Constrained Flow Optimization via Sequential Fine-Tuning for Molecular Design

Sven Gutjahr* ETH Zurich sgutjahr@student.ethz.ch

Riccardo De Santi* ETH Zurich rdesanti@ethz.ch

Luca Schaufelberger* ETH Zurich schaluca@ethz.ch

Kjell Jorner ETH Zurich kjell.jorner@ethz.chemail

Andreas Krause ETH Zurich krausea@ethz.ch

Abstract

Adapting generative foundation models to optimize rewards of interest (e.g., binding affinity) while satisfying constraints (e.g., molecular synthesizability) is of fundamental importance to render them applicable in real-world discovery campaigns such as molecular design. While recent works have introduced scalable methods for reward-guided fine-tuning of diffusion and flow models, it remains an open problem how to algorithmically trade off property maximization and constraint satisfaction in a reliable and predictable manner. Towards tackling this challenging problem, we first present a rigorous formulation for constrained generative optimization. Then, we introduce **C**onstrained **F**low **O**ptimization (CFO), an augmented Lagrangian method that renders it possible to arbitrarily control the aforementioned trade-off between reward maximization and constraint satisfaction. We provide convergence guarantees for the proposed scheme. Ultimately, we present an experimental evaluation on both synthetic, yet illustrative, settings, and a molecular design task optimizing molecular properties while constraining energy.

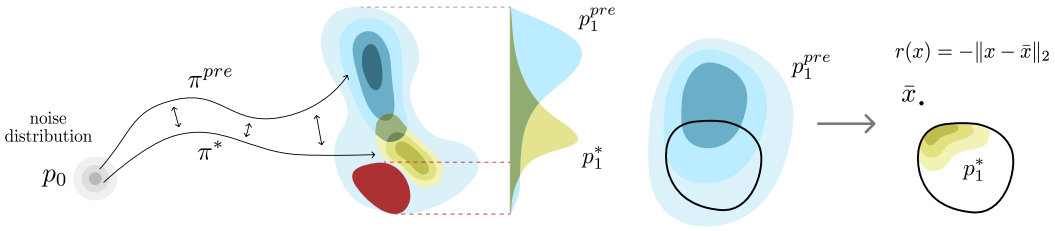
1 Introduction

Recent advances in generative modeling, particularly the advent of diffusion [15, 28, 27] and flow models [20], have led to state-of-the-art performances in several biological tasks, including generating protein structures [36], drug-like molecules [11], and DNA sequences [29], among others. These foundation models excel at capturing complex data distributions and generating realistic samples. However, approximately sampling from the data distribution is insufficient for most real-world discovery applications, where one typically wishes to generate candidates maximizing task-specific properties, or *rewards*, such as binding affinity or druglikeness in drug discovery. Recent works have introduced scalable fine-tuning methods that adapt a pre-trained flow or diffusion model to maximize a given reward under KL-regularization from the pre-trained model, using formulations from control theory or reinforcement learning [10, 33, 30]. However, shifting model density toward high-reward regions can make regularization insufficient to enforce validity constraints [32]. In chemistry and biology, such constraints may include the physical validity of docking poses [5], or the toxicity and synthesizability of drug candidates [12, 23]. Motivated by this, we ask the following question:

How can we fine-tune a pre-trained flow or diffusion model to controllably trade-off reward optimization and satisfiability of known constraints?

Our approach. In this work, we answer this question by first formalizing the problem of *constrained generative optimization*, extending current fine-tuning formulations to the constrained case (Sec. 3). Next, we introduce **C**onstrained **F**low **O**ptimization (CFO), a fine-tuning method based on the

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.



(a) Constrained generative optimization. Red: high-cost area.

(b) Adaptation to a low-cost area.

Figure 1: (a) Pre-trained and fine-tuned policies yield p_1^{pre} and p_1^* for reward r (increasing downwards); red indicates high cost. (b) p_1^{pre} adapts to p_1^* to maximize r while constrained to the low-cost area.

augmented Lagrangian (AL) scheme [3], which sequentially adapts the model to balance reward maximization with constraint satisfaction, thereby enabling controllable and reliable sample generation (Sec. 4). The proposed principled algorithm enables the transfer of classic constrained optimization guarantees from the AL method to generative model finetuning (Sec. 5). Finally, we demonstrate the effectiveness of CFO on both synthetic settings and a molecular design task, where it generates molecules with large dipole moments [22] while satisfying energetic constraints (Sec. 6).

Our contributions. To sum up, we present the following contributions:

- We present the *constrained generative optimization via fine-tuning* problem for flow models (Sec. 3).
- We introduce **C**onstrained **F**low **O**ptimization (CFO), an augmented Lagrangian-based fine-tuning method for flow and diffusion models (Sec. 4).
- We provide constrained optimization guarantees for CFO based on the AL scheme (Sec. 5).
- We evaluate CFO's ability to controllably trade-off reward maximization and constraint satisfaction in both synthetic settings and a molecular design task (Sec. 6).

2 Background and Notation

Generative Flow Models. Flow models aim to approximately sample from a data distribution p^{data}, by transforming samples from an initial distribution p^{init} into samples from p^{data} [7, 20]. A flow is a map $\psi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ denoted by $\psi_t(x)$, and it is be defined by a velocity field $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, via the *flow ODE*:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x_0) = u_t(\psi_t(x_0)) \quad \text{with} \quad \psi_0(x_0) = x_0, \tag{1}$$

A generative flow model is a continuous-time process $\{X_t\}_{0 \le t \le 1}$ induced by a flow ψ via $X_0 \sim p^{\text{init}}$ as $X_t = \psi_t(X_0), t \in [0,1]$, such that $X_1 = \psi_1(X_0) \sim p^{\text{data}}$. A flow model induces a probability path of marginal densities $p = \{p_t\}_{0 \le t \le 1}$ such that at time t: $X_t = \psi_t(X_0) \sim p_t$.

Continuous-time Reinforcement Learning (ctRL). We present finite-horizon ctRL [35, 31, 38] as a specific case of stochastic optimal control. Let $\mathcal{X} := \mathbb{R}^d \times [0,1]$ be a state space and $\mathcal{A}$ an action space, with the transition dynamics governed by: $\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = a_t(\psi_t(x))$ and denote by $\pi(X_t, t) \in \mathcal{A}$ the policy, which is as mapping form a state $(x,t) \in \mathcal{X}$ to an action $a \in \mathcal{A}$ such that $a_t = \pi(X_t, t)$. Denote with p_t^π the marginal density at time t induced by policy π. Thus, a pre-trained flow model with velocity field u^{pre} can be interpreted as an action process $a_t^{\text{pre}} := u^{\text{pre}}(X_t, t)$, where a_t^{pre} is determined by a policy via $a_t^{\text{pre}} = \pi^{\text{pre}}(X_t, t)$ [8]. Therefore, we can express the flow ODE induced by a pre-trained flow model by replacing u_t with a^{pre} in Eq. 1, and denote the pre-trained model by its policy π^{pre}, which induces a marginal density $p_1^{\text{pre}} := p_1^{\pi^{\text{pre}}}$ approximating p^{data}.

