LEARNING TO RESTART: DIFFERENTIABLE RESTART POLICIES FOR OPTIMIZATION DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive restart mechanisms, which reset the momentum or other auxiliary state of an iterative optimizer, play a central role in the empirical performance of accelerated first-order methods, yet existing restart rules remain almost entirely hand-designed. The obstruction to learning restart policies is that, unlike the typical Learning to Optimize (L2O) setting, the discrete restart events make the resulting trajectory non-differentiable in the algorithm parameters and thus block gradient-based optimization of the restart policy. We resolve this obstruction by developing a differentiable framework for restart policies through their continuous-time limit. We introduce *surrogate gradients* for the restart events and optimization trajectories that are efficient to compute and provably converge to their continuous-time counterparts. We validate the effectiveness of our surrogate gradients in L2O settings, training restart policies that improve upon standard handcrafted adaptive restart criteria on quadratic and linear programs, and that transfer zero-shot from synthetic quadratics to logistic regression on real datasets.

1 INTRODUCTION

Optimization algorithms underlie a wide range of modern applications, including operations research (Taha, 1997), the training of large-scale machine learning models (Zhao et al., 2025), and decision-making in finance (Pilbeam, 2018). Substantial effort has therefore been devoted to accelerating these algorithms, both by analytical design and, more recently, by data-driven approaches such as hyperparameter optimization (Feurer & Hutter, 2019) and learning to optimize (L2O) (Chen et al., 2022; 2024). In this line of work, the training objective is differentiable in the parameters to be optimized, which enables efficient gradient-based optimization.

What is largely missing from this data-driven toolbox is the *restart* mechanism: adaptive resets of the momentum or other auxiliary state of an iterative algorithm. Restart is routinely used to accelerate convergence on a wide range of convex optimization problems (O'donoghue & Candes, 2015; Giselsson & Boyd, 2014; Fercoq & Qu, 2019, etc.), and it is also a key ingredient behind the empirical performance of PDLP on large-scale linear programs (Applegate et al., 2021; 2023; Lu & Yang, 2025; Applegate et al., 2025). The vast majority of existing restart rules, however, are designed analytically in a case-by-case manner. It is therefore natural to follow the L2O paradigm and *learn* the restart criterion, by parameterizing it via a neural network and optimizing it directly against a downstream optimization objective. The key obstacle is that the restart events are discrete in time, so the resulting optimization trajectory is *non-differentiable*, which precludes a naive application of gradient-based learning.

In this work, we resolve this obstacle by developing a differentiable framework for learning restart rules in optimization dynamics. Our starting point is the observation that, although the restart events of a discrete-time algorithm are non-differentiable in the algorithm parameters, the restart events in its continuous-time limit dynamics *are* differentiable. Our main contributions are summarized as follows:

- **Continuous-time differentiable restart.** In continuous-time dynamics, we apply the implicit function theorem to obtain closed-form representations of the gradients for both the restart times and the trajectory (Theorem 2.2). We demonstrate that by using these gradient surrogates, a parameterized restart rule can be optimized to outperform the standard hand-designed adaptive restart criteria on the underlying ODE dynamics (Section 2.4).

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.

- **Efficient discrete approximation.** We develop a *surrogate gradient* (Definition 3.2)—the discrete-time analogue of the continuous restart sensitivity that can be implemented efficiently (e.g., through the standard automatic-differentiation framework). We prove that, whenever the discrete dynamics are *consistent* with their continuous-time limit, the surrogate gradients converge to the continuous gradients (Theorem 3.4).
- **Applications to L2O.** We instantiate the framework in the L2O setting through experiments on Quadratic Programming (Section 4.1) and on Linear Programming (Section 4.2), each with a synthetic training family and real test problems. The learned restart policies consistently outperform the standard handcrafted restart criteria on both problem families, confirming the effectiveness of our discrete surrogate gradients. In particular, a policy trained only on synthetic quadratics transfers zero-shot to logistic regression on real datasets and to tomographic reconstruction from real scanner measurements (Section 4.1), where it improves over the best handcrafted criterion.

Related work Due to space constraints, we defer the full literature review to Appendix A.

2 Differentiable Restarts: A Continuous-time Perspective

In the following, we first motivate the technical obstacle of applying the L2O paradigm to learn the adaptive restart policies.

A motivating example Consider NAG with parameter-dependent stepsize $\alpha_k(\theta)$ and momentum coefficient $\beta_k(\theta)$ applied to a smooth convex objective f:

$$x_{(k+1)} = y_{(k)} - \alpha_k(\theta)\,\nabla f(y_{(k)}), \qquad y_{(k+1)} = x_{(k+1)} + \beta_k(\theta)\big(x_{(k+1)} - x_{(k)}\big), \tag{1}$$

where both parameter schedules are differentiable in the hyperparameter θ. Without restarts, the iterate $x_{(N)}$ produced after N steps is a smooth function of θ, and any objective built from it—for instance $f(x_{(N)})$—admits ordinary gradient-based optimization in θ.

To equip (1) with an adaptive restart in the spirit of O'donoghue & Candes (2015), the natural construction is to introduce a parameterized restart score $\bar{V}(x, y; \theta_{\mathrm{rst}})$ and trigger a restart whenever $\bar{V}(x_{(k)}, y_{(k)}; \theta_{\mathrm{rst}}) \le 0$, at which point the momentum is reset by setting $y_{(k)} \leftarrow x_{(k)}$. The full parameter vector is then $(\theta, \theta_{\mathrm{rst}})$, and the L2O objective becomes to learn the within-phase schedule and the restart criterion jointly. However, the indices at which $\bar{V}$ crosses below zero are *integer-valued* and locally constant in $(\theta, \theta_{\mathrm{rst}})$. The iterate $x_{(N)}$ is therefore a piecewise-constant function of the parameters that is generally discontinuous across these jumps. The issue is particularly severe for θ_{rst}, which influences the trajectory *only* through the location of the restart events: away from those events, the derivative of $x_{(N)}$ with respect to θ_{rst} vanishes and carries no information.[1]

Our approach To circumvent this obstruction, we adopt the continuous-time perspective of optimization algorithms. Most discrete optimizers (including NAG and the primal–dual schemes considered later) admit a continuous-time limit (e.g., a gradient flow) in which the restart triggered by $\bar{V} = 0$ becomes *smooth*. In this section, we formalize this intuition, derive the continuous-time gradients, and validate their effectiveness. Section 3 then transfers these continuous-time gradients back to the discrete algorithm via an efficient surrogate, closing the loop between the continuous-time analysis and the practical iterative optimizers.

2.1 ODE Dynamics and Restarts

We use the word *state* broadly: depending on the algorithm, it may comprise the optimization variable, momentum variables, primal–dual pairs, running averages, or any other auxiliary quantity maintained by the optimizer. The evolution of the state in the dynamical systems of interest can typically be written as a parameterized ordinary differential equation (ODE):

$$\dot{X}(t) = -\mathcal{A}_{\mathsf{X}}(X(t), t; \theta), \qquad X(0) = X_0, \tag{2}$$

[1] The straight-through estimator (Bengio et al., 2013) corresponds to setting the integer-step derivative to zero and would therefore produce $\partial_{\theta_{\mathrm{rst}}} x_{(N)} \equiv 0$ identically in this setting, providing no gradient information for θ_{rst}.

where $\mathcal{A}_{\mathsf{X}} : \mathbb{R}^D \times \Theta \to \mathbb{R}^D$ encodes the instantaneous direction of state motion induced by the algorithm with (hyper-)parameter $\theta \in \Theta$.

Many classical optimization algorithms—gradient descent, momentum methods, accelerated gradient methods, and primal–dual gradients—admit such a gradient-flow representation, in which the state X collects the iterate together with any momentum or auxiliary variables maintained by the algorithm and the field $\mathcal{A}_{\mathsf{X}}$ encodes the corresponding update rule. We illustrate the abstraction with three concrete examples below.

Gradient flow. The simplest example is the (rescaled) gradient flow,

$$\dot{x}(t) = -\alpha(t;\theta)\,\nabla f(x(t)),$$

where f is the objective and the scaling function $\alpha(t;\theta)$ is time- and parameter-dependent. In the form of (2), the field is $\mathcal{A}_{\mathsf{X}}(x,t;\theta) = \alpha(t;\theta)\,\nabla f(x)$.

Nesterov's accelerated gradient (NAG). Several ODE models of NAG[2] have been proposed in the literature (Su et al., 2016; Shi et al., 2022). On the smooth convex minimization problem $\min_{x\in\mathbb{R}^d} f(x)$, the limiting behaviour of NAG with stepsize $s > 0$ can be captured by the *high-resolution* ODE (Shi et al., 2022):

$$\ddot{x}(t) + \tfrac{3}{t+\tau}\dot{x}(t) + \sqrt{s}\,\nabla^2 f(x(t))\,\dot{x}(t) + \left(1 + \tfrac{3\sqrt{s}}{2(t+\tau)}\right)\nabla f(x(t)) = 0, \tag{3}$$

with $\tau = 3\sqrt{s}/2$, $x(0) = x_0$ and $\dot{x}(0) = -\sqrt{s}\nabla f(x_0)$, which is shown to accurately track the trajectory of the discrete iteration. This second-order ODE fits the abstract form (2) if the state $X(t) = [x(t); v(t)]$ consists of the iterate and the momentum $v(t) = \dot{x}(t)$. We use these dynamics throughout Section 2.4 to validate the continuous-time gradients derived in Section 2.3.

Primal–dual gradient dynamics. For minimax (primal–dual) problems $\min_x \max_y \mathcal{L}(x,y)$, the analogous continuous-time model is the primal–dual gradient flow (Arrow et al., 1958; Feijer & Paganini, 2010),

$$\dot{x}(t) = -\nabla_x \mathcal{L}(x(t),y(t)), \qquad \dot{y}(t) = \nabla_y \mathcal{L}(x(t),y(t)).$$

Closely related dynamics have been studied for constrained minimax problems (Feijer & Paganini, 2010; Cherukuri et al., 2016) and for accelerated primal–dual methods such as the primal–dual hybrid gradient (PDHG) method (Chambolle & Pock, 2011; Li & Shi, 2024).

2.2 Dynamics with Restarts

ODE dynamics with restarts can then be (informally) described by augmenting the ODE (2) with an additional scalar score $V(t)$ that determines the restarting (with $x(0) = x_0$ and $V(0) = 1$):

$$\begin{cases} \frac{dX(t)}{dt} = -\mathcal{A}_{\mathsf{X}}(X(t),t;\theta), \\ \frac{dV(t)}{dt} = -\mathcal{A}_{\mathsf{V}}(X(t),V(t),t;\theta), \end{cases} \qquad \text{while } V(t) > 0. \tag{4}$$

At any time t such that $V(t-) = 0$, a restart is triggered: we reset $V(t) = 1$ and apply a restart map $R : \mathbb{R}^D \to \mathbb{R}^D$ to obtain $X(t) = R(X(t-))$. Here $\mathcal{A} = (\mathcal{A}_{\mathsf{X}}, \mathcal{A}_{\mathsf{V}})$ specifies the joint dynamics of the state and the restart score, while R specifies the restart operation, which may, for instance, leave the iterate component unchanged but reset the momentum variables to zero.

The system in (4) is hybrid: between consecutive restarts the state evolves continuously according to a smooth ODE, while at each restart it is reinitialized in a discontinuous fashion via the map R together with the score reset $V \leftarrow 1$. Such hybrid evolutions can be analyzed rigorously within the framework of hybrid dynamical systems (Van Der Schaft & Schumacher, 2007; Goebel et al., 2009) or neural event functions for ODEs (Chen et al., 2020). In the following, we develop the minimal notation required to describe such systems.

[2]The corresponding discrete iteration and the standard hand-crafted restart rules are summarized in Appendix B.1.

General flow-map dynamics. To describe dynamics with restarts, we start by specifying the evolution within each restart *phase*. Let

$$F = (F_{\mathsf{X}}, F_{\mathsf{V}}) : \mathbb{R}_+ \times \mathbb{R}^D \times \Theta \to \mathbb{R}^D \times \mathbb{R}$$

be a differentiable flow map, with $F_{\mathsf{X}}(0; x, \theta) = x$ and $F_{\mathsf{V}}(0; x, \theta) > 0$ at each phase initialization, where $F_{\mathsf{X}}(t; x, \theta)$ ($F_{\mathsf{V}}(t; x, \theta)$) denotes the state X (score V) reached after time t when the phase starts from x. In the ODE setting above, $F(t; x, \theta)$ is the solution $(X(t), V(t))$ at time t of ODE (4) initialized at $X(0) = x$ and $V(0) = 1$, but our framework does not require F to be of this specific form.

Intuitively, the whole trajectory is obtained by repeatedly flowing under F until the learned event condition $V \le 0$ is met, and then starting the next phase from the resulting state. Formally, we consider a parameterized global trajectory $(X(t) = X(t; \theta))_{t \ge 0}$ obtained by concatenating phases. On the i-th phase $[T_i, T_{i+1})$, the trajectory follows

$$X(T_i + t) = F_{\mathsf{X}}(t; X(T_i), \theta), \qquad 0 \le t < L_i := T_{i+1} - T_i, \tag{5}$$

where L_i denotes the length of the phase determined by the score $V(T_i + t) := F_{\mathsf{V}}(t; X(T_i), \theta)$ via

$$L_i := \inf\{t \ge 0 : F_{\mathsf{V}}(t; X(T_i), \theta) \le 0\}, \qquad \inf \varnothing := +\infty. \tag{6}$$

At a finite event, the pre-reset score satisfies $V(T_{i+1}-) = F_{\mathsf{V}}(L_i; X(T_i), \theta) = 0$, and the post-reset state is $X(T_{i+1}) = R(F_{\mathsf{X}}(L_i; X(T_i), \theta))$. If $L_i = +\infty$, the current phase continues without a further reset.

We summarize this construction in the following definition.

Definition 2.1 (Continuous restarted trajectory)**.** The restarted trajectory $(X(t))_{t \ge 0}$ is defined recursively, starting with $T_0 = 0$ and initial point $X(0) = x_0$. For each phase i with starting time T_i and initial point $x_i := X(T_i)$, we define

$$X(T_i + t) = F_{\mathsf{X}}(t; x_i, \theta), \qquad 0 \le t < L_i, \tag{7}$$

with restart time L_i defined in (6). For finite L_i, we then set

$$T_{i+1} = T_i + L_i, \qquad x_{i+1} = X(T_{i+1}) = R(F_{\mathsf{X}}(L_i; x_i, \theta)). \tag{8}$$

The notation here is deliberately compact, but it straightforwardly covers the standard restart rules used in practice. As a concrete instance, suppose that the restart score depends only on the elapsed time t, the current state $X(T_i + t)$, and the state $x_i = X(T_i)$ at the start of the current phase. Then F_{V} admits the form

$$F_{\mathsf{V}}(t; x_i, \theta) = \bar{V}\big(t, X(T_i + t), x_i; \theta\big), \tag{9}$$

where $\bar{V}$ is a parameterized function of the elapsed time, the current state, and the phase-start state. Handcrafted restart rules—such as the function-value and gradient-based rules of O'donoghue & Candes (2015)—correspond to such fixed, problem-specific choices of $\bar{V}$. In our setting, $\bar{V}$ is parameterized by a small neural network and its parameters are learned jointly with any parameters of the within-phase dynamics.

2.3 Differentiating Restart

The restart time is defined implicitly by the event equation $F_{\mathsf{V}}(L_i; x_i, \theta) = 0$, and its derivative can therefore be derived through the chain rule and the implicit function theorem, as in the differentiation of stopping times (Chen et al., 2020; Xie et al., 2025). [3]

Theorem 2.2 (Sensitivity of learnable restart dynamics)**.** *Fix a finite horizon T and an initial state x_0. Suppose F and R are C^1. At θ_0, assume exactly $q < \infty$ restarts occur at $0 < T_1 < \cdots < T_q < T$. For every realized event $0 \le i < q$, assume $\partial_t F_{\mathsf{V}}(L_i; x_i, \theta_0) < 0$. Then L_i and x_{i+1} are C^1 in a neighborhood of θ_0, with recursion*

$$\partial_\theta L_i = -\frac{\partial_\theta F_{\mathsf{V}}(L_i; x_i, \theta) + \partial_x F_{\mathsf{V}}(L_i; x_i, \theta) \cdot \partial_\theta x_i}{\partial_t F_{\mathsf{V}}(L_i; x_i, \theta)}, \tag{10}$$

[3] Throughout, ∂_θ applied to a phase quantity such as L_i, x_i, or $X(t)$ denotes the *total* derivative with respect to θ, while $\partial_\theta F$, $\partial_\theta V$ denote the partial derivatives of the flow and the score in their explicit θ argument; ∂_x and ∂_t are interpreted similarly.

and

$$\partial_\theta x_{i+1} = \partial_t F_{\mathsf{R}}(L_i; x_i, \theta)\, \partial_\theta L_i + \partial_x F_{\mathsf{R}}(L_i; x_i, \theta)\, \partial_\theta x_i + \partial_\theta F_{\mathsf{R}}(L_i; x_i, \theta), \tag{11}$$

where $\partial_x F_{\mathsf{V}}$ denotes the derivative of the score-flow component F_{V} with respect to the phase initialization x (i.e., the second argument), and we denote $F_{\mathsf{R}} := R \circ F_{\mathsf{X}}$.

Consequently, for a fixed global time $t \in [0, T]$ with $t \in (T_i, T_{i+1})$, the state $X(t) = F_{\mathsf{X}}(t - T_i; x_i, \theta)$ is also differentiable with respect to θ, with gradient

$$\partial_\theta X(t) = -\partial_t F_{\mathsf{X}}(t - T_i; x_i, \theta)\, \partial_\theta T_i + \partial_x F_{\mathsf{X}}(t - T_i; x_i, \theta)\, \partial_\theta x_i + \partial_\theta F_{\mathsf{X}}(t - T_i; x_i, \theta), \tag{12}$$

where $\partial_\theta T_i = \sum_{j=0}^{i-1} \partial_\theta L_j$ by definition. The first term is a boundary-shift correction: at a fixed global time, perturbing θ moves the earlier restart boundaries, which changes the local time corresponding to t. A full proof of Theorem 2.2 based on a phase-by-phase induction with the implicit function theorem is deferred to Appendix C.1.

In the remainder of this section, we validate the effectiveness of the gradients. In Section 3, we then develop a discrete-time counterpart of these continuous-time sensitivities that retains the same structure while avoiding the cost of exact ODE differentiation.

2.4 Validation: Optimizing the Restarts in the Continuous-time Dynamics

In the following, we demonstrate the effectiveness of the restart gradients of Theorem 2.2 by optimizing the restarts for the ODE (3). We consider QP problems of the form

$$\min_{x \in \mathbb{R}^d} f(x) = \tfrac{1}{2} x^\top A x - b^\top x + c \tag{13}$$

with dimension $d = 30$ and condition number $\kappa = 80$; the precise generation protocol of f is deferred to Appendix D. On such problems the ODE (3) and the corresponding sensitivity equations can be simulated with high accuracy.

We parameterize the learnable restart score via a small MLP $V(t; \theta) = \mathsf{MLP}(\phi(x(t)); \theta)$, where ϕ is a simple feature map (see Appendix D.1). The parameter θ is optimized with respect to the following loss (integrated function value):

$$J(\theta) \;:=\; \int_0^T f(x(t; \theta))\, dt, \tag{14}$$

with gradients computed through (10)–(12). We compare the learned policy against two handcrafted continuous-time restart criteria, which are continuous analogues of the standard discrete adaptive-restart rules of O'donoghue & Candes (2015):

- **Function restart**: $V_{\mathsf{func}}(t) = -\nabla f(x(t))^\top \dot{x}(t)$, the continuous analogue of the function-based rule "restart iff $f(x_{k+1}) > f(x_k)$".
- **Gradient restart**: $V_{\mathsf{grad}}(t) = -\nabla f(x(t) + \sqrt{s}\, \dot{x}(t))^\top \dot{x}(t)$, the continuous analogue of the gradient-based rule "restart iff $\nabla f(y_k)^\top (x_{k+1} - x_k) > 0$".

Figure 1 (*right*) reports the objective gap along the optimization trajectories induced by the hand-designed restart policies and the learned restart policy. The learned restart policy achieves a better final gap $f(x(t)) - \min f$ than the baseline restart criteria, and its convergence is more stable. As a reference-only ceiling, we also plot the strongly-convex accelerated ODE that uses the true strong-convexity parameter μ (detailed in Appendix D.1).[4] This validates the learnability of the restart criterion in the continuous-time regime via the gradients derived in Section 2.3; more extensive experiments in the discrete-time setting are deferred to Section 4.1.

