Constrained Molecular Generation via Sequential Flow Model Fine-Tuning

Sven Gutjahr [*1] **Riccardo De Santi** [*1 2 3] **Luca Schaufelberger** [*3 4] **Kjell Jorner** [3 4] **Andreas Krause** [1 2 3]

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 or protein engineering. 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, in this work, we first present a rigorous formulation for constrained generation. Then, we introduce **A**ugmented **L**agrangian **F**lows **F**ine-tuning (ALF^2), an augmented Lagrangian method that renders 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 (Ho et al., 2020; Song et al., 2021; 2022) and flow models (Lipman et al., 2023), have led to state-of-the-art performances in several biological tasks, including generating protein structures (Wu et al., 2024), drug-like molecules (Dunn & Koes, 2024), and DNA sequences (Stark et al., 2024) among others.

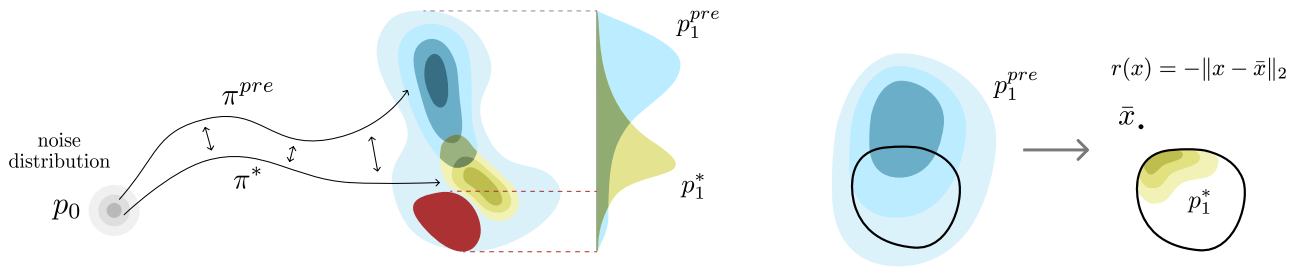
These foundation models excel at capturing complex data distributions and generating realistic samples. However, approximately sampling from the data distribution is insufficient for most real-world discovery applications, where one typically wishes to generate candidates maximizing task-specific properties, 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 (e.g., Domingo-Enrich et al., 2025; Uehara et al., 2024b; Tang, 2024). However, transporting the generative model density towards a region of higher rewards can render regularization insufficient to ensure validity constraints are fulfilled (Uehara et al., 2024a). In chemistry and biology, these constraints could, for example, constitute the physical validity of generated docking poses (Buttenschoen et al., 2024), toxicity, chemical synthesizability of drug candidates (Ertl & Schuffenhauer, 2009; Neeser et al., 2024) or sequence-driven aggregation risk (Fernandez-Escamilla et al., 2004). Driven by this observation, we aim to answer the following question in this work:

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 generation via fine-tuning*, extending current fine-tuning formulations to the constrained case (Sec. 3). Next, we introduce **A**ugmented **L**agrangian **F**lows **F**ine-tuning (ALF^2), a fine-tuning method based on the augmented Lagrangian (AL) scheme (Birgin & Martínez, 2014), which progressively adjusts the model to navigate the trade-off between reward maximization and constraint fulfillment thus enabling controllable and reliable sample generation (Sec. 4). The proposed principled algorithm renders possible to transport classic constrained optimization guarantees of the AL method to generative model finetuning (Sec. 5). Ultimately, we demonstrate the effectiveness of ALF^2 on both synthetic, yet illustrative, settings, and on a molecular design task. In the latter case, we utilize ALF^2 to generate molecular structures with large molecular dipole moment (Minkin

[*]Equal contribution [1]Department of Computer Science, ETH Zurich, Zurich, Switzerland [2]ETH AI Center, Zurich, Switzerland [3]NCCR Catalysis, Switzerland [4]Institute of Chemical and Bioengineering, Department of Chemistry and Applied Biosciences, ETH Zurich, Zurich CH-8093, Switzerland. Correspondence to: Riccardo De Santi, Luca Schaufelberger <rdesanti@ethz.ch, schaluca@ethz.ch>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.



(a) Constrained generation via fine-tuning problem. In red, a high-cost area.

(b) Adaptation to low-cost area within black line.

Figure 1. (1a) Pre-trained and fine-tuned policies inducing densities p_1^{pre} and optimal density p_1^* w.r.t. reward r increasing downwards, and red high-cost area. (1b) Pre-trained model p_1^{pre} adapts into p_1^* to maximize r and stay within the low-cost area inside the black line.

et al., 1970) while satisfying energetic constraints (Sec. 6).

