CONSTRAINED FLOW MATCHING VIA LAGRANGIAN DUAL FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow matching is a powerful tool for generative modeling, but emerging applications in robotics, planning, and control require inference-time constraints on generated outputs. Such constraints are often complex and highly nonlinear. As a result, methods designed for linear constraints like image inpainting are rarely sufficient, and projection or optimization-based alternatives can be prohibitively expensive. In this paper, we introduce Lagrangian Dual Flows, a new family of constrained generation techniques based on Lagrangian dual dynamics. By flowing a dual co-state alongside generated samples, we can guarantee nonlinear constraint satisfaction without expensive optimization subproblems, pseudoinverses, or projection steps during the denoising process. The resulting constrained generation algorithms are simple, effective, and open new theoretical connections between flow matching and primal-dual methods in numerical optimization.

1 INTRODUCTION

Flow matching (Lipman et al., 2023) has emerged as an effective and widely-used approach for generative modeling, with applications in image and video generation (Lipman et al., 2023; Ho et al., 2022; Jin et al., 2025), robotics (Black et al., 2024; Zhang et al., 2025; Kurtz & Burdick, 2025), physics (Utkarsh et al., 2025; Baldan et al., 2026), and beyond. Flow matching learns a vector field that is integrated at inference time to transport samples from a reference distribution (e.g., standard Gaussian) to an approximation of the data distribution.

However, many generative modeling applications require the generated output to satisfy constraints. In robotics, for example, generated actions should avoid collisions while generated trajectories must satisfy dynamic feasibility constraints. In these emerging application areas, constraints are critical but (i) are only known at inference time and (ii) are often nonlinear. To meet these needs, we aim to take a pre-trained flow matching model and modify the generation procedure such that the generated samples satisfy constraints.

Existing methods for doing so either (i) require simple linear constraints (Song et al., 2023; Pokle et al., 2024), or (ii) solve a computationally expensive subproblem at each integration step in the denoising process (Li et al., 2025; Utkarsh et al., 2025). Linear constraints are a special case of linear inverse problems, a well-studied problem class that includes image inpainting and super-resolution, but these techniques are not directly applicable to general nonlinear constraints. These and related methods often rely on computing a pseudoinverse of the constraint, which can be difficult to compute or ill-defined. Methods that support nonlinear constraints require projection to the constraint manifold (Utkarsh et al., 2025) or the solution of optimization subproblems (Li et al., 2025; Dai et al., 2025) in each inference step, both of which can be computationally demanding.

Note that while linear methods are typically derived as approximate Bayesian posterior sampling, steering samples toward a conditional distribution given the constraints (Song et al., 2023), nonlinear constrained methods typically focus only on constraint satisfaction. We follow the latter group, and ask only that generated samples be *feasible* with respect to the constraints.

To this end, we present a family of efficient methods for constraint satisfaction in flow matching, which we call *Lagrangian Dual Flows* (LDF). Inspired by the classical view of constrained optimization as a continuous-time dynamical system (Arrow et al., 1958; Platt & Barr, 1987), we augment the state of the flow matching model with a dual variable that plays the role of a Lagrange multiplier.

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

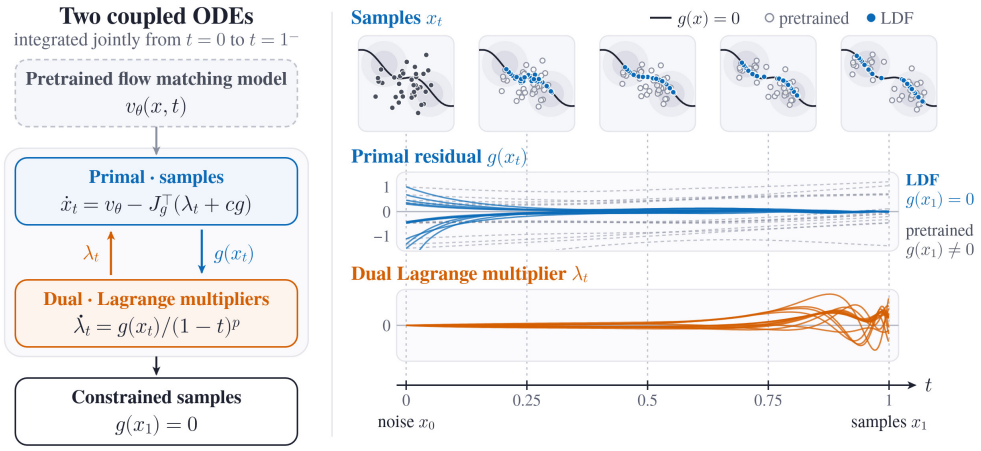

Figure 1: Visual summary of LDF, which integrates two coupled ODEs. Lagrange multipliers λ_t flow alongside the sample x_t, applying an adaptive correction that guarantees constraint satisfaction to tight numerical tolerances. LDF is simple, efficient, supports nonlinear equality and inequality constraints, and requires no specialized tooling beyond the off-the-shelf ODE solver already used for unconstrained generation.

The dual dynamics are a simple ordinary differential equation (ODE) defined by the constraint; the same off-the-shelf ODE integration tools typically used for flow matching can thus generate samples satisfying constraints. LDF handles both nonlinear equality and inequality constraints and does not require any pseudoinverse or computationally expensive optimization sub-problems.

Beyond these structural advantages, LDF comes with provable guarantees. We show that the constraint violation is driven to zero at an explicit rate $O\big((1-t)^\alpha\big)$ as $t \to 1^-$ for an explicit constant $\alpha > 0$. This result holds under standard regularity and constraint-qualification assumptions, together with a small-variation condition requiring that the Gram matrix associated with the constraint Jacobian stay sufficiently close to some fixed reference matrix along the trajectory, which holds automatically for affine constraints. We emphasize that this is a guarantee of *feasibility* of the generated sample, not of convergence to any particular conditional distribution.

In summary, our contributions are as follows:

- We introduce *Lagrangian Dual Flows*, a family of methods for enforcing inference-time constraints on a pre-trained flow matching model by augmenting the generative ODE with dual (and, for inequality constraints, slack) variables. Sampling requires only the same off-the-shelf ODE solvers used for unconstrained flow matching, enabling simple and computationally efficient implementations.
- Our proposed approach supports general nonlinear equality and inequality constraints known only at inference time. Inequalities are handled via a slack reformulation whose dynamics keep the slack nonnegative automatically (Lemma 2).
- We prove that the constraint violation converges to zero as $t \to 1^-$ with an explicit algebraic rate (Theorems 1 and 3).
- We demonstrate the effectiveness of our approach on a variety of challenging constrained generation problems, including nonlinear equality constraints, image inpainting, collision-free path planning, and legged robot locomotion.

The remainder of this paper is organized as follows: Section 2 reviews existing methods for constrained flow matching, while Section 3 presents background and a formal problem setting. Section 4 presents our main results, including algorithms for equality and inequality-constrained generation and theoretical convergence analysis. We illustrate the effectiveness of LDF with several examples in Section 5, and conclude with Section 6.

2 Related Work

Inference-time constrained generation with flow matching is an active area of research. We summarize the main families of approaches here.

Penalty and guidance with quadratic costs. Some training-free methods add the gradient of a quadratic constraint-violation penalty to the generative dynamics (Ho et al., 2022; Chung et al., 2022; Feng et al., 2025). This guidance is inherently soft. Tight constraint satisfaction requires a larger penalty which, in turn, stiffens the ODE. Inspired by primal-dual methods in numerical optimization, LDF retains a penalty term but augments it with a dual variable that drives the residual to zero exactly while limiting numerical ill-conditioning.

Pseudoinverse and posterior sampling guidance. A second family approximates the conditional score with a Gaussian posterior, resulting in corrections built from the pseudoinverse of the constraint Jacobian (Song et al., 2023; Chung et al., 2023; Peng et al., 2024; Pokle et al., 2024; Pourya et al., 2025; Black et al., 2026). These methods are designed for posterior sampling under (approximately) linear-Gaussian measurement models, and the per-step cost scales with the number of constraints. LDF includes a dual correction term that recovers a similar structure when specialized to linear constraints, but replaces the pseudoinverse with a dual variable governed by a simple ODE.

Projection and reflection-based methods. A third family enforces constraints geometrically by projecting iterates onto the feasible set or by reflecting the dynamics at the boundary of this set (Cheng et al., 2025; Deng et al., 2024; Xie et al., 2024; Utkarsh et al., 2025; Ma et al., 2026). These methods are effective for simple constraints when projection is cheap, but exact projection onto a general nonlinear constraint set is itself an optimization problem, and hard projection inside an ODE yields discontinuous dynamics (Nagurney & Zhang, 2012). Our soft-projected slack dynamics (Section 4.3) keep the dynamics Lipschitz while retaining convergence guarantees.

Control-theoretic methods. A fourth family borrows machinery from trajectory optimization, model predictive control, and control barrier functions (CBFs), enforcing constraints through an optimization subproblem or safety filter at each integration step (Li et al., 2025; Xiao et al., 2025; Dai et al., 2025; Yang et al., 2025). These methods ensure tight constraint satisfaction, but the per-step optimization subproblem can be computationally expensive. Additionally, CBF-based methods require finding a barrier function compatible with the dynamics (Ames et al., 2019). This can pose a considerable challenge when the dynamics are defined by a large pre-trained flow matching model.

ODE-based optimization. Finally, our approach draws on the classical view of optimization as a dynamical system, beginning with the basic differential multiplier method of Platt and Barr (Platt & Barr, 1987) and the broader literature on continuous-time primal-dual and saddle-point flows (Arrow et al., 1958; Feijer & Paganini, 2010; Cherukuri et al., 2017; Qu & Li, 2019). LDF couples the dual dynamics to the generative ODE and integrates both with the same off-the-shelf solver such that feasibility is achieved at the end of the denoising process.

3 Background

Generative modeling and flow matching. The goal of generative modeling is to draw new samples from an unknown data distribution $\mathcal{D}$ on $\mathbb{R}^n$ given access only to a finite collection of samples $x^{(i)} \sim \mathcal{D}$, $i \in \{1, \dots, N\}$. Flow matching (Lipman et al., 2023) approaches this challenge by learning a time-varying vector field $v_\theta : \mathbb{R}^n \times [0,1] \to \mathbb{R}^n$ that transports a simple reference distribution into $\mathcal{D}$, where θ denotes trainable parameters. Typically, a sample is generated by drawing an initial state $x_0 \sim \mathcal{N}(0, I)$ from a standard normal distribution and integrating the ODE

$$\dot{x}_t = v_\theta(x_t, t) \tag{1}$$

from $t = 0$ to $t = 1$. The endpoint x_1 is then a sample approximately distributed according to $\mathcal{D}$. The vector field v_θ is trained so that its flow matches a prescribed probability path connecting the Gaussian reference to the data distribution. We treat v_θ as frozen and refer the reader to (Lipman et al., 2023) for details on how to train such a model. A key feature of flow matching is that sampling requires integrating (1), typically with off-the-shelf ODE solvers.

Constraints. In many applications, the generated sample x_1 is required to satisfy constraints. We consider equality and inequality constraints,

$$g(x_1) = 0, \qquad h(x_1) \leq 0, \tag{2}$$

where $g : \mathbb{R}^n \to \mathbb{R}^m$ and $h : \mathbb{R}^n \to \mathbb{R}^k$. We understand inequality constraints componentwise, and the constraints are permitted to be nonlinear.

Crucially, we assume that g and h are only known at inference time. A robot may encounter a new obstacle, an inpainting mask may be specified by a user, or a physical conservation law may be imposed only at deployment. The vector field v_θ is thus trained without knowledge of these constraints and $\mathcal{D}$ may be supported outside of the feasible set. As a result, an unconstrained sample x_1 may violate the constraints (2).

Objective. Our goal is to take a pre-trained v_θ and, at inference time, modify the sampling process such that the generated sample satisfies the constraints. We seek a modification that satisfies three properties. First, it should certify constraint satisfaction: the generated x_1 should provably satisfy (2) at least up to small numerical tolerances. Second, it should accommodate nonlinear g and h. Third, it should remain computationally lightweight, ideally requiring no more than the integration of an augmented ODE with the same off-the-shelf solvers used for unconstrained sampling. In particular, we would like to avoid solving optimization subproblems (e.g., projection onto the constraint set) or expensive nonlinear pseudoinverse computations at each integration step.

4 Lagrangian Dual Flows

In this section we introduce LDF, our family of methods for enforcing constraints on a pre-trained flow matching model at inference time. Before presenting our approach, we motivate its form by examining two classical approaches to enforcing an equality constraint $g(x) = 0$ on the output of a flow (1). Both approaches admit natural formulations within the flow matching framework, and each has a shortcoming that LDF addresses.

Penalty method. A simple way to drive the sample toward the constraint set is to add the negative gradient of a quadratic penalty $\frac{c}{2}\|g(x)\|^2$ to the flow:

$$\dot{x}_t = v_\theta(x_t, t) - c\, J_g(x_t)^\top\, g(x_t), \tag{3}$$

where $c > 0$ is a constant and $J_g(x) = \nabla_x g(x)$ is the Jacobian of g. This correction pulls trajectories toward the constraint manifold $\{x \in \mathbb{R}^n \mid g(x) = 0\}$ but exact constraint satisfaction requires $c \to \infty$. For any finite $c > 0$, the penalty must balance the pre-trained v_θ and a residual persists at $t = 1$. At the same time, large c makes the ODE stiff and computationally expensive to integrate.