3 Efficient Discrete Approximation of Restart Sensitivities

To apply the insights of Section 2.3 to practical optimization algorithms, which operate in *discrete* time, we now develop the discrete analogue of the continuous-time restart gradients. Concretely, in

[4]This last curve relies on knowing μ; adaptive restart is precisely the mechanism that recovers the same exponential convergence rate without such knowledge.

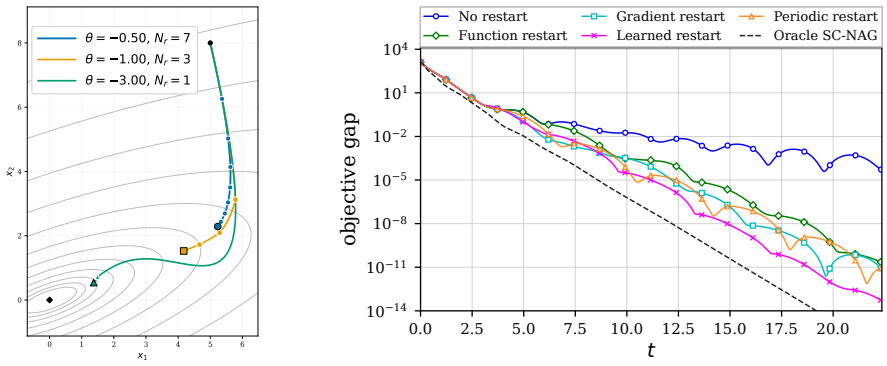


Figure 1: Continuous-time validation on the QP family of (13). *Left:* ODE trajectories induced by (3) under different restart policies on a 2-dimensional quadratic instance, illustrating how the learned policy reshapes the iterate path. *Right:* objective gap $f(x(t)) - \min f$ along the optimization trajectories induced by the hand-designed and learned restart policies ($d = 30$, $\kappa = 80$). The learned restart policy consistently improves over both hand-designed criteria and approaches the strongly-convex reference.

this section we introduce a notion of *surrogate gradients* for both the restart step (Definition 3.2) and the restarted trajectory in the discrete setting, and we show that the surrogate gradients closely approximate the continuous-time gradient of Definition 2.1 whenever the discrete dynamics are sufficiently close to the continuous-time dynamics.

Following the notation in Section 2, we assume that the discrete-time dynamics are described by a flow map $\bar{F} = (\bar{F}_{\mathsf{X}}, \bar{F}_{\mathsf{V}})$ as follows.

Definition 3.1 (Discrete restarted trajectory)**.** The restarted discrete trajectory $(\bar{X}(n) = \bar{X}(n;\theta))_{n\geq 0}$ is defined recursively, starting with $N_0 = 0$ and $\bar{x}_0 = x_0$. Assume $\bar{F}_{\mathsf{X}}(0; x, \theta) = x$ and $\bar{F}_{\mathsf{V}}(0; x, \theta) > 0$ at every phase initialization. For each phase i with starting step N_i and initial point $\bar{x}_i := \bar{X}(N_i)$, define

$$\bar{X}(N_i + k) = \bar{F}_{\mathsf{X}}(k; \bar{x}_i, \theta), \qquad \bar{V}(N_i + k) = \bar{F}_{\mathsf{V}}(k; \bar{x}_i, \theta), \tag{15}$$

for each $0 \leq k < S_i$, where the discrete restart time S_i is defined as

$$S_i := \min\{k \geq 1 : \bar{F}_{\mathsf{V}}(k; \bar{x}_i, \theta) \leq 0\}, \qquad \min \varnothing := +\infty. \tag{16}$$

For finite S_i, we then set $N_{i+1} = N_i + S_i$ and $\bar{x}_{i+1} = \bar{X}(N_{i+1}) = R(\bar{F}_{\mathsf{X}}(S_i; \bar{x}_i, \theta))$.

As we have mentioned, the integer-valued restart step S_i is locally constant in θ, and therefore does not admit a non-trivial ordinary derivative. To define a meaningful surrogate, we assume that $\bar{F}$ is a discretization of a continuous-time system in the sense of Definition 2.1: we consider the time grid $t_n = nh$ for $n = 0, 1, \ldots, N$, where $h > 0$ is the discretization stepsize and N is the total number of steps with $T = Nh$. Then, following Definition 2.1, we can introduce the following surrogate gradients.

Definition 3.2 (Surrogate gradient)**.** For a finite first crossing S_i, let $\Delta_i := \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) - \bar{F}_{\mathsf{V}}(S_i - 1; \bar{x}_i, \theta)$, which satisfies $\Delta_i < 0$. Given $\mathsf{grad}(\bar{x}_i)$, define the formal step sensitivity $\mathsf{grad}(S_i)$ by

$$\mathsf{grad}(S_i) = -\frac{\partial_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) + \partial_x \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i)}{\Delta_i}. \tag{17}$$

The restart-state sensitivity is defined by

$$\mathsf{grad}(\bar{x}_{i+1}) = \big(\bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) - \bar{F}_{\mathsf{R}}(S_i - 1; \bar{x}_i, \theta)\big)\mathsf{grad}(S_i) + \partial_x \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i) + \partial_\theta \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta), \tag{18}$$

where we denote $\bar{F}_{\mathsf{R}} := R \circ \bar{F}_{\mathsf{X}}$. For interior global steps $N_i < n < N_{i+1}$, the global-trajectory sensitivity is defined by the discrete analogue of (12),

$$\begin{aligned}\mathsf{grad}(\bar{X}(n)) = &-\big(\bar{F}_{\mathsf{X}}(n - N_i; \bar{x}_i, \theta) - \bar{F}_{\mathsf{X}}(n - 1 - N_i; \bar{x}_i, \theta)\big)\mathsf{grad}(N_i) \\ &+ \partial_x \bar{F}_{\mathsf{X}}(n - N_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i) + \partial_\theta \bar{F}_{\mathsf{X}}(n - N_i; \bar{x}_i, \theta),\end{aligned} \tag{19}$$

where $\mathsf{grad}(N_i) = \sum_{j<i} \mathsf{grad}(S_j)$ accumulates the boundary-shift surrogate gradients from previous phases.

Intuition We can mimic the continuous derivation in Section 2.3 and treat the discrete event equation $\bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \approx 0$ as an implicit constraint and apply the implicit function theorem formally, replacing the derivative with respect to the (integer) step index by a backward difference. Symbolically, using the chain rule,

$$\begin{aligned} 0 \;\approx\; \nabla_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \;\approx\; \partial_k \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \cdot \partial_\theta S_i + \partial_x \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \cdot \partial_\theta \bar{x}_i + \partial_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \\ \approx (\bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) - \bar{F}_{\mathsf{V}}(S_i - 1; \bar{x}_i, \theta)) \cdot \partial_\theta S_i + \partial_x \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \cdot \partial_\theta \bar{x}_i + \partial_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta), \end{aligned}$$

where $\partial_k \bar{F}_{\mathsf{V}}$, $\partial_\theta S_i$, and $\partial_\theta \bar{x}_i$ are formal symbols: they do not exist in the strict sense, since S_i is integer-valued and $\bar{x}_i$ inherits this non-differentiability. Although this manipulation is purely formal, it exposes the right structure: S_i plays the role of the continuous restart time L_i, with the time derivative $\partial_t V$ replaced by the discrete backward difference Δ_i defined below. Solving this formal equation for $\partial_\theta S_i$ then motivates Definition 3.2. We emphasize that this derivation is purely intuitive and does not by itself justify the surrogate; the actual theoretical guarantee comes from connecting Definition 3.2 to the continuous-time gradients of Definition 2.1, which is the subject of Theorem 3.4.

Computational efficiency The sensitivities defined in (17)–(18) are designed to be plugged directly into a standard automatic-differentiation framework (Appendix B.2). Appendix B.3 gives a single-pass forward-sensitivity implementation, including the restart and boundary-shift updates, with storage independent of trajectory length.

Theoretical guarantees We next show that the surrogate gradients are *accurate*: provided that the discrete flow map $\bar{F}$ approximates F in the quantitative sense of the following assumption, the surrogate gradients of the restart time, the restart state, and the global trajectory all converge to their continuous counterparts.

Assumption 3.3. Assume that there are constants $\gamma, M > 0$ such that the following holds.

(1) $\partial_t F_{\mathsf{V}}(L_i; x_i, \theta) \le -\gamma$ for each i.

(2) Smoothness: $\|\partial_{(t,x,\theta)} F\| + \|\nabla R\| \le M$, and $\partial_{(t,x,\theta)} F$ and ∇R are both M-Lipschitz.

(3) Consistency: the flow maps satisfy the following for any (x, θ) and $0 \le k \le N$

$$\|\bar{F}(k; x, \theta) - F(t_k; x, \theta)\| + \|\partial_{(x,\theta)} \bar{F}(k; x, \theta) - \partial_{(x,\theta)} F(t_k; x, \theta)\| \le \epsilon_h, \tag{20}$$

$$\left\| \tfrac{1}{h} \big(\bar{F}(k; x, \theta) - \bar{F}(k-1; x, \theta) \big) - \partial_t F(t_k; x, \theta) \right\| \le \epsilon_h, \tag{21}$$

where $\epsilon_h \to 0$ as $h \to 0$.

Theorem 3.4 (Closeness of surrogate gradients to continuous gradients)**.** *Suppose that Assumption 3.3 holds and h is sufficiently small. Then, the surrogate gradient in Definition 3.2 satisfies*

$$|hS_i - L_i| + \|h\mathsf{grad}(S_i) - \partial_\theta L_i\| = O(\epsilon_h + h), \tag{22}$$

$$\|\bar{x}_{i+1} - x_{i+1}\| + \|\mathsf{grad}(\bar{x}_{i+1}) - \partial_\theta x_{i+1}\| = O(\epsilon_h + h), \tag{23}$$

$$\|\bar{X}(n) - X(t_n)\| + \|\mathsf{grad}(\bar{X}(n)) - \partial_\theta X(t_n)\| = O(\epsilon_h + h), \tag{24}$$

for any phase $i \ge 0$ and $n \in [N]$.

The precise phase and sampling scope, including endpoint restrictions, and the more general local formulation are given in Appendix C.3. Finally, in Appendix D.2, we empirically validate the surrogate gradient of Definition 3.2 against its continuous-time counterpart in a setting where both can be computed accurately.

Policy improvement guarantees Built upon our framework, we establish second-order smoothness of the optimization dynamics with respect to the restart policy parameter (Appendix C.1). In particular, projected gradient descent (PGD) on the policy parameter with an appropriate stepsize is guaranteed to improve, or at least preserve, performance on the problems used for training. Therefore, if the restart policy is initialized at a given restart method, the learned policy is guaranteed not to perform worse than the given method on the empirical fixed-horizon training objective.

In addition, the convergence guarantees of the restarted dynamics can be inherited from the optimization backbone. Therefore, the learned policy can exploit accelerated NAG/PDLP dynamics when beneficial, while the worst case is bounded by the convergence guarantee of vanilla gradient methods.

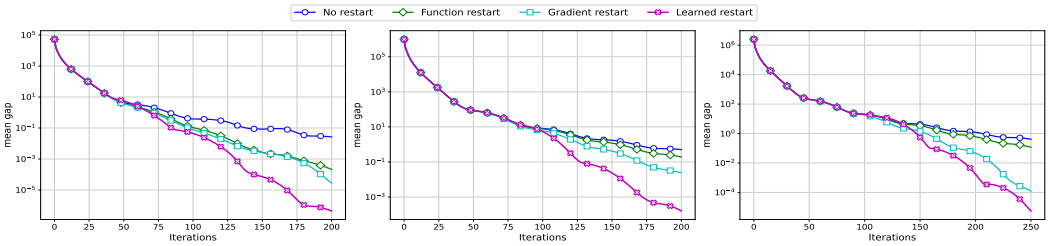


Figure 2: Learning a restart policy for NAG on the QP family (13): mean objective gap $f(x_k) - \min f$ along the trajectory for a policy trained on $d = 20$ and evaluated on $d = 20$ (*left*), 400 (*middle*), and 1000 (*right*).

Table 1: Terminal gap $f(x_N) - f^\star$ on real-measurement CT ridge reconstruction (77) ($N = 800$; geometric mean over 8 initial points). Per dataset, the two rows are the regularization levels $\lambda_{\text{rel}} = 3 \cdot 10^{-4}$ and 10^{-4}. Details in Appendix F.

Dataset	No restart	Function	Gradient	Learned
walnut82	6.90e-09	2.46e-09	5.17e-11	**1.76e-13**
	1.72e-08	1.72e-08	1.72e-08	**2.65e-10**
cheeseL45	1.56e-07	9.75e-11	4.91e-11	**1.97e-12**
	5.72e-07	5.81e-08	5.01e-08	**8.95e-09**
cheeseF45	2.00e-07	3.50e-11	3.23e-11	**2.27e-12**
	7.81e-07	8.35e-08	7.92e-08	**1.77e-08**

4 Experiments

We conduct experiments on two problem classes, QP with the NAG backbone and LP with the PDHG backbone of PDLP. All problem constructions, feature maps, and training hyperparameters are deferred to Appendices D to G.

4.1 Learning to Restart for QP

Throughout this subsection, the optimizer is NAG and we compare the learned restart policy against two standard handcrafted adaptive restart criteria from O'donoghue & Candes (2015): **Function restart** (restart iff $f(x_{k+1}) > f(x_k)$) and **Gradient restart** (restart iff $\nabla f(y_k)^\top (x_{k+1} - x_k) > 0$), together with **No restart**.

Synthetic quadratics. We evaluate the proposed differentiable restart mechanism by learning a restart policy for NAG on a family of diagonal QPs (13). The restart policy is a small two-layer MLP (4-32-1) taking four phase-level features as input. Training uses 20-dimensional problems with a two-stage procedure: warm-start from function-restart teacher labels, followed by end-to-end fine-tuning on a ratio objective. Full details are in Appendix D.3. Figure 2 reports the mean objective gap along the optimization trajectory, averaged over test problems. Results on unseen condition numbers are reported in Appendix D.3.

Zero-shot transfer to real data. We next ask whether a policy trained *only on synthetic QPs* transfers to real problems without retraining. We train a 7-32-1 MLP on synthetic quadratics with the same two-stage procedure and test it on two real problem families: ℓ_2-regularized logistic regression on five LIBSVM datasets (Chang & Lin, 2011) and ridge reconstruction from physically measured X-ray tomography data (Hämäläinen et al., 2015; Bubba et al., 2017) in dimensions up to $d = 16384$. Although both tasks are out of distribution, the learned policy still attains the smallest terminal gap in every configuration. Table 1 reports the tomography results; the logistic-regression results and all experimental details are given in Appendices E and F.

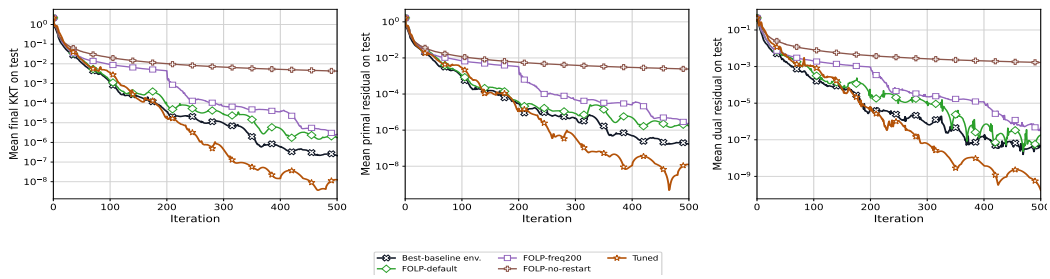


Figure 3: KKT residual (*left*), primal residual (*middle*), and dual residual (*right*) vs. iteration for the learned restart policy and PDLP-style baselines on synthetic LPs (dimension $n = 12$, $m = 6$).

Table 2: Netlib LP benchmark: median final residual under a fixed iteration budget; n is the number of retained instances per tier.

Tier	Residual	Learned	PDLP default	Periodic-200	No restart	vs. default
small n=32	KKT	**2.29e-3**	3.45e-3	4.62e-2	1.80e-1	1.51×
	primal	**1.73e-5**	1.85e-5	5.06e-3	1.48e-4	1.07×
	dual	**3.16e-6**	4.47e-6	6.73e-6	6.14e-5	1.41×
mid n=24	KKT	**4.83e-4**	1.05e-3	2.37e-2	1.60e-2	2.16×
	primal	**5.61e-5**	9.88e-5	4.12e-3	2.12e-4	1.76×
	dual	**3.86e-6**	9.91e-6	5.26e-5	6.87e-5	2.57×
large n=10	KKT	**1.25e-3**	1.43e-2	3.63e-2	2.26e-1	11.45×
	primal	**8.71e-7**	3.18e-6	6.87e-4	2.38e-5	3.65×
	dual	**4.85e-5**	3.31e-4	8.71e-4	5.68e-5	6.82×

4.2 Learning to Restart for LP

We next apply our framework to primal–dual hybrid gradient (PDHG) methods on standard-form LPs, $\min_{x \in \mathbb{R}^n} c^\top x$ s.t. $Ax = b$, $x \geq 0$, using a Python reimplementation of the PDHG backbone and adaptive restart scheme of PDLP (Chambolle & Pock, 2011; Applegate et al., 2021); the update and restart rule are given in Appendix G. Throughout, we compare against the PDLP restart rule, periodic restarting, and no restarting, together with their instance-wise best envelope. The training procedure is the same as for QP; all training details are in Appendix G.

Synthetic LPs. We parametrize the restart score by a small MLP, $\bar{V}_k = \mathsf{MLP}\big(\phi(\bar{x}_k, \bar{x}_{N_i}, k); \theta\big)$, with 183 trainable parameters. Figure 3 reports the KKT, primal, and dual residuals on held-out instances: the learned policy improves upon the best-baseline envelope in all three metrics, and the improvement carries over to larger unseen instances (6×18) without retraining (Figure 6).

Netlib benchmark. Finally, we train a restart policy directly on real instances from the Netlib LP benchmark. Here we deliberately use a *minimal* policy class: the restart score is an affine function of 15 log-scale phase features, with only 16 trainable parameters. The policy is trained with the surrogate gradients on the 7 smallest instances and evaluated on the remaining instances across three problem-size tiers (Appendix G.4). Table 2 reports the result. The learned policy is best in every tier and residual.

5 Conclusion

We developed a differentiable framework for learning restart policies. Restart events are non-differentiable in discrete time but differentiable in the continuous-time limit; we derived the continuous-time gradients in closed form, introduced a discrete surrogate gradient that fits into standard automatic differentiation, and proved that it converges to the continuous-time gradient. Learned restart policies outperform handcrafted restart criteria on quadratic and linear programs, and transfer to unseen problem sizes and to real problems without retraining.

AI DISCLOSURE

In this work, we used AI tools to polish the paper and write code for generating the figures based on experimental data. We have reviewed all AI-assisted work carefully. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3981–3989, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/fb87582825f9d28a8d42c5e5e5e8b23d-Abstract.html.

David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Practical large-scale linear programming using primal-dual hybrid gradient. *Advances in Neural Information Processing Systems*, 34:20243–20257, 2021.

David Applegate, Oliver Hinder, Haihao Lu, and Miles Lubin. Faster first-order primal-dual methods for linear programming using restarts and sharpness. *Mathematical Programming*, 201(1):133–184, 2023.

David Applegate, Mateo Díaz, Oliver Hinder, Haihao Lu, Miles Lubin, Brendan O'Donoghue, and Warren Schudy. Pdlp: A practical first-order method for large-scale linear programming. *arXiv preprint arXiv:2501.07018*, 2025.

Kenneth Joseph Arrow, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in linear and non-linear programming*, volume 2. Stanford University Press Stanford, 1958.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Quentin Bertrand, Quentin Klopfenstein, Mathurin Massias, Mathieu Blondel, Samuel Vaiter, Alexandre Gramfort, and Joseph Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *Journal of Machine Learning Research*, 23(149):1–43, 2022.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5230–5242, 2022.

Jérôme Bolte, Tam Le, Edouard Pauwels, and Antonio Silveti-Falls. Nonsmooth implicit differentiation for machine learning and optimization, 2022. URL https://arxiv.org/abs/2106.04350.

