CONICFLOW: CONSTRAINED INFERENCE-TIME OPTIMAL CONTROL FOR FLOW MATCHING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose ConicFlow, which frames constrained sampling from a flow matching model at inference time as an optimal control problem that balances deviation from the pretrained trajectory against reward, subject to hard terminal constraints. We discretize this problem and adapt receding-horizon control, optimizing only a short window of the trajectory at each step for computational tractability. We address the resulting sequence of receding-horizon subproblems with a receding-horizon primal–dual algorithm. Its dual multipliers capture the difficulty of satisfying each constraint and are refined throughout sampling, steering generation toward feasible final samples. We further propose a terminal-estimate variant that avoids accumulating local truncation errors by evaluating the reward and constraints at the predicted terminal point rather than at the end of the current horizon. Under standard regularity assumptions, we establish finite-horizon guarantees on sub-optimality. Under convex-affine structure, we derive a sharper bound that scales linearly with omitted horizon length, and we show that the terminal-estimate variant is provably tighter when the flow dynamics change slowly. Experiments in both low- and high-dimensional settings with a recent large-scale flow matching model show that ConicFlow achieves high reward and strong constraint satisfaction while deviating less from the pretrained flow than recent baselines.

1 INTRODUCTION

Diffusion models and flow matching models generate high-quality samples across a variety of modalities (Ho et al., 2020; Lipman et al., 2022; Rombach et al., 2022). However, sample quality alone is often insufficient: many applications require samples that both achieve high reward and satisfy hard constraints. Rewards can encode user preferences, such as image aesthetics evaluated by models trained on human feedback (Wu et al., 2023), consistency with observations in inverse problems (Ben-Hamu et al., 2024), or safety-related criteria (Schramowski et al., 2023). Optimizing these rewards alone, however, does not guarantee satisfaction of hard constraints. The challenge is to achieve high reward and terminal feasibility while limiting deviations from the pretrained dynamics.

In both theory and practice, ensuring that samples generated by these models satisfy hard constraints is underexplored. Existing approaches fall broadly into two categories. One approach is to adapt model parameters through fine-tuning (Black et al., 2024; Domingo i Enrich et al., 2025), but it can be prohibitively expensive. Furthermore, such an approach would require finetuning again for each new reward/constraint. Guidance, inference-time steering, and projection based methods for constraint satisfaction (Ben-Hamu et al., 2024; Christopher et al., 2026), avoid the computational overhead of fine-tuning, usually at the cost of additional overhead during sampling. Our approach is an inference-time approach. We provide extended discussion on related approaches in Appendix A.

A pre-trained flow matching model describes the dynamics of the sample evolving across time. We need to deviate from these dynamics in order to satisfy constraints and high reward at the endpoint, meanwhile ideally we would want these deviations to be small, because we still want to end up reasonably close to the data manifold. Given this description, the problem naturally lends itself to an optimal control formulation, where the deviation from the flow dynamics can be viewed as the control input that we are optimizing over. This leads to the continuous-time constrained inference problem (P-C) where r: $\mathbb{R}^d \to \mathbb{R}$ encodes a soft preference over the generated sample, while h:

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.

$\mathbb{R}^d \to \mathbb{R}^m$ represents m hard terminal requirements.

$$\begin{aligned} \min_{u(t)} \quad & J_C(u) \;:=\; \int_0^1 \frac{\|u(t)\|^2}{2\lambda}\, dt - r(x_1^u) \\ \text{s.t.} \quad & \dot{x}_t^u \;=\; b_t(x_t^u) + u(t), \quad \text{given } x_0, \\ & h(x_1^u) \;\le\; 0. \end{aligned} \tag{P-C}$$

Full-horizon constrained control can be computationally demanding, particularly for high-dimensional flow models. We therefore adopt receding-horizon control (RHC), which replaces the full problem by a sequence of shorter optimizations and replans after each intervention. (García et al., 1989; Mayne et al., 2000; Mattingley et al., 2011; Bemporad et al., 2000). This improves tractability at the cost of horizon-truncation error, motivating our formulations (RH-H) and (RH-T) in Section 3.

Specifically, in RHC, rather than optimizing the full trajectory at once, at each step of the dynamics (in the case of flow matching, at each sampling step) we optimize over a short window and then apply the first optimized control action before shifting the window forward by one step and repeating this procedure until we reach terminal time. Note that in (P-C), both the reward and the constraints are evaluated only at the terminal time. Classical receding-horizon control introduced in (RH-H) instead evaluates them at the end of each truncated window, which creates a mismatch with the original objective. We therefore propose (RH-T), which uses the pretrained flow map (Boffi et al., 2025) to predict the terminal state reached from the end of each window and evaluates the reward and constraints at this prediction. To the best of our knowledge, this is the first work to use flow maps within receding-horizon control, allowing each subproblem to account for the full-horizon objective without extending the control horizon. Our contributions are fourfold.

(i) **Constrained-control formulation and theory.** We formulate reward-guided generation with hard terminal constraints as an optimal control problem that penalizes deviations from pretrained dynamics, and bound the errors introduced by discretization and horizon truncation.

(ii) **Terminal prediction through flow maps.** We propose a variant of receding-horizon control (RH-T) that uses the pretrained flow map to predict the uncontrolled continuation beyond each optimization window. This enables evaluation of rewards and constraints at the final sampling time without extending the optimized control horizon.

(iii) **Primal-dual algorithm and convergence guarantees.** We develop a receding-horizon primal-dual algorithm and, under affine dynamics, establish sublinear convergence rates. We further quantify how local optimization errors propagate and establish convergence to the exact receding-horizon trajectory under uniform bounds.

(iv) **Experiments.** On synthetic data and image editing, ConicFlow improves the reward-control trade-off and achieves near-100% empirical feasibility with up to a 20-fold reduction in control energy relative to the baseline.

Preliminaries on flow matching and flow maps. A flow matching model generates samples by evolving an initial state $x_0 \sim p_0$ according to $\dot{x}_t = b_t(x_t)$ for $t \in [0,1]$, where b: $[0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is a learned velocity field. Training aims to transport the source distribution p_0 to the target distribution p_{data} at $t = 1$, and can be formulated using stochastic interpolants (Albergo et al., 2025). Assuming the ODE has unique solutions over the time interval, its *flow map* $X_{s,t} : \mathbb{R}^d \to \mathbb{R}^d$ sends a state at time s to the corresponding state at time t: $X_{s,t}(x_s) = x_t$, with $X_{s,s}(x) = x$ for $0 \le s \le t \le 1$. Learned approximations of these solution operators enable direct prediction across time intervals, accelerating sampling without explicitly integrating every intermediate step (Boffi et al., 2025).

2 Problem Formulation

Problem statement. We consider the problem of steering a pretrained flow model toward samples that achieve a desired terminal reward while simultaneously satisfying hard output constraints. Rather than modifying or retraining the underlying flow model, we introduce an inference-time control u_t that perturbs the nominal vector field. This leads to the continuous-time constrained inference problem (P-C). The reward r and the constraint h play distinct roles: The constraint specifies strict

properties that the generated output must satisfy, while the reward ranks feasible terminal samples. The quadratic energy term penalizes deviation from the pretrained generative dynamics. In particular, smaller λ places a larger penalty on intervention and keeps the controlled trajectory closer to the nominal flow, while larger λ permits stronger inference-time steering.

Discrete-time constrained inference. In practice, flow-based inference is carried out using finitely many vector-field evaluations. We introduce a uniform forward-Euler discretization of the flow dynamics in (P-C) over $t_0 < t_1 < \cdots < t_N$: $x_{k+1} = x_k + \Delta t\big(b_{t_k}(x_k) + u_k\big)$, where $t_0 = 0$, $t_N = 1$, and $\Delta t = 1/N$. By applying Euler discretization, we approximate (P-C) as a discrete-time constrained inference problem,

$$\begin{aligned} \min_{u_{0:N-1}} \quad & J_D(u) \;:=\; \sum_{k=0}^{N-1} \Delta t \frac{\|u_k\|^2}{2\lambda} - r(x_N^u) \\ \text{s.t.} \quad & x_{k+1} \;=\; x_k + \Delta t\big(b_{t_k}(x_k) + u_k\big), \quad \text{given } x_0, \\ & h(x_N^u) \;\le\; 0. \end{aligned} \tag{P-D}$$

The constrained inference problem (P-D) is finite-dimensional. To capture the approximation error, we prove that the optimal value of the discrete-time problem (P-D) approximates that of the continuous-time problem (P-C) with error $O(1/N)$ in Lemma 1. Let $u_C^\star$ be an optimal control for (P-C), and $u_D^\star(N)$ be an optimal control for the N-step discretized problem (P-D).

Assumption 1 (Regularity, feasibility, and repairability)**.** (i) *On the relevant energy-bounded reachable set, $b_t(x)$ is L_b-Lipschitz in state and L_b^t-Lipschitz in time, while r and h are L_r- and L_h-Lipschitz, respectively;* (ii) *there exist $u^0, u_C^\star \in L^2$ such that $h(x_1^{u^0}) \le 0$, $J_C(u^0) < \infty$, and $u_C^\star \in \operatorname{argmin}_{u:\, h(x_1^u)\le 0} J_C(u)$;* (iii) *there exist $\kappa, \rho_0 > 0$, independent of N, such that any relevant continuous- or discrete-time control with terminal violation at most ρ_0 admits a feasible perturbation whose control norm is at most κ times the violation.*

We state Assumption 1 in a compact form with its precise statement deferred to Appendix B.1.

Lemma 1 (Discretization error)**.** *Let Assumption 1 hold. Then, there exist constants $N_1 \in \mathbb{N}$, and $C_{\text{disc}} > 0$ that is independent of N, such that for all $N \ge N_1$,*

$$\big|J_D\big(u_D^\star(N)\big) - J_C(u_C^\star)\big| \le \frac{C_{\text{disc}}}{N}. \tag{1}$$

Lemma 1's proof is provided in Appendix B.1. In Lemma 1, we show that the objective gap between the discrete- and continuous-time problems, (P-D) and (P-C), decays to zero at a rate $O(1/N)$.

Path-space constrained inference. Although (P-D) is finite-dimensional, solving it directly requires a full-horizon constrained nonlinear optimization over all N control inputs, with repeated evaluations of the pretrained flow map. For high-dimensional flow maps, this can be computationally prohibitive at inference time. We therefore rewrite Problem (P-D) in terms of the generated trajectory rather than the control sequence. The control input u_k in (P-D) is directly expressed by $u_k = (x_{k+1} - x_k)/\Delta t - b_{t_k}(x_k)$. Thus, we can remove the explicit flow dynamics in (P-D) and introduce a discrete-time path-space problem as follows.

$$\begin{aligned} \min_{x_{1:N}} \quad & J_D(u) \;:=\; \sum_{k=0}^{N-1} \frac{1}{2\lambda\Delta t} \left\|x_{k+1} - x_k - \Delta t\, b_{t_k}(x_k)\right\|^2 - r(x_N) \\ \text{s.t.} \quad & h(x_N) \le 0, \quad \text{given } x_0. \end{aligned} \tag{P-DP}$$

Like Problem (P-D), (P-DP) remains finite-dimensional, but it does not include constraints from the flow dynamics. Each stage cost of (P-DP) measures the discrepancy between the optimized displacement $x_{k+1} - x_k$ and the pretrained flow drift $\Delta t\, b_{t_k}(x_k)$. Thus, path-space optimization can be interpreted as deforming the nominal generative trajectory while penalizing the magnitude of that deformation. However, solving (P-DP) still requires optimizing the entire N-step generation trajectory. For flow matching models where the vector field is typically given by a deep neural network, repeatedly optimizing such a full path requires a large number of evaluations of the high-dimensional vector field. This becomes more costly as the number of sampling steps grows. To address this challenge, we will establish a receding-horizon primal-dual approach that allows us to retain the reward and constraint structure of the full inference problem (P-C) while optimizing a short portion of the future trajectory at each time step during inference.

3 RECEDING-HORIZON PRIMAL-DUAL

To make (P-DP) more tractable, we first propose to replace the whole trajectory with a local subset of it. At generation step j, starting from current state $x_j = \bar{x}_j$, we select next H_j states and optimize over them: $(x_{j+1}, \ldots, x_{j+H_j})$, where $H_j \ll N$. We apply the first induced control, and then repeat the optimization from the resulting state. This produces a receding-horizon procedure analogous to model predictive control (García et al., 1989; Mayne et al., 2000; Alonso et al., 2022), but expressed directly in the path space of the pretrained flow.

Where should the terminal objective and constraint be evaluated? Denote a local path $x_{j+1:j+H_j} = (x_{j+1}, \ldots, x_{j+H_j})$ and abbreviate it as $\mathbf{x}_j$. Restricting the optimization horizon introduces an important design choice. A direct truncation evaluates the reward and constraint at the end of the local optimization window:

$$\begin{aligned} \min_{\mathbf{x}_j} \quad & \sum_{\ell=j}^{j+H_j-1} \frac{1}{2\lambda\Delta t} \left\| x_{\ell+1} - x_\ell - \Delta t\, b_{t_\ell}(x_\ell) \right\|^2 - r(x_{j+H_j}) \\ \text{s.t.} \quad & h(x_{j+H_j}) \le 0. \end{aligned} \tag{RH-H}$$

Formulation (RH-H) is computationally attractive, since it involves shorter paths to optimize. Meanwhile, (RH-H) relaxes the original inference problem (P-C): the terminal reward and feasibility requirement are now imposed at time t_{j+H_j} rather than at the final generation time. Once the local cost has been optimized at $\mathbf{x}_j^\star$, the first residual intervention is recovered as $u_j^\star = (x_{j+1}^\star - \bar{x}_j)/\Delta t - b_{t_j}(\bar{x}_j)$. Only this first intervention is executed before shifting the window forward by one step and replanning (Mayne et al., 2000; Alonso et al., 2022).

A distinctive feature of our setting is that the pretrained flow itself provides a natural prediction of how a locally optimized state would evolve up to the true terminal time. Rather than evaluating reward and feasibility prematurely at x_{j+H_j}, we can propagate this state under the uncontrolled flow and evaluate the original terminal objective at the resulting prediction. This brings each truncated subproblem closer to the full-horizon problem (P-DP) without extending the control horizon.

To make this precise, we introduce the discrete flow map induced by the pretrained flow, $X_{i,j} : \mathbb{R}^d \to \mathbb{R}^d$, defined recursively by $X_{i,i}(x) = x$ and $X_{i,k+1}(x) = X_{i,k}(x) + \Delta t\, b_{t_k}\big(X_{i,k}(x)\big)$ for $k \ge i$, so that $X_{i,j}(x)$ is the state obtained by propagating x from t_i to t_j under the pretrained flow. Given the endpoint x_{j+H_j} of the local window, the terminal prediction is then $\widehat{x}_N = X_{j+H_j,N}(x_{j+H_j})$. Evaluating the terminal objective at $\widehat{x}_N$ in place of x_{j+H_j} yields our terminal-rollout formulation, RH-T, described below. This gives the terminal-rollout formulation

$$\begin{aligned} \min_{\mathbf{x}_j} \quad & \sum_{\ell=j}^{j+H_j-1} \frac{1}{2\lambda\Delta t} \left\| x_{\ell+1} - x_\ell - \Delta t\, b_{t_\ell}(x_\ell) \right\|^2 - r\big(\widehat{x}_N\big) \\ \text{s.t.} \quad & h\big(\widehat{x}_N\big) \le 0. \end{aligned} \tag{RH-T}$$

Compared with (RH-H), (RH-T) truncates only the control horizon while retaining the terminal objective and constraint. Our analysis in Section 4.1 identifies when this yields a tighter guarantee; the comparison is detailed in Remark 1.

Receding-Horizon Primal-Dual Approach. Directly solving a new hard-constrained nonlinear program at every generation step itself is expensive. We therefore introduce a primal–dual approach in which terminal constraints are represented by dual variables $\mu \in \mathbb{R}_+^m$. To unify our analysis, we represent an approximate terminal state by $\Phi_j(x)$, which is x or $X_{j+H_j,N}(x)$ for (RH-H) and (RH-T), respectively. Denote $\mathcal{E}_j(\mathbf{x}_j) := \sum_{\ell=j}^{j+H_j-1} \frac{1}{2\lambda\Delta t} \left\| x_{\ell+1} - x_\ell - \Delta t\, b_{t_\ell}(x_\ell) \right\|^2$ and $g_j(\mathbf{x}_j) := h\big(\Phi_j(x_{j+H_j})\big)$. For each j, we have a frozen local problem,

$$p_j^\star(\bar{x}_j) \;:=\; \min_{\mathbf{x}_j} \; \big\{ \varphi_j(\mathbf{x}_j; \bar{x}_j) := \mathcal{E}_j(x_{j:j+H_j}) - r\big(\Phi_j(x_{j+H_j})\big) \big\} \quad \text{s.t.} \quad g_j(\mathbf{x}_j) \le 0. \tag{RH$_j$}$$

The Lagrangian for (RH$_j$) is given by $\mathcal{L}_j(\mathbf{x}_j, \mu; \bar{x}_j) = \varphi_j(\mathbf{x}_j; \bar{x}_j) + \mu^\top g_j(\mathbf{x}_j)$. Define the dual function by $d_j(\mu; \bar{x}_j) := \min_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j, \mu; \bar{x}_j)$. The associated dual problem is $\max_{\mu \ge 0} d_j(\mu; \bar{x}_j)$, whose

Algorithm 1 Receding-Horizon Primal-Dual Algorithm

Require: Pretrained flow b_t, reward r, constraints h, horizon H, local optimization steps $K \geq 1$, dual step size η_μ, a choice of (RH-H) and (RH-T) through the terminal map Φ_j.

1: Initialize x_0 and $\mu_0^{(0)} \geq 0$
2: **for** $j = 0, \ldots, N-1$ **do**
3: Set $H_j \leftarrow \min\{H, N-j\}$, $\bar{x}_j \leftarrow x_j$ and initial $\mu_j^{(0)} \geq 0$.
4: **for** $s = 0, 1, \ldots, K-1$ **do**
5: Compute the primal response: $\mathbf{x}_j^{(s)} \leftarrow \operatorname{argmin}_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j, \mu_j^{(s)}; \bar{x}_j)$.
6: Perform the projected dual update: $\mu_j^{(s+1)} \leftarrow \left[\mu_j^{(s)} + \eta_\mu g_j(\mathbf{x}_j^{(s)})\right]_+$.
7: **end for**
8: Compute the primal response at $\mu_j^{(K)}$: $\mathbf{x}_j^{(K)} \leftarrow \operatorname{argmin}_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j, \mu_j^{(K)}; \bar{x}_j)$.
9: Execute the first state of $\mathbf{x}_j^{(K)}$: $x_{j+1} \leftarrow x_{j+1}^{(K)}$, and recover the residual control: $u_j \leftarrow (x_{j+1}^{(K)} - \bar{x}_j)/\Delta t - b_{t_j}(\bar{x}_j)$.
10: If $j < N-1$, warm start the next physical window with $\mu_{j+1}^{(0)} = \mu_j^{(K)}$.
11: **end for**
return The executed path $(x_0, \ldots, x_N)$ and residual controls $(u_0, \ldots, u_{N-1})$.

solution set is denoted by $\mathcal{M}_j^\star(\bar{x}_j)$. At each receding-horizon window, we hold the current state $\bar{x}_j$ fixed and alternate primal minimization with projected dual ascent: $\mathbf{x}_j^{(s)} \in \arg\min_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j, \mu_j^{(s)}; \bar{x}_j)$ and $\mu_j^{(s+1)} = \left[\mu_j^{(s)} + \eta_\mu g_j(\mathbf{x}_j^{(s)})\right]_+$, where $\eta_\mu > 0$ is the dual step size. The primal step optimizes the planned path for the current multipliers, while the dual step increases the weight of violated constraints and decreases that of strictly satisfied constraints. After K dual updates, we recompute the primal response at $\mu_j^{(K)}$ and execute only its first transition. We then advance the window and warm-start its multiplier using $\mu_{j+1}^{(0)} = \mu_j^{(K)}$. This transfers the learned constraint weights between successive windows without assuming that their optimal multipliers coincide. We summarize the receding-horizon inference procedure in Algorithm 1 and establish its theoretical guarantees for solving the constrained inference problem (P-C) in Section 4.

4 Theoretical Analysis

To bridge the original continuous-time problem and the implementable Algorithm 1, we proceed in two steps. First, we bound the approximation error of (RH-H) and (RH-T) relative to the full discretized problem (P-D). We then bound the error introduced by finitely solving the local primal–dual problem (RH_j). Combined with Lemma 1, these results connect Algorithm 1 to the original continuous formulation (P-C).

4.1 Bounding the Suboptimality

We first quantify the error introduced by restricting the control horizon. Specifically, we compare the idealized constrained trajectories of (RH-H) and (RH-T) with the optimal solution $\boldsymbol{x}^\star$ of the full discretized problem (P-D). Both approximations optimize only H_j future steps, but differ in where the terminal reward and constraint are evaluated.

Throughout this section, Assumption 1 remains in force. For the receding-horizon comparison, we additionally impose the following structure. Unlike the continuous-to-discrete analysis, terminal feasibility here is handled through a uniform Slater condition rather than local repairability.

Assumption 2 (Affine flow and bounded amplification)**.** *The discretized flow is affine, $x_{k+1} = x_k + \Delta t(A_k x_k + c_k + u_k)$. On the relevant reachable set, $\|A_k x + c_k\| \leq M$, and further let $\kappa_A := \sup_{0 \leq k \leq \ell \leq N} \left\|\prod_{s=k}^{\ell-1}(I + \Delta t A_s)\right\| < \infty$, with M and κ_A independent of the horizon.*

Assumption 3 (Convexity and uniform feasibility)**.** *The reward r is concave and L_r-Lipschitz, and h is componentwise convex and L_h-Lipschitz. Either subproblem* (RH-H) *or* (RH-T) *admits a strictly feasible trajectory with uniform margin $\gamma > 0$ and objective at most $\overline{J}_D$.*

Let $\underline{J}_D$ be a uniform lower bound on the objectives of (RH-H) or (RH-T) and define $\chi_h := L_h(\overline{J}_D - \underline{J}_D)/\gamma$ and $G_h := L_r + \chi_h$. Here, G_h represents the effective terminal sensitivity, with L_r capturing reward variation and χ_h the additional sensitivity required to maintain feasibility.