Our contributions. We present the following contributions:

- We present the *constrained generation via fine-tuning* problem for flow and diffusion models (Sec. 3).
- We introduce **A**ugmented **L**agrangian **F**lows **F**ine-tuning (ALF^2), an augmented Lagrangian-based fine-tuning method for flow and diffusion models (Sec. 4).
- We provide constrained optimization guarantees based on the AL scheme for the proposed method (Sec. 5).
- We evaluate ALF^2'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 generative models, aim to approximately sample from a data distribution p^{data}, by transforming samples $X_0 \sim p^{\text{init}}$ from an initial distribution into $X_1 \sim p^{\text{data}}$ (Chen et al., 2018; Lipman et al., 2023). A flow is a map $\psi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ denoted by $\psi_t(x)$. In particular, the flow ψ can be defined by a velocity field $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, via the following *flow ODE*:

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

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: $X_t = \psi_t(X_0) \sim p_t$.

Pre-trained Flow Models as an RL policy. We present finite-horizon continuous-time reinforcement learning (RL) (Wang et al., 2020; Treven et al., 2023; Zhao et al., 2024) as a specific case of stochastic optimal control. Let $\mathcal{X}$ be a state space and $\mathcal{A}$ an action space, with the transition dynamics governed by the following ODE, where $a_t \in \mathcal{A}$ is a selected action:

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = a_t(\psi_t(x)) \quad (2)$$

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

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 $a_t^{pre} = \pi^{pre}(X_t, t)$ (De Santi et al., 2025). Therefore, we can express the flow ODE induced by a pre-trained flow model by replacing a_t with a^{pre} in Eq. (2), and denote the pre-trained model by its policy π^{pre}, which induces a marginal density $p_1^{pre} := p_1^{\pi^{pre}}$ approximating p_{data}.

3. Problem Setting

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

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = a_t^{\text{fine}}(\psi_t(x)), \text{ with } a_t^{\text{fine}} = \pi^*(x_t, t) \quad (3)$$

The fine-tuned model should induce a distribution $p_1^* := p_1^{\pi^*}$ maximizing the expected value of a property of interest, while preserving prior information from the pre-trained model π^{pre}, and satisfying arbitrary constraints violation. This *constrained generation via fine-tuning* problem is illustrated in Fig. 1, and formalized it in the following:

Constrained Generation via Fine-Tuning

$$\begin{aligned} \arg\max_{\pi} \ & \mathbb{E}_{x\sim p_1^\pi}[r(x)] - \alpha D_{KL}(p_1^\pi || p_1^{\text{pre}}) \\ \text{s.t. } & \mathbb{E}_{x\sim p_1^\pi}[c(x)] \le B \end{aligned} \quad (4)$$

Where $r : \mathcal{X} \to \mathbb{R}$ and $c : \mathcal{X} \to \mathbb{R}$ are respectively scalar reward and constraint functions, $\alpha \in \mathbb{R}$ determines the KL-regularization strength, and $B \in \mathbb{R}$ is a controllable parameter imposing the degree of permissible constraint violation.

Crucially, the *constrained generation via fine-tuning* problem in Eq. (4) renders possible to compute a fine-tuned flow model that maximizes the given reward function r, while attaining at most a constraint violation of arbitrary value B.

The trade-off between reward r and constraint c becomes particularly important in challenging settings like the one shown in Fig. 1b, where the reward-maximizing region lies outside the set of valid data points. This is a common

situation when leveraging learned reward functions as property predictors (Uehara et al., 2024a). In such cases, naive optimization may drive the model toward high-reward samples that violate the constraint.

In the next section, we propose an algorithm that can tackle the constrained generation problem in Eq. (4) by leveraging as a subroutine any unconstrained fine-tuning method (e.g., Domingo-Enrich et al., 2025; Uehara et al., 2024b).

4. Algorithm

In the following, we present **A**ugmented **L**agrangian **F**lows **F**ine-tuning (ALF^2), see Alg. 1, which reduces the constrained generation problem in Eq. (4) to sequentially fine-tuning the pre-trained model π_{pre} for a specific sequence of auxiliary objectives determined via an augmented Lagrangian scheme (Rockafellar, 1976; Fortin, 1975).

At each iteration $k \in \{1, \dots, K\}$, ALF^2 first fine-tunes π_{k-1} into π_k to maximize an auxiliary Augmented Lagrangian objective f_k. Then, it uses the current model π_k to estimate the AL objective f_{k+1} used at the next iteration. This renders possible to tackle the constrained problem in Eq. (4) as a sequence of unconstrained subproblems, which can be solved via established methods (e.g., Domingo-Enrich et al., 2025; Uehara et al., 2024b).