Tatiana A. Bubba, Markus Juvonen, Jonatan Lehtonen, Maximilian März, Alexander Meaney, Zenith Purisha, and Samuli Siltanen. Tomographic X-ray data of carved cheese, 2017. URL https://doi.org/10.5281/zenodo.1254210.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Learning neural event functions for ordinary differential equations. *arXiv preprint arXiv:2011.03902*, 2020.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

Xiaohan Chen, Jialin Liu, and Wotao Yin. Learning to optimize: A tutorial for continuous and mixed-integer optimization. *Science China Mathematics*, 67(6):1191–1262, 2024.

Ashish Cherukuri, Enrique Mallada, and Jorge Cortés. Asymptotic convergence of constrained primal–dual dynamics. *Systems & Control Letters*, 87:10–15, 2016.

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

Olivier Fercoq. Quadratic error bound of the smoothed gap and the restarted averaged primal-dual hybrid gradient. *arXiv preprint arXiv:2206.03041*, 2022.

Olivier Fercoq and Zheng Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. *IMA Journal of Numerical Analysis*, 39(4):2069–2095, 2019.

Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pp. 3–33. Springer International Publishing, 2019.

Pontus Giselsson and Stephen Boyd. Monotonicity and restart in fast gradient methods. In *53rd IEEE Conference on Decision and Control*, pp. 5058–5063. IEEE, 2014.

Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE control systems magazine*, 29(2):28–93, 2009.

Keijo Hämäläinen, Lauri Harhanen, Aki Kallonen, Antti Kujanpää, Esa Niemi, and Samuli Siltanen. Tomographic X-ray data of a walnut, 2015. URL `https://doi.org/10.5281/zenodo.1254206`.

Uwe Helmke and John B Moore. *Optimization and dynamical systems*. Springer Science & Business Media, 2012.

Oliver Hinder. Worst-case analysis of restarted primal-dual hybrid gradient on totally unimodular linear programs. *Operations Research Letters*, 57:107199, 2024.

Maxime Laborde and Adam Oberman. A lyapunov analysis for accelerated gradient methods: From deterministic to stochastic case. In *International conference on artificial intelligence and statistics*, pp. 602–612. PMLR, 2020.

Bowen Li and Bin Shi. Understanding the pdhg algorithm via high-resolution differential equations. *arXiv preprint arXiv:2403.11139*, 2024.

Jialin Liu, Xiaohan Chen, Zhangyang Wang, Wotao Yin, and HanQin Cai. Towards constituting mathematical structures for learning to optimize. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 21426–21449, 2023.

Haihao Lu and Jinwen Yang. On the geometry and refined rate of primal–dual hybrid gradient for linear programming. *Mathematical Programming*, 212(1):349–387, 2025.

Kaifeng Lv, Shunhua Jiang, and Jian Li. Learning gradient descent: Better generalization and longer horizons. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2247–2255. PMLR, 2017. URL `http://proceedings.mlr.press/v70/lv17a.html`.

Céline Moucer, Adrien Taylor, and Francis Bach. A systematic approach to lyapunov analyses of continuous-time models in convex optimization. *SIAM Journal on Optimization*, 33(3):1558–1586, 2023.

Brendan O'donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

Keith Pilbeam. *Finance and financial markets*. Bloomsbury Publishing, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Bin Shi, Simon S Du, Michael I Jordan, and Weijie J Su. Understanding the acceleration phenomenon via high-resolution differential equations. *Mathematical Programming*, 195:79–148, 2022.

Weijie Su, Stephen Boyd, and Emmanuel J Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17(1): 5312–5354, 2016.

Hamdy A Taha. *Operations research: an introduction*. Prentice hall Upper Saddle River, NJ, 6 edition, 1997.

Arjan J Van Der Schaft and Hans Schumacher. *An introduction to hybrid dynamical systems*, volume 251. springer, 2007.

Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

Zhonglin Xie, Wotao Yin, and Zaiwen Wen. ODE-based Learning to Optimize, 2024. URL https://arxiv.org/abs/2406.02006.

Zhonglin Xie, Yiman Fong, Haoran Yuan, and Zaiwen Wen. Accelerating optimization via differentiable stopping time. *arXiv preprint arXiv:2505.22509*, 2025.

Ming Xu, Timothy L Molloy, and Stephen Gould. Revisiting implicit differentiation for learning problems in optimal control. In *Advances in Neural Information Processing Systems*, volume 36, pp. 60060–60076, 2023.

Junjie Yang, Tianlong Chen, Mingkang Zhu, Fengxiang He, Dacheng Tao, Yingbin Liang, and Zhangyang Wang. Learning to generalize provably in learning to optimize, 2023a. URL https://arxiv.org/abs/2302.11085.

Junjie Yang, Xuxi Chen, Tianlong Chen, Zhangyang Wang, and Yingbin Liang. M-l2o: Towards generalizable learning-to-optimize by test-time fast self-adaptation. *arXiv preprint arXiv:2303.00039*, 2023b.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2025. URL https://arxiv.org/abs/2303.18223.

A Related Work

Adaptive restarts. For accelerated gradient methods, O'donoghue & Candes (2015) introduced two simple adaptive restart criteria and showed that they recover a linear rate on strongly convex problems without knowledge of the strong-convexity constant. Convergence guarantees for these and related rules have then been refined in (Giselsson & Boyd, 2014; Fercoq & Qu, 2019; Fercoq, 2022). For primal–dual methods on linear programming, adaptive restart policies are the central mechanism behind the practical performance of PDLP and its variants (Applegate et al., 2021; 2023; Lu & Yang, 2025), with worst-case and sharpness-based analyses developed in (Hinder, 2024; Applegate et al., 2023). All of these rules are derived analytically and are governed by a small number of hand-tuned parameters.

Learning to optimize. L2O is a paradigm in which optimization algorithms are designed by training a parameterized model on a distribution of problem instances; see (Chen et al., 2022; 2024) for surveys. One line of work parameterizes either the per-step update directly by a neural network (Andrychowicz et al., 2016; Lv et al., 2017) or the algorithm's coefficients by a small parametric family motivated by an ODE limit (Xie et al., 2024). Another line of work studies when L2O can come with convergence and generalization guarantees (Liu et al., 2023; Yang et al., 2023a;b). The dominant training objective in both lines is to minimize the loss reached after a fixed iteration budget in the absence of *restarts*, which is differentiable in the algorithm parameters and admits standard gradient-based meta-learning.

ODE perspectives on optimization. The gradient flow perspective provides both theoretical justification and a practical design guide for gradient-based algorithms (e.g., Helmke & Moore, 2012; Wibisono et al., 2016) by relating discrete iterations to their continuous-time limit. Lyapunov analyses of these continuous-time models translate directly into convergence guarantees for the corresponding discrete iterations (Laborde & Oberman, 2020; Moucer et al., 2023), and a canonical instance is the ODE limit of Nesterov's accelerated gradient methods (Su et al., 2016; Shi et al., 2022). Building on this perspective, recent work (Xie et al., 2024; 2025) formulates a differentiable notion of stopping time for iterative algorithms and uses it to optimize convergence speed against a target accuracy. The underlying machinery—differentiating through state-dependent events of an ODE solve—is closely related to neural event functions for ODEs (Chen et al., 2020) and, more broadly, to implicit differentiation through optimization procedures (Blondel et al., 2022; Bertrand et al., 2022; Bolte et al., 2022; Xu et al., 2023).

B Additional Details

B.1 Nesterov's Accelerated Gradient with Restarts

On the smooth convex minimization problem $\min_{x\in\mathbb{R}^d} f(x)$ with stepsize $s > 0$, the standard NAG iteration with momentum coefficient $\beta_k = (k-1)/(k+2)$ can be written,[5] in terms of the iterate x_k and a velocity-like momentum variable $v_k := (x_k - x_{k-1})/\sqrt{s}$, as

$$\begin{aligned} y_k &= x_k + \beta_k \sqrt{s}\, v_k, \\ x_{k+1} &= y_k - s\,\nabla f(y_k), \\ v_{k+1} &= (x_{k+1} - x_k)/\sqrt{s}. \end{aligned} \tag{25}$$

The state in the abstract flow-map formulation is the pair $\bar{x} = (x, v)$, and the restart map zeroes the velocity:

$$R(x, v) = (x, 0). \tag{26}$$

With $h = \sqrt{s}$, the iteration (25) is a discretization of the high-resolution ODE (3) of Shi et al. (2022); consistency for initial-position derivatives and time differences away from phase starts is verified in Appendix C.5.

The within-phase update (25) is augmented with a restart score $\bar{V}(\,\cdot\,;\theta_{\mathrm{rst}})$ that triggers a reset whenever it crosses zero. Algorithm 1 summarizes the resulting iteration.

[5] An equivalent formulation uses the auxiliary sequence $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$ with $\beta_k = (t_k - 1)/t_{k+1}$, which yields the same asymptotic behaviour.

Algorithm 1 NAG with adaptive restart.

Require: Initial point $x_0 \in \mathbb{R}^d$, stepsize $s > 0$, restart score $\bar{V}(\,\cdot\,;\theta_{\mathrm{rst}})$, horizon N.
Set $x_{-1} \leftarrow x_0$, $v_0 \leftarrow 0$, phase counter $k_{\mathrm{rst}} \leftarrow 0$, anchor $x_{\mathrm{anc}} \leftarrow x_0$.
for $k = 0, 1, \ldots, N-1$ **do**
  $\beta_k \leftarrow (k - k_{\mathrm{rst}} - 1)/(k - k_{\mathrm{rst}} + 2)$ {phase-local momentum coefficient}
  $y_k \leftarrow x_k + \beta_k \sqrt{s}\, v_k$
  $x_{k+1} \leftarrow y_k - s\, \nabla f(y_k)$
  $v_{k+1} \leftarrow (x_{k+1} - x_k)/\sqrt{s}$
  if $\bar{V}\big(k+1, x_{k+1}, v_{k+1}, x_{\mathrm{anc}}; \theta_{\mathrm{rst}}\big) \le 0$ **then**
    $v_{k+1} \leftarrow 0,\ k_{\mathrm{rst}} \leftarrow k+1,\ x_{\mathrm{anc}} \leftarrow x_{k+1}$ {restart: zero momentum, reset phase clock and anchor}
  end if
end for
return x_N

The two standard hand-crafted restart rules of O'donoghue & Candes (2015), used as baselines in Sections 2.4 and 4.1, are recovered by the following choices of $\bar{V}$:

- *Function restart*: restart iff $f(x_{k+1}) > f(x_k)$, equivalently $\bar{V}_k^{\mathrm{func}} = f(x_k) - f(x_{k+1})$.
- *Gradient restart*: restart iff $\nabla f(y_k)^\top (x_{k+1} - x_k) > 0$, equivalently $\bar{V}_k^{\mathrm{grad}} = -\nabla f(y_k)^\top (x_{k+1} - x_k)$.

The continuous-time analogues used in Section 2.4 are obtained by replacing $x_{k+1} - x_k$ by $\sqrt{s}\,\dot{x}(t)$ and y_k by $x(t) + \sqrt{s}\,\dot{x}(t)$:

$$V_{\mathsf{func}}(t) = -\nabla f(x(t))^\top \dot{x}(t), \qquad V_{\mathsf{grad}}(t) = -\nabla f(x(t) + \sqrt{s}\,\dot{x}(t))^\top \dot{x}(t).$$

B.2 Implementation via Automatic Differentiation

We now describe how the surrogate gradients of Definition 3.2 can be realized as a single backward pass of a standard automatic-differentiation engine, with no manual chain-rule bookkeeping across phases. In what follows it is essential to keep two distinct gradient operators in mind:

- $\mathsf{grad}(\cdot)$ denotes the *surrogate gradients* introduced in Definition 3.2. It is a mathematical object defined by the recursion (17)–(18), and is in general not the strict derivative of any quantity.
- $\mathsf{autograd}(\cdot)$ denotes the *autodiff gradient* produced by an autodiff engine (e.g., JAX or PyTorch) on a given computation graph; it is the exact derivative of whatever expression it is applied to.
- $\mathsf{stopgrad}(\cdot)$ denotes the stop-gradient operator: it acts as the identity in the forward pass and detaches its argument from the computation graph in the backward pass.

Our goal is to construct, at each phase i, two auxiliary quantities—a *soft restart step* $\hat{S}_i$ and a *soft phase-end state* $\hat{x}_{i+1}$—that are numerically identical to S_i and $\bar{x}_{i+1}$ on the forward pass, but whose autodiff gradients reproduce the symbolic sensitivities of Definition 3.2:

$$\mathsf{autograd}(\hat{S}_i) = \mathsf{grad}(S_i), \qquad \mathsf{autograd}(\hat{x}_{i+1}) = \mathsf{grad}(\bar{x}_{i+1}). \tag{27}$$

We assume inductively that the engine has been initialized so that $\mathsf{autograd}(\bar{x}_i) = \mathsf{grad}(\bar{x}_i)$ at the start of phase i.

Soft restart step and soft phase-end state. Building on the construction in Section 3, at each phase i we define the *soft restart time*

$$\tau_i := \frac{\bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)}{\mathsf{stopgrad}(\Delta_i)} = \frac{-\bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)}{\mathsf{stopgrad}\big(\bar{F}_{\mathsf{V}}(S_i - 1; \bar{x}_i, \theta) - \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)\big)}, \tag{28}$$

and the interpolation mask

$$m_i := 1 - \tau_i + \mathsf{stopgrad}(\tau_i). \tag{29}$$

With this notation, the soft restart step and the soft phase-end state at phase i are

$$\hat{S}_i := (S_i - 1) + m_i, \qquad \hat{x}_{i+1} := m_i\, \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) + (1 - m_i)\, \bar{F}_{\mathsf{R}}(S_i - 1; \bar{x}_i, \theta), \tag{30}$$

where, as in Definition 3.2, $\bar{F}_{\mathsf{R}} = R \circ \bar{F}_{\mathsf{X}}$.

Both $\hat{S}_i$ and $\hat{x}_{i+1}$ have a transparent forward-pass behavior. Since τ_i and $\mathsf{stopgrad}(\tau_i)$ share the same numerical value, the mask satisfies $m_i = 1$ in the forward pass, and therefore

$$\hat{S}_i = S_i, \qquad \hat{x}_{i+1} = \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) = \bar{x}_{i+1}.$$

That is, replacing the integer-valued boundary by its soft counterpart leaves the trajectory unchanged on the forward pass, so $\hat{x}_{i+1}$ can be fed as the phase-end state into the next phase without altering the simulation.

Backward-pass agreement with the symbolic sensitivities. With S_i held fixed and $\mathsf{autograd}(\bar{x}_i) = \mathsf{grad}(\bar{x}_i)$,

$$\begin{aligned}
\mathsf{autograd}\big(\bar{F}_{\mathsf{X}}(S_i; \bar{x}_i, \theta)\big) &= \partial_x \bar{F}_{\mathsf{X}}(S_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i) + \partial_\theta \bar{F}_{\mathsf{X}}(S_i; \bar{x}_i, \theta), \\
\mathsf{autograd}\big(\bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)\big) &= \partial_x \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i) + \partial_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta).
\end{aligned} \tag{31}$$

Using $\mathsf{autograd}(\mathsf{stopgrad}(\tau_i)) = \mathsf{autograd}(\mathsf{stopgrad}(\Delta_i)) = 0$,

$$\mathsf{autograd}(m_i) = -\mathsf{autograd}(\tau_i) = -\frac{\partial_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) + \partial_x \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i)}{\Delta_i} = \mathsf{grad}(S_i),$$

$$\mathsf{autograd}(\hat{S}_i) = \mathsf{autograd}(m_i) = \mathsf{grad}(S_i).$$

Since $m_i = 1$ on the forward pass,

$$\begin{aligned}
\mathsf{autograd}(\hat{x}_{i+1}) &= \big(\bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) - \bar{F}_{\mathsf{R}}(S_i - 1; \bar{x}_i, \theta)\big)\mathsf{autograd}(m_i) \\
&\quad + \mathsf{autograd}\big(\bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta)\big) \\
&= \big(\bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) - \bar{F}_{\mathsf{R}}(S_i - 1; \bar{x}_i, \theta)\big)\mathsf{grad}(S_i) \\
&\quad + \partial_x \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta)\mathsf{grad}(\bar{x}_i) + \partial_\theta \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) \\
&= \mathsf{grad}(\bar{x}_{i+1}).
\end{aligned} \tag{32}$$

Implementation summary. Putting the pieces together, the implementation of phase i reduces to: (i) run the discrete dynamics until $\bar{F}_{\mathsf{V}}$ first crosses zero, recording S_i and the snapshots $\bar{F}_{\mathsf{X}}(S_i - 1)$, $\bar{F}_{\mathsf{X}}(S_i)$, $\bar{F}_{\mathsf{V}}(S_i - 1)$, $\bar{F}_{\mathsf{V}}(S_i)$; (ii) build τ_i and m_i via (28)–(29); (iii) apply the restart map R to obtain $\bar{F}_{\mathsf{R}}(S_i - 1)$ and $\bar{F}_{\mathsf{R}}(S_i)$ and feed the soft state $\hat{x}_{i+1}$ from (30) into the next phase. The stop-gradient inserted across the phase boundary is what makes the backward pass substitute the symbolic sensitivity (17) for the formally undefined derivative of the integer step S_i, and the soft-mask construction makes this substitution invisible from the user side: the resulting code is a drop-in replacement for the unrolled discrete loop and inherits the approximation guarantee of Theorem 3.4.

B.3 Efficient Implementation by Forward Sensitivity Propagation

The surrogate recursions admit a single forward pass that propagates sensitivities alongside the optimizer state. This realizes the same quantities as Appendix B.2 without retaining a computation graph for the whole trajectory. We describe the propagation both within a phase and across a restart, including the correction for evaluating a trajectory at a fixed global step.

Within a phase. Let $p = \dim \theta$ and let $Z_k \in \mathbb{R}^m$ contain the current optimizer state X_k, restart score V_k, and any auxiliary variables needed for a one-step update. Within phase i, write

$$Z_{k+1} = \Phi_k(Z_k; \theta), \qquad Z_0 = \mathcal{I}(\bar{x}_i; \theta).$$

Here k is the local step index, held fixed when differentiating. If the update or score depends explicitly on the phase-start state, include that state as a constant component of Z_k. The initialization map $\mathcal{I}$ sets the score and auxiliary variables, including any prescribed initial momentum. Let $G_i = \mathsf{grad}(\bar{x}_i) \in \mathbb{R}^{D \times p}$ and initialize

$$H_0 = D_x \mathcal{I}(\bar{x}_i; \theta) G_i + D_\theta \mathcal{I}(\bar{x}_i; \theta). \tag{33}$$

Propagate the $m \times p$ sensitivity matrix by

$$H_{k+1} = D_Z \Phi_k(Z_k; \theta) H_k + D_\theta \Phi_k(Z_k; \theta). \tag{34}$$

This is the ordinary chain rule at fixed local indices, with the incoming surrogate G_i as its initial tangent. In particular, its state and score rows satisfy

$$H_k^{\mathsf{X}} = \partial_x \bar{F}_{\mathsf{X}}(k; \bar{x}_i, \theta) G_i + \partial_\theta \bar{F}_{\mathsf{X}}(k; \bar{x}_i, \theta), \qquad H_k^{\mathsf{V}} = \partial_x \bar{F}_{\mathsf{V}}(k; \bar{x}_i, \theta) G_i + \partial_\theta \bar{F}_{\mathsf{V}}(k; \bar{x}_i, \theta).$$

An evaluated score can either be included in Φ_k or differentiated directly by this same chain rule. The rows of H_k are phase-local sensitivities; the global-time correction below is applied when an output is evaluated.

At a restart. Keep the preceding state and score until the next step has been checked. When the first crossing occurs at $k = S_i$, set

$$\Delta_i = V_k - V_{k-1} < 0, \qquad d_i = -H_k^{\mathsf{V}} / \Delta_i = \mathsf{grad}(S_i), \tag{35}$$

$$\bar{x}_{i+1} = R(X_k), \qquad G_{i+1} = DR(X_k) H_k^{\mathsf{X}} + \big(R(X_k) - R(X_{k-1})\big) d_i. \tag{36}$$

The second term in G_{i+1} is a column–row outer product. Substituting the expression for H_k^{X} and applying the chain rule to $R \circ \bar{F}_{\mathsf{X}}$ shows that (36) is exactly (18), including for a nonlinear reset. Reinitialize Z_0, H_0 with (33) for the next phase. Thus neither past phase snapshots nor past computation graphs are needed after this update.