We first consider (RH-H), which evaluates reward and feasibility at the end of the truncated horizon. Consequently, its approximation error depends on how far the nominal flow can move over the omitted portion of the trajectory, captured by the drift bound M.

Lemma 2 (Constrained RH-H bound)**.** *Let Assumptions 1–3 hold, and $M < \infty$, $G_h < \infty$, and $\gamma > 0$. Then, the optimal trajectory $\boldsymbol{x}^{H,\star}$ of* (RH-H) *satisfies*

$$0 \;\le\; J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^{\star}) \;\le\; C_H^{\mathrm{C}} \frac{N-H}{N}, \qquad \text{where } C_H^{\mathrm{C}} := 2\max\left\{ G_h M + \frac{1}{2}\lambda G_h^2, \frac{M^2}{2\lambda} \right\}.$$

Thus, (RH-H) incurs a horizon error linear in the omitted fraction $(N-H)/N$. Its constant depends on the absolute motion of the pretrained flow through M, reflecting that reward and feasibility are evaluated before the true terminal time.

(RH-T) instead propagates the local endpoint to the original terminal time N using the pretrained flow map. This removes dependence on the absolute nominal motion after the control horizon; the relevant quantity is instead how strongly the remaining flow amplifies perturbations, measured by κ_A.

Lemma 3 (Constrained RH-T bound)**.** *Let Assumptions 1–3 hold, and assume $\kappa_A < \infty$, $G_h < \infty$, $\gamma > 0$. Then, the optimal trajectory $\widehat{\boldsymbol{x}}^{H,\star}$ of* (RH-T) *satisfies*

$$0 \;\le\; J(\widehat{\boldsymbol{x}}^{H,\star}) - J(\boldsymbol{x}^{\star}) \;\le\; C_T^{\mathrm{C}} \frac{N-H}{N}, \qquad \text{where } C_T^{\mathrm{C}} := \frac{1}{2}\kappa_A^2 G_h^2 \lambda.$$

Both formulations thus achieve an $O((N-H)/N)$ approximation to the full discretized problem, but their constants capture different properties of the pretrained dynamics: (RH-H) depends on nominal flow motion, whereas (RH-T) depends on perturbation amplification. The resulting comparison, including conditions under which (RH-T) yields the tighter guarantee, is detailed in Remark 1.

Remark 1 (When is terminal rollout beneficial?)**.** *For our constrained problem, $\kappa_A \le \sqrt{2}$ implies $C_T^{\mathrm{C}} \le C_H^{\mathrm{C}}$. Thus, when the remaining pretrained flow does not strongly amplify intermediate-state perturbations, terminal rollout gives the tighter guarantee. The analogous unconstrained comparison is given in Appendix B.2. Intuitively,* (RH-T) *is most useful when the state continues to evolve after the control horizon, but this evolution remains predictable.*

4.2 Convergence of Dynamic Primal–Dual

In Lemmas 2–3, we assume that each subproblem of (RH$_j$) is solved, exactly. We next quantify the error introduced by a finite number of primal–dual updates in Algorithm 1. For the jth subproblem (RH$_j$), we denote by B_j a lower block diagonal matrix with off-diagonals $-(I_d + \Delta t A_{j+1})$, S_j a row vector with first block $I_d + \Delta t A_j$, and Q_j the linear part of the terminal map Φ_j ($Q_j = I$ for (RH-H)). Using the strong convexity and dual smoothness, we first show that the inner-loop primal-dual iterates (Lines 4–7 of Algorithm 1) converge to the optimal, sublinearly.

Lemma 4 (Local primal–dual convergence)**.** *Let Assumptions 2–3 hold for the jth frozen subproblem problem* (RH$_j$)*. Then, if $0 < \eta_\mu < 2/L_{d,j}$, then the inner-loop primal-dual iterates (Lines 4–7 of Algorithm 1) converge their optima, i.e., $\mu_j^{(s)} \to \bar{\mu}_j \in \mathcal{M}_j^\star$ and $\mathbf{x}_j^{(s)} \to \mathbf{x}_j^\star$. Moreover, if $0 < \eta_\mu \le 1/L_{d,j}$, then the gaps between the iterates $(\mathbf{x}_j^{(s)}, \mu_j^{(s)})$ and the optima are bounded by*

$$p_j^\star - d_j(\mu_j^{(s)}) \;\le\; \frac{R_j^2}{2\eta_\mu s}, \qquad \|\mathbf{x}_j^{(s)} - \mathbf{x}_j^\star\| \;\le\; \frac{R_j}{\sqrt{\sigma_j \eta_\mu s}}, \qquad \|[g_j(\mathbf{x}_j^{(s)})]_+\| \;\le\; \frac{L_{g,j} R_j}{\sqrt{\sigma_j \eta_\mu s}}$$

where $R_j := \mathrm{dist}(\mu_j^{(0)}, \mathcal{M}_j^\star)$, $\sigma_j := \sigma_{\min}^2(B_j)/(\lambda \Delta t)$ is the strong convexity modulus, $L_{g,j} := L_h \|Q_j\|$ and $L_{d,j} := L_{g,j}^2/\sigma_j$ are the Lipschitz constants of the constraint and dual functions.

We defer the proof of Lemma 4 to Appendix B.5. In Lemma 4, we show that the primal-dual gap $p_j^\star - d_j(\mu_j^{(s)})$, and the primal gap $\|\mathbf{x}_j^{(s)} - \mathbf{x}_j^\star\|$ and constraint violation $\|[g_j(\mathbf{x}_j^{(s)})]_+\|$ decay to zero, sublinearly. This result shows that the primal-dual update (Lines 4–7 of Algorithm 1) solves the subproblem (RH_j) up to an error $\mathcal{O}(1/\sqrt{s})$. We next quantify how these errors propagate through the receding-horizon trajectory, and establish the global convergence of Algorithm 1 in Theorem 1; see Appendix B.6 for proofs. Denote the first state of the solution $\mathbf{x}_j^\star(\xi)$ to (RH_j) initialized at ξ as $\pi_j(\xi)$. We let the exact trajectory generated by π_j be $x_{j+1}^{\mathrm{RH}} = \pi_j(x_j^{\mathrm{RH}})$ with $x_0^{\mathrm{RH}} = x_0$.

Theorem 1 (Global convergence)**.** *Let Assumptions 2–3 hold for the jth frozen subproblem problem (RH_j). Then, if $0 < \eta_\mu \le 1/\bar{L}_d$ and each inner-loop primal-dual update of Algorithm 1 take K iterations, then the generated trajectory x_j for $j = 1, \dots, N$ satisfies*

$$\|x_j - x_j^{\mathrm{RH}}\| \le \frac{\bar{R}\,\Gamma_j(\bar{\kappa})}{\sqrt{\underline{\sigma}\,\eta_\mu K}}, \qquad \|[h(x_N)]_+\| \le \frac{L_h \bar{R}}{\sqrt{\underline{\sigma}\,\eta_\mu K}}$$

where $\bar{L}_d := \max_j L_{d,j}$, $\bar{R} := \max_j \mathrm{dist}(\mu_j^{(0)}, \mathcal{M}_j^\star(x_j))$, $\underline{\sigma} := \min_j \sigma_j$, $\kappa_j := \frac{\|B_j^\top S_j\|}{\lambda \Delta t\, \sigma_j}$ $\bar{\kappa} := \max_j \kappa_j$, *and* $\Gamma_j(\bar{\kappa}) := \sum_{q=0}^{j-1} \bar{\kappa}^q$.

In Theorem 1, we prove that the trajectories generated by Algorithm 1 converge to the exact trajectory induced by the policy π_j, in terms of the primal gap $\|x_j - x_j^{\mathrm{RH}}\|$ and the constraint violation $\|[h(x_N)]_+\|$. This implies that a finite iterations of primal-dual optimization contribute an $O(1/\sqrt{K})$ error, modulated by the sensitivity $\Gamma_j(\bar{\kappa})$ of the exact policy.

Corollary 1 (Error decomposition)**.** *Let the assumptions of Theorem 1 hold. Suppose J is L_J-Lipschitz on the relevant set. Then, the generated trajectory $\boldsymbol{x}^{(K)}$ by Algorithm 1 satisfies*

$$J(\boldsymbol{x}^{(K)}) - J_C(u_C^\star) \;\le\; \frac{C_{\mathrm{disc}}}{N} + C_{\mathrm{hor}} \frac{N-H}{N} + \frac{L_J \sqrt{N}\, \bar{R}\, \Gamma_N(\bar{\kappa})}{\sqrt{\underline{\sigma}\,\eta_\mu K}},$$

where $C_{\mathrm{hor}} := C_H^{\mathrm{C}}$ *for* (RH-H) *and* $C_{\mathrm{hor}} := C_T^{\mathrm{C}}$ *for* (RH-T).

The proof of Corollary 1 is straightforward by adding and subtracting $J_D\big(u_D^\star(N)\big)$ and $J(\boldsymbol{x}^{\mathrm{RH}})$, and applying Lemmas 1–3 and Theorem 1. The bound separates the discretization error $O(1/N)$ and horizon bias $O((N-H)/N)$ from the $O(1/\sqrt{K})$ finite-solver error. Taking $H = N$ eliminates the horizon bias $O((N-H)/N)$, and letting $K \to \infty$ for fixed N eliminates the finite-solver error, leaving only the discretization error bounded by $O(1/N)$, which vanishes as $N \to \infty$, yielding global convergence to the continuous-time optimum.

5 Experiments

We evaluate ConicFlow in two regimes: *unconstrained* reward-guided generation, where the objective trades terminal reward against control energy, and *constrained* generation, where the sample must additionally satisfy hard terminal constraints. We compare against the state-of-the-art approach in each regime: (i) FMRG (Huang et al., 2026) and (ii) HardFlow (Li et al., 2026). FMRG is a training-free, single-trajectory method that uses soft reward guidance, and it is theoretically near-optimal only when small deviations from the pretrained flow suffice. HardFlow enforces constraints at terminal time, but with a different methodology compared to our approach, and it has been shown to outperform projection-based alternatives. See Appendix A for more discussion on related approaches.

Unconstrained low-dimensional setting. We first use a flow map model trained on 2D samples from the synthetic two moons dataset and consider two rewards, both taking values in $[0, 1]$. The first reward increases closer to the upper moon. As shown in Figure 1 (left), all methods saturate the reward at small λ (i.e., large control-energy penalties). On average, small deviations from the pretrained flow suffice to land trajectories on the upper moon. Both ConicFlow variants: (RH-H) and (RH-T) achieve the best trade-off between terminal reward and control energy, closely matched by FMRG, which is theoretically near-optimal in the small-λ regime (Huang et al., 2026).

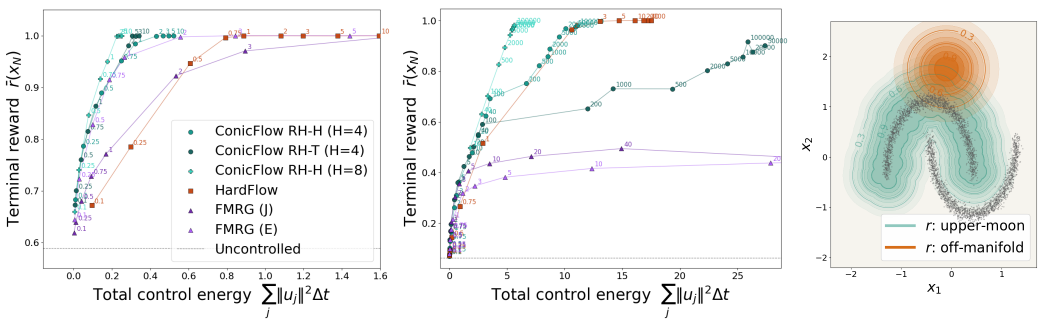


Figure 1: Trade-off between terminal reward and total control cost, traced by varying the control-cost weight λ (upper left is better). λ value displayed on each point. **Left:** easy reward (upper moon). **Middle:** difficult reward (off-manifold). **Right:** uncontrolled sample distribution and both rewards.

To study settings that need large deviations (thus large λ), the second reward peaks off the pretrained data manifold (middle in Figure 1). FMRG fails outside the small-λ regime as its control energy increases without noticeable reward gains. ConicFlow (RH-H) with $H = 4$ matches HardFlow (and improves on it for small deviations), while increasing the horizon to $H = 8$ improves the trade-off across the board, outperforming all baselines and saturating the reward with the least control energy. Neither (RH-H) nor (RH-T) consistently outperforms the other.

In short, FMRG does well for small λ and HardFlow for larger λ. *ConicFlow with a short horizon combines the strengths of FMRG and HardFlow across all λ, and improves further as H increases.*

Unconstrained image setting. In high-dimensional, prompt-conditioned image generation, we use FLUX.1-dev with the flow-map LoRA released by (Huang et al., 2026), distilled from the base model as our flow matching model (Boffi et al., 2025). We consider a well-established task: reward-guided image editing (Li et al., 2026; Kim et al., 2022; Patashnik et al., 2021). A base prompt c generates the sample, while the reward $r(x; c')$ is evaluated against a *different* edit prompt c'. The control must introduce the attribute named by c' while deviating as little as possible from the original image. We use three commonly used rewards CLIP (Radford et al., 2021), ImageReward (Xu et al., 2023), and PickScore (Kirstain et al., 2023). The RH-T variant of ConicFlow was used across all image experiments as it led to better performance.

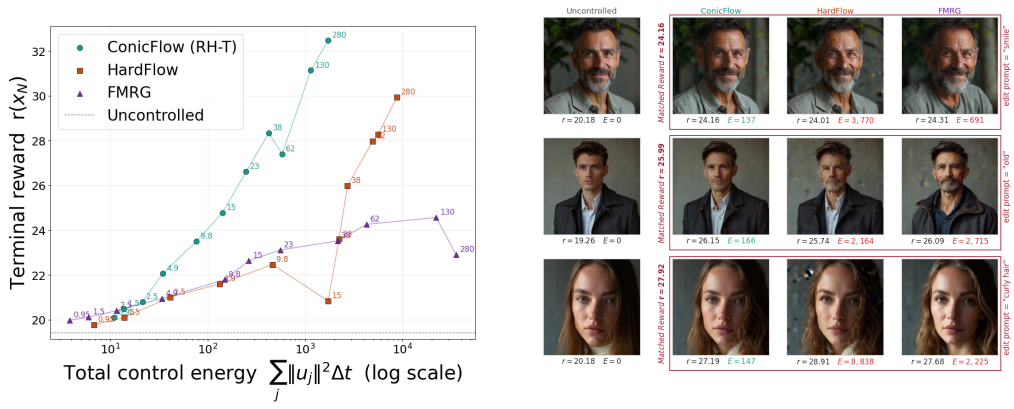


Figure 2: Trade-off between terminal reward and total control cost. **Left:** Reward vs. control energy on the editing task, traced by sweeping λ (upper left is better). **Right:** Edits from each method at approximately matched reward; ConicFlow reaches the same reward with much lower control energy.

ConicFlow achieves the best trade-off between reward and control energy (Figure 2, left). At matched reward, its edits stay closer to the original image (Figure 2, right). Table 1 quantifies this across multiple rewards, seeds and prompts, each describing a new optimization problem.

Table 1: Mean reward gain over a shared energy budget. Each method's reward–energy frontier, averaged over the control-energy range attained by all three methods and measured relative to the uncontrolled sample (units are those of each reward). Mean over n=4 (prompt, seed) pairs per reward.

Method	CLIP (↑)	ImageReward (↑)	PickScore (↑)
ConicFlow	$\mathbf{0.130} \pm 0.018$	$\mathbf{2.048} \pm 0.455$	$\mathbf{6.158} \pm 0.728$
HardFlow	0.118 ± 0.021	1.279 ± 0.374	1.736 ± 0.472
FMRG	0.065 ± 0.008	1.278 ± 0.504	2.670 ± 0.073

Constrained image setting. We consider the same image editing task, but now enforce the desired attribute through a hard constraint $r(x; c') \geq \tau$, i.e., $h(x) = \tau - r(x; c') \leq 0$, using PickScore as r. In the multi-constraint setting, we impose one such constraint per edit prompt, $h_k(x) = \tau_k - r(x; c'_k) \leq 0$ for $k = 1, \ldots, K$, requiring the sample to exhibit all K attributes simultaneously. For each problem, the threshold is set to $\tau = m \cdot r(x^{\text{unc}}; c')$, a multiple of the score achieved by the uncontrolled sample, with m ranging from 1.1 (easier) to 1.5 (harder). We report the control energy and the feasibility rate, i.e., the fraction of (prompt, seed) problems whose final sample satisfies the constraint. Since FMRG has no mechanism for enforcing hard constraints, we do not include it in this comparison.

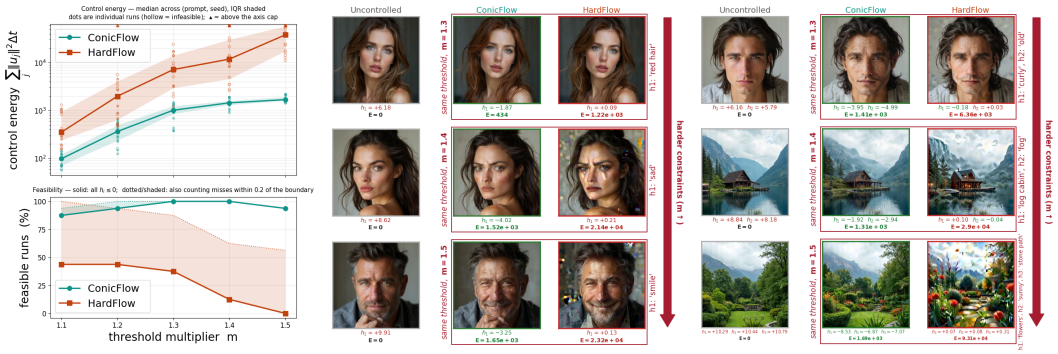


Figure 3: Constrained image editing comparison. **Top left:** control energy vs. threshold multiplier m, median over (prompt, seed) pairs. **Bottom left:** feasibility rate vs. m. **Middle:** single-constraint edits with increasing m. **Right:** multi-constraint edits with increasing m.

As m increases, both methods expend more control energy (Figure 3, top left), but ConicFlow satisfies the constraints with substantially less energy (up to $20\times$). ConicFlow also maintains a feasibility rate close to 100%, whereas HardFlow's drops as the threshold grows (bottom left). The examples demonstrate the same message: as m increases, both methods express the edit concept more strongly, but HardFlow deviates increasingly from the uncontrolled image, eventually producing visible noise and artifacts. This is most pronounced in the multi-constraint setting (right). See Appendix C for additional implementation details and results.

6 Conclusion

We introduced ConicFlow, a receding-horizon primal-dual framework for constrained inference with pretraned flow matching models. This framework combines reward optimization, control-energy regularization, and explicit terminal constraints. We adapt receding-horizon control and propose a terminal-estimate variant that uses the pretrained flow map to evaluate the reward and constraints at the predicted terminal state rather than at the end of the truncated window. Our analysis quantifies discretization and horizon truncation errors and establishes primal-dual convergence guarantees under convex-affine assumptions, including the propagation of local optimization errors across successive windows. Experiments on synthetic data and image editing demonstrate that this approach can improve the reward–control trade-off while maintaining high empirical feasibility.

AI USE STATEMENT

In this work, we used generative AI tools to assist in proving mathematical claims and writing proofs, and to provide feedback on research methodology and experimental design. We have not used generative AI tools to generate synthetic datasets, develop theoretical or conceptual frameworks, formulate mathematical claims, propose or refine hypotheses, implement methods, assist with translation, clean or reformat datasets, support qualitative or thematic data analysis, or interpret experimental results.

Additionally, we used generative AI tools to draft, edit, and polish parts of the manuscript; source and identify relevant literature; create or edit software code; and create or refine scientific figures. We have reviewed all AI-assisted work. AI-assisted mathematical arguments and proofs were independently checked by the authors for correctness, suggested references were verified against the original sources, and AI-assisted code, experimental designs, and figures were reviewed and validated by the authors before use.

We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Carmen Amo Alonso, Jing Shuang Li, James Anderson, and Nikolai Matni. Distributed and localized model-predictive control—part i: Synthesis and implementation. *IEEE Transactions on Control of Network Systems*, 10(2):1058–1068, 2022. doi: 10.1109/TCNS.2022.3219770.

Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N Pistikopoulos. The explicit solution of model predictive control via multiparametric quadratic programming. In *Proceedings of the 2000 American control conference. ACC (IEEE Cat. No. 00CH36334)*, volume 2, pp. 872–876. IEEE, 2000.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. *arXiv preprint arXiv:2402.14017*, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv preprint arXiv:2406.07507*, 2024.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *arXiv preprint arXiv:2505.18825*, 2025.

Jean-Baptiste Bouvier, Kanghyun Ryu, Kartik Nagpal, Qiayuan Liao, Koushil Sreenath, and Negar Mehr. DDAT: Diffusion policies enforcing dynamically admissible robot trajectories. In *Robotics: Science and Systems (RSS)*, 2025.

Raphael Cagienard, Pascal Grieder, Eric C. Kerrigan, and Manfred Morari. Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6):563–570, 2007. doi: 10.1016/j.jprocont.2007.01.001.

Jacob K. Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 89307–89333, 2024.

Jacob K Christopher, James E Warner, and Ferdinando Fioretto. Constraint-aware flow matching: Decision aligned end-to-end training for constrained sampling. *arXiv preprint arXiv:2605.12754*, 2026.