Tangential projection. A more principled approach is to require that the corrected flow remain tangent to the constraint manifold, so that trajectories never move away from $g(x) = 0$ after reaching it. Augmenting (1) with a correction of the form $-J_g(x)^\top\, \mu$ and imposing $\frac{d}{dt} g(x_t) = 0$ determines $\mu(x, t) = \left(J_g J_g^\top\right)^{-1} J_g\, v_\theta(x, t)$. Substituting back and retaining a penalty term yields

$$\dot{x}_t = \left(I - J_g(x_t)^\dagger\, J_g(x_t)\right) v_\theta(x_t, t) - c J_g(x_t)^\top\, g(x_t), \tag{4}$$

where $J_g(x)^\dagger = J_g(x)^\top\, (J_g(x)\, J_g(x)^\top)^{-1}$ is the Moore-Penrose pseudoinverse of $J_g(x)$ and the penalty term ensures that g converges to zero. Note that we can choose a smaller $c > 0$ in this case, as $\mu(x, t)$ eliminates movement away from the constraint manifold. The cost is that every evaluation of the right-hand side requires solving an $m \times m$ system with $J_g J_g^\top$, which becomes expensive for complex and high-dimensional constraints, and unstable when that matrix is ill-conditioned.

4.1 The Proposed Method

Let $g : \mathbb{R}^n \to \mathbb{R}^m$ be the equality constraint in (2) and let $J_g(x) \in \mathbb{R}^{m \times n}$ denote its Jacobian matrix evaluated at $x \in \mathbb{R}^n$. Inspired by constrained differential optimization (Platt & Barr, 1987), we augment the flow matching state x with a dual variable $\lambda \in \mathbb{R}^m$, and evolve the pair (x, λ) jointly.

For $t \in [0, 1)$ and fixed constants $c > 0, p \in [1, \infty)$, the *Lagrangian dual flow* is the system

$$\dot{x}_t = v_\theta(x_t, t) - J_g(x_t)^\top \lambda_t - c\, J_g(x_t)^\top g(x_t), \tag{5a}$$

$$\dot{\lambda}_t = \frac{g(x_t)}{(1-t)^p}, \tag{5b}$$

with initial conditions $x_0 \sim \mathcal{N}(0, I_n)$, $\lambda_0 = 0$. A constrained sample is generated by integrating (5) from $t = 0$ to $t = 1$ and returning x_1, as outlined in Algorithm 1.

Algorithm 1 Equality-constrained Lagrangian Dual Flows

Require: Trained flow model $v_\theta(x, t)$, constraint $g(x)$, penalty $c > 0$, constant $p \geq 1$.
Ensure: Final state x_1 such that $g(x_1) = 0$.

1: Initialize the state $x_0 \sim \mathcal{N}(0, I_n)$.
2: Initialize the dual $\lambda_0 = 0$.
3: Integrate from $t = 0$ to $t = 1$:

$$\dot{x}_t = v_\theta(x_t, t) - J_g(x_t)^\top \lambda_t - c\, J_g(x_t)^\top g(x_t),$$
$$\dot{\lambda}_t = g(x_t)/(1-t)^p.$$

4: (Optional) perform N_{proj} Gauss-Newton iterations to project x_1 to the constraint manifold.
return x_1

The dynamics (5) modify the unconstrained flow (1) through two correction terms in (5a) and the introduction of an auxiliary equation (5b). First, the term $-cJ_g(x)^\top g(x)$ in (5a) pulls the sample toward the constraint set with the strength of the pull controlled by the weight $c > 0$, as in a quadratic penalty method. Second, the term $-J_g(x)^\top \lambda$ applies a correction in directions normal to the constraint set with magnitude set by the dual variable λ. Rather than a fixed penalty or a quantity computed from x, λ is a system state and is integrated over time according to the dynamics (5b). The time-rescaling factor $(1-t)^{-p}$ makes this accumulation increasingly aggressive as $t \to 1^-$, concentrating the correction near the end of the integration interval. For numerical stability, our implementation uses $(1 - t + \epsilon)^{-p}$, with $\epsilon = 1 \times 10^{-6}$. This rescaling is necessary because, unlike in (Platt & Barr, 1987) where integration is from $t = 0$ to $t = \infty$, flow models integrate from $t = 0$ to $t = 1$. The exponent $p > 0$ governs the rate of this concentration. We take $p = 2$ throughout and justify this choice in Section 4.2. Intuitively, growth in $(1-t)^{-p}$ is balanced by decay in $g(x_t)$.

At the end of Algorithm 1, optional Gauss-Newton iterations project the generated sample onto the set $\{x \mid g(x) = 0\}$. We adapt this step from (Utkarsh et al., 2025). It is computationally cheap, since $v_\theta(x, t)$ is not used, and allows us to achieve constraint satisfaction to tight numerical tolerances. In practice, we find that this projection step does not change the quantitative nature of generated samples, and we limit ourselves to $N_{\text{proj}} = 2$ projection iterations.

Computational cost. Sampling from (5) only requires the integration of an augmented ODE, carried out with the same off-the-shelf integrators used for unconstrained flow matching. Compared to unconstrained flow matching, LDF adds m scalar ODE channels for the dual variables and, per evaluation of the right-hand-side, a single vector-Jacobian product (VJP) $[\lambda_t + cg(x_t)]^\top J_g(x_t)$. This VJP is cheaply available via automatic differentiation.

4.2 Theoretical Analysis

We now establish that LDF drives the equality constraint violation to zero as $t \to 1^-$. We discuss inequality constraints in Section 4.3. Our analysis rests on the following standing assumptions, which we take to hold along the trajectory of the dynamics (5).

- (A1) *Smoothness:* $g \in C^2(\mathbb{R}^n; \mathbb{R}^m)$ and $v_\theta \in C^1(\mathbb{R}^n \times [0, 1); \mathbb{R}^n)$, and there exist constants $L, B \geq 0$ such that $\|J_g(x)\|_2 \leq L$ and $\|v_\theta(x, t)\| \leq B$ along the trajectory.
- (A2) *Constraint qualification:* There exist constants $0 < a_{\min} \leq a_{\max} < \infty$ such that $a_{\min} I_m \preceq J_g(x) J_g(x)^\top \preceq a_{\max} I_m$ along the trajectory.
- (A3) *Bounded trajectory:* The trajectory (x_t, λ_t) exists on $[0, 1)$ with x_t lying in a compact set.

Assumption (A1) is a standard regularity condition. Assumption (A2) is a uniform full-rank constraint qualification condition on the constraint Jacobian, ensuring that the constraint directions remain well-conditioned along the trajectory. Assumption (A3) rules out finite-time blowup.

Main result ($p = 2$). Before stating the theorem, we describe the change of variables underlying the analysis. The singularity of (5b) at $t = 1$ is naturally handled by passing to the time variable $\tau := -\log(1 - t)$, which maps $[0, 1)$ onto $[0, \infty)$, together with the rescaled dual variable $\eta := (1 - t)\lambda$. In these variables convergence as $t \to 1^-$ corresponds to convergence as $\tau \to \infty$. The analysis then proceeds by constructing a Lyapunov function for the pair (g, η). To ensure that the Lyapunov function decreases along solution trajectories, the construction additionally requires a condition controlling the variation of the constraint Jacobian:

(A4) *Small variation:* There exists a reference matrix $A_* \succ 0$ such that, along the trajectory, $\rho := K_{A_*} \sup_t \left\| J_g(x_t) J_g(x_t)^\top - A_* \right\|_2 < 1$, where $K_{A_*} = \left\| A_*^{-1} + \frac{1}{2} I \right\|_2 + 1$.

We note that Assumption (A4) holds automatically with $\rho = 0$ for affine constraints, where $J_g(x)J_g(x)^\top$ is constant. Let A_* and $\rho < 1$ be the reference matrix and small-variation constant from (A4), let $P := \begin{pmatrix} 2A_*^{-1} + I_m & -I_m \\ -I_m & I_m + A_* \end{pmatrix} \succ 0$, and set $\alpha := (1 - \rho)/\lambda_{\max}(P) > 0$, where $\lambda_{\max}(P)$ is the maximum eigenvalue of P. We then obtain the following result:

Theorem 1 (Convergence at $p = 2$)**.** *Under Assumptions* (A1)–(A4) *with $p = 2$, the trajectory of* (5a)–(5b) *satisfies, as $t \to 1^-$,*

$$\|g(x_t)\| = O\big((1 - t)^\alpha\big) \quad \textit{and} \quad (1 - t)\,\|\lambda_t\| = O\big((1 - t)^\alpha\big).$$

In particular, since $\alpha > 0$, both quantities vanish as $t \to 1^-$ and the generated sample is feasible.

In particular, for affine constraints the conclusions of Theorem 1 hold without Assumption (A4).

The role of p. The exponent p in (5b) controls how aggressively λ accumulates as $t \to 1^-$. For $p \in [1, 2)$, we can prove (see the Appendix, Proposition 5) that $\|g(x_t)\|$ converges as $t \to 1^-$ but that its limiting value is generally nonzero. For $p > 2$ we have no general proof of convergence, although a scalar example converges analytically and numerical experiments suggest convergence more broadly. On the other hand, the dynamics become increasingly challenging to integrate as p increases. We therefore suggest $p = 2$ as a reasonable tradeoff between constraint satisfaction and ease of integration. We include more thorough theoretical analysis of different p in Appendix D.

4.3 Inequality Constraints

We now extend the method to inequality constraints $h(x) \leq 0$. We convert the inequality $h(x) \leq 0$ into an equality by introducing a slack variable $s \in \mathbb{R}^k$ and writing

$$h(x) + s = 0, \qquad s \geq 0,$$

so that $h(x) + s = 0$ together with $s \geq 0$ recovers $h(x) \leq 0$. Treating $\tilde{g}(x, s) := h(x) + s$ as an equality constraint on the augmented state (x, s) and applying the flow of Section 4.1 would yield dynamics for (x, s, λ) analogous to (5). The difficulty is the nonnegativity requirement $s \geq 0$: the slack dynamics produced this way do not keep s nonnegative on their own, and enforcing $s \geq 0$ by hard projection in the ODE produces a projected dynamical system (Nagurney & Zhang, 2012), which is discontinuous.

Soft projection. We can instead enforce $s \geq 0$ through a Lipschitz relaxation. The resulting Lagrangian dual flow for inequality constraints is

$$\dot{x}_t = v_\theta(x_t, t) - J_h(x_t)^\top \lambda_t - c\, J_h(x_t)^\top (h(x_t) + s_t), \tag{6a}$$

$$\dot{s}_t = c \left[-s_t + \mathrm{ReLU}\left(-h(x_t) - \frac{1}{c}\lambda_t \right) \right], \tag{6b}$$

$$\dot{\lambda}_t = \frac{h(x_t) + s_t}{(1 - t)^p}, \tag{6c}$$

with $x_0 \sim \mathcal{N}(0, I_n)$, $\lambda_0 = 0$, $s_0 = \mathrm{ReLU}(-h(x_0))$, and $\mathrm{ReLU}(y) = \max(y, 0)$. The slack equation (6b) is a globally Lipschitz relaxation of the projected dynamics. This approach is inspired by work on continuous-time projected and proximal gradient dynamics (Gao, 2003; Hassan-Moghaddam & Jovanović, 2021; Davydov et al., 2025). Moreover, these dynamics ensure nonnegativity of the slack variable.

Lemma 2 (Positive invariance of the slack variable)**.** *Along the slack dynamics* (6b) *with* $s_0 \geq 0$*, the slack satisfies* $s_t \geq 0$ *for all* $t \in [0, 1)$.

Convergence. Under similar assumptions as in the equality-constrained case, but now imposed on h in place of g (see Assumptions (B1)-(B4) in Appendix C for specifics), we establish the soft-projected dynamics drive the generated sample to satisfy the inequality constraints as $t \to 1^-$.

Let A_*, κ_0, and $\rho < \kappa_0$ be the reference matrix, dissipation margin, and small-variation constant from (B4), let $P \succ 0$ be the matrix defining the Lyapunov function in Lemma 4, and set $\alpha := (\kappa_0 - \rho)/\lambda_{\max}(P) > 0$.

Theorem 3 (Convergence for inequality constraints at $p = 2$)**.** *Suppose Assumptions* (B1)–(B4) *in Appendix C hold. Then with* $p = 2$ *the soft-projected dynamics* (6a)–(6c) *satisfy, as* $t \to 1^-$*,*

$$\left\|\mathrm{ReLU}\big(h(x_t)\big)\right\| = O\big((1-t)^\alpha\big) \quad \textit{and} \quad (1-t)\,\|\lambda_t\| = O\big((1-t)^\alpha\big).$$

In particular, since $\alpha > 0$*, both quantities vanish as* $t \to 1^-$ *and the generated sample is feasible.*

For affine constraints, (B4) holds automatically and the conclusions follow from (B1)–(B3) alone.

Reduced dynamics. Equation (6b) has s tend toward $s^\star(x, \lambda) = \mathrm{ReLU}(-h(x) - \lambda/c)$. For large $c > 0$, this slack variable converges quickly and we may replace s_t by $s^\star(x_t, \lambda_t)$. This substitution gives the reduced dynamics

$$\dot{x}_t = v_\theta(x_t, t) - J_h(x_t)^\top \mathrm{ReLU}(\lambda_t + c\,h(x_t)) \tag{7a}$$

$$\dot{\lambda}_t = \frac{\mathrm{ReLU}(\lambda_t + c\,h(x_t)) - \lambda_t}{c\,(1-t)^p} \tag{7b}$$

These dynamics are compatible with Algorithm 1, and are the variant we use in our experiments. It removes the slack channels and we find it integrates at least as accurately as the full system (6a)–(6c) at a lower cost. We leave a comprehensive convergence analysis of these dynamics to future work.

5 Numerical Results

We implemented LDF in JAX, using FLAX/NNX for neural networks and Diffrax (Kidger, 2021) for ODE integration. Experiments were run on a small laptop GPU (NVIDIA RTX Pro 2000, 8GB). Code at `https://anonymous.4open.science/r/constrained_flow_matching`.

