Silver Rate Is (Almost) Optimal for Gradient Descent: The Strongly Convex Case

Kaizhao Liu*
MIT
mrzt@mit.edu

Yuhan Ye*
MIT
yyh03@mit.edu

September 22, 2026

Abstract

Understanding the limitations of gradient descent (GD) has long been a central problem in optimization. On smooth strongly convex objectives, a recent breakthrough by Altschuler and Parrilo [AP25a] showed that GD can be accelerated to $O(\kappa^{1/p_{\mathrm{sil}}}\log(1/\varepsilon))$ iterations using the *silver stepsize schedule*, where κ is the condition number, ε is the target relative accuracy, and $p_{\mathrm{sil}} = \log_2(1+\sqrt{2})$, improving the polynomial dependence on κ compared to the classical $O(\kappa\log(1/\varepsilon))$ bound for constant-step GD. However, the best-known lower bound has remained $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$, established by Nemirovsky and Yudin [NY79] over four decades ago.

In this paper, we prove that GD with any predetermined stepsize schedule requires

$$\Omega\left(\kappa^{\frac{1}{p_{\mathrm{sil}}}-o(1)}\log\frac{1}{\varepsilon}\right)$$

iterations to reduce either the squared distance to the minimizer or function suboptimality by a factor ε. This matches the polynomial exponent of κ achieved by the silver stepsize schedule.

As in our previous work on smooth convex objectives [YL26b], we construct the hard instance from a chain of *local hard functions*, where the next coordinate in the chain becomes active at selected steps called *checkpoints*. The main difficulty arises from the additional quadratic term required by strong convexity, whose step-by-step contraction of the iterates toward the minimizer can destroy the lower bound if the total stepsize between consecutive checkpoints is too large. We overcome this by sequentially scanning the stepsize schedule *block by block* to choose the checkpoints, while inserting *repair components* at carefully chosen steps whenever the accumulated unselected stepsize grows too large.

*Authors are listed in random order.

Contents

1 Introduction **3**
- 1.1 Our Contributions . . . 4
- 1.2 Related Work . . . 6
  - 1.2.1 The special case of quadratic optimization. . . . 6
  - 1.2.2 General smooth (strongly) convex optimization. . . . 6
  - 1.2.3 Performance estimation. . . . 7
  - 1.2.4 Asymptotic lower bounds for GD. . . . 7

2 Overview of Proof Idea **8**
- 2.1 A Review of Existing Techniques . . . 8
- 2.2 Our Construction: Handling Quadratic Contraction . . . 11

3 Local Hard Functions: Bending, Repairing, and Bridging **13**
- 3.1 Bending . . . 15
- 3.2 Repairing . . . 15
- 3.3 Bridging . . . 16

4 Selecting Checkpoints for Every Schedule **17**
- 4.1 Choosing Checkpoints within a Block . . . 17
- 4.2 Repairing Accumulated Unselected Steps . . . 18

5 Assembling the Hard Function and Proving Theorem 1.1 **19**
- 5.1 Constructing Components at a Maximizing Checkpoint Set . . . 19
- 5.2 Constructing the Global Hard Function . . . 20

6 Concluding Remarks **24**

A Proofs for the Local Hard Functions **29**
- A.1 Proof of Lemma 3.1 . . . 29
- A.2 Proof of Lemma 3.2 . . . 31
- A.3 Proof of Lemma 3.3 . . . 32

B Proof of Lemma 4.1 **37**

C Proof of Lemma 4.2 **45**

1 Introduction

Gradient descent (GD) is one of the simplest and most fundamental algorithms in optimization. To minimize a differentiable function $F : \mathbb{R}^d \to \mathbb{R}$, GD proceeds via the iteration

$$x_t = x_{t-1} - h_t \nabla F(x_{t-1}), \qquad 1 \le t \le n,$$

where the stepsize h_t controls how far the iterate moves along the negative gradient direction.

In this paper, we study the fundamental limits of GD for smooth strongly convex minimization when its *stepsize schedule* $H = (h_1, \dots, h_n) \in [0,\infty)^n$ is predetermined. Throughout, we assume F is μ-strongly convex and L-smooth, with unique minimizer x_*. Let $\kappa := L/\mu$ denote the condition number.