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.

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

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Weiguo Gao, Ming Li, and Qianxiao Li. Terminally constrained flow-based generative models from an optimal control perspective. *arXiv preprint arXiv:2601.09474*, 2026.

Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989. doi: 10.1016/0005-1098(89)90002-2.

Paul J. Goulart and Eric C. Kerrigan. Output feedback receding horizon control of constrained systems. *International Journal of Control*, 80(1):8–20, 2007. doi: 10.1080/00207170600892949.

Paul J. Goulart, Eric C. Kerrigan, and Jan M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006. doi: 10.1016/j.automatica.2005.08.023.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Jerry Y Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M Boffi. How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*, 2026.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2416–2425. IEEE, 2022.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Vince Kurtz and Alexander Davydov. Constrained flow matching via lagrangian dual flows. *arXiv preprint arXiv:2607.04513*, 2026.

Zeyang Li, Kaveh Alim, and Navid Azizan. Hardflow: Hard-constrained sampling for flow-matching models via trajectory optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

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

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

Xingchao Liu, Lemeng Wu, Shujian Zhang, Chengyue Gong, Wei Ping, and Qiang Liu. FlowGrad: Controlling the output of generative ODEs with gradients. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24335–24344, 2023.

Hao Luan, Yi Xian Goh, See-Kiong Ng, and Chun Kai Ling. Projected coupled diffusion for test-time constrained joint generation. In *International Conference on Learning Representations (ICLR)*, 2026.

Jacob Mattingley, Yang Wang, and Stephen Boyd. Receding horizon control: Automatic generation of high-speed solvers. *IEEE Control Systems Magazine*, 31(3):52–65, 2011. doi: 10.1109/MCS.2011.940571.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. doi: 10.1016/S0005-1098(99)00214-9.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 2065–2074. IEEE, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Ralf Römer, Alexander von Rohr, and Angela P. Schoellig. Diffusion predictive control with constraints. In *Proceedings of the 7th Annual Learning for Dynamics & Control Conference (L4DC)*, volume 283 of *Proceedings of Machine Learning Research*, pp. 791–803. PMLR, 2025.

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22522–22531. IEEE, 2023.

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

Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. Training free guided flow matching with optimal control. In *International Conference on Learning Representations (ICLR)*, 2025.

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

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

Stefano Zampini, Jacob K. Christopher, Luca Oneto, Davide Anguita, and Ferdinando Fioretto. Training-free constrained generation with stable diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained diffusers for safe planning and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Related Work

Inference-time control of pretrained diffusion and flow matching models has developed along two largely separate lines: *soft* guidance methods, which trade reward against fidelity to the pretrained dynamics but offer no feasibility guarantee, and *hard*-constraint methods, which enforce feasibility, most often by projecting onto the constraint set during sampling. ConicFlow belongs to the second line. It is a hard-constrained sampling method whose terminal constraints are enforced through dual multipliers, and whose objective additionally accommodates a reward and a control-energy penalty so that feasible samples are ranked by preference and stay close to the pretrained dynamics. To evaluate it we compare against a recent state-of-the-art representative from each line: Flow Map Reward Guidance (FMRG) (Huang et al., 2026) for soft guidance, and HardFlow (Li et al., 2026) for hard-constrained sampling in the tradition of projection-based methods. We review both families below and then discuss the control-theoretic tools we build on.

Flow matching and flow maps. The pretrained models we steer are trained with flow matching (Lipman et al., 2022; Liu et al., 2022) or, equivalently, stochastic interpolants (Albergo et al., 2025), which learn a velocity field b_t whose ODE transports a simple base distribution to the data distribution. Flow maps (Boffi et al., 2024; 2025), which directly predict the ODE solution between two arbitrary times, were introduced to accelerate sampling; we use a pretrained flow map both to integrate the dynamics and, in the RH-T variant, to predict the terminal sample from an intermediate state.

Soft guidance and reward-tilted sampling. Classifier guidance (Dhariwal & Nichol, 2021) and classifier-free guidance (Ho & Salimans, 2022) steer diffusion models by adding a (learned or amortized) gradient of a conditional likelihood to the score. Training-free variants replace the classifier with the gradient of a reward evaluated at a one-step estimate of the clean sample, as in diffusion posterior sampling (Chung et al., 2023) and its flow-based counterparts. A second line differentiates through the sampling ODE itself: FlowGrad (Liu et al., 2023) and D-Flow (Ben-Hamu et al., 2024) optimize the initial noise or the whole trajectory by backpropagating a terminal loss, and OC-Flow (Wang et al., 2025) recasts this as an optimal control problem solved via the Pontryagin maximum principle, with convergence guarantees in Euclidean space and on $\mathrm{SO}(3)$. OC-Flow is the closest precursor to our unconstrained formulation, and it shows that the KL divergence between the controlled and uncontrolled terminal distributions is bounded by the expected control cost, which justifies the quadratic energy term in (P-C); it does not, however, handle constraints, and solving the full-horizon adjoint problem is considerably more expensive than a receding-horizon scheme. Most recently, FMRG (Huang et al., 2026) reformulates guidance as *deterministic* optimal control and shows that the flow map arises naturally in the optimal feedback, yielding a single-trajectory, training-free method that matches or surpasses multi-particle guidance schemes with as few as three function evaluations at text-to-image scale. FMRG is provably near-optimal in the small-λ regime (small deviations from the pretrained flow), which is exactly the regime in which we observe it performing best (Section 5); it has no mechanism for enforcing hard terminal constraints, and its greedy, myopic control degrades when the reward lies far from the data manifold. We therefore use FMRG as the representative soft-guidance baseline, and, since it subsumes DPS-style methods as special cases, we do not compare against those separately.

Fine-tuning for reward alignment. An orthogonal approach adapts the model weights to a reward, via reinforcement learning (Black et al., 2024; Fan et al., 2023) or stochastic optimal control (Domingo i Enrich et al., 2025; Uehara et al., 2024). Fine-tuning amortizes the cost of steering over many samples but must be repeated for every new reward or constraint, can be prohibitively expensive for large models, and offers no per-sample feasibility guarantee. Our method is inference-time only and complementary to these approaches.

Hard-constrained sampling via projection. The dominant approach to hard constraints projects the iterate onto the feasible set at every (or every late) sampling step. Projected diffusion models (Christopher et al., 2024) interpret reverse diffusion as projected Langevin dynamics and require a projection operator for the constraint set at each step; Zampini et al. (2025) extend this to latent (Stable Diffusion) models and, notably, to copyright-style constraints on images, but still require a function measuring distance to the feasible set and a projection in latent space. Projected coupled diffusion (Luan et al., 2026) applies the same per-step projection to the joint generation of correlated

samples from several pretrained models. In robotics, diffusion predictive control with constraints (Römer et al., 2025) and DDAT (Bouvier et al., 2025) project generated trajectories onto sets that are dynamically admissible for the robot, either at every step or only late in sampling; these methods are tailored to state–action constraints and system dynamics and are less directly comparable to the general terminal-constraint problem we consider. As several authors have noted (Li et al., 2026; Kurtz & Davydov, 2026), projecting the *entire* sampling path is overly restrictive: intermediate iterates need not be feasible, exact projection onto a general nonlinear set is itself an optimization problem, and projecting inside an ODE produces a discontinuous, projected dynamical system that degrades sample quality.

Hard constraints via penalties and dual dynamics. Several recent works instead enforce terminal constraints through the sampling dynamics themselves. TOCFlow (Gao et al., 2026) formulates terminally constrained sampling as an optimal control problem, characterizes the value function through a Hamilton–Jacobi–Bellman equation, and shows that, as the control penalty vanishes, the terminal law converges to a generalized Wasserstein projection onto the constraint manifold $\{x : h(x) = 0\}$. Algorithmically, however, constraints enter through a quadratic terminal penalty $\|h(x)\|^2$, so exact feasibility is recovered only in the vanishing-penalty limit; the theoretical analysis is formulated around a smooth zero-level-set constraint manifold. Lagrangian dual flows (Kurtz & Davydov, 2026) augment the generative ODE with a dual co-state (and a slack variable for inequalities) so that constraint violation is driven to zero along the flow without projections or inner optimization; the authors also show why quadratic penalties of the TOCFlow type are a poor substitute for genuine dual variables. This is conceptually close to our dynamic primal–dual scheme, but LDF flows the multiplier continuously alongside the sample with no notion of look-ahead, and has so far been demonstrated only on low-dimensional and MNIST-scale problems. Constrained diffusers (Zhang et al., 2025) bring projected, primal–dual, and augmented-Lagrangian updates to reverse Langevin sampling for safe planning; they target a different, *distribution-level* problem, $\max_{\lambda \geq 0} \min_q \mathrm{KL}(q \,\|\, p) + \lambda^\top \mathbb{E}_{x \sim q}[g(x)]$, whose dual update requires an expectation over samples at every step, and their setting is diffusion-based trajectory planning rather than deterministic flows.

Hard constraints via trajectory optimization. HardFlow (Li et al., 2026) is the work most closely related to ours. It reformulates hard-constrained sampling from a flow matching model as a trajectory optimization problem in which constraints are imposed only at the terminal time, uses model predictive control ideas to decompose the long-horizon problem into a sequence of single-step subproblems, allows integral costs that limit distribution shift, and provides a control-theoretic bound on the gap between its surrogate and the ideal problem. It substantially outperforms projection-based methods on robotics, PDE boundary-control, and text-guided image-editing tasks, and it is therefore our representative of the hard-constraint family. The differences from ConicFlow are, first, that HardFlow optimizes a single step at a time and solves each constrained subproblem with an off-the-shelf solver, whereas we optimize over a horizon $H \geq 1$ with a dynamic primal–dual scheme whose multipliers are warm-started across windows; second, that HardFlow evaluates the constraint at the end of the current step, whereas our RH-T variant evaluates it at the predicted terminal point; and third, that we provide truncation and finite-iteration guarantees for both variants. Empirically, HardFlow is competitive precisely when large deviations from the pretrained flow are required, complementing FMRG's strength in the small-deviation regime, and ConicFlow combines the strengths of both (Section 5).

Receding-horizon and constrained control. Receding-horizon control provides a classical framework for handling finite-horizon objectives and hard constraints through repeated online optimization (García et al., 1989; Mayne et al., 2000). Several works have addressed its computational and robustness limitations. In particular, Kerrigan and collaborators studied move blocking to reduce the online decision dimension while retaining feasibility and stability guarantees Cagienard et al. (2007), convex parametrizations of feedback policies for robust constrained finite-horizon control (Goulart et al., 2006), and constrained output-feedback receding-horizon control under process and measurement uncertainty (Goulart & Kerrigan, 2007). Related work has developed high-speed RHC solvers Mattingley et al. (2011) and scalable localized MPC architectures Alonso et al. (2022). Our setting differs in that the controlled dynamics are given by a pretrained generative flow and the objective and hard constraints are imposed on the generated terminal sample.

B PROOFS

B.1 PROOF OF LEMMA 1

Proof. **Proof strategy.** We prove the result by comparing the continuous and discretized optimal values in both directions. The argument proceeds in four steps.

First, the quadratic control penalty together with the Lipschitz terminal reward yields a coercive lower bound on the objective. Using the finite-cost feasible trajectory from Assumption 1 i, this gives a uniform energy bound on all controls relevant to the comparison and, consequently, a common bounded region containing their trajectories.

Second, we show that replacing a continuous control by its block averages and propagating the dynamics with Euler discretization changes the terminal state by $O(\Delta t)$. Conversely, the piecewise-constant interpolation of a discrete control incurs the same order of terminal-state error.

Third, the resulting $O(\Delta t)$ terminal constraint violation is corrected using Assumption 1 ii, requiring only an $O(\Delta t)$ perturbation of the control. This construction also establishes feasibility and existence of the discrete optimum for sufficiently large N.

Finally, we apply these constructions in both directions. Discretizing and repairing the continuous optimum gives an upper bound on the discrete optimal value, while interpolating and repairing the discrete optimum gives the reverse inequality. Combining the two yields

$$\left|J_D\big(U_D^\star(N)\big) - J_C(u_C^\star)\right| = O(\Delta t) = O\left(\frac{1}{N}\right).$$

We next make the quantitative content of Assumptions 1 from i to iii explicit and introduce the constants used in the proof.

Quantitative form of the assumptions. By Assumption 1 i, there exist constants $L_b, L_b^t, L_r, L_h > 0$ such that

$$\|b_t(x) - b_s(y)\| \le L_b\|x - y\| + L_b^t|t - s|,$$

and

$$|r(x) - r(y)| \le L_r\|x - y\|, \qquad \|h(x) - h(y)\| \le L_h\|x - y\|.$$

We also define

$$B_0 := \sup_{t\in[0,1]} \|b_t(x_0)\| < \infty.$$

By Assumption 1 ii, there exists a feasible continuous-time control u^0 satisfying

$$h(x_1^{u^0}) \le 0, \qquad J_C(u^0) < \infty,$$

and the continuous problem admits an optimal control $u_C^\star$.

Assumption 1 iii is used in the following quantitative form. On the energy ball introduced below, there exist constants $\kappa, \rho_0 > 0$, independent of N, such that any continuous or discrete control whose terminal violation is at most ρ_0 can be made feasible by a control perturbation whose norm is at most κ times the violation. For continuous controls we use the L^2 norm, while for discrete controls we use

$$\|U\|_{\ell^2_{\Delta t}}^2 := \sum_{k=0}^{N-1} \Delta t\, \|u_k\|^2.$$

A priori energy bound. We first derive the energy radius on which the remainder of the proof is carried out. For any continuous control u, the dynamics and Assumption 1 i imply

$$\|x_t^u - x_0\| \le B_0 t + L_b \int_0^t \|x_s^u - x_0\|\, ds + \int_0^t \|u_s\|\, ds.$$

By Cauchy–Schwarz and Grönwall's inequality,

$$\sup_{0\le t\le 1} \|x_t^u - x_0\| \le e^{L_b}\left(B_0 + \|u\|_{L^2}\right).$$

Hence, by Lipschitz continuity of r,

$$r(x_1^u) \le |r(x_0)| + L_r e^{L_b} \left(B_0 + \|u\|_{L^2}\right).$$

Define

$$a_r := L_r e^{L_b}, \qquad \beta_r := |r(x_0)| + a_r B_0 + \lambda a_r^2.$$

Young's inequality gives

$$a_r \|u\|_{L^2} \le \frac{1}{4\lambda} \|u\|_{L^2}^2 + \lambda a_r^2.$$

Therefore,

$$J_C(u) \ge \frac{1}{4\lambda} \|u\|_{L^2}^2 - \beta_r.$$

The same argument applied to the Euler recursion gives

$$J_D(U) \ge \frac{1}{4\lambda} \|U\|_{\ell^2_{\Delta t}}^2 - \beta_r.$$

Define

$$R_\star := 4\lambda\big(J_C(u^0) + \beta_r\big), \qquad \overline{R} := R_\star + 1.$$

Since $u_C^\star$ is optimal and u^0 is feasible,

$$J_C(u_C^\star) \le J_C(u^0).$$

Using the coercivity estimate above,

$$\|u_C^\star\|_{L^2}^2 \le R_\star < \overline{R}.$$

Step 1: Uniform confinement of relevant trajectories.

Consider any continuous control satisfying

$$\|u\|_{L^2}^2 \le \overline{R}.$$

The preceding trajectory estimate gives

$$\sup_{0 \le t \le 1} \|x_t^u - x_0\| \le \big(B_0 + \sqrt{\overline{R}}\big) e^{L_b}.$$

Define

$$\rho_{\overline{R}} := \big(B_0 + \sqrt{\overline{R}}\big) e^{L_b},$$

and

$$M_{\overline{R}} := B_0 + L_b \rho_{\overline{R}}.$$

Then

$$\|b_t(x_t^u)\| \le M_{\overline{R}}.$$

The same conclusion holds for Euler trajectories. Let $U = (u_0, \dots, u_{N-1})$ satisfy

$$\|U\|_{\ell^2_{\Delta t}}^2 \le \overline{R}.$$

The Euler recursion gives

$$\|x_{k+1}^U - x_0\| \le (1 + L_b \Delta t) \|x_k^U - x_0\| + B_0 \Delta t + \Delta t \, \|u_k\|.$$

Moreover,

$$\sum_{k=0}^{N-1} \Delta t \, \|u_k\| \le \left(\sum_{k=0}^{N-1} \Delta t \, \|u_k\|^2 \right)^{1/2} \le \sqrt{\overline{R}}.$$

Iterating the recursion and using $(1 + L_b \Delta t)^N \le e^{L_b}$ gives

$$\max_{0 \le k \le N} \|x_k^U - x_0\| \le \rho_{\overline{R}}.$$

Thus, both continuous and Euler trajectories generated by controls in the energy ball $\overline{R}$ remain uniformly bounded, and the drift along all such trajectories is bounded by $M_{\overline{R}}$.

Step 2: Continuous–discrete approximation.

We next establish the approximation estimates required in the remainder of the proof.

Continuous-to-discrete approximation. Let $u \in L^2([0,1];\mathbb{R}^d)$ satisfy

$$\|u\|_{L^2}^2 \le \overline{R},$$

and define its block averages by

$$\overline{u}_k := \frac{1}{\Delta t}\int_{t_k}^{t_{k+1}} u_s\, ds.$$

By Jensen's inequality,

$$\sum_{k=0}^{N-1} \Delta t\, \|\overline{u}_k\|^2 \le \int_0^1 \|u_t\|^2\, dt \le \overline{R}.$$

Let $x^u(t)$ denote the exact continuous trajectory and let x_k^Δ denote the Euler trajectory driven by $\overline{U} = (\overline{u}_0, \dots, \overline{u}_{N-1})$:

$$x_{k+1}^\Delta = x_k^\Delta + \Delta t\big(b_{t_k}(x_k^\Delta) + \overline{u}_k\big).$$

Set

$$e_k := x^u(t_k) - x_k^\Delta.$$

By construction,

$$\int_{t_k}^{t_{k+1}} \big(u_s - \overline{u}_k\big)\, ds = 0.$$

Thus the integrated control contribution is reproduced exactly over each grid interval. Subtracting the Euler recursion from the exact dynamics gives

$$e_{k+1} = e_k + \Delta t\big(b_{t_k}(x^u(t_k)) - b_{t_k}(x_k^\Delta)\big) + \tau_k,$$

where

$$\tau_k := \int_{t_k}^{t_{k+1}} [b_s(x^u(s)) - b_{t_k}(x^u(t_k))]\, ds.$$

Define

$$q_k := \int_{t_k}^{t_{k+1}} \|u_s\|^2\, ds.$$

For $s \in [t_k, t_{k+1}]$, Step 1 and Cauchy–Schwarz imply

$$\|x^u(s) - x^u(t_k)\| \le M_{\overline{R}}(s - t_k) + \sqrt{s - t_k}\,\sqrt{q_k}.$$

Therefore,

$$\|\tau_k\| \le L_b M_{\overline{R}} \Delta t^2 + \frac{1}{2} L_b^t \Delta t^2 + L_b \Delta t^{3/2}\sqrt{q_k}.$$

Moreover,

$$\sum_{k=0}^{N-1} \sqrt{q_k} \le \sqrt{N}\left(\sum_{k=0}^{N-1} q_k\right)^{1/2} \le \frac{\sqrt{\overline{R}}}{\sqrt{\Delta t}}.$$

Consequently,

$$\sum_{k=0}^{N-1} \|\tau_k\| \le \left(L_b M_{\overline{R}} + \frac{1}{2} L_b^t + L_b\sqrt{\overline{R}}\right)\Delta t.$$

Since

$$\|e_{k+1}\| \le (1 + L_b \Delta t)\|e_k\| + \|\tau_k\|,$$

discrete Grönwall yields

$$\max_{0 \le k \le N} \|x^u(t_k) - x_k^\Delta\| \le C_0 \Delta t,$$

where

$$C_0 := e^{L_b}\left(L_b M_{\overline{R}} + \frac{1}{2}L_b^t + L_b\sqrt{\overline{R}}\right).$$

Discrete-to-continuous interpolation. Conversely, let $U = (u_0, \dots, u_{N-1})$ satisfy

$$\|U\|_{\ell^2_{\Delta t}}^2 \le \overline{R},$$

and define its piecewise-constant interpolation

$$\widehat{u}(t) := u_k, \qquad t \in [t_k, t_{k+1}).$$

Then

$$\|\widehat{u}\|_{L^2}^2 = \|U\|_{\ell^2_{\Delta t}}^2.$$

Since the block average of $\widehat{u}$ on each interval is exactly u_k, the preceding argument applies directly and gives

$$\max_{0\le k\le N} \|\widehat{x}(t_k) - x_k^U\| \le C_0\Delta t,$$

where $\widehat{x}$ is the exact continuous trajectory generated by $\widehat{u}$.

Sensitivity to control perturbations. Let $u, v \in L^2([0,1];\mathbb{R}^d)$. Then

$$\|x_t^u - x_t^v\| \le L_b \int_0^t \|x_s^u - x_s^v\|\, ds + \int_0^t \|u_s - v_s\|\, ds.$$

Grönwall and Cauchy–Schwarz imply

$$\sup_{0\le t\le 1} \|x_t^u - x_t^v\| \le e^{L_b}\|u - v\|_{L^2}.$$

Similarly, for two discrete controls U and V,

$$\max_{0\le k\le N} \|x_k^U - x_k^V\| \le e^{L_b}\|U - V\|_{\ell^2_{\Delta t}}.$$

Thus Step 2 provides the two estimates used below:

$$\text{Euler endpoint error} = O(\Delta t), \qquad \text{control perturbation } \delta \Longrightarrow \text{state perturbation } O(\delta).$$

Step 3: Discrete feasibility, existence, and energy bound.