Fixed global-time outputs and losses. Maintain the row vector $b_i = \mathsf{grad}(N_i)$, starting from $b_0 = 0$, and update it at a restart by $b_{i+1} = b_i + d_i$. At an interior step $n = N_i + k$, $k \geq 1$, the global-trajectory surrogate is

$$J_n := \mathsf{grad}(\bar{X}(n)) = H_k^{\mathsf{X}} - (X_k - X_{k-1}) b_i. \tag{37}$$

This reproduces (19); applying only (34) would omit the shift of earlier phase boundaries. The matrix J_n is used for the output and does not replace H_k in the local propagation. At a phase anchor, use G_i as specified in Section 3. For a differentiable terminal loss $\ell(\bar{X}(n), \theta)$, its surrogate row gradient is $D_X \ell \, J_n + D_\theta \ell$. For a sum of per-step losses, these rows can be accumulated during the forward pass. More general trajectory-dependent losses require whatever additional state or storage their own evaluation needs.

Runtime and memory. No dense $m \times m$ update Jacobian is required. For parameter direction e_j, the jth column of (34) is the Jacobian–vector product of the joint map $(Z, \theta) \mapsto \Phi_k(Z; \theta)$ in direction $(H_k e_j, e_j)$. The full sensitivity can therefore be propagated with p such products per step, or with a batched equivalent. For gradient-based optimizers these products may involve Hessian–vector products, without forming a full Hessian.

If C_Φ is the cost of one state update, C_{JVP} the cost of one such directional derivative, and C_R the cost of a reset together with its sensitivity update, the total cost for N iterations and q restarts is

$$O\big(N(C_\Phi + pC_{\mathrm{JVP}}) + qC_R\big).$$

The additional sensitivity storage is $O(mp + p)$, with $O(m)$ storage for the current and preceding states and the temporary workspace for a single update. Thus, for fixed state and parameter dimensions and a terminal or streaming loss, storage is independent of trajectory length and runtime is linear in the iteration count. The parameter-dimension factor is explicit: forward propagation can be attractive for small restart policies, while a large p can make the sensitivity matrix costly. Propagating only selected parameter directions reduces this cost but returns only the corresponding directional derivatives. The stop-gradient construction in Appendix B.2 is an alternative realization of the same surrogate; a reverse-mode execution of its entire unrolled graph does not by itself give this trajectory-length-independent storage bound.

C Proofs from Section 2 and Section 3

C.1 Proof of Theorem 2.2

Proof. Let $L_i^0 = L_i(\theta_0)$ and $H_i(t,\theta) = F_{\mathsf{V}}(t; x_i(\theta), \theta)$. Starting from $D_\theta x_0 = D_\theta T_0 = 0$, induction and the implicit function theorem give

$$\left.\begin{array}{c} x_i, T_i \in C^1, \qquad H_i(L_i^0, \theta_0) = 0, \\ \partial_t H_i(L_i^0, \theta_0) < 0 \end{array}\right\} \Longrightarrow \begin{array}{c} \ell_i \in C^1(U_i), \quad \ell_i(\theta_0) = L_i^0, \\ H_i(\ell_i(\theta), \theta) = 0. \end{array}$$

For sufficiently small $r_i > 0$ and a smaller neighborhood U_i,

$$\begin{aligned} &0 < r_i < L_i^0, \qquad m_i := \min_{0 \le t \le L_i^0 - r_i} H_i(t, \theta_0) > 0, \\ &\qquad H_i(t,\theta) \ge m_i/2 && (0 \le t \le L_i^0 - r_i,\ \theta \in U_i), \\ &\qquad \partial_t H_i(t,\theta) \le -c_i < 0 && (|t - L_i^0| \le r_i,\ \theta \in U_i), \\ &|\ell_i(\theta) - L_i^0| < r_i, \qquad H_i(t,\theta) > 0 \quad (0 \le t < \ell_i(\theta)). \end{aligned}$$

Hence $L_i = \ell_i$ on U_i, and

$$\begin{aligned} 0 &= \partial_t F_{\mathsf{V}}\, D_\theta L_i + \partial_x F_{\mathsf{V}}\, D_\theta x_i + \partial_\theta F_{\mathsf{V}}, \\ D_\theta L_i &= -\frac{\partial_x F_{\mathsf{V}}\, D_\theta x_i + \partial_\theta F_{\mathsf{V}}}{\partial_t F_{\mathsf{V}}}, \\ D_\theta x_{i+1} &= \partial_t F_{\mathsf{R}}\, D_\theta L_i + \partial_x F_{\mathsf{R}}\, D_\theta x_i + \partial_\theta F_{\mathsf{R}}, \\ D_\theta T_{i+1} &= D_\theta T_i + D_\theta L_i. \end{aligned}$$

All derivatives of F and F_{R} in this display are evaluated at $(L_i; x_i, \theta)$. For fixed t strictly inside phase i, shrink the parameter neighborhood so that $T_i(\theta) < t < T_{i+1}(\theta)$. Then

$$D_\theta X(t) = -\partial_t F_{\mathsf{X}}(t - T_i; x_i, \theta) D_\theta T_i + \partial_x F_{\mathsf{X}}(t - T_i; x_i, \theta) D_\theta x_i + \partial_\theta F_{\mathsf{X}}(t - T_i; x_i, \theta).$$

□

C.2 Monotone improvement

Theorem C.1 (Monotone improvement)**.** *Let $\Theta \subset \mathbb{R}^p$ be open and convex, with a fixed initial state x_0. Suppose the restart hypotheses of Theorem 2.2 hold throughout Θ, with the same finite number of events before T, positive phase-start scores, and no event at T. Assume Assumption 3.3(1)–(2) hold uniformly, and $\mathcal{L}$ is uniformly Lipschitz and smooth. Then the terminal loss $\ell(\theta) = \mathcal{L}(X(T;\theta))$ is M_ℓ-smooth on Θ for some $M_\ell > 0$.*

Let $K \subset \Theta$ be nonempty, closed, and convex, and suppose $\inf_K \ell > -\infty$ (in particular, this holds for compact K). For $\theta_0 \in K$ and $0 < \eta < 1/M_\ell$, projected gradient descent (PGD) with the exact terminal-loss gradient satisfies

$$\theta_{k+1} = \Pi_K\big(\theta_k - \eta \nabla \ell(\theta_k)\big), \qquad G_\eta(\theta_k) = \frac{\theta_k - \theta_{k+1}}{\eta}, \tag{38}$$

$$\ell(\theta_{k+1}) \le \ell(\theta_k) - \eta\left(1 - \frac{M_\ell \eta}{2}\right) \|G_\eta(\theta_k)\|^2. \tag{39}$$

Thus every nonzero update strictly improves the loss, and $\|G_\eta(\theta_k)\| \to 0$. If projection is eventually inactive, then $\|\nabla \ell(\theta_k)\| \to 0$ as well. The latter conclusion also holds for ordinary gradient descent whose iterates stay in Θ and whose losses are bounded below.

Proof. Let q be the event count and $A \ge 1$ bound the first derivatives and their Lipschitz constants in Assumption 3.3(2). For functions on Θ, write $\|\cdot\|_\infty$ and $\mathrm{Lip}(\cdot)$ for the supremum norm and Lipschitz constant. By Appendix C.1, $x_i, L_i, T_i \in C^1(\Theta)$.

Phase sensitivities. Set $P_i = D_\theta x_i$, $b_i = D_\theta L_i$, $c_i = \|P_i\|_\infty$, and $d_i = \|b_i\|_\infty$. Equations (10)–(11) give

$$c_0 = 0, \qquad d_i \le \frac{A}{\gamma}(1 + c_i), \qquad c_{i+1} \le A^2(1 + c_i + d_i), \qquad \|D_\theta T_i\|_\infty \le \sum_{j<i} d_j. \tag{40}$$

For $z_i(\theta) = (L_i(\theta), x_i(\theta), \theta)$ and $\theta_s = \theta' + s(\theta - \theta')$,

$$z_i(\theta) - z_i(\theta') = \int_0^1 Dz_i(\theta_s)(\theta - \theta')\, ds, \qquad \mathrm{Lip}(z_i) \le D_i := 1 + c_i + d_i.$$

Define

$$\alpha_i = \partial_\theta F_{\mathsf{V}}(z_i) + \partial_x F_{\mathsf{V}}(z_i) P_i, \quad \nu_i = \partial_t F_{\mathsf{V}}(z_i), \quad \lambda_i = \mathrm{Lip}(P_i), \quad \kappa_i = \mathrm{Lip}(b_i).$$

Then

$$\begin{aligned} &\|\alpha_i\|_\infty \le A(1 + c_i), && \mathrm{Lip}(\nu_i) \le AD_i, \quad \nu_i \le -\gamma, \\ &\mathrm{Lip}(\alpha_i) \le AD_i(1 + c_i) + A\lambda_i, \\ b_i = -\alpha_i/\nu_i \quad \Longrightarrow \quad &\kappa_i \le \frac{AD_i(1 + c_i) + A\lambda_i}{\gamma} + \frac{A^2(1 + c_i)D_i}{\gamma^2}. \end{aligned}$$

Moreover,

$$\begin{aligned} DF_{\mathsf{R}}(z) &= DR(F_{\mathsf{X}}(z))DF_{\mathsf{X}}(z), \qquad \|DF_{\mathsf{R}}\|_\infty \le A^2, \\ \|DF_{\mathsf{R}}(z) - DF_{\mathsf{R}}(z')\| &\le A\|DF_{\mathsf{X}}(z) - DF_{\mathsf{X}}(z')\| + A^2\|F_{\mathsf{X}}(z) - F_{\mathsf{X}}(z')\| \\ &\le C_R\|z - z'\|, \qquad C_R = A^2 + A^3, \\ \lambda_{i+1} &\le C_R D_i(1 + c_i + d_i) + A^2(\lambda_i + \kappa_i), \qquad \lambda_0 = 0. \end{aligned}$$

Thus $c_i, d_i, \lambda_i, \kappa_i < \infty$ for all realized phases, and $\mathrm{Lip}(D_\theta T_q) \le \sum_{j<q} \kappa_j$.

Terminal loss. Put $Y(\theta) = F_{\mathsf{X}}(T - T_q; x_q, \theta)$ and $J = 1 + c_q + \sum_{j<q} d_j$. From (12),

$$\begin{aligned} DY &= -\partial_t F_{\mathsf{X}}\, D_\theta T_q + \partial_x F_{\mathsf{X}}\, P_q + \partial_\theta F_{\mathsf{X}}, \qquad (t, x, \theta) = (T - T_q, x_q, \theta), \\ B &:= AJ, \qquad C := AJ^2 + A\left(\lambda_q + \sum_{j<q} \kappa_j\right), \\ \|DY\|_\infty &\le B, \qquad \|DY(\theta) - DY(\theta')\| \le C\|\theta - \theta'\|. \end{aligned} \tag{41}$$

Consequently,

$$\begin{aligned} \|Y(\theta) - Y(\theta')\| &= \left\| \int_0^1 DY(\theta_s)(\theta - \theta')\, ds \right\| \le B\|\theta - \theta'\|, \\ \nabla\ell(\theta) &= DY(\theta)^\top \nabla\mathcal{L}(Y(\theta)), \qquad \|\nabla\mathcal{L}\|_\infty \le L_{\mathcal{L}}, \\ \|\nabla\ell(\theta) - \nabla\ell(\theta')\| &\le BM_{\mathcal{L}}\|Y(\theta) - Y(\theta')\| + L_{\mathcal{L}}\|DY(\theta) - DY(\theta')\| \\ &\le (M_{\mathcal{L}}B^2 + L_{\mathcal{L}}C)\|\theta - \theta'\|. \end{aligned}$$

Take $M_\ell = \max\{1, M_{\mathcal{L}}B^2 + L_{\mathcal{L}}C\}$.

Projected descent. With $g_k = \nabla\ell(\theta_k)$ and $s_k = \theta_{k+1} - \theta_k$, the projection inequality gives

$$\begin{aligned} \langle \theta_k - \eta g_k - \theta_{k+1}, \theta_k - \theta_{k+1} \rangle &\le 0, \\ \langle g_k, s_k \rangle &\le -\|s_k\|^2/\eta, \\ \ell(\theta_{k+1}) &\le \ell(\theta_k) + \langle g_k, s_k \rangle + \tfrac{M_\ell}{2}\|s_k\|^2 \\ &\le \ell(\theta_k) - \eta\left(1 - \tfrac{M_\ell \eta}{2}\right)\|G_\eta(\theta_k)\|^2. \end{aligned}$$

For $\ell_* := \inf_K \ell > -\infty$ and $0 < \eta < 1/M_\ell$,

$$s_k \neq 0 \implies \ell(\theta_{k+1}) < \ell(\theta_k),$$
$$\eta\left(1 - \tfrac{M_\ell \eta}{2}\right) \sum_{k=0}^{N-1} \|G_\eta(\theta_k)\|^2 \leq \ell(\theta_0) - \ell(\theta_N) \leq \ell(\theta_0) - \ell_*,$$
$$\sum_{k\geq 0} \|G_\eta(\theta_k)\|^2 < \infty \implies \|G_\eta(\theta_k)\| \longrightarrow 0.$$

If projection is inactive for $k \geq k_0$, then

$$G_\eta(\theta_k) = \nabla\ell(\theta_k) \quad (k \geq k_0) \implies \|\nabla\ell(\theta_k)\| \longrightarrow 0.$$

For ordinary gradient descent with $\theta_k \in \Theta$ and $\underline{\ell} := \inf_k \ell(\theta_k) > -\infty$,

$$\theta_{k+1} = \theta_k - \eta g_k \implies \eta(1 - M_\ell\eta/2) \sum_{k=0}^{N-1} \|g_k\|^2 \leq \ell(\theta_0) - \underline{\ell} \implies \|g_k\| \to 0.$$

□

C.3 Detailed Formulation and Proof of Theorem 3.4

The main-text statement uses a compact notation. Its phase indices refer to realized finite events, backward differences require $k \geq 1$, and the global-trajectory comparison is interpreted at sample times in matching phases away from moving restart boundaries. The following detailed formulation makes this scope explicit and also allows local regularity, an initial time layer, and an h-dependent continuous reference.

Assumption C.2. Fix a finite horizon T and a slightly larger local-time interval $[0, H]$, $H > T$. All bounds below hold on a neighborhood of the realized phase initializations and parameters.

(1) The event hypotheses of Theorem 2.2 hold, and $\partial_t F_{\mathsf{V}}(L_i; x_i, \theta) \leq -\gamma < 0$ for $0 \leq i < q$.

(2) The first derivatives of F and R are bounded by M, and DR is M-Lipschitz. For every fixed $\delta > 0$, the first derivatives of F are jointly M_δ-Lipschitz on local times $[\delta, H]$.

(3) For $t_k = kh \in [0, H]$, the value and initialization/parameter derivatives satisfy

$$\|\bar{F}(k; x, \theta) - F(t_k; x, \theta)\| + \|\partial_{(x,\theta)}\bar{F}(k; x, \theta) - \partial_{(x,\theta)}F(t_k; x, \theta)\| \leq \epsilon_h. \tag{42}$$

For every fixed $\delta > 0$, the backward time differences satisfy

$$\left\| \frac{\bar{F}(k; x, \theta) - \bar{F}(k-1; x, \theta)}{h} - \partial_t F(t_k; x, \theta) \right\| \leq C_\delta \epsilon_h, \qquad \delta \leq t_k \leq H. \tag{43}$$

Here $\epsilon_h \to 0$, and all constants are independent of h. The second estimate uses $k \geq 1$ and is not required in the initial time layer.

Theorem C.3 (Closeness of surrogate gradients to continuous gradients)**.** *Suppose Assumption C.2 holds, $\bar{x}_0 = x_0$ is independent of θ, and h is sufficiently small. The discrete and continuous trajectories then have the same q events before T, and*

$$|hS_i - L_i| + \|h\mathsf{grad}(S_i) - \partial_\theta L_i\| = O(\epsilon_h + h), \qquad 0 \leq i < q, \tag{44}$$
$$\|\bar{x}_{i+1} - x_{i+1}\| + \|\mathsf{grad}(\bar{x}_{i+1}) - \partial_\theta x_{i+1}\| = O(\epsilon_h + h), \qquad 0 \leq i < q. \tag{45}$$

For every fixed $\delta > 0$,

$$\|\bar{X}(n) - X(t_n)\| + \|\mathsf{grad}(\bar{X}(n)) - \partial_\theta X(t_n)\| = O_\delta(\epsilon_h + h) \tag{46}$$

uniformly over $t_n = nh \leq T$ with $\mathrm{dist}(t_n, \{0, T_1, \ldots, T_q\}) \geq \delta$. Such sample times belong to corresponding phases for small h. The value and sensitivity at the fixed initial point agree exactly.

The theorem also applies when the continuous reference is a family $F = F_h$, as in the high-resolution NAG model, provided all constants and neighborhoods above are uniform. In addition, require a fixed event count q and uniform event margins: there exist $\rho, m > 0$ such that $L_i \geq 2\rho$, $T - T_q \geq \rho$,

the score is at least m on $[0, L_i - \rho]$, and its time derivative is at most $-\gamma/2$ on $[L_i - \rho, L_i + \rho]$ for every $i < q$. In the final phase, require the score to be at least m through local time $T - T_q + \rho$. Take $H > T + \rho$. For a fixed reference flow, these margins follow by compactness, strict initial positivity, and the finitely many transverse crossings; for a family they are additional uniform hypotheses.

All regularity and consistency bounds are local to the relevant trajectories. In Appendix C.4 and Appendix C.5, we establish consistency with $\epsilon_h = O(h)$ for forward Euler and $\epsilon_h = O(h \log(1/h))$ for NAG. For NAG, the phase coordinate is the initial position with prescribed initial velocity, and the backward time-difference estimate is restricted to positive local times as above. Under the event hypotheses, the corresponding surrogate sensitivities converge at these rates.

