Flow Density Control: Generative Optimization Beyond Entropy-Regularized Fine-Tuning

Riccardo De Santi
ETH Zurich
ETH AI Center
rdesanti@ethz.ch

Marin Vlastelica
ETH Zurich
ETH AI Center
marin.vlastelica@inf.ethz.ch

Ya-Ping Hsieh
ETH Zurich
yaping.hsieh@inf.ethz.ch

Zebang Shen
ETH Zurich
zebang.shen@inf.ethz.ch

Niao He
ETH Zurich
ETH AI Center
niaohe@ethz.ch

Andreas Krause
ETH Zurich
ETH AI Center
krausea@ethz.ch

Abstract

Adapting large-scale foundational flow and diffusion generative models to optimize task-specific objectives while preserving prior information is crucial for real-world applications such as molecular design, protein docking, and creative image generation. Existing principled fine-tuning methods aim to maximize the expected reward of generated samples, while retaining knowledge from the pre-trained model via KL-divergence regularization. In this work, we tackle the significantly more general problem of optimizing general utilities beyond average rewards, including risk-averse and novelty-seeking reward maximization, diversity measures for exploration, and experiment design objectives among others. Likewise, we consider more general ways to preserve prior information beyond KL-divergence, such as optimal transport distances and Rényi divergences. To this end, we introduce **F**low **D**ensity **C**ontrol (FDC), a simple algorithm that reduces this complex problem to a specific sequence of simpler fine-tuning tasks, each solvable via scalable established methods. We derive convergence guarantees for the proposed scheme under realistic assumptions by leveraging recent understanding of mirror flows. Finally, we validate our method on illustrative settings, text-to-image, and molecular design tasks, showing that it can steer pre-trained generative models to optimize objectives and solve practically relevant tasks beyond the reach of current fine-tuning schemes.

1 Introduction

Large-scale generative modeling has recently seen remarkable advancements, with flow [30, 31] and diffusion models [52, 53, 23] standing out for their ability to produce high-fidelity samples across a wide range of applications, from chemistry [24] and biology [9] to robotics [8]. However, approximating the data distribution is insufficient for real-world applications such as scientific discovery [6, 60], where one typically wishes to generate samples optimizing specific utilities, e.g., molecular stability and diversity, while preserving certain information from a pre-trained model. This problem has recently been tackled via fine-tuning in the case where the utility corresponds

	Current		This work
Utility	$\mathbb{E}_{x\sim p_1^\pi}[r(x)]$	$\rightarrow$	$\mathcal{F}(p_1^\pi)$
Divergence	$D_{KL}(p_1^\pi \| p_1^{pre})$	$\rightarrow$	$\mathcal{D}(p_1^\pi \| p_1^{pre})$

Figure 1: We extend the capabilities of current fine-tuning schemes from KL-regularized expected reward maximization (left) to the optimization of arbitrary distributional utilities $\mathcal{F}$ under general divergences $\mathcal{D}$ (right).

First Exploration in AI Today Workshop at ICML (EXAIT at ICML 2025).

to the expected reward of generated samples, and pre-trained model information is retained via KL-divergence regularization, as shown in Fig. 1 (left). Crucially, this specific fine-tuning problem can be solved via entropy-regularized control formulations [e.g., 14, 56, 54] with successful applications in real-world domains such as image generation [14], molecular design [57], or protein engineering [57].

Unfortunately, many practically relevant tasks cannot be captured by this formulation. For instance, consider the tasks of *risk-averse* and *novelty-seeking* reward maximization. In the former case, one wishes to steer the generative model toward distributions with controlled worst-case rewards, thereby improving validity and safety. In the latter case, one aims to control the upper tail of the reward distribution to maximize the probability of generating exceptionally promising designs, e.g., for scientific discovery. Other applications that cannot be captured via maximization of simple expectations include manifold exploration [12], model de-biasing [13], and optimal experimental design [39, 10] among others. Similarly, preserving prior information via a KL divergence has known drawbacks. For instance, it can lead to missing of low-probability yet valuable modes [29, 44], and it prevents from leveraging the geometry of the space even when this is known, e.g., in protein docking [9]. Replacing KL with alternative divergences can address these shortcomings. Driven by these motivations, in this work we aim to answer the following fundamental question (see Fig. 1):

How can we provably fine-tune a flow or diffusion model to optimize any user-specified utility while preserving prior information via an arbitrary divergence?

Answering this would contribute to the algorithmic-theoretical foundations of *generative optimization*.

Our approach We tackle this challenge by first introducing the formal problem of *generative optimization via fine-tuning*. Then, we shed light on why this formulation is strictly more expressive than current fine-tuning problems [14, 54], and present a sample of novel practically relevant utilities and divergences (Sec. 3). Next, we introduce **F**low **D**ensity **C**ontrol (FDC), a simple sequential scheme that can fine-tune models to optimize general objectives beyond the reach of entropy-regularized control methods. This is achieved by leveraging recent machinery from Convex [20] and General Utilities RL [61] (Sec. 4). We provide rigorous convergence guarantees for the proposed algorithm in both a simplified scenario, via convex optimization analysis [43, 33], and in a realistic setting, by building on recent understanding of mirror flows [25] (Sec. 5). Finally, we provide an experimental evaluation of the proposed method, demonstrating its practical relevance on both synthetic and high-dimensional image and molecular generation tasks, showing how it can steer pre-trained models to solve tasks beyond the inherent limits of current fine-tuning schemes (Sec. 6).

Our contributions To sum up, in this work we contribute

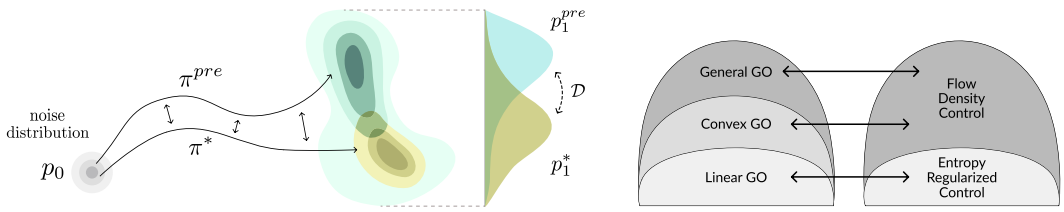
- A formalization of the *generative optimization* problem, which extends current fine-tuning formulations beyond linear utilities and general divergences (Sec. 3).
- ***F**low **D**ensity **C**ontrol (*FDC*)*, a principled algorithm capable of optimizing functionals beyond the reach of current fine-tuning schemes based on entropy-regularized control/RL (Sec. 4).
- Convergence guarantees for the presented algorithm both under simplified and realistic assumptions leveraging recent understanding of mirror flows (Sec. 5).
- An experimental evaluation of FDC showcasing its practical relevance on both illustrative and high-dimensional text-to-image and molecular design tasks, showing how it can steer pre-trained models to solve tasks beyond the capabilities of current fine-tuning schemes. (Sec. 6).

2 Background and Notation

General Notation. We denote with $\mathcal{X} \subseteq \mathbb{R}^d$ an arbitrary set. Then, we indicate the set of Borel probability measures on $\mathcal{X}$ with $\mathbb{P}(\mathcal{X})$, and the set of functionals over the set of probability measures $\mathbb{P}(\mathcal{X})$ as $\mathbb{F}(\mathcal{X})$. Given an integer N, we define $[N] \coloneqq \{1, \dots, N\}$.

Generative Flow Models. Generative models aim to approximately sample novel data points from a data distribution p_{data}. Flow models tackle this problem by transforming samples $X_0 = x_0$ from a source distribution p_0 into samples $X_1 = x_1$ from the target distribution p_{data}[31, 17]. Formally, a *flow* is a time-dependent map $\psi : [0, 1] \times \mathbb{R}^d \to \mathbb{R}$ such that $\psi : (t, x) \to \psi_t(x)$. A *generative flow model* is a continuous-time Markov process $\{X_t\}_{0 \leq t \leq 1}$ obtained by applying a flow ψ_t to $X_0 \sim p_0$ as $X_t = \psi_t(X_0)$, $t \in [0, 1]$, such that $X_1 = \psi_1(X_0) \sim p_{data}$. In particular, the flow ψ can be defined by a *velocity field* $u : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$, which is a vector field related to ψ via the following ordinary differential equation (ODE), typically referred to as *flow ODE*:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = u_t(\psi_t(x)) \tag{1}$$



(a) Generative Optimization via Flow Model Fine-tuning.

(b) GO expressivity hierarchy

Figure 2: (2a) Pre-trained and fine-tuned policies inducing densities p_1^{pre} and optimal density p_1^* w.r.t. utility $\mathcal{F}$ and divergence $\mathcal{D}$. (2b) Expressivity and control hierarchy for generative optimization.

with initial condition $\psi_0(x) = 0$. A flow model $X_t = \psi_t(X_0)$ induces a probability path of *marginal densities* $p = \{p_t\}_{0 \leq t \leq 1}$ such that at time t we have that $X_t \sim p_t$. Given a velocity field u and marginal densities p, we say that u generates the marginal densities $p = \{p_t\}_{0 \leq t \leq 1}$ if $X_t = \psi_t(X_0) \sim p_t$ for all $t \in [0, 1)$. This is the case if the pair (u, p) satisfy the *Continuity Equation*:

$$\frac{\mathrm{d}}{\mathrm{d}t} p_t(x) + \mathrm{div}(p_t u_t)(x) = 0 \tag{2}$$

In this case, we denote by p^u the probability path of marginal densities induced by the velocity field u. Flow matching [30, 32, 1, 31] can estimate a velocity field u^θ s.t. the induced marginal densities p^{u_θ} satisfy $p_0^{u_\theta} = p_0$ and $p_1^{u_\theta} = p_{data}$, where p_0 denotes the source distribution, and p_{data} the target data distribution. Interestingly, diffusion models [53] (DMs) admit an equivalent ODE-based formulation with identical marginal densities to their original SDE dynamics [31, Chapter 10]. Consequently, although in this work we adopt the notation of flow models, our contributions carry over directly to DMs.

Continuous-time Reinforcement Learning. We formulate finite-horizon continuous-time reinforcement learning (RL) as a specific class of optimal control problems [58, 26, 55, 62]. Given a state space $\mathcal{X}$ and an action space $\mathcal{A}$, we consider the transition dynamics governed by the following ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t} \psi_t(x) = a_t(\psi_t(x)) \tag{3}$$

where $a_t \in \mathcal{A}$ is a selected action. We consider a state space $\mathcal{X} := \mathbb{R}^d \times [0, 1]$, and denote by (Markovian) deterministic policy a function $\pi_t(X_t) := \pi(X_t, t) \in \mathcal{A}$ mapping a state $(x, t) \in \mathcal{X}$ to an action $a \in \mathcal{A}$ such that $a_t = \pi(X_t, t)$, and denote with p_t^π the marginal density at time t induced by policy π.