ALF^2 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 $\eta \geq 1$, and a contraction value $0 < \tau < 1$. At each iteration, ALF^2 performs four main steps. First, it computes the Augmented Lagrangian objective f_k according to the classic augmented Lagrangian scheme for constrained optimization (Rockafellar, 1976; Fortin, 1975) (Step 1). Then, it computes policy π_k by solving a classic KL-regularized fine-tuning problem:

$$\arg\max_{\pi} \quad \mathbb{E}_{x \sim p_1^{\pi}}[f_k(x)] - \alpha D_{KL}(p_1^{\pi} || p_1^{\text{pre}}) \tag{5}$$

where the AL 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) (Domingo-Enrich et al., 2025), of which we report a possible implementation in Apx. B. Next, ALF^2 computes a proposal λ_{k+1} for the Lagrange multiplier via a sample-based estimate of the expected infeasibility of policy π_k (Step 3). Lastly, ALF^2 tests whether the penalty parameter ρ should grow or not, by checking the progress in terms of fulfillment of the constraint, and the new Lagrange multiplier is set (Step 4). Ultimately, ALF^2 returns the fine-tuned model π_K.

Nevertheless, it is still unclear whether ALF^2 is guaranteed to solve the constrained generation problem in Eq. (4), which demands reward optimality and arbitrary constraint satisfaction. In the next section, we provide an affirmative answer by presenting an analysis based on the convergence properties of the AL scheme (Birgin & Martínez, 2014).

Algorithm 1 Augmented Lagrangian Flows Fine-Tuning

Input: π_{pre}: pre-trained model, K: number of iterations, $\lambda_{\min} < 0$: min. Lagrange multiplier, $\rho_1 > 0$: initial penalty parameter, $\eta \geq 1$: growth rate, $0 < \tau < 1$: contraction value
Init: Set initial Lagrange multiplier $\bar{\lambda}_1 = 0$
for $k = 1, 2, \dots, K$ **do**
  Step 1: Update fine-tuning AL objective:

$$f_k(x) := r(x) - \frac{\rho_k}{2}\left[\max\left(0, c(x) - B - \frac{\bar{\lambda}_k}{\rho_k}\right)\right]^2 \tag{6}$$

  Step 2: Compute π_k via fine-tuning:

$$\pi_k \leftarrow \text{FINETUNINGSOLVER}(f_k, \pi_{\text{pre}}) \tag{7}$$

  Step 3: Compute Lagrange multiplier proposal:

$$\lambda_{k+1} \leftarrow \min\left\{0, \bar{\lambda}_k - \rho_k\left(\mathbb{E}_{x \sim p_1^{\pi_k}}[c(x)] - B\right)\right\}$$

  Step 4: Set

$$V_k = \min\left\{\mathbb{E}_{x \sim p_1^{\pi_k}}[c(x)] - B, -\bar{\lambda}_k/\rho_k\right\} \tag{8}$$

  and calculate new penalty and Lagrange multiplier:

$$\rho_{k+1} = \begin{cases} \rho_k, & \text{if } k = 1 \text{ or } V_k \leq \tau V_{k-1}, \\ \eta\rho_k, & \text{otherwise} \end{cases} \tag{9}$$

$$\bar{\lambda}_{k+1} = \max\{\lambda_{k+1}, \lambda_{\min}\} \tag{10}$$

end for
Return: π_K

5. Constrained Generation Guarantees

Before presenting convergence guarantees for ALF^2, we introduce the following realistic assumption for FINETUNINGSOLVER in Alg. 1.

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

$$L_{\rho_k}(\pi_k, \bar{\lambda}_k) \geq L_{\rho_k}(\pi, \bar{\lambda}_k) - \varepsilon_k \quad \forall \pi \tag{11}$$

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

This assumption is standard in the AL framework and captures the approximate nature of typical fine-tuning schemes, along the lines of recent works (e.g., De Santi et al., 2025).

With this condition in place, we present two main results that establish the convergence behavior of ALF^2, with proofs in Apx. D. Theorem 5.2 states that ALF^2 finds a policy that minimizes infeasibility, i.e.

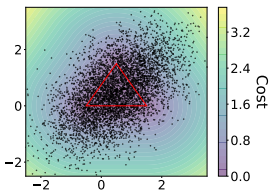
$$\langle G(\pi) \rangle_+ = \left\langle \mathbb{E}_{x \sim p_1^{\pi}}[c(x)] - B \right\rangle_+ \geq 0.$$

Theorem 5.2 (Feasibility of ALF^2)**.** *Let $\{\pi_k\}$ be a sequence generated by Alg. 1 under the oracle Assumption 5.1. Let $\bar{\pi}$ be a limit of the sequence $\{\pi_k\}$. Then, we have:*

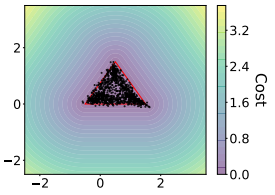
$$\langle G(\bar{\pi}) \rangle_+ \leq \langle G(\pi) \rangle_+ \quad \forall \pi \tag{12}$$