Proof of Theorem C.3. Empty maxima below are zero. Set $r_h = \epsilon_h + h$, $a_i = \partial_\theta x_i$, $\bar{a}_i = \mathsf{grad}(\bar{x}_i)$, $u_i = \|\bar{x}_i - x_i\|$, and $w_i = \|\bar{a}_i - a_i\|$. Constants are independent of h; for a family F_h, use the uniform margins stated above. The finite recursion (10)–(11) gives

$$\max_{i<q} \|\partial_\theta L_i\| + \max_{i \le q} \big(\|a_i\| + \|\partial_\theta T_i\|\big) \le C, \qquad u_0 = w_0 = 0.$$

First crossings. Write $v_i(t) = F_{\mathsf{V}}(t; x_i, \theta)$ and choose $0 < \rho < L_i$ with

$$\min_{[0, L_i - \rho]} v_i = m_i > 0, \qquad v_i'(t) \le -\gamma/2 \quad (|t - L_i| \le \rho).$$

By (42),

$$|\bar{F}_{\mathsf{V}}(k; \bar{x}_i, \theta) - v_i(kh)| \le C u_i + \epsilon_h =: b_i. \tag{47}$$

For $b_i < m_i$ and $2b_i/\gamma + h < \rho$,

$$\begin{aligned} v_i(t) &\ge \tfrac{\gamma}{2}(L_i - t) && (L_i - \rho \le t \le L_i), \\ v_i(t) &\le -\tfrac{\gamma}{2}(t - L_i) && (L_i \le t \le L_i + \rho), \\ kh < L_i - 2b_i/\gamma &\implies \bar{F}_{\mathsf{V}}(k; \bar{x}_i, \theta) > 0, \\ k_i^+ := \left\lceil \frac{L_i + 2b_i/\gamma}{h} \right\rceil &\implies \bar{F}_{\mathsf{V}}(k_i^+; \bar{x}_i, \theta) \le 0. \end{aligned}$$

Consequently,

$$L_i - 2b_i/\gamma \le hS_i \le hk_i^+ \le L_i + 2b_i/\gamma + h, \qquad |hS_i - L_i| \le C(u_i + r_h). \tag{48}$$

The reset and accumulated-time errors satisfy

$$\begin{aligned} u_{i+1} &\le C\big(\epsilon_h + u_i + |hS_i - L_i|\big) \le C(u_i + r_h), \\ \max_{i \le q} u_i + \max_{i<q} |hS_i - L_i| &\le C r_h, \qquad |hN_i - T_i| \le \sum_{j<i} |hS_j - L_j| \le C r_h. \end{aligned}$$

For the final-phase positive margin m_q on $[0, T - T_q + \rho]$,

$$kh \le T - hN_q \implies kh \le T - T_q + C r_h, \qquad \bar{F}_{\mathsf{V}}(k; \bar{x}_q, \theta) \ge m_q - C u_q - \epsilon_h > 0.$$

Event sensitivities. Define

$$\begin{aligned} d_i &= \partial_t F_{\mathsf{V}}(L_i; x_i, \theta), & \bar{d}_i &= \Delta_i / h, \\ A_i &= \partial_\theta F_{\mathsf{V}}(L_i; x_i, \theta) + \partial_x F_{\mathsf{V}}(L_i; x_i, \theta) a_i, \\ \bar{A}_i &= \partial_\theta \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) + \partial_x \bar{F}_{\mathsf{V}}(S_i; \bar{x}_i, \theta) \bar{a}_i. \end{aligned}$$

The consistency estimates at $hS_i \ge L_i/2 > 0$ give

$$|\bar{d}_i - d_i| \le C r_h, \quad |\bar{d}_i| \ge \gamma/2, \quad \|A_i\| \le C, \quad \|\bar{A}_i - A_i\| \le C(r_h + w_i).$$

Thus

$$\|h\mathsf{grad}(S_i) - \partial_\theta L_i\| \le \frac{\|\bar{A}_i - A_i\|}{|\bar{d}_i|} + \frac{\|A_i\|\,|\bar{d}_i - d_i|}{|d_i \bar{d}_i|} \le C(r_h + w_i). \tag{49}$$

Reset sensitivities. Let $y = \bar{F}_{\mathsf{X}}(S_i; \bar{x}_i, \theta)$, $y^- = \bar{F}_{\mathsf{X}}(S_i - 1; \bar{x}_i, \theta)$, and $z = F_{\mathsf{X}}(L_i; x_i, \theta)$. Then

$$y - z = O(r_h), \qquad y - y^- = h\partial_t F_{\mathsf{X}}(L_i; x_i, \theta) + O(hr_h) = O(h),$$

$$\begin{aligned} R(y) - R(y^-) &= \left[\int_0^1 DR\big(y^- + s(y - y^-)\big)\, ds\right](y - y^-) \\ &= DR(y)(y - y^-) + O(h^2) = h\partial_t F_{\mathsf{R}}(L_i; x_i, \theta) + O(hr_h). \end{aligned} \tag{50}$$

With $B_i = (R(y) - R(y^-))/h$,

$$\begin{aligned} B_i &= \partial_t F_{\mathsf{R}}(L_i; x_i, \theta) + O(r_h), \\ \partial_{(x,\theta)} \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta) &= DR(y)\partial_{(x,\theta)} \bar{F}_{\mathsf{X}}(S_i; \bar{x}_i, \theta) \\ &= \partial_{(x,\theta)} F_{\mathsf{R}}(L_i; x_i, \theta) + O(r_h), \\ \bar{a}_{i+1} &= B_i\, h\mathsf{grad}(S_i) + \partial_x \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta)\bar{a}_i + \partial_\theta \bar{F}_{\mathsf{R}}(S_i; \bar{x}_i, \theta), \\ w_{i+1} &\le C\big(r_h + w_i + \|h\mathsf{grad}(S_i) - \partial_\theta L_i\|\big) \le C(r_h + w_i). \end{aligned}$$

Finite induction yields

$$\max_{i \le q} w_i + \max_{i < q} \|h\mathsf{grad}(S_i) - \partial_\theta L_i\| \le C r_h,$$

which proves (44)–(45).

Matched global times. For $\mathrm{dist}(t_n, \{0, T_1, \ldots, T_q\}) \ge \delta$ and $Cr_h < \delta/2$,

$$t_n \in (T_i, T_{i+1}) \implies N_i < n < N_{i+1},$$

$$k = n - N_i, \qquad \ell = t_n - T_i, \qquad |kh - \ell| = |hN_i - T_i| \le Cr_h, \qquad kh, \ell \ge \delta/2,$$

$$\|\bar{X}(n) - X(t_n)\| \le \epsilon_h + C(u_i + |kh - \ell|) \le C_\delta r_h,$$

$$\|h\mathsf{grad}(N_i) - \partial_\theta T_i\| \le \sum_{j<i} \|h\mathsf{grad}(S_j) - \partial_\theta L_j\| \le Cr_h.$$

For the final phase, the same implications hold with its right endpoint beyond T. Define

$$\bar{Q}_i = \frac{\bar{F}_{\mathsf{X}}(k; \bar{x}_i, \theta) - \bar{F}_{\mathsf{X}}(k-1; \bar{x}_i, \theta)}{h}, \quad Q_i = \partial_t F_{\mathsf{X}}(\ell; x_i, \theta), \quad J_i = \partial_x F_{\mathsf{X}}(\ell; x_i, \theta).$$

By (42)–(43),

$$\bar{Q}_i - Q_i = O_\delta(r_h), \qquad \partial_{(x,\theta)} \bar{F}_{\mathsf{X}}(k; \bar{x}_i, \theta) - \partial_{(x,\theta)} F_{\mathsf{X}}(\ell; x_i, \theta) = O_\delta(r_h).$$

$$\begin{aligned} \mathsf{grad}(\bar{X}(n)) - \partial_\theta X(t_n) &= -\bar{Q}_i\big(h\mathsf{grad}(N_i) - \partial_\theta T_i\big) - (\bar{Q}_i - Q_i)\partial_\theta T_i \\ &\quad + J_i(\bar{a}_i - a_i) + O_\delta(r_h) = O_\delta(r_h). \end{aligned}$$

□

C.4 Consistency for Forward Euler Discretization

We first verify Assumption 3.3 for the simplest case: forward Euler discretization of the generic restart-augmented ODE (4). The argument is standard but is included here to make the use of Theorem 3.4 self-contained.

Setup. Consider the ODE (4):

$$\dot{X}(t) = -\mathcal{A}_{\mathsf{X}}(X(t); \theta), \qquad \dot{V}(t) = -\mathcal{A}_{\mathsf{V}}(X(t), V(t); \theta), \qquad X(0) = x,\ V(0) = 1,$$

and the forward Euler discretization with stepsize $h > 0$:

$$\bar{X}_{k+1} = \bar{X}_k - h\,\mathcal{A}_{\mathsf{X}}(\bar{X}_k; \theta), \qquad \bar{V}_{k+1} = \bar{V}_k - h\,\mathcal{A}_{\mathsf{V}}(\bar{X}_k, \bar{V}_k; \theta), \qquad t_k = kh. \tag{51}$$

The flow map F and $\bar{F}$ are defined accordingly.

In the following, we verify the consistency condition (20) and (21) separately. We pack the variable $Z = (X, V)$ and define $\mathcal{A}(Z; \theta) := (\mathcal{A}_{\mathsf{X}}(X; \theta), \mathcal{A}_{\mathsf{V}}(X, V; \theta))$.

Consistency condition (20) We first prove (20) for this setting.

Proposition C.4 (Forward-Euler consistency I)**.** *Suppose there exists $M > 0$ such that $\|\mathcal{A}(Z;\theta)\| \le M$, $\|\partial_{(Z,\theta)}\mathcal{A}(Z;\theta)\| \le M$ and $\|\partial_{(Z,\theta)}\mathcal{A}(Z;\theta) - \partial_{(Z,\theta)}\mathcal{A}(Z';\theta)\| \le M\|Z - Z'\|$ for all Z, Z'. Then there exists $C = C(M,T) > 0$ such that, for any stepsize $h > 0$ and any integer k with $t_k \le T$,*

$$\|\bar{F}(k;x,\theta) - F(t_k;x,\theta)\| \le Ch, \tag{52}$$

$$\|\partial_{(x,\theta)}\bar{F}(k;x,\theta) - \partial_{(x,\theta)}F(t_k;x,\theta)\| \le Ch. \tag{53}$$

Proof. Let $Z = (X,V)$, $\dot{Z} = -\mathcal{A}(Z;\theta)$, $\bar{Z}_{k+1} = \bar{Z}_k - h\mathcal{A}(\bar{Z}_k;\theta)$, and $Z(0) = \bar{Z}_0 = (x,1)$.

State error. For $e_k = \|Z(t_k) - \bar{Z}_k\|$,

$$\ddot{Z} = \partial_Z\mathcal{A}(Z;\theta)\mathcal{A}(Z;\theta), \qquad \|\ddot{Z}\|_\infty \le M^2,$$

$$r_k := Z(t_{k+1}) - Z(t_k) + h\mathcal{A}(Z(t_k);\theta) = \int_{t_k}^{t_{k+1}} (t_{k+1} - s)\ddot{Z}(s)\,ds, \qquad \|r_k\| \le M^2h^2/2,$$

$$e_{k+1} \le (1 + Mh)e_k + M^2h^2/2, \qquad e_0 = 0,$$

$$e_k \le \frac{M^2h^2}{2}\sum_{j=0}^{k-1}(1+Mh)^j \le \frac{Mh}{2}(e^{MT} - 1).$$

Sensitivity error. Set $W = D_{(x,\theta)}Z$, $\bar{W}_k = D_{(x,\theta)}\bar{Z}_k$, and $B(Z) = \partial_Z\mathcal{A}(Z;\theta)$, $C(Z) = [0\ \partial_\theta\mathcal{A}(Z;\theta)]$. Then

$$\dot{W} = -B(Z)W - C(Z),$$
$$\bar{W}_{k+1} = \bar{W}_k - hB(\bar{Z}_k)\bar{W}_k - hC(\bar{Z}_k), \qquad W(0) = \bar{W}_0.$$

$$\sup_t \|W(t)\| + \sup_k \|\bar{W}_k\| \le C(M,T),$$
$$\|Z(t) - Z(s)\| + \|W(t) - W(s)\| \le C|t - s|.$$

$$\|\dot{W}(t) - \dot{W}(s)\| \le \|B(Z(t)) - B(Z(s))\|\,\|W(t)\| + \|B(Z(s))\|\,\|W(t) - W(s)\| + \|C(Z(t)) - C(Z(s))\| \le C_1|t - s|,$$

$$r_k^W := W(t_{k+1}) - W(t_k) - h\dot{W}(t_k) = \int_{t_k}^{t_{k+1}} [\dot{W}(s) - \dot{W}(t_k)]\,ds,$$

$$\|r_k^W\| \le C_1h^2/2.$$

Thus, with $\tilde{e}_k = \|W(t_k) - \bar{W}_k\|$,

$$\tilde{e}_{k+1} \le (1 + Mh)\tilde{e}_k + Mh(\|W(t_k)\| + 1)e_k + C_1h^2/2 \le (1 + Mh)\tilde{e}_k + Ch^2, \qquad \tilde{e}_0 = 0,$$

$$\tilde{e}_k \le Ch^2\sum_{j=0}^{k-1}(1+Mh)^j \le Ch.$$

□

Consistency condition (21) Next, we prove (21).

Proposition C.5 (Forward-Euler consistency II)**.** *Under the assumptions of Proposition C.4, there exists $C' = C'(M,T) > 0$ such that, for any stepsize $h > 0$ and any integer $k \ge 1$ with $t_k \le T$,*

$$\left\| \frac{\bar{F}(k;x,\theta) - \bar{F}(k-1;x,\theta)}{h} - \partial_t F(t_k;x,\theta) \right\| \le C'h. \tag{54}$$

Proof. For $1 \le k \le T/h$, (51) and Proposition C.4 give

$$\begin{aligned}\frac{\bar{F}(k;x,\theta)-\bar{F}(k-1;x,\theta)}{h} &= -\mathcal{A}(\bar{Z}_{k-1};\theta),\\ \partial_t F(t_k;x,\theta) &= -\mathcal{A}(Z(t_k);\theta),\\ \|\mathcal{A}(\bar{Z}_{k-1};\theta)-\mathcal{A}(Z(t_k);\theta)\| &\le M\big(\|\bar{Z}_{k-1}-\bar{Z}_k\|+\|\bar{Z}_k-Z(t_k)\|\big)\\ &\le M\left(Mh+\frac{Mh}{2}(e^{MT}-1)\right)=C'h.\end{aligned}$$

□

C.5 Consistency for NAG and its ODE

We verify consistency for the indexing used in (25), with $h=\sqrt{s}$, against the high-resolution ODE (3). The continuous reference itself depends on h. The derivatives below are with respect to the phase-start *position*; the initial velocity is prescribed as a function of that position.

Write $g=\nabla f$, $H=\nabla^2 f$, $t_k=kh$, and $\tau=3h/2$. The discrete and continuous phase dynamics are

$$\begin{aligned} y_k &= x_k+\beta_k h v_k, \qquad & v_{k+1} &= \beta_k v_k - hg(y_k),\\ x_{k+1} &= x_k + h v_{k+1}, \qquad & \beta_k &= \frac{k-1}{k+2}.\end{aligned} \tag{55}$$

$$\dot{x}=v, \qquad \dot{v}=-\frac{3}{t+\tau}v-hH(x)v-\left(1+\frac{3h}{2(t+\tau)}\right)g(x). \tag{56}$$

At a phase start a, set $(x_0,v_0)=(a,hb_d(a))$ and $(x(0),v(0))=(a,hb_c(a))$. In particular, $b_d=0$ is the discrete momentum reset, while $b_c=-g$ gives the initialization in (3). The proof allows this $O(h)$ initialization discrepancy, as well as either common initialization.

Proposition C.6 (NAG consistency on the prescribed initialization manifold)**.** *Let $f\in C^3$ be convex and fix a finite horizon $T>0$. Suppose that, for all sufficiently small $h>0$ and initial positions in a fixed neighborhood of a compact set, the continuous trajectories, discrete iterates, and extrapolated points through T lie in a common compact subset of the domain of f. Assume that g, H, DH are bounded on a neighborhood containing the line segments used below, and that b_c, b_d have uniformly bounded derivatives through order two there. Define*

$$Z_h(t;a)=(x(t;a),v(t;a)), \qquad \bar{Z}_k(a)=(x_k(a),v_k(a)), \qquad \eta_h=h\big[1+\log(1+\lfloor T/h\rfloor)\big].$$

There is a constant C, independent of h,k,a, such that, for $0\le k\le\lfloor T/h\rfloor$,

$$\|\bar{Z}_k(a)-Z_h(t_k;a)\|\le C\eta_h, \tag{57}$$

$$\|D_a\bar{Z}_k(a)-D_aZ_h(t_k;a)\|\le C\eta_h. \tag{58}$$

For every fixed $\delta>0$, there is C_δ independent of h,k,a such that

$$\left\|\frac{\bar{Z}_k(a)-\bar{Z}_{k-1}(a)}{h}-\dot{Z}_h(t_k;a)\right\|\le C_\delta\eta_h, \qquad \delta\le t_k\le T. \tag{59}$$

The first two bounds include $t=0$ and, when T is on the grid, $t=T$. The last bound excludes an initial interval of fixed physical length δ, rather than merely a fixed number of initial steps. No derivative with respect to an independently varied initial velocity is asserted.

Proof. Constants depend on T and the compact-set bounds, but not on $0<h\le 1$. All norms are Euclidean or induced operator norms.

Transformed dynamics. With $u=v+hg(x)$, $\zeta=(x,u)$, and $\tau=3h/2$,

$$\dot{x}=u-hg(x), \qquad \dot{u}=-\frac{3}{t+\tau}u-\frac{t}{t+\tau}g(x), \qquad u(0)=h(b_c(a)+g(a)). \tag{60}$$

Multiplication by $(t+\tau)^3$ gives

$$u(t)=\frac{\tau^3u(0)}{(t+\tau)^3}-\frac{1}{(t+\tau)^3}\int_0^t s(s+\tau)^2g(x(s))\,ds. \tag{61}$$

Hence

$$\|u(t)\| \le Ch + C\int_0^t \frac{s(s+\tau)^2}{(t+\tau)^3}\,ds \le C(t+h), \qquad \|v(t)\| \le C(t+h),$$
$$\|\dot u(t)\| + \|\ddot x(t)\| \le C, \qquad \ddot x = \dot u - hH(x)v,$$
$$\ddot u = \frac{3u}{(t+\tau)^2} - \frac{3\dot u}{t+\tau} - \frac{\tau g(x)}{(t+\tau)^2} - \frac{t}{t+\tau}H(x)v, \qquad \|\ddot\zeta(t)\| \le C\left(1+\frac{1}{t+h}\right).$$

Initial-position derivatives. For $P = D_a x$ and $Q = D_a u$,

$$\begin{aligned} \dot P &= Q - hH(x)P, && P(0) = I,\\ \dot Q &= -\frac{3}{t+\tau}Q - \frac{t}{t+\tau}H(x)P, && Q(0) = hD_a(b_c+g)(a). \end{aligned} \tag{62}$$

The integral equations yield

$$Q(t) = \frac{\tau^3 Q(0)}{(t+\tau)^3} - \int_0^t \frac{s(s+\tau)^2}{(t+\tau)^3}H(x(s))P(s)\,ds,$$
$$\|Q(t)\| \le Ch + C\int_0^t \|P(s)\|\,ds,$$
$$\|P(t)\| \le 1 + \int_0^t \big(\|Q(s)\| + Ch\|P(s)\|\big)ds \le C + C\int_0^t \|P(s)\|\,ds,$$
$$\|P(t)\| \le Ce^{Ct}, \quad \|Q(t)\| + \|\dot P(t)\| \le C(t+h), \quad \|\dot Q(t)\| \le C.$$

Let $K(t) = DH(x)[v]P + H(x)\dot P$. Then

$$\|K(t)\| \le C(t+h), \qquad \ddot P = \dot Q - hK,$$
$$\ddot Q = \frac{3Q}{(t+\tau)^2} - \frac{3\dot Q}{t+\tau} - \frac{\tau H(x)P}{(t+\tau)^2} - \frac{t}{t+\tau}K,$$
$$\left\|\frac{d^2}{dt^2}D_a\zeta(t;a)\right\| \le C\left(1+\frac{1}{t+h}\right). \tag{63}$$

For $P_2 = D_a^2 x$ and $Q_2 = D_a^2 u$,

$$\begin{aligned} \dot P_2 &= Q_2 - h(HP_2 + DH[P,P]), && P_2(0) = 0,\\ \dot Q_2 &= -\frac{3}{t+\tau}Q_2 - \frac{t}{t+\tau}(HP_2 + DH[P,P]), && Q_2(0) = O(h), \end{aligned}$$
$$\|Q_2(t)\| \le Ch + C\int_0^t (1 + \|P_2(s)\|)\,ds,$$
$$\|P_2(t)\| \le C + C\int_0^t \|P_2(s)\|\,ds.$$

Therefore $\|P_2\| \le C$ and $\|Q_2(t)\| \le C(t+h)$.