We compare against several constrained generation algorithms. The **penalty** baseline adds a simple quadratic penalty on constraint violation, integrating (3) during the denoising process. The **PCFM** (physics-constrained flow matching) baseline (Utkarsh et al., 2025) solves an optimization subproblem at each denoising step to project samples to the constraint manifold. **ΠGDM** extends the approach of (Song et al., 2023; Pokle et al., 2024) to handle nonlinear constraints via iterative linearization. The **CBF** baseline enforces constraints by solving a (relaxed) CBF quadratic program at each denoising step, as described in (Dai et al., 2025). We test against two versions of our proposed approach: **LDF** without optional projection steps, and **LDF + projection** with $N_{\mathrm{proj}} = 2$ Gauss-Newton projection iterations following the denoising process.

Note that PCFM and ΠGDM support equality constraints only, while CBF supports inequality constraints only. Our proposed LDF methods support both types of constraints.

We test these algorithms on six constrained generation problems of increasing complexity.

In the **Star (eq.)** example, a simple flow model generates samples from a star-shaped submanifold of $\mathbb{R}^2$, as shown in Fig. 2. Samples are constrained to lie on the unit circle. We also consider an inequality constrained version of the same problem, **Star (ineq.)**, where samples are constrained to the right half of the plane (Fig. 4a). The star examples provide a simple, low-dimensional illustration

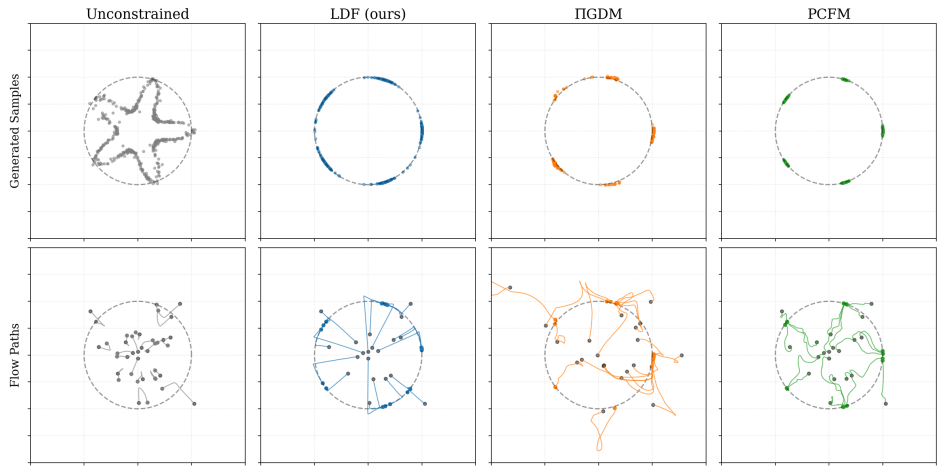

Figure 2: Generated star examples subject to a unit circle constraint. Our LDF method produces clean paths that drive samples directly toward the constraint manifold, unlike the PCFM (Utkarsh et al., 2025) and ΠGDM (Song et al., 2023; Pokle et al., 2024) baselines. LDF and PCFM both achieve tight constraint satisfaction. ΠGDM, as a method designed for linear constraints, produces samples slightly off the nonlinear constraint manifold.

of the method's ability to handle different constraint types, as well as an opportunity to visualize the generation process. Fig. 2 shows that LDF produces clean generation paths that drive directly toward the constraint manifold, while ΠGDM and PCFM take more circuitous generation paths.

Increasing the problem dimension, we come to **MNIST** image inpainting. Here an unconditional flow model is trained on MNIST (LeCun, 1998) to generate arbitrary digits. The top half of the image is fixed to match a reference from the MNIST test set, as shown in Fig. 3. This simple linear constraint is a special case of the nonlinear equality constraints that LDF can handle. Nonetheless, we find that LDF provides performance competitive with inpainting-specific algorithms like ΠGDM.

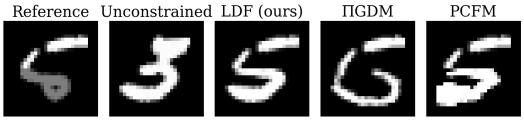

Figure 3: A pre-trained model generates MNIST handwritten digits. We then constrain the top half of the image to match the reference (left). LDF produces clean digits consistent with the inpainting mask. PCFM (Utkarsh et al., 2025) achieves tight constraint satisfaction, but at the cost of visible artifacts in the generated image. Additional samples are shown in Appendix A.2.3.

The **Obstacle** avoidance examples generate control points of a Bezier spline representing a robot's path through a field of circular obstacles, as shown in Fig. 4b. The constraint is that the path must avoid collisions with obstacles, as evaluated along 90 evenly-spaced points along the path. This is an example of a highly nonconvex inequality constraint. The CBF baseline can also produce collision-free paths, but is much slower than LDF. Additionally, the CBF-QP is by default infeasible and must be relaxed with an elastic formulation as suggested in (Dai et al., 2025). This relaxation eliminates any constraint satisfaction guarantees and some samples as a result do collide with the obstacles. We also find that the CBF method occasionally diverges, resulting in NaN errors.

Finally, we consider the **Hopper** and **Walker** examples described in (Xiao et al., 2025; Yang et al., 2026). In these examples, a flow matching network is trained on expert data to generate trajectories (state-action pairs over time) for planar robots. The constraint is a velocity-dependent ceiling on the robot's height, as illustrated in Fig. 4c. LDF-generated trajectories preserve the general shape of the phase portrait (torso height vs vertical velocity) while also enforcing the constraint.

Table 1 shows average generation time and constraint violation across 20 samples from each scenario. The two LDF variants consistently produce tight constraint satisfaction and fast generation times. This is thanks in large part to the simplicity of the method: an LDF implementation adds only a few lines of code and a single vector-Jacobian product; no expensive pseudoinverses or optimization subproblems are required during the denoising process.

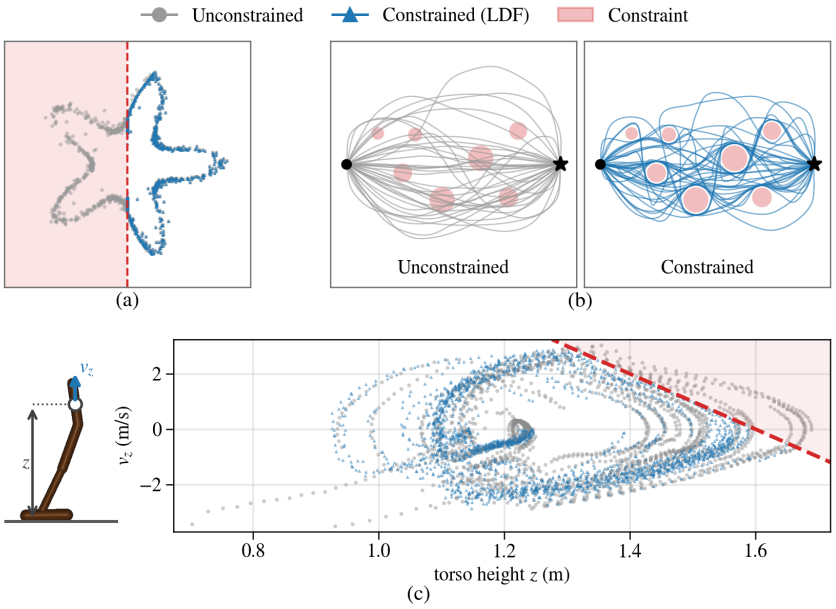

Figure 4: Examples of inequality-constrained test problems. (a) Inequality-constrained star; LDF preserves the shape of the data manifold while producing only samples that fall on the right half of the plane. (b) Obstacle avoidance; LDF avoids obstacles while providing smooth paths from a start to a goal. (c) Hopper; LDF generates trajectories for a hopping robot (left) that enforce a velocity dependent ceiling, as shown in the phase portrait on the right.

	Penalty only		CBF		PCFM		IIGDM		LDF (no proj.)		LDF + proj.	
Problem	Time	Viol.	Time	Viol.	Time	Viol.	Time	Viol.	Time	Viol.	Time	Viol.
Star (eq.)	**7.9**	3.6×10^{-2}	—	—	42.9	$\mathbf{2.4\times10^{-8}}$	15.3	3.1×10^{-2}	**9.2**	2.6×10^{-3}	**8.4**	$\mathbf{3.3\times10^{-8}}$
MNIST	**182.0**	3.3×10^{-1}	—	—	224.6	$\mathbf{1.0\times10^{-7}}$	543.4	1.1×10^{-1}	**182.3**	5.0×10^{-2}	**185.9**	$\mathbf{6.9\times10^{-8}}$
Star (ineq.)	**7.6**	2.2×10^{-2}	331.5	**0**	—	—	—	—	**8.7**	9.3×10^{-3}	**8.0**	**0**
Obstacles (2)	**48.4**	1.0×10^{-3}	2365.1	**0** †	—	—	—	—	**45.2**	7.3×10^{-5}	**46.8**	$\mathbf{7.5\times10^{-10}}$
Obstacles (6)	**46.6**	5.9×10^{-3}	3909.3	**0** †	—	—	—	—	**46.2**	2.0×10^{-4}	**48.6**	1.6×10^{-5}
Walker2D	**134.2**	8.1×10^{-3}	8336.6	**0**	—	—	—	—	**138.3**	9.8×10^{-4}	**137.0**	$\mathbf{1.2\times10^{-8}}$
Hopper	**134.0**	6.2×10^{-3}	4494.4	**0**	—	—	—	—	**135.3**	6.7×10^{-4}	**135.6**	$\mathbf{6.0\times10^{-9}}$

Table 1: Mean per-sample generation time (ms) and constraint violation across 20 samples. Violation is $\|g(x_1)\|$ for equality-constrained problems (top) and $\|\mathrm{ReLU}(h(x_1))\|$ for inequality-constrained problems (bottom). Bold times are within 20% of the lowest time in that row; bold violations are below 10^{-6} (near machine precision for single-precision float). †At least one NaN error produced during generation; average shown is across non-NaN samples.

6 Conclusion

Lagrangian dual flows (LDF) provide a simple, computationally efficient means to apply inference-time constraints to pre-trained flow matching policies. By flowing a dual state along with the samples, we can guarantee constraint satisfaction under standard constraint-qualification assumptions and a small-variation assumption without resorting to projections, pseudoinverses, or optimization subproblems. In fact, with careful implementation, a single vector-Jacobian product is all that is needed to implement LDF. Constraints need not be linear or even invertible. This makes LDF a promising candidate for emerging flow matching applications like robotics, planning, and control where constraints are critical but complex, and inference-time compute is limited.

Future work will also focus on improving efficiency, reducing ODE stiffness to enable few-step generation, and characterizing the conditional distribution of constrained samples. More broadly, LDF draws a direct connection between flow matching and the classical theory of primal-dual and saddle point dynamics. We believe that the literature on continuous-time optimization offers tools that may translate into faster and tighter constrained generation. Pursuing this connection and moving from feasibility to constraint-conditioned generation remains an exciting avenue for future work.

AI Disclosure Statement

In this work, we used generative AI tools for assistance in formulating and proving mathematical claims (specifically Theorems 1, 3, and Lemma 4), and for coding assistance in implementing methods and baselines. Additionally, we used generative AI tools for (i) drafting and formatting figures, (ii) identifying relevant literature, and (iii) polishing writing. We have verified all mathematical proofs and derivations, have tested all code, and have confirmed all suggested references.

We have not used generative AI tools for (i) generating synthetic data sets, (ii) developing theoretical models or conceptual frameworks, (iii) proposing or refining hypotheses, (iv) designing or providing feedback on research methodology or experiments, (v) assisting with translation, (vi) cleaning and reformatting datasets, (vii) supporting qualitative and thematic data analyses, or (viii) interpreting results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics Statement

This work contributes to the fundamental challenge of enforcing constraints on pre-trained flow matching models at inference time. Generative AI models (such as flow matching) raise many ethical questions concerning misuse and bias in generated content. This work does not claim to address any of these challenges, though it may serve as a first step toward safer content generation. Our experiments use publicly available datasets and simulation environments. Moreover, we do not involve any human subjects or private data.

Reproducibility Statement

To help in reproducing the results of this paper, we have made our code available at `https://anonymous.4open.science/r/constrained_flow_matching`. Implementation details for each of the experiments described in Section 5 are presented in Appendix A. Additional ablations can be found in Appendix B, and proofs of all mathematical claims in Appendix C.

References

Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *European Control Conference (ECC)*, pp. 3420–3431, 2019.

Jon Arrizabalaga, Kevin Tracy, and Zachary Manchester. A differentiable interior-point method in single precision. *arXiv preprint arXiv:2605.17913*, 2026.

Kenneth J. Arrow, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in Linear and Non-Linear Programming*. Stanford University Press, 1958.

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Physics vs distributions: Pareto optimal flow matching with physics constraints. In *International Conference on Learning Representations (ICLR)*, 2026.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:33383–33407, 2026.

Chaoran Cheng, Boran Han, Danielle C. Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W. Mahoney, and Bernie Wang. Gradient-free generation for hard-constrained systems. In *International Conference on Learning Representations (ICLR)*, 2025.

Ashish Cherukuri, Bahman Gharesifard, and Jorge Cortés. Saddle-point dynamics: Conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511, 2017.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

Xiaobing Dai, Zewen Yang, Dian Yu, Fangzhou Liu, Hamid Sadeghian, Sami Haddadin, and Sandra Hirche. SafeFlow: Safe robot motion planning with flow matching via control barrier functions. *arXiv preprint arXiv:2504.08661*, 2025.

Alexander Davydov, Veronica Centorrino, Anand Gokhale, Giovanni Russo, and Francesco Bullo. Time-varying convex optimization: A contraction and equilibrium tracking approach. *IEEE Transactions on Automatic Control*, 70(11):7446–7460, 2025.