3 Constrained Generative Optimization via Flow Fine-Tuning

We aim to fine-tune a pre-trained diffusion model π^{pre} to obtain a new model π^*, inducing a process:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = a_t^{\text{fine}}(\psi_t(x)), \text{ with } a_t^{\text{fine}} = \pi^*(x_t, t) \tag{2}$$

The fine-tuned model should induce a distribution $p_1^* := p_1^{\pi^*}$ that maximizes the expected value of a target property, while retaining prior information from the pre-trained model π^{pre} and respecting arbitrary constraints. The problem is illustrated in Fig. 1 and formalized as follows:

Constrained Generative Optimization via Flow Fine-Tuning

$$\begin{aligned} &\arg\max_{\pi} \mathbb{E}_{x \sim p_1^\pi}[r(x)] - \alpha D_{KL}(p_1^\pi || p_1^{\text{pre}}) \\ &\text{s.t. } \mathbb{E}_{x \sim p_1^\pi}[c(x)] \le B \end{aligned} \tag{3}$$

Algorithm 1 **C**onstrained **F**low **O**ptimization

Init: Set initial Lagrange multiplier $\lambda_1 = 0$ and $V_0 = -\infty$
for $k = 1, 2, \ldots, K$ **do**
  Step 1: Update fine-tuning AL objective:

$$f_k(x) \coloneqq r(x) - \frac{\rho_k}{2}\left[\max\left(0,\, c(x) - B - \frac{\lambda_k}{\rho_k}\right)\right]^2 \tag{4}$$

  Step 2: Compute π_k via fine-tuning:

$$\pi_k \leftarrow \textsc{FineTuningSolver}(f_k, \pi_{\text{pre}}) \tag{5}$$

  Step 3: Update Lagrange multiplier:

$$\bar{\lambda}_{k+1} \leftarrow \min\Big\{0,\, \lambda_k - \rho_k\big(\mathbb{E}_{x\sim p_1^{\pi_k}}[c(x)] - B\big)\Big\} \qquad \lambda_{k+1} = \max\{\bar{\lambda}_{k+1}, \lambda_{\min}\}. \tag{6}$$

  Step 4: Set $V_k = \min\Big\{\mathbb{E}_{x\sim p_1^{\pi_k}}[c(x)] - B,\, -\lambda_k/\rho_k\Big\}$ and update $\rho_{k+1} = \begin{cases}\rho_k, & \text{if } V_k \le \tau V_{k-1},\\ \eta\rho_k, & \text{otherwise,}\end{cases}$
end for
Return: π_K

Where r and $c : \mathbb{R}^d \to \mathbb{R}$ are respectively scalar reward and constraint functions, $\alpha \in \mathbb{R}$ determines the KL-regularization strength, and $B \in \mathbb{R}$ controls the permissible degree of constraint violation.

Crucially, the *constrained generative optimization via fine-tuning* problem in Eq. 3 allows constructing a flow model that maximizes reward r while limiting constraint violation to any value B. The reward-constraint trade-off is critical in settings such as Fig. 1b, where the reward-maximizing region lies outside the valid data points, a typical case when learned reward functions act as property predictors [32]. In such cases, naive optimization may lead to high-reward but constraint-violating samples.

In the next section, we propose an algorithm that can tackle the constrained generative optimization problem in Eq. 3 by leveraging as a subroutine any unconstrained fine-tuning method [e.g., 10, 33].

4 Constrained Flow Optimization (CFO)

We present **C**onstrained **F**low **O**ptimization (CFO) (Alg. 1), an algorithm that tackles the constrained generative optimization problem in Eq. 3 by reducing it to a sequence of unconstrained fine-tuning subproblems. In each iteration, the algorithm constructs an Augmented Lagrangian (AL) objective based on the current policy, fine-tunes the model to optimize this auxiliary objective, and updates the AL parameters. This renders it possible to tackle the constrained problem in Eq. 3 as a sequence of unconstrained subproblems, which can be solved via established methods [e.g., 10, 33].

CFO requires as inputs a pre-trained model π_{pre}, the number of iterations K, a minimal Lagrange multiplier $\lambda_{\min} < 0$, an initial penalty parameter $\rho_1 > 0$, a growth rate $\eta \ge 1$, and a contraction value $0 < \tau < 1$. At each iteration, CFO performs four main steps. First, it computes the Augmented Lagrangian objective f_k according to the classic augmented Lagrangian scheme for constrained optimization [25, 13] (Step 1). Then, it computes policy π_k by solving a classic KL-regularized fine-tuning problem:

$$\arg\max_{\pi} \;\; \mathbb{E}_{x\sim p_1^{\pi}}\left[f_k(x)\right] - \alpha D_{KL}(p_1^{\pi} || p_1^{\text{pre}}) \tag{7}$$

where the Augmented Lagrangian objective f_k is the one computed at the previous step (Step 2). This can be achieved by leveraging established fine-tuning schemes such as Adjoint Matching (AM) [10] (Apx. B). Next, CFO computes a proposal $\bar{\lambda}_{k+1}$ for the Lagrange multiplier via a sample-based estimate of the expected infeasibility of policy π_k, and the new Lagrange multiplier is set with a railguard (Step 3). Lastly, CFO tests whether the penalty parameter ρ should increase or not by checking the progress toward satisfying the constraint (Step 4). Ultimately, CFO returns the fine-tuned model π_K.

However, it is still unclear whether CFO is guaranteed to solve the *constrained generative optimization* problem (Eq. 3). Next, we affirm this by analyzing the convergence properties of the AL scheme [3].

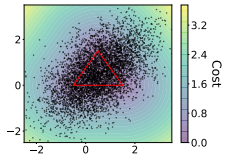
5 Constrained Generative Optimization Guarantees

We first introduce the following realistic assumption, which captures the approximate nature of typical fine-tuning schemes, along the lines of recent works [e.g., 8], and is standard in AL schemes.

