROTEIN DESIGN

C.3.1 IMPLEMENTATION DETAILS

Experimental setup. We study constrained protein sequence optimization on the TrpB benchmark using the pretrained model and fitness oracle released with SGPO (Yang et al., 2026). The generative backbone is the Continuous-ESM diffusion model, which operates in a continuous embedding space and generates sequences of length 389. Fitness and mutation distance are evaluated on the 15 designated residue positions $\mathcal{S} = \{117, 118, 119, 162, 166, 182, 183, 184, 185, 186, 227, 228, 230, 231, 301\}$, using one-based indexing. The reward $f(x)$ is the predicted fitness from the released ensemble of 20 multilayer perceptrons applied to one-hot encodings of these residues. We average the nonnegative predictions of the ensemble members and disable the additional Hamming-distance penalty. All reported fitness values are oracle predictions, rather than experimental measurements. For the main results, we use the KL regularization $\alpha = 0.5$.

Let x^{WT} denote the reference sequence of the wild-type protein and define

$$d(x, x^{\mathrm{WT}}) = \sum_{j \in \mathcal{S}} \mathbf{1}\{x_j \neq x_j^{\mathrm{WT}}\}. \tag{42}$$

We consider the feasibility event $F(x) = \{d(x, x^{\mathrm{WT}}) \geq 8\}$ and seek to maximize expected fitness subject to $\mathbb{P}_{p_\theta}(F) \geq q$. Oracle-invalid outputs are assigned zero reward and treated as infeasible. The constraint encourages sequence-level departure from the wild-type protein at the designated sites.

Reward-weighted fine-tuning. We implement fine-tuning through reward-weighted endpoint denoising as in (Lee et al., 2023). A shared offline reference pool contains 2,048 samples from the pretrained generator, with an additional 512 independent reference samples used for support diagnostics. For a method-specific utility R, we construct globally normalized weights

$$w_i(R) = \frac{\exp(R(x_i)/\alpha)}{\sum_{j=1}^{N} \exp(R(x_j)/\alpha)}, \qquad N = 2,048. \tag{43}$$

These weights approximate the exponential tilt $p_R(x) \propto p^{\mathrm{pre}}(x) \exp(R(x)/\alpha)$.

At each optimization step, we sample reference endpoints with replacement according to these weights. Given an endpoint z_i, we draw $t \sim \mathcal{U}(0, 1)$ and $\xi \sim \mathcal{N}(0, I)$, construct

$$z_t = \sin\left(\frac{\pi t}{2}\right) z_i + \cos\left(\frac{\pi t}{2}\right) \xi, \tag{44}$$

and train the endpoint predictor $D_\theta(z_t, t)$ to reconstruct z_i using a mean-squared error loss. Consequently, neither the fitness oracle nor the discrete constraint needs to be differentiable.

Methods and constraint calibration. Gated reward fine-tuning and CFO use

$$R_{\mathrm{gated}}(x) = f(x)\mathbf{1}_F(x), \qquad R_{\mathrm{CFO}}(x) = f(x) + \mu\big(d(x, x^{\mathrm{WT}}) - 8\big), \tag{45}$$

respectively. Their target weights remain fixed throughout training. For CFO, we sweep $\mu \in \{0.189108, 0.241837, 0.306832, 0.404155\}$. These coefficients are calibrated on the reference pool to cover a range of ideal feasibility levels.

The hard-constraint CCFO utility is $R_{\mathrm{CCFO}}(x) = f(x) + \lambda \mathbf{1}_F(x)$. We initialize λ using the estimated optimal multiplier. To compensate for imperfect distribution fitting, we subsequently update it using separate batches of 256 generated samples every 40 optimization steps:

$$\lambda_{k+1} = \Pi_{[0,4]}\left[\lambda_k + \mathrm{clip}\left(\eta_k(q - \widehat{p}_k), -0.4, 0.4\right)\right], \tag{46}$$

where $\widehat{p}_k$ is the estimated hard feasibility.

For the higher-target CCFO settings, we additionally use the constraint-shaping feature

$$h_{\epsilon,\tau}(x) = (1 - \epsilon)\mathbf{1}_F(x) + \epsilon\, \sigma\left(\frac{d(x, x^{\mathrm{WT}}) - 7.5}{\tau}\right), \tag{47}$$

with both terms set to zero for oracle-invalid outputs, and replace the training utility by $f(x) + \lambda h_{\epsilon,\tau}(x)$. Feedback and evaluation always use the original hard event. This is a practical skill used to handle the sparse training signal induced by the indicator function.