Pre-trained Flow Models as an RL policy. A pre-trained flow model with velocity field u^{pre} can be interpreted as an action process $a_t^{pre} := u^{pre}(X_t, t)$, where a_t^{pre} is determined by a continuous-time RL policy via $a_t^{pre} = \pi^{pre}(X_t, t)$ [12]. Therefore, we can express the flow ODE induced by a pre-trained flow model by replacing a_t with a^{pre} in Eq. (3), and denote the pre-trained model by its (implicit) policy π^{pre}, which induces a marginal density $p_1^{pre} := p_1^{\pi^{pre}}$ approximating p_{data}.

We present a thorough analysis of related works in Apx. A.

3 Formal Problem: a General Framework for Generative Optimization

In this section, we aim to formally introduce the general problem of generative optimization (GO) via fine-tuning. Formally, we wish to adapt a pre-trained generative flow model π^{pre} to obtain a new model π^* inducing an ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t} \psi_t(x) = a_t^*(\psi_t(x)) \quad \text{with} \quad a_t^* = \pi^*(x, t), \tag{4}$$

such that instead of imitating the data distribution p_{data}, as typically in generative modeling, it induces a marginal density $p_1^{\pi^*}$ that maximizes a utility measure $\mathcal{F} : \mathbb{P}(\mathcal{X}) \to \mathbb{R}$, while preserving information from the pre-trained model π^{pre} via regularization with an arbitrary divergence $\mathcal{D}(\cdot \,\|\, p^{pre})$. This algorithmic problem is illustrated in Fig. 2a, and formalized in the following.

Generative Optimization via Flow Model Fine-Tuning

$$\arg\max_{\pi} \quad \mathcal{F}(p_1^\pi) - \alpha \mathcal{D}(p_1^\pi \,\|\, p_1^{pre}) \text{ s.t.} \frac{\mathrm{d}}{\mathrm{d}t} p_t(x) + \mathrm{div}(p_t a_t)(x) = 0 \text{ with } a_t = \pi(x, t) \tag{5}$$

Application	Functional $\mathcal{F}$ / $\mathcal{D}$	Linear GO	Non-Linear GO	
			Concave	General
Reward optimization [14, 56]	$\mathbb{E}_{x\sim p^\pi}[r(x)]$	✓	✓	✓
Manifold Exploration [12] Gen. model de-biasing	$\mathcal{H}(p^\pi) := -\mathop{\mathbb{E}}_{x\sim p^\pi}[\log p^\pi(x)]$	✗	✓	✓
Risk-averse optimization	$\mathrm{CVaR}^r_\beta(p^\pi) := \mathop{\mathbb{E}}_{x\sim p^\pi}[r(x) \mid r(x) \le \mathrm{q}^r_\beta(p^\pi)]$	✗	✓	✓
	$\mathbb{E}_{x\sim p^\pi}[r(x)] - \mathbb{V}\mathrm{ar}(p^\pi)$	✗	✗	✓
Novelty-seeking optimization	$\mathrm{SQ}^r_\beta(p^\pi) := \mathop{\mathbb{E}}_{x\sim p^\pi}[r(x) \mid r(x) \ge \mathrm{q}^r_\beta(p^\pi)]$	✗	✗	✓
Optimal Experiment Design	$\mathrm{s}\left(\mathop{\mathbb{E}}_{x\sim p^\pi}[\Phi(x)\Phi(x)^\top - \lambda \mathbb{I}]\right)$ $\mathrm{s}(\cdot) \in \{\log\det(\cdot), -\mathrm{Tr}(\cdot)^{-1}, -\lambda_{max}(\cdot)\}$	✗	✓	✓
Diverse modes discovery	$-\mathop{\mathbb{E}}_{z}[D_{KL}(p^{\pi,z} \Vert \mathop{\mathbb{E}}_{k} p^{\pi,k})]$	✗	✗	✓
Log-Barrier Constrained Generation	$\mathbb{E}_{x\sim p^\pi}[r(x)] - \beta \log(\langle p^\pi, c\rangle - C)$	✗	✓	✓
Kullback–Leibler divergence [14, 56]	$D_{KL}(p^\pi \Vert p^{pre}) = \int p^\pi(x) \log \frac{p^\pi(x)}{p^{pre}(x)}\, dx$	✓	✓	✓
Rényi divergences	$D_\beta(p^\pi \Vert p^{pre}) := \frac{1}{\beta - 1} \log \int (p^\pi(x))^\beta (p^{pre})^{1-\beta}\, dx$	✗	✗	✓
Optimal Transport distances	$W_p(p^\pi \Vert p^{pre}) := \inf_{\gamma\in\Gamma(p^\pi, p^{pre})} \mathop{\mathbb{E}}_{(x,y)\sim\gamma}[d(x,y)^p]^{\frac{1}{p}}$	✗	✗	✓
Maximum Mean Discrepancy	$\mathrm{MMD}_k(p^\pi \Vert p^{pre}) := \Vert \mu_{p^\pi} - \mu_{p^{pre}} \Vert, \mu_p := \mathop{\mathbb{E}}_{x\sim p}[k(x,\cdot)]$	✗	✓	✓

Table 1: Examples of practically relevant utilities $\mathcal{F}$ (blue) and divergences $\mathcal{D}$ (orange). Apx. B provides mathematical details and practical applications for each functional. Notice that besides $\mathcal{H}$, all non-linear functionals presented are novel in the context of fine-tuning of diffusion and flow models.

In this formulation, $\mathcal{F}$ and $\mathcal{D}$ are both functionals mapping the marginal density p_1^π induced by policy π to a scalar real number, namely $\mathcal{F}, \mathcal{D} : \mathbb{P}(\mathcal{X}) \to \mathbb{R}$. The constraint in Eq. (5) is the (*controlled*) Continuity Equation (see Eq. (2)), which relates the control policy π to the induced marginal density p_1^π.

3.1 The sub-case of KL-regularized reward maximization via entropy-regularized control

Current fine-tuning schemes for flow generative models based on RL and control-theoretic formulations [e.g., 14, 56] aim to tackle the following problem, where we omit the flow constraint for clarity:

Linear Generative Optimization via Flow Model Fine-Tuning

$$\arg\max_{\pi} \quad \mathop{\mathbb{E}}_{x\sim p_1^\pi}[r(x)] - \alpha D_{KL}(p_1^\pi \,\Vert\, p_1^{pre}) \tag{6}$$

Crucially, the common problem in Eq. (6), which we denote by *Linear*[1] GO, is the specific sub-case of the generative optimization problem in Eq. (5), where the utility $\mathcal{F}$ is a linear functional corresponding to the expectation of a (reward) function $r : \mathcal{X} \to \mathbb{R}$, and $\mathcal{D}$ is the Kullback–Leibler divergence:

$$\mathcal{F}(p_1^\pi) = \langle p_1^\pi, r\rangle = \mathop{\mathbb{E}}_{x\sim p_1^\pi}[r(x)] \quad \text{and} \quad \mathcal{D}(p_1^\pi \,\Vert\, p_1^{pre}) = D_{KL}(p_1^\pi \,\Vert\, p_1^{pre}) \tag{7}$$

This specific fine-tuning problem can be solved via entropy-regularized (or relaxed) control [14].

3.2 Beyond Linear Generative Optimization

Let $\mathcal{G}(p_1^\pi) = \mathcal{F}(p_1^\pi) - \alpha\, \mathcal{D}(p_1^\pi \,\Vert\, p_1^{\mathrm{pre}})$ be the functional in Eq. (5). Then we denote by *Convex* GO the case where $\mathcal{G}$ is concave in p_1^π, and by *General* GO the case for arbitrary, possibly non-convex functionals [2]. In terms of expressivity **Linear GO** $\subset$ **Convex GO** $\subset$ **General GO**, as depicted in Fig. 2b (left). In Table 1 we classify into these tree tiers a sample of practically relevant utilities ($\mathcal{F}$, blue) and divergences ($\mathcal{D}$, orange). In Apx. B we report complete definitions and applications. Except for entropy [12] and KL, all non-linear functionals in Table 1 are to our knowledge explicitly used for the first time in the flow and diffusion model fine-tuning literature, while vastly employed in other areas. Moreover, the framework presented in this work for GO (Eq. 5) applies to any new choice of $\mathcal{F}$ or $\mathcal{D}$.

[1] For clarity, we adopt the term *linear* motivated by the linear utility even though the KL is non-linear.

[2] We use the term *convex* GO, rather than concave GO, to denote the problem class with concave functionals.

Algorithm 1 **F**low **D**ensity **C**ontrol (FDC)

1: **input:** $\mathcal{G}$: general utility functional, K : number of iterations, π^{pre} : pre-trained flow generative model, $\{\eta_k\}_{k=1}^K$ regularization coefficients
2: **Init:** $\pi_0 := \pi^{pre}$
3: **for** $k = 1, 2, \ldots, K$ **do**
4: Estimate: $\nabla_x g_k = \nabla_x \delta\mathcal{G}(p_1^{k-1})$
5: Compute π_k via first-order linear fine-tuning:

$$\pi_k \leftarrow \textsc{EntropyRegularizedControlSolver}(\nabla_x g_k, \eta_k, \pi_{k-1})$$

6: **end for**
7: **output:** policy $\pi := \pi_K$

Given the generality of generative optimization (Eq.(5)), a natural question arises: how can it be solved algorithmically? In the next section, we answer this by leveraging recent machinery from Convex [20] and General-Utilities RL [61], to derive a fine-tuning scheme that handles both convex and general GO, thus going beyond current entropy-regularized control methods, as illustrated in Fig. 2b (right).

4 Algorithm: Flow Density Control

In this section, we introduce **F**low **D**ensity **C**ontrol (FDC), see Alg. 1, which provably solves the generative optimization problem in Eq. (5) via sequential fine-tuning of the pre-trained model π^{pre}. To this end, we recall the notion of first variation of a functional over a space of probability measures [25]. A functional $\mathcal{G} \in \mathbb{F}(\mathcal{X})$, where $\mathcal{G} : \mathbb{P}(\mathcal{X}) \to \mathbb{R}$, has first variation at $\mu \in \mathbb{P}(\mathcal{X})$ if there exists a function $\delta\mathcal{G}(\mu) \in \mathbb{F}(\mathcal{X})$ such that for all $\mu' \in \mathbb{P}(\mathcal{X})$ it holds that:

$$\mathcal{G}(\mu + \epsilon\mu') = \mathcal{G}(\mu) + \epsilon\langle \mu', \delta\mathcal{G}(\mu)\rangle + o(\epsilon).$$

where the inner product has to be interpreted as an expectation. Intuitively, the first variation of $\mathcal{G}$ at μ, namely $\delta\mathcal{G}(\mu)$, can be interpreted as an infinite-dimensional gradient in the space of probability measures. Given this notion, and a pair of generative models represented via policies π and π', we can now state the following *entropy-regularized first variation maximization* fine-tuning problem.