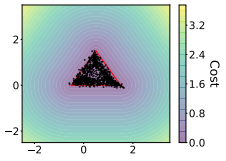
Assumption 5.1 (Solver)**.** For all $k \in \mathbb{N}$, the FINETUNINGSOLVER returns π_k such that:

$$L_{\rho_k}(\pi_k, \lambda_k) \ge L_{\rho_k}(\pi, \lambda_k) - \varepsilon_k \quad \forall \pi \tag{8}$$

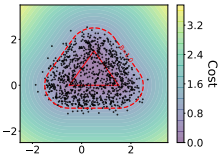
where $L_{\rho_k}(\pi_k, \lambda_k) = \mathbb{E}_{x\sim p_1^{\pi}}\left[f_k(x)\right] - \alpha D_{KL}(p_1^{\pi} || p_1^{\text{pre}})$ and the sequence $\{\varepsilon_k\} \subseteq \mathbb{R}_+$ is bounded.



(a) Pre-trained samples



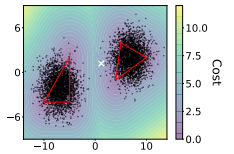
(b) CFO with $B = 0$



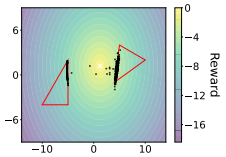
(c) CFO with $B = 1$

Experiment	$\mathbb{E}[c(x)]$	Violation Rate
Pre-trained	0.57	77.2%
CFO $B = 0$	0.003	$6.5 \pm 2.5\%$
CFO $B = 1$	0.47	$4.1 \pm 1.5\%$
Pre-trained	0.53	52.6 %
CFO	0.023	$6.1 \pm 5.9\%$

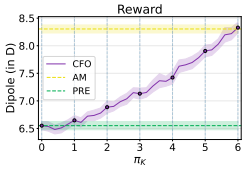
(d) Evaluation



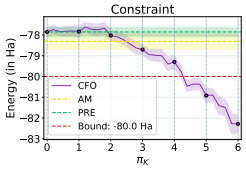
(e) Pre-trained samples



(f) CFO samples



(g) Dipole Moment



(h) Energy

Figure 2: (a-d) Constrained generation ($r = 0$) via CFO for different constraint violations B. (e-f) Constrained generative optimization via CFO. (g-h) Energy-constrained dipole moment maximization.

With this condition in place, we present two main results that establish the convergence behavior of CFO. The proofs are in Apx. G. Theorem 5.2 states that CFO finds a policy that minimizes infeasibility, i.e.: $\langle G(\pi)\rangle_+ = \langle \mathbb{E}_{x\sim p_1^\pi}[c(x)] - B\rangle_+ \geq 0$, with $\langle\cdot\rangle_+ := \max\{0, \cdot\}$.

Theorem 5.2 (Feasibility of CFO)**.** *Let $\{\pi_k\}$ be a sequence generated by Alg. 1 under Assumption 5.1 with limit $\bar{\pi}$. Then, we have $\langle G(\bar{\pi})\rangle_+ \leq \langle G(\pi)\rangle_+$, for all π, where $G(\pi) = \mathbb{E}_{x\sim p_1^\pi}[c(x)] - B$.*

By requiring a stronger condition on FINETUNINGSOLVER, namely that $\varepsilon_k \to 0$, CFO not only ensures constrained generation but also optimal reward maximization, as stated in the following.

Theorem 5.3 (Optimality of CFO)**.** *Let $\{\pi_k\}$ be a sequence generated by Alg. 1 under Assumption 5.1 with limit $\bar{\pi}$ and $\lim_{k\to\infty} \varepsilon_k = 0$. Suppose that $\langle G(\bar{\pi})\rangle_+ = 0$, then $\bar{\pi}$ is a global maximizer.*

6 Experiments

We validate the ability of CFO to solve the *constrained generative optimization* problem (Eq. 3) via two experiments: (1) a synthetic, yet illustrative, setting that enables visual interpretability, and (2) a molecular design task demonstrating CFO's relevance to real-world high-dimensional problems. While our current molecular design experiments focus on simplified constraints (e.g., energy), the framework is general and can incorporate more realistic conditions, such as ensuring candidates are synthesizable [12] or non-toxic [24]. Further experimental details are provided in Apx. D-E.

(1) Illustrative Settings. We consider two cases. First, we consider a pre-trained model density p_1^{pre} corresponding to a simple Gaussian (Fig. 2a), and evaluate the constrained generation capability of CFO, i.e., reward $r = 0$, and using a constraint c that assigns positive costs outside the red triangle (Fig. 2a). CFO can successfully steer the pre-trained model to fulfill the constraint for varying bounds $B \in \{0.0, 1.0\}$, as shown in Fig. 2b and 2c, respectively, where we report the fine-tuned density p_1^π. Next, we consider the problem of reward maximization under constraints, where p_1^{pre} is a mixture of two non-overlapping Gaussians (Fig. 2e), and the constraints c and reward r are illustrated via the color gradients in Fig. 2e and 2f, respectively. As shown in Fig. 2f, CFO can move the prior density within valid regions according to c, which is positive outside the red triangles, while maximizing the reward function. Numerical results for both experiments are reported in Tab. 2d.

(2) Molecular Design. To demonstrate the practical relevance of CFO in high-dimensional settings, we apply CFO to a molecular design. Specifically, we adapt FlowMol [11], a flow model pre-trained on GEOM Drugs [1], and maximize the dipole moment [22] as reward while ensuring constraint fulfillment. As constraints, we impose an upper bound on the total `xTB` energy (i.e., -80 Ha), to be used as a proxy for chemical stability. Both functions are computed via GNN-based predictors trained on `GFN2-xTB` [2]. In Fig. 2g-2h, we show the performance of CFO. The optimal policy π^* computed by CFO ($K = 6, N = 10$) increases the dipole moment from 6.55 Debye of the pre-trained model to 8.33 Debye. Simultaneously, π^* shifts the flow model density to generate predominantly low-energy samples, effectively achieving an expected energy of -82.28 Ha, thus satisfying the upper bound B of -80 Ha. For reference, running Adjoint Matching ($N = 60$) [10] purely for reward maximization, without enforcing the constraint, achieves a similar reward of 8.30 Debye, yet results in an expected constraint of -78.31 Ha, thus not fulfilling the constraint. Further details can be found in Apx. E.

Acknowledgments

This publication was made possible by the ETH AI Center doctoral fellowship to Riccardo De Santi. The project has received funding from the Swiss National Science Foundation under NCCR Catalysis (grant number 180544 and 225147) and NCCR Automation (grant agreement 51NF40 180545). This work was supported by an ETH Zurich Research Grant.