Let $\overline{U}^0$ denote the block average of the feasible reference control u^0. From the coercivity estimate,

$$\|u^0\|_{L^2}^2 \le R_\star,$$

and Jensen's inequality gives

$$\|\overline{U}^0\|_{\ell^2_{\Delta t}}^2 \le R_\star < \overline{R}.$$

By Step 2,

$$\|x_N^{\overline{U}^0} - x_1^{u^0}\| \le C_0\Delta t.$$

Since u^0 is feasible and h is L_h-Lipschitz,

$$\|[h(x_N^{\overline{U}^0})]_+\| \le L_h C_0 \Delta t.$$

Define

$$N_0 := \left\lceil \max\left\{1, \frac{L_h C_0}{\rho_0}\right\}\right\rceil.$$

For $N \ge N_0$, Assumption 1 iii provides a feasible discrete control $\widetilde{U}^0$ satisfying

$$\|\widetilde{U}^0 - \overline{U}^0\|_{\ell^2_{\Delta t}} \le \delta_N,$$

where

$$\delta_N := \kappa L_h C_0 \Delta t.$$

The change in quadratic control cost satisfies

$$\frac{1}{2\lambda}\left(\|\widetilde{U}^0\|_{\ell^2_{\Delta t}}^2 - \|\overline{U}^0\|_{\ell^2_{\Delta t}}^2\right) \le \frac{\sqrt{\overline{R}}}{\lambda}\delta_N + \frac{1}{2\lambda}\delta_N^2.$$

Moreover, the control-sensitivity estimate and Lipschitz continuity of r give

$$\left|r(x_N^{\widetilde{U}^0}) - r(x_N^{\overline{U}^0})\right| \le L_r e^{L_b}\delta_N.$$

Using also the Euler endpoint estimate and the fact that block averaging cannot increase the quadratic control cost,

$$J_D(\widetilde{U}^0) \le J_C(u^0) + L_r C_0 \Delta t + \left(\frac{\sqrt{\overline{R}}}{\lambda} + L_r e^{L_b}\right)\delta_N + \frac{1}{2\lambda}\delta_N^2.$$

Since $\Delta t^2 \le \Delta t$, define

$$C_{\mathrm{disc}} := C_0\left[L_r + \left(\frac{\sqrt{\overline{R}}}{\lambda} + L_r e^{L_b}\right)\kappa L_h + \frac{\kappa^2 L_h^2 C_0}{2\lambda}\right].$$

Then

$$J_D(\widetilde{U}^0) \le J_C(u^0) + C_{\mathrm{disc}}\Delta t.$$

Thus the discrete feasible set is nonempty for all $N \ge N_0$.

For fixed N, the map

$$U \mapsto x_N^U$$

is continuous, and therefore the feasible set is closed. Moreover,

$$J_D(U) \ge \frac{1}{4\lambda}\|U\|_{\ell^2_{\Delta t}}^2 - \beta_r,$$

so J_D is coercive. Since the discrete control space is finite-dimensional, the discrete problem admits a global minimizer $U_D^\star(N)$.

By optimality,

$$J_D\big(U_D^\star(N)\big) \le J_D(\widetilde{U}^0) \le J_C(u^0) + C_{\mathrm{disc}}\Delta t.$$

Using the coercivity estimate again,

$$\|U_D^\star(N)\|_{\ell^2_{\Delta t}}^2 \le R_\star + 4\lambda C_{\mathrm{disc}}\Delta t.$$

Define

$$N_2 := \lceil \max\{1, 4\lambda C_{\mathrm{disc}}\} \rceil, \qquad N_1 := \max\{N_0, N_2\}.$$

Then, for every $N \ge N_1$,

$$\|U_D^\star(N)\|_{\ell^2_{\Delta t}}^2 \le R_\star + 1 = \overline{R}.$$

Step 4: Two-sided comparison of the optimal values.

We now apply the preceding constructions to the two optimal controls.

Continuous optimum to discrete feasible control. Since

$$\|u_C^\star\|_{L^2}^2 \le R_\star < \overline{R},$$

let $\overline{U}^\star$ denote the block average of $u_C^\star$. By Jensen's inequality,

$$\|\overline{U}^\star\|_{\ell^2_{\Delta t}}^2 \le R_\star.$$

Step 2 gives

$$\|x_N^{\overline{U}^\star} - x_1^{u_C^\star}\| \le C_0\Delta t.$$

Since $u_C^\star$ is feasible,

$$\|[h(x_N^{\overline{U}^\star})]_+\| \le L_h C_0 \Delta t.$$

For $N \geq N_0$, terminal repairability provides a feasible discrete control $\widetilde{U}^\star$ satisfying

$$\|\widetilde{U}^\star - \overline{U}^\star\|_{\ell^2_{\Delta t}} \leq \delta_N.$$

Repeating the preceding objective estimate gives

$$J_D(\widetilde{U}^\star) \leq J_C(u_C^\star) + C_{\mathrm{disc}}\Delta t.$$

Since $\widetilde{U}^\star$ is feasible for the discrete problem,

$$J_D\big(U_D^\star(N)\big) - J_C(u_C^\star) \leq C_{\mathrm{disc}}\Delta t.$$

Discrete optimum to continuous feasible control. For $N \geq N_1$, Step 3 gives

$$\|U_D^\star(N)\|^2_{\ell^2_{\Delta t}} \leq \overline{R}.$$

Define its piecewise-constant interpolation by

$$\widehat{u}_D^\star(t) := u_{D,k}^\star, \qquad t \in [t_k, t_{k+1}).$$

The quadratic control energy is preserved exactly:

$$\|\widehat{u}_D^\star\|^2_{L^2} = \|U_D^\star(N)\|^2_{\ell^2_{\Delta t}}.$$

Let $\widehat{x}_D^\star$ denote the corresponding exact continuous trajectory. Step 2 gives

$$\|\widehat{x}_D^\star(1) - x_N^{U_D^\star(N)}\| \leq C_0\Delta t.$$

Since $U_D^\star(N)$ is feasible for the discrete problem,

$$\|[h(\widehat{x}_D^\star(1))]_+\| \leq L_h C_0 \Delta t.$$

For $N \geq N_0$, Assumption 1 iii provides a feasible continuous control $\widetilde{u}_D^\star$ satisfying

$$\|\widetilde{u}_D^\star - \widehat{u}_D^\star\|_{L^2} \leq \delta_N.$$

The piecewise-constant interpolation preserves the quadratic running cost exactly, and the same endpoint and repair estimates therefore yield

$$J_C(\widetilde{u}_D^\star) \leq J_D\big(U_D^\star(N)\big) + C_{\mathrm{disc}}\Delta t.$$

Since $\widetilde{u}_D^\star$ is feasible for the continuous problem, optimality of $u_C^\star$ implies

$$J_C(u_C^\star) - J_D\big(U_D^\star(N)\big) \leq C_{\mathrm{disc}}\Delta t.$$

Combining the two directions gives

$$\big|J_D\big(U_D^\star(N)\big) - J_C(u_C^\star)\big| \leq C_{\mathrm{disc}}\Delta t = \frac{C_{\mathrm{disc}}}{N},$$

for every $N \geq N_1$. This proves the claim.

□

Remark 2 (What determines the discretization rate)**.** *The proof separates the error into two mechanisms. Regularity of the flow, together with block averaging of the additive control, gives the first-order Euler endpoint estimate*

$$\|x_1^u - x_N^{\overline{U}}\| = O(\Delta t).$$

Local terminal repairability then converts the resulting $O(\Delta t)$ constraint violation into an $O(\Delta t)$ perturbation of the control. Consequently, enforcing exact terminal feasibility does not change the first-order discretization rate.

B.2 Unconstrained Receding-Horizon Bounds

For completeness, we record the unconstrained counterparts of the constrained receding-horizon bounds in the main text. These results are special cases in which the terminal feasibility constraint is absent.

Lemma 5 (Unconstrained RH-H bound)**.** *Suppose the discretized dynamics are affine,*

$$x_{k+1} = x_k + \Delta t(A_k x_k + c_k + u_k), \qquad \Delta t = \frac{1}{N},$$

with $\|A_k x + c_k\| \le M$ on the relevant reachable set, and let r be concave and L_r-Lipschitz. For $1 \le H \le N$, let $\boldsymbol{x}^{H,\star}$ denote the exact unconstrained RH-H closed-loop trajectory with horizon H, and let $\boldsymbol{x}^\star$ denote the full-horizon optimum. Then

$$0 \le J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^\star) \le C_H^{\mathrm{U}} \frac{N-H}{N},$$

where

$$C_H^{\mathrm{U}} := 2\max\left\{ L_r M + \frac{\lambda L_r^2}{2}, \frac{M^2}{2\lambda} \right\}.$$

In particular, C_H^{U} is independent of N and H.

Proof. Recall that

$$J_{j,L}^\star(x) := \min_{u_{j:j+L-1}} \left\{ \sum_{k=j}^{j+L-1} \frac{\Delta t}{2\lambda} \|u_k\|^2 - r(x_{j+L}) \right\},$$

subject to

$$x_{k+1} = x_k + \Delta t(A_k x_k + c_k + u_k), \qquad x_j = x.$$

We write $J_j^\star(x) := J_{j,N-j}^\star(x)$ for the full remaining-horizon value.

Under the affine dynamics, the terminal state is affine in the control sequence. Since r is concave, $-r$ is convex, while the quadratic control cost is coercive. Hence all finite-horizon unconstrained problems considered below attain their minima.

Step 1: Local sensitivity to the terminal horizon.

Fix j, L, q such that $j + L + q \le N$, and set

$$\tau := q\Delta t.$$

We first show that

$$\left| J_{j,L+q}^\star(x) - J_{j,L}^\star(x) \right| \le C_{\mathrm{loc}} \tau,$$

where

$$C_{\mathrm{loc}} := \max\left\{ L_r M + \frac{\lambda L_r^2}{2}, \frac{M^2}{2\lambda} \right\}.$$

Let $u_{j:j+L+q-1}^\star$ be optimal for $J_{j,L+q}^\star(x)$, with associated trajectory $x_k^\star$, and denote the energy of its last q controls by

$$E := \sum_{k=j+L}^{j+L+q-1} \Delta t \, \|u_k^\star\|^2.$$

Using only the first L controls gives an admissible candidate for $J_{j,L}^\star(x)$. Therefore,

$$J_{j,L}^\star(x) - J_{j,L+q}^\star(x) \le r(x_{j+L+q}^\star) - r(x_{j+L}^\star) - \frac{E}{2\lambda}.$$

By the L_r-Lipschitz continuity of r,

$$J_{j,L}^\star(x) - J_{j,L+q}^\star(x) \le L_r \|x_{j+L+q}^\star - x_{j+L}^\star\| - \frac{E}{2\lambda}.$$

Using the dynamics and the drift bound $\|A_k x_k + c_k\| \le M$ from Assumption 2,

$$\|x^\star_{j+L+q} - x^\star_{j+L}\| \le \sum_{k=j+L}^{j+L+q-1} \Delta t\,(M + \|u^\star_k\|)$$
$$\le M\tau + \sqrt{\tau E},$$

where the second inequality follows from Cauchy–Schwarz. Hence

$$J^\star_{j,L}(x) - J^\star_{j,L+q}(x) \le L_r M\tau + L_r\sqrt{\tau E} - \frac{E}{2\lambda}.$$

Completing the square in $\sqrt{E}$ gives

$$L_r\sqrt{\tau E} - \frac{E}{2\lambda} \le \frac{\lambda L_r^2}{2}\tau,$$

and therefore

$$J^\star_{j,L}(x) - J^\star_{j,L+q}(x) \le \left(L_r M + \frac{\lambda L_r^2}{2}\right)\tau.$$

For the reverse direction, let $u^\star_{j:j+L-1}$ be optimal for $J^\star_{j,L}(x)$ and let $\bar{x} := x^\star_{j+L}$ be its terminal state. Extend this trajectory over the next q steps by choosing

$$u_k = -\big(A_k\bar{x} + c_k\big), \qquad k = j+L, \dots, j+L+q-1.$$

Then

$$x_{k+1} = \bar{x},$$

so the state, and hence the terminal reward, remains unchanged throughout the extension. Since $\|A_k\bar{x} + c_k\| \le M$, the additional control cost is at most

$$\sum_{k=j+L}^{j+L+q-1} \frac{\Delta t}{2\lambda}\|u_k\|^2 \le \frac{M^2}{2\lambda}\tau.$$

Thus,

$$J^\star_{j,L+q}(x) - J^\star_{j,L}(x) \le \frac{M^2}{2\lambda}\tau.$$

Combining the two directions yields

$$\big|J^\star_{j,L+q}(x) - J^\star_{j,L}(x)\big| \le C_{\mathrm{loc}} q\Delta t.$$

Step 2: Accumulation along the RH-H trajectory.

Set

$$m := N - H,$$

so that $m\Delta t = (N-H)/N$. For $j = 0, \dots, m$, define

$$W_j(x) := J^\star_{j,H}(x).$$

At $j = m$, the horizon H reaches the true terminal time, and therefore

$$W_m(x) = J^\star_m(x).$$

Let $x_j^{H,\star}$ denote the RH-H closed-loop state and $u_j^{H,\star}$ the first control returned by the H-step problem at time j. By optimality of the truncated problem,

$$W_j(x_j^{H,\star}) = \frac{\Delta t}{2\lambda}\|u_j^{H,\star}\|^2 + J^\star_{j+1,H-1}(x_{j+1}^{H,\star}),$$

for $j = 0, \dots, m-1$. Therefore,

$$\frac{\Delta t}{2\lambda}\|u_j^{H,\star}\|^2 = W_j(x_j^{H,\star}) - W_{j+1}(x_{j+1}^{H,\star})$$
$$+ \Big[W_{j+1}(x_{j+1}^{H,\star}) - J^\star_{j+1,H-1}(x_{j+1}^{H,\star})\Big].$$

Applying the local horizon estimate with $q = 1$ gives

$$W_{j+1}(x_{j+1}^{H,\star}) - J_{j+1,H-1}^{\star}(x_{j+1}^{H,\star}) \le C_{\mathrm{loc}}\Delta t.$$

Hence

$$\frac{\Delta t}{2\lambda}\|u_j^{H,\star}\|^2 \le W_j(x_j^{H,\star}) - W_{j+1}(x_{j+1}^{H,\star}) + C_{\mathrm{loc}}\Delta t.$$

Summing from $j = 0$ to $m - 1$ yields

$$\sum_{j=0}^{m-1} \frac{\Delta t}{2\lambda}\|u_j^{H,\star}\|^2 \le W_0(x_0) - W_m(x_m^{H,\star}) + mC_{\mathrm{loc}}\Delta t.$$

From time $m = N - H$ onward, the remaining horizon is at most H, so RH-H coincides with full remaining-horizon re-optimization. Consequently, its remaining cost from $x_m^{H,\star}$ is exactly $J_m^{\star}(x_m^{H,\star}) = W_m(x_m^{H,\star})$. Thus

$$J(\boldsymbol{x}^{H,\star}) \le W_0(x_0) + mC_{\mathrm{loc}}\Delta t.$$

Finally, $W_0(x_0) = J_{0,H}^{\star}(x_0)$, while $J(\boldsymbol{x}^{\star}) = J_{0,N}^{\star}(x_0)$. Applying the local horizon estimate once more, now with $L = H$ and $q = m$, gives

$$W_0(x_0) - J(\boldsymbol{x}^{\star}) \le C_{\mathrm{loc}} m\Delta t.$$

Therefore,

$$J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^{\star}) \le 2C_{\mathrm{loc}} m\Delta t.$$

Since $\boldsymbol{x}^{\star}$ is globally optimal,

$$J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^{\star}) \ge 0.$$

Using $m = N - H$ and $\Delta t = 1/N$,

$$0 \le J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^{\star}) \le 2C_{\mathrm{loc}}\frac{N-H}{N}.$$

Defining

$$C_H^{\mathrm{U}} := 2C_{\mathrm{loc}} = 2\max\left\{L_r M + \frac{\lambda L_r^2}{2}, \frac{M^2}{2\lambda}\right\}$$

proves the result. □

Lemma 6 (Unconstrained RH-T bound)**.** *Suppose the hypotheses of Lemma 5 hold and, in addition,*

$$\kappa_A := \sup_{0\le k\le \ell\le N}\left\|\prod_{s=k}^{\ell-1}(I + \Delta t A_s)\right\| < \infty.$$

For $1 \le H \le N$, let $\widehat{\boldsymbol{x}}^{H,\star}$ denote the exact unconstrained RH-T closed-loop trajectory obtained by optimizing controls over an H-step window and propagating the local endpoint to time N under the uncontrolled pretrained flow. Then

$$0 \le J(\widehat{\boldsymbol{x}}^{H,\star}) - J(\boldsymbol{x}^{\star}) \le C_T^{\mathrm{U}}\frac{N-H}{N},$$

where

$$C_T^{\mathrm{U}} := \frac{\lambda}{2}\kappa_A^2 L_r^2.$$

In particular, C_T^{U} is independent of N and H.

Proof. As in the proof of Lemma 5, the quadratic control penalty and Lipschitz terminal reward ensure that all finite-dimensional optimization problems below attain their minima. We therefore focus only on the additional structure introduced by terminal rollout.

For $j + L \le N$, define the rollout value

$$\widehat{J}_{j,L}^{\star}(x) := \min_{u_{j:j+L-1}}\left\{\sum_{k=j}^{j+L-1}\frac{\Delta t}{2\lambda}\|u_k\|^2 - r(\widehat{x}_N)\right\},$$

where the state follows

$$x_{k+1} = x_k + \Delta t(A_k x_k + c_k + u_k)$$

for $k = j, \dots, j + L - 1$, and is subsequently propagated to time N under the uncontrolled flow, i.e., $u_k = 0$ for $k \geq j + L$.

Step 1: Terminal rollout does not accumulate local horizon errors.

Increasing the control horizon only enlarges the feasible set of control sequences. Hence

$$J_j^\star(x) \leq \widehat{J}_{j,L+1}^\star(x) \leq \widehat{J}_{j,L}^\star(x).$$

Let

$$m := N - H,$$

and define

$$\widehat{W}_j(x) := \widehat{J}_{j,H}^\star(x), \qquad j = 0, \dots, m.$$

At $j = m$, exactly H steps remain until the true terminal time, and therefore

$$\widehat{W}_m(x) = J_m^\star(x).$$

Let $\widehat{x}_j^{H,\star}$ denote the RH-T closed-loop state and $\widehat{u}_j^{H,\star}$ the first control selected at time j. By optimality of the H-step rollout problem,

$$\widehat{W}_j(\widehat{x}_j^{H,\star}) = \frac{\Delta t}{2\lambda}\|\widehat{u}_j^{H,\star}\|^2 + \widehat{J}_{j+1,H-1}^\star(\widehat{x}_{j+1}^{H,\star}).$$

Since allowing one additional control action cannot increase the optimal value,

$$\widehat{W}_{j+1}(\widehat{x}_{j+1}^{H,\star}) = \widehat{J}_{j+1,H}^\star(\widehat{x}_{j+1}^{H,\star}) \leq \widehat{J}_{j+1,H-1}^\star(\widehat{x}_{j+1}^{H,\star}).$$

Consequently,

$$\frac{\Delta t}{2\lambda}\|\widehat{u}_j^{H,\star}\|^2 \leq \widehat{W}_j(\widehat{x}_j^{H,\star}) - \widehat{W}_{j+1}(\widehat{x}_{j+1}^{H,\star}).$$

Summing from $j = 0$ to $m - 1$ gives

$$\sum_{j=0}^{m-1} \frac{\Delta t}{2\lambda}\|\widehat{u}_j^{H,\star}\|^2 \leq \widehat{W}_0(x_0) - \widehat{W}_m(\widehat{x}_m^{H,\star}).$$

From time m onward, the control horizon reaches the true terminal time, so the remaining closed-loop cost is $J_m^\star(\widehat{x}_m^{H,\star}) = \widehat{W}_m(\widehat{x}_m^{H,\star})$. Therefore,

$$J(\widehat{\boldsymbol{x}}^{H,\star}) \leq \widehat{J}_{0,H}^\star(x_0).$$

Since the full-horizon problem has a larger control set,

$$J(\boldsymbol{x}^\star) = J_0^\star(x_0) \leq \widehat{J}_{0,H}^\star(x_0).$$

It is therefore sufficient to bound the single initial rollout gap $\widehat{J}_{0,H}^\star(x_0) - J_0^\star(x_0)$.

Step 2: Bounding the initial rollout gap.

Let $u_0^\star, \dots, u_{N-1}^\star$ denote the controls associated with the full-horizon optimum $\boldsymbol{x}^\star$. Define

$$\tau_H := (N - H)\Delta t = \frac{N - H}{N},$$

and let

$$E_H := \sum_{k=H}^{N-1} \Delta t \, \|u_k^\star\|^2$$

be the control energy of its omitted tail.