Entropy-Regularized First Variation Maximization

$$\arg\max_{\pi} \quad \langle \delta\mathcal{G}\left(p_1^{\pi'}\right), p_1^{\pi}\rangle - \eta D_{KL}(p_1^{\pi} \,\|\, p_1^{\pi'}) \tag{8}$$

Crucially, we can introduce a function $g : \mathcal{X} \to \mathbb{R}$ defined for all $x \in \mathcal{X}$ such that:

$$g(x) := \delta\mathcal{G}\left(p_1^{\pi'}\right)(x) \quad \text{and} \quad \mathop{\mathbb{E}}_{x\sim p^{\pi}}[g(x)] = \langle \delta\mathcal{G}\left(p_1^{\pi'}\right), p_1^{\pi}\rangle \tag{9}$$

As a consequence, by rewriting Eq. (8) expressing the first term via an expectation as shown in Eq. (9), it corresponds to a common Linear GO problem (see Eq. (6)), which can be optimized by utilizing established entropy-regularized control methods [e.g., 57, 14, 62].

We can finally present **F**low **D**ensity **C**ontrol (FDC), see Alg. 1, a mirror descent (MD) scheme [43] that reduces optimization of non-linear functionals $\mathcal{G}$ to a specific sequence of Linear GO problems. FDC takes three inputs: a pre-trained flow or diffusion model π^{pre}, the number of iterations K, and a sequence of regularization weights $\{\eta_k\}_{k=1}^K$. At each iteration, FDC first estimates the gradient of the functional first variation at the previous policy π_{k-1}, i.e., $\nabla_x \delta\mathcal{G}\left(p_1^{k-1}\right)$ (line 4). Then, it updates the flow model π_k by solving the fine-tuning problem in Eq. (8) via an entropy-regularized control solver such as Adjoint Matching [14], using $\nabla_x g_k := \nabla_x \delta\mathcal{G}\left(p_1^{k-1}\right)$ as in Eq. (9) (line 5). Ultimately, it returns a final policy $\pi := \pi_K$. We report a detailed implementation of FDC in Apx. F. An introduction to computation and estimation of the gradient of first variations can be found in Apx. C.

Given the approximate gradient estimates and the generality of the objective functions, it is still unclear whether the proposed algorithm provably converges to the optimal flow model π^*. In the next section, we answer this question by developing a theoretical analysis via recent results on mirror flows [25].

5 Guarantees for Generative Optimization via Flow Density Control

In this section, we recast (5) as *constrained* optimization over stochastic processes, where the constraint is given by the Continuity Equation (2). This formulation enables the application of **mirror**

descent for constrained optimization and the notion of *relative smoothness* [3]. In our framework, convergence speed is governed by: 1. the structural complexity of the functional $\mathcal{G}$ (cf. Section 4), 2. the accuracy of the estimator g from (9), and 3. the quality of the oracle EntropyRegularizedControlSolver in Alg. 1. To handle these cases, we will analyze two representative regimes:

- **Idealized.** $\mathcal{G}$ is *concave*, and both g and EntropyRegularizedControlSolver are exact. In this setting, classical results yield sharp step-size prescriptions and fast convergence rates.
- **General.** $\mathcal{G}$ is *non-concave*, with g and the oracle subject to noise and bias. While fast convergence is generally out of reach [34, 27], convergence to a stationary point remains attainable under mild assumptions.

Theoretical analysis: Idealized setting. We now present a framework leading to convergence guarantees for FDC (i.e., Alg. 1) for *concave* functionals $\mathcal{G} \in \mathbb{F}(\mathcal{X})$. We report in Apx. D, we report background knowledge regarding L relative smoothness and l relative strong convexity of $\mathcal{G}$ w.r.t. a given functional, and recall the notion of Bregman divergence induced by a given functional.

In the following, we interpret line (6) of FDC as a step of mirror descent [43], and the KL divergence term as the Bregman divergence induced by an entropic mirror map $\mathcal{Q} = \mathcal{H}$, i.e., $D_{KL}(\mu, \nu) = D_{\mathcal{H}}(\mu \,\|\, \nu)$. We can finally state the following set of assumptions as well as the convergence guarantee for an arbitrary functional $\mathcal{G}(\cdot) = \mathcal{F}(\cdot) - \alpha\mathcal{D}(\cdot \,\|\, p^{pre}) \in \mathbb{F}(\mathcal{X})$.

Assumption 5.1 (Exact estimation and optimization)**.** *We consider the following assumptions:*

1. *Exact estimation:* $\nabla_x \delta\mathcal{G}(p_1^k)$ *is estimated exactly* $\forall k \in [K]$.
2. *The optimization problem in Eq.* (8) *is solved exactly.*

Theorem 5.1 (Convergence guarantee of Flow Density Control with concave functionals)**.** *Given Assumptions 5.1, fine-tuning a pre-trained model* π^{pre} *via* FDC *(Algorithm 1) with* $\eta_k = L$ $\forall k \in [K]$*, leads to a policy* π *inducing a marginal distribution* p_1^π *such that:*

$$\mathcal{G}(p_1^*) - \mathcal{G}(p_1^\pi) \leq \frac{L - l}{K} D_{KL}(p_1^* \,\|\, p_1^{pre}) \tag{10}$$

where $p_1^* := p_1^{\pi^*}$ *is the marginal distribution induced by the optimal policy* $\pi^* \in \arg\max_\pi \mathcal{G}(p_1^\pi) := \mathcal{F}(p_1^\pi) - \alpha\mathcal{D}(p_1^\pi \,\|\, p_1^{pre})$.

Theorem 5.1 provides a fast convergence rate under a specific step-size choice ($\eta_k = L$). However, it critically depends on Assumption 5.1, which typically does not hold in practice. To address this limitation, we now consider a more general scenario where this key assumption is relaxed.

Theoretical analysis: General setting. Recall that $p_1^k := p_1^{\pi_k}$ represents the (stochastic) density produced by the EntropyRegularizedControlSolver oracle at the k-th step of FDC, and consider the following *mirror descent* iterates, where $1/\lambda_k = \eta_k$ in Algorithm 1:

$$p_\sharp^k := \arg\max_{p \in \mathbb{P}(\Omega_{pre})} \quad \langle \delta\mathcal{G}\left(p_T^{\pi_{k-1}}\right), p\rangle - \frac{1}{\gamma_k} D_{KL}(p \,\|\, p_T^{\pi_{k-1}}) \tag{MD$_k$}$$

In realistic settings, where only noisy *and* biased approximations of (MD$_k$) are available, it is essential to quantify the deviations from the idealized iterates in (MD$_k$). To this end, denote by $\mathcal{T}_k$ the filtration up to step k, and consider the decomposition of the oracle into its *noise* and *bias* parts:

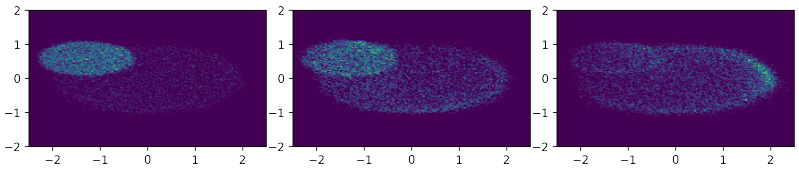
$$b_k := \mathbb{E}\left[\delta\mathcal{G}(p_T^{\pi_k}) - \delta\mathcal{G}(p_\sharp^k) \,|\, \mathcal{T}_k\right], \qquad U_k := \delta\mathcal{G}(p_T^{\pi_k}) - \delta\mathcal{G}(p_\sharp^k) - b_k \tag{11}$$

Conditioned on $\mathcal{T}_k$, U_k has zero mean, while b_k captures the *systematic* error. We then impose:

Assumption 5.2 (Noise and Bias)**.** *The following events happen almost surely:*

$$\|b_k\|_\infty \to 0, \qquad \sum_k \mathbb{E}\left[\gamma_k^2 \left(\|b_k\|_\infty^2 + \|U_k\|_\infty^2\right)\right] < \infty, \qquad \sum_k \gamma_k \|b_k\|_\infty < \infty \tag{12}$$

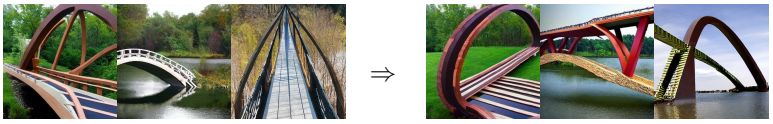
The first condition is a *necessary* requirement for convergence since when violated, it is easy to construct scenarios where no practical algorithm can solve the generative optimization problem. The second and third inequalities manage the trade-off between *accuracy* of the approximate oracle EntropyRegularizedControlSolver and *aggressiveness* of the step sizes, γ_k. Intuitively, lower noise and bias in the oracle enable the use of larger step sizes. To this end, Assumption 5.2 provides a concrete criterion that guarantees the success of finding the optimal policy with probability one.



(a) Pre-trained samples (b) FDC $\alpha = 0.5$ samples (c) FDC $\alpha = 0.0$ samples

	$\mathcal{H}(p^\pi)$
Pre-trained	6.78
FDC $\alpha = 0.5$	7.00
FDC $\alpha = 0.0$	7.14

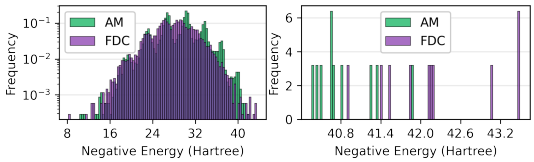
(d) Entropy evaluation



(e) Pre-trained samples (f) FDC samples

	Vendi	CLIP
Pre-trained	2.36	0.19
FDC $\alpha = 0.001$	**2.47**	**0.22**

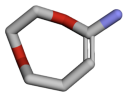
(g) Images evaluation



(h) Energy distributions (i) Top 0.2% Energies

	$\mathbb{E}[r(x)]$	SQ_β
Pre	15.4	24.2
AM	29.1	39.7
FDC-A	27.5	41.8

(j) SQ Negative Energy



(k) FDC-generated design

Figure 3: (top) Illustrative manifold exploration experiment via KL-regularized entropy maximization, (mid) High-dimensional manifold exploration via text-to-image model fine-tuning for prompt "A creative bridge design". Left: images from pre-trained model, Right: images from model fine-tuned via FDC, with higher diversity as indicated by a higher Vendi score. (bottom) Novelty-seeking molecular design for Energy (kcal/mol) maximization by fine-tuning FlowMol [15]. FDC shows enhanced control capabilities for optimizing such complex objectives than AM, a classic fine-tuning scheme.