Wei Deng, Yu Chen, Nicole Tianjiao Yang, Hengrong Du, Qi Feng, and Ricky T. Q. Chen. Reflected Schrödinger bridge for constrained generative modeling. In *Uncertainty in Artificial Intelligence (UAI)*, 2024. URL `https://proceedings.mlr.press/v244/deng24b.html`.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 8780–8794, 2021.

Diego Feijer and Fernando Paganini. Stability of primal–dual gradient dynamics and applications to network optimization. *Automatica*, 46(12):1974–1981, 2010.

Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. In *International Conference on Machine Learning (ICML)*, 2025.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Xing-Bao Gao. Exponential stability of globally projected dynamic systems. *IEEE Transactions on Neural Networks*, 14(2):426–431, 2003.

Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 1993.

Sepideh Hassan-Moghaddam and Mihailo R. Jovanović. Proximal gradient flow and Douglas–Rachford splitting dynamics: Global exponential stability via integral quadratic constraints. *Automatica*, 123:109311, 2021. doi: https://doi.org/10.1016/j.automatica.2020.109311.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations (ICLR)*, pp. 23378–23402, 2025.

Hassan K. Khalil. *Nonlinear Systems*, volume 3. Prentice Hall, 2002.

Patrick Kidger. *On Neural Differential Equations*. PhD thesis, University of Oxford, 2021.

Vince Kurtz and Joel W. Burdick. Generative predictive control: Flow matching policies for dynamic and difficult-to-demonstrate tasks. *arXiv preprint arXiv:2502.13406*, 2025.

Yann LeCun. The MNIST database of handwritten digits, 1998. http://yann.lecun.com/exdb/mnist/.

Zeyang Li, Kaveh Alim, and Navid Azizan. HardFlow: Hard-constrained sampling for flow-matching models via trajectory optimization. *arXiv preprint arXiv:2511.08425*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Jianming Ma, Qiyue Yang, Yang Zhang, Liyun Yan, Zhanxiang Cao, Yazhou Zhang, and Yue Gao. PolyFlow: Safe and efficient polytope-constrained flow matching with constraint embedding and projection-free update. In *International Conference on Machine Learning (ICML)*, 2026.

Anna Nagurney and Ding Zhang. *Projected Dynamical Systems and Variational Inequalities With Applications*, volume 2. Springer Science & Business Media, 2012.

Xinyu Peng, Ziyang Zheng, Wenrui Dai, Nuoqian Xiao, Chenglin Li, Junni Zou, and Hongkai Xiong. Improving diffusion models for inverse problems using optimal posterior covariance. *arXiv preprint arXiv:2402.02149*, 2024.

John Platt and Alan Barr. Constrained differential optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1987.

Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://openreview.net/forum?id=PLIt3a4yTm.

Mehrsa Pourya, Bassam El Rawas, and Michael Unser. Flower: A flow-matching solver for inverse problems. In *International Conference on Learning Representations (ICLR)*, 2025.

Guannan Qu and Na Li. On the exponential stability of primal-dual gradient dynamics. *IEEE Control Systems Letters*, 3(1):43–48, 2019.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. SafeDiffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations (ICLR)*, 2025.

Tianyu Xie, Yu Zhu, Longlin Yu, Tong Yang, Ziheng Cheng, Shiyue Zhang, Xiangyu Zhang, and Cheng Zhang. Reflected flow matching. In *International Conference on Machine Learning (ICML)*, 2024.

Jeongyong Yang, Seunghwan Jang, and SooJean Han. Safeflowmatcher: Safe and fast planning using flow matching with control barrier functions. In *International Conference on Learning Representations*, volume 2026, pp. 54058–54087, 2026.

Zewen Yang, Xiaobing Dai, Dian Yu, Zhijun Li, Majid Khadiv, Sandra Hirche, and Sami Haddadin. UniConFlow: A unified constrained flow-matching framework for certified motion planning. *arXiv preprint arXiv:2506.02955*, 2025.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. FlowPolicy: Enabling fast and robust 3D flow-based policy via consistency flow matching for robot manipulation. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 14754–14762, 2025.

A Implementation Details

Our implementation of LDF, along with baselines and code for reproducing all figures in this paper is available at https://anonymous.4open.science/r/constrained_flow_matching. We use JAX and Flax NNX for model training and deployment; ODEs are solved with Diffrax (Kidger, 2021).

A.1 Constrained Generation Methods

A.1.1 Lagrangian Dual Flows

We implement Algorithm 1 in JAX, leaving the penalty c and dual flow exponent p as free parameters. The correction terms in (5a) are implemented using a single VJP: the explicit Jacobian $J_g(x)$ is never formed or stored in memory.

We wrap the dynamics of the sample x, Lagrange multipliers λ, and any slack variables s into a single ODE, and solve this ODE with Diffrax. For fixed-step integration, we use the second-order midpoint method. For error-controlled integration, we use Diffrax's default step-size controller with a minimum step size of 10^{-5}.

Since the ODE is well-defined only as $t \to 1^-$, we integrate only to $1 - dt$, where dt is the step size. Additionally, we add a small constant $\epsilon = 1 \times 10^{-6}$ to the dual rescaling term such that (5b) is implemented as $\dot{\lambda}_t = g(x_t)/(1 - t + \epsilon)^p$.

A.1.2 Penalty Only Baseline

The penalty-only baseline exercises the same code path as our proposed approach, but we fix $\lambda = 0$. The net ODE is therefore

$$\dot{x}_t = v_\theta(x_t, t) - \frac{c}{2} \nabla_x \|g(x_t)\|^2, \tag{8}$$

a simple quadratic penalty guiding the generation process. For inequality constraints, we instead use the ODE

$$\dot{x}_t = v_\theta(x_t, t) - \frac{c}{2} \nabla_x \|\mathrm{ReLU}(h(x_t))\|^2, \tag{9}$$

which penalizes violations in the inequality constraint $h(x_1) \leq 0$.

A.1.3 Pseudoinverse Guidance Baseline

For pseudoinverse guidance, we use JAX and Diffrax to flow samples according to the ODE defined in (Pokle et al., 2024, Algorithm 1). Note that this algorithm is designed for linearly constrained problems of the form

$$Ax = y. \tag{10}$$

To support generic nonlinear constraints $g(x) = 0$, we simply replace A with the Jacobian $J_g(x_t)$ and the residual $Ax - y$ with $g(x)$.

To maximize runtime performance, we use LU decomposition and a linear solve to avoid full computation of the Moore-Penrose pseudoinverse, and a single VJP to avoid storing the full Jacobian of last-state predictions with respect to the current estimate.

A.1.4 Physics-Constrained Flow Matching Baseline

We similarly implement (Utkarsh et al., 2025, Algorithm 1) using JAX and Diffrax. This PCFM algorithm is considerably more complex than other baselines, involving several interleaved steps of ODE solving, projection, and optimization subproblems. We solve the optimization subproblem (Utkarsh et al., 2025, Algorithm 1, line 8) with 10 steps of gradient descent at learning rate 0.1. We found that PCFM's generation paths were quite sensitive to the projection step (Utkarsh et al., 2025, Algorithm 1, line 6). In particular, performing this projection to a linearization of the constraint manifold resulted in extremely noisy generation paths for the unit circle constraint, as shown in Fig. 5 below.

To avoid this issue, our implementation repeats this projection step N times, re-linearizing the constraint after each projection. We found that $N = 8$ offered a reasonable balance of performance and computation speed for our examples. We use the improved $N = 8$ version as our baseline for all of our numerical experiments.

A.1.5 Control Barrier Function (CBF) Baseline

For the CBF baseline, we implement (Dai et al., 2025, Algorithm 1) in JAX, using QPAX (Arrizabalaga et al., 2026) to solve the CBF-QP. This quadratic program is not feasible in general, so we

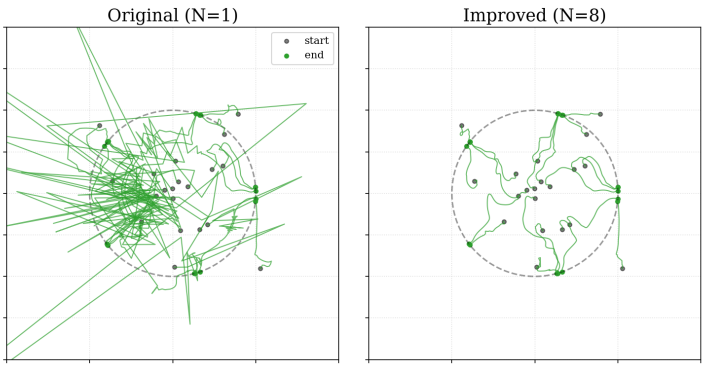

Figure 5: The original PCFM algorithm (left, (Utkarsh et al., 2025, Algorithm 1)) produces extremely noisy generation paths for our star examples, subject to a unit circle constraint. We compare with an improved baseline (right), where increasing the number of projection steps avoids overshoot and results in more reasonable generation paths.

use the relaxed (elastic-mode) formulation to solve it. Note that the analytical QP solution described in (Dai et al., 2025) is not applicable in our case of general inequality constraints.

Note that the need for this relaxation is an indication that we do not have a true control barrier function for the problem at hand. Finding such a barrier function is difficult, requiring careful analysis and a priori knowledge of the constraint set and system dynamics, much like constructing a Lyapunov function. The relaxed solution that we use makes a "best effort" attempt to satisfy these conditions, but they are infeasible at some point in the generation process for all of our examples. As a result, the CBF approach cannot guarantee constraint satisfaction, and in practice sometimes violates constraints or results in divergence during the denoising process.

A.2 Example Details

Here we provide a brief summary regarding our example systems. Full details can be found in the source code at `https://anonymous.4open.science/r/constrained_flow_matching`. Note that our pretrained models and architectures are not particularly optimized or finely tuned; their primary purpose is rather to provide a testbed for applying inference-time constraints.

A.2.1 Equality Constrained Star

We first consider a simple example with nonlinear equality constraints. In particular, we pre-train a small feed-forward network to generate samples from a star-shaped distribution, shown in the leftmost column of Fig. 2. This flow model $v_\theta(x, t)$ is a simple MLP with 4 hidden layers of 64 units and swish activations. We train on 1024 data points generated on a 2D star manifold. Training this small network takes only a few minutes on a laptop.

We then impose a unit circle constraint

$$g(x) = \|x\|^2 - 1$$

at inference time. All methods use 100 steps ($dt = 0.01$) of the second-order midpoint rule for ODE integration.

A.2.2 Inequality Constrained Star

To illustrate LDF's ability to handle inequality constraints, we impose a simple constraint on the same pretrained star model,

$$h\left(\begin{bmatrix} x^0 \\ x^1 \end{bmatrix}\right) = -x^0,$$

which constrains generated samples to the right-hand side of the plane. The resulting samples are shown in Fig. 4a. We again use 100 denoising steps and midpoint integration.

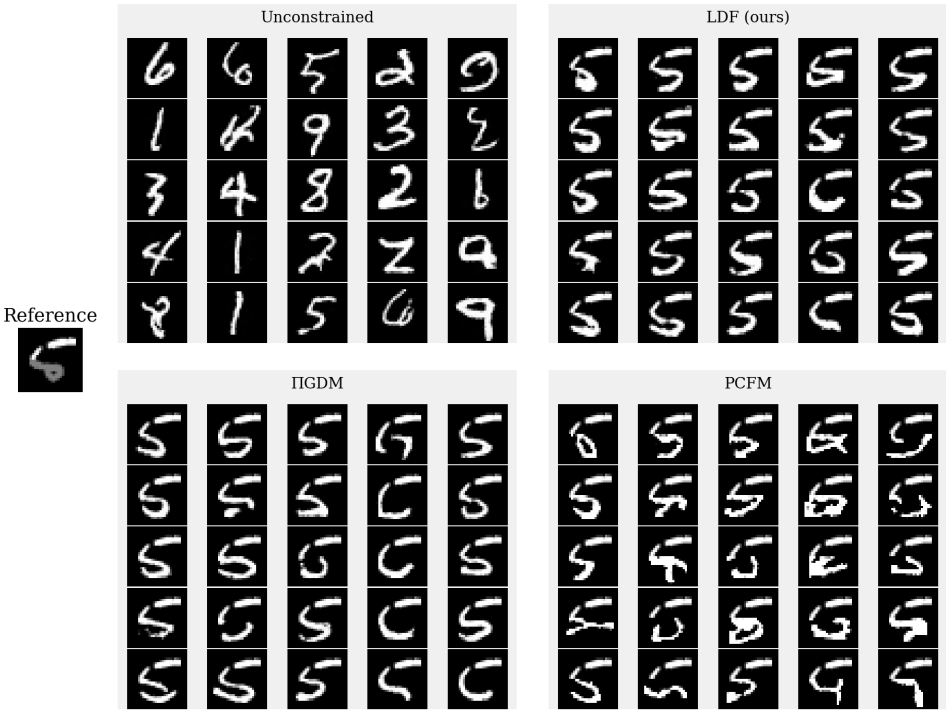

Figure 6: Constrained samples generated with different random seeds.

Adding this constraint is computationally cheap: generating 1000 LDF-constrained samples (in parallel, 100 denoising steps, midpoint integration) takes 0.0121 seconds, compared to 0.0114 seconds for unconstrained generation.

A.2.3 MNIST Inpainting

Image inpainting, where only a certain subset of pixels are generated by a pre-trained model, is a special case of equality constraints $g(x) = 0$. In this case, $g(x)$ is a linear constraint specifying the value of fixed pixels. We train a $\sim$6M parameter UNet to generate MNIST digits, then apply an inpainting constraint, which fixes the top half of the image.