Discrete stability. Let Φ_k be the map in (55), $\mathcal{T}_h(x,v) = (x, v + hg(x))$, and $\Psi_k = \mathcal{T}_h \circ \Phi_k \circ \mathcal{T}_h^{-1}$. With $y = x + \beta_k h v$,

$$D\Phi_k = \begin{pmatrix} I - h^2H(y) & h\beta_k I - h^3\beta_k H(y) \\ -hH(y) & \beta_k I - h^2\beta_k H(y) \end{pmatrix}, \qquad |\beta_k| \le 1.$$

For directions $\xi_j = (p_j, r_j)$ and $e_j = p_j + \beta_k h r_j$,

$$D^2\Phi_k[\xi_1,\xi_2] = \begin{pmatrix} -h^2 DH(y)[e_1,e_2] \\ -hDH(y)[e_1,e_2] \end{pmatrix}, \qquad \|D\Phi_k\| \le 1 + Ch, \quad \|D^2\Phi_k\| \le Ch,$$
$$D\mathcal{T}_h^{\pm 1} = \begin{pmatrix} I & 0 \\ \pm hH(x) & I \end{pmatrix}, \qquad D^2\mathcal{T}_h^{\pm 1}[\xi_1,\xi_2] = \begin{pmatrix} 0 \\ \pm hDH(x)[p_1,p_2] \end{pmatrix}.$$

The chain rule gives

$$\|D\Psi_k\| \le 1 + Ch, \qquad \|D^2\Psi_k\| \le Ch. \tag{64}$$

Local defects. Write $\|r\|_{1,a} = \|r\| + \|D_a r\|$. Along the continuous curve at $t = t_k$, the transformed update is

$$v = u - hg(x), \qquad y = x + \beta_k h v, \qquad x^+ = y - h^2 g(y),$$
$$u^+ = \beta_k u - \beta_k h g(x) + h\{g(x^+) - g(y)\}.$$

Let B_h denote the vector field in (60) and $d_k = \Psi_k(\zeta(t_k)) - \zeta(t_k) - hB_h(t_k, \zeta(t_k))$. Then

$$d_k^x = h(\beta_k - 1)v - h^2 g(y), \qquad \beta_k - 1 = -\frac{3h}{t_k + 2h},$$
$$\|d_k^x\|_{1,a} \le Ch^2, \qquad \|v(t_k)\|_{1,a} + \|u(t_k)\|_{1,a} \le C(t_k + h),$$

$$d_k^u = c_{1,k} u + h c_{2,k} g(x) + h\{g(x^+) - g(y)\},$$
$$c_{1,k} = \frac{3h^2}{2(t_k + \tau)(t_k + 2h)}, \qquad c_{2,k} = \frac{3h(t_k + h)}{2(t_k + \tau)(t_k + 2h)}. \tag{65}$$

Furthermore,

$$x^+ - y = -h^2 g(y),$$
$$D_a x^+ - D_a y = -h^2 H(y) D_a y,$$
$$D_a\{g(x^+) - g(y)\} = [H(x^+) - H(y)] D_a y + H(x^+)(D_a x^+ - D_a y),$$
$$\|h\{g(x^+) - g(y)\}\|_{1,a} \le Ch^3,$$
$$\|d_k^u\|_{1,a} \le C\left(\frac{h^2}{t_k + h} + h^3\right).$$

With $r_k = \zeta(t_{k+1}) - \Psi_k(\zeta(t_k))$,

$$r_k = \int_{t_k}^{t_{k+1}} (t_{k+1} - s)\ddot{\zeta}(s)\, ds - d_k,$$
$$\left\| \int_{t_k}^{t_{k+1}} (t_{k+1} - s)\ddot{\zeta}(s)\, ds \right\|_{1,a} \le Ch^2 \left(1 + \frac{1}{t_k + h}\right),$$

$$\|r_k\| + \|D_a r_k\| \le C\left(h^2 + \frac{h}{k+1}\right). \tag{66}$$

Accumulation. Set $\bar{\zeta}_k = \mathcal{T}_h(\bar{Z}_k)$, $e_k = \|\bar{\zeta}_k - \zeta(t_k)\|$, and $w_k = \|D_a \bar{\zeta}_k - D_a \zeta(t_k)\|$. Then

$$\bar{\zeta}_0 - \zeta(0) = \big(0, h(b_d(a) - b_c(a))\big), \qquad e_0 + w_0 \le Ch,$$
$$e_{k+1} \le (1 + Ch)e_k + \|r_k\|, \qquad w_{k+1} \le (1 + Ch)w_k + Che_k + \|D_a r_k\|,$$
$$e_k + w_k \le e^{CT}\left[Ch + C\sum_{j<k}\left(h^2 + \frac{h}{j+1}\right)\right] \le C\eta_h,$$
$$\|\bar{Z}_k - Z_h(t_k)\| + \|D_a \bar{Z}_k - D_a Z_h(t_k)\| \le C(e_k + w_k) \le C\eta_h.$$

Backward differences. For $k \ge 1$,

$$\frac{x_k - x_{k-1}}{h} = v_k, \qquad \frac{v_k - v_{k-1}}{h} = -\frac{3}{t_k + h} v_{k-1} - g(y_{k-1}).$$

For $t_k \geq \delta$ and $h < \delta/2$,

$$\begin{aligned}
\|v_{k-1} - v(t_k)\| + \|y_{k-1} - x(t_k)\| &\leq C(\eta_h + h),\\
\left|\frac{3}{t_k + h} - \frac{3}{t_k + \tau}\right| &\leq C_\delta h,\\
\left\|\frac{v_k - v_{k-1}}{h} - \dot{v}(t_k)\right\| &\leq \frac{3}{t_k + h}\|v_{k-1} - v(t_k)\| + C\|y_{k-1} - x(t_k)\|\\
&\quad + \left|\frac{3}{t_k + h} - \frac{3}{t_k + \tau}\right| \|v(t_k)\| + h\|H(x(t_k))v(t_k)\|\\
&\quad + \frac{3h}{2(t_k + \tau)}\|g(x(t_k))\| \leq C_\delta \eta_h,\\
\left\|\frac{x_k - x_{k-1}}{h} - \dot{x}(t_k)\right\| &= \|v_k - v(t_k)\| \leq C\eta_h.
\end{aligned}$$

□

Scores. For a C^2 score S with uniformly bounded derivatives, let

$$V_h(t; a, \theta) = S(t, Z_h(t; a), a; \theta), \qquad \bar{V}_k(a, \theta) = S(t_k, \bar{Z}_k(a), a; \theta).$$

Writing $\xi = (a, \theta)$, the chain rule gives

$$D_\xi V_h = S_Z D_\xi Z_h + S_\xi, \qquad D_\xi \bar{V}_k = S_Z D_\xi \bar{Z}_k + S_\xi,$$

$$|\bar{V}_k - V_h(t_k)| + \|D_\xi \bar{V}_k - D_\xi V_h(t_k)\| \leq C\big(\|\bar{Z}_k - Z_h(t_k)\| + \|D_a \bar{Z}_k - D_a Z_h(t_k)\|\big) \leq C\eta_h.$$

The derivatives of S are evaluated at their corresponding continuous or discrete arguments. Put $\bar{w}_k = (t_k, \bar{Z}_k)$, $w(t) = (t, Z_h(t))$, and hold a, θ fixed. For $t_k \geq \delta$,

$$\begin{aligned}
\frac{\bar{V}_k - \bar{V}_{k-1}}{h} &= \int_0^1 D_w S(\bar{w}_{k-1} + s(\bar{w}_k - \bar{w}_{k-1}), a; \theta)\frac{\bar{w}_k - \bar{w}_{k-1}}{h}\, ds,\\
\left|\frac{\bar{V}_k - \bar{V}_{k-1}}{h} - \dot{V}_h(t_k)\right| &\leq C_\delta \left(\eta_h + h + \left\|\frac{\bar{Z}_k - \bar{Z}_{k-1}}{h} - \dot{Z}_h(t_k)\right\|\right) \leq C_\delta \eta_h.
\end{aligned}$$

For a C^2 integrated score field $\mathcal{A}_\mathsf{V}$ with uniformly bounded derivatives,

$$\dot{V}_h = -\mathcal{A}_\mathsf{V}(Z_h, V_h, t; \theta), \qquad \bar{V}_{k+1} = \bar{V}_k - h\mathcal{A}_\mathsf{V}(\bar{Z}_k, \bar{V}_k, t_k; \theta), \qquad V_h(0) = \bar{V}_0 = 1.$$

Let $U = D_\xi V_h$, $\bar{U}_k = D_\xi \bar{V}_k$, and $\partial_\xi^{\exp}$ denote explicit dependence on ξ. Then

$$\begin{aligned}
\dot{U} &= -\partial_Z \mathcal{A}_\mathsf{V}\, D_\xi Z_h - \partial_V \mathcal{A}_\mathsf{V}\, U - \partial_\xi^{\exp}\mathcal{A}_\mathsf{V},\\
\|\dot{Z}_h\| + \|D_a \dot{Z}_h\| + |\dot{V}_h| + \|U\| + \|\dot{U}\| &\leq C,\\
\|r_k^V\| + \|r_k^U\| &\leq Ch^2,\\
E_k := |\bar{V}_k - V_h(t_k)| + \|\bar{U}_k - U(t_k)\|, \qquad E_0 &= 0,\\
E_{k+1} &\leq (1 + Ch)E_k + Ch\eta_h + Ch^2,\\
E_k &\leq C(\eta_h + h)h\sum_{j<k}(1 + Ch)^j \leq C\eta_h,\\
\left|\frac{\bar{V}_k - \bar{V}_{k-1}}{h} - \dot{V}_h(t_k)\right| &\leq C(E_{k-1} + \eta_h + h) \leq C\eta_h.
\end{aligned}$$

Here r_k^V, r_k^U are the continuous Euler residuals. For second parameter derivatives,

$$\begin{aligned}
\frac{d}{dt}D_\xi^2 V_h &= -\partial_V \mathcal{A}_\mathsf{V}\, D_\xi^2 V_h + B_\xi(t), \qquad \|B_\xi(t)\| \leq C,\\
D_\xi^2 V_h(0) &= 0, \qquad \sup_t \|D_\xi^2 V_h(t)\| \leq C.
\end{aligned}$$

For either score construction,

$$\|D_{(t,a,\theta)} V_h\| \leq C, \qquad \|D^2_{(t,a,\theta)} V_h\| \leq C_\delta \quad (t \geq \delta),$$

which supplies the continuous regularity in Assumption C.2.

Application to restarted trajectories. To apply Theorem C.3, use the phase-start position a as the abstract state coordinate, set $F_{\mathsf{X}}(t;a) = x(t;a)$ and $\bar{F}_{\mathsf{X}}(k;a) = x_k(a)$, and take R to be the identity on positions. Each phase internally initializes its velocity with $hb_c(a)$ or $hb_d(a)$; the score may still observe the full position–velocity state. This is permitted by the general flow-map formulation and ensures that the derivative in Assumption C.2 is exactly the D_a derivative proved above. Under the positive-score and uniform event-margin hypotheses of that theorem, the consistency rate is $\epsilon_h = O(\eta_h) = O(h\log(1/h))$. Full-state observations inside matched phases also converge with this rate away from phase starts and restart boundaries, by the same chain rule. This is a comparison with the h-dependent high-resolution model; it does not assert convergence of gradients to a separate $h = 0$ model.

D Experiment Details for QP

We provide details for the experiments in Section 2.4, Appendix D.2, and Section 4.1.

Problem instances. With a given dimension d and condition number κ, we generate QP problems of the form (13) as [6]

$$f(x) = \tfrac{1}{2}(x - x^\star)^\top Q(x - x^\star), \qquad Q = \mathrm{diag}(\lambda_1, \ldots, \lambda_d), \tag{67}$$

where the diagonal entries are placed on a geometric grid,

$$\lambda_i = \kappa^{(i-1)/(d-1)}, \qquad i = 1, \ldots, d, \tag{68}$$

so that the smallest and largest eigenvalues are $\mu = 1$ and $L = \kappa = 80$. This corresponds to setting $A = Q$, $b = Qx^\star$ and $c = \frac{1}{2}(x^\star)^\top Qx^\star$.

The minimizer $x^\star$ is sampled from the standard Gaussian distribution with random seed 0. The initial point is sampled as $x_0 \sim \mathrm{Unif}([-20, 20]^d)$.

D.1 Details of Section 2.4

ODE dynamics. Within each restart phase, the continuous-time dynamics are the high-resolution ODE (3), written in joint state $(x(t), v(t))$ with $v = \dot{x}$:

$$\dot{x}(t) = v(t), \qquad \dot{v}(t) = -\frac{3}{t+\tau}v(t) - \sqrt{s}\,\nabla^2 f(x(t))\,v(t) - \left(1 + \frac{3\sqrt{s}}{2(t+\tau)}\right)\nabla f(x(t)). \tag{69}$$

with $\tau = 3\sqrt{s}/2$, $x(0) = x_0$ and $\dot{x}(0) = -\sqrt{s}\nabla f(x_0)$. The reference stepsize is $s = 1/L = 0.0125$ and the terminal time is $T = K\sqrt{s}$ with $K = 200$. Trajectories are integrated with a fourth-order Runge–Kutta method at fixed step $\Delta t = 5 \times 10^{-3}$. When a restart is triggered, the phase is reset by $V \leftarrow 1$ and the current state is stored as the new phase anchor.

As a reference-only ceiling, in Figure 1 we also report the high-resolution accelerated ODE for μ-strongly-convex objectives of Shi et al. (2022):

$$\ddot{x}(t) + 2\sqrt{\mu}\,\dot{x}(t) + \sqrt{s}\,\nabla^2 f(x(t))\,\dot{x}(t) + \big(1 + \sqrt{\mu s}\big)\nabla f(x(t)) = 0, \tag{70}$$

with $x(0) = x_0$ and $\dot{x}(0) = -\frac{2\sqrt{s}}{1+\sqrt{\mu s}}\nabla f(x_0)$. Unlike the ODE (3), this dynamics requires the strong-convexity parameter μ explicitly, and is therefore not a practical baseline; it is included here only as a reference that adaptive restart aims to recover without knowledge of μ.

Network architecture and training. The learnable score $V(X;\theta) = \mathsf{MLP}(\phi(X,t);\theta)$ is an MLP with embedding map $\phi : \mathbb{R}^d \to \mathbb{R}^5$ defined as

$$\phi([x;v],t) = \big(f(x),\ \|\nabla f(x)\|,\ \|x - x_{\mathrm{anc}}\|,\ \sqrt{s}\,\|v\|,\ 1 + t/\sqrt{s}\big), \tag{71}$$

where x_{anc} is the starting point of the current phase. The MLP has input dimension 5, a single hidden layer of width 16 with $\tanh$ activations, and a scalar output logit, for a total of 113 trainable parameters.

[6] We remark that it is without loss of generality to assume the matrix A in (13) is diagonal as the NAG updates are rotation-invariant.

Training proceeds in two stages. We first *warm-start* the network on a dataset of feature–label pairs $\mathcal{D} = \{(\phi_i, y_i)\}_{i=1}^{M}$ with $M = 800$ examples collected from trajectories driven by the function-restart ODE; the warm-start is fitted by minimizing the binary cross-entropy with logits, with positive-class reweighting to compensate for the rarity of restart labels. Starting from the warm-started parameters, we then *fine-tune* the network using the surrogate gradients of Definition 3.2 on the loss (14) for 20 epochs, with the AMSGrad optimizer at learning rate 10^{-2}.

Restart policies. Figure 1 compares six trajectories on the test instance:

- *ODE no restart*: the ODE (69) with no restart applied.
- *Function-restart ODE*: restart triggered by the score $V_{\mathsf{func}}(t) = -\nabla f(X)^\top \dot{X}$, the continuous analogue of the discrete rule "restart iff $f(x_{k+1}) > f(x_k)$".
- *Gradient-restart ODE*: restart triggered by the score $V_{\mathsf{grad}}(t) = -\nabla f(X + \sqrt{s}\,\dot{X})^\top \dot{X}$, the continuous analogue of the discrete rule "restart iff $\nabla f(y_k)^\top (x_{k+1} - x_k) > 0$".
- *Periodic-restart ODE*: restart triggered at $t = \tau, 2\tau, \cdots$ for a fixed choice of restart length τ.
- *Oracle SC-NAG ODE*: the strongly-convex accelerated ODE that uses the true strong-convexity constant μ; this is a reference-only ceiling, since μ is not available in practice.
- *Learned restart*: the trained policy.

For both handcrafted scores, a restart is triggered when the corresponding score crosses from nonnegative to negative between two consecutive RK4 states (i.e., $V \le 0$); the precise crossing time is approximated by linear interpolation within the step.

Final results on the test instance. Table 3 reports the terminal objective gaps at $t = T$ for the trajectories plotted in Figure 1.

Table 3: Terminal objective gaps $f(x(T)) - \min f$ on the held-out test initialization ($d = 30$, $\kappa = 80$, $T \approx 22.36$). The learned restart policy improves over both handcrafted scores by several orders of magnitude and approaches the oracle strongly-convex curve.

Method	Terminal gap $f(x(T)) - \min f$
ODE no restart	4.99×10^{-5}
Function-restart ODE	2.12×10^{-11}
Gradient-restart ODE	7.02×10^{-12}
Learned restart	4.75×10^{-14}
Oracle SC-NAG ODE (reference)	1.28×10^{-17}

D.2 Validation of the Approximation Accuracy

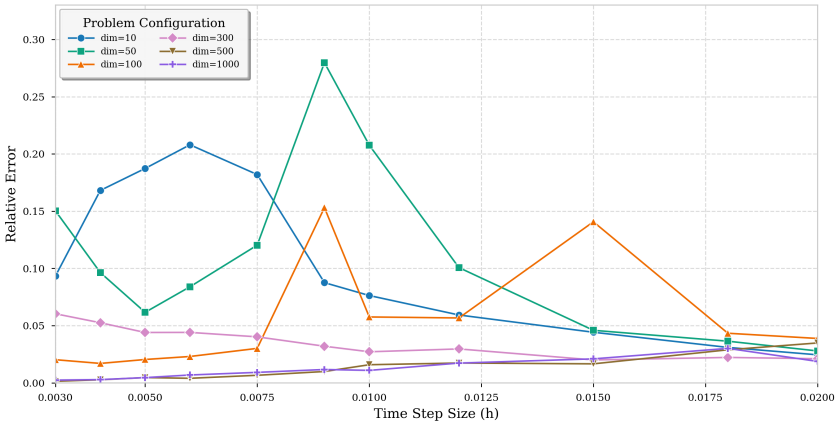
We now empirically validate the surrogate gradient of Definition 3.2 against its continuous-time counterpart in a setting where both can be computed accurately. We reuse the QP problem family of Section 2.4 together with a simple parameterized restart score $V(\cdot;\theta)$. We compare:

- Continuous-time dynamics $(X(t))$ driven by ODE (3) with gradients computed via Definition 2.1.
- Discrete-time dynamics $(\bar{X}(n))$ driven by NAG iteration with stepsize s (detailed in Appendix B.1) with gradients computed via Definition 3.2 (via the autodiff implementation of Appendix B.2).

We evaluate the average relative error

$$\mathrm{Err}_{\mathrm{grad}} = \frac{\|\mathsf{grad}(\bar{X}(n)) - \partial_\theta X(t_n)\|_2}{\|\partial_\theta X(t_n)\|_2 + \|\mathsf{grad}(\bar{X}(n))\|_2}, \tag{72}$$

across a range of stepsizes s and dimensions d, and report them in Figure 4. As predicted by Theorem 3.4 together with the consistency analysis of Appendix C.5, both errors decrease as $s \to 0$ and remain stable across dimensions, confirming that the surrogate gradient faithfully approximates its continuous counterpart at the same order as the underlying discretization error.



Terminal Gap Gradient Relative Error vs Time Step Size

Figure 4: Relative error between the discrete surrogate gradient $\mathsf{grad}(f(X(T)))$ and the continuous-time gradient $\partial_\theta f(X(T))$, plotted as a function of the stepsize h (with $s = h^2$) for dimensions d ranging from 10 to 1000. The error decreases as $h \to 0$ at a rate consistent with Theorem 3.4 and remains stable across dimensions. Details are in Appendix D.1.

D.3 Details for Section 4.1

The base optimizer is the NAG iteration of Appendix B.1 with step size $\eta = 1/\lambda_{\max}(H)$. The learned restart rule is a two-layer $\tanh$ MLP with embedding dimension 4 and hidden dimension 32 and input features

$$\phi_k = \big(f(x_{k+1}),\ \|x_{k+1} - \bar{x}_i\|_2,\ 1 + (k - N_i)/\sqrt{s},\ \cos\angle(x_{k+1} - x_k,\ x_k - x_{k-1})\big), \tag{73}$$

where $\bar{x}_i$ is the last restart state.

Training uses 20-dimensional problems with condition numbers $\kappa \in \{180, 240, 320\}$ and the two-stage procedure of Section 4.1: warm-start from function-restart teacher labels, followed by end-to-end fine-tuning on a ratio objective that normalizes the learned policy's terminal gap by the best handcrafted baseline. The mean objective gap along the trajectory on $d = 20$, 400, and 1000 is reported in Figure 2 in the main text.

Additional out-of-distribution evaluation. To further probe generalization, we evaluate the same policy on condition numbers *not seen during training*: $\kappa \in \{140, 200, 280, 360\}$. Figure 5 shows results on $d = 500$ and $d = 2000$; the learned policy continues to outperform both handcrafted baselines on these unseen κ values.

E Experiment Details for Real-Data Logistic Regression

This appendix collects the details deferred from Section 4.1: the construction of the evaluation instances, the exact computation of $f^\star$, the training distribution and protocol of the transferred policy, and the evaluation protocol.