Theorem 5.2 (Convergence guarantee of Flow Density Control for general functionals)**.** *Given the Robbins-Monro step-size rule: $\sum_k \gamma_k = \infty, \sum_k \gamma_k^2 < \infty$, under Assumption 5.2 and technical assumptions (see Appendix E), the sequence of marginal densities p_1^k induced by the iterates π_k of Algorithm 1 converges weakly to a stationary point $\tilde{p}_1$ of $\mathcal{G}$ almost surely, formally: $p_1^k \rightharpoonup \tilde{p}_1 \quad a.s.$.*

6 Experimental Evaluation

We analyze the ability of **F**low **D**ensity **C**ontrol (FDC) to induce policies optimizing complex non-linear objectives, and compare its performance with Adjoint Matching (AM) [14], a classic fine-tuning method. In the following, we present three experiments: (i) an illustrative and visually interpretable exploration task , (ii) a novelty-seeking molecular design problem for single-point energy minimization [18] , and (iii) manifold exploration for text-to-image *creative bridge design* generation. In Apx. G we provide further experiments for risk-averse and novelty-seeking utilities, as well as regularization via Wasserstein distances. Additional details are provided in Apx. H.

Conservative manifold exploration. We tackle manifold exploration [12] by fine-tuning a pre-trained model π^{pre} to maximize the entropy utility ($\mathcal{H}$ in Tab. 1) under a KL regularization of strength α, a capability not possible with prior methods [12]. As in previous work, we consider the common setting where the pre-trained model density p_1^{pre} concentrates most of its mass in a specific region as shown in Fig. 3a, where $N = 10000$ samples are shown. By fine-tuning π^{pre} via FDC, the density of the fine-tuned model shifts into low-coverage areas (see Fig. 3b and 3c). In particular, Fig. 3d demonstrates that reducing α from 0.5 to 0.0 yields progressively higher Monte Carlo entropy estimates (7.00 at $\alpha = 0.5$, 7.14 at $\alpha = 0$), thus enabling control of the trade-off between preserving the original distribution and exploring novel regions, a capability not supported by prior methods [12].

Molecular design for single-point energy minimization. We fine-tune FlowMol [15], pre-trained on QM9 [47], to discover molecules minimizing the single-point total energy computed via extended tight-binding at the GFN1-xTB level of theory [18]. Concretely, we maximize the negative energy. We do not aim to maximize the average sample reward, but rather that of the top 0.2% samples. We employ FDC with novelty-seeking SQ utility (see Tab. 1) with $\beta = 0.998$, and make 2 gradient steps per $K = 10$ iterations. We compare it with AM run for 240 steps. Fig. 3j shows that while AM generates better samples in average (namely 29.1 over 27.5 of FDC), the average quality of the top 0.2% molecules, indicated by SQ_β is higher for FDC than for AM (namely 41.8 over 39.7 of AM). This confirms (see Fig. 3i and 3h) that FDC can sacrifice the average reward to generate a few truly high-reward designs.

Text-to-image bridge designs conservative exploration. We perform manifold exploration by fine-tuning Stable Diffusion (SD) 1.4 [50] with prompt "A creative bridge design.". To this end, we maximize the KL-regularized entropy (see Tab. 1) with $\alpha = 0.001$ via FDC for $K = 2$ steps. As a diversity metric, we utilize the Vendi score [19] with cosine similarity kernel on the extracted CLIP [21] features from a sample of 100 images and compared it to the baseline pre-trained model in Fig. 3g. Beyond increasing the Vendi score, FDC also increases the CLIP score of the initial model.

Acknowledgements

This publication was made possible by the ETH AI Center doctoral fellowship to Riccardo De Santi, and postdoctoral fellowship to Marin Vlastelica. The project has received funding from the Swiss National Science Foundation under NCCR Catalysis grant number 180544 and NCCR Automation grant agreement 51NF40 180545.

References

[1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[3] Pierre-Cyril Aubin-Frankowski, Anna Korba, and Flavien Léger. Mirror descent with relative smoothness in measure spaces, with application to sinkhorn and em. *Advances in Neural Information Processing Systems*, 35:17263–17275, 2022.

[4] Anas Barakat, Ilyas Fatkhullin, and Niao He. Reinforcement learning with general utilities: Simpler variance reduction and large state-action space. In *International Conference on Machine Learning*, pages 1753–1800. PMLR, 2023.

[5] Michel Benaïm. Dynamics of stochastic approximation algorithms. In *Seminaire de probabilites XXXIII*, pages 1–68. Springer, 2006.

[6] Camille Bilodeau, Wengong Jin, Tommi Jaakkola, Regina Barzilay, and Klavs F Jensen. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1608, 2022.

[7] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[8] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[9] 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.

[10] Riccardo De Santi, Federico Arangath Joseph, Noah Liniger, Mirco Mutti, and Andreas Krause. Geometric active exploration in markov decision processes: the benefit of abstraction. *arXiv preprint arXiv:2407.13364*, 2024.

[11] Riccardo De Santi, Manish Prajapat, and Andreas Krause. Global reinforcement learning: Beyond linear and convex rewards via submodular semi-gradient methods. *arXiv preprint arXiv:2407.09905*, 2024.

[12] Riccardo De Santi, Marin Vlastelica, Ya-Ping Hsieh, Zebang Shen, Niao He, and Andreas Krause. Provable maximum entropy manifold exploration via diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

[13] Alexander Decruyenaere, Heidelinde Dehaene, Paloma Rabaey, Johan Decruyenaere, Christiaan Polet, Thomas Demeester, and Stijn Vansteelandt. Debiasing synthetic data generated by deep generative models. *Advances in Neural Information Processing Systems*, 37:41539–41576, 2024.

[14] 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.

[15] Ian Dunn and David Ryan Koes. Mixed continuous and categorical flow matching for 3d de novo molecule generation. *ArXiv*, pages arXiv–2404, 2024.

[16] Pavel Dvurechensky and Jia-Jie Zhu. Analysis of kernel mirror prox for measure optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2350–2358. PMLR, 2024.

[17] Jesse Farebrother, Matteo Pirotta, Andrea Tirinzoni, Rémi Munos, Alessandro Lazaric, and Ahmed Touati. Temporal difference flows. *arXiv preprint arXiv:2503.09817*, 2025.

[18] Marvin Friede, Christian Hölzer, Sebastian Ehlert, and Stefan Grimme. dxtb—an efficient and fully differentiable framework for extended tight-binding. *The Journal of Chemical Physics*, 161(6), 2024.

[19] Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

[20] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, 2019.

[21] 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*, pages 7514–7528, 2021.

[22] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.

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

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

[25] Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. Finding mixed nash equilibria of generative adversarial networks. In *International Conference on Machine Learning*, pages 2810–2819. PMLR, 2019.

[26] Yanwei Jia and Xun Yu Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *Journal of Machine Learning Research*, 23(154):1–55, 2022.

[27] Mohammad Reza Karimi, Ya-Ping Hsieh, and Andreas Krause. Sinkhorn flow as mirror flow: A continuous-time framework for generalizing the sinkhorn algorithm. In *International Conference on Artificial Intelligence and Statistics*, pages 4186–4194. PMLR, 2024.

[28] Flavien Léger. A gradient descent perspective on sinkhorn. *Applied Mathematics & Optimization*, 84(2):1843–1855, 2021.

[29] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. *Advances in neural information processing systems*, 29, 2016.

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

[31] 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.

[32] 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.

[33] Haihao Lu, Robert M Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

[34] Panayotis Mertikopoulos, Ya-Ping Hsieh, and Volkan Cevher. A unified stochastic approximation framework for learning in games. *Mathematical Programming*, 203(1):559–609, 2024.

[35] Alexander Mielke and Jia-Jie Zhu. Hellinger-kantorovich gradient flows: Global exponential decay of entropy functionals. *arXiv preprint arXiv:2501.17049*, 2025.

[36] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

[37] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[38] Mojmír Mutnỳ. *Modern Adaptive Experiment Design: Machine Learning Perspective*. PhD thesis, ETH Zurich, 2024.

[39] Mojmir Mutny, Tadeusz Janik, and Andreas Krause. Active exploration via experiment design in Markov chains. In *International Conference on Artificial Intelligence and Statistics*, 2023.

[40] Mirco Mutti, Riccardo De Santi, Piersilvio De Bartolomeis, and Marcello Restelli. Challenging common assumptions in convex reinforcement learning. *Advances in Neural Information Processing Systems*, 35:4489–4502, 2022.

[41] Mirco Mutti, Riccardo De Santi, Piersilvio De Bartolomeis, and Marcello Restelli. Convex reinforcement learning in finite trials. *Journal of Machine Learning Research*, 24(250):1–42, 2023.

[42] Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*, pages 16223–16239. PMLR, 2022.

[43] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[44] Kushagra Pandey, Jaideep Pathak, Yilun Xu, Stephan Mandt, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Heavy-tailed diffusion models. *arXiv preprint arXiv:2410.14171*, 2024.

[45] Manish Prajapat, Mojmír Mutnỳ, Melanie N Zeilinger, and Andreas Krause. Submodular reinforcement learning. *arXiv preprint arXiv:2307.13372*, 2023.

[46] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[47] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[48] R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[49] R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[51] Marta Skreta, Lazar Atanackovic, Avishek Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the it\^ o density estimator. *arXiv preprint arXiv:2412.17762*, 2024.

[52] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

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

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

[55] Lenart Treven, Jonas Hübotter, Bhavya Sukhija, Florian Dorfler, and Andreas Krause. Efficient exploration in continuous-time model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 36:42119–42147, 2023.

[56] 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.

[57] Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Sergey Levine, and Tommaso Biancalani. Feedback efficient online fine-tuning of diffusion models. *arXiv preprint arXiv:2402.16359*, 2024.

[58] Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34, 2020.

[59] Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex mdps. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

[60] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, et al. Mattergen: a generative model for inorganic materials design. *arXiv preprint arXiv:2312.03687*, 2023.

[61] Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4572–4583. Curran Associates, Inc., 2020.

[62] Hanyang Zhao, Haoxian Chen, Ji Zhang, David D Yao, and Wenpin Tang. Scores as actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. *arXiv preprint arXiv:2409.08400*, 2024.

Contents

A Related Works **13**

B Functionals and Derivation of Gradients of First-order Variations **14**

B.1 Overview of utilities and divergences in Table 1 14

B.2 A brief tutorial on first variation derivation 14

B.3 Derivation of gradients of first-order variation for functionals in Table 1 15

C Gradient of first variation: computation and estimation **18**

D Proof for Theorem 5.1 **19**

D.1 Optimization background 19

D.2 Convergence Proof 19

E Proof for Theorem 5.2 **21**

F Detailed Example of Algorithm Implementation **23**

F.1 Implementation of EntropyRegularizedControlSolver 23

F.2 Discussion: computational complexity and cost of FDC 23

G Further Experiments **25**

H Experimental Details **26**

H.1 Used computational resources 26

H.2 Experiments in Illustrative Settings 26

H.3 Real-World Experiments 26

A Related Works

Flow and diffusion models fine-tuning via optimal control. Recent works have framed fine-tuning of diffusion and flow models to maximize expected reward under KL regularization as an entropy-regularized optimal control problem [e.g., 56, 54, 57, 14]. Crucially, as shown in Sec. 3, the problem tackled by these studies is the specific sub-case of generative optimization (Eq. (5)), where the utility $\mathcal{F}$ is linear, and $\mathcal{D} = D_{KL}$. In this work, we propose a principled method with guarantees for the far more general class of non-linear utilities and divergences beyond KL, including the ones listed in Tab. 1. The framework introduced has strictly higher expressive power and control capabilities for fine-tuning generative model (see Sec. 3). This renders possible to tackle relevant tasks e.g., scientific discovery, beyond the capabilities of the aforementioned fine-tuning schemes.