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

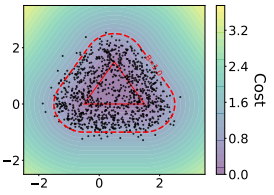
Interestingly, by requiring a stronger condition on FINETUNINGSOLVER, namely that $\varepsilon_k \to 0$, ALF^2 not only ensures



(a) Pre-trained samples



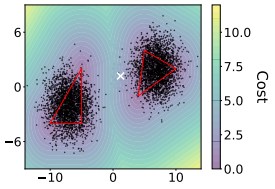
(b) ALF^2 with $B = 0$



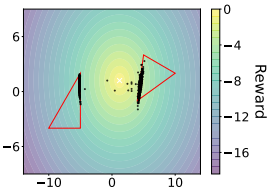
(c) ALF^2 with $B = 1$

Experiment	$\mathbb{E}[c(x)]$	Violation Rate
Pre-trained	0.57	77.2%
ALF^2 $B = 0$	0.003	$6.5 \pm 2.5\%$
ALF^2 $B = 1$	0.47	$4.1 \pm 1.5\%$
Pre-trained	0.53	52.6 %
ALF^2	0.023	$6.1 \pm 5.9\%$

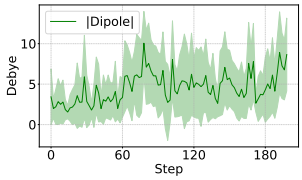
(d) Numerical Evaluation



(e) Pre-trained samples



(f) ALF^2 samples



(g) Dipole moment

Property	Stage	Value
Dipole moment	Pre	$3.43 \pm 3.45D$
	ALF^2	$8.66 \pm 4.50D$
Energy	Pre	$-16.72 \pm 2.48E_{\mathrm{h}}$
	ALF^2	$-39.40 \pm 4.01E_{\mathrm{h}}$
Violations	Pre	65 %
	ALF^2	0 %

(h) Numerical Evaluation

Figure 2. (top) Constrained generation via ALF^2 for different thresholds B of constraint violation. (bottom) Constrained reward maximization via ALF^2 (left), and energy-constrained dipole moment maximization for molecular design (right).

constrained generation but also optimal reward maximization, as stated in the following.

Theorem 5.3 (Optimality of ALF^2)**.** *Let $\{\pi_k\}$ be a sequence generated by Alg. 1 under Assumption 5.1 and $\lim_{k\to\infty} \varepsilon_k = 0$. Let $\bar{\pi}$ be a limit of the sequence $\{\pi_k\}$. Suppose that $\langle G(\bar{\pi})\rangle_+ = 0$, then $\bar{\pi}$ is a global maximizer.*

6. Experiments

We validate the ability of ALF^2 to solve the *constrained generation* problem in Eq. (4) via two type of experiments: (1) a synthetic, yet illustrative, setting that enables visual interpretability, and (2) a molecular design task demonstrating ALF^2 '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 (Ertl & Schuffenhauer, 2009) or non-toxic (Raies & Bajic, 2016). Further experimental details are provided in Apx. C.

(1) Illustrative Settings. In this experiment, we consider two cases. First, we consider a pre-trained model density p_1^{pre} corresponding to a simple Gaussian (see Fig. 2a), and evaluate the constrained generation capability of ALF^2 without an additional reward r, and using a constraint c that assigns positive costs outside the red triangle in Fig. 2a. ALF^2 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 (see Figure 2e), and the constraints c and reward r are illustrated via the color gradients in Fig. Figure 2e and Fig. 2f, respectively. As shown in Fig. 2f, ALF^2 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 Fig. 2d.

(2) Molecular Design. We illustrate the potential of ALF^2 for ensuring constraint satisfaction in molecular design on a proof-of-principle task. To this end, we fine-tune FlowMol (Dunn & Koes, 2024), a flow model pre-trained on QM9 (Ramakrishnan et al., 2014) data to generate small molecules, and employ `dxtb` (Friede et al., 2024) to estimate differentiable rewards and constraints. In particular, we aim to maximize the norm of the dipole moment (Minkin et al., 1970) while constraining the total energy to remain below $-18E_{\mathrm{h}}$. As shown in Tab. 2h, the pre-trained model π^{pre} violates such constraint with 65 % of samples. On the contrary, the model fine-tuned via ALF^2 can successfully achieve zero constraint violation (30 Monte Carlo samples, all below the threshold) while increasing the average norm of the dipole moment from 3.43 ± 3.45 to 8.66 ± 4.50, as shown in Fig 2g. While we highlight the proof-of-principle nature of the optimization setup, given the limited diversity of QM9 and performing optimization on non-relaxed geometries, this setup demonstrates the potential of ALF^2 for more complex applications, such as drug discovery (Schneuing et al., 2024).

7. Conclusion

This work tackles the problem of constrained generation 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 Augmented Lagrangian Flows Fine-Tuning, a method that turns the constrained objective into a sequence of fine-tuning steps and renders possible to provide constrained generation guarantees via the classic AL scheme. Empirical results on both illustrative settings and a molecular design task confirm the ability of ALF^2 to steer models toward high-reward valid regions.