Problem instances. We use the five binary-classification datasets `a9a`, `w8a`, `ijcnn1`, `covtype` (binarized) and `phishing` from the LIBSVM collection (Chang & Lin, 2011), with raw dimensions 123, 300, 22, 54 and 68 respectively. Each *instance* is generated from a dataset $(a_i, y_i)_{i=1}^M$, $y_i \in \{-1, +1\}$, by the following deterministic-given-seed procedure.

1. Draw $m = 1000$ rows without replacement and standardize each raw feature column to zero mean and unit variance.

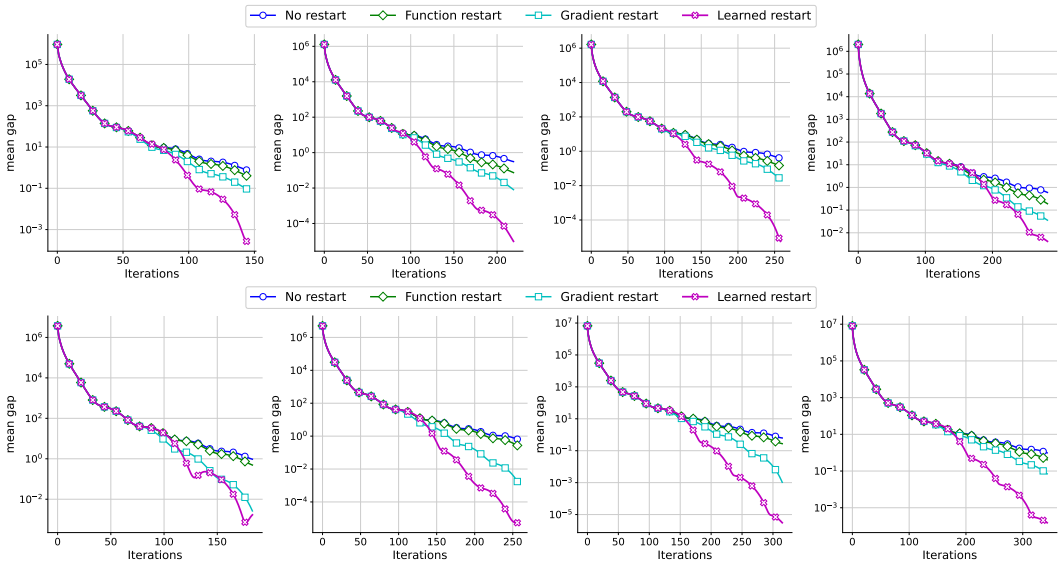


Figure 5: Out-of-distribution transfer to unseen condition numbers. The policy is trained on $\kappa \in \{180, 240, 320\}$ at $d = 20$ and evaluated zero-shot on $d = 500$ (*top*) and $d = 2000$ (*bottom*) with $\kappa \in \{140, 200, 280, 360\}$. The learned policy transfers across both dimensions and condition numbers not seen during training.

2. Draw random Fourier features (Rahimi & Recht, 2007) $z(a) = \sqrt{2/D}\cos(Wa + b)$ with $D = 2000$, $W_{ij} \sim \mathcal{N}(0, \sigma^{-2})$ and $b_i \sim \mathrm{Unif}[0, 2\pi]$, where the bandwidth σ^2 is set by the median heuristic on a 300-point subsample of pairwise squared distances.
3. Rescale the resulting design matrix $Z \in \mathbb{R}^{m\times D}$ to unit spectral norm, so that the data-term smoothness constant $L_0 = 0.25/m$ is shared across instances (using $\sup_t \sigma''(t) \le 1/4$ for the logistic link).
4. Set $\lambda = \lambda_{\mathrm{rel}} L_0$ and $x_0 \sim \mathcal{N}(0, I_D)$.

The resulting objective is the ℓ_2-regularized logistic regression

$$\min_{x\in\mathbb{R}^D} \; f(x) = \frac{1}{m}\sum_{i=1}^{m} \log\!\left(1 + e^{-y_i\langle z_i, x\rangle}\right) + \frac{\lambda}{2}\|x\|^2, \qquad y_i \in \{-1, +1\}, \tag{74}$$

where $\lambda = \lambda_{\mathrm{rel}} L_0$ with $L_0 = 0.25/m$ the smoothness constant of the data term. Since $\nabla^2 f \preceq \frac{1}{4m} Z^\top Z + \lambda I = (L_0 + \lambda) I$ and $\nabla^2 f \succeq \lambda I$, the instance is $(L_0 + \lambda)$-smooth and λ-strongly convex, so the effective condition number is $\kappa_{\mathrm{eff}} = (L_0 + \lambda)/\lambda = 1 + 1/\lambda_{\mathrm{rel}}$.

Exact minimizer. Unlike the QP family, (74) has no closed-form minimizer. We compute $x^\star$ by damped Newton with Armijo backtracking in the D-dimensional primal, started from the origin and iterated until $\|\nabla f\|_\infty \le 10^{-15} \cdot \max\{1, \|\nabla f(0)\|_\infty\}$; the achieved residual is recorded per instance and is of order 10^{-18} in all runs. The reported gaps $f(x_N) - f^\star$ are therefore exact to machine precision and involve no long reference run, which matters here because the gaps of interest are as small as 10^{-6}.

Training distribution and policy. The policy has the same architecture family as in Section 4.1 and is trained exclusively on synthetic quadratics of the form (67); no real data and no logistic objective is seen during training. The training distribution draws, per problem, a dimension $d \in \{10, 20, 40, 100\}$, a condition number $\kappa \in \{120, 180, 240, 320, 420\}$, a spectrum shape in {geometric, power, log-uniform}, and an initialization scale $x_0 \sim \mathrm{Unif}([-c, c]^d)$ with $c \in \{5, 20, 50\}$; each problem is a batch of 8 instances run for $N_{\mathrm{train}} = 150$ iterations.

For transfer across problem scales the feature map (73) is replaced by a scale-free extension: magnitudes enter only through $\log_{10}$ and all remaining coordinates are dimensionless ratios,

$$\phi_k = \Big(\tfrac{1}{10}\log_{10} f(x_{k+1}),\ \tfrac{1}{10}\log_{10}\|x_{k+1}-\bar{x}_i\|,\ \log_{10}(1+k-N_i),\ \cos\angle(x_{k+1}-x_k, x_k-x_{k-1}),\ r_k,\ c_k,\ p_k\Big), \tag{75}$$

where $r_k = \frac{f(x_k)-f(x_{k+1})}{f(x_k)+f(x_{k+1})}$ is the normalized one-step decrease (the scale-free analogue of the function-restart signal), $c_k = \frac{\langle \nabla f(x_{k+1}), v_{k+1}\rangle}{\|\nabla f(x_{k+1})\|\|v_{k+1}\|}$ is the gradient–momentum alignment (the scale-free analogue of the gradient-restart signal), and $p_k = \frac{1}{10}\log_{10}\big(f(x_{k+1})/f(\bar{x}_i)\big)$ measures within-phase progress relative to the phase anchor $\bar{x}_i$. No coordinate depends explicitly on d, m, or the scale of f, which is what makes evaluation at $D = 2000$ meaningful for a policy trained at $d \le 100$. The network is a two-layer $\tanh$ MLP of width 32 with a scalar output logit (7-32-1, 289 trainable parameters), whose output bias is initialized to $+0.5$ so that the initial policy does not restart.

Training protocol. Training follows the same two stages as Section 4.1. In the *warm-start* stage we roll out the function-restart rule on 24 sampled training problems and record every $(\phi_k, \mathbb{1}\{\text{restart}\})$ pair along the trajectories, giving $24 \times 8 \times 150 = 28{,}800$ examples; the network is fitted for 300 full-batch Adam steps at learning rate 10^{-2} on the binary cross-entropy with logits, with positive-class reweighting to compensate for the rarity of restart labels (converged loss 0.062–0.070 across seeds). Because a restart is a hard threshold event whose straight-through gradient vanishes identically, this cloned policy is also the strongest derivative-free trainable baseline available.

In the *fine-tuning* stage the warm-started network is trained end-to-end through the surrogate gradient of Definition 3.2 for 30 epochs of 8 freshly sampled problems each, minimizing the clipped log-ratio of the terminal gap to the best handcrafted baseline on the same problem,

$$J(\theta) = \mathbb{E}\Big[\max\big\{\log\big(f(x_{N_{\text{train}}};\theta)-f^\star\big)-\log g_{\text{base}},\ -3\log 10\big\}\Big], \qquad g_{\text{base}} = \min_{\text{base}}\big(f(x^{\text{base}}_{N_{\text{train}}})-f^\star\big), \tag{76}$$

where the minimum is over {no restart, function restart, gradient restart} and the floor at $-3\log 10$ prevents a single easy problem from dominating the batch. We use AMSGrad at learning rate 10^{-3} with gradient-norm clipping at 5, and select the checkpoint attaining the best value of (76) on a held-out set of 16 validation problems drawn from the same synthetic distribution. The entire two-stage procedure takes about two CPU-minutes per seed. We repeat it for 5 independent seeds and report all of them; no seed selection is performed anywhere in Section 4.1.

Evaluation protocol. All methods are run on the same instance, from the same x_0, with the same stepsize $s = 1/(L_0+\lambda)$ and the same horizon N; the only difference is the restart rule. We evaluate 6 instances per configuration and all 5 training seeds, giving 30 paired (instance, seed) comparisons per cell and 300 in total across the ten configurations. Per-configuration numbers are geometric means of the terminal gap over instances (and additionally over seeds for the learned policies). Because different datasets have gaps of different magnitude, aggregate improvements are reported as geometric means of the per-configuration ratios. We compare against the instance-wise *envelope* of the three handcrafted rules, $g_{\text{env}} = \min\{g_{\text{none}}, g_{\text{func}}, g_{\text{grad}}\}$, which is a strictly stronger competitor than any fixed single rule since it is allowed to select the best criterion separately on each instance.

Full results. Table 4 reports the terminal gaps of the learned policy against the three handcrafted rules.

F Experiment Details for Real-Measurement Computed Tomography

This appendix collects the details deferred from Section 4.1, the second zero-shot transfer experiment, complementary to Appendix E: the same synthetic-QP training pipeline is applied, without retraining, to ridge reconstruction from *physically measured* X-ray tomography data.

Problem instances. We use three open tomographic datasets released by the Finnish Inverse Problems Society: a walnut scanned at an 82×82 reconstruction grid (Hämäläinen et al., 2015),

Table 4: Terminal gap $f(x_N) - f^\star$ on real-data logistic regression (74) ($m = 1000$, $D = 2000$). Each dataset occupies two rows, one per regime: the **first row** is the regularization level $\lambda_{\text{rel}} = 10^{-4}$ with iteration budget $N = 500$, and the **second row** is $\lambda_{\text{rel}} = 3 \cdot 10^{-5}$ with $N = 1000$; recall $\lambda = \lambda_{\text{rel}} L_0$.

Dataset	No restart	Function	Gradient	Learned
a9a	2.89e-04	1.16e-04	1.13e-04	**8.19e-05**
	1.74e-04	1.74e-04	1.74e-04	**2.33e-05**
w8a	3.15e-05	2.14e-05	2.08e-05	**7.82e-06**
	1.30e-05	3.51e-06	3.45e-06	**1.55e-06**
ijcnn1	1.45e-04	9.52e-05	7.73e-05	**4.35e-05**
	7.48e-05	5.02e-05	4.93e-05	**1.00e-05**
covtype	2.64e-04	1.66e-04	1.63e-04	**7.41e-05**
	1.71e-04	6.90e-05	6.67e-05	**2.81e-05**
phishing	1.06e-04	4.59e-05	4.54e-05	**3.03e-05**
	6.71e-05	5.06e-05	5.04e-05	**1.07e-05**

and the carved-cheese phantom scanned at a 128×128 grid in both a full-angle and a limited-angle configuration (Bubba et al., 2017). Each dataset supplies a sparse fan-beam system matrix $A \in \mathbb{R}^{m \times d}$ and the measured sinogram $b \in \mathbb{R}^m$; no forward projection is simulated and no synthetic noise is added. We denote the three problems `walnut82`, `cheeseF45` and `cheeseL45`. Their dimensions are $m \times d = 9840 \times 6724$ with $9.70 \cdot 10^5$ nonzeros (1.47% dense, 120 projection angles $\times$ 82 detector bins) for `walnut82`, and 44505×16384 with $8.54 \cdot 10^6$ nonzeros (1.17% dense, 45 projection angles $\times$ 989 detector bins) for both cheese problems. The two cheese operators have identical size and differ only in angular coverage: `cheeseF45` distributes its 45 angles over the full circle, whereas `cheeseL45` confines them to a restricted arc, which makes the limited-angle operator substantially more ill-posed. This is a standard difficulty axis in tomography and is the reason both configurations are reported.

The reconstruction problem is the ridge least-squares objective

$$\min_{x \in \mathbb{R}^d} \; f(x) = \frac{1}{2m}\|Ax - b\|^2 + \frac{\lambda}{2}\|x\|^2, \qquad \lambda = \lambda_{\text{rel}} L_0, \quad L_0 = \sigma_{\max}^2(A)/m, \tag{77}$$

where L_0 is computed by 300 power iterations on $A^\top A$ and cached. Since $\nabla^2 f = \frac{1}{m} A^\top A + \lambda I$, the instance is $(L_0 + \lambda)$-smooth and λ-strongly convex, so the effective condition number is $\kappa_{\text{eff}} = (L_0 + \lambda)/\lambda = 1 + 1/\lambda_{\text{rel}}$ exactly as in Appendix E. The three datasets differ by an order of magnitude in the scale of the data themselves — $L_0 = 0.254$ for `walnut82` against 10.8 and 11.4 for `cheeseF45` and `cheeseL45`, with sinogram entries in $[0, 0.92]$ versus $[0.013, 2.45]$ — so a policy that is not genuinely scale-free cannot transfer across them.

A *configuration* is a triple (dataset, λ_{rel}, N), and each configuration contains 8 instances that share (A, b, λ) and differ only in the initial point, $x_0 = c_j \cdot \frac{\|x^\star\|}{\sqrt{d}} \cdot \xi_j$ with $\xi_j \sim \mathcal{N}(0, I_d)$ drawn from a fixed seed and c_j alternating over $\{0.5, 2.0\}$ across $j = 0, \ldots, 7$, so that half the instances start closer to $x^\star$ than the solution's own coordinate scale and half start further away. This makes the 8 instances a controlled sweep over initialization distance rather than 8 i.i.d. repetitions.

Exact minimizer. Unlike (74), (77) is a quadratic, so $x^\star$ solves the normal equations $\big(\frac{1}{m} A^\top A + \lambda I\big) x = \frac{1}{m} A^\top b$. We solve them by conjugate gradients to machine precision (`rtol` $= 10^{-16}$) and certify the result by the strong-convexity bound $f(x) - f^\star \le \|\nabla f(x)\|^2/(2\lambda)$, which is evaluated at the returned iterate and asserted to be below $10^{-13} f^\star$; the measured certificates lie between $6.7 \cdot 10^{-29}$ and $8.6 \cdot 10^{-27}$, i.e. at least 14 orders of magnitude below the smallest gap appearing in Table 1. The reported gaps therefore involve no long reference run, which matters here because the smallest entry of the table is of order 10^{-13}. As a sanity check on the reconstructions themselves, the relative data residual $\|Ax^\star - b\|/\|b\|$ at $\lambda_{\text{rel}} = 3 \cdot 10^{-4}$ is 2.8%, 4.4% and 7.2% on `walnut82`, `cheeseL45` and `cheeseF45`.

Training distribution and policy. The policy is trained exclusively on synthetic diagonal quadratics of the form (67); no tomographic data, no measured operator and no sparse problem is seen during

training. Per problem the training distribution draws a dimension $d \in \{50, 200, 500\}$, a condition number $\kappa \in \{300, 10^3, 3 \cdot 10^3, 10^4\}$, a spectrum shape in {geometric, power, log-random, bulk-tail}, and an initialization scale $x_0 \sim \mathrm{Unif}([-c, c]^d)$ with $c \in \{5, 20\}$; each problem is a batch of 6 instances. The feature map is the scale-free map (75) and the network is the same two-layer $\tanh$ MLP as in Appendix E (7-32-1, 289 parameters, output bias initialized to $+0.5$ so the initial policy never restarts); all arithmetic is in float64.

Training protocol. Training follows the two stages of Appendix E. The *warm-start* stage rolls out the function-restart rule on 24 sampled training problems and fits the network for 300 full-batch Adam steps at learning rate 10^{-2} on the positively reweighted binary cross-entropy. The *fine-tuning* stage then trains end-to-end through the surrogate gradient of Definition 3.2 for 40 epochs of 5 freshly sampled problems each, minimizing the clipped log-ratio objective (76) with AMSGrad at learning rate 10^{-3} and gradient-norm clipping at 5; the checkpoint with the best score on 8 held-out synthetic validation problems is retained. No tomographic result is used at any point to select the checkpoint.

Evaluation protocol. All methods are run on the same instance, from the same x_0, with the same stepsize $s = 1/(L_0 + \lambda)$, the same NAG momentum $\beta_k = (k-1)/(k+2)$ and the same horizon $N = 800$; the only difference is the restart rule. Per-configuration numbers are geometric means of the terminal gap $f(x_N) - f^\star$ over the 8 instances. As in Appendix E we additionally compare against the instance-wise *envelope* $g_{\mathrm{env}} = \min\{g_{\mathrm{none}}, g_{\mathrm{func}}, g_{\mathrm{grad}}\}$, a strictly stronger competitor than any single handcrafted rule.

Results. Table 1 reports the terminal gaps. The learned policy attains the smallest gap in all six configurations. Against the envelope it improves the geometric-mean terminal gap by $23.6\times$ (range $4.5\times$ to $295\times$) and wins 40 of the 48 paired comparisons.

G Experiment Details for LP

This appendix collects the details deferred from Section 4.2: the PDHG backbone used in our implementation and its associated runtime state, the default FirstOrderLP restart logic, the feature design and architecture of the learned monitor, the synthetic LP generation and training setup, and the precise definition of each curve in Figure 3.

G.1 Problem Generation

Our best small-scale result is obtained on a synthetic dense LP family with $(m, n) = (6, 12)$. Each instance is a standard-form primal LP

$$\min_{x \in \mathbb{R}^n} c^\top x \quad \text{s.t.} \quad Ax = b, \quad x \geq 0, \tag{78}$$

with dual

$$\max_{y \in \mathbb{R}^m,\, s \in \mathbb{R}^n} -b^\top y \quad \text{s.t.} \quad c + A^\top y = s, \quad s \geq 0. \tag{79}$$

Then $s_{\mathrm{d}}(y^\star) = c + A^\top y^\star = s^\star$, so $(x^\star, y^\star, s^\star)$ satisfies primal feasibility, dual feasibility, and complementarity exactly. A problem family is defined by the dimensions (m, n). We first sample a matrix $\widetilde{A} \in \mathbb{R}^{m \times n}$ entrywise from $\mathcal{N}(0, 1)$ and normalize by its spectral norm:

$$A = \widetilde{A} / \sigma_{\max}(\widetilde{A}). \tag{80}$$

We then construct a certified primal–dual optimal solution $(x^\star, y^\star, s^\star)$ and back-solve for (b, c). Specifically, we sample an active support set $S \subseteq \{1, \ldots, n\}$ of size

$$|S| = \min\big(n, \max(m + 2, \lfloor n/3 \rfloor)\big), \tag{81}$$

set $x_i^\star \sim \mathrm{Unif}[0.25, 1.25]$ for $i \in S$ and $x_i^\star = 0$ otherwise, sample $y^\star \sim \mathcal{N}(0, 0.6^2 I_m)$, set $s_i^\star = 0$ on S, and sample $s_i^\star \sim \mathrm{Unif}[0.25, 1.25]$ on the inactive coordinates. We then define

$$b = Ax^\star, \qquad c = s^\star - A^\top y^\star. \tag{82}$$

By construction, $(x^\star, y^\star, s^\star)$ satisfies primal feasibility, dual feasibility, and complementarity exactly, so each generated instance comes with a certified KKT point.

For each LP instance, the initial state is

$$z_0 = (x_0, y_0), \qquad x_0 \sim \mathrm{Unif}([-2.5, 2.5]^n), \qquad y_0 \sim \mathrm{Unif}([-2.5, 2.5]^m), \tag{83}$$

with independent coordinates.

G.2 PDHG Backbone and Restart Score

We describe the primal–dual algorithm for LP as implemented in our codebase, following the FirstOrderLP(FOLP) algorithm from Applegate et al. (2023).