Convex and General Utilities Reinforcement Learning. Convex and General (Utilities) RL [20, 59, 61] generalizes RL to the case where one wishes to maximize a concave [20, 59], or general [61, 4] functional of the state distribution induced by a policy over a dynamical system's state space. The introduced generative optimization problem (in Eq. (5)) is related, with p_1^π representing the state distribution induced by policy π over a subset of the state space. Recent works tackled the finite samples budget setting [e.g., 42, 40, 41, 45, 11]. Ultimately, to our knowledge, this is the first work leveraging an algorithmic scheme resembling General RL for the practically relevant task of generative optimization of general non-linear functionals via fine-tuning of diffusion and flow models.

Optimization over probability measures via mirror flows. Recently, there has been a growing interest in building theoretical guarantees for optimization problems over spaces of probability measures in a variety of applications. These include GANs [25], optimal transport [3, 28, 27], kernelized methods [16], and manifold exploration [12]. We present the first use of this framework to establish guarantees for the generative optimization problem in Eq. (5). This novel link to probability-space optimization sheds new light on large-scale flow and diffusion models fine-tuning.

B Functionals and Derivation of Gradients of First-order Variations

B.1 Overview of utilities and divergences in Table 1

In the following, we report the missing details for the functionals presented within Table 1, and discuss some possible applications.

Manifold Exploration and Generative Model De-biasing As mentioned within Sec. 3, maximization of the entropy functional as been recently introduced as a principled objective for manifold exploration [12]. Moreover, we wish to point out that it can be interpreted also from the viewpoint of de-biasing a prior generative model to re-distribute more uniformly its density while preserving a certain notion of support, e.g., via sufficient KL-divergence regularization.

Risk-averse and Novelty-seeking reward maximization A definition of q_β^r can be found below, explanations of these utilities can be found in Sec. 1, and experimental illustrative examples are provided in Sec. 6.

Optimal Experiment Design The task of Optimal Experimental Design (OED) [7] involves choosing a sequence of experiments so as to minimize some uncertainty metric for an unknown *quantity of interest* $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the set of all possible experiments. From a probabilistic standpoint, an optimal design may be viewed as a probability distribution over $\mathcal{X}$, prescribing how frequently each experiment should be performed to achieve maximal reduction in uncertainty about f [46]. This problem has been recently studied in the case where f is an element of a reproducing kernel Hilbert space (RKHS), i.e., $f \in \mathcal{H}_k$, induced by a known kernel $k(x, x') = \Phi(x)^\top \Phi(x')$ where $x, x' \in \mathcal{X}$ [38]. Given this setting, one might aim to acquire information about f according to different *criteria* captured by the scalarization function $s(\cdot)$ [39]. In particular, in Table 1, we report three illustrative choices for s:

- D-design: $\log\det(\cdot)$ (Information)
- A-design: $-\mathrm{Tr}(\cdot)$ (Parameter error)
- E-design: $\lambda_{max}(\cdot)$ (Worst projection error)

as reported in previous work [Table 1 39].

Diverse Mode Discovery This objective corresponds to a re-interpretation of the Diverse Skill Discovery objective introduced in the context of Reinforcement Learning [59]. Consider the case where it is given a discrete and finite set $\mathcal{S}$ of symbols interpretable as latent variables, which can be leveraged to (exactly or approximately) perform conditional generation. This objective captures the task of assuring maximal diversity, in terms of KL divergence between the different conditional components, represented as $p^{\pi,k}$ with $k \in \mathcal{S}$.

Log-barrier constrained generation This formulation can be found within the General Utilities RL literature [61]. In particular, here we show the case where constraints are enforced via a log-barrier function, namely $\log(\cdot)$. Nonetheless, the functional presented in Table 1 remains meaningful for general penalty functions.

Optimal transport distances OT distances within Table 1 and their relative notation are introduced below in the context of their first variation computation.

Maximum Mean Discrepancy Here k denotes a positive-definite kernel, which measures similarity between two points in sample space. Moreover, μ_p denotes a kernel mean embedding of distribution p [37]. In terms of applications, choosing a proper kernel k could render possible to preserve specific structure of the initial pre-trained model that would be otherwise lost via KL regularization.

B.2 A brief tutorial on first variation derivation

In this work, we focus on the functionals that are Fréchet differentiable: Let V be a normed spaces. Consider a functional $F : V \to \mathbb{R}$. There exists a linear operator $A : V \to \mathbb{R}$ such that the following

limit holds

$$\lim_{\|h\|_V \to 0} \frac{|F(f+h) - F(f) - A[h]|}{\|h\|_V} = 0. \tag{13}$$

We further assume that V admits certain structure such that every element in its dual space (the space of bounded linear operator on V) admits some compact representation. For example, when V is the set of compact-supported continuous bounded functions, there exists a unique positive Borel measure μ with the same support, which can be identified as the linear functional. We denote this element as $\delta F[f]$ such that $\langle \delta F[f], h\rangle = A[h]$. Sometimes we also denote it as $\frac{\delta F}{\delta f}$. We will refer to $\delta F[f]$ as the first-order variation of F at f.

In this section, we briefly review strategies for deriving the first-order variation of two broad classes of functionals: those defined in closed form with respect to the density (e.g., expectation and entropy) and those defined via variational formulations (e.g., CVaR, Wasserstein distance, and MMD).

- **Category 1: Functional defined in a closed form w.r.t. the density.** For this class of functionals, the first-order variations can typically be computed using its definition and chain rule.

  With definition (13) in mind, we can try to calculate the first-order variation of the mean functional. Consider a continuous and bounded function $r : \mathbb{R}^d \to \mathbb{R}$ and a probability measure μ on $\mathbb{R}^d$. Consider the functional $F(\mu) = \int r(x)\mu(x)dx$. We have

  $$|F(\mu + \delta\mu) - F(\mu) - \langle r, \delta\mu\rangle| = 0. \tag{14}$$

  We therefore obtain $\delta F[\mu] = r$ for all μ. We will compute the first-order variations for other functionals in the next subsection.

- **Category 2: Functionals defined through a variational formulation.** Another important subclass of functionals considered in this paper is the ones defined via a variational problem

  $$F[f] = \sup_{g \in \Omega} G[f, g], \tag{15}$$

  where Ω is a set of functions or vectors independent of the choice of f, and g is optimized over the set Ω. We will assume that the maximizer $g^*(f)$ that reaches the optimal value for $G[f, \cdot]$ is unique (which is the case for the functionals considered in this project). It is known that one can use the Danskin's theorem (also known as the envelope theorem) to compute

  $$\frac{\delta F[f]}{\delta f} = \partial_f G[f, g^*(f)], \tag{16}$$

  under the assumption that F is differentiable [36].

B.3 Derivation of gradients of first-order variation for functionals in Table 1

- **Risk-Averse Optimization (Category 2)** Recall that $q^r_\beta(p^\pi) = \sup\{v \in \mathbb{R} | F_Z(v) \le \beta\}$, where the random variable Z is defined as $Z = r(x)$ with $x \sim p^\pi(x)$. From [49], we have

  $$\mathrm{CVaR}^r_\beta(p^\pi) = \mathbb{E}[r(x)|r(x) \le q^r_\beta(p^\pi)] = \beta \inf_\zeta \left\{ \zeta + \frac{1}{\beta}\mathbb{E}\left[\min\{r(x) - \zeta, 0\}\right] \right\}.$$

  Moreover, we have ζ^* that solves the above optimization problem is exactly $\zeta^* = q^r_\beta(p^\pi)$. By Danskin's theorem, one has (in a weak sense)

  $$\frac{\delta \mathrm{CVaR}^r_\beta(p^\pi)}{\delta p^\pi} = \beta \min\{r(x) - q^r_\beta(p^\pi), 0\}. \tag{17}$$

- **Risk-Seeking Optimization (Category 2)** Recall that $q^r_\beta(p^\pi) = \sup\{v \in \mathbb{R} | F_Z(v) \le \beta\}$, where the random variable Z is defined as $Z = r(x)$ with $x \sim p^\pi(x)$. From [49], we have

  $$\mathrm{SQ}^r_\beta(p^\pi) = \mathbb{E}[r(x)|r(x) \ge q^r_\beta(p^\pi)] = (1-\beta) \inf_\zeta \left\{ \zeta + \frac{1}{1-\beta}\mathbb{E}\left[\max\{r(x) - \zeta, 0\}\right] \right\}.$$

  Moreover, we have ζ^* that solves the above optimization problem is exactly $\zeta^* = q^r_\beta(p^\pi)$. By Danskin's theorem, one has (in a weak sense)

  $$\frac{\delta \mathrm{SQ}^r_\beta(p^\pi)}{\delta p^\pi} = (1-\beta) \max\{r(x) - q^r_\beta(p^\pi), 0\}. \tag{18}$$