Figure 6 shows samples generated with this inpainting constraint. Both pseudoinverse guidance (Pokle et al., 2024) and our approach produce visually plausible images. Interestingly, while physics-constrained flow matching (Utkarsh et al., 2025) produces the tightest constraint satisfaction (Table 1), clear artifacts are visible in the resulting images. This suggests that the projection substeps used in PCFM may be overly aggressive, pushing samples off of the data manifold in an attempt to enforce constraint satisfaction.

For the pretrained model, we use an encoder/decoder U-net with a (64, 128, 256) dimensional channel schedule. We use a 128-dim sinusoidal time embedding, conditioned via AdaGN (Dhariwal & Nichol, 2021). This model is trained to generate unconditional samples: classification labels are not used during training. Training this model takes about 30 minutes. At inference time, we use 100 denoising steps.

A.3 Obstacle Avoidance

For obstacle avoidance, the unconstrained temporal UNet model is trained to produce a variety of paths between a start and a goal configuration. Samples from the unconstrained model are shown in the left column of Fig. 7. At test time, a random set of circular obstacles is generated. Configurations are rejected if obstacles are overlapping or too close; see the code for full details.

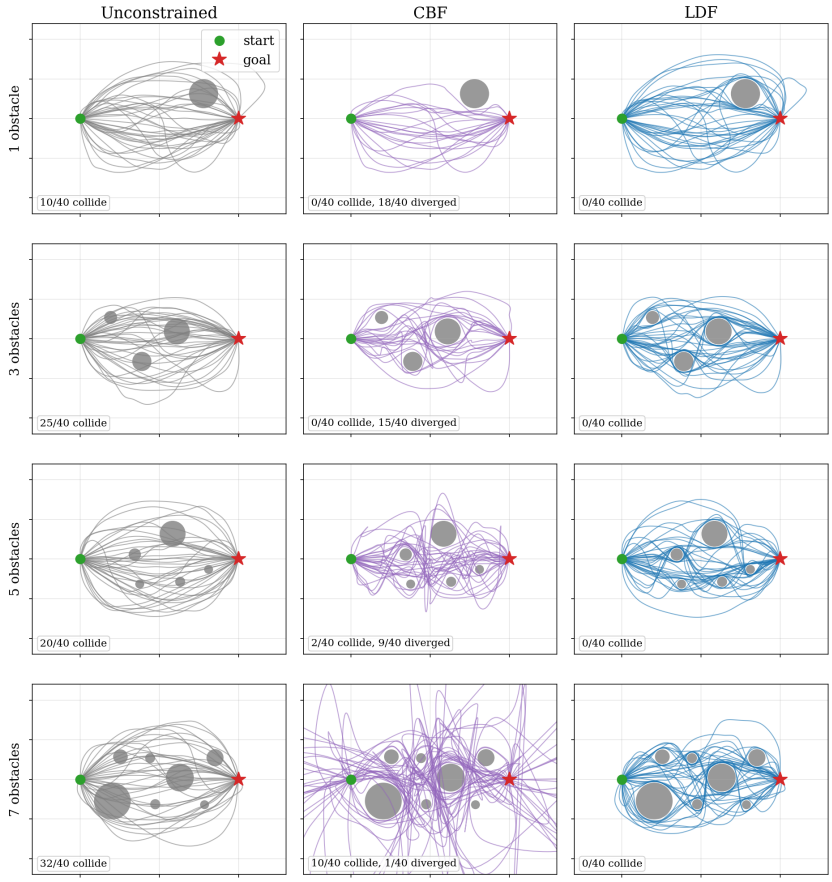

Figure 7: Collision-free path planning comparison between LDF and CBF methods.

Obstacle avoidance constraints are formulated by sampling evenly spaced points along each path, which is represented as a Bezier spline. We use 500 steps of the midpoint rule for this example.

Constrained samples are shown in Fig. 7. Beyond requiring an expensive QP subproblem at each iteration of the denoising process, the CBF method frequently produces collisions and divergences (e.g., NaN errors during the denoising process). In contrast, LDF requires simply integrating a standard ODE and produces reliable performance as the number of obstacles increases.

A.4 Legged Locomotion

For the hopper and walker examples, we train flow models to generate trajectories based on the D4RL (Fu et al., 2020) dataset. Our setup follows (Xiao et al., 2025; Yang et al., 2026). In particular, each trajectory consists of actions (3 actions for the hopper, 6 for the walker) and observations (11 for the hopper, 17 for the walker) over a 32-step horizon. The unconstrained models are 3.3M parameter temporal UNets.

We adopt the constraint, a velocity-dependent maximum height, from (Yang et al., 2026). In particular, the robot height z must remain below a threshold $h_r - \varphi v_z$, where h_r and φ are constants. The hopper uses $(h_r = 1.6, \varphi = 0.1)$ while the walker uses $(h_r = 1.4, \varphi = 0.1)$.

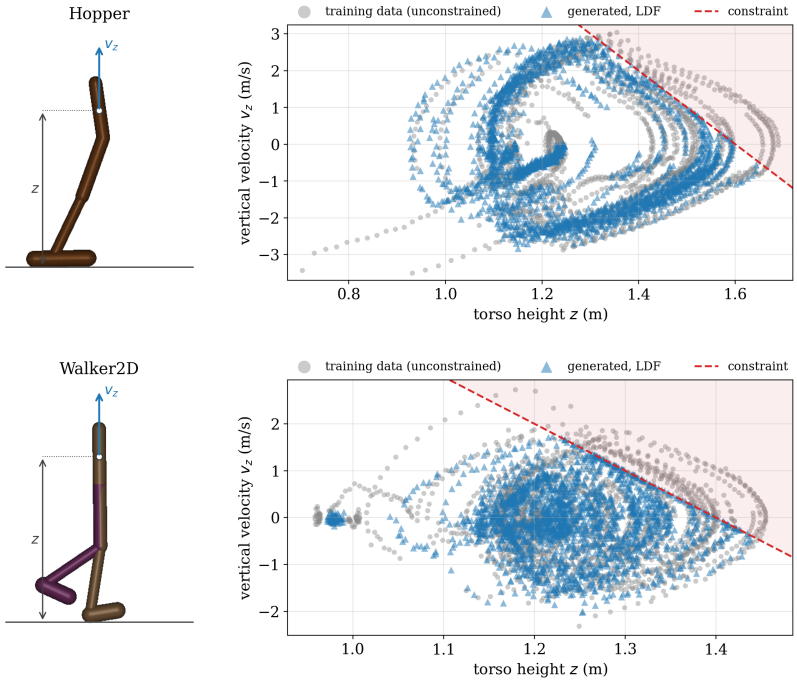

Figure 8: Illustrations and constrained phase portraits for the hopper (top) and walker (bottom) examples. LDF preserves the general character of the pretrained trajectories while enforcing the constraint.

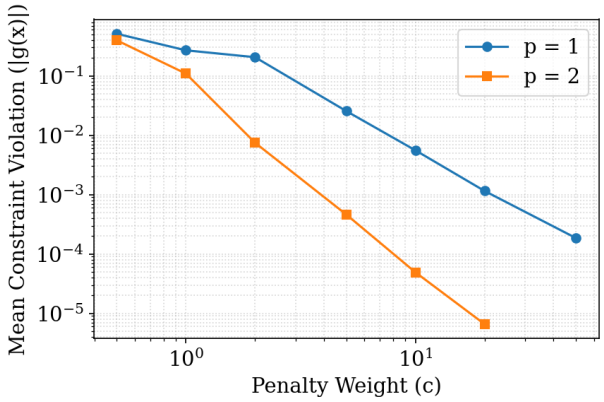

Figure 9: Average constraint violation for the star example, subject to a unit circle constraint.

B Additional Experiments

B.1 Tightening Constraints

Our theoretical analysis indicates that exact constraint satisfaction is achieved (under sufficient regularity conditions) for $p = 2$ and an exact solution of the ODE. We find that performance in practice depends on choices of coefficients c and p, as well as integration accuracy.

We investigate impact of the penalty weight c and time-rescaling factor p on constraint violation. To minimize the impact of integration errors, we use a 5th order Runge-Kutta scheme with error-controlled integration at accuracy 10^{-5}, as implemented in Diffrax (Kidger, 2021). Figure 9 shows

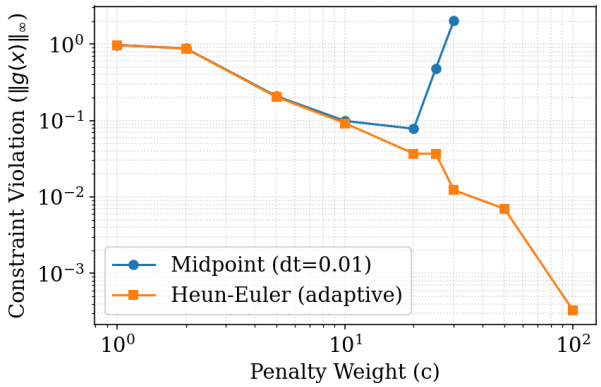

Figure 10: For MNIST image inpainting, constraint tightness saturates around $c = 20$ with a fixed-step midpoint integrator. Error-controlled integration (in this case adaptive-step Heun-Euler) (Hairer et al., 1993) allows us to further tighten constraint satisfaction.

that constraint violation in the final generated samples decreases predictably with the penalty weight c, and that using $p = 2$ provides tighter constraint satisfaction for the same penalty weight.

While $p = 1$ produces reasonable performance in this case, there are cases in which $p = 1$ fails to enforce the constraint for any c. See Appendix D.1 below for details.

B.2 ODE Stiffness

Enforcing constraints with LDF does produce a stiffer ODE than the original unconstrained model. While increasing the penalty c can tighten constraint satisfaction, as shown in Fig. 9 above, this also makes the ODE stiffer and more difficult to integrate.

For fixed-step integration schemes, this results in a tradeoff: higher penalties will produce a stiff-enough ODE that numerical integration error dominates, ultimately increasing constraint violation. This is shown for the MNIST inpainting example in Fig. 10. Error-controlled integration methods (Hairer et al., 1993) avoid this problem by adapting the step size to bound the impact of numerical error. Of course, these adaptive integration methods also tend to introduce some computational overhead.

Developing LDF formulations that limit the stiffness of the denoising ODE is an important area for future research.

B.3 Role of Dual Dynamics

To illustrate the role of the flowed dual λ_t, we compare with the penalty-only approach where we ignore the Lagrangian term. This is similar to the quadratic penalty approach used in (Ho et al., 2022; Chung et al., 2022). Using error-controlled integration as above, we generate samples using a range of penalty weights c. We record the resulting constraint violation, averaged over 200 samples, and number of denoising steps required. The number of denoising steps is a rough proxy for ODE stiffness: a more difficult ODE will require more steps to resolve to the requested accuracy.

The resulting data is shown in Fig. 11. Even with the stiffest penalty $c = 50$, the penalty-only method is only able to reduce constraint violation to around 10^{-3}. More critically, the penalty-only dynamics in this case become extremely stiff, requiring over 800 denoising steps to resolve. Meanwhile, LDF allows us to decrease constraint violations by two orders of magnitude with half the denoising steps.

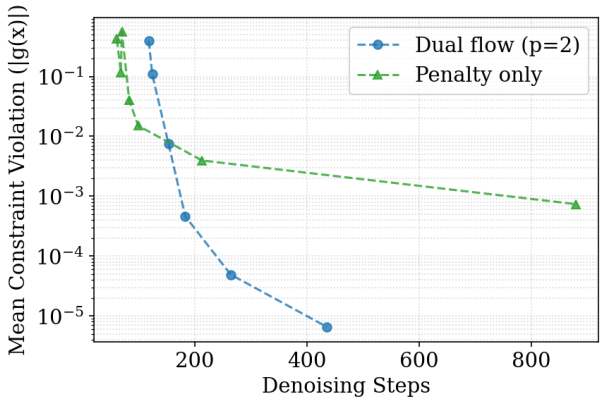

Figure 11: Comparison with a simple penalty method (e.g., $\lambda = 0$). Flowed Lagrangian dynamics allows us to further reduce constraint violations on MNIST inpainting while limiting the stiffness of the ODE.

C Proofs

C.1 Proof of Theorem 1

Proof. The proof has four steps. First, we remove the singularity at $t = 1$ by an exponential time change. Second, we construct a Lyapunov function and verify its positive definiteness and its exact dissipation along the autonomous limit. Third, we decompose the Lyapunov function's derivative along the true dynamics into the autonomous part plus two controlled corrections. Finally, we apply the comparison lemma to the resulting inequality, obtaining an explicit decay rate.

Throughout we abbreviate

$$A(x) := J_g(x)J_g(x)^\top \in \mathbb{R}^{m\times m}, \qquad b(x,t) := J_g(x)\, v_\theta(x,t) \in \mathbb{R}^m.$$

Step 1 (Exponential time change). Introduce $\tau := -\log(1-t)$, mapping $[0,1)$ onto $[0,\infty)$ with $1-t = e^{-\tau}$ and $\mathrm{d}\tau/\mathrm{d}t = e^{\tau}$, and the rescaled dual variable

$$\eta(\tau) := (1-t)\lambda(t) = e^{-\tau}\lambda. \tag{11}$$

The chain rule applied to $g(x_t)$ along (5a) gives $\dot g = J_g(x)\dot x = b(x,t) - A(x)\lambda - c\,A(x)g$. Converting to τ-time via $\mathrm{d}g/\mathrm{d}\tau = e^{-\tau}\dot g$ and $\lambda = e^{\tau}\eta$,

$$\frac{\mathrm{d}g}{\mathrm{d}\tau} = -A(x)\,\eta + e^{-\tau}\big[b(x,t) - c\,A(x)\,g\big]. \tag{12}$$

For the dual variable, $\dot\lambda = g/(1-t)^2$ gives $\mathrm{d}\lambda/\mathrm{d}\tau = e^{-\tau}\dot\lambda = e^{\tau}g$, so differentiating (11),

$$\frac{\mathrm{d}\eta}{\mathrm{d}\tau} = -e^{-\tau}\lambda + e^{-\tau}\frac{\mathrm{d}\lambda}{\mathrm{d}\tau} = -\eta + g. \tag{13}$$

Under assumption (A4), as $\tau \to \infty$ the factor $e^{-\tau}$ vanishes, and (12)-(13) are close to the autonomous linear system $\mathrm{d}g/\mathrm{d}\tau = -A_*\eta$, $\mathrm{d}\eta/\mathrm{d}\tau = g - \eta$, with $A(x)$ frozen at A_*. The deviations $A(x) - A_*$ and the $e^{-\tau}$ forcing are treated as corrections in Step 3.