Classical Upper Bounds. The convergence analysis of GD has a long history. For quadratic objectives, Young [You53] showed that Chebyshev stepsizes reduce the squared distance to the minimizer by a factor of $0 < \varepsilon \le 1/4$ in $O(\sqrt{\kappa}\log(1/\varepsilon))$ iterations. [1]

For general smooth strongly convex objectives, classical guarantees for constant-step GD are established by bounding the progress of each iteration separately [Pol63]. This yields an $O(\kappa\log(1/\varepsilon))$ iteration complexity for reducing either the squared distance to the minimizer or function suboptimality by a factor $0 < \varepsilon \le 1/4$; see also the textbooks [Pol87, Nes04, Bub15].

Further improving this rate, particularly its dependence on κ, has long motivated the development of accelerated first-order methods. Standard approaches often add momentum or auxiliary sequences, as in Polyak's heavy-ball method [Pol64] and Nesterov's accelerated method [Nes83]. The latter achieves an $O(\sqrt{\kappa}\log(1/\varepsilon))$ iteration complexity. These methods, however, modify the original GD update.

Acceleration via Stepsize Schedules. Compared with general smooth strongly convex objectives, quadratic objectives have additional structure [Alt18, Chapter 8], allowing Chebyshev stepsizes to attain the accelerated $O(\sqrt{\kappa}\log(1/\varepsilon))$ iteration complexity discussed above. Beyond quadratics, however, whether predetermined stepsizes alone could accelerate GD remained open for decades.

Altschuler observed that analyzing steps individually fails to capture the benefits from coordinating short and long steps in his thesis [Alt18, Chapter 8]. A joint analysis of multiple steps is therefore necessary to establish any improvement over the best constant stepsize. Building on this observation, Altschuler and Parrilo [AP25a] achieved a major breakthrough by introducing the recursive *silver stepsize schedule*. This schedule improves the iteration complexity from $O(\kappa\log(1/\varepsilon))$ to $O(\kappa^{1/p_{\mathrm{sil}}}\log(1/\varepsilon))$, where

$$p_{\mathrm{sil}} := \log_2(1+\sqrt{2}) \approx 1.27155.$$

In the same work, they also conjectured its optimality. The classical first-order lower bound $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$ [NY79, Nes04] leaves a polynomial gap with respect to the silver upper bound. Therefore, the central question is whether we can close this gap by proving a stronger lower bound for GD.

Progress on Lower Bounds. We briefly review the history of lower bound constructions and the foundational ideas that motivate our approach. For a quadratic objective $F(x) = \frac{1}{2}(x-x_*)^\top A(x-x_*)$, the component of the error along an eigenvector of the Hessian A with eigenvalue λ is scaled by $\prod_{t=1}^n (1-h_t\lambda)$ after n GD steps. Among predetermined stepsize schedules, the Chebyshev schedule minimizes the maximum absolute value of this product over $\lambda \in [\mu, L]$, yielding the tight $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$ lower bound [Gav50].

[1] These stepsizes are the reciprocals of the roots of a degree-n Chebyshev polynomial of the first kind, shifted and scaled to $[\mu, L]$.

Two one-dimensional quadratics are sufficient to yield a lower bound for constant-step GD for the general class of smooth strongly convex functions. Indeed, any constant stepsize h satisfies [CC55]

$$\max\{|1-h\mu|,|1-hL|\} \geq \frac{\kappa-1}{\kappa+1}.$$

Thus, for at least one of the functions $F(x) = \mu x^2/2$ and $F(x) = Lx^2/2$, the squared distance after n steps remains at least $((\kappa-1)/(\kappa+1))^{2n}$ times the initial squared error. These examples yield the $\Omega(\kappa\log(1/\varepsilon))$ lower bound for constant-step GD on the full function class. See [Alt18, Section 2.1] for details.

For the broader class of *deterministic first-order methods*, which access the objective through function values and gradients at arbitrary points, Nemirovsky and Yudin [NY79] established the classical $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$ lower bound. Their *resisting oracle* chooses the unrevealed coordinate directions as new queries arrive, even when those queries depend adaptively and nonlinearly on previous function values and gradients. These choices keep the unrevealed part of the minimizer orthogonal to the queried points and returned gradients, without changing the function values or gradients at any previously queried point. For methods satisfying the *linear-span assumption*[2], Nesterov [Nes04] provides a simpler quadratic-chain construction. By initializing at the origin, each iteration discovers at most one new coordinate direction, while the minimizer has nonzero coordinates further along the chain. The gap between the optimal objective value over the first n coordinates and that over the full space establishes the desired lower bound.

Subsequent work established finite-horizon lower bounds with exact constants for squared distance to the minimizer [DT22] and function suboptimality [Dro26]. These works also employ a resisting-oracle argument similar to that of Nemirovsky and Yudin [NY79] to establish their lower bounds for arbitrary deterministic first-order methods. To obtain sharp constants, [DT22] constructs smooth strongly convex functions from finitely many points alongside their corresponding function values and gradients. This approach is intimately connected to the interpolation techniques used in *performance estimation problems* (PEPs), discussed in Section 1.2.3.