Application	Functional $\mathcal{F}$ / $\mathcal{D}$	First-order Variation	Density Control	
			convex	general
Reward optimization [14, 56]	$\mathbb{E}_{x\sim p^\pi}[r(x)]$	r	✓	✓
Manifold Exploration Gen. model de-biasing	$\mathcal{H}(p^\pi) := -\mathbb{E}_{x\sim p^\pi}[\log p^\pi(x)]$	$-1-\log p^\pi$	✓	✓
Risk-averse optimization	$\mathrm{CVaR}^r_\beta(p^\pi) := \mathbb{E}_{x\sim p^\pi}[r(x) \mid r(x) \le \mathrm{q}^r_\beta(p^\pi)]$	$\beta \min\{r(x) - q^r_\beta(p^\pi), 0\}$	✓	✓
	$\mathbb{E}_{x\sim p^\pi}[r(x)] - \mathbb{V}\mathrm{ar}(p^\pi)$	$r(x) - \left(r(x)^2 - 2\mathbb{E}_{x\sim p^\pi}[r(x)]r(x)\right)$	✗	✓
Risk-seeking optimization	$\mathrm{SQ}^r_\beta(p^\pi) := \mathbb{E}_{x\sim p^\pi}[r(x) \mid r(x) \ge q^r_\beta(p^\pi)]$	$(1-\beta)\max\{r(x) - q^r_\beta(p^\pi), 0\}$	✗	✓
Optimal Experiment Design	$\mathrm{s}(\mathbb{E}_{x\sim p^\pi}[\Phi(x)\Phi(x)^\top - \lambda \mathbb{I}])$ $\mathrm{s}(\cdot) \in \{\log\det(\cdot), -\mathrm{Tr}(\cdot)^{-1}, -\lambda_{max}(\cdot)\}$	see Equation (28)	✓	✓
Diverse modes discovery	$-\mathbb{E}_z[D_{KL}(p^{\pi,z} \,\|\, \mathbb{E}_k\, p^{\pi,k})]$	See Equation (30)	✗	✓
Log-Barrier Constrained Generation	$\mathbb{E}_{x\sim p^\pi}[r(x)] - \beta \log(\langle p^\pi, c\rangle - C)$	See Equation (29)	✓	✓
Kullback–Leibler divergence	$D_{KL}(p^\pi \,\|\, p^{pre}) = \int p^\pi(x) \log \frac{p^\pi(x)}{p^{pre}(x)}\, dx$	$1 + \log p^\pi - \log p^{pre}$	✓	✓
Rényi divergences	$D_\beta(p^\pi \,\|\, p^{pre}) := \frac{1}{\beta-1} \log \int (p^\pi(x))^\beta (p^{pre}(x))^{1-\beta}\, dx$	$\frac{\beta}{\beta-1}\left(\int \left(\frac{p}{q}\right)^\beta dq(x)\right)^{-1} \left(\frac{p}{q}\right)^{\beta-1}$	✓	✓
Optimal Transport distances	$W_p(p^\pi \,\|\, p^{pre}) := \inf_{\gamma\in\Gamma(p^\pi, p^{pre})} \mathbb{E}_{(x,y)\sim\gamma}[d(x,y)^p]^{\frac{1}{p}}$	see Equation (27)	✓	✓
Maximum Mean Discrepancy	$\mathrm{MMD}_k(p^\pi, p^{pre}) := \|\mu_{p^\pi} - \mu_{p^{pre}}\|, \mu_p := \mathbb{E}_{x\sim p}[k(x,\cdot)]$	$\arg\max_{\phi\in\mathcal{H}} \langle \phi, p^\pi - p^{pre}\rangle$	✓	✓

Table 2: Examples of practically relevant utilities $\mathcal{F}$ (blue) and divergences $\mathcal{D}$ (orange), and their first-order variations.

- **Rényi Divergence (Category 1)** Recall the definition of Rényi Divergence

$$D_\beta(p\|q) = \frac{1}{\beta-1} \log \int \left(\frac{p}{q}\right)^\beta dq(x). \tag{19}$$

We ignore higher-order terms like $O((\delta p)^2)$.

$$D_\beta(p+\delta p\|q) - D_\beta(p\|q) = \frac{1}{\beta-1} \log \frac{\int \left(\frac{p+\delta p}{q}\right)^\beta dq(x)}{\int \left(\frac{p}{q}\right)^\beta dq(x)} \tag{20}$$

$$= \frac{1}{\beta-1} \log \frac{\int \left(\frac{p}{q}\right)^\beta + \beta \left(\frac{p}{q}\right)^{\beta-1} \frac{\delta p}{q} dq(x)}{\int \left(\frac{p}{q}\right)^\beta dq(x)} \tag{21}$$

$$= \frac{1}{\beta-1} \log 1 + \frac{\int \beta \left(\frac{p}{q}\right)^{\beta-1} \frac{\delta p}{q} dq(x)}{\int \left(\frac{p}{q}\right)^\beta dq(x)} \tag{22}$$

$$= \frac{1}{\beta-1} \frac{\int \beta \left(\frac{p}{q}\right)^{\beta-1} \frac{\delta p}{q} dq(x)}{\int \left(\frac{p}{q}\right)^\beta dq(x)} \tag{23}$$

$$\frac{\delta}{\delta p} R_\beta(p,q) = \frac{\beta}{\beta-1} \left(\int \left(\frac{p}{q}\right)^\beta dq(x)\right)^{-1} \left(\frac{p}{q}\right)^{\beta-1} \tag{24}$$

- **Optimal transport and Wasserstein-p distance (Category 2)** Consider the optimal transport problem

$$\mathrm{OT}_c(u,v) = \inf_\gamma \left\{ \int\int c(x,y) d\gamma(x,y) : \int \gamma(x,y)dx = u(y), \int \gamma(x,y)dy = v(x) \right\} \tag{25}$$

where

$$\Gamma = \left\{ \gamma : \int \gamma(x,y)dx = u(y), \int \gamma(x,y)dy = v(x) \right\}$$

It admits the following equivalent dual formulation

$$\mathrm{OT}_c(u,v) = \sup_{f,g} \left\{ \int f du + \int g dv : f(x) + g(y) \leq c(x,y) \right\} \tag{26}$$

By taking $c(x,y) = \|x-y\|^p$, we recover $\mathrm{OT}_c(u,v) = W_p(u,v)^p$. Let f^* and g^* be the solution to the above dual optimization problem. From the Danskin's theorem, we have

$$\frac{\delta}{\delta u} W_p(u,v)^p = f^*. \tag{27}$$

In the special case of $p = 1$, we know that $g^* = -f^*$ (note that the constraint can be equivalently written as $\|\nabla f\| \leq 1$), in which case f^* is typically known as the critic in the WGAN framework.

- **Optimal Experiment Design. (Category 1)** We take $\mathrm{s}(M) = \log\det(M)$ as example. By chain rule, we have

$$\delta F[p^\pi] = \mathrm{Tr}\left[\left(\mathop{\mathbb{E}}_{x\sim p^\pi}[\Phi(x)\Phi(x)^\top - \lambda \mathbb{I}] \right)^{-1} \left(\Phi(x)\Phi(x)^\top - \lambda \mathbb{I} \right) \right]. \tag{28}$$

- **Log-Barrier Constrained Generation. (Category 1)** By chain rule, we obtain

$$\delta F[p^\pi] = r - \frac{\beta c}{\langle p^\pi, c\rangle - C}. \tag{29}$$

- **Diverse modes discovery. (Category 1)** By chain rule, we obtain

$$\begin{aligned} \frac{\delta F}{\delta p^{\pi,z}} &= -\frac{\delta}{\delta p^{\pi,z}} \mathbb{E}_z \left[\int p^{\pi,z} \log p^{\pi,z} dx - \int p^{\pi,z} \log\left(\mathbb{E}_k[p^{\pi,k}]\right) dx \right] \\ &= -\mathbb{E}_z \left[\frac{\delta}{\delta p^{\pi,z}} \left(\int p^{\pi,z} \log p^{\pi,z} dx \right) - \frac{\delta}{\delta p^{\pi,z}} \left(\int p^{\pi,z} \log\left(\mathbb{E}_k[p^{\pi,k}]\right) dx \right) \right] \\ &= -\mathbb{E}_z \left[\log p^{\pi,z} + 1 - \log\left(\mathbb{E}_k[p^{\pi,k}]\right) - \frac{p^{\pi,z}}{\mathbb{E}_k[p^{\pi,k}]} \right] \end{aligned} \tag{30}$$

- **Entropy. (Category 1)** As a first example, consider the entropy functional $\mathcal{F}(p) = -\int p \log p, dx$. By the definition of the first-order variation, we have $\frac{\delta \mathcal{F}}{\delta p}(p) = -1 - \log p$, and therefore $\nabla \frac{\delta \mathcal{F}}{\delta p}(p) = -\nabla \log p$. This gradient term can be effectively estimated using standard score approximations; see [12].

C Gradient of first variation: computation and estimation

Surprisingly, estimating $\nabla_x g_k$ in Alg. 1 (line 4) rarely requires density estimation. Among the functionals in Table 1, only the Rényi divergence does, for which one can leverage the recent Itô density estimator [51]. All other functionals admit straightforward plug-in or sample-based approximations detailed in Apx. B. As an illustrative example, in the following we showcase three examples from Table 1:

$$\nabla_x \delta\mathcal{Q}(p^\pi)(x) = \begin{cases} -\nabla_x \log p^\pi(x) & \text{Entropy } (\mathcal{H}) \\ \nabla_x r(x) \cdot \mathbf{1}\{r(x) \leq q^r_\beta(p^\pi)\} & \text{CVaR} \\ \nabla_x \phi^*(x) \text{ where } \phi^* = \arg\max_{\phi:\|\nabla_x \phi\| \leq 1} \langle \phi, p^\pi - p^{pre} \rangle & \text{Wasserstein-1 } (W_1) \end{cases}$$

Here $\mathcal{Q}$ denotes either a utility $\mathcal{F}$ or a divergence $\mathcal{D}$, and $q^r_\beta(p^\pi)$ is the β-quantile of $Z = r(X)$ with $X \sim p^\pi$ [48]. These gradients can be easily implemented. For entropy, the score term can be approximated via the score network in the case of diffusion models [12], and obtained via a known linear transformation of the learned velocity field in the case of flows [14, Eq.(8)]. For CVaR, any standard sample-based estimator of $q^r_\beta(p^\pi)$ [48] can be used. For Wasserstein-1, ϕ^* actually corresponds to the discriminator in Wasserstein-GAN, which can be learned with established methods [2]. In Apx. B, we report the gradient of the first variation for all functionals in Table 1, explain their practical estimation, and present a tutorial to derive the first variation of any new functionals not mentioned within Table 1.

D Proof for Theorem 5.1

D.1 Optimization background

We start by recalling the notion of Bregman divergence induced by a functional $\mathcal{Q} \in \mathbb{F}(\mathcal{X})$ between densities $\mu, \nu \in \mathbb{P}(\mathcal{X})$, namely:

$$D_{\mathcal{Q}}(\mu \,\|\, \nu) \coloneqq \mathcal{Q}(\mu) - \mathcal{Q}(\nu) - \langle \delta\mathcal{Q}(\nu), \mu - \nu \rangle$$

Next, we introduce two structural properties for our analysis.

Definition 1 (Relative smoothness and relative strong concavity [33])**.** *Let $\mathcal{G} : \mathbb{P}(\mathcal{X}) \to \mathbb{R}$ a concave functional. We say that $\mathcal{G}$ is L-smooth relative to $\mathcal{Q} \in \mathbb{F}(\mathcal{X})$ over $\mathbb{P}(\mathcal{X})$ if $\exists$ L scalar s.t. for all $\mu, \nu \in \mathbb{P}(\mathcal{X})$:*

$$\mathcal{G}(\nu) \geq \mathcal{G}(\mu) + \langle \delta\mathcal{G}(\mu), \nu - \mu \rangle - L D_{\mathcal{Q}}(\nu \,\|\, \mu) \tag{31}$$

and we say that $\mathcal{G}$ is l-strongly concave relative to $\mathcal{Q} \in \mathbb{F}(\mathcal{X})$ over $\mathbb{P}(\mathcal{X})$ if $\exists$ $l \geq 0$ scalar s.t. for all $\mu, \nu \in \mathbb{P}(\mathcal{X})$:

$$\mathcal{G}(\nu) \leq \mathcal{G}(\mu) + \langle \delta\mathcal{G}(\mu), \nu - \mu \rangle - l D_{\mathcal{Q}}(\nu \,\|\, \mu) \tag{32}$$

D.2 Convergence Proof

Theorem 5.1 (Convergence guarantee of Flow Density Control with concave functionals)**.** *Given Assumptions 5.1, fine-tuning a pre-trained model π^{pre} via* FDC *(Algorithm 1) with $\eta_k = L \; \forall k \in [K]$, leads to a policy π inducing a marginal distribution p_1^{π} such that:*

$$\mathcal{G}(p_1^*) - \mathcal{G}(p_1^{\pi}) \leq \frac{L - l}{K} D_{KL}(p_1^* \,\|\, p_1^{pre}) \tag{10}$$

where $p_1^* \coloneqq p_1^{\pi^*}$ is the marginal distribution induced by the optimal policy $\pi^* \in \arg\max_{\pi} \mathcal{G}(p_1^{\pi}) \coloneqq \mathcal{F}(p_1^{\pi}) - \alpha \mathcal{D}(p_1^{\pi} \,\|\, p_1^{pre})$.