8. Impact Statement

This work presents research aimed at advancing the field of generative modeling and molecular design. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

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

Birgin, E. G. and Martínez, J. M. *Practical Augmented Lagrangian Methods for Constrained Optimization*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, May 2014. ISBN 978-1-61197-335-8.

Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., and Chen, H. Application of generative autoencoder in de novo molecular design. *Molecular informatics*, 37(1-2): 1700123, 2018.

Brinck, T. and Liljenberg, M. The use of quantum chemistry for mechanistic analyses of sear reactions. *Arene Chemistry: Reaction Mechanisms and Methods for Aromatic Compounds*, pp. 83–105, 2015.

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

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Advances in neural information processing systems*, volume 31. Curran Associates, Inc., 2018.

De Santi, R., Vlastelica, M., Hsieh, Y.-P., Shen, Z., He, N., and Krause, A. Provable maximum entropy manifold exploration via diffusion models. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2025.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs on Image Synthesis, June 2021.

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Q. Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control, January 2025.

Dunn, I. and Koes, D. R. Mixed Continuous and Categorical Flow Matching for 3D De Novo Molecule Generation, April 2024.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, June 2009. ISSN 1758-2946.

Fernandez-Escamilla, A.-M., Rousseau, F., Schymkowitz, J., and Serrano, L. Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins. *Nature biotechnology*, 22(10):1302–1306, 2004.

Fortin, M. 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. ISSN 1432-0606.

Friede, M., Hölzer, C., Ehlert, S., and Grimme, S. dxtb—An efficient and fully differentiable framework for extended tight-binding. *The Journal of Chemical Physics*, 161(6), 2024.

Gao, W., Luo, S., and Coley, C. W. Generative artificial intelligence for navigating synthesizable chemical space. 2024.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Graikos, A., Jojic, N., and Samaras, D. Fast constrained sampling in pre-trained diffusion models, April 2025.

Guo, J. and Schwaller, P. Directly optimizing for synthesizability in generative molecular design using retrosynthesis models. *Chemical Science*, 16(16):6943–6956, 2025.

Ho, J. and Salimans, T. Classifier-Free Diffusion Guidance, July 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising Diffusion Probabilistic Models, December 2020.

Jensen, J. H. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572, 2019.

Khalafi, S., Ding, D., and Ribeiro, A. Constrained Diffusion Models via Dual Training, November 2024.

Kong, L., Du, Y., Mu, W., Neklyudov, K., De Bortoli, V., Wang, H., Wu, D., Ferber, A., Ma, Y.-A., Gomes, C. P., and Zhang, C. Diffusion Models as Constrained Samplers for Optimization with Unknown Constraints, April 2024.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow Matching for Generative Modeling, February 2023.

Loeffler, H. H., He, J., Tibo, A., Janet, J. P., Voronov, A., Mervin, L. H., and Engkvist, O. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

Minkin, V. I., Osipov, O. A., and Zhdanov, Y. A. *Dipole Moments in Organic Chemistry*. Springer US, Boston, MA, 1970.

Neeser, R. M., Correia, B., and Schwaller, P. FSscore: A Personalized Machine Learning-Based Synthetic Feasibility Score. *Chemistry–Methods*, 4(11), 2024.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

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

Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, August 2014.

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

Schaufelberger, L., Blaskovits, J. T., Laplaza, R., Jorner, K., and Corminboeuf, C. Inverse design of singlet-fission materials with uncertainty-controlled genetic optimization. *Angewandte Chemie*, 137(3):e202415056, 2025.

Schneuing, A., Harris, C., Du, Y., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Blundell, T., Lio, P., Welling, M., Bronstein, M., and Correia, B. Structure-based Drug Design with Equivariant Diffusion Models, September 2024.

Song, J., Meng, C., and Ermon, S. Denoising Diffusion Implicit Models, October 2022.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021.

Stark, H., Jing, B., Wang, C., Corso, G., Berger, B., Barzilay, R., and Jaakkola, T. Dirichlet Flow Matching with Applications to DNA Sequence Design, May 2024.

Tang, W. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond, March 2024.

Treven, L., Hübotter, J., Sukhija, B., Dorfler, F., and Krause, A. Efficient exploration in continuous-time model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 36:42119–42147, 2023.

Tripp, A. and Hernández-Lobato, J. M. Genetic algorithms are strong baselines for molecule generation. 2023.

Uehara, M., Zhao, Y., Biancalani, T., and Levine, S. Understanding Reinforcement Learning-Based Fine-Tuning of Diffusion Models: A Tutorial and Review, July 2024a.