References

[1] Simon Axelrod and Rafael Gómez-Bombarelli. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, April 2022.

[2] Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions. *Journal of Chemical Theory and Computation*, 15(3):1652–1671, March 2019. Publisher: American Chemical Society.

[3] E. G. Birgin and J. M. Martínez. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, May 2014.

[4] Matthieu Blanke, Yongquan Qu, Sara Shamekh, and Pierre Gentine. Strictly constrained generative modeling via split augmented langevin sampling, 2025.

[5] Martin Buttenschoen, Garrett M. Morris, and Charlotte M. Deane. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

[6] Luiz F. O. Chamon, Mohammad Reza Karimi, and Anna Korba. Constrained sampling with primal-dual langevin monte carlo, 2025.

[7] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, volume 31. Curran Associates, Inc., 2018.

[8] 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*, 2025.

[9] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis, June 2021.

[10] Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control, January 2025.

[11] Ian Dunn and David Ryan Koes. Mixed Continuous and Categorical Flow Matching for 3D De Novo Molecule Generation, April 2024.

[12] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, June 2009.

[13] Michel Fortin. Minimization of some non-differentiable functionals by the Augmented Lagrangian Method of Hestenes and Powell. *Applied Mathematics and Optimization*, 2(3):236–250, September 1975.

[14] Alexandros Graikos, Nebojsa Jojic, and Dimitris Samaras. Fast constrained sampling in pre-trained diffusion models, April 2025.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020.

[16] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, July 2022.

[17] Shervin Khalafi, Dongsheng Ding, and Alejandro Ribeiro. Constrained Diffusion Models via Dual Training, November 2024.

[18] Shervin Khalafi, Ignacio Hounie, Dongsheng Ding, and Alejandro Ribeiro. Composition and alignment of diffusion models using constrained learning, 2025.

[19] Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Haorui Wang, Dongxia Wu, Aaron Ferber, Yi-An Ma, Carla P. Gomes, and Chao Zhang. Diffusion Models as Constrained Samplers for Optimization with Unknown Constraints, April 2024.

[20] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023.

[21] Hao Ma, Sabrina Bodmer, Andrea Carron, Melanie Zeilinger, and Michael Muehlebach. Constraint-aware diffusion guidance for robotics: Real-time obstacle avoidance for autonomous racing, 2025.

[22] Vladimir I. Minkin, Osip A. Osipov, and Yurii A. Zhdanov. *Dipole Moments in Organic Chemistry*. Springer US, Boston, MA, 1970.

[23] Rebecca M. Neeser, Bruno Correia, and Philippe Schwaller. FSscore: A Personalized Machine Learning-Based Synthetic Feasibility Score. *Chemistry–Methods*, 4(11), 2024.

[24] Arwa B Raies and Vladimir B Bajic. In silico toxicology: computational methods for the prediction of chemical toxicity. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 6(2):147–172, 2016.

[25] R Tyrrell Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976.

[26] 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 *The Exploration in AI Today Workshop at ICML 2025*, 2025.

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models, October 2022.

[28] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021.

[29] Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet Flow Matching with Applications to DNA Sequence Design, May 2024.

[30] Wenpin Tang. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond, March 2024.

[31] 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.

[32] Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding Reinforcement Learning-Based Fine-Tuning of Diffusion Models: A Tutorial and Review, July 2024.

[33] 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, February 2024.

[34] 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. In *Proceedings of the 41st International Conference on Machine Learning*, pages 48892–48918. PMLR, July 2024.

[35] 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.

[36] Kevin E. Wu, Kevin K. Yang, Rianne van den Berg, Sarah Alamdari, James Y. Zou, Alex X. Lu, and Ava P. Amini. Protein structure generation via folding diffusion. *Nature Communications*, 15(1):1059, February 2024.

[37] Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained diffusers for safe planning and control, 2025.

[38] 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, September 2024.

A Related Works and Conclusion

Control-based fine-tuning of flow and diffusion models. Recent works have tackled fine-tuning of diffusion and flow models to maximize rewards under KL regularization as an entropy-regularized optimal control problem [e.g., 33, 30, 34, 10]. Such methods have been successfully applied to real-world domains such as image generation [10], molecular design [34], or protein engineering [34]. These methods have also been adopted as subroutines to tackle settings beyond reward maximization, such as manifold exploration [8] or optimization of distributional objectives [26]. CFO extends fine-tuning methods for reward maximization to leverage known constraint functions and can be straightforwardly used as a plug-in oracle in more complex settings (e.g.,, exploration and distributional fine-tuning).

Constrained Generative Modeling and Optimization. Most prior work addresses constraint-aware generative modeling, developing tools for handling linear [14], differentiable [17], and black-box [19] constraints. Enforcement spans training-time dual/penalty formulations [17] and inference-time strategies such as reward-weighted denoising for non-differentiable objectives [19] and classifier or classifier-free guidance for differentiable surrogates [9, 16]. These techniques have been applied in domains such as molecular design [19] and constrained planning [21]. The closest work to ours is arguably [17], with the main difference that our setting is for post-training, i.e., at fine-tuning time, constrained generative optimization rather than a training-time scheme enforcing given constraints.

Augmented Lagrangian and Dual Methods in Constrained Sampling. Augmented Lagrangian and dual formulations turn equality and inequality constraints into auxiliary updates that run with the sampler, enabling draws from unnormalized targets while enforcing feasibility either per-sample or in expectation [18, 4, 6]. In planning and control, it has been shown that employing an augmented Lagrangian method to steer diffusion rollouts toward time-varying safety sets without requiring retraining of the base model [37]. Dual schemes similarly maintain physical invariants during sampling or data assimilation while still retaining sufficient exploration of feasible states [4]. In addition to constrained generation or sampling, CFO also performs reward-driven optimization under the augmented formulation.

Conclusion. This work tackles the problem of *constrained generative optimization* via fine-tuning pre-trained flow and diffusion models, a relevant and challenging task in discovery applications such as molecular design and protein structure generation. After proposing a constrained optimization formulation of the problem, we introduced **C**onstrained **F**low **O**ptimization, a method that turns the constrained objective into a sequence of off-the-shelf fine-tuning steps, and renders it possible to provide constrained generative optimization guarantees via the classic AL scheme. Empirical results on both illustrative settings and a molecular design task confirm the ability of CFO to steer models toward high-reward valid regions.