The core CCFO sweep uses $q \in \{0.6, 0.7, 0.8, 0.9\}$. We set $\epsilon = 1.25(q - 0.5)$.

Training details. All models are initialized from the same original pretrained checkpoint and trained for 1,000 AdamW updates with learning rate 5×10^{-6}, zero weight decay, and gradient-norm clipping at 1. Generation uses 50 ancestral denoising steps with a cosine schedule, followed by the full frozen pretrained terminal decoder and deterministic tokenwise argmax decoding. Training is performed on a single NVIDIA RTX 5090 GPU, and the final checkpoint is used for evaluation.

Evaluation metrics. The core Pareto comparison aggregates final-checkpoint results over three training seeds per setting and reports the mean and sample standard deviation across seeds. Its horizontal axis is the realized feasibility, and its vertical axis is mean raw fitness over all generated samples. For detailed evaluation of representative checkpoints, we generate 4,096 fresh samples per model. We report $\mathbb{E}[f(X)]$, $\mathbb{P}(F)$, $\mathbb{E}[f(X) \mid F]$, and $\mathbb{P}(F, f(X) > c)$. Diversity is measured by the average sitewise Shannon entropy over the 15 designated positions:

$$H(\mathcal{D}) = -\frac{1}{15} \sum_{j \in \mathcal{S}} \sum_{a \in \mathcal{A}} \widehat{p}_j(a) \log_2 \widehat{p}_j(a), \tag{48}$$

where $\mathcal{A}$ is the canonical amino-acid alphabet. We also report the number of distinct canonical 15-residue designs satisfying both feasibility and $f(x) > 1$.

C.3.2 Additional results

Ablation on different $q = 1 - \beta$. Table 6 reports the performance of CCFO under different target feasibility levels q. CCFO consistently achieves the desired feasibility across $q \in \{0.6, 0.7, 0.8, 0.9\}$. As q increases, the optimization places greater emphasis on constraint satisfaction, leading to a gradual decrease in overall mean fitness. Interestingly, the mean fitness among feasible samples remains relatively stable for $q = 0.6, 0.7, 0.8$, but decreases at $q = 0.9$, suggesting that enforcing an excessively high feasibility level can compromise the quality of feasible designs. Despite this decrease, CCFO still achieves higher fitness than CFO at a comparable feasibility level. Moreover, CCFO produces more distinct designs that are both feasible and high-fitness.

Table 6: Protein design performance with $\alpha = 0.5$ on protein sequence generation averaged over 3 seeds.

Method / checkpoint	Mean fitness (all)	Feasibility (%)	Mean fitness (given feasible)	Entropy (all, bits/site)	Entropy (given feasible, bits/site)	Distinct designs $F \cap \{f > 1\}$
CFO μ_1	0.407	50.69	0.342	1.645	1.778	73
CFO μ_2	0.368	63.54	0.327	1.705	1.783	85
CFO μ_3	0.331	79.57	0.312	1.763	1.802	84
CFO μ_4	0.266	89.50	0.259	1.848	1.866	65
CCFO $q = 0.6$	0.411	66.96	0.402	1.608	1.677	129
CCFO $q = 0.7$	0.406	71.48	0.399	1.592	1.653	136
CCFO $q = 0.8$	0.392	82.30	0.384	1.657	1.703	142
CCFO $q = 0.9$	0.301	90.01	0.297	1.717	1.741	88

Ablation on different KL-regularization parameter α. Table 7 shows how α influences the performance. We observe that CCFO achieves the target feasibility $q = 0.7$ across three settings of $\alpha = \{0.2, 0.5, 1.0\}$. A small value of α leads to larger fitness values, with, however, the diversity decreased compared to the pre-trained model. This is also common for the KL-regularized fine-tuning problem. At the intermediate level $\alpha = 0.5$, it improves the feasibility and fitness compared to the pre-trained model, while maintaining the diversity at the same level.

Table 7: Effect of α on CCFO for TrpB design with target feasibility $q = 0.7$. Each model is trained for 1,000 steps with seed 0 and evaluated using 4,096 samples.