Uehara, M., Zhao, Y., Black, K., Hajiramezanali, E., Scalia, G., Diamant, N. L., Tseng, A. M., Biancalani, T., and Levine, S. Fine-Tuning of Continuous-Time Diffusion Models as Entropy-Regularized Control, February 2024b.

Uehara, M., Zhao, Y., Black, K., Hajiramezanali, E., Scalia, G., Diamant, N. L., Tseng, A. M., Levine, S., and Biancalani, T. Feedback Efficient Online Fine-Tuning of Diffusion Models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 48892–48918. PMLR, July 2024c.

Uehara, M., Zhao, Y., Wang, C., Li, X., Regev, A., Levine, S., and Biancalani, T. Inference-Time Alignment in Diffusion Models with Reward-Guided Generation: Tutorial and Review, January 2025.

Wang, H., Zariphopoulou, T., and Zhou, X. Y. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34, 2020.

Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A. M., and Gershoni-Poranne, R. Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882, 2023.

Wu, K. E., Yang, K. K., van den Berg, R., Alamdari, S., Zou, J. Y., Lu, A. X., and Amini, A. P. Protein structure generation via folding diffusion. *Nature Communications*, 15(1):1059, February 2024. ISSN 2041-1723.

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

A. Related Works

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., Uehara et al., 2024b; Tang, 2024; Uehara et al., 2024c; Domingo-Enrich et al., 2025). These methods have been successfully applied to real-world domains such as image generation (Domingo-Enrich et al., 2025), molecular design (Uehara et al., 2024c), or protein engineering (Uehara et al., 2024c). In this work, we propose a principled extension of such formulations to the case where known constraints are available (see Sec. 3) and one wishes to have constrained generation guarantees (see Sec. 5). In particular, Augmented Lagrangian Flows Fine-Tuning (see Alg. 1) renders possible to tackle this more complex, constrained setting, by leveraging as oracles the aforementioned control-based methods for unconstrained fine-tuning.

Molecular Design. Molecular design is the discovery of molecules with desired properties for a given problem in chemistry and biology. In earlier works, quantum chemical methods were used to identify reactivity descriptors and rationally optimize reactions (Brinck & Liljenberg, 2015). Genetic algorithms have shown strong performance on efficient exploration of chemical space (Tripp & Hernández-Lobato, 2023). They have been applied for molecular materials design (Schaufelberger et al., 2025), or optimizing log$P$ values with a constraint for synthetic accessibility (Jensen, 2019). With the advent of deep learning, RNN-based autoencoders that represent and explore the latent space were employed to optimize for drug-like structures (Gómez-Bombarelli et al., 2018; Blaschke et al., 2018). Reinforcement learning techniques have also been applied to SMILES strings (Olivecrona et al., 2017; Loeffler et al., 2024), while enforcing synthesizability constraints (Guo & Schwaller, 2025). Diffusion and flow models can be adapted to sample molecules with targeted properties, which has been used for a variety of single- and multi-objective optimization tasks (Weiss et al., 2023), for example, with online finetuning (Uehara et al., 2024c). Recent work also incorporated synthesizability constraints into generative modeling by producing synthetic pathways that ensure that candidates are synthetically tractable (Gao et al., 2024). Compared with previous work, Augmented Lagrangian Flows Fine-Tuning renders possible to trade-off reward maximization and domain-specific constraint fulfillment (e.g., non-toxicity, synthesizability) in a more controllable way supported by constrained generation guarantees.

Constrained Generative Modeling. Recent works increasingly focus on the challenge of embedding constraints into generative models. These methods seek to ensure that generated samples not only reflect the learned data distribution but also satisfy downstream constraints. Typically, constraints include linear (Graikos et al., 2025) and differentiable functions (Khalafi et al., 2024). Kong et al. (2024) extend this by proposing a method able to satisfy black-box constraints. The strategies for enforcing constraints range from dual optimization frameworks (Khalafi et al., 2024) to classifier-guided sampling (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) and reward-weighted denoising (Kong et al., 2024). We extend this line of work by tackling the constraint generation problem with an Augmented Lagrangian approach (in Alg. 1) that does not need any dual information or time-dependent classifier. Future works include studying the problem of constraint satisfaction at inference-time (Uehara et al., 2025; Graikos et al., 2025).

B. Adjoint Matching (Domingo-Enrich et al., 2025) implementation of FINETUNINGSOLVER

To ensure completeness, below we provide pseudocode for one concrete realization of a FINETUNINGSOLVER as in Eq. (7). We describe exactly the version employed in Sec. 6, which builds on the Adjoint Matching framework (Domingo-Enrich et al., 2025), 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 (Ho et al., 2020), effectively following the flow models notation introduced by Adjoint Matching (Domingo-Enrich et al., 2025, Sec. 5.2). The full procedure is in Algorithm 2.