Proof. We prove this result using the framework of relative smoothness and relative strong convexity introduced in Section 5.

The analysis is based on the classical mirror descent framework under relative properties [33]. For notational simplicity, we let $\mu_k \coloneqq p_T^{\pi_k}$, and fix an arbitrary reference density $\mu \in \mathbb{P}(\Omega_{\text{pre}})$. To better align the notation of our theory with existing literature, we will proceed with the *convex* functional $\tilde{\mathcal{G}} \coloneqq -\mathcal{G}$ below.

We begin by showing the following inequality:

$$\begin{aligned} \tilde{\mathcal{G}}(\mu_k) &\leq \tilde{\mathcal{G}}(\mu_{k-1}) + \langle \delta\tilde{\mathcal{G}}(\mu_{k-1}), \mu_k - \mu_{k-1} \rangle + L D_{\mathcal{Q}}(\mu_k, \mu_{k-1}) && (33) \\ &\leq \tilde{\mathcal{G}}(\mu_{k-1}) + \langle \delta\tilde{\mathcal{G}}(\mu_{k-1}), \mu - \mu_{k-1} \rangle + L D_{\mathcal{Q}}(\mu, \mu_{k-1}) - L D_{\mathcal{Q}}(\mu, \mu_k). && (34) \end{aligned}$$

The first inequality follows from the L-smoothness of $\mathcal{G}$ relative to $\mathcal{Q}$ as defined in Definition 1. The second inequality uses the three-point inequality of the Bregman divergence [33, Lemma 3.1] with $\phi(\mu) = \frac{1}{L}\langle \delta\mathcal{G}(\mu_{k-1}), \mu - \mu_{k-1} \rangle$, $z = \mu_{k-1}$, and $z^+ = \mu_k$.

Next, using the l-strong concavity of $\mathcal{G}$ relative to $\mathcal{Q}$, again from Definition 1, we obtain:

$$\tilde{\mathcal{G}}(\mu_k) \leq \tilde{\mathcal{G}}(\mu) + (L - l) D_{\mathcal{Q}}(\mu, \mu_{k-1}) - L D_{\mathcal{Q}}(\mu, \mu_k). \tag{35}$$

By recursively applying the above inequality and using the monotonicity of $\mathcal{G}(\mu_k)$ along with the non-negativity of the Bregman divergence, we obtain [33]:

$$\sum_{k=1}^{K} \left(\frac{L}{L - l} \right)^k \left(\tilde{\mathcal{G}}(\mu_k) - \tilde{\mathcal{G}}(\mu) \right) \leq L D_{\mathcal{Q}}(\mu, \mu_0) - L \left(\frac{L}{L - l} \right)^K D_{\mathcal{Q}}(\mu, \mu_K) \leq L D_{\mathcal{Q}}(\mu, \mu_0). \tag{36}$$

Letting

$$\frac{1}{C_K} \coloneqq \sum_{k=1}^{K} \left(\frac{L}{L - l} \right)^k, \tag{37}$$

and rearranging terms, we arrive at the convergence rate:

$$\tilde{\mathcal{G}}(\mu_K) - \tilde{\mathcal{G}}(\mu) \leq C_K L D_{\mathcal{Q}}(\mu, \mu_0) = \frac{l D_{\mathcal{Q}}(\mu, \mu_0)}{\left(1 + \frac{l}{L-l}\right)^K - 1}. \tag{38}$$

Finally, the convergence rate stated in the theorem follows by observing that $\left(1 + \frac{l}{L-l}\right)^K \geq 1 + \frac{Kl}{L-l}$.

□

E Proof for Theorem 5.2

To establish our main convergence result, we introduce two additional technical assumptions that are satisfied in virtually all practical settings:

Assumption E.1 (Support Compatibility)**.** *We assume that the support of $p_T^{\pi_k}$ is contained in a fixed compact domain $\tilde{\Omega}$ for all k, and that for some j, we have supp$(p_j^{\pi_k}) = \tilde{\Omega}$.*

Assumption E.2 (Precompactness)**.** *The sequence $\{\delta\mathcal{H}(p_T^{\pi_k})\}_k$ is precompact in the topology induced by the L_∞ norm.*

We are now ready to present the full proof. For the reader's convenience, we restate the theorem:

Theorem 5.2 (Convergence guarantee of Flow Density Control for general functionals)**.** *Given the Robbins-Monro step-size rule: $\sum_k \gamma_k = \infty, \sum_k \gamma_k^2 < \infty$, under Assumption 5.2 and technical assumptions (see Appendix E), the sequence of marginal densities p_1^k induced by the iterates π_k of Algorithm 1 converges weakly to a stationary point $\tilde{p_1}$ of $\mathcal{G}$ almost surely, formally: $p_1^k \rightharpoonup \tilde{p_1} \quad a.s.$.*

Proof. To facilitate readability, we begin with an outline of the key steps.

Proof Outline The main idea is to relate the discrete iterates $\{p_T^k\}_{k\in\mathbb{N}}$ produced by Algorithm 1 to a continuous-time dynamical system. Let us define the initial dual variable as:

$$h_0 = \delta\mathcal{H}(p_{pre}) = -\log p_{pre},$$

and consider the following gradient flow:

$$\begin{cases} \dot{h}_t = \delta\mathcal{G}(p_t), \\ p_t = \delta(-\mathcal{H})^\star(h_t), \end{cases} \tag{MF}$$

where $(-\mathcal{H})^\star(h) = \log \int_\Omega e^h$ is the Fenchel dual of the negative entropy functional [25, 22].

To connect this with our algorithm, we construct a continuous-time interpolation of the dual iterates $h^k = \delta\mathcal{H}(p_T^{\pi_k})$. Define the effective time $\tau^k = \sum_{r=0}^k \alpha_r$, and let the interpolated process $h(t)$ be given by:

$$h(t) = h^k + \frac{t - \tau^k}{\tau + 1^k - \tau^k}(\tau + 1^k - h^k). \tag{Int}$$

Intuitively, our convergence result follows if two conditions hold:

Informal Assumption 1 (Closeness to Continuous-Time Flow)**.** *The interpolated process $h(t)$ asymptotically follows the dynamics of* (MF) *as $k \to \infty$.*

Informal Assumption 2 (Convergence of the Flow)**.** *The trajectories of* (MF) *converge to a stationary point of $\mathcal{G}$.*

To formalize this, we invoke the stochastic approximation framework of [5]. Let $\mathcal{Z}$ be the space of integrable functions on Ω, and let Θ denote the flow of (MF). We define:

Definition 2 (Asymptotic Pseudotrajectory (APT))**.** *We say $h(t)$ is an asymptotic pseudotrajectory (APT) of* (MF) *if for all $T > 0$,*

$$\lim_{t\to\infty} \sup_{0\le h\le T} \|h(t+h) - \Theta_h(h(t))\|_\infty = 0.$$

If $h(t)$ is a precompact APT, then [5] show:

Theorem E.1 (APT Limit Set Theorem)**.** *Let $h(t)$ be a precompact APT for the flow* (MF)*. Then, almost surely, the limit set of $h(t)$ is contained in the set of internally chain-transitive (ICT) points of* (MF)*.*

The proof of our result follows from two claims:

1. The iterates $\{h^k\}$ generate a precompact APT under Assumptions E.1 and5.2.

2. The ICT set of (MF) consists only of stationary points of $\mathcal{G}$.

The second claim holds because (MF) is a gradient flow—specifically, the spherical Hellinger–Kantorovich flow [35]. By Sard's theorem and standard results in dynamical systems [5], the ICT set must consist of stationary points.

For the first claim, Assumptions E.1 and E.2 ensure that the interpolated process is well-defined and precompact, while Assumption 5.2 allows us to apply standard stochastic approximation arguments [27]. We conclude the proof by applying Theorem E.1. □

F Detailed Example of Algorithm Implementation

F.1 Implementation of EntropyRegularizedControlSolver

To ensure completeness, below we provide pseudocode for one concrete realization of a EntropyRegularizedControlSolver as in Eq. (8) using a first-order optimization routine. In particular, we describe exactly the version employed in Sec. 6, which builds on the Adjoint Matching framework [14], casting linear fine-tuning as a stochastic optimal control problem and tackling it via regression.

Let u^{pre} be the initial, pre-trained vector field, and $u^{finetuned}$ its fine-tuned counterpart. We also use $\bar{\alpha}$ to refer to the accumulated noise schedule from [23] effectively following the flow models notation introduced by Adjoint Mathing [14, Sec. 5.2]. The full procedure is in Algorithm 2.