B Adjoint Matching [10] implementation of FINETUNINGSOLVER

To ensure completeness, below we provide pseudocode for one concrete realization of a FINETUNINGSOLVER as in Eq. 5. We describe exactly the version employed in Sec. 6, which builds on the Adjoint Matching framework [10], 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 $u^{\text{finetuned}}$ its fine-tuned counterpart. We also use $\bar{\alpha}$ to refer to the accumulated noise schedule from [15], effectively following the flow models notation introduced by Adjoint Matching [10, Sec. 5.2]. The full procedure is in Algorithm 2.

Algorithm 2 FINETUNINGSOLVER (Adjoint Matching [10]) based implementation

Input: N : number of iterations, u^{pre} : pre-trained flow vector field, α regularization coefficient as in Eq. 3, ∇f: objective function gradient, m batch size, h step size
Init: $u^{\text{finetuned}} := u^{\text{pre}}$ with parameter θ
for $n = 0, 1, 2, \ldots, N-1$ **do**
    Sample m trajectories $\{X_t\}_{0 \le t \le 1}$ via a memoryless noise schedule $\sigma(t)$ [10], e.g.,

$$\text{sample } \varepsilon_t \sim \mathcal{N}(0, I),\ X_0 \sim \mathcal{N}(0, I), \text{ then:} \tag{9}$$

$$X_{t+h} = X_t + h\left(2u_\theta^{\text{finetuned}}(X_t, t) - \frac{\bar{\alpha}_t}{\alpha_t} X_t\right) + \sqrt{h}\sigma(t)\varepsilon_t \tag{10}$$

    Use objective function gradient:

$$\tilde{a}_1 = -\frac{1}{\alpha}\nabla f(X_1)$$

    For each trajectory, solve the lean adjoint ODE, see [10, Eq. 38-39], from $t = 1$ to 0:

$$\tilde{a}_{t-h} = \tilde{a}_t + h\tilde{a}_t^\top \nabla_{X_t}\left(2u^{\text{pre}}(X_t, t) - \frac{\bar{\alpha}_t}{\alpha_t} X_t\right) \tag{11}$$

    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 [10, Eq. 37]:

$$\mathcal{L}_\theta = \sum_{t \in \{0, h, \ldots, 1-h\}} \left\| \frac{2}{\sigma(t)}\left(u_\theta^{\text{finetuned}}(X_t, t) - u^{\text{pre}}(X_t, t)\right) + \sigma(t)\tilde{a}_t \right\|^2 \tag{12}$$

    Compute the gradient $\nabla_\theta \mathcal{L}(\theta)$ and update θ.
end for
Output: Fine-tuned flow vector field $u_\theta^{\text{finetuned}}$

For further implementation details, we refer to [10, Appendix G].

C Computational Overhead

While CFO has K outer iterations, typical fine-tuning solvers [10, 34, 30] require N steps to compute the optimal iterates. This makes CFO a double loop algorithm. But in practice, we run CFO under a fixed solver-step budget of $M = K \cdot N$ for all experiments, thus keeping the total compute constant. This leads to a trade-off between the exactness of the *inner* solver and the *outer* dual updates. Increasing K reallocates budget from a more exact inner solver to more frequent updates of the Lagrange parameters, effectively making the FINETUNINGSOLVER less precise at every *outer* step.

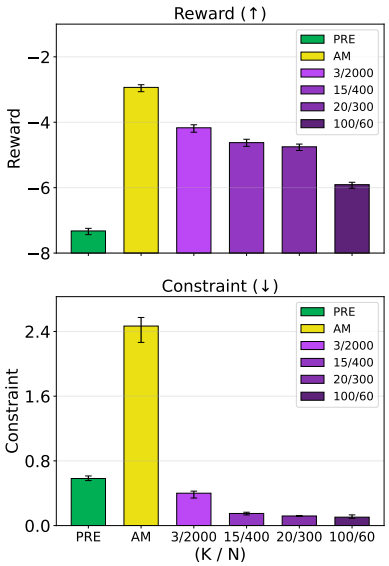


Figure 3: Reward and constraint for different values of (K/N)

To show that CFO can effectively work with an approximate fine-tuning oracle, we probe the setting shown in Fig. 2e-2f. Empirically, under a fixed budget of $M = 6000$, varying K reveals a clear trade-off between constraint satisfaction and reward. When using very few dual updates ($K = 3$), the inner solver remains highly accurate ($N = 2000$), resulting in high reward but also high expected constraint violations (0.40). Conversely, using $K = 100$ produces very frequent dual updates, but makes the inner solver approximate ($N = 60$), which almost eliminates the expected constraint violations (0.10) but substantially decreases the reward (-5.91). An intermediate configuration ($K = 20$) achieves a favorable balance, yielding both low constraint violation (0.12) and high reward (2.47), as shown in Fig. 3. Thus CFO effectively acts as a fixed-budget allocator, balancing solver precision and update frequency, where moderately inexact inner solvers allow more dual updates, and thus better constraint satisfaction. This implies that from a practical standpoint, the computational cost of CFO is comparable to that of standard fine-tuning schemes such as AM [10].

Importantly, this observation also holds for the molecular design task in Fig. 2g-2h. CFO ($K = 6$, $N = 10$) and AM ($N = 60$) have comparable computational cost, as both perform 60 gradient steps. Concretely, CFO. has a total runtime of 37.18 min and compares well to the runtime of AM with 35.35 min. This 5% increase arises from the extra sampling and constraint evaluation performed in Step 3 of Alg. 1. Thus demonstrating that CFO can operate effectively in high-dimensional domains even with an approximate oracle.

D Further Details - Illustrative Examples

The Mixture of Gaussians (Fig. 2e) is generated by

$$p(x) = \frac{1}{2}\left(\mathcal{N}\left(x \mid \begin{bmatrix}-7\\-2\end{bmatrix}, \mathbf{\Sigma}\right) + \mathcal{N}\left(x \mid \begin{bmatrix}2\\7\end{bmatrix}, \mathbf{\Sigma}\right)\right), \text{ with } \mathbf{\Sigma} = \begin{bmatrix}3 & 0\\0 & 3\end{bmatrix},$$

We sample $20k$ points ($80/20$ train/validation split) and train a MLP with 3 hidden layers, each with 256 nodes, for the vector field v. The same setting is used for the experiment on the correlated Gaussian (Fig. 2a), with:

$$p(x) = \mathcal{N}\left(x \mid \begin{bmatrix}0.5\\0.5\end{bmatrix}, \begin{bmatrix}1 & 0.5\\0.5 & 1\end{bmatrix}\right)$$

The constraint triangles have the following vertices:

1. **MoG:**

$$\triangle^{I} : \left(\begin{bmatrix}-10\\-4\end{bmatrix}, \begin{bmatrix}-5\\-4\end{bmatrix}\begin{bmatrix}-5\\2\end{bmatrix}\right) \text{ and } \triangle^{II} : \left(\begin{bmatrix}4\\-1\end{bmatrix}, \begin{bmatrix}10\\2\end{bmatrix}, \begin{bmatrix}5\\4\end{bmatrix}\right)$$

2. **Correlated Gaussian:**

$$\triangle : \left(\begin{bmatrix}-1\\-0.5\end{bmatrix}, \begin{bmatrix}1\\-0.5\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}\right)$$

(a) 15.7 D / −86.9 Ha

(b) 8.0 D / −90.7 Ha

	$\mathbb{E}[r(x)] \uparrow$	$\mathbb{E}[c(x)] \downarrow$
PRE	6.55 ± 0.07	-77.86 ± 0.22
CFO	$\mathbf{8.33 \pm 0.10}$	$\mathbf{-82.28 \pm 0.41}$
AM	$\mathbf{8.30 \pm 0.07}$	-78.31 ± 0.38

(c) Evaluation

Figure 4: (4a-4a) Drug-like molecules sampled from the fine-tuned model, together with ground-truth dipole moments (D) and energies (Ha). 4c: Numeric Evaluation of CFO ($K = 6$, $N = 10$) and AM ($N = 60$) in Fig. 2g-2h on the molecular design task (best are bold, mean and 95% CI (32 seeds)).

E Further Details - Molecular Design

Molecular Design. For the molecular design task, we fine-tune FlowMol [11]. FlowMol models the molecules as graphs $g = (X, A, C, E)$, where $X = \{x_i\}_{i=1}^N \in \mathbb{R}^{N\times 3}$ is the atom position matrix, $A = \{a_i\}_{i=1}^N \in \mathbb{R}^{N\times n_a}$ are the atom types, $C = \{c_i\}_{i=1}^N \in \mathbb{R}^{N\times n_c}$ denote the formal charges, and $E = \{e_{ij} \mid \forall i, j \in [N] | i \neq j\} \in \mathbb{R}^{N^2-N\times n_e}$ the bond order matrix. Where n_a, n_c, and n_e are the number of possible atom types, charges, and bond orders, these are categorical variables represented by one-hot vectors. We refer to [11] for the sampling of categorical and initial values. We use Gaussian sampling for the experiments in the main text on GEOM-Drugs.

GNN Details and Generalization. To verify that optimization targets the intended physical objective rather than exploiting the surrogate, we evaluate the ground-truth `xTB` values for every molecule sampled during the execution of CFO and compare their properties to the GNN predictions. For the energy (used as a constraint), surrogate predictions are essentially indistinguishable from `xTB`, indicating faithful approximation within the explored region. For the dipole moment (the maximization target), the surrogate systematically underestimates the true xTB values by 10%, yet the two remain strongly correlated and move in lockstep throughout the fine-tuning. Consequently, improvements under the surrogate translate to larger gains under `xTB`. Overall, these checks indicate that CFO does not exploit model artifacts and remains within the training distribution.

F Parameters

Discussion of the most important Hyperparameter of CFO and FINETUNINGSOLVER:

- **Initial penalty ρ_{init}.** Larger ρ_{init} penalizes constraint violations more strongly, thus effectively reducing early exploration inside the base distribution. Smaller ρ_{init} does the opposite.
- **Penalty growth rate $\eta \geq 1$.** Controls the penalty growth across updates. Larger η accelerates enforcement and thus can reduce exploration of high-reward regions. Smaller η tightens feasibility more gradually, allowing for early reward progress, but potentially slower constraint satisfaction.
- **Contraction rate $\tau \in (0, 1)$.** Determines when the penalty parameter ρ is updated. Smaller τ triggers more frequent updates, values near one update conservatively.
- **Multiplier lower bound $\lambda_{\min} < 0$.** Safeguards the Lagrange multiplier via clipping. Smaller $\lambda_{\min}$ permits larger corrective signals of the offset, see Sec. 4. If set to a large negative value, its influence on the final output is typically small, since $\lambda_{\min}$ is not achieved.
- **FINETUNINGSOLVER regularization α.** Trade-off between staying close to the base distribution and reallocating mass. Larger α enforces stronger KL-regularization of the policy. A smaller α allows greater deviation from the base policy.
- **Sampling for constraint estimation (sample count/batch size).** Larger samples reduce estimator variance, stabilizing updates and improving feasibility. If the sample size is too small, this yields volatile or biased estimates that can steer CFO to off-target solutions.

Table 1: Hyperparameters for CFO and Adjoint Matching

	SG(2a)	**MoG**(2e)	**MD-GEOM**(2g-2h)
CFO			
Lagrangian Updates K	20	20	6
ρ_{init}	0.5	0.5	1.0
η	1.25	1.25	1.25
τ	0.99	0.99	0.99
λ_{min}	-50.0	-50.0	-50.0
Adjoint Matching			
$(1/\alpha)$	1e5	1e5	50
Number of Iterations N	300	300	10
Effective Batch Size	256	256	20
Clip Grad Norm	0.7	0.7	0.4
Learning Rate	5e-6	5e-6	5e-6
Integration Steps	40	40	40
Total Steps	6000	6000	60

G Proofs

Before we present a proof of the theorems in Sec. 5. We will transform the main problem in Eq. 3 to a simpler form. First, we recall that the policy π is a vector field. It has been shown before that the ODE in Eq. 1 and a stochastic differential equation (SDE) of the form

$$dX_t = b(X_t, t)dt + \sigma(t)dB_t,\ X_0 \sim p_0, \tag{13}$$

with drift $b : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$, diffusion coefficient $\sigma : [0,1] \to \mathbb{R}_{\geq 0}$ and Brownian motion B_t induce the same marginals $\{p_t\}$. For an exact definition of b and a proof of this statement, we refer to [10]. Controlling this SDE can be done by adjusting the drift as follows [10]:

$$dX_t = (b(X_t, t) + \sigma(t)u(X_t, t))\, dt + \sigma(t)dB_t,\ X_0 \sim p_0,$$

where $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is a control vector field, this means the pre-trained model is a controlled model with $u \equiv 0$. With these notational changes, we reformulate the optimization problem in Eq. 3 in terms of the controlled diffusion process $\mathbf{X}^u \sim p^u$:

$$\begin{aligned} \max_{u \in \mathcal{U}} \quad & \mathbb{E}_{\mathbf{X}^u \sim p^u}\left[r(X_1)\right] - \alpha D_{KL}(p_1^u(\cdot)||p_1^{\text{pre}}(\cdot)) \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{X}^v \sim p^v}\left[c(X_1)\right] \leq B \end{aligned} \tag{14}$$