Construct a feasible candidate for the H-step rollout problem by keeping the first H optimal controls and setting all subsequent controls to zero:

$$\widetilde{u}_k = \begin{cases} u_k^\star, & k = 0, \dots, H-1, \\ 0, & k = H, \dots, N-1. \end{cases}$$

Let $\widetilde{x}_k$ denote the resulting trajectory. Since the two trajectories coincide at time H, define

$$\delta_k := x_k^\star - \widetilde{x}_k, \qquad \delta_H = 0.$$

For $k \geq H$, affine dynamics give

$$\delta_{k+1} = (I + \Delta t A_k)\delta_k + \Delta t\, u_k^\star.$$

Writing

$$\Phi_{\ell,k} := \prod_{s=k}^{\ell-1} (I + \Delta t A_s), \qquad \Phi_{k,k} := I,$$

we obtain

$$\delta_N = \sum_{k=H}^{N-1} \Delta t\, \Phi_{N,k+1} u_k^\star.$$

By the definition of κ_A,

$$\|\delta_N\| \leq \kappa_A \sum_{k=H}^{N-1} \Delta t\, \|u_k^\star\|.$$

Cauchy–Schwarz then gives

$$\|\delta_N\| \leq \kappa_A \sqrt{\tau_H E_H}.$$

Because $\widetilde{u}$ is feasible for the rollout problem,

$$\widehat{J}_{0,H}^\star(x_0) - J_0^\star(x_0) \leq J(\widetilde{\boldsymbol{x}}) - J(\boldsymbol{x}^\star).$$

The first H control costs cancel, while the omitted tail energy appears with a negative sign:

$$J(\widetilde{\boldsymbol{x}}) - J(\boldsymbol{x}^\star) = -\frac{E_H}{2\lambda} + r(x_N^\star) - r(\widetilde{x}_N).$$

Using the L_r-Lipschitz continuity of r,

$$\widehat{J}_{0,H}^\star(x_0) - J_0^\star(x_0) \leq L_r \kappa_A \sqrt{\tau_H E_H} - \frac{E_H}{2\lambda}.$$

Completing the square in $\sqrt{E_H}$ yields

$$L_r \kappa_A \sqrt{\tau_H E_H} - \frac{E_H}{2\lambda} \leq \frac{\lambda}{2} \kappa_A^2 L_r^2 \tau_H.$$

Therefore,

$$0 \leq \widehat{J}_{0,H}^\star(x_0) - J_0^\star(x_0) \leq \frac{\lambda}{2} \kappa_A^2 L_r^2 \frac{N-H}{N}.$$

Combining this estimate with Step 1 and using the global optimality of $\boldsymbol{x}^\star$ gives

$$0 \leq J(\widehat{\boldsymbol{x}}^{H,\star}) - J(\boldsymbol{x}^\star) \leq \frac{\lambda}{2} \kappa_A^2 L_r^2 \frac{N-H}{N}.$$

Thus,

$$C_T^{\mathrm{U}} = \frac{\lambda}{2} \kappa_A^2 L_r^2,$$

which proves the claim. □

B.3 Proof of Lemma 2

Proof. We follow the argument of Lemma 5, with one additional step: truncating a feasible long-horizon trajectory need not leave a feasible endpoint. The uniform Slater condition allows us to restore feasibility by convexly mixing the truncated trajectory with a strictly feasible one.

For $j + L \le N$, let

$$J_{j,L}^{C,\star}(x) := \min_{u_{j:j+L-1}} \left\{ \sum_{k=j}^{j+L-1} \frac{\Delta t}{2\lambda} \|u_k\|^2 - r(x_{j+L}) \right\}$$

subject to the affine dynamics, $x_j = x$, and the terminal constraint $h(x_{j+L}) \le 0$. We write

$$J_j^{C,\star}(x) := J_{j,N-j}^{C,\star}(x).$$

Recall the constants

$$\chi_h := \frac{L_h(\overline{J} - \underline{J})}{\gamma}, \qquad G_h := L_r + \chi_h,$$

where γ is the uniform Slater margin, $\overline{J}$ uniformly bounds the objective of the strictly feasible Slater trajectories, and $\underline{J}$ is a uniform lower bound on the relevant finite-horizon objectives. The latter exists by the quadratic control penalty and the Lipschitz terminal reward, as in the coercivity argument used previously.

Step 1: Local constrained horizon sensitivity.

Fix j, L, q with $j + L + q \le N$, and set

$$\tau := q\Delta t.$$

We show that

$$\left| J_{j,L+q}^{C,\star}(x) - J_{j,L}^{C,\star}(x) \right| \le C_{\mathrm{loc}}^{\mathrm{C}} \tau,$$

where

$$C_{\mathrm{loc}}^{\mathrm{C}} := \max\left\{ G_h M + \frac{\lambda G_h^2}{2}, \frac{M^2}{2\lambda} \right\}.$$

Let $u_{j:j+L+q-1}^{\star}$ be optimal for $J_{j,L+q}^{C,\star}(x)$, with trajectory $x_k^{\star}$, and let

$$E := \sum_{k=j+L}^{j+L+q-1} \Delta t \, \|u_k^{\star}\|^2$$

denote the energy in the omitted tail. Define

$$d := M\tau + \sqrt{\tau E}.$$

As in the proof of Lemma 5,

$$\|x_{j+L+q}^{\star} - x_{j+L}^{\star}\| \le d.$$

Although $x_{j+L+q}^{\star}$ is feasible, the truncated endpoint $x_{j+L}^{\star}$ need not be. Since $h(x_{j+L+q}^{\star}) \le 0$ and h is L_h-Lipschitz, each component satisfies

$$h_i(x_{j+L}^{\star}) \le L_h d.$$

Set

$$\delta := L_h d.$$

By Assumption 3, the L-step problem admits a strictly feasible Slater trajectory $(x_k^{\mathrm{s}}, u_k^{\mathrm{s}})$ from the same initial state x, satisfying

$$h(x_{j+L}^{\mathrm{s}}) \le -\gamma \mathbf{1}, \qquad J_{j,L}(x^{\mathrm{s}}, u^{\mathrm{s}}) \le \overline{J}.$$

Choose

$$\alpha := \frac{\delta}{\gamma + \delta}.$$

In particular,

$$0 \le \alpha \le \frac{\delta}{\gamma}.$$

Consider the convexly mixed controls

$$\widetilde{u}_k := (1-\alpha)u_k^\star + \alpha u_k^{\mathrm{s}}, \qquad k = j, \dots, j+L-1.$$

Because the dynamics are affine and both trajectories start from the same state, their corresponding states satisfy

$$\widetilde{x}_k = (1-\alpha)x_k^\star + \alpha x_k^{\mathrm{s}}.$$

By componentwise convexity of h,

$$\begin{aligned} h_i(\widetilde{x}_{j+L}) &\le (1-\alpha)h_i(x^\star_{j+L}) + \alpha h_i(x^{\mathrm{s}}_{j+L}) \\ &\le (1-\alpha)\delta - \alpha\gamma = 0. \end{aligned}$$

Thus the mixed trajectory is feasible for the L-step problem.

Since the dynamics are affine, the quadratic running cost is convex in the controls, and $-r$ is convex because r is concave. Hence the truncated objective is convex, so

$$J_{j,L}(\widetilde{x}, \widetilde{u}) \le (1-\alpha)J_{j,L}(x^\star, u^\star) + \alpha J_{j,L}(x^{\mathrm{s}}, u^{\mathrm{s}}).$$

Using $J_{j,L}(x^\star, u^\star) \ge \underline{J}$ and $J_{j,L}(x^{\mathrm{s}}, u^{\mathrm{s}}) \le \overline{J}$ gives

$$J_{j,L}(\widetilde{x}, \widetilde{u}) \le J_{j,L}(x^\star, u^\star) + \alpha(\overline{J} - \underline{J}).$$

Therefore,

$$J^{C,\star}_{j,L}(x) \le J_{j,L}(x^\star, u^\star) + \alpha(\overline{J} - \underline{J}).$$

The objective of the truncated prefix and that of the original $(L+q)$-step optimum differ by

$$\begin{aligned} J_{j,L}(x^\star, u^\star) - J^{C,\star}_{j,L+q}(x) &= r(x^\star_{j+L+q}) - r(x^\star_{j+L}) - \frac{E}{2\lambda} \\ &\le L_r d - \frac{E}{2\lambda}. \end{aligned}$$

Moreover,

$$\alpha(\overline{J} - \underline{J}) \le \frac{L_h(\overline{J} - \underline{J})}{\gamma} d = \chi_h d.$$

Combining the preceding inequalities,

$$J^{C,\star}_{j,L}(x) - J^{C,\star}_{j,L+q}(x) \le G_h d - \frac{E}{2\lambda}.$$

Substituting $d = M\tau + \sqrt{\tau E}$,

$$J^{C,\star}_{j,L}(x) - J^{C,\star}_{j,L+q}(x) \le G_h M\tau + G_h\sqrt{\tau E} - \frac{E}{2\lambda}.$$

Completing the square gives

$$G_h\sqrt{\tau E} - \frac{E}{2\lambda} \le \frac{\lambda G_h^2}{2}\tau,$$

and hence

$$J^{C,\star}_{j,L}(x) - J^{C,\star}_{j,L+q}(x) \le \left(G_h M + \frac{\lambda G_h^2}{2}\right)\tau.$$

For the reverse direction, let an optimal trajectory for $J^{C,\star}_{j,L}(x)$ terminate at $\bar{x}$. Since $h(\bar{x}) \le 0$, extend it for the next q steps using

$$u_k = -\big(A_k\bar{x} + c_k\big), \qquad k = j+L, \dots, j+L+q-1.$$

This keeps the state fixed at $\bar{x}$, so feasibility and the terminal reward are preserved. By the drift bound,

$$\|u_k\| \le M,$$

and therefore the extension costs at most

$$\frac{M^2}{2\lambda}\tau.$$

Thus

$$J^{C,\star}_{j,L+q}(x) - J^{C,\star}_{j,L}(x) \le \frac{M^2}{2\lambda}\tau.$$

Combining the two directions,

$$\left|J^{C,\star}_{j,L+q}(x) - J^{C,\star}_{j,L}(x)\right| \le C^{\mathrm{C}}_{\mathrm{loc}} q\Delta t.$$

Step 2: Accumulation along the constrained RH-H trajectory.

The remainder follows the telescoping argument in Lemma 5. We include the main steps for completeness.

Let

$$m := N - H, \qquad W^{\mathrm{C}}_j(x) := J^{C,\star}_{j,H}(x), \qquad j = 0, \dots, m.$$

At $j = m$, the horizon reaches the true terminal time, so

$$W^{\mathrm{C}}_m(x) = J^{C,\star}_m(x).$$

Let $x^{H,\star}_j$ and $u^{H,\star}_j$ denote the constrained RH-H closed-loop state and applied control. Bellman's principle for the H-step constrained problem gives

$$W^{\mathrm{C}}_j(x^{H,\star}_j) = \frac{\Delta t}{2\lambda}\|u^{H,\star}_j\|^2 + J^{C,\star}_{j+1,H-1}(x^{H,\star}_{j+1}).$$

Applying the local horizon estimate with $q = 1$,

$$W^{\mathrm{C}}_{j+1}(x^{H,\star}_{j+1}) - J^{C,\star}_{j+1,H-1}(x^{H,\star}_{j+1}) \le C^{\mathrm{C}}_{\mathrm{loc}}\Delta t.$$

Consequently,

$$\frac{\Delta t}{2\lambda}\|u^{H,\star}_j\|^2 \le W^{\mathrm{C}}_j(x^{H,\star}_j) - W^{\mathrm{C}}_{j+1}(x^{H,\star}_{j+1}) + C^{\mathrm{C}}_{\mathrm{loc}}\Delta t.$$

Summing from $j = 0$ to $m - 1$ yields

$$\sum_{j=0}^{m-1} \frac{\Delta t}{2\lambda}\|u^{H,\star}_j\|^2 \le W^{\mathrm{C}}_0(x_0) - W^{\mathrm{C}}_m(x^{H,\star}_m) + mC^{\mathrm{C}}_{\mathrm{loc}}\Delta t.$$

From time $m = N - H$ onward, the remaining horizon is at most H, so RH-H coincides with full remaining-horizon re-optimization. Its remaining cost is therefore

$$J^{C,\star}_m(x^{H,\star}_m) = W^{\mathrm{C}}_m(x^{H,\star}_m).$$

Hence

$$J(\boldsymbol{x}^{H,\star}) \le W^{\mathrm{C}}_0(x_0) + mC^{\mathrm{C}}_{\mathrm{loc}}\Delta t.$$

Finally,

$$W^{\mathrm{C}}_0(x_0) = J^{C,\star}_{0,H}(x_0), \qquad J(\boldsymbol{x}^\star) = J^{C,\star}_{0,N}(x_0).$$

Applying the local constrained horizon estimate once more with $L = H$ and $q = m$ gives

$$W^{\mathrm{C}}_0(x_0) - J(\boldsymbol{x}^\star) \le C^{\mathrm{C}}_{\mathrm{loc}} m\Delta t.$$

Therefore,

$$J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^\star) \le 2C^{\mathrm{C}}_{\mathrm{loc}} m\Delta t.$$

Since $\boldsymbol{x}^\star$ is globally optimal among feasible trajectories,

$$J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^\star) \geq 0.$$

Using $m = N - H$ and $\Delta t = 1/N$,

$$0 \leq J(\boldsymbol{x}^{H,\star}) - J(\boldsymbol{x}^\star) \leq C_H^{\mathrm{C}} \frac{N-H}{N},$$

where

$$C_H^{\mathrm{C}} := 2C_{\mathrm{loc}}^{\mathrm{C}} = 2\max\left\{ G_h M + \frac{\lambda G_h^2}{2}, \frac{M^2}{2\lambda} \right\}.$$

This proves the claim. □

B.4 Proof of Lemma 3

Proof. The proof combines the no-accumulation argument of Lemma 6 with the Slater-mixing argument used in Lemma 2. The only new issue is that truncating the control horizon and rolling forward with zero control may violate the terminal constraint; the uniform Slater condition is used to repair this candidate without changing the order of the bound.

For $j + L \leq N$, define the constrained rollout value

$$\widehat{J}_{j,L}^{C,\star}(x) := \min_{u_{j:j+L-1}} \left\{ \sum_{k=j}^{j+L-1} \frac{\Delta t}{2\lambda} \|u_k\|^2 - r(\widehat{x}_N) \right\},$$

where the state follows the controlled affine dynamics for $k = j, \ldots, j+L-1$, is subsequently propagated with $u_k = 0$ until time N, and satisfies

$$h(\widehat{x}_N) \leq 0.$$

As before, $J_j^{C,\star}(x)$ denotes the constrained full remaining-horizon value.

Recall

$$\chi_h := \frac{L_h(\overline{J} - \underline{J})}{\gamma}, \qquad G_h := L_r + \chi_h,$$

where $\underline{J}$ is a uniform lower bound on the relevant rollout objectives and $\overline{J}$ bounds the objective of the corresponding strictly feasible Slater trajectories.

Step 1: Terminal rollout does not accumulate horizon errors.

The nesting argument from Lemma 6 remains valid in the constrained case. Indeed, any feasible L-step rollout control remains feasible for the $(L+1)$-step problem by setting the additional control to zero; the resulting terminal state is unchanged. Hence

$$J_j^{C,\star}(x) \leq \widehat{J}_{j,L+1}^{C,\star}(x) \leq \widehat{J}_{j,L}^{C,\star}(x).$$

Let

$$m := N - H, \qquad \widehat{W}_j^{\mathrm{C}}(x) := \widehat{J}_{j,H}^{C,\star}(x), \qquad j = 0, \ldots, m.$$

At $j = m$, the control horizon reaches the true terminal time, so

$$\widehat{W}_m^{\mathrm{C}}(x) = J_m^{C,\star}(x).$$

Let $\widehat{x}_j^{H,\star}$ and $\widehat{u}_j^{H,\star}$ denote the constrained RH-T closed-loop state and applied control. Optimality gives

$$\widehat{W}_j^{\mathrm{C}}(\widehat{x}_j^{H,\star}) = \frac{\Delta t}{2\lambda} \|\widehat{u}_j^{H,\star}\|^2 + \widehat{J}_{j+1,H-1}^{C,\star}(\widehat{x}_{j+1}^{H,\star}).$$

By nestedness,

$$\widehat{W}_{j+1}^{\mathrm{C}}(\widehat{x}_{j+1}^{H,\star}) \leq \widehat{J}_{j+1,H-1}^{C,\star}(\widehat{x}_{j+1}^{H,\star}),$$

and therefore

$$\frac{\Delta t}{2\lambda}\|\widehat{u}_j^{H,\star}\|^2 \le \widehat{W}_j^{\mathrm{C}}(\widehat{x}_j^{H,\star}) - \widehat{W}_{j+1}^{\mathrm{C}}(\widehat{x}_{j+1}^{H,\star}).$$

Summing over $j = 0, \dots, m-1$ and using $\widehat{W}_m^{\mathrm{C}} = J_m^{C,\star}$ yields

$$J(\widehat{\boldsymbol{x}}^{H,\star}) \le \widehat{J}_{0,H}^{C,\star}(x_0).$$

Consequently,

$$0 \le J(\widehat{\boldsymbol{x}}^{H,\star}) - J(\boldsymbol{x}^\star) \le \widehat{J}_{0,H}^{C,\star}(x_0) - J_0^{C,\star}(x_0).$$

It therefore remains only to bound this initial constrained rollout gap.

Step 2: Bounding the constrained rollout gap.

Let $u_0^\star, \dots, u_{N-1}^\star$ be the controls associated with the full constrained optimum $\boldsymbol{x}^\star$, and define

$$\tau_H := (N-H)\Delta t = \frac{N-H}{N},$$

together with its omitted-tail energy

$$E_H := \sum_{k=H}^{N-1} \Delta t\, \|u_k^\star\|^2.$$

As in Lemma 6, retain the first H optimal controls and set the remaining controls to zero:

$$\widetilde{u}_k = \begin{cases} u_k^\star, & k = 0, \dots, H-1, \\ 0, & k = H, \dots, N-1. \end{cases}$$

Let $\widetilde{x}_N$ denote the resulting terminal state. The affine perturbation argument of Lemma 6 gives

$$\|x_N^\star - \widetilde{x}_N\| \le \kappa_A \sqrt{\tau_H E_H}.$$

Since the full-horizon optimum is feasible, $h(x_N^\star) \le 0$, and h is L_h-Lipschitz. Thus each terminal constraint violation of the truncated rollout is bounded by

$$h_i(\widetilde{x}_N) \le \delta_H, \qquad \delta_H := L_h \kappa_A \sqrt{\tau_H E_H}.$$

The zero-tail rollout need not itself be feasible. By Assumption 3, however, the H-step rollout problem admits a strictly feasible trajectory $(x^{\mathrm{s}}, u^{\mathrm{s}})$ from the same initial state satisfying

$$h(x_N^{\mathrm{s}}) \le -\gamma \mathbf{1}, \qquad \widehat{J}_{0,H}(x^{\mathrm{s}}, u^{\mathrm{s}}) \le \overline{J}.$$

Set

$$\alpha := \frac{\delta_H}{\gamma + \delta_H} \le \frac{\delta_H}{\gamma},$$

and form the mixed rollout controls

$$u_k^\alpha := (1-\alpha)\widetilde{u}_k + \alpha u_k^{\mathrm{s}}, \qquad k = 0, \dots, H-1,$$

with zero control thereafter.

Because the dynamics are affine, the resulting terminal state satisfies

$$x_N^\alpha = (1-\alpha)\widetilde{x}_N + \alpha x_N^{\mathrm{s}}.$$

By componentwise convexity of h,

$$\begin{aligned} h_i(x_N^\alpha) &\le (1-\alpha)h_i(\widetilde{x}_N) + \alpha h_i(x_N^{\mathrm{s}}) \\ &\le (1-\alpha)\delta_H - \alpha\gamma = 0. \end{aligned}$$

Hence the mixed rollout is feasible.

The rollout objective is convex in the active controls, since the dynamics are affine, the running cost is quadratic, and $-r$ is convex. Therefore,

$$\widehat{J}_{0,H}(u^\alpha) \le (1-\alpha)\widehat{J}_{0,H}(\widetilde{u}) + \alpha \widehat{J}_{0,H}(u^{\mathrm{s}}).$$

Using the uniform lower and upper bounds $\widehat{J}_{0,H}(\widetilde{u}) \geq \underline{J}$ and $\widehat{J}_{0,H}(u^{\mathrm{s}}) \leq \overline{J}$,

$$\widehat{J}_{0,H}(u^{\alpha}) \leq \widehat{J}_{0,H}(\widetilde{u}) + \alpha(\overline{J} - \underline{J}).$$

Moreover,

$$\alpha(\overline{J} - \underline{J}) \leq \frac{L_h(\overline{J} - \underline{J})}{\gamma} \kappa_A \sqrt{\tau_H E_H} = \chi_h \kappa_A \sqrt{\tau_H E_H}.$$

Since u^{α} is feasible for the constrained rollout problem,

$$\widehat{J}^{C,\star}_{0,H}(x_0) \leq \widehat{J}_{0,H}(u^{\alpha}).$$

On the other hand, comparing the zero-tail trajectory with the full optimum gives

$$\widehat{J}_{0,H}(\widetilde{u}) - J^{C,\star}_0(x_0) = -\frac{E_H}{2\lambda} + r(x^{\star}_N) - r(\widetilde{x}_N).$$

Using the L_r-Lipschitz continuity of r,

$$\widehat{J}_{0,H}(\widetilde{u}) - J^{C,\star}_0(x_0) \leq L_r \kappa_A \sqrt{\tau_H E_H} - \frac{E_H}{2\lambda}.$$

Combining the reward and feasibility-repair contributions,

$$\widehat{J}^{C,\star}_{0,H}(x_0) - J^{C,\star}_0(x_0) \leq G_h \kappa_A \sqrt{\tau_H E_H} - \frac{E_H}{2\lambda}.$$

Completing the square in $\sqrt{E_H}$ gives

$$G_h \kappa_A \sqrt{\tau_H E_H} - \frac{E_H}{2\lambda} \leq \frac{\lambda}{2} \kappa_A^2 G_h^2 \tau_H.$$

Therefore,

$$0 \leq \widehat{J}^{C,\star}_{0,H}(x_0) - J^{C,\star}_0(x_0) \leq \frac{\lambda}{2} \kappa_A^2 G_h^2 \frac{N-H}{N}.$$

Combining this estimate with Step 1 yields

$$0 \leq J(\widehat{\boldsymbol{x}}^{H,\star}) - J(\boldsymbol{x}^{\star}) \leq \frac{\lambda}{2} \kappa_A^2 G_h^2 \frac{N-H}{N}.$$

Thus

$$C_T^{\mathrm{C}} = \frac{\lambda}{2} \kappa_A^2 G_h^2,$$

which proves the claim. □

B.5 Proof of Lemma 4

We first collect two properties of each frozen RH problem that are used in the proof. Fix a physical index j and current state $\bar{x}_j$. Let $H_j = \min\{H, N-j\}$ and write $\mathbf{x}_j = (x_{j+1}, \ldots, x_{j+H_j})$. Under Assumption 2, the stacked residuals satisfy

$$\mathrm{col}(e_j, \ldots, e_{j+H_j-1}) = B_j \mathbf{x}_j - a_j - S_j \bar{x}_j,$$

where

$$B_j = \begin{bmatrix} I_d & 0 & \cdots & 0 \\ -(I_d + \Delta t A_{j+1}) & I_d & \ddots & \vdots \\ 0 & \ddots & \ddots & 0 \\ 0 & \cdots & -(I_d + \Delta t A_{j+H_j-1}) & I_d \end{bmatrix}, \qquad S_j = \begin{bmatrix} I_d + \Delta t A_j \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

and $a_j = \Delta t \ \mathrm{col}(c_j, \ldots, c_{j+H_j-1})$. Recall

$$\sigma_j := \frac{\sigma^2_{\min}(B_j)}{\lambda \Delta t} > 0.$$

For RH-T, let Q_j denote the linear part of the terminal map Φ_j; for RH-H, $Q_j = I$. Thus

$$L_{g,j} := L_h \|Q_j\|, \qquad L_{d,j} := \frac{L^2_{g,j}}{\sigma_j}.$$

Lemma 7 (Strong convexity and well-posedness)**.** *Under Assumptions 2–3, $\varphi_j(\cdot;\bar{x}_j)$ and $\mathcal{L}_j(\cdot,\mu;\bar{x}_j)$ are σ_j-strongly convex for every $\mu \geq 0$. Moreover, the primal response*

$$\mathbf{x}_j(\mu) := \operatorname*{argmin}_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j, \mu; \bar{x}_j)$$

exists and is unique, and the constrained local problem has a unique optimizer $\mathbf{x}_j^\star$.