Scaled state and PDHG update. The implementation first applies Ruiz scaling to the LP data: with diagonal row and column rescalings D_r, D_c, the scaled data are $\tilde{A} = D_r A D_c$, $\tilde{b} = D_r b$, $\tilde{c} = D_c c$, and the iterates $(\tilde{x}_k, \tilde{y}_k)$ live in the scaled coordinates $\tilde{x} = D_c^{-1} x$, $\tilde{y} = D_r^{-1} y$. The runtime maintains a scalar stepsize s_k and primal weight ω_k, with primal/dual stepsizes $\tau_k = s_k/\omega_k$ and $\sigma_k = s_k \omega_k$. The PDHG half-step is

$$\begin{aligned} \tilde{x}_{k+\frac{1}{2}} &= \Pi_{\mathbb{R}^n_+}\big(\tilde{x}_k - \tau_k(\tilde{c} + \tilde{A}^\top \tilde{y}_k)\big), \\ \bar{x}_{k+\frac{1}{2}} &= 2\tilde{x}_{k+\frac{1}{2}} - \tilde{x}_k, \\ \tilde{y}_{k+\frac{1}{2}} &= \tilde{y}_k + \sigma_k(\tilde{A}\,\bar{x}_{k+\frac{1}{2}} - \tilde{b}). \end{aligned} \tag{84}$$

An adaptive movement test compares a quadratic primal–dual displacement $m_k = \frac{1}{2}\omega_k \|\Delta x_k\|^2 + \frac{1}{2}\omega_k^{-1}\|\Delta y_k\|^2$ to a cross-interaction term $q_k = |\langle \Delta x_k, \tilde{A}^\top(\tilde{y}_{k+\frac{1}{2}} - \tilde{y}_k)\rangle|$ and accepts the half-step iff $m_k \le \varepsilon$ or $s_k \le m_k/q_k$; otherwise $(\tilde{x}_{k+1}, \tilde{y}_{k+1})$ is set to $(\tilde{x}_k, \tilde{y}_k)$. The stepsize is updated multiplicatively at every iteration toward the local stability limit. Throughout, the runtime maintains weighted running averages

$$\tilde{x}_k^{\mathrm{avg}} = \frac{\sum_{t=1}^k s_t \tilde{x}_t}{\sum_{t=1}^k s_t}, \qquad \tilde{y}_k^{\mathrm{avg}} = \frac{\sum_{t=1}^k s_t \tilde{y}_t}{\sum_{t=1}^k s_t}, \tag{85}$$

which are used at restart checks and as the post-restart anchor.

Stepsize and primal-weight initialization. With the scaled data $(\tilde{A}, \tilde{b}, \tilde{c})$, the runtime is initialized by

$$s_0 = \frac{1}{\max(\|\tilde{A}\|_\infty, 1)}, \qquad \omega_0 = \begin{cases} \pi_{\mathrm{imp}} \cdot \frac{\|\tilde{c}\|_2}{\|\tilde{b}\|_2}, & \text{if both norms are nonzero,} \\ \pi_{\mathrm{imp}}, & \text{otherwise,} \end{cases} \tag{86}$$

where $\|\tilde{A}\|_\infty$ is the maximum absolute row sum of $\tilde{A}$ and π_{imp} is the `primal_importance` hyperparameter. Define the displacement and cross-interaction

$$m_k = \tfrac{1}{2}\omega_k \|\Delta \tilde{x}_k\|_2^2 + \tfrac{1}{2}\omega_k^{-1}\|\Delta \tilde{y}_k\|_2^2, \qquad q_k = |\langle \Delta \tilde{x}_k,\, \tilde{A}^\top(\tilde{y}_{k+\frac{1}{2}} - \tilde{y}_k)\rangle|, \tag{87}$$

with per-step stability limit $s_k^{\max} = m_k/q_k$ when $q_k > \varepsilon$ and $s_k^{\max} = \infty$ otherwise. The stepsize is updated by

$$s_{k+1} = \min\Big\{\big(1 - (k+1)^{-a}\big)\, s_k^{\max},\, \big(1 + (k+1)^{-b}\big)\, s_k\Big\}, \tag{88}$$

with defaults $a = 0.3$, $b = 0.6$.

KKT residual. For a state $z = (x, y)$ in the unscaled coordinates, define the normalized residual blocks

$$\begin{aligned} r_{\mathrm{peq}}(z) &= \tfrac{1}{\sqrt{m}}(Ax - b), & r_{\mathrm{pineq}}(z) &= \tfrac{1}{\sqrt{n}}[-x]_+, \\ r_{\mathrm{dineq}}(z) &= \tfrac{1}{\sqrt{n}}[-c - A^\top y]_+, & r_{\mathrm{comp}}(z) &= \tfrac{1}{\sqrt{n}}\, x \odot (c + A^\top y), \end{aligned} \tag{89}$$

where $[u]_+ = \max(u, 0)$ is applied elementwise. The KKT residual used as our evaluation and training metric is

$$\mathcal{R}(z) = \sqrt{\|r_{\mathrm{peq}}(z)\|_2^2 + \|r_{\mathrm{pineq}}(z)\|_2^2 + \|r_{\mathrm{dineq}}(z)\|_2^2 + \|r_{\mathrm{comp}}(z)\|_2^2}. \tag{90}$$

Duality gap and proxy. The adaptive restart scheme of Applegate et al. (2021) triggers a restart according to the *normalized duality gap*

$$\rho_r(z) \;=\; \frac{1}{r}\max_{\tilde{z}\in B_r(z)\cap Z}\big\{\mathcal{L}(x,\tilde{y})-\mathcal{L}(\tilde{x},y)\big\}, \tag{91}$$

where $B_r(z)=\{\tilde{z}\in Z:\|z-\tilde{z}\|\le r\}$ and $\mathcal{L}(x,y)=c^\top x+y^\top(Ax-b)$. Computing ρ_r requires solving a trust-region subproblem at each restart check. For LP, substituting the Lagrangian into (91) yields

$$\rho_r(z)=\frac{1}{r}\max_{\tilde{x}\ge 0,\,\tilde{y}:\,\|(\tilde{x}-x,\tilde{y}-y)\|_2\le r}\big\{\underbrace{(c+A^\top y)^\top(x-\tilde{x})}_{\text{primal term}}+\underbrace{(Ax-b)^\top(\tilde{y}-y)}_{\text{dual term}}\big\}. \tag{92}$$

Since $\mathcal{L}(x,\tilde{y})-\mathcal{L}(\tilde{x},y)$ is linear in $(\tilde{x},\tilde{y})$, the trust-region maximization in (92) reduces to aligning $(\tilde{x}-x,\tilde{y}-y)$ with the residual vector $(c-A^\top y,\ b-Ax)$ subject to $\tilde{x}\ge 0$ and $\|(\tilde{x}-x,\tilde{y}-y)\|\le r$. In the regime where z is approximately feasible and interior, the non-negativity constraint $\tilde{x}\ge 0$ is inactive and the maximizer simply points along the residual direction, giving

$$\rho_r(z)\;\approx\;\|[-\bar{s}]_+\|_2+\|Ax-b\|_2+(\text{boundary correction from }\tilde{x}\ge 0), \tag{93}$$

where $\bar{s}=c+A^\top y$ is the dual slack (we write $\bar{s}$ to avoid confusion with the stepsize s_k). In other words, ρ_r is well-approximated by the ℓ_2 norm of the KKT residual components.

Motivated by this observation, our implementation replaces the trust-region solve with the following closed-form proxy. Let $(\tilde{x}^r,\tilde{y}^r)$ denote the last restart anchor. For any restart candidate $z=(x,y)$ and anchor $\tilde{z}=(\tilde{x},\tilde{y})$, define the scaled distance

$$\|z-\tilde{z}\|_k:=\sqrt{\frac{\omega_k\|x-\tilde{x}\|_2^2+\omega_k^{-1}\|y-\tilde{y}\|_2^2}{s_k}}, \tag{94}$$

and the LP-specific restart proxy

$$\hat{\rho}(z;\tilde{z})=|c^\top x+b^\top y|+\max(\|z-\tilde{z}\|_k,1)\left(\|r_{\text{peq}}(z)\|_2+\|r_{\text{pineq}}(z)\|_2+\|r_{\text{dineq}}(z)\|_2+\|r_{\text{comp}}(z)\|_2\right). \tag{95}$$

The duality-gap term $|c^\top x+b^\top y|$ bounds the Lagrangian gap that appears in the numerator of (92) for unconstrained dual directions, and the residual terms capture the same constraint-violation and complementarity components that the trust-region maximizer in (92) selects. The $\max(\|z-\tilde{z}\|_k,1)$ prefactor plays the role of the radius r in (91), normalizing the residual contribution by the travel distance from the phase anchor.

We use $\widehat{\rho}$ as a computationally cheap proxy for the localized normalized duality gap used in PDLP. The proxy is designed to capture the same qualitative residual information used by the restart rule, including primal feasibility, dual feasibility, complementarity, and primal–dual objective mismatch. However, it does not solve the trust-region maximization problem that defines the localized normalized duality gap in the original Julia implementation. Therefore, we do not claim that $\widehat{\rho}(z;\tilde{z})/\|z-\tilde{z}\|_k$ is equal or theoretically equivalent to the exact PDLP gap measure. We use it only as a low-cost restart monitor in our Python implementation. Once the residual components are available, evaluating $\widehat{\rho}$ costs $O(n+m)$ and requires no inner solve, which allows restart checks to be performed frequently.

Default FirstOrderLP restart rule. At a restart checkpoint, let $z^r=(\tilde{x}^r,\tilde{y}^r)$ denote the current restart anchor. The default FOLP logic first compares the current iterate and the averaged iterate through

$$\phi_{\text{cur}}=\frac{\hat{\rho}(z_k;\,z^r)}{\|z_k-z^r\|_k^2},\qquad \phi_{\text{avg}}=\frac{\hat{\rho}(z_k^{\text{avg}};\,z^r)}{\|z_k^{\text{avg}}-z^r\|_k^2}, \tag{96}$$

where $z_k=(\tilde{x}_k,\tilde{y}_k)$ and $z_k^{\text{avg}}=(\tilde{x}_k^{\text{avg}},\tilde{y}_k^{\text{avg}})$. The restart candidate is whichever of the two has the smaller ϕ value. If z^{cand} is the chosen candidate, define

$$\rho_k=\frac{\hat{\rho}(z^{\text{cand}};\,z^r)\;/\;\|z^{\text{cand}}-z^r\|_k}{\hat{\rho}_{\text{last}}\;/\;\|\cdot\|_{\text{last}}}, \tag{97}$$

where $(\hat{\rho}_{\text{last}},\|\cdot\|_{\text{last}})$ denote the gap proxy and distance of the previous restart segment. The default FOLP implementation restarts if either the artificial safeguard

$$\ell_k\ge 0.5\,k, \tag{98}$$

where ℓ_k is the current phase length, or the normalized reduction test

$$\rho_k < 0.9 \quad \text{and} \quad \big(\rho_k < 0.1 \text{ or } \rho_k > \rho_{k-1}^{\text{trial}}\big). \tag{99}$$

After restart, the iterate is reset to the chosen candidate, the averaging buffers (85) are cleared, and the primal weight is updated from the primal/dual movement ratio with smoothing coefficient 0.5.

G.3 Learning Restart Policy for LP

Network architecture. The policy is a small MLP $V(z;\theta) = \mathsf{MLP}(\phi(z);\theta)$ with 4 layers (embedding dimensions 10, 8, 7, 4) and ReLU activations. The feature vector $\phi(\bar{x}_k, \bar{x}_{N_i}, k) \in \mathbb{R}^{10}$ has entries being the logarithm of: the duality gap $|c^\top \bar{x}_k + b^\top \bar{y}_k|$, the residual sum $\|r_{\text{peq}}\| + \|r_{\text{pineq}}\| + \|r_{\text{dineq}}\| + \|r_{\text{comp}}\|$, the primal/dual travel distances $d_{x,k} = \|\tilde{x}_k^{\text{avg}} - \tilde{x}_{N_i}\|$ and $d_{y,k} = \|\tilde{y}_k^{\text{avg}} - \tilde{y}_{N_i}\|$, the runtime stepsize s_k and primal weight ω_k, summary statistics $\widehat{G}_{k-1}, \widehat{D}_{k-1}, \widehat{\rho}_{k-1}$ from the previous phase, and the relative phase length $\varphi_k = (k - N_i)/\max(k, 1)$. The first output coordinate is the restart logit; the remaining three outputs predict log-scale phase-summary statistics ($\log \widehat{G}$, $\log \widehat{D}$, $\log \widehat{\rho}$) used to update the monitor state after each restart. The total number of trainable parameters is 183.

Training objective. The training objective is the weighted-path KKT residual

$$J(\theta) = \mathbb{E}_{P,z_0}\left[\sum_{k=1}^{N} w_k \, \mathcal{R}\big(\bar{X}(k;\theta)\big)\right], \qquad w_k = \frac{2k}{N(N+1)}, \tag{100}$$

with $\mathcal{R}$ as in (90) and $w_k \propto k$ up-weighting later iterates. Training proceeds in two stages. We first *warm-start* the network on a dataset of feature–label pairs $\mathcal{D} = \{(\phi_i, y_i)\}_{i=1}^{M}$ with $M = 880$ examples collected from the baselines; the warm-start is fitted by minimizing the binary cross-entropy with logits. Starting from the warm-started parameters, we then *fine-tune* the network using the surrogate gradients of Definition 3.2 on the loss (100) for 6 epochs with AdamW (learning rate 3×10^{-4} and weight decay 10^{-4}). The selected checkpoint is epoch 5, chosen by validation performance.

Baseline set. We compare against three FirstOrderLP baselines and their pointwise best envelope:

1. `FOLP-default`: the default FirstOrderLP reference solver;
2. `FOLP-freq200`: the same FirstOrderLP solver but with restart checks every 200 iterations;
3. `FOLP-no-restart`: the same FirstOrderLP solver with restart checks effectively disabled.
4. `Best-baseline-env`: the pointwise-best KKT residual achieved by the above three baselines at each iteration, serving as a conservative comparison envelope.

Out-of-distribution performance. To test generalization beyond the training scale, we evaluate the learned policy on synthetic LP instances with $n = 18$ variables and $m = 6$ constraints—50% larger than the training instances—without any additional fine-tuning. As shown in Figure 6, the learned restart policy continues to match or improve upon the best-baseline envelope on these out-of-distribution instances, indicating that the policy captures transferable phase-level restart structure rather than overfitting to the training problem size.

G.4 Netlib Benchmark: Policy Class, Training, and Evaluation

This appendix details the Netlib experiment reported in Section 4.2. It differs from the synthetic-LP study above in three respects: the instances are real benchmark LPs, the policy class is an affine score rather than an MLP with monitor state, and the policy is trained on a small, a-priori-chosen subset of the benchmark and evaluated on the rest.

Policy class. Every `check_every` iterations the solver asks whether to restart, i.e. whether to take the running average as the new anchor. The learned score is affine in standardized features,

$$V(\phi_k;\theta) = w^\top \frac{\phi_k - \mu}{\sigma} + b, \qquad \text{restart} \iff V(\phi_k;\theta) \le 0, \tag{101}$$

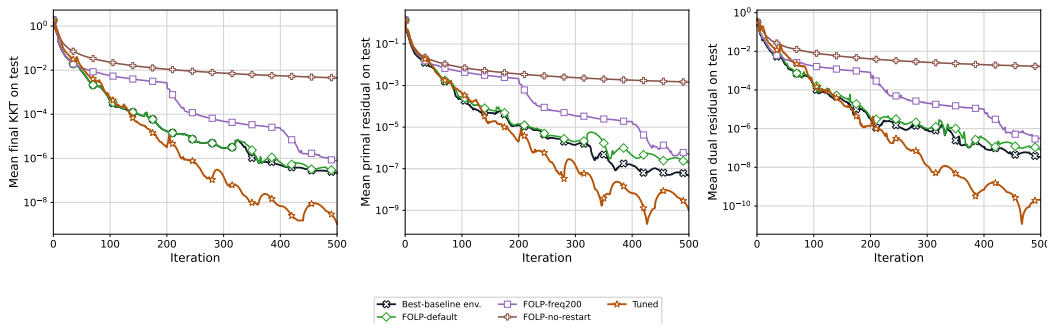


Figure 6: KKT residual (*left*), primal residual (*middle*), and dual residual (*right*) vs. iteration for the learned restart policy and FirstOrderLP (Applegate et al., 2023) baselines on synthetic LPs (dimension $n = 18$, $m = 6$).

with $\phi_k \in \mathbb{R}^{15}$ the phase features listed below and $\theta = (w, b) \in \mathbb{R}^{16}$ the only trainable parameters.

The reason for this restricted class is that it *contains the handcrafted rule as a special case*: taking w supported on the coordinate $\log \rho_k$ and $b = \mu_{14} - \log 0.85$ recovers the PDLP sufficient-decrease test $\rho_k \le 0.85$ exactly. Training can therefore be initialized at the baseline rule and can be read as improving a known rule inside a class that contains it, rather than fitting an unconstrained function that might be worse. With 16 parameters and 7 training instances the capacity is also small enough that overfitting is implausible, and the fitted weights are directly interpretable. We note explicitly that this is a simplification relative to the architecture of Appendix G; we adopt it because it suffices on these features, not because the MLP fails.

Features. The feature map $\phi_k \in \mathbb{R}^{15}$ collects, all on a logarithmic scale: the duality gap and residual sum at the current iterate; the KKT residuals R_{cur} and R_{avg} of the current and averaged iterates; the squared primal and dual movements $\|\Delta x\|^2$, $\|\Delta y\|^2$ within the current phase; the stepsize s and primal weight ω; the candidate's normalized duality gap η, the value η_{last} from the previous phase, and the previous trial ratio ρ_{trial}; the phase length ℓ and the relative phase length ℓ/k; the per-iteration decrease rate of $\log \eta$; and finally the test statistic of the handcrafted rule, $\log \rho_k$ with $\rho_k = \eta/\eta_{\text{last}}$, which is the coordinate that makes (101) contain the PDLP rule.

Training instances. Of the 86 Netlib instances that our parser handles, we take the 7 with the fewest nonzeros in the MPS file: `afiro`, `sc50b`, `sc50a`, `kb2`, `sc105`, `adlittle`, `scagr7`. The selection rule is fixed in advance and uses no evaluation outcome. These seven instances are removed from every statistic we report. No mid- or large-tier instance is used for training, so both of those tiers are size extrapolation.

Randomized initial points. Rather than training from one fixed starting point, each rollout draws a fresh initialization,

$$x_0 \sim \text{Unif}[0, x_{\text{hi}})^n, \quad y_0 \sim \mathcal{N}(0, \sigma_y^2)^m, \quad x_{\text{hi}} \sim \text{LogUnif}[0.5, 8], \quad \sigma_y \sim \text{Unif}[0, 2], \tag{102}$$

so that the magnitudes themselves vary across rollouts and the policy sees a wide family of trajectories instead of a single one.

Objective and gradients. We minimize the same weighted-path KKT residual as in (100), in logarithmic form so that instances of different difficulty contribute on a common scale,

$$\mathcal{L}(\theta) = \sum_{k=1}^{N} w_k \log R_k, \qquad w_k = \frac{2k}{N(N+1)}, \tag{103}$$

with the linearly increasing weights biasing the optimization towards terminal accuracy. Restart times are integer-valued and hence not differentiable in θ; gradients are obtained with the surrogate of Definition 3.2.

Optimization. The parameters are initialized at the PDLP rule as described above. We train for 40 epochs with AdamW (learning rate 3×10^{-4}, weight decay 10^{-4}, gradient-norm clipping at 1.0). Checkpoints are selected on a fixed panel of 7 instances $\times$ 3 held-out initial points (seeds 2001–2003) that never enter any gradient.

Evaluation protocol. Iteration budgets are $N = 20000$, 25000 and 50000 for the small, mid and large tier, with `check_every` $= 1$. Each instance is run from 4 independent random initial points (seeds 0–3); a cell of Table 2 is the median across seeds within each instance, then the median across instances in the tier. To avoid letting non-converged runs dominate the medians, an instance is dropped from the comparison—for *all* methods simultaneously—whenever any method ends above a KKT residual of 10^5, leaving $n = 32$, 24 and 10 instances. The baselines are the PDLP default rule, periodic restarting every 200 iterations, and no restarting.