Eq. 14 may seem the same as Eq. 3, but it is in terms of a diffusion process. This way we can calculate the KL efficiently, see [Eq. 18, 10], by using Girsanov's theorem, which gives the relationship between the control process u and the KL-Divergence:

$$D_{\text{KL}}(p^u(\mathbf{X}|X_0)\ ||\ p^{\text{pre}}(\mathbf{X}|X_0)) = \mathbb{E}_{\mathbf{X}^u \sim p^u}\left[\int_0^1 \frac{1}{2}\left\|u(X_t, t)\right\|^2 dB_t\right]$$

Meaning if both processes have the same initial value X_0, the KL divergence between the controlled and uncontrolled process is equal to the expected value of the squared norm of the control u [10, 33, 30]. This dependence on the initial value can be dropped when using a specific noise schedule [10]. Recalling that marginals at time t are $p_t(x)$, i.e. $X_t \sim p_t(x)$, then we can equivalently write the optimization problem as:

$$\begin{aligned} \max_{u \in \mathcal{U}} \quad & \mathbb{E}_{\mathbf{X}^u \sim p^u}\left[r(X_1)\right] - \alpha \mathbb{E}\left[\int_0^1 \frac{1}{2}\|u(X_t^u, t)\|^2 dt\right] \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{X}^u \sim p^u}\left[r(X_1)\right] \leq B \end{aligned}$$

Where the expectation is taken over the controlled process $\mathbf{X}^u$. For numerical optimization, we now assume that the control u is a parametric model, typically a neural network, with parameters θ. The resulting optimization problem is then:

$$\begin{aligned} \max_{\theta \in \mathbb{R}^m} \quad & F(\theta) := F_r(\theta) - \alpha F_{KL}(\theta) \\ & = \mathbb{E}_{x \sim p_1^{u_\theta}}\left[r(x)\right] - \alpha \mathbb{E}\left[\int_0^1 \frac{1}{2}\|u_\theta(X_t, t)\|^2 dt\right] \\ \text{s.t.} \quad & G(\theta) := \mathbb{E}_{x \sim p_1^{u_\theta}}\left[c(x)\right] - B \leq 0 \end{aligned} \tag{15}$$

For some function $F : \mathbb{R}^m \to \mathbb{R}$ and function $G : \mathbb{R}^m \to \mathbb{R}$. This is finite-dimensional optimization over θ.

Next, we present a proof that Alg. 1 can find a parameterized policy π_θ, with $\theta \in \mathbb{R}^m$ that minimizes the infeasibility while maximizing the reward. The proof is mostly the same as in "Practical Augmented Lagrangian Methods for Constrained Optimization" [3, Chapter 5].

The augmented Lagrangian objective in Eq. 7 becomes:

$$L_\rho(\theta, \lambda) = F(\theta) - \frac{\rho}{2}\left[\max\left(0, G(\theta) - \frac{\lambda}{\rho}\right)\right]^2 \tag{16}$$

where $\lambda \in \mathbb{R}_{\leq 0}$ is the Lagrange multiplier, $\rho > 0$ is a penalty parameter.

With this notation, the assumption on the solver becomes:

Assumption G.1 (Solver)**.** For all $k \in \mathbb{N}$, we obtain u such that:

$$L_{\rho_k}(\theta_k, \lambda_k) \geq L_{\rho_k}(\theta, \lambda_k) - \varepsilon_k \quad \forall\ \theta \in \mathbb{R}^m \tag{17}$$

where the sequence $\{\varepsilon_k\} \subseteq \mathbb{R}_+$ is bounded.

This corresponds to Assumption 5.1 from [3]. Assumption G.1 states that the solver can find an approximate maximizer of the subproblem.

Next we state and prove the main result for the algorithm. Namely, in the limit, we obtain a minimizer of the infeasibility measure.

Theorem G.2 (Feasibility of **C**onstrained **F**low **O**ptimization)**.** *Let $\{\theta_k\}$ be a sequence generated by Alg. 1 under the solver Assumption G.1. Let $\bar{\theta}$ be a limit of the sequence $\{\theta_k\}$. Then, we have:*

$$\left\langle G(\bar{\theta})\right\rangle_+ \leq \langle G(\theta)\rangle_+ \quad \forall \theta \in \mathbb{R}^m, \tag{18}$$

where $G(\theta) := \mathbb{E}_{x\sim p_1^{u_\theta}}[c(x)] - B \leq 0$ and $\langle\cdot\rangle_+ := \max\{0, \cdot\}$.

Proof. By definition $\mathbb{R}^m$ is closed and $\theta_k \in \mathbb{R}^m$ thus $\bar{\theta} \in \mathbb{R}^m$. We consider two cases: $\{\rho_k\}$ bounded and $\rho_k \to \infty$. First we assume $\{\rho_k\}$ is bounded, there exists k_0 such that $\rho_k = \rho_{k_0}$ for all $k \geq k_0$. Therefore, for all $k \geq k_0$, the upper bracket in Step 4 of Alg. 1 holds. This implies that $|V_k| \to 0$, so $\langle G(\theta_k)\rangle_+ \to 0$. Thus, the limit point is feasible.

Now, assume that $\rho_k \to \infty$. Let $K \subseteq \mathbb{N}$ be such that:

$$\theta_k \to \bar{\theta} \text{ for } k \in K \text{ and } k \to \infty$$

Assume by contradiction that there exists $\theta \in \mathbb{R}^d$ such that

$$\left\langle G(\bar{\theta})\right\rangle_+^2 > \langle G(\theta)\rangle_+^2$$

By the continuity of G, the boundedness of $\{\lambda_k\}$, and the fact that $\rho_k \to \infty$, there exists $c > 0$ and $k_0 \in \mathbb{N}$ such that for all $k \in K, k \geq k_0$:

$$\left\langle G(\theta_k) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2 > \left\langle G(\theta) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2 + c$$

Therefore, for all $k \in K, k \geq k_0$:

$$F(\theta_k) - \frac{\rho_k}{2}\left[\left\langle G(\theta_k) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] < F(\theta) - \frac{\rho_k}{2}\left[\left\langle G(\theta) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] - \frac{\rho_k c}{2} + F(\theta_k) - F(\theta)$$