Step 2 (Lyapunov function). With the reference matrix $A_* \succ 0$ from Assumption (A4), define the candidate Lyapunov function

$$V(g,\eta) := g^\top\big(A_*^{-1} + \tfrac{1}{2}I_m\big)g - g^\top\eta + \tfrac{1}{2}\,\|\eta\|^2 + \tfrac{1}{2}\eta^\top A_*\eta. \tag{14}$$

First, we establish that this candidate is positive definite. Stacking $w := (g,\eta) \in \mathbb{R}^{2m}$, we have $V = \frac{1}{2}w^\top P w$ with

$$P = \begin{pmatrix} 2A_*^{-1} + I_m & -I_m \\ -I_m & I_m + A_* \end{pmatrix}. \tag{15}$$

Diagonalizing $A_* = Q\,\mathrm{diag}(\mu_1,\dots,\mu_m)\,Q^\top$ with Q orthogonal and conjugating P by $\mathrm{diag}(Q,Q)$ block-diagonalizes P into m blocks

$$P_i = \begin{pmatrix} 2\mu_i^{-1}+1 & -1 \\ -1 & 1+\mu_i \end{pmatrix}. \tag{16}$$

Each $P_i \succ 0$ since it is strictly diagonally dominant. Hence $P \succ 0$, so V is positive definite and

$$c_1\big(\|g\|^2+\|\eta\|^2\big) \le V(g,\eta) \le c_2\big(\|g\|^2+\|\eta\|^2\big), \qquad c_1 := \tfrac{1}{2}\lambda_{\min}(P),\ c_2 := \tfrac{1}{2}\lambda_{\max}(P). \tag{17}$$

Dissipation along the autonomous limit. Along $\mathrm{d}g/\mathrm{d}\tau = -A_*\eta$, $\mathrm{d}\eta/\mathrm{d}\tau = g-\eta$ we claim

$$\frac{\mathrm{d}V}{\mathrm{d}\tau}\Big|_{\text{autonomous}} = -\|g\|^2 - \|\eta\|^2. \tag{18}$$

Differentiating (14) term by term:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\tau}\Big[g^\top(A_*^{-1}+\tfrac{1}{2}I)g\Big] &= 2g^\top(A_*^{-1}+\tfrac{1}{2}I)(-A_*\eta) = -2g^\top\eta - g^\top A_*\eta,\\
\frac{\mathrm{d}}{\mathrm{d}\tau}\Big[-g^\top\eta\Big] &= -(-A_*\eta)^\top\eta - g^\top(g-\eta) = \eta^\top A_*\eta - \|g\|^2 + g^\top\eta,\\
\frac{\mathrm{d}}{\mathrm{d}\tau}\Big[\tfrac{1}{2}\|\eta\|^2\Big] &= \eta^\top(g-\eta) = g^\top\eta - \|\eta\|^2,\\
\frac{\mathrm{d}}{\mathrm{d}\tau}\Big[\tfrac{1}{2}\eta^\top A_*\eta\Big] &= \eta^\top A_*(g-\eta) = g^\top A_*\eta - \eta^\top A_*\eta,
\end{aligned}$$

using the symmetry of A_* for $(A_*\eta)^\top\eta = \eta^\top A_*\eta$ and $\eta^\top A_* g = g^\top A_*\eta$. Summing, we are left with $-\|g\|^2 - \|\eta\|^2$, which is (18).

Step 3 (Decomposition along the true dynamics). Now we differentiate V along (12)-(13), where $A(x)$ replaces A_* and the forcing $e^{-\tau}[b - cAg]$ is present. With $\nabla_g V = 2(A_*^{-1}+\frac{1}{2}I)g - \eta$ and $\nabla_\eta V = -g + \eta + A_*\eta$, substituting and decomposing $A(x) = A_* + (A(x)-A_*)$ gives

$$\frac{\mathrm{d}V}{\mathrm{d}\tau} = -\|g\|^2 - \|\eta\|^2 + R_A + e^{-\tau}R_p, \tag{19}$$

where the autonomous part is exactly (18), and

$$R_A := -2g^\top\big(A_*^{-1}+\tfrac{1}{2}I\big)(A(x)-A_*)\eta + \eta^\top(A(x)-A_*)\eta, \tag{20}$$

$$R_p := \big(2g^\top\big(A_*^{-1}+\tfrac{1}{2}I\big) - \eta^\top\big)\xi, \qquad \xi := b(x,t) - c\,A(x)\,g. \tag{21}$$

Bound on R_A. By Cauchy–Schwarz on each term of (20) and Young's inequality $2\|g\|\|\eta\| \le \|g\|^2 + \|\eta\|^2$,

$$\begin{aligned}
|R_A| &\le 2\|g\|\,\big\|A_*^{-1}+\tfrac{1}{2}I\big\|_2\,\|A(x)-A_*\|_2\,\|\eta\| + \|A(x)-A_*\|_2\,\|\eta\|^2\\
&\le \|A(x)-A_*\|_2\Big[\big\|A_*^{-1}+\tfrac{1}{2}I\big\|_2\big(\|g\|^2+\|\eta\|^2\big) + \|\eta\|^2\Big]\\
&\le \big(\big\|A_*^{-1}+\tfrac{1}{2}I\big\|_2 + 1\big)\,\|A(x)-A_*\|_2\big(\|g\|^2+\|\eta\|^2\big)\\
&= K_{A_*}\,\|A(x)-A_*\|_2\big(\|g\|^2+\|\eta\|^2\big),
\end{aligned}$$

using $\|\eta\|^2 \le \|g\|^2 + \|\eta\|^2$. By Assumption (A4), $K_{A_*}\|A(x)-A_*\|_2 \le \rho$, so

$$|R_A| \le \rho\big(\|g\|^2+\|\eta\|^2\big). \tag{22}$$

Bound on R_p. The vector ξ is bounded: by (A1)-(A3), $\|\xi\| \le \|b\| + c\,\|A\|_2\,\|g\| \le LB + c\,a_{\max}G =: M_\xi$. Then by Cauchy-Schwarz,

$$|R_p| \le 2\|g\|\,\big\|A_*^{-1}+\tfrac{1}{2}I\big\|_2\,\|\xi\| + \|\eta\|\,\|\xi\| \le K_p\big(\|g\|+\|\eta\|\big), \tag{23}$$

with $K_p := M_\xi\,\max\{2\big\|A_*^{-1}+\frac{1}{2}I\big\|_2, 1\}$ depending only on $L, B, c, a_{\max}, A_*, G$.

Combined inequality. Substituting (22) and (23) into (19),

$$\frac{\mathrm{d}V}{\mathrm{d}\tau} \le -(1-\rho)\big(\,\|g\|^2 + \|\eta\|^2\,\big) + e^{-\tau}K_p\big(\,\|g\| + \|\eta\|\,\big). \tag{24}$$

Step 4 (Comparison lemma and explicit rate). Write $\|w\|^2 := \|g\|^2 + \|\eta\|^2$. Since $\|g\| + \|\eta\| \le \sqrt{2}\,\|w\|$, the combined inequality (24) gives

$$\frac{\mathrm{d}V}{\mathrm{d}\tau} \le -(1-\rho)\,\|w\|^2 + e^{-\tau}\sqrt{2}\,K_p\,\|w\|\,.$$

Set $y := \sqrt{V} \ge 0$. Where $V > 0$, using $\|w\|^2 \ge V/c_2 = y^2/c_2$ on the dissipation term and $\|w\| \le \sqrt{V/c_1} = y/\sqrt{c_1}$ on the forcing term (both from (17)),

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} = \frac{1}{2\sqrt{V}}\frac{\mathrm{d}V}{\mathrm{d}\tau} \le -\alpha y + \gamma e^{-\tau}, \qquad \alpha := \frac{1-\rho}{2c_2} > 0, \quad \gamma := \frac{K_p}{\sqrt{2c_1}} > 0.$$

The comparison lemma (Khalil, 2002, Lemma 3.4) then gives, for all $\tau \ge 0$,

$$y(\tau) \le y(0)e^{-\alpha\tau} + \gamma\int_0^\tau e^{-\alpha(\tau-u)}e^{-u}\,\mathrm{d}u = y(0)e^{-\alpha\tau} + \gamma\begin{cases}\dfrac{e^{-\alpha\tau}-e^{-\tau}}{1-\alpha}, & \text{if } \alpha \ne 1,\\ \tau\,e^{-\tau}, & \text{if } \alpha = 1.\end{cases} \tag{25}$$

Both terms on the right of (25) decay so we can conclude that $y(\tau) \to 0$ as $\tau \to \infty$.

Additionally, for all $i \in \{1,\dots,m\}$, $\lambda_{\max}(P) \ge P_i^{11} = 2\mu_i^{-1} + 1 > 1$ for a symmetric matrix, and $2c_2 = \lambda_{\max}(P)$ with $\rho \ge 0$, we have $\alpha = (1-\rho)/\lambda_{\max}(P) < 1$ and thus there is a constant $M > 0$ such that $y(\tau) \le M\,e^{-\alpha\tau}$ for all $\tau \ge 0$. Converting to original time via $e^{-\tau} = 1 - t$ and using $\|g(x_t)\| \le \|w\| \le y/\sqrt{c_1}$,

$$\|g(x_t)\| \le \frac{M}{\sqrt{c_1}}\,(1-t)^\alpha = O\big((1-t)^\alpha\big) \quad \text{and} \quad (1-t)\,\|\lambda_t\| = \|\eta\| = O\big((1-t)^\alpha\big)$$

as $t \to 1^-$. In particular both tend to zero. □

C.2 Proof of Lemma 2 and Theorem 3

Proof of Lemma 2. If the i-th component of s_t, s_t^i, is equal to zero at some time $t \in [0,1)$, then from (6b), $\dot{s}_t^i = c\ \mathrm{ReLU}(-h^i(x_t) - \lambda_t^i/c) \ge 0$. Hence s_t^i cannot decrease past zero and with $s_0 \ge 0$, we conclude that $s_t \ge 0$ for all $t \in [0,1)$. □

For self-containedness, we reproduce the assumptions for convergence here, stated in terms of h. We assume the following hold along the trajectory of (6a)-(6c).

- (B1) *Smoothness:* $h \in C^2(\mathbb{R}^n;\mathbb{R}^k)$ and $v_\theta \in C^1(\mathbb{R}^n \times [0,1);\mathbb{R}^n)$, with $\|J_h(x)\|_2 \le L$ and $\|v_\theta(x,t)\| \le B$;
- (B2) *Constraint qualification:* $a_{\min}I_k \preceq J_h(x)J_h(x)^\top \preceq a_{\max}I_k$ for constants $0 < a_{\min} \le a_{\max} < \infty$.
- (B3) *Bounded trajectory:* the trajectory (x_t, s_t, λ_t) exists on $[0,1)$ with x_t and s_t in a compact set.

These assumptions are identical to those assumed in the equality-constrained case with the additional assumption that the slack variable s_t evolves in a compact set. This additional assumption can be enforced by replacing the $\mathrm{ReLU}(-h(x_t) - \lambda_t/c)$ in (6b) with $\min\{\mathrm{ReLU}(-h(x_t) - \lambda_t/c), R\}$ for some $R > 0$. This enforces boundedness at the expense of enforcing the two-sided constraint $-R \le h(x_1) \le 0$ for the generated sample where we understand $-R \le h(x_1)$ to mean that each entry of $h(x_1)$ is greater than or equal to $-R$.

Unlike the equality case, the soft projection switches between active sets, so the proof relies on a Lyapunov function that dissipates simultaneously in every possible regime. Because the dissipation margin κ_0 it produces enters the small-variation condition, we state it first.

First, for notation, for a subset $\mathcal{S} \subseteq \{1,\dots,k\}$, let $E_{\mathcal{S}} := \mathrm{diag}(\mathbf{1}_{i\in\mathcal{S}})$ be the corresponding $0/1$ diagonal matrix.

Lemma 4. *Let $A_* \succ 0$ and define*

$$V(\tilde{g},\eta) := \tfrac{1}{2}(\tilde{g},\eta)^\top P\,(\tilde{g},\eta), \qquad P := \begin{pmatrix} I_k & -\frac{1}{2}I_k \\ -\frac{1}{2}I_k & A_* + \frac{1}{2}I_k \end{pmatrix}. \tag{26}$$

Then $P \succ 0$, and for every *projector $E_{\mathcal{S}}$ ($\mathcal{S} \subseteq \{1,\dots,k\}$), the matrix $M_{\mathcal{S}} := \begin{pmatrix} 0 & -(A_*+E_{\mathcal{S}}) \\ I & -I \end{pmatrix}$ satisfies*

$$M_{\mathcal{S}}{}^\top P + P M_{\mathcal{S}} = \begin{pmatrix} -I & I - E_{\mathcal{S}} \\ I - E_{\mathcal{S}} & E_{\mathcal{S}} - A_* - I \end{pmatrix} =: D_{\mathcal{S}} \prec 0. \tag{27}$$

Consequently the dissipation margin

$$\kappa_0 := \kappa_0(A_*) := \frac{1}{2} \min_{\mathcal{S}\subseteq\{1,\dots,k\}} \lambda_{\min}(-D_{\mathcal{S}}) > 0 \tag{28}$$

is uniform over all active sets, and $\mathrm{d}V/\mathrm{d}\tau \le -\kappa_0\Big(\|\tilde{g}\|^2 + \|\eta\|^2\Big)$ along $\mathrm{d}\tilde{g}/\mathrm{d}\tau = -(A_* + E_{\mathcal{S}})\eta$, $\mathrm{d}\eta/\mathrm{d}\tau = \tilde{g} - \eta$. Moreover $\nabla_{\tilde{g}} V = \tilde{g} - \frac{1}{2}\eta$, so $\|\nabla_{\tilde{g}} V\| \le \frac{\sqrt{5}}{2}\,\|(\tilde{g},\eta)\|$.