α	Mean fitness (all)	Feasibility (%)	Mean fitness (given feasible)	Entropy (all, bits/site)	Entropy (given feasible, bits/site)	Distinct designs $F \cap \{f > 1\}$
0.2	0.843	75.81	0.855	1.154	1.184	894
0.5	0.425	70.07	0.418	1.631	1.689	146
1.0	0.299	77.29	0.293	1.710	1.764	79

Figure 11 plots the training curves of three settings with the target feasibility $q = 0.7$. We observe that feasibility gets to the target $q = 0.7$ at around 240 optimizer steps for all settings of α, and then oscillate around the target feasibility. For the fitness, the value of α plays a big role. A large value of α can make fitness hardly improved since it enforce a strong penalty of moving far from the pre-trained model. A similar pattern is found for the target feasibility $q = 0.8$ in Figure 12.

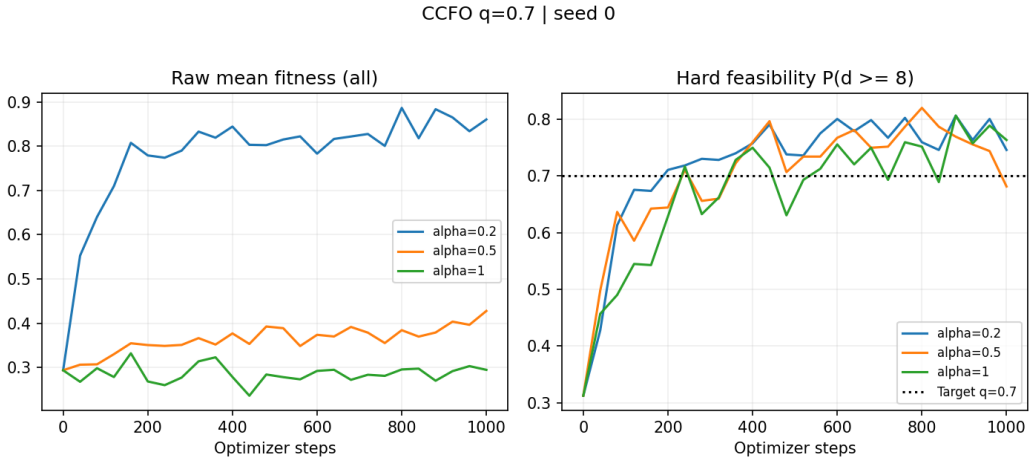


Figure 11: Training curves for different values of α with target feasibility $q = 0.7$.

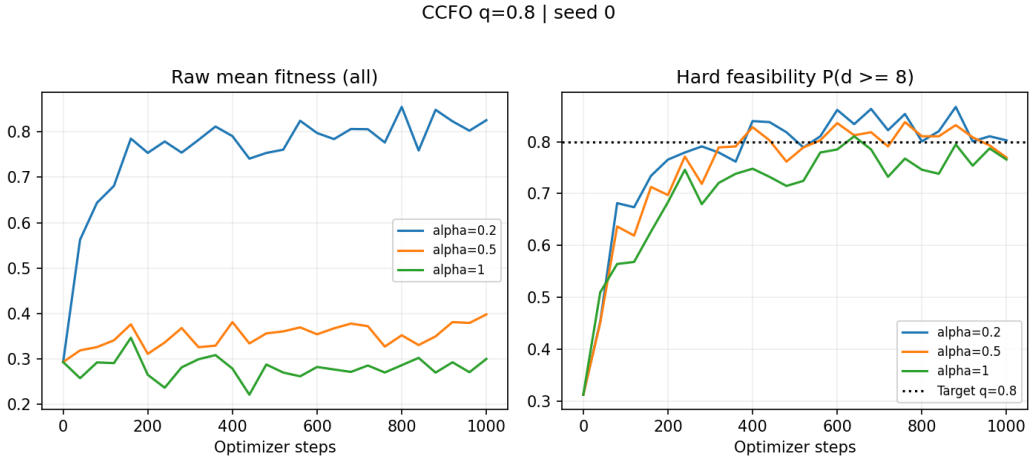


Figure 12: Training curves for different values of α with target feasibility $q = 0.8$.

D TEXT-TO-IMAGE SAMPLES



Figure 13: Generated samples by the pre-trained model, CFO and CCFO.