Since $\lim_{k\in K}\theta_k = \bar{\theta}$, the continuity of F, and the boundedness of $\{\varepsilon_k\}$, there exists $k_1 \geq k_0$ such that, for $k \in K$ $k \geq k_1$:

$$\frac{\rho_k c}{2} - F(\theta_k) + F(\theta) > \varepsilon_k$$

Therefore,

$$F(\theta_k) - \frac{\rho_k}{2}\left[\left\langle G(\theta_k) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] < F(\theta) - \frac{\rho_k}{2}\left[\left\langle G(\theta) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] - \varepsilon_k$$

for $k \in K, k \geq k_1$. This contradicts Assumption G.1. □

Theorem G.2 and its proof correspond to [3, Sec. 5.1]. Theorem G.2 establishes that Alg. 1, under the iterates given in Assumption G.1, identifies minimizers of the infeasibility, i.e.,

$$\langle G(\theta)\rangle_+ := \left\langle \mathbb{E}_{x\sim p_1^{u_\theta}}[c(x)] - B \leq 0\right\rangle_+.$$

Consequently, if the original optimization problem is feasible, then every limit point of the sequence produced by the algorithm is also feasible.

Next, we will see that, assuming that ε_k tends to zero, it is possible to prove that, in the feasible case, the algorithm asymptotically finds a global maximizer of the problem in Eq. 3.

Theorem G.3 (Optimality of **C**onstrained **F**low **O**ptimization)**.** *Let $\{\theta_k\} \subset \mathbb{R}^d$ be a sequence generated by Alg. 1 under Assumption G.1 and $\lim_{k\to\infty}\varepsilon_k = 0$. Let $\bar{\theta} \in \mathbb{R}^m$ be a limit of the sequence $\{\theta_k\}$. Suppose that $\langle G(\theta)\rangle_+ = 0$, then $\bar{\theta}$ is a global maximizer of Eq. 3.*

Proof. Let $K \subseteq \mathbb{N}$ be such that.

$$\theta_k \to \bar{\theta} \text{ for } k \in K \text{ and } k \to \infty$$

By assumption, the problem is feasible, thus, by Theorem G.2, we have that $\bar{\theta}$ is feasible. Let $\theta \in \mathbb{R}^m$ be such that $G(\theta) \leq 0$. By the definition of the algorithm, we have that

$$F(\theta_k) - \frac{\rho_k}{2}\left[\left\langle G(\theta_k) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] \geq F(\theta) - \frac{\rho_k}{2}\left[\left\langle G(\theta) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] - \varepsilon_k \tag{19}$$

for all $k \in \mathbb{N}$, as well as by assumption $G(\theta) \leq 0$, we have that

$$\left\langle G(\theta) - \frac{\lambda_k}{\rho_k} \right\rangle_+^2 \leq \left(\frac{\lambda_k}{\rho_k}\right)^2. \tag{20}$$

We again consider the two cases: $\rho_k \to \infty$ and $\{\rho_k\}$ bounded.

In the first case, we assume $\rho_k \to \infty$. By Eq. 19 and Eq. 20, we have

$$F(\theta_k) \geq F(\theta_k) - \frac{\rho_k}{2}\left[\left\langle G(\theta_k) - \frac{\lambda_k}{\rho_k}\right\rangle_+^2\right] \geq F(\theta) - \frac{(\lambda_k)^2}{2\rho_k} - \varepsilon_k.$$

Taking limits for $k \in K$, and using that $\theta_k \to \bar{\theta}$, we have that $\lim_{k\in K}(\lambda_k)^2/\rho_k = 0$ and $\lim_{k\in K}\varepsilon_k = 0$, by the continuity of F and the convergence of θ_k, we get

$$F(\bar{\theta}) \geq F(\theta).$$

Since θ is an arbitrary feasible element of $\mathbb{R}^m$, $\bar{\theta}$ is a global optimizer.

For the second case, we assume $\{\rho_k\}$ is bounded, there exists $k_0 \in \mathbb{N}$ such that $\rho_k = \rho_{k_0}$ for all $k \geq k_0$. Therefore, by Assumption G.1, Eq. 19 holds for all $k \geq k_0$, and Eq. 20 holds with $\rho = \rho_{k_0}$. Thus,

$$F(\theta_k) - \frac{\rho_{k_0}}{2}\left[\left\langle G(\theta_k) - \frac{\lambda_k}{\rho_{k_0}}\right\rangle_+^2\right] \geq F(\theta) - \frac{(\lambda_k)^2}{2\rho_{k_0}} - \varepsilon_k.$$

for all $k \geq k_0$. Let $K_1 \subseteq \mathbb{N}$ and $\lambda^* \in \mathbb{R}_{\leq 0}$ be such that: $\lim_{k\in K_1}\lambda_k = \lambda^*$. By the feasibility of $\bar{\theta}$, taking limits in the inequality above for $k \in K_1$, we get

$$F(\bar{\theta}) - \frac{\rho_{k_0}}{2}\left[\left\langle G(\bar{\theta}) - \frac{\lambda^*}{\rho_{k_0}}\right\rangle_+^2\right] \geq F(\theta) - \frac{(\lambda^*)^2}{2\rho_{k_0}} - \varepsilon_k. \tag{21}$$

Now, if $G(\bar{\theta}) = 0$, since $\lambda^*/\rho_{k_0} \geq 0$, we have that

$$\left\langle G(\bar{\theta}) - \frac{\lambda^*}{\rho_{k_0}}\right\rangle_+^2 = \left(\frac{\lambda^*}{\rho_{k_0}}\right)^2$$

Therefore, by Eq. 21,

$$F(\bar{\theta}) - \frac{\rho_{k_0}}{2}\left[\left\langle G(\bar{\theta}) - \frac{\lambda^*}{\rho_{k_0}}\right\rangle_+^2\right] \geq F(\theta) - \frac{(\lambda^*)^2}{2\rho_{k_0}}. \tag{22}$$

But, by condition in Step 4 for Alg. 1 (definition of V_k, and interaction with ρ_k), we have $\lim_{k\to\infty}\min\{G(\theta_k), -\lambda^*/\rho_{k_0}\} = 0$. Therefore, if $G(\bar{\theta}) < 0$, we necessarily have that $\lambda^* = 0$. Therefore, Eq. 22 implies that $F(\bar{\theta}) \geq F(\theta)$. Since θ is an arbitrary feasible element of $\mathbb{R}^m$, $\bar{\theta}$ is a global optimizer. □

We want to make two remarks about Theorem G.3: the first is that having access to such an solver is difficult and in practice rarely the case. Secondly, we refer to [3, Sec. 5.2] for a discussion about the sets K and K_1, how they are connected to the convexity of F and G.