Proof. Since B_j is block lower triangular with identity diagonal blocks, it is nonsingular. The Hessian of the quadratic path cost is

$$\frac{1}{\lambda \Delta t} B_j^\top B_j \succeq \sigma_j I.$$

Moreover, Φ_j is affine, r is concave, and each component of h is convex. Hence $-r \circ \Phi_j$ and $g_{j,i} = h_i \circ \Phi_j$ are convex. Therefore φ_j is σ_j-strongly convex, and so is $\mathcal{L}_j = \varphi_j + \mu^\top g_j$ for every $\mu \geq 0$.

It remains to show attainment. Let $X = \mathbf{x}_j$, $\zeta_j = a_j + S_j \bar{x}_j$, and write

$$q_j(X) := \frac{1}{2\lambda \Delta t} \|B_j X - \zeta_j\|^2, \qquad \psi_\mu(X) := -r(\Phi_j(x_{j+H_j})) + \mu^\top g_j(X).$$

The finite-valued convex function ψ_μ admits an affine lower bound: for any fixed $\widehat{X}$ and $s_\mu \in \partial \psi_\mu(\widehat{X})$,

$$\psi_\mu(X) \geq \psi_\mu(\widehat{X}) + s_\mu^\top (X - \widehat{X}).$$

Also,

$$\begin{aligned} q_j(X) &\geq \frac{1}{4\lambda \Delta t} \|B_j X\|^2 - \frac{1}{2\lambda \Delta t} \|\zeta_j\|^2 \\ &\geq \frac{\sigma_{\min}^2(B_j)}{4\lambda \Delta t} \|X\|^2 - \frac{1}{2\lambda \Delta t} \|\zeta_j\|^2. \end{aligned}$$

Thus

$$\mathcal{L}_j(X, \mu; \bar{x}_j) \geq \frac{\sigma_{\min}^2(B_j)}{4\lambda \Delta t} \|X\|^2 - \|s_\mu\| \, \|X\| - C$$

for a finite constant C, and hence the Lagrangian is coercive. Therefore its minimum is attained; strong convexity makes the minimizer unique.

Finally, Assumption 3 gives a nonempty convex feasible set for the frozen RH problem. Since its objective is strongly convex and coercive, the constrained minimizer exists and is unique. □

Lemma 8 (Dual regularity)**.** *Under Assumptions 2–3, strong duality holds, $\mathcal{M}_j^\star$ is nonempty and compact, and d_j is differentiable with*

$$\nabla d_j(\mu) = g_j(\mathbf{x}_j(\mu)).$$

Furthermore,

$$\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\| \leq \frac{L_{g,j}}{\sigma_j} \|\mu - \nu\|,$$

and

$$\|\nabla d_j(\mu) - \nabla d_j(\nu)\| \leq L_{d,j} \|\mu - \nu\|.$$

Proof. By Assumption 3, the local convex problem satisfies Slater's condition. Standard convex duality therefore gives strong duality and existence of an optimal multiplier.

To see boundedness, let $\widetilde{\mathbf{x}}_j$ be a Slater point with $g_j(\widetilde{\mathbf{x}}_j) \leq -\gamma \mathbf{1}$. For any $\mu^\star \in \mathcal{M}_j^\star$,

$$\begin{aligned} p_j^\star = d_j(\mu^\star) &\leq \mathcal{L}_j(\widetilde{\mathbf{x}}_j, \mu^\star; \bar{x}_j) \\ &\leq \varphi_j(\widetilde{\mathbf{x}}_j; \bar{x}_j) - \gamma \|\mu^\star\|_1. \end{aligned}$$

Hence

$$\|\mu^\star\|_1 \leq \frac{\varphi_j(\widetilde{\mathbf{x}}_j; \bar{x}_j) - p_j^\star}{\gamma}.$$

Thus $\mathcal{M}_j^\star$ is bounded; closedness follows from continuity of d_j, proved below, so it is compact.

We next establish Lipschitzness of the primal response. Strong convexity and optimality of $\mathbf{x}_j(\mu)$ and $\mathbf{x}_j(\nu)$ give

$$\mathcal{L}_j(\mathbf{x}_j(\nu), \mu) \geq \mathcal{L}_j(\mathbf{x}_j(\mu), \mu) + \frac{\sigma_j}{2}\|\mathbf{x}_j(\nu) - \mathbf{x}_j(\mu)\|^2$$

and

$$\mathcal{L}_j(\mathbf{x}_j(\mu), \nu) \geq \mathcal{L}_j(\mathbf{x}_j(\nu), \nu) + \frac{\sigma_j}{2}\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\|^2.$$

Adding these inequalities and cancelling the common objective terms yields

$$\sigma_j\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\|^2 \leq (\mu - \nu)^\top \big(g_j(\mathbf{x}_j(\nu)) - g_j(\mathbf{x}_j(\mu))\big).$$

Using Cauchy–Schwarz and the $L_{g,j}$-Lipschitzness of g_j,

$$\sigma_j\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\|^2 \leq L_{g,j}\|\mu - \nu\|\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\|,$$

which proves

$$\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\| \leq \frac{L_{g,j}}{\sigma_j}\|\mu - \nu\|.$$

To identify the dual gradient, optimality of the two primal responses gives

$$d_j(\nu) \leq d_j(\mu) + (\nu - \mu)^\top g_j(\mathbf{x}_j(\mu))$$

and

$$d_j(\mu) \leq d_j(\nu) + (\mu - \nu)^\top g_j(\mathbf{x}_j(\nu)).$$

Therefore

$$\begin{aligned}\big|d_j(\nu) - d_j(\mu) - (\nu - \mu)^\top g_j(\mathbf{x}_j(\mu))\big| &\leq \|\nu - \mu\|\,\|g_j(\mathbf{x}_j(\nu)) - g_j(\mathbf{x}_j(\mu))\| \\ &\leq \frac{L_{g,j}^2}{\sigma_j}\|\nu - \mu\|^2.\end{aligned}$$

Hence $\nabla d_j(\mu) = g_j(\mathbf{x}_j(\mu))$. Finally,

$$\begin{aligned}\|\nabla d_j(\mu) - \nabla d_j(\nu)\| &\leq L_{g,j}\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\| \\ &\leq \frac{L_{g,j}^2}{\sigma_j}\|\mu - \nu\| = L_{d,j}\|\mu - \nu\|.\end{aligned}$$

□

Proof of Lemma 4. Fix j and suppress the dependence on $\bar{x}_j$. From Lemma 8,

$$\begin{aligned}\langle \nabla d_j(\mu) - \nabla d_j(\nu), \mu - \nu\rangle &= (\mu - \nu)^\top \big(g_j(\mathbf{x}_j(\mu)) - g_j(\mathbf{x}_j(\nu))\big) \\ &\leq -\sigma_j\|\mathbf{x}_j(\mu) - \mathbf{x}_j(\nu)\|^2 \\ &\leq -\frac{1}{L_{d,j}}\|\nabla d_j(\mu) - \nabla d_j(\nu)\|^2.\end{aligned}$$

Let

$$T(\mu) := \mathrm{Proj}_{\mathbb{R}^m_+}\big(\mu + \eta_\mu \nabla d_j(\mu)\big).$$

A point z maximizes d_j over $\mathbb{R}^m_+$ if and only if $T(z) = z$. For $z \in \mathcal{M}_j^\star$, nonexpansiveness of the projection and the preceding inequality imply

$$\begin{aligned}\|\mu_j^{(s+1)} - z\|^2 &\leq \|\mu_j^{(s)} - z + \eta_\mu(\nabla d_j(\mu_j^{(s)}) - \nabla d_j(z))\|^2 \\ &\leq \|\mu_j^{(s)} - z\|^2 - \eta_\mu\left(\frac{2}{L_{d,j}} - \eta_\mu\right)\|\nabla d_j(\mu_j^{(s)}) - \nabla d_j(z)\|^2.\end{aligned}$$

Thus, whenever $0 < \eta_\mu < 2/L_{d,j}$, the sequence is Fejér monotone with respect to $\mathcal{M}_j^\star$ and therefore bounded.

Using firm nonexpansiveness of the projection gives more precisely

$$\|\mu_j^{(s+1)} - z\|^2 + \|\mu_j^{(s)} - \mu_j^{(s+1)} + \eta_\mu(\nabla d_j(\mu_j^{(s)}) - \nabla d_j(z))\|^2 \\ \leq \|\mu_j^{(s)} - z\|^2 - \eta_\mu \left(\frac{2}{L_{d,j}} - \eta_\mu\right) \|\nabla d_j(\mu_j^{(s)}) - \nabla d_j(z)\|^2.$$

Summing over s shows both squared terms on the left are summable. Consequently,

$$\|\mu_j^{(s+1)} - \mu_j^{(s)}\| \to 0.$$

Since the sequence is bounded, it admits a convergent subsequence $\mu_j^{(s_k)} \to \bar{\mu}_j$. Passing to the limit in $\mu_j^{(s+1)} = T(\mu_j^{(s)})$ gives $T(\bar{\mu}_j) = \bar{\mu}_j$; hence $\bar{\mu}_j \in \mathcal{M}_j^\star$. Fejér monotonicity with respect to $\bar{\mu}_j$ then implies convergence of the full sequence:

$$\mu_j^{(s)} \to \bar{\mu}_j.$$

Lemma 8 gives continuity of the primal response, and Lemma 7 gives uniqueness. Therefore

$$\mathbf{x}_j^{(s)} = \mathbf{x}_j(\mu_j^{(s)}) \longrightarrow \mathbf{x}_j(\bar{\mu}_j) = \mathbf{x}_j^\star.$$

Finite-iteration bounds. Assume now $0 < \eta_\mu \leq 1/L_{d,j}$. The projection optimality condition gives, for any $z \geq 0$,

$$\left\langle \nabla d_j(\mu_j^{(k)}), z - \mu_j^{(k+1)} \right\rangle \leq \frac{\|z - \mu_j^{(k)}\|^2 - \|z - \mu_j^{(k+1)}\|^2 - \|\mu_j^{(k+1)} - \mu_j^{(k)}\|^2}{2\eta_\mu}.$$

By concavity and $L_{d,j}$-smoothness,

$$d_j(z) - d_j(\mu_j^{(k+1)}) \leq \left\langle \nabla d_j(\mu_j^{(k)}), z - \mu_j^{(k+1)} \right\rangle \\ + \frac{L_{d,j}}{2}\|\mu_j^{(k+1)} - \mu_j^{(k)}\|^2.$$

Hence

$$d_j(z) - d_j(\mu_j^{(k+1)}) \leq \frac{\|z - \mu_j^{(k)}\|^2 - \|z - \mu_j^{(k+1)}\|^2}{2\eta_\mu} \\ - \left(\frac{1}{2\eta_\mu} - \frac{L_{d,j}}{2}\right) \|\mu_j^{(k+1)} - \mu_j^{(k)}\|^2.$$

Taking $z = \mu_j^{(k)}$ shows that $d_j(\mu_j^{(k)})$ is nondecreasing. Let $\mu_j^\dagger \in \mathcal{M}_j^\star$ satisfy

$$\|\mu_j^{(0)} - \mu_j^\dagger\| = R_j := \operatorname{dist}(\mu_j^{(0)}, \mathcal{M}_j^\star).$$

Summing the preceding inequality with $z = \mu_j^\dagger$ gives

$$\sum_{k=0}^{s-1} \left(p_j^\star - d_j(\mu_j^{(k+1)})\right) \leq \frac{R_j^2}{2\eta_\mu}.$$

Since these gaps are nonincreasing,

$$0 \leq p_j^\star - d_j(\mu_j^{(s)}) \leq \frac{R_j^2}{2\eta_\mu s}.$$

Finally, strong convexity of $\mathcal{L}_j(\cdot, \mu)$ and feasibility of $\mathbf{x}_j^\star$ imply

$$\frac{\sigma_j}{2}\|\mathbf{x}_j(\mu) - \mathbf{x}_j^\star\|^2 \leq \mathcal{L}_j(\mathbf{x}_j^\star, \mu) - d_j(\mu) \\ \leq p_j^\star - d_j(\mu).$$

Taking $\mu = \mu_j^{(s)}$ yields

$$\|\mathbf{x}_j(\mu_j^{(s)}) - \mathbf{x}_j^\star\| \le \frac{R_j}{\sqrt{\sigma_j \eta_\mu s}}.$$

Since $[\,\cdot\,]_+$ is nonexpansive and $g_j(\mathbf{x}_j^\star) \le 0$,

$$\begin{aligned}\|[g_j(\mathbf{x}_j(\mu_j^{(s)}))]_+\| &\le \|g_j(\mathbf{x}_j(\mu_j^{(s)})) - g_j(\mathbf{x}_j^\star)\| \\ &\le L_{g,j}\|\mathbf{x}_j(\mu_j^{(s)}) - \mathbf{x}_j^\star\| \\ &\le \frac{L_{g,j} R_j}{\sqrt{\sigma_j \eta_\mu s}}.\end{aligned}$$

This proves the lemma. □

B.6 Proof of Theorem 1

We next propagate the local optimization error through the receding-horizon closed loop. For a current state ξ, let $\mathbf{x}_j^\star(\xi)$ denote the exact optimizer of the jth RH problem and define

$$\pi_j(\xi) := [\mathbf{x}_j^\star(\xi)]_1.$$

Thus, the exact RH trajectory satisfies $x_0^{\mathrm{RH}} = x_0$ and $x_{j+1}^{\mathrm{RH}} = \pi_j(x_j^{\mathrm{RH}})$.

Lemma 9 (Lipschitz exact RH policy)**.** *Under Assumptions 2–3,*

$$\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\| \le \kappa_j \|\xi - \zeta\|, \qquad \kappa_j := \frac{\|B_j^\top S_j\|}{\lambda \Delta t\, \sigma_j}.$$

Consequently,

$$\|\pi_j(\xi) - \pi_j(\zeta)\| \le \kappa_j \|\xi - \zeta\|.$$

Proof. The constraint $g_j(\mathbf{x}_j) = h(\Phi_j(x_{j+H_j}))$ does not depend explicitly on the current state. Hence the feasible set is the same for initial states ξ and ζ.

Since $\varphi_j(\cdot;\xi)$ is σ_j-strongly convex and $\mathbf{x}_j^\star(\xi)$ minimizes it over this common convex feasible set,

$$\varphi_j(\mathbf{x}_j^\star(\zeta);\xi) \ge \varphi_j(\mathbf{x}_j^\star(\xi);\xi) + \frac{\sigma_j}{2}\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\|^2.$$

Exchanging ξ and ζ gives

$$\varphi_j(\mathbf{x}_j^\star(\xi);\zeta) \ge \varphi_j(\mathbf{x}_j^\star(\zeta);\zeta) + \frac{\sigma_j}{2}\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\|^2.$$

Adding the two inequalities,

$$\begin{aligned}\sigma_j\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\|^2 \le{}& \varphi_j(\mathbf{x}_j^\star(\xi);\zeta) - \varphi_j(\mathbf{x}_j^\star(\xi);\xi) \\ &+ \varphi_j(\mathbf{x}_j^\star(\zeta);\xi) - \varphi_j(\mathbf{x}_j^\star(\zeta);\zeta).\end{aligned}$$

The terminal-reward terms cancel, and expanding the two quadratic residual costs gives

$$\begin{aligned}\sigma_j\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\|^2 &\le \frac{1}{\lambda\Delta t}\left\langle B_j^\top S_j(\xi - \zeta), \mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\right\rangle \\ &\le \frac{\|B_j^\top S_j\|}{\lambda\Delta t}\|\xi - \zeta\|\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\|.\end{aligned}$$

Cancelling the final factor proves

$$\|\mathbf{x}_j^\star(\xi) - \mathbf{x}_j^\star(\zeta)\| \le \kappa_j\|\xi - \zeta\|.$$

Since π_j extracts the first block of the optimized path and this projection has operator norm one, the same bound holds for π_j. □

Lemma 10 (Propagation of local solve errors)**.** *Suppose the implemented trajectory satisfies*

$$\|x_{j+1} - \pi_j(x_j)\| \le \epsilon_j, \qquad j = 0, \dots, N-1.$$

Then, for every $p = 1, \dots, N$*,*

$$\|x_p - x_p^{\mathrm{RH}}\| \le \sum_{j=0}^{p-1} \left(\prod_{\ell=j+1}^{p-1} \kappa_\ell \right) \epsilon_j,$$

where an empty product equals one. Moreover, if $h(\pi_{N-1}(x_{N-1})) \le 0$*, then*

$$\|[h(x_N)]_+\| \le L_h \epsilon_{N-1}.$$

Proof. Since $x_0 = x_0^{\mathrm{RH}}$, the claim is immediate for $p = 1$:

$$\|x_1 - x_1^{\mathrm{RH}}\| = \|x_1 - \pi_0(x_0)\| \le \epsilon_0.$$

Assume the bound holds at time p. Then

$$\begin{aligned}
\|x_{p+1} - x_{p+1}^{\mathrm{RH}}\| &\le \|x_{p+1} - \pi_p(x_p)\| + \|\pi_p(x_p) - \pi_p(x_p^{\mathrm{RH}})\| \\
&\le \epsilon_p + \kappa_p \|x_p - x_p^{\mathrm{RH}}\| \\
&\le \epsilon_p + \sum_{j=0}^{p-1} \left(\prod_{\ell=j+1}^{p} \kappa_\ell \right) \epsilon_j \\
&= \sum_{j=0}^{p} \left(\prod_{\ell=j+1}^{p} \kappa_\ell \right) \epsilon_j,
\end{aligned}$$

which proves the result by induction.

For terminal feasibility, since $h(\pi_{N-1}(x_{N-1})) \le 0$ and h is L_h-Lipschitz,

$$\begin{aligned}
\|[h(x_N)]_+\| &= \|[h(x_N)]_+ - [h(\pi_{N-1}(x_{N-1}))]_+\| \\
&\le \|h(x_N) - h(\pi_{N-1}(x_{N-1}))\| \\
&\le L_h \|x_N - \pi_{N-1}(x_{N-1})\| \\
&\le L_h \epsilon_{N-1}.
\end{aligned}$$

□

Proof of Theorem 1. At physical window j, Lemma 4 with $s = K$ gives

$$\|\mathbf{x}_j^{(K)} - \mathbf{x}_j^\star(x_j)\| \le \frac{R_j}{\sqrt{\sigma_j \eta_\mu K}},$$

where

$$R_j = \operatorname{dist}\big(\mu_j^{(0)}, \mathcal{M}_j^\star(x_j)\big).$$

Since the implemented next state is the first block of $\mathbf{x}_j^{(K)}$, while $\pi_j(x_j)$ is the first block of $\mathbf{x}_j^\star(x_j)$,

$$\|x_{j+1} - \pi_j(x_j)\| \le \frac{R_j}{\sqrt{\sigma_j \eta_\mu K}} =: \epsilon_j.$$

Define

$$\bar{R} := \max_j R_j, \qquad \underline{\sigma} := \min_j \sigma_j, \qquad \bar{\kappa} := \max_j \kappa_j.$$

Lemma 10 therefore gives

$$\begin{aligned}
\|x_p - x_p^{\mathrm{RH}}\| &\le \sum_{j=0}^{p-1} \left(\prod_{\ell=j+1}^{p-1} \kappa_\ell \right) \frac{R_j}{\sqrt{\sigma_j \eta_\mu K}} \\
&\le \frac{\bar{R}}{\sqrt{\underline{\sigma}\, \eta_\mu K}} \sum_{j=0}^{p-1} \bar{\kappa}^{\,p-1-j} \\
&= \frac{\bar{R}\, \Gamma_p(\bar{\kappa})}{\sqrt{\underline{\sigma}\, \eta_\mu K}},
\end{aligned}$$

where

$$\Gamma_p(\bar{\kappa}) := \sum_{q=0}^{p-1} \bar{\kappa}^q.$$

Finally, at the last physical window $j = N - 1$, we have $H_{N-1} = 1$. Hence $\pi_{N-1}(x_{N-1})$ is the terminal state of the exact feasible local solution, so

$$h(\pi_{N-1}(x_{N-1})) \leq 0.$$

Applying Lemma 10,

$$\begin{aligned} \|[h(x_N)]_+\| &\leq L_h \epsilon_{N-1} \\ &\leq \frac{L_h \bar{R}}{\sqrt{\underline{\sigma}\, \eta_\mu K}}. \end{aligned}$$

This proves the theorem. □

C Additional Experimental Details and Results

C.1 Implementation Details

C.1.1 Low-dimensional experiment implementation details

Model and sampler. The pretrained model is a flow map on the two-moons density, parameterised as $X_\theta(s,t,x) = x + (t-s)\,\sigma_d\,\phi_\theta\big(s,\ t,\ x/\sigma_d\big)$ with ϕ_θ an MLP of three hidden layers of 256 GELU units, input $(s,t,x/\sigma_d) \in \mathbb{R}^4$ and output in $\mathbb{R}^2$. We use the EMA weights at 15k training steps. The data are `sklearn`'s `make_moons` with noise 0.05, standardised per coordinate and then scaled by 0.75; $\sigma_d = 0.750$ is the standard deviation of that training set, and the base law is $p_0 = \mathcal{N}(0, \sigma_d^2 I)$. Unlike the image experiment, time runs *forward*, $t_j = j/N$ with $N = 16$ uniform Euler steps and $\Delta t = 1/16$; Every run uses the same batch of 512 particles drawn from the same seed, shared across all methods and all λ, so each frontier is computed on identical initial conditions.