Algorithm 2 EntropyRegularizedControlSolver (Adjoint Matching [14]) based implementation

1: **Input:** N : number of iterations, u^{pre} : pre-trained flow vector field, η regularization coefficient as in Eq. (8), h : step size, ∇f: reward function gradient, m batch size
2: **Init:** $u^{finetuned} := u^{pre}$ with parameter θ
3: **for** $n = 0, 1, 2, \ldots, N-1$ **do**
4: Sample m trajectories $\{X_t\}_{t=1}^{T}$ via memoryless noise schedule [14], e.g.,

$$\text{sample } \epsilon_t \sim \mathcal{N}(0, I),\ X_0 \sim \mathcal{N}(0, I), \text{ then:}$$

$$X_{t+h} = X_t + h\left(2v_\theta^{finetuned}(X_t, t) - \frac{\bar{\alpha}_t}{\alpha_t} X_t\right) + \sqrt{h}\sigma(t)\epsilon_t$$

Use reward gradient:

$$\tilde{a}_T = -\frac{1}{\eta}\nabla f(X_1)$$

For each trajectory, solve the lean adjoint ODE, see [14, Eq. 38-39], from $t = 1$ to 0, e.g.,:

$$\tilde{a}_{t-h} = \tilde{a}_t + h\tilde{a}_t^\top \nabla_{X_t}\left(2u^{pre}(X_t, t) - \frac{\bar{\alpha}_t}{\alpha_t} X_t\right)$$

Where X_t and $\tilde{a}_t$ are computed without gradients, i.e., $X_t = \texttt{stopgrad}(X_t), \tilde{a}_t = \texttt{stopgrad}(\tilde{a}_t)$. For each trajectory compute the Adjoint Matching objective [14, Eq. 37]:

$$\mathcal{L}_\theta = \sum_{t=0}^{1-h} \| \frac{2}{\sigma(t)}\left(u_\theta^{finetuned}(X - t, t) - u^{pre}(X_t, t)\right) + \sigma(t)\tilde{a}_t \|$$

Compute the gradient $\nabla_\theta \mathcal{L}(\theta)$ and update θ.
5: **end for**
6: **output:** Fine-tuned noise predictor $u_\theta^{finetuned}$

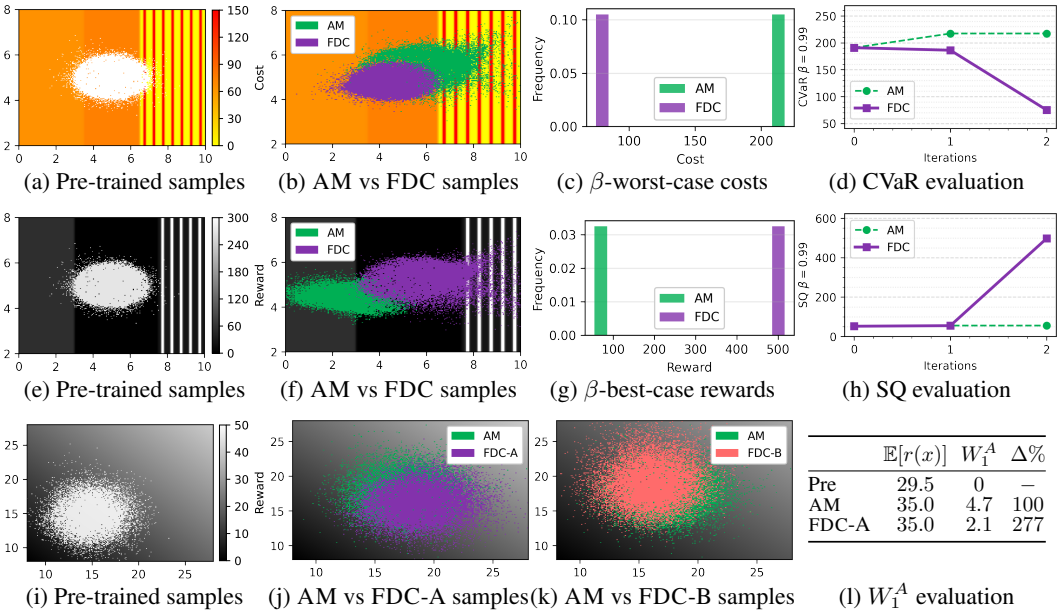
F.2 Discussion: computational complexity and cost of FDC

Flow Density Control (see Algorithm 1) is a sequential fine-tuning scheme, which performs K iterations of a base fine-tuning oracle, as shown in Algorithm 1. Typically, as for the case of Adjoint Matching [14], which is contextualized in Algorithm 2, the inner oracle also performs N iterations to solve the classic fine-tuning problem. As a consequence, at first glance, this lead to FDC having a computational complexity scaling linearly in K the one of classic fine-tuning. Nonetheless, this does not seem to capture well the practical computational cost. In particular, we wish to point out the two following observations:

- As discussed for the molecular design experiment in Sec. 6 and further in Appendix H, the FDC scheme might work well even with a very approximate oracle to solve the entropy-regularized control problem at each iteration.
- For many real-world problems a very small number of iterations K might be sufficient to approximate the non-linear functional sufficiently well and hence obtain useful fine-tuned

models. This is shown in text-to-image bridge design experiment in Sec. 6 and in Appendix H. In this case, merely $K = 2$ iterations of FDC lead to promising results.

G Further Experiments



(a) Pre-trained samples (b) AM vs FDC samples (c) β-worst-case costs (d) CVaR evaluation

(e) Pre-trained samples (f) AM vs FDC samples (g) β-best-case rewards (h) SQ evaluation

(i) Pre-trained samples (j) AM vs FDC-A samples (k) AM vs FDC-B samples (l) W_1^A evaluation

	$\mathbb{E}[r(x)]$	W_1^A	$\Delta\%$
Pre	29.5	0	–
AM	35.0	4.7	100
FDC-A	35.0	2.1	277

Figure 4: Illustrative settings with visually interpretable results. (top) Risk-averse reward maximization for valid or safe generation, (mid) Novelty-seeking reward maximization for discovery, (bottom) Expected rewards maximization under optimal transport distance regularization. Crucially, FDC can optimize well these complex objectives, while AM [14], a classic fine-tuning scheme, fails at this.

Risk-averse reward maximization for better worst-case validity or safety. We fine-tune a pre-trained policy π^{pre} (see Fig. 4a) by optimizing the CVaR$_\beta$ utility i.e., expected outcome in the β-worst-case (see Tab. 1) with KL regularization, and costs interpreted as negative rewards. The cost has three regions: a high-cost plateau (dark orange), where the initial density lies; a moderate-cost left area (light orange); and a predominantly low-cost right zone (yellow) punctuated by narrow, but catastrophic red-stripes. As shown in Fig. 4b, AM moves the model density into the yellow region, lowering average cost but exposing it to rare extreme costs. In contrast, FDC, run with $K = 2$ iterations and $\beta = 0.01$, successfully steers density into the safer, moderate-cost area, cutting the 1%-worst-case cost from 217.0 achieved by AM to 75.0, well below the initial 190.9, as shown in Fig. 4c and 4d.

Novelty-seeking reward maximization for discovery. We fine-tune a pre-trained policy π^{pre} to maximize the SQ$_\beta$ utility, i.e., expected outcome in the β-best-case (see Tab. 1). The reward shown in Fig. 4e has a moderately high-reward left area (light gray), a medium-reward central plateau (darker gray) where the initial density lies, and a low-reward right region (black) with sparse, extreme-reward spikes depicted by thin white lines. As shown in Fig. 4f, AM drifts the density into the safer left basin — improving the average reward but only reaching a best-1% expected reward of 55.5, as shown in Fig. 4g and Fig. 4h. In contrast, FDC, run for $K = 2$ iterations and $\beta = 0.99$, pushes the density rightwards, elevating the top-1% reward to 497.7 (see Fig. 4h) — far above both AM and the initial 52.1.

Reward maximization regularized via optimal transport distance. We fine-tune the pre-trained model with density in Fig. 4i to maximize a reward function that increases moving top right. We consider two W_1 distances induced by two ground metrics: d_A, which makes vertical moves more costly than horizontal ones, and d_B, which does the opposite. Under d_A, both AM and the OT-regularized model reach an expected reward of 35.0, but FDC-A incurs only $W_1^A = 2.1$ versus 4.7 for AM, and achieves a mean shift that is 277% larger in the horizontal than in the vertical direction (Fig. 4j and Tab. 4l). By contrast, FDC-B under d_B preferentially shifts the density upward (Fig. 4k).

H Experimental Details

H.1 Used computational resources

We run all experiments on a single Nvidia H100 GPU.

H.2 Experiments in Illustrative Settings

Shared experimental setup. For all illustrative experiments we utilize Adjoint Matching (AM) [14] for the entropy-regularized fine-tuning solver in Algorithm 1. Moreover, the stochastic gradient steps within the AM scheme are performed via an Adam optimizer.

Risk-averse reward maximization for better worst-case validity or safety. In this experiment, we execute FDC for $K = 2$ iterations with a total of 1000 gradient steps within each iteration, AM solver (within the FDC scheme) with learning rate of $2e^{-2}$, $\alpha = 10^9$, and $\eta = 10$. Meanwhile, the AM baseline, is run for 1000 gradient steps with $\alpha = 0.2857$, and learning rate of $1e^{-5}$. The resulting CVaR is computed via the standard torch quantile method. The values of β reported in the main paper effectively refers to the value of $1 - \beta$.

Novelty-seeking reward maximization for discovery. We run FDC for $K = 2$ iterations with a total of 1000 gradient steps within each iteration, AM solver (within the FDC scheme) with learning rate of $3e^{-6}$, $\alpha = 10^5$, and $\eta = 0.625$, and 8000 samples are used to estimate the first variation gradient as explained in Appendix B. Meanwhile, the AM baseline, is run for 1000 gradient steps with $\alpha = 0.333$, and learning rate of $1e^{-5}$. The resulting SQ is computed via the standard torch quantile method.

Reward maximization regularized via optimal transport distance. Within this experiment, we present two runs of FDC, namely FDC-A and FDC-B, compared against AM. Both FDC-A and FDC-B have been run for $K = 6$ iterations of FDC, with $\alpha = 0.1$, AM oracle learning rate of $1e^{-6}$, $\eta = 6.666$. Both their discriminators to solve the dual OT problem as presented in Appendix B and mentioned within Sec. 4, have been learned via a simple MLP architecture with 800 gradient steps, by enforcing the 1-Lip. condition via the standard gradient penalty technique with regularization strength of $\lambda_{GP} = 10.0$ and learning rate of $1e^{-4}$. In particular, FDC-A is based on the distance defined, for two 2-dimensional points $x = (x_1, x_2)$ and $y = (y_1, y_2)$ by:

$$d_A(x, y) = \sqrt{(x_1 - y_1)^2 + (K(x_2 - y_2))^2}$$

Analogously, FDC-B leverages d_B defined as:

$$d_A(x, y) = \sqrt{(K(x_1 - y_1))^2 + (x_2 - y_2)^2)}$$

Where $K = 7$ in both cases. On the other hand, the AM baseline is run for 1000 gradient steps with learning rate of $1e^{-3}$ and $\alpha = 1.538$.

Conservative manifold exploration. We ran FDC for $K = 50$ iterations and 2500 gradient steps in total with $\eta = 10$ and $\alpha = 0.0, 0.01, 0.1, 0.5, 1.0$. We set the AM learning rate to $2e^{-4}$ and sample trajectories of length 400 for computing the AM loss.

H.3 Real-World Experiments

Molecular design for single-point energy minimization. In this experiment FDC is run for $K = 10$ iterations, with merely 2 gradient steps at each iteration (i.e., the AM oracle is very approximate), AM learning rate of $1e^{-4}$, $\eta = 0.01$ and $\alpha = 0$. Meanwhile, the AM baseline is run for 240 gradient steps with $\alpha = 0.0045$.

Text-to-image bridge designs conservative exploration. For this experiment we ran FDC on a single Nvidia H100 GPU, with $K = 2$, $\eta = 200$, $\alpha = 0.001$ and a 100 gradient steps in total. Similarly to previous work, we tuned the vector field resulting from applying classifier-free guidance with guidance scale $w = 8$ in SD-1.5.