The proof of Lemma 4 is given at the end of this subsection.

With κ_0 defined, we state the small-variation condition that replaces Assumption (A4) in the inequality setting. It is the exact analog of the equality case, with the dissipation margin κ_0 playing the role of the constant 1.

(B4) *Small variation:* there exists a reference matrix $A_* \succ 0$ such that along the trajectory

$$\rho := \frac{\sqrt{5}}{2} \sup_t \Big\| J_h(x_t) J_h(x_t)^\top - A_* \Big\|_2 < \kappa_0(A_*). \tag{29}$$

As in the equality case, (B4) holds automatically with $\rho = 0$ for affine constraints $h(x) = Hx + d$: then $J_h J_h{}^\top = HH^\top$ is constant, and the choice $A_* = HH^\top$ gives $\rho = 0 < \kappa_0$.

Proof of Theorem 3. The proof follows the same four steps as the equality-constrained case (Theorem 1). As in the equality case, the nonlinear constraint is handled directly by working with the reference matrix A_* of the small-variation condition (29). Mirroring the abbreviations $A(x) = J_g J_g{}^\top$, $b(x,t) = J_g\, v_\theta$ of the equality case, we set

$$A(x) := J_h(x) J_h(x)^\top, \qquad b(x,t) := J_h(x)\, v_\theta(x,t), \qquad r := \mathrm{ReLU}\Big(-h(x) - \tfrac{1}{c}\lambda\Big), \tag{30}$$

so that the slack equation (6b) reads $\dot{s} = c(r - s)$.

Write $\tilde{g}(x,s) := h(x) + s$. By Lemma 2 and $s_0 \ge 0$ we have $s_t \ge 0$ for all t, hence $\|\mathrm{ReLU}(h(x_t))\| \le \|\tilde{g}(x_t,s_t)\|$ and it suffices to prove $\|\tilde{g}(x_t,s_t)\| \to 0$ as $t \to 1^-$. By Assumptions (B1)–(B3) we may set $G := \sup_t \|h(x_t)\| < \infty$, $S := \sup_t \|s_t\| < \infty$, and $\|b(x_t,t)\| \le LB$.

Step 1 (Exponential time change). Let $\tau := -\log(1-t)$, so $1 - t = e^{-\tau}$, and $\eta := (1-t)\lambda = e^{-\tau}\lambda$. Applying the chain rule to $\tilde{g} = h + s$, $\dot{\tilde{g}} = J_h(x)\dot{x} + \dot{s}$; substituting (6a), (6b) and converting to τ-time via $\mathrm{d}\tilde{g}/\mathrm{d}\tau = e^{-\tau}\dot{\tilde{g}}$ and $\lambda = e^{\tau}\eta$,

$$\frac{\mathrm{d}\tilde{g}}{\mathrm{d}\tau} = -A(x)\,\eta + e^{-\tau}\Big[b(x,t) - c\,A(x)\,\tilde{g}\Big] + c\,e^{-\tau}\,(r - s), \tag{31}$$

For the dual variable, $\dot{\lambda} = \tilde{g}/(1-t)^2$ gives, exactly as in the equality case,

$$\frac{\mathrm{d}\eta}{\mathrm{d}\tau} = \tilde{g} - \eta, \qquad \text{and} \qquad \frac{\mathrm{d}s}{\mathrm{d}\tau} = e^{-\tau}\dot{s} = c\,e^{-\tau}\,(r - s). \tag{32}$$

The ReLU acts componentwise; let $\mathcal{S}(\tau) := \{i \mid -h^i - \frac{1}{c}\lambda^i > 0\}$ be the active set (the indices where ReLU is identity) and $E_{\mathcal{S}} := \mathrm{diag}(\mathbf{1}_{i\in\mathcal{S}})$ the corresponding $0/1$ diagonal matrix. On active

components $r^i = -h^i - \frac{1}{c}\lambda^i$; substituting (using $\lambda = e^\tau \eta$, $h + s = \tilde{g}$) the term $c\,e^{-\tau} r$ contributes $-\eta_i$ to $\mathrm{d}\tilde{g}_i/\mathrm{d}\tau$ beyond $-A(x)\eta$, while inactive components contribute $r^i = 0$. Hence

$$\frac{\mathrm{d}\tilde{g}}{\mathrm{d}\tau} = -\Big(A(x) + E_{\mathcal{S}}\Big)\eta + e^{-\tau}\Big[b - c(A(x) + E_{\mathcal{S}})\tilde{g}\Big] - c\,e^{-\tau} s_{\mathcal{S}^c}, \tag{33}$$

where $\mathcal{S}^c$ is the complement of $\mathcal{S}$ and $s_{\mathcal{S}^c} = E_{\mathcal{S}^c} s$. As $\tau \to \infty$ the factor $e^{-\tau}$ vanishes, and (33) together with $\mathrm{d}\eta/\mathrm{d}\tau = \tilde{g} - \eta$ approaches, in each active-set regime $\mathcal{S}$, the switching system $\mathrm{d}\tilde{g}/\mathrm{d}\tau = -(A(x) + E_{\mathcal{S}})\eta$, $\mathrm{d}\eta/\mathrm{d}\tau = \tilde{g} - \eta$. Freezing $A(x)$ at a fixed reference A_* renders this autonomous in each active set. Because the active set $\mathcal{S}$ may change along the trajectory, this is a linear switching system and this Lyapunov function must dissipate simultaneously in every regime.

Step 2 (Lyapunov function). Let $A_* \succ 0$ be the reference matrix of the small-variation condition (29), and take the Lyapunov function V of Lemma 4 built from A_*. Write $z := (\tilde{g}, \eta)$ so that $\|z\|^2 = \|\tilde{g}\|^2 + \|\eta\|^2$. By Lemma 4, $P \succ 0$, so V is positive definite and, with $c_1 := \frac{1}{2}\lambda_{\min}(P)$ and $c_2 := \frac{1}{2}\lambda_{\max}(P)$,

$$c_1 \|z\|^2 \le V(\tilde{g}, \eta) \le c_2 \|z\|^2 . \tag{34}$$

Dissipation along the autonomous limit. Fix an active set $\mathcal{S}$. Along the autonomous system $\mathrm{d}\tilde{g}/\mathrm{d}\tau = -(A_* + E_{\mathcal{S}})\eta$, $\mathrm{d}\eta/\mathrm{d}\tau = \tilde{g} - \eta$ obtained by freezing $A(x)$ at A_* and dropping the $e^{-\tau}$ forcing in (33), Lemma 4 gives

$$\frac{\mathrm{d}V}{\mathrm{d}\tau}\Big|_{\text{autonomous}} = z^\top P\, M_{\mathcal{S}}\, z \le -\kappa_0 \|z\|^2 , \tag{35}$$

with $M_{\mathcal{S}}$ from Lemma 4, and $\kappa_0 > 0$ independent of $\mathcal{S}$.

Step 3 (Decomposition along the true dynamics). Fix the active set $\mathcal{S}$ and split the constraint matrix in (33) as $A(x) = A_* + \Delta(x)$, $\Delta(x) := A(x) - A_*$. Differentiating V along (33) and the η-dynamics (32) and substituting this splitting,

$$\frac{\mathrm{d}V}{\mathrm{d}\tau} = \underbrace{z^\top P\, M_{\mathcal{S}} z}_{\le -\kappa_0\|z\|^2 \text{ (35)}} \underbrace{- (\nabla_{\tilde{g}} V)^\top \Delta(x)\,\eta}_{=: R_\Delta} + (\nabla_{\tilde{g}} V)^\top \underbrace{\Big(e^{-\tau}\Big[b - c(A(x) + E_{\mathcal{S}})\tilde{g}\Big] - c\,e^{-\tau} s_{\mathcal{S}^c}\Big)}_{\text{forcing}},$$

where $\nabla_{\tilde{g}} V = \tilde{g} - \frac{1}{2}\eta$. As in the equality case, the deviation $\Delta(x)$ enters only through the constraint block, and does so identically in every regime, since it shifts each $A_* + E_{\mathcal{S}}$ by the same amount.

Bound on R_Δ (Jacobian variation). Using $\|\nabla_{\tilde{g}} V\| \le \frac{\sqrt{5}}{2}\|z\|$ from Lemma 4 and $\|\Delta(x)\eta\| \le \|\Delta(x)\|_2 \|\eta\| \le \|\Delta(x)\|_2 \|z\|$,

$$|R_\Delta| \le \tfrac{\sqrt{5}}{2}\, \|\Delta(x)\|_2 \|z\|^2 \le \rho\, \|z\|^2 ,$$

where $\rho = \frac{\sqrt{5}}{2} \sup_t \|\Delta(x_t)\|_2$ as in (29).

Bound on the forcing. By (B1), $\|b\| \le LB$; by (B2), $\|A(x) + E_{\mathcal{S}}\|_2 \le a_{\max} + 1$; by (B3), $\|\tilde{g}\| \le \|h\| + \|s\| \le G + S =: G'$ and $\|s\| \le S$. Hence the forcing is bounded in norm by $e^{-\tau} B_0$ with $B_0 := LB + c(a_{\max} + 1)G' + cS$, and by Cauchy–Schwarz,

$$(\nabla_{\tilde{g}} V)^\top (\text{forcing}) \le e^{-\tau} \tfrac{\sqrt{5}}{2} B_0 \|z\| =: e^{-\tau} K \|z\| , \qquad K = \tfrac{\sqrt{5}}{2}\Big(LB + c(a_{\max} + 1)G' + cS\Big).$$

Combined inequality. Collecting the three terms,

$$\frac{\mathrm{d}V}{\mathrm{d}\tau} \le -(\kappa_0 - \rho)\|z\|^2 + e^{-\tau} K \|z\| , \tag{36}$$

with $\kappa_0 - \rho > 0$ by the small-variation condition (29).

Step 4 (Comparison lemma and explicit rate). Set $y := \sqrt{V} \ge 0$. Where $V > 0$, $\frac{\mathrm{d}y}{\mathrm{d}\tau} = \frac{1}{2\sqrt{V}} \frac{\mathrm{d}V}{\mathrm{d}\tau}$; bounding the dissipation term in (36) from below via $\|z\|^2 \ge V/c_2 = y^2/c_2$ and the forcing term from above via $\|z\| \le \sqrt{V/c_1} = y/\sqrt{c_1}$ (both from (34)),

$$\frac{\mathrm{d}y}{\mathrm{d}\tau} \le \frac{1}{2y}\Big[-(\kappa_0 - \rho)\frac{y^2}{c_2}\Big] + \frac{1}{2y}\, e^{-\tau} K \frac{y}{\sqrt{c_1}} = -\alpha y + \gamma e^{-\tau},$$

where $\alpha := \frac{\kappa_0 - \rho}{2c_2} > 0$ and $\gamma := \frac{K}{2\sqrt{c_1}} > 0$. We are left with a scalar differential inequality for y. The comparison lemma (Khalil, 2002, Lemma 3.4) implies that for all $\tau \in [0, \infty)$

$$y(\tau) \le y(0)e^{-\alpha\tau} + \gamma \int_0^\tau e^{-\alpha(\tau-u)} e^{-u}\, \mathrm{d}u = y(0)e^{-\alpha\tau} + \gamma \begin{cases} \dfrac{e^{-\alpha\tau} - e^{-\tau}}{1-\alpha}, & \alpha \neq 1, \\ \tau\, e^{-\tau}, & \alpha = 1. \end{cases}$$

We thus conclude that $y(\tau) \to 0$. Moreover, the top-left block of P in Lemma 4 is I_k, so $\lambda_{\max}(P) \ge 1$, and with $\kappa_0 \le \frac{1}{2}$ and $\rho \ge 0$ we have $\alpha = (\kappa_0 - \rho)/\lambda_{\max}(P) \le \kappa_0 \le \frac{1}{2} < 1$. Therefore, there exists a constant $M > 0$ such that $y(\tau) \le M\, e^{-\alpha\tau}$ for all $\tau \ge 0$. Converting to original time via $e^{-\tau} = 1 - t$, and using $\|\mathrm{ReLU}(h(x_t))\| \le \|\tilde{g}\| \le \|z\| \le y/\sqrt{c_1}$,

$$\|\mathrm{ReLU}(h(x_t))\| \le \frac{M}{\sqrt{c_1}} (1-t)^\alpha = O\big((1-t)^\alpha\big) \quad \text{and} \quad (1-t)\|\lambda_t\| = \|\eta\| = O\big((1-t)^\alpha\big),$$

as $t \to 1^-$. In particular $\|\mathrm{ReLU}(h(x_t))\| \to 0$, so the generated sample is feasible in the limit. □

We now prove Lemma 4.

Proof of Lemma 4. Positive definiteness. The top-left block of P is $I \succ 0$, and its Schur complement is $(A_* + \frac{1}{2}I) - (\frac{1}{2}I)(\frac{1}{2}I)^\top = A_* + \frac{1}{4}I \succ 0$, so $P \succ 0$.