Rewards. Two terminal rewards are used, both bounded in $(0, 1]$ so that reward gains are directly comparable across methods. The *upper-moon* reward is an RBF envelope over one mode, $r(x) = \exp\big(-\min_j \|x - x^{(j)}\|^2/2\sigma^2\big)$ with $\sigma = 0.35$ and 400 reference points $x^{(j)}$ drawn from the upper moon; it is ≈ 1 on that mode and decays off it. The uncontrolled sampler already attains $\bar{r} = 0.589$, so the attainable headroom is small. The *off-manifold* reward is a single Gaussian bump $r(x) = \exp\big(-\|x - p\|^2/2\sigma^2\big)$ with $p = (-0.1,\ 1.75)$ and $\sigma = 0.5$. Its optimum lies off the data manifold: the upper moon's apex $(-0.36,\ 1.41)$ scores only 0.69, and the uncontrolled sampler attains $\bar{r} = 0.062$. Reaching high reward therefore requires leaving the support of the learned density, which is what separates the methods.

Control methods. All three methods share the sampler, the initial batch and the reward, and no constraints are active, so ConicFlow again reduces to $K = 1$ in Algorithm 1. Two departures from the image configuration are specific to this setting. First, the window subproblem is *not* warm-started across windows: each window is initialised from a fresh uncontrolled rollout of its own H steps, which isolates the effect of the horizon from that of solver warm-starting. Second, we report two choices of the terminal map Φ_j in (RH-T): *endpoint*, $\Phi_j = X_{j+H,N}$, which predicts $t = 1$ from the window's terminal state through one flow-map call (the image-experiment choice), and *horizon*, $\Phi_j = \mathrm{id}$, which evaluates the reward directly at the window's terminal state x_{j+H} and makes no network call beyond the rollout.

HardFlow is run in its unconstrained form, so its terminal subproblem reduces to proximal reward steering, $\widehat{x}_N^\star = \arg\min_x -r(x) + \frac{\alpha_t^2}{2\Delta t\,\lambda}\|x - \bar{x}_N\|^2$, solved by the augmented-Lagrangian routine with $\rho_0 = 1$ and $\rho \times 2$ per outer round, warm-started across sampling steps. Its released code parameterises the proximal weight as $w = \lambda_{\mathrm{oc}}\alpha_t^2/(2\Delta t)$, i.e. with the opposite sense: larger λ_{oc} holds the sample *closer* to the pretrained path. We therefore set $\lambda_{\mathrm{oc}} = 1/\lambda$ so that a single λ axis means the same thing for all three methods — larger λ is a weaker control penalty and more reward — which is what makes the frontiers superimposable. FMRG is run in both variants, J (differentiating through $X_{j,N}$) and E (stop-gradient at the map), with the step $z \leftarrow z + (\Delta t/n_{\mathrm{inner}})\lambda\,\nabla r$.

λ grids and per-λ selection. Because the state space is two-dimensional, the inner hyperparameters can be swept exhaustively rather than range-found. For every (method, variant, λ) we run the full inner grid of Table 2 and report the configuration minimising the control objective itself,

$$J \;=\; \frac{1}{2\lambda}\sum_j \|\Delta x_j\|^2 \;-\; \bar{r}(x_N), \tag{2}$$

where $\Delta x_j = x_{j+1} - \bar{x}_{j+1}$ is the applied deviation from the uncontrolled Euler step. Each plotted marker is that single best run, so the curves are attainment frontiers under a fixed selection rule rather than non-dominated envelopes. The horizontal axis of both figures is the mean integrated control effort $\sum_j \|\Delta x_j\|^2$ averaged over the batch, which equals $\Delta t\, E$ for the energy E of (3), i.e. $E/16$ at $N = 16$.

For the upper-moon reward all three methods share $\lambda \in \{0.1,\ 0.25,\ 0.5,\ 0.75,\ 1,\ 2,\ 3\}$; beyond $\lambda \approx 3$ every method has saturated, since the reward is already near 1 on the mode the sampler visits.

For the off-manifold reward HardFlow and FMRG saturate by $\lambda \approx 5$ and are swept over the 13 values $\{0.1, \ldots, 100\}$, whereas ConicFlow is still improving at $\lambda = 100$ and is extended geometrically to $\lambda = 10^5$ (22 values in total) until its reward plateaus.

Table 2: Hyperparameters for the low-dimensional experiments.

Shared across methods		
Steps N		16
Batch / seed		512 particles / 42, shared across all runs
Base law p_0		$\mathcal{N}(0, \sigma_d^2 I)$, $\sigma_d = 0.750$
Backbone		MLP, 3×256 GELU; EMA weights at 15k steps
Per method		
ConicFlow	horizon H	$\{1, 2, 4, 8, 16\}$
	terminal map Φ_j	*endpoint* $X_{j+H,N}$ or *horizon* id
	inner optimizer	Adam
	inner iterations	$\{25, 50\}$, extended to $\{100, 200\}$ at large λ
	learning rate	$\{0.01, 0.05\}$, extended with $\{0.001, 0.005, 0.2\}$ at large λ
	window initialisation	fresh uncontrolled rollout (no cross-window warm start)
HardFlow	inner optimizer	L-BFGS (strong Wolfe) or Adam
	outer $\times$ inner	$\{3, 5, 10\} \times \{10, 20\}$
	learning rate	$\{0.1, 0.5\}$ (L-BFGS), $\{0.01, 0.05\}$ (Adam)
	penalty ρ_0 / growth	1 / $\times 2$ per outer round
	proximal weight	$\alpha_i^2/(2\Delta t\,\lambda)$, i.e. $\lambda_{\mathrm{oc}} = 1/\lambda$
	warm start	across sampling steps
FMRG	variant	J (through $X_{j,N}$) and E (stop-gradient)
	guidance steps	$\{1, 3, 5, 10\}$ per sampling step

C.1.2 Unconstrained image experiment implementation details

Model and sampler. We use FLUX.1-dev with the FlowMap LoRA adapter, so one forward pass evaluates the flow map $b_t(z,\, t \to t')$: sampling advances with $t' = t - \Delta t$, while $t' = 0$ estimates the terminal latent in a single call. The N steps are uniform, $t_j = 1 - j/N$, so $\Delta t = 1/N$ for the sampler. Each run draws one latent from a seed, and the uncontrolled sample from that same seed is the reference throughout.

Reward. CLIP (Radford et al., 2021) is the cosine similarity between normalised CLIP-ViT-H-14 image and text features. PickScore (Kirstain et al., 2023) is $\exp(\tau)$ times the same cosine under its own finetuned encoder, τ being the checkpoint's learned logit scale. ImageReward (Xu et al., 2023) is the score its checkpoint already standardises to zero mean and unit variance. All three are evaluated on the decoded image, so every reward gradient is taken through the VAE decoder as well as the scorer. The three scales are unrelated, so a reward gain is only ever compared within a reward. No constraints are active in this experiment.

Evaluation suite. Table 1 and the frontier figures are computed on twelve problems: three reward models $\times$ four (prompt, seed) instances. The instances are *"a portrait of a man"* $\to$ *"a smiling face"* at seeds 42 and 7, *"a photo of a sleeping cat"* $\to$ *"a photo of a cat with wide open eyes"* at seed 42, and *"a field of green grass"* $\to$ *"a field of red poppies"* at seed 42. Within a problem all three methods share the initial latent and the λ grid, so each problem contributes one matched triple of frontiers, and the $\pm$ in Table 1 is the standard error over the four instances of that reward. Figure 2 (left) is the PickScore man $\to$ smiling instance at seed 42; Figure 4 shows two of the others.

The image rows in Figure 2 (right) are selected at matched *achieved reward* rather than at matched λ, and use three editing pairs: *"a portrait of a man"* $\to$ *"a smiling face"* (seed 7), *"a portrait of a young man"* $\to$ *"a portrait of an old man"* (seed 317), and *"a portrait of a woman with straight hair"* $\to$ *"a portrait of a woman with curly hair"* (seed 1289). The latter two come from a separate editing sweep and do not enter Table 1.

Control methods. All three methods share the sampler, the initial latent and the reward. We run ConicFlow in its (RH-T) form with $H = 2$; since this experiment is unconstrained, Algorithm 1

reduces to $K = 1$ and its primal response is approximated by n_{inner} Adam steps at learning rate 10^{-2}, warm-started from the previous window's solution rather than from the pretrained path. We evaluate $\Phi_j = X_{j+H,N}$ through the recursion, i.e. $N - j - H$ pretrained steps. Within the window the drift is evaluated without gradient for $\ell > 0$, dropping the $J_b^\top$ term from the inner gradient.

HardFlow, ported here to FLUX latents, instead optimizes the predicted terminal latent $\widehat{x} = X_{j,N}(x_j)$ obtained from a single call and never differentiates the transformer. FMRG-J takes n_{inner} Jacobian-guidance steps per sampling step, $z \leftarrow z + (\Delta t / n_{\text{inner}})\, \lambda\, \nabla_z r\big(X_{j,N}(z)\big)$.

Choosing λ. The useful range of the control-cost weight λ moves with the reward's scale and with the problem, so we range-find it per (reward, prompt) pair at a cheap $N = 4$ proxy of the $N = 8$ sweep rather than fixing one grid. We step $\times 10$ down from a $\|\nabla_z r\|$-scaled starting value until Δr vanishes into the noise and $\times 10$ up until it saturates, then place six λ per method across the logistic rise fitted to $\Delta r(\lambda)$, densified with log-midpoints where a finer frontier was wanted. The same grid is reused across seeds. The three per-method grids are merged into one, so every arm runs at identical λ, at which all three apply the same first-order step $\lambda\, \Delta t\, \nabla r$ per sampling step. The resulting grids differ by orders of magnitude across the suite; the PickScore man $\rightarrow$ smiling cell behind Figure 2 (left) runs at $\lambda \in \{0.95, 1.5, 2.5, 4.9, 9.8, 15, 23, 38, 62, 130, 280\}$.

Mean reward gain over a shared energy budget. Comparing methods at a common λ is fragile, since λ enters each method differently and the same value buys different amounts of control. We therefore summarise each method by its whole frontier instead. Define $E = \sum_{k=0}^{N-1} \Delta t\, \|u_k\|^2$ to denote a run's control energy. Fix a problem (reward, prompt, seed). For method m we first discard every run that another run of the *same* method beats on both axes (i.e., no less energy and no more reward), leaving a set of non-dominated runs $\{(E_i, r_i)\}$ along which r increases with E; write $\widehat{r}_m$ for its piecewise-linear interpolant in $\log_{10} E$. Because the methods span different energies, we evaluate all of them on the interval attained by all three, $E_1 = \max_m \min_i E_i$ and $E_2 = \min_m \max_i E_i$, and report

$$G_m \;=\; \frac{1}{\log_{10}(E_2/E_1)} \int_{\log_{10} E_1}^{\log_{10} E_2} \big(\widehat{r}_m(E) - r_{\text{unc}}\big)\, \mathrm{d} \log_{10} E, \tag{3}$$

the mean height of the frontier over that budget range. G_m carries the units of the reward, and $G_m = 0$ means the method does no better than the uncontrolled sample anywhere in the range. Both endpoints lie inside every method's frontier by construction, so no extrapolation is involved, and (3) is invariant to any uniform rescaling of the energy axis. Integrating in $\log_{10} E$ rather than in E weights each decade of budget equally; on a linear axis the top decade would carry almost all the weight. ConicFlow is taken at 24 inner iterations, as plotted in Fig. 2.

Two properties of (3) are worth stating. It is conservative for the method that reaches furthest: the interval stops at E_2, the lowest of the three maximum energies, so any reward attained beyond it — typically ConicFlow's — is discarded. And it scores the attainment frontier rather than the sweep itself, so a method whose reward is non-monotone in λ is represented by its upper envelope.

Hardware and cost. One run occupies a single NVIDIA RTX A5000 (24 GB), one process per card, peaking at 16.3–17.7 GB. Median wall clock over the 11 values of λ: 317 s and 626 s for ConicFlow at 12 and 24 inner iterations, 46 s for HardFlow, 43 s for FMRG-J, against 3–4 s for the uncontrolled sample.

C.1.3 Constrained image experiments implementation details

The sampler, resolution, precision, guidance scale, PickScore scorer, $N = 8$, and $H = 2$ are those of Appendix C.1.2; only what differs is described here.

Problem. There is no reward: $r \equiv 0$, so each method's objective is its control energy alone and the two methods are compared on what it costs them to reach the same feasible set. An instance is a base prompt c, one to three *attribute prompts* $c_1, \ldots, c_C$, and a seed. Constraint i is

$$h_i(z) \;=\; \tau_i \;-\; s\big(\mathrm{dec}(z),\, c_i\big) \;\le\; 0, \tag{4}$$

where $s(\cdot, c_i)$ is the PickScore of the decoded image against c_i, exactly the unnormalised score of Appendix C.1.2 with c_i in place of the edit prompt. Every constraint is evaluated at the predicted

Table 3: Hyperparameters for the unconstrained image experiment (Fig 2)

Shared across methods		
Steps N / horizon H		8 / 2
Resolution / latent		256×256 / $16 \times 32 \times 32$
Guidance scale		3.5
Precision		bf16 sampler, int8 transformer, fp32 reward
Reward models		CLIP, PickScore (both CLIP-ViT-H-14), ImageReward
Seeds / batch		42 and 7 / 1
λ grid		range-found per (reward, prompt)
Per method		
ConicFlow	inner iterations	12 and 24 (reported as two configurations)
	Inner optimizer	Adam
	Learning rate	10^{-2}
	Terminal estimate	rollout of the remaining $N - j - H$ steps
	drift term	detached for $\ell > 0$
HardFlow	inner iterations	40
	Learning rate	2.5 (replaced by $0.5/\mathrm{reg}$ where $\mathrm{lr} \cdot \mathrm{reg} \geq 1$)
	terminal estimate	single flow-map call to $t = 0$
	effort penalty	$\mathrm{reg} = N/(2\lambda)$, active from step 0
FMRG-J	guidance steps	3 per sampling step

terminal latent, and a run is *feasible* when $\max_i h_i \leq 0$ on the final decoded image. Multiple constraints are independent PickScore heads that share nothing but the image. The generation prompt stays c throughout.

Thresholds. Absolute PickScore values are not comparable across prompts or seeds, so thresholds are set relative to each instance's own uncontrolled sample. For each (prompt, seed) pair we first draw the uncontrolled sample from that seed and score it with every constraint to obtain baselines s_i^0 (19.2–22.5 across the instances here). A *tightness multiplier* m then sets every threshold of the instance at once,

$$\tau_i(m) \;=\; s_i^0 + (m-1)\,|s_i^0| \;=\; m\,s_i^0 \quad (s_i^0 > 0), \tag{5}$$

so $m = 1$ places the boundary exactly on the uncontrolled image and $m = 1.5$ demands a 50% higher score on every attribute simultaneously. (The absolute-value form is what the code implements; it keeps the same grid meaningful for scorers with negative baselines, which does not arise for PickScore.) One (prompt, seed, m) triple is one comparison: both methods receive the same λ, the same thresholds and the same initial latent, and the two uncontrolled images are checked to be byte-identical before the figure is drawn.

ConicFlow. The window problem is (RH-T) with the reward term replaced by the Lagrangian term $\sum_i \mu_i\, h_i\big(\widehat{e}(x_H)\big)$, $\mu_i \geq 0$. After each window solve one projected dual-ascent step is taken on the constraint value the optimiser saw at its horizon endpoint,

$$\mu_i \;\leftarrow\; \max\big(0,\; \mu_i + \eta\, h_i(\widehat{e}(x_H))\big), \tag{6}$$

and the multipliers persist across sampling steps. All multipliers start at $\mu_{\mathrm{init}} = 0.5$ with $\eta = 0.5$; both were chosen on the pilot instance ("a portrait of a man" $\rightarrow$ "a smiling face") and then frozen for every run reported. The remaining difference from the unconstrained runs is the terminal estimate *inside* the inner objective: rather than rolling the remaining $N - j - H$ steps out, $\widehat{e}$ is a single flow-map call from x_H to $t = 0$, which makes the cost of one sampling step linear rather than quadratic in N. Only the surrogate the optimiser differentiates changes; the applied trajectory, the final image and everything scored from it are computed as before.

HardFlow. Constraints enter HardFlow's inner problem as an augmented Lagrangian on the predicted terminal latent $\widehat{x}$, as in the released code: the ascent objective is $-\sum_i \mu_i h_i(\widehat{x}) - \frac{\rho}{2}\sum_i \max(0, h_i(\widehat{x}))^2 - \mathrm{reg}\,\alpha^2\|\widehat{x} - \bar{x}\|^2$, climbed with 40 explicit gradient steps; every 8 inner steps the multipliers take the projected step (6) with $\eta = 0.2$, and they are reset to 0 at the start of every sampling step (the released code's semantics, retained). Two settings depart from that code and

both were forced by stability. First, the released code doubles ρ whenever a violation exceeds 0.02 at a dual update; over 40 inner steps that lets ρ grow from 5 to 160 within a single sampling step, and at the tighter thresholds it produced control energies orders of magnitude above ConicFlow's on the same instance. We hold $\rho = 5$ fixed. Second, the inner ascent is an explicit fixed-step scheme whose error contracts by $1 - 2\,\mathrm{lr} \cdot \mathrm{reg} \cdot \alpha^2$ per step, with $\alpha \to 1$ at the final sampling step; with the authors' $\mathrm{lr} = 2.5$ and $\mathrm{reg} = N/(2\lambda) = 0.2$ that factor reaches 0 and the pilot runs diverged, so the inner learning rate is 0.5 ($\mathrm{lr} \cdot \mathrm{reg} = 0.1$). The constraint margin and activation fraction are 0, as in Appendix C.1.2.

Choosing λ. $\lambda = 20$ for both methods throughout. With no reward, λ does not trade reward against effort; it only weights energy against the constraint term and therefore selects *which* feasible point each method settles on, not whether it reaches one.

Instances and sweep. Table 5 lists the instances. Nine prompt pairs carry one or two or three constraints and are swept over a five-point grid of tightness multipliers, from $m = 1.1$ to $m = 1.5$ in steps of 0.1. Every instance is run at both seeds, giving 86 comparisons, i.e. 172 controlled runs plus one uncontrolled probe per (prompt, seed). The attribute prompts were written so that the base prompt does not already imply the attribute (e.g., "a young man with straight hair" $\to$ "curly hair").

Table 4: Hyperparameters for the constrained image experiments. Everything not listed (sampler, resolution, precision, guidance scale) is as in Table 3.

Shared across methods		
Reward		none ($r \equiv 0$; objective is control energy)
Constraint scorer		PickScore, one head per attribute prompt
Constraint form		$h_i = \tau_i - s_i \le 0$, evaluated at the endpoint
Thresholds		$\tau_i = m\, s_i^0$, Eq. (5)
Steps N / horizon H		8 / 2
λ		20 (fixed)
Seeds		310, 698
Per method		
ConicFlow	inner iterations	24
	inner optimiser	Adam, lr 10^{-2}, warm-started
	terminal estimate	one flow-map call $x_H \to t = 0$ ("jump")
	drift term	detached for $\ell > 0$
	multipliers	$\mu_{\mathrm{init}} = 0.5$, persist across steps
	dual step η	0.5, once per sampling step, Eq. (6)
HardFlow	inner iterations	40
	inner lr	0.5 ($\mathrm{lr} \cdot \mathrm{reg} = 0.1$)
	terminal estimate	single flow-map call to $t = 0$
	effort penalty	$\mathrm{reg} = N/(2\lambda) = 0.2$, active from step 0
	multipliers	$\mu_{\mathrm{init}} = 0$, reset every sampling step
	dual step η	0.2, every 8 inner steps
	penalty weight ρ	5, held fixed (released code doubles it)
	margin / activation	0 / 0

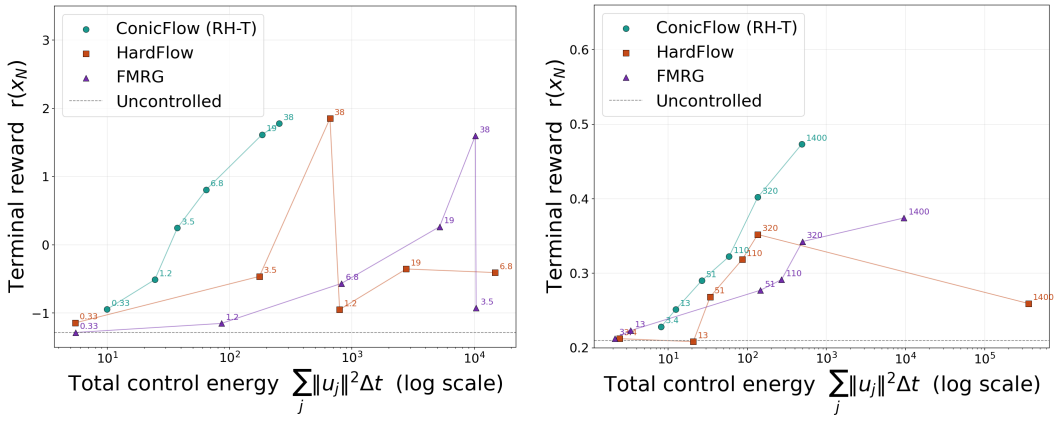
C.2 Additional Results

Unconstrained. We present additional visualizations of the unconstrained image editing comparisons in Figure 5, similar to Figure 2 in the main paper. We also present two additional pareto plots of instances in Figure 4 similar to the one in Figure 2 but with different reward/prompts which define a different problem.