Algorithm 2 FINETUNINGSOLVER (Adjoint Matching (Domingo-Enrich et al., 2025)) based implementation

input N : number of iterations, u^{pre} : pre-trained flow vector field, α regularization coefficient as in Eq. (4), ∇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 $\sigma(t)$ (Domingo-Enrich et al., 2025), e.g.,

$$\text{sample } \varepsilon_t \sim \mathcal{N}(0, I),\ X_0 \sim \mathcal{N}(0, I), \text{ then:} \tag{13}$$

$$X_{t+h} = X_t + h\left(2u_\theta^{\text{finetuned}}(X_t, t) - \frac{\bar{\alpha}_t}{\alpha_t} X_t\right) + \sqrt{h}\sigma(t)\varepsilon_t \tag{14}$$

Use objective function gradient:

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

For each trajectory, solve the lean adjoint ODE, see (Domingo-Enrich et al., 2025, Eq. 38-39), from $t = 1$ to 0:

$$\tilde{a}_{t-h} = \tilde{a}_t + h\tilde{a}_t^\top \nabla_{X_t}\left(2u^{\text{pre}}(X_t, t) - \frac{\bar{\alpha}_t}{\alpha_t} X_t\right) \tag{15}$$

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 (Domingo-Enrich et al., 2025, Eq. 37):

$$\mathcal{L}_\theta = \sum_{t \in \{0, h, \ldots, 1-h\}} \left\| \frac{2}{\sigma(t)}\left(u_\theta^{\text{finetuned}}(X_t, t) - u^{\text{pre}}(X_t, t)\right) + \sigma(t)\tilde{a}_t \right\|^2 \tag{16}$$

Compute the gradient $\nabla_\theta \mathcal{L}(\theta)$ and update θ.

end for

output: Fine-tuned flow vector field $u_\theta^{\text{finetuned}}$

For the molecular design task, we fine-tune FlowMol (Dunn & Koes, 2024). 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 (Dunn & Koes, 2024) for the sampling of categorical and initial values, we use CTMC sampling in all our experiments.

The reward and constraint, building the objective function, are implemented in `dxtb` (Friede et al., 2024). `dxtb` provides gradients with respect to the positions of each atom. Thus, the adjoint in Eq. (15) and loss in Eq. (16) are calculated for atomic positions.

For further implementation details, we refer to (Domingo-Enrich et al., 2025, Appendix G).

C. Parameter Augmented Lagrangian Flows Fine-Tuning for and Adjoint Matching

For the synthetic datasets present in Figure 2a and Figure 2e and the molecular design task, we used the following parameters:

Table 1. Hyperparameters for ALF^2 and Adjoint Matching

	Simple Gaussian	**Mixure of Gaussians**	**Molecular Design**
ALF^2			
Lagrangian Updates K	35	35	20
ρ_{init}	0.5	0.5	2
η	1.25	1.25	1.1
τ	0.99	0.99	0.999
Adjoint Matching			
$(1/\alpha)$	1e5	1e6	1e2
Number of Iterations N	300	300	10
Effective Batch Size	512	512	40
Loss Clipping	-	-	1e5
Clip Grad Norm	0.7	0.7	0.5
Learning Rate	5e-6	5e-6	1e-4
Integration Steps	40	40	50

For the synthetic datasets, an MLP with 3 hidden layers, each with 256 nodes, is trained on 20k samples (80/20 train validation split) for 500 epochs. For the molecular design task, we fine-tune FlowMol (Dunn & Koes, 2024), with CTMC sampling for the discrete features.

D. Proofs

Before we present a proof of the theorems in Section 5. We will transform the main problem in Eq. (4) 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

$$dX_t = b(X_t, t)dt + \sigma(t)dB_t,\ X_0 \sim p_0, \tag{17}$$

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 (Domingo-Enrich et al., 2025). Controlling this SDE can be done by adjusting the drift as follows (Domingo-Enrich et al., 2025):

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

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. (4) in terms of the controlled diffusion process $\mathbf{X}^u \sim p^u$:

$$\begin{aligned} \max_{u \in \mathcal{U}} \quad & \mathbb{E}_{\mathbf{X}^u \sim p^u}\left[r(X_1)\right] - \alpha D_{KL}(p_1^u(\cdot)||p_1^{\text{pre}}(\cdot)) \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{X}^v \sim p^v}\left[c(X_1)\right] \leq B \end{aligned} \tag{18}$$

Eq. (18) may seem the same as Eq. (4), but it is in terms of a diffusion process. This way we can calculate the KL efficiently, see (Eq. 18, Domingo-Enrich et al., 2025), 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 (Domingo-Enrich et al., 2025; Uehara et al., 2024b; Tang, 2024). This dependence on the initial value can be dropped when using a specific noise schedule (Domingo-Enrich et al., 2025). 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:

$$\begin{aligned} \max_{\theta \in \mathbb{R}^m} \quad & F(\theta) := F_r(\theta) - \alpha F_{KL}(\theta) \\ & \qquad = \mathbb{E}_{x \sim p_1^{u_\theta}}[r(x)] - \alpha \mathbb{E}\left[\int_0^1 \frac{1}{2}\|u_\theta(X_t, t)\|^2 dt\right] \\ \text{s.t.} \quad & G(\theta) := \mathbb{E}_{x \sim p_1^{u_\theta}}[c(x)] - B \leq 0 \end{aligned} \tag{19}$$

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 Algorithm 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" (Birgin & Martínez, 2014, Chapter 5).