Dissipation. A direct block computation gives ${M_\mathcal{S}}^\top P + P M_\mathcal{S} = \left(\begin{smallmatrix} -I & I-E \\ I-E & E-A_*-I \end{smallmatrix}\right) =: D_\mathcal{S}$. The top-left block is $-I \prec 0$; the Schur complement of $D_\mathcal{S}$ is

$$(E_\mathcal{S} - A_* - I) - (I - E_\mathcal{S})(-I)^{-1}(I - E_\mathcal{S}) = (E_\mathcal{S} - A_* - I) + (I - E_\mathcal{S})^2.$$

Because $E_\mathcal{S}$ is a $0/1$ diagonal projector, $E_\mathcal{S}^2 = E_\mathcal{S}$, hence $(I - E_\mathcal{S})^2 = I - E_\mathcal{S}$, and the Schur complement collapses to

$$(E_\mathcal{S} - A_* - I) + (I - E_\mathcal{S}) = -A_* \prec 0.$$

Thus $D_\mathcal{S} \prec 0$ for every $E_\mathcal{S}$. Since the active sets $\mathcal{S}$ form a finite set, $\kappa_0 = \frac{1}{2} \min_\mathcal{S} \lambda_{\min}(-D_\mathcal{S}) > 0$. Along the autonomous flow, $\mathrm{d}V/\mathrm{d}\tau = z^\top P M_\mathcal{S} z = \frac{1}{2} z^\top D_\mathcal{S} z \le -\frac{1}{2}\lambda_{\min}(-D_\mathcal{S}) \|z\|^2 \le -\kappa_0 \|z\|^2$, which gives the stated bound.

Gradient bound. $\nabla_{\tilde{g}} V = \tilde{g} - \frac{1}{2}\eta$, so by Cauchy-Schwarz, $\|\nabla_{\tilde{g}} V\| \le \|\tilde{g}\| + \frac{1}{2}\|\eta\| \le \frac{\sqrt{5}}{2} \|(\tilde{g}, \eta)\|$. □

D Different Choices for p

In this section, we provide some more analysis for the equality-constrained case for $p \neq 2$. To summarize our results,

- For $p \in [1, 2)$, we show that the residual $\|g(x_t)\|$ converges to a limit as $t \to 1^-$, but this limit is generally nonzero;
- $p = 2$ is the case studied in Theorem 1. We prove constraint satisfaction under Assumptions (A1)-(A4);
- For $p > 2$, we do not have general convergence results though analysis of a simplified scalar setting in Appendix D.2 indicates that we can see constraint satisfaction in this case.

To facilitate the presentation, we first present an essential change of coordinates. For $p \in [1, \infty)$, define

$$\eta(t) := (1-t)^{p-1}\lambda(t), \qquad \tau := -\log(1-t), \tag{37}$$

so $\tau \in [0, \infty)$ and $1 - t = e^{-\tau}$. Note that when $p = 1$, $\eta(t) = \lambda(t)$.

Using these new coordinates, our dynamics become

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = -e^{(p-2)\tau} \nabla g(x)^\top \eta + e^{-\tau} F(x, \tau), \tag{38}$$

$$\frac{\mathrm{d}\eta}{\mathrm{d}\tau} = g(x) - (p-1)\eta, \tag{39}$$

where $F(x, \tau) := v_\theta(x, t(\tau)) - c\, \nabla g(x)^\top g(x)$.

D.1 $p \in [1, 2)$

The exponent p enters (38) only through the gain $e^{(p-2)\tau}$ multiplying the dual correction, and this gain determines the steady-state behavior of these dynamics. At $p = 2$ the gain is constant, and (38)-(39) approach an autonomous system as $\tau \to \infty$ (this is the case studied in Theorem 1). For $p > 2$ the gain grows without bound. For $p \in [1, 2)$, by contrast, the gain decays exponentially in τ. The following proposition makes this precise.

Proposition 5. *Let $p \in [1, 2)$ and suppose Assumptions* (A1)-(A3) *hold along the trajectory of* (5)*. Then there exists $x_1^\star \in \mathbb{R}^n$ such that $x_t \to x_1^\star$ as $t \to 1^-$, and hence $g(x_t) \to g_1 := g(x_1^\star)$. The constraint is satisfied in the limit if and only if $g_1 = 0$, which the dynamics do not enforce.*

Proof. By Assumptions (A1) and (A3), there are constants $G, L, B \geq 0$ with $\|g(x_\tau)\| \leq G$, $\|\nabla g(x_\tau)\|_2 \leq L$, and $\|F(x_\tau, \tau)\| \leq B + cLG =: M_F$ along the trajectory.

First we bound η. Solving the linear equation (39) by variation of constants, with $\eta(0) = \lambda_0 = 0$,

$$\eta(\tau) = \int_0^\tau e^{-(p-1)(\tau-u)}\, g(x(u))\, \mathrm{d}u,$$

so for $p \in (1, 2)$ we get $\|\eta(\tau)\| \leq G/(p-1)$ for all τ, and for $p = 1$ we get $\|\eta(\tau)\| \leq G\tau$.

Substituting these bounds into (38),

$$\left\| \frac{\mathrm{d}x}{\mathrm{d}\tau} \right\| \leq L\, e^{(p-2)\tau} \|\eta(\tau)\| + M_F\, e^{-\tau} =: \varphi(\tau),$$

and since $p - 2 < 0$, the function φ is integrable on $[0, \infty)$ in both cases.

Since $\mathrm{d}x/\mathrm{d}\tau$ is bounded and integrable on $[0, \infty)$, the following integral is absolutely convergent which lets us define the limit

$$x_1^\star := x_0 + \int_0^\infty \frac{\mathrm{d}x}{\mathrm{d}\tau} \mathrm{d}\tau. \tag{40}$$

We can then see that $\|x(\tau) - x_1^\star\| \leq \int_\tau^\infty \varphi(u)\mathrm{d}u \to 0$ as $\tau \to \infty$. Therefore we conclude that $x(\tau) \to x_1^\star$ as $\tau \to \infty$ (or equivalently that $x_t \to x_1^\star$ as $t \to 1^-$) and that $g(x_t) \to g(x_1^\star)$ as $t \to 1^-$ by continuity. □

We exhibit a simple instance in which the failure described by Proposition 5 can be established by a direct argument in the τ-coordinates. Let $n = m = 1$ with $g(x) = x$, and take $v_\theta = 0$ and $c = 0$: the flow has no drift and no penalty, so the dual variable is the only mechanism acting on the sample, and its only task is to bring x to the constraint set $\{x = 0\}$. With $p = 1$ we have $\eta = \lambda$, and the general dynamics (38)-(39) become the simple system

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = -e^{-\tau}\, \eta, \qquad \frac{\mathrm{d}\eta}{\mathrm{d}\tau} = x, \qquad x(0) = x_0, \quad \eta(0) = 0, \tag{41}$$

where $\tau = \log \frac{1}{1-t} \in [0, \infty)$ and constraint satisfaction as $t \to 1^-$ corresponds to $x(\tau) \to 0$ as $\tau \to \infty$.

Integrating the two equations of (41) from 0 gives the two identities

$$\eta(\tau) = \int_0^\tau x(u)\, \mathrm{d}u, \qquad x(\tau) = x_0 - \int_0^\tau e^{-u}\, \eta(u)\, \mathrm{d}u. \tag{42}$$

Proposition 6. *Let $x_0 > 0$. Along* (41)*, $x(\tau) > 0$ and is strictly decreasing for all $\tau > 0$, and*

$$x_\infty := \lim_{\tau \to \infty} x(\tau) \quad \textit{exists with} \quad 0 < x_\infty \leq \tfrac{1}{2} x_0.$$

Hence $x_\infty \neq 0$ whenever $x_0 \neq 0$, and the constraint is not satisfied in the limit.

Proof. We will use the following fact several times:

$$\int_0^\infty u\, e^{-u}\, \mathrm{d}u = 1. \tag{43}$$

First we show that when $x_0 > 0$, we have that $x_\tau > 0$ for all $\tau > 0$. To this end, let $\tau^\star := \sup\{T \mid x(u) > 0 \text{ for all } u \in [0, T)\}$, which is positive since $x(0) = x_0 > 0$ and x is continuous. On $[0, \tau^\star)$ we have $x > 0$, so by the first identity in (42) $\eta > 0$, hence $\mathrm{d}x/\mathrm{d}\tau = -e^{-\tau}\eta < 0$: the residual is strictly decreasing, and in particular $x(u) < x_0$ for $u \in (0, \tau^\star)$. Therefore $\eta(u) = \int_0^u x(r)\,\mathrm{d}r < x_0\,u$ for every $u \in (0, \tau^\star)$, and the second identity in (42) gives, for any $\tau \in (0, \tau^\star)$,

$$x(\tau) = x_0 - \int_0^\tau e^{-u}\eta(u)\,\mathrm{d}u > x_0 - x_0 \int_0^\tau u\,e^{-u}\,\mathrm{d}u > x_0 - x_0 \int_0^\infty u\,e^{-u}\,\mathrm{d}u = 0,$$

using (43) in the last step. Thus x is bounded away from zero on every compact subinterval and can never reach it, so $\tau^\star = \infty$: the residual stays positive and strictly decreasing for all τ.

Being positive and strictly decreasing, $x(\tau)$ has a limit $x_\infty \in [0, x_0)$. Since $x(r) < x_0$ for every $r > 0$, the strict inequality $\eta(u) < x_0 u$ holds for all $u > 0$. Multiplying by the positive weight e^{-u} and integrating over $(0, \infty)$,

$$\int_0^\infty e^{-u}\eta(u)\,\mathrm{d}u < x_0 \int_0^\infty u\,e^{-u}\,\mathrm{d}u = x_0.$$

Letting $\tau \to \infty$ in the second identity of (42) therefore yields

$$x_\infty = x_0 - \int_0^\infty e^{-u}\eta(u)\,\mathrm{d}u > x_0 - x_0 = 0.$$

For the upper bound, $x(u) \geq x_\infty$ gives $\eta(u) \geq x_\infty u$, hence $\int_0^\infty e^{-u}\eta(u)\,\mathrm{d}u \geq x_\infty$ by (43), so $x_\infty = x_0 - \int_0^\infty e^{-u}\eta \leq x_0 - x_\infty$, i.e. $x_\infty \leq \frac{1}{2}x_0$. The case that $x_0 < 0$ is handled by symmetry and is omitted. □

D.2 $p > 2$

To provide supporting evidence that we can see constraint satisfaction for $p > 2$, we revisit the scalar example from the previous section ($n = m = 1$, $g(x) = x$, and $v_\theta = 0$). Using the same change of coordinates $\tau = -\log(1 - t)$ and $\eta = (1 - t)^{p-1}\lambda$, we are left with the coupled system

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = -e^{(p-2)\tau}\,\eta, \qquad \frac{\mathrm{d}\eta}{\mathrm{d}\tau} = x - (p-1)\eta, \qquad x(0) = x_0,\ \eta(0) = 0. \tag{44}$$

Differentiating the first equation and substituting the second turns (44) into a single scalar equation for the residual,

$$\frac{\mathrm{d}^2x}{\mathrm{d}\tau^2} + \frac{\mathrm{d}x}{\mathrm{d}\tau} + e^{(p-2)\tau}\,x = 0, \qquad x(0) = x_0, \quad \frac{\mathrm{d}x}{\mathrm{d}\tau}(0) = 0, \tag{45}$$

which is a damped oscillator with unit damping and a stiffness $e^{(p-2)\tau}$ that grows unbounded when $p > 2$. Constraint satisfaction as $t \to 1^-$ corresponds to $x(\tau) \to 0$. Note that when $p = 2$, this is a standard damped oscillator whose solution trajectories tend to the origin exponentially quickly, agreeing with the results of Theorem 1.

First, we will choose the substitution $x(\tau) = e^{-\tau/2}y(\tau)$. Computing the derivatives of $x(\tau)$ and substituting them into the equation (45), we get the differential equation for y:

$$\frac{\mathrm{d}^2y}{\mathrm{d}\tau^2} + \left(e^{(p-2)\tau} - \frac{1}{4}\right)y = 0. \tag{46}$$

Let $Q(\tau) = e^{(p-2)\tau} - \frac{1}{4}$. Since $p > 2$, we see that Q is strictly positive and non-decreasing, i.e., $Q(\tau) > 0$ and $Q'(\tau) \geq 0$. Going forward, we will use $'$ to denote a derivative with respect to τ.

Define the energy function

$$V(\tau) = \frac{1}{Q(\tau)}(y')^2 + y^2. \tag{47}$$

Since $Q(\tau) > 0$, $V(\tau)$ is always nonnegative and satisfies that $y^2 \leq V(\tau)$, meaning that if we can show that V is bounded, we will conclude that y is as well. Taking a τ-derivative of V yields

$$\begin{aligned}\frac{\mathrm{d}V}{\mathrm{d}\tau} &= \frac{2y'y''Q(\tau) - (y')^2Q'(\tau)}{Q(\tau)^2} + 2yy' = \frac{-2yy'Q(\tau)^2 - (y')^2Q'(\tau)}{Q(\tau)^2} + 2yy' \\ &= -\frac{Q'(\tau)}{Q(\tau)^2}(y')^2 \leq 0.\end{aligned}$$

Therefore, for all $\tau \in [0, \infty)$, we have that $V(\tau) \leq V(0)$. Since $y^2(\tau) \leq V(\tau)$, we conclude that y is bounded and cannot grow to infinity. Mapping back to x, $|x(\tau)| = e^{-\tau/2}|y(\tau)| \to 0$ as $\tau \to \infty$. Therefore, we get constraint satisfaction in this case for any initial condition and any $p > 2$.