E VISUALIZATION OF GENERATED PROTEIN SEQUENCES

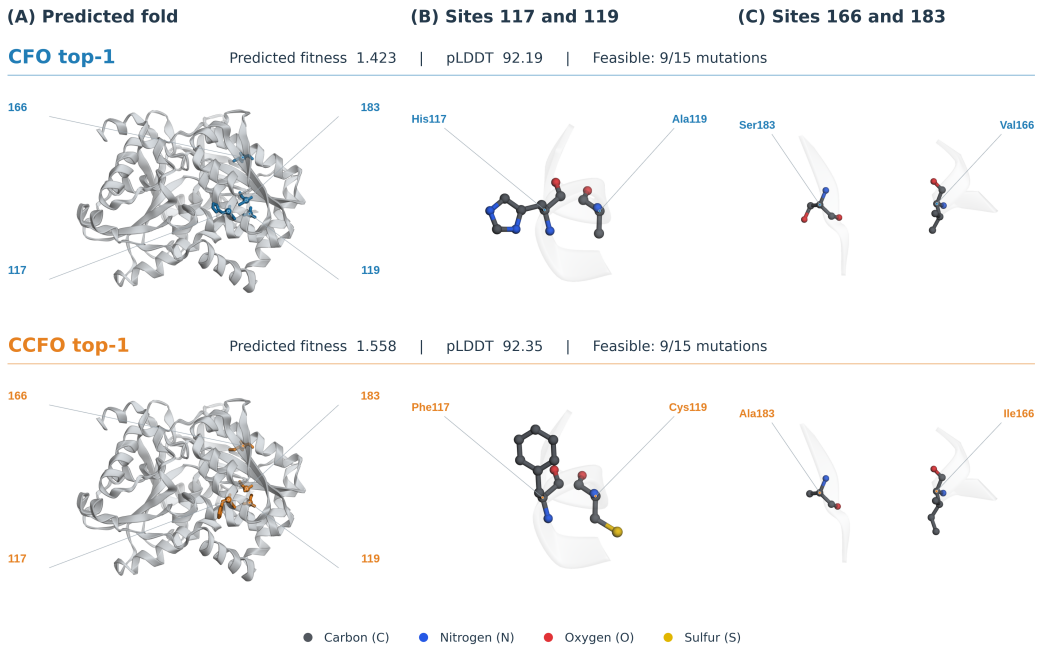


Figure 14: Structural visualization of the highest-fitness feasible TrpB design from CFO and CCFO. (A) Overall predicted folds, highlighting the four positions at which the two designs differ. (B) Detailed views of positions 117 and 119. (C) Detailed views of positions 166 and 183.

Feasible TrpB sequence designs

Design	117	118	119	162	166	182	183	184	185	186	227	228	230	231	301	Predicted fitness	Mutations to WT	Occurrences in 4,096 draws
WT	T	A	A	L	I	Y	V	F	G	S	V	S	G	S	Y	—	0/15	—
CFO-1	H	V	A	M	V	Y	S	L	G	S	L	G	G	G	Y	1.423	9/15	1
CFO-2 †	F	V	A	M	I	F	A	L	G	S	M	G	G	G	Y	1.368	9/15	1
CFO-3	F	A	C	M	I	Y	I	L	G	S	L	G	G	G	Y	1.364	8/15	1
CFO-4	Y	V	A	D	I	Y	A	L	G	S	M	G	G	G	Y	1.315	8/15	1
CFO-5	L	V	A	M	I	Y	A	L	G	S	I	G	G	G	Y	1.307	8/15	1
CCFO-1	F	V	C	M	I	Y	A	L	G	S	L	G	G	G	Y	1.558	9/15	1
CCFO-2	F	V	A	M	I	F	I	L	G	S	L	G	G	G	Y	1.392	9/15	1
CCFO-3	V	V	A	M	V	Y	S	L	G	S	L	G	G	G	Y	1.392	9/15	1
CCFO-4	F	V	A	M	V	F	A	L	G	S	I	G	G	G	Y	1.371	10/15	1
CCFO-5 †	F	V	A	M	I	F	A	L	G	S	M	G	G	G	Y	1.368	9/15	1

Same as WT | CFO mutation | CCFO mutation | † Same design in both methods

Figure 15: Top-5 feasible TrpB sequence designs generated by CFO and CCFO. Blue and orange cells indicate substitutions relative to WT for CFO and CCFO, respectively; gray cells indicate unchanged residues. The right columns report oracle-predicted fitness, mutation count, and the number of occurrences in the corresponding sample set. A dagger (†) marks a design shared by both methods.