The augmented Lagrangian objective in Eq. (5) becomes:

$$L_\rho(\theta, \lambda) = F(\theta) - \frac{\rho}{2}\left[\max\left(0, G(\theta) - \frac{\lambda}{\rho}\right)\right]^2 \tag{20}$$

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

$$L_{\rho_k}(\theta_k, \bar{\lambda}_k) \geq L_{\rho_k}(\theta, \bar{\lambda}_k) - \varepsilon_k \quad \forall\, \theta \in \mathbb{R}^m \tag{21}$$

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

This corresponds to Assumption 5.1 from (Birgin & Martínez, 2014). Assumption D.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 D.2 (Feasibility of Augmented Lagrangian Flows Fine-Tuning)**.** *Let $\{\theta_k\}$ be a sequence generated by Algorithm 1 under the solver Assumption D.1. Let $\bar{\theta}$ be a limit of the sequence $\{\theta_k\}$. Then, we have:*

$$\left\langle G(\bar{\theta}) \right\rangle_+ \leq \langle G(\theta) \rangle_+ \quad \forall \theta \in \mathbb{R}^m, \tag{22}$$

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 of Eq. (8) 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 $\left\{\bar{\lambda}_k\right\}$, and the fact that $\rho_k \to \infty$, there exists $c > 0$ and $k_0 \in \mathbb{N}$ such that for all $k \in K, k \geq k_0$:

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

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

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

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,

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

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

Theorem D.2 and its proof correspond to Birgin & Martínez (2014, Sec. 5.1). Theorem D.2 establishes that Algorithm 1, under the iterates given in Assumption D.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 Equation (4).

Theorem D.3 (Optimality of Augmented Lagrangian Flows Fine-Tuning)**.** *Let $\{\theta_k\} \subset \mathbb{R}^d$ be a sequence generated by Algorithm 1 under Assumption D.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 Equation (4).*

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

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

By assumption, the problem is feasible, thus, by Theorem D.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

$$F(\theta_k) - \frac{\rho_k}{2}\left[\left\langle G(\theta_k) - \frac{\bar{\lambda}_k}{\rho_k}\right\rangle_+^2\right] \geq F(\theta) - \frac{\rho_k}{2}\left[\left\langle G(\theta) - \frac{\bar{\lambda}_k}{\rho_k}\right\rangle_+^2\right] - \varepsilon_k \tag{23}$$

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

$$\left\langle G(\theta) - \frac{\bar{\lambda}_k}{\rho_k}\right\rangle_+^2 \leq \left(\frac{\bar{\lambda}_k}{\rho_k}\right)^2. \tag{24}$$

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 Equation (23) and Equation (24), we have

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

Taking limits for $k \in K$, and using that $\theta_k \to \bar{\theta}$, we have that $\lim_{k\in K}(\bar{\lambda}_k)^2/\rho_k = 0$ 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 D.1, Equation (23) holds for all $k \geq k_0$, and Equation (24) holds with $\rho = \rho_{k_0}$. Thus,

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

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

$$F(\bar{\theta}) - \frac{\rho_{k_0}}{2}\left[\left\langle G(\bar{\theta}) - \frac{\bar{\lambda}^*}{\rho_{k_0}}\right\rangle_+^2\right] \geq F(\theta) - \frac{(\bar{\lambda}^*)^2}{2\rho_{k_0}} - \varepsilon_k. \tag{25}$$

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

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

Therefore, by Equation (25),

$$F(\bar{\theta}) - \frac{\rho_{k_0}}{2}\left[\left\langle G(\bar{\theta}) - \frac{\bar{\lambda}^*}{\rho_{k_0}}\right\rangle_+^2\right] \geq F(\theta) - \frac{(\bar{\lambda}^*)^2}{2\rho_{k_0}}. \tag{26}$$

But, by Equation (8), $\lim_{k\to\infty}\min\{G(\theta_k), -\bar{\lambda}^*/\rho_{k_0}\} = 0$. Therefore, if $G(\bar{\theta}) < 0$, we necessarily have that $\bar{\lambda}^* = 0$. Therefore, Equation (26) 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 D.3: the first is that having access to such an solver is difficult and in practice rarely the case. Secondly, we refer the reader to Birgin & Martínez (2014, Sec. 5.2) for a discussion about the sets K and K_1, how they are connected to the convexity of F and G, and the corresponding theorem and proof.