Constrained We present additional visualizations of the constrained image editing comparisons in Figure 6, 7, and 8, similar to Figure 3 in the main paper.

Table 5: Constrained instances. Each attribute prompt is one PickScore constraint (4); semicolons separate the constraints of an instance.

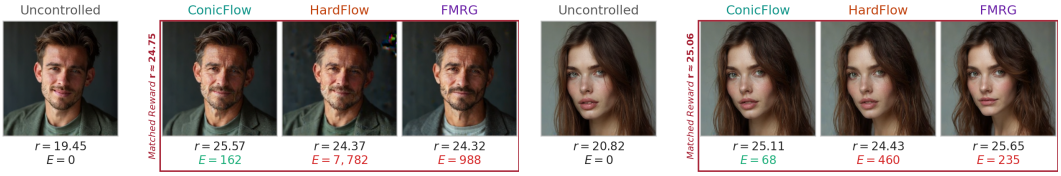
base prompt c	attribute prompts $c_1, \ldots, c_C$
a portrait of a man	a smiling face
a photo of a golden retriever	a dog wearing sunglasses
a city street at noon	a city street at night
a portrait of a woman	a woman with bright red hair
a portrait of a woman	sad frowning woman
portrait of a young man with straight hair	man with curly hair; photo of very old man
a portrait of a woman	a woman wearing glasses; a woman with short blonde hair
a house by a lake	a wooden log cabin; a house in heavy fog
a green garden on an overcast day	a garden with red flowers; a garden in bright sunshine; a garden with a stone path



(a) ImageReward, "a photo of a sleeping cat" → "a photo of a cat with wide open eyes", seed 42.

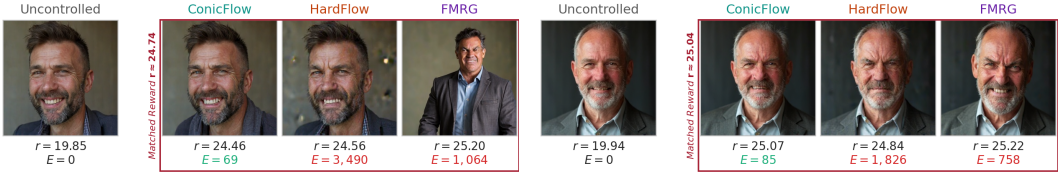
(b) CLIP, "a portrait of a man" → "a smiling face", seed 7.

Figure 4: Reward–effort trade-off on two further editing instances, $N = 8$, $H = 2$. The horizontal axis is control energy $\sum_k \|u_k\|^2 \Delta t$ and the vertical axis the achieved reward $r(x_N)$, so up and to the left is better. Each marker is one value of λ.



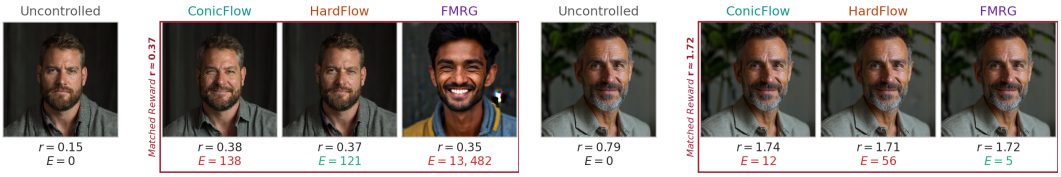
(a) PickScore, seed 1289: *"a portrait of a young man"* → *"a portrait of an old man"*

(b) PickScore, seed 317: *"a woman with straight hair"* → *"... with curly hair"*



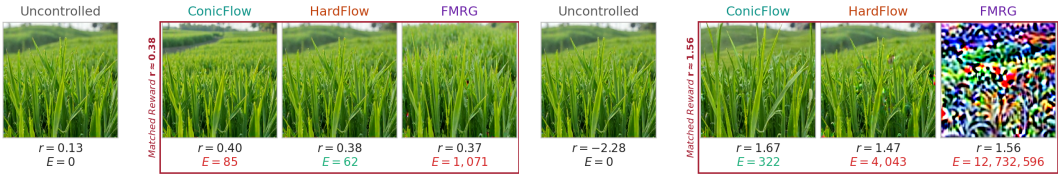
(c) PickScore, seed 317: *"a portrait of a smiling man"* → *"... an angry man"*

(d) PickScore, seed 1289: *"a portrait of a smiling man"* → *"... an angry man"*



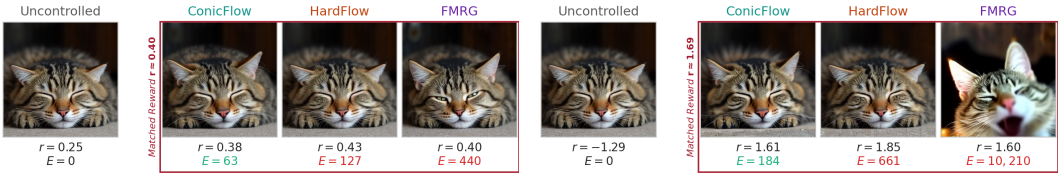
(e) CLIP, seed 42: *"a portrait of a man"* → *"a smiling face"*

(f) ImageReward, seed 7: *"a portrait of a man"* → *"a smiling face"*



(g) CLIP, seed 42: *"a field of green grass"* → *"a field of red poppies"*

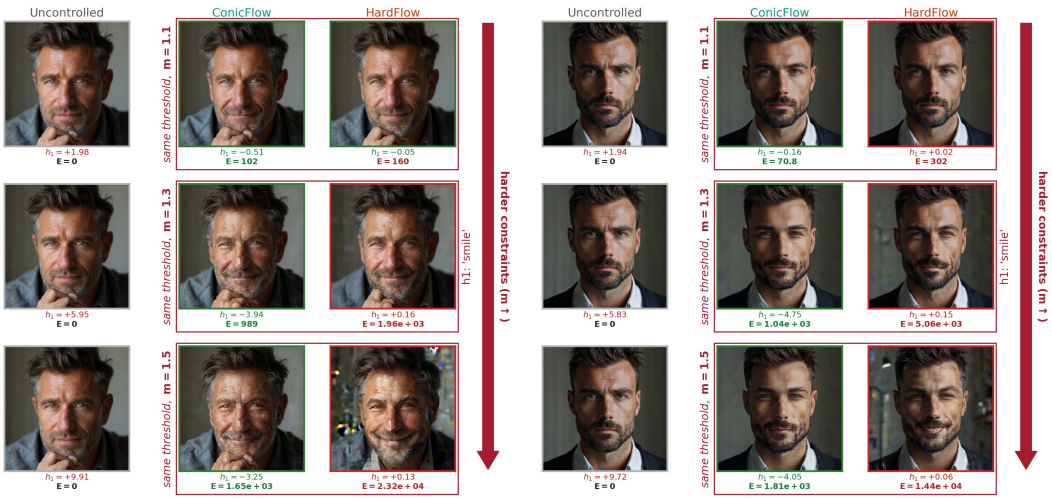
(h) ImageReward, seed 42: *"a field of green grass"* → *"a field of red poppies"*



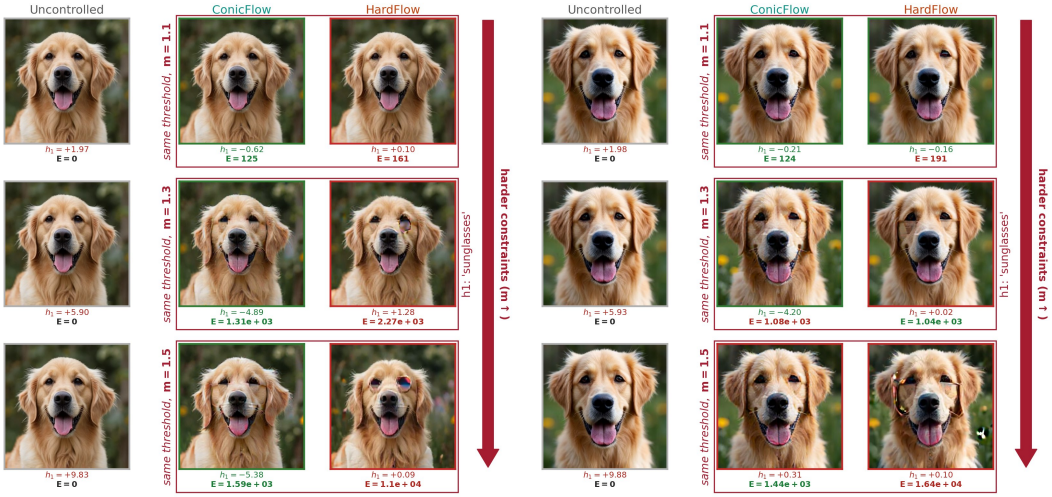
(i) CLIP, seed 42: *"a photo of a sleeping cat"* → *"... with wide open eyes"*

(j) ImageReward, seed 42: *"a photo of a sleeping cat"* → *"... with wide open eyes"*

Figure 5: More examples of unconstrained editing. Each panel is one (reward, prompt, seed) instance at a matched-reward level. Panels g–j are the harder case: turning a green field red, or opening a sleeping cat's eyes, asks for the image to be restructured rather than adjusted, and no method delivers it.



(a) *"a portrait of a man"* → *"a smiling face"*

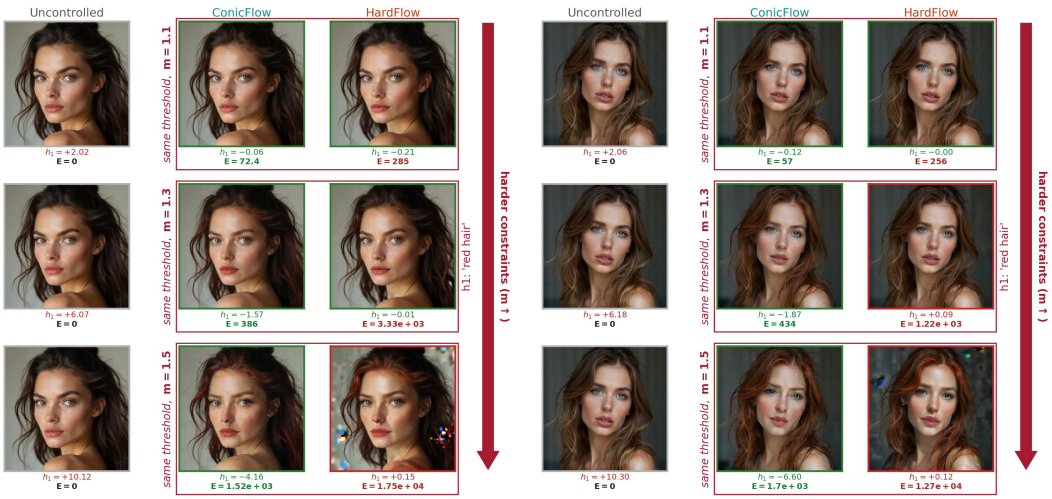


(b) *"a photo of a golden retriever"* → *"a dog wearing sunglasses"*

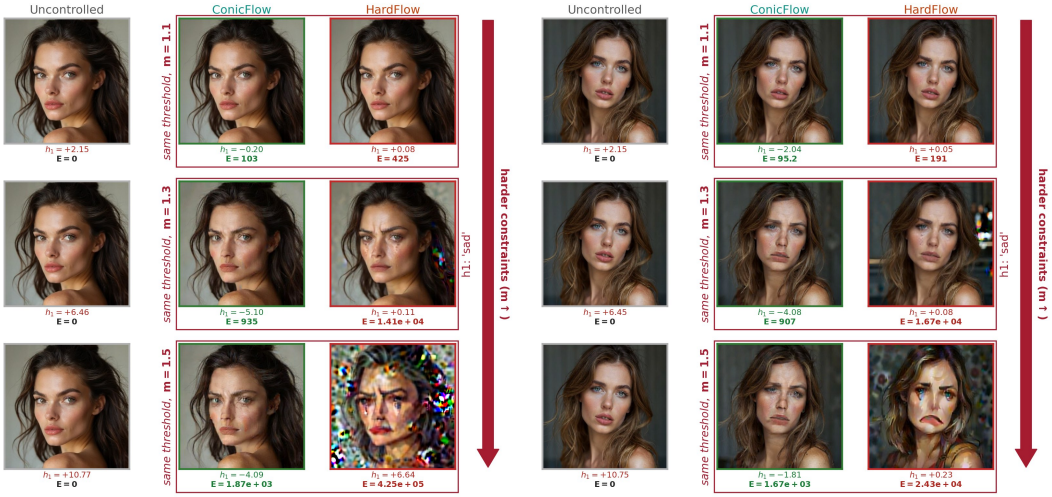


(c) *"a city street at noon"* → *"a city street at night"*

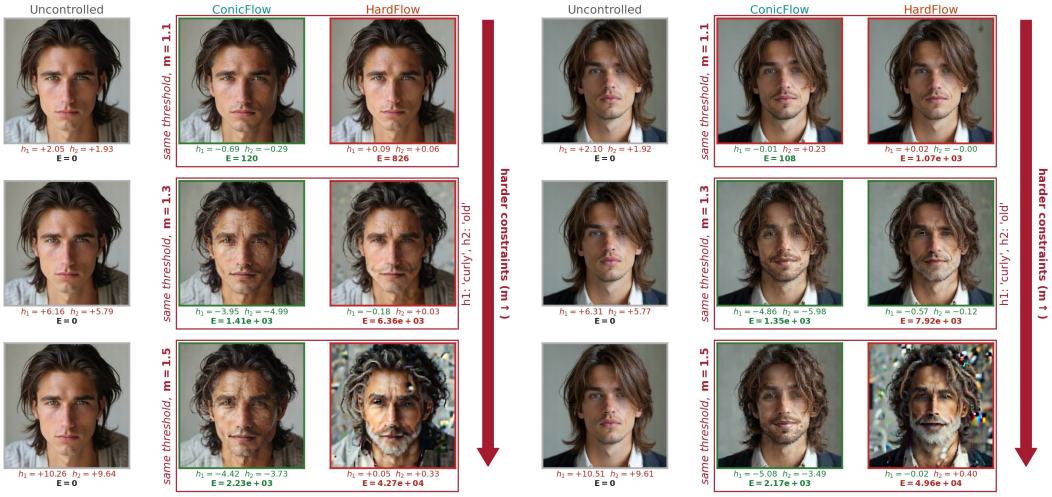
Figure 6: More examples of constrained editing. Each panel is one (prompt, seed) instance. Instance details are in Table 5.



(a) *"a portrait of a woman"* → *"a woman with bright red hair"*

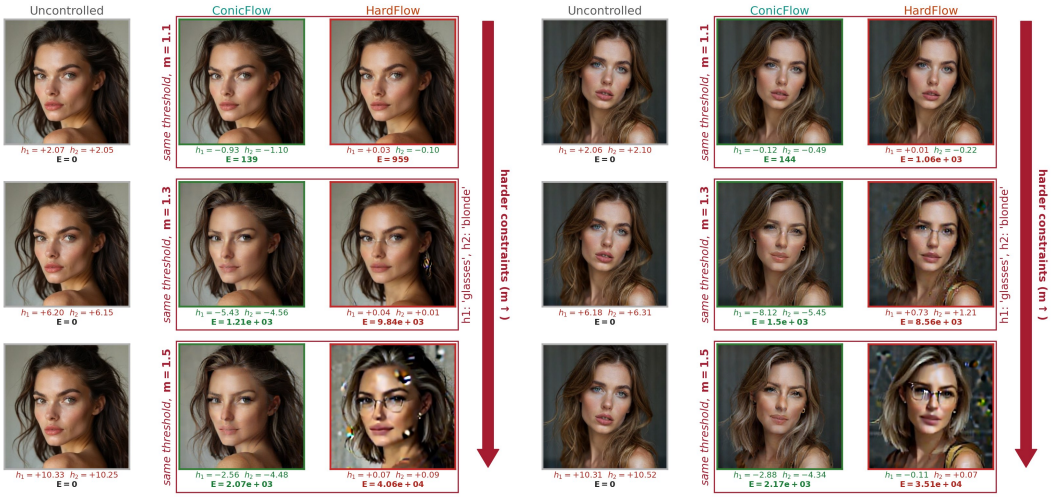


(b) *"a portrait of a woman"* → *"sad frowning woman"*



(c) *"portrait of a young man with straight hair"* → *"man with curly hair; photo of very old man"*

Figure 7: More examples of constrained editing. Each panel is one (prompt, seed) instance. Instance details are in Table 5.



(a) *"a portrait of a woman"* → *"a woman wearing glasses; a woman with short blonde hair"*



(b) *"a house by a lake"* → *"a wooden log cabin; a house in heavy fog"*



(c) *"a green garden on an overcast day"* → *"a garden with red flowers; a garden in bright sunshine; a garden with a stone path"*

Figure 8: More examples of constrained editing. Each panel is one (prompt, seed) instance. Instance details are in Table 5.

D ABLATIONS

Three questions are relevant to the comparison in Section 5: (1) whether ConicFlow's advantage is an artifact of the planning horizon we happened to choose, (2) whether it survives once the baselines are given the same compute, and (3) how much of it comes from where the horizon objective is evaluated. We address all three in this section on the editing task at $N = 8$, on exactly the shared λ grids of Section 5.

D.1 PLANNING HORIZON

ConicFlow plans over H steps and applies the first. We re-ran it at $H \in \{1, 4, 6, 8\}$ against the shipped $H = 2$ configuration: four editing prompts at seed 317, $n_{\text{inner}} = 24$, $\text{lr} = 10^{-2}$, terminal estimate by a single flow-map call ("jump"), six λ per cell, so 24 operating points per H. HardFlow and FMRG-J have no planning horizon and are not re-run.

Table 6: Planning horizon, editing, $N = 8$, seed 317; Δr, E and time are medians over the 24 operating points in each H. Wall clock is measured on shared cards and is indicative only.

H	Δr	E	time
1	3.98	78.5	207 s
2	4.45	**68.7**	249 s
4	4.39	70.6	288 s
6	4.67	90.3	321 s
8	**4.95**	101.4	264 s

Moving from a one-step to a two-step horizon improves both quantities at once: Δr rises 12% and E falls 12%. Beyond $H = 2$ the two diverge. Reward keeps rising, apart from a plateau at $H = 4$, and reaches its maximum at $H = 8$, 24% above $H = 1$ for 24% more NFE, where $\text{NFE} = n_{\text{inner}}\big(N(N-1)/2 + H\big)$. Energy grows much faster over the same range: from $H = 2$ to $H = 8$, E rises 48% while Δr gains only 11%. Longer horizons therefore buy reward with disproportionately more control effort, and on the objective $H = 2$ is the best of the five.

D.2 MATCHED WALL CLOCK

ConicFlow is the only arm that backpropagates through the transformer, so at equal λ it also spends more compute: 239 s against 36 s for HardFlow and 33 s for FMRG-J in Section 5. We removed that advantage by retuning n_{inner} to 12, 167 and 13 respectively, which brings all three to $\approx$122 s per run, and re-ran the four editing prompts at seeds 317 and 1289 on the same shared λ grids: 48 operating points at equal λ *and* equal compute.

Table 7: Equal λ and equal wall clock, editing, $N = 8$, seeds 317 and 1289; Δr, E and time are medians.

	n_{inner}	time	Δr	E
ConicFlow	12	124 s	**4.16**	84.6
HardFlow	167	123 s	2.39	555.2
FMRG-J	13	128 s	1.84	**32.5**

At matched compute ConicFlow still delivers the largest reward gain, a median Δr of 4.16 against 2.39 for HardFlow and 1.84 for FMRG-J (Table 7). The two baselines fall short in opposite directions. HardFlow, given more than three times its shipped compute, reaches only 57% of ConicFlow's reward gain and pays 6.6× the control energy for it ($E = 555.2$ against 84.6). FMRG-J spends the least energy of any arm, $E = 32.5$, but only because it barely moves the sample: its reward gain is under half of ConicFlow's. ConicFlow is the only arm that combines a large reward gain with moderate energy, and extra compute alone does not move either baseline there.

D.3 OBJECTIVE EVALUATION POINT

The RH-T variant of ConicFlow scores the horizon objective at the flow-map prediction of the terminal latent rather than at the planning-window state x_H it actually reaches. We ran a comparison between RH-H and RH-T variants. We used four editing prompts at 2 seeds, $n_{\text{inner}} = 24$, $\text{lr} = 10^{-2}$, six λ values per cell, so 48 operating points. As seen in Table 8, evaluating at the terminal point is worth more than the compute. RH-H runs $2.6\times$ faster but costs 1.54 in median Δr while *raising* median energy by a quarter, so the RH-H is dominated by RH-T on both axes.

Table 8: Where the horizon objective is evaluated, editing, $N = 8$, seeds 317 and 1289; Δr, E and time are medians.

	objective at	time	Δr	E
ConicFlow	endpoint	238 s	**4.79**	**68.7**
	horizon	90 s	3.25	86.9
HardFlow	—	36 s	2.00	553.9
FMRG-J	—	33 s	2.39	87.0