Our recent work [YL26b] proved an $\Omega(n^{-p_{\text{sil}}-o(1)})$ lower bound in the smooth convex setting, matching the polynomial exponent of the corresponding silver upper bound. Although upper bounds can be transferred between smooth convex and strongly convex optimization via standard black-box reductions [AZH16], these techniques do not readily yield an analogous lower bound in the strongly convex regime. Instead, one should manually extend the lower-bound construction, where the primary difficulty becomes accommodating the quadratic term required to ensure strong convexity.

1.1 Our Contributions

We prove a lower bound of $\Omega(\kappa^{1/p_{\text{sil}}-o(1)}\log(1/\varepsilon))$ for the iteration complexity of GD with predetermined nonnegative stepsizes. This matches the silver exponent in the upper bound of Altschuler and Parrilo [AP25a].

To state our results, let $\mathscr{F}_{\mu,L}(\mathbb{R}^d)$ denote the class of differentiable μ-strongly convex functions on $\mathbb{R}^d$ with L-Lipschitz gradients. For a normalized[3] schedule H, which may depend on κ and the horizon n, define the worst-case ratios for squared distance to the minimizer and function suboptimality, each relative to its

[2]The assumption is $x_t \in x_0 + \text{span}\{\nabla F(x_0),\dots,\nabla F(x_{t-1})\}$ [Nes04, Assumption 2.1.4].

[3]Dividing the objective by L and multiplying each stepsize by L preserves the iterates. We therefore normalize $L = 1$ and $\mu = 1/\kappa$ for our strongly convex results.

initial value, by

$$D_{n,\kappa}(H) := \sup_{d\in\mathbb{N}} \sup_{F\in\mathscr{F}_{1/\kappa,1}(\mathbb{R}^d)} \sup_{x_0\neq x_*} \frac{\|x_n - x_*\|^2}{\|x_0 - x_*\|^2}, \tag{1}$$

$$E_{n,\kappa}(H) := \sup_{d\in\mathbb{N}} \sup_{F\in\mathscr{F}_{1/\kappa,1}(\mathbb{R}^d)} \sup_{x_0\neq x_*} \frac{F(x_n) - F(x_*)}{F(x_0) - F(x_*)}.$$

Both ratios satisfy the following lower bound.

Theorem 1.1 (Convergence lower bound)**.** *There are absolute constants $C_1, C_2, \kappa_0 > 0$ such that, for every $\kappa \geq \kappa_0$, every integer $n \geq 1$, and every predetermined nonnegative schedule $H \in [0,\infty)^n$,*

$$\min\{D_{n,\kappa}(H), E_{n,\kappa}(H)\} \geq \exp\left[-C_1 n\, \kappa^{-1/p_{\mathrm{sil}}} \log\kappa\, e^{C_2\sqrt{\log\kappa}}\right]. \tag{2}$$

For a target reduction factor $0 < \varepsilon < 1$, define the iteration complexities by

$$T_D(\kappa,\varepsilon) := \inf\{n \in \mathbb{N}_{\geq 1} :\ \exists H \in [0,\infty)^n \text{ with } D_{n,\kappa}(H) \leq \varepsilon\},$$
$$T_E(\kappa,\varepsilon) := \inf\{n \in \mathbb{N}_{\geq 1} :\ \exists H \in [0,\infty)^n \text{ with } E_{n,\kappa}(H) \leq \varepsilon\}.$$

Corollary 1.2 (Iteration complexity)**.** *There are absolute $c, C > 0$ such that, for sufficiently large κ and every $0 < \varepsilon < 1$,*

$$\min\{T_D(\kappa,\varepsilon), T_E(\kappa,\varepsilon)\} \geq c\, \frac{\kappa^{1/p_{\mathrm{sil}}}}{e^{C\sqrt{\log\kappa}}\log\kappa} \log\frac{1}{\varepsilon}. \tag{3}$$

The $\Omega(\kappa^{1/p_{\mathrm{sil}}-o(1)}\log(1/\varepsilon))$ iteration lower bound in Corollary 1.2 matches the $O(\kappa^{1/p_{\mathrm{sil}}}\log(1/\varepsilon))$ upper bound of [AP25a] up to a subpolynomial factor in κ.[4] The lower and upper bounds contain the same multiplicative factor $\log(1/\varepsilon)$. Table 1 compares our iteration lower bound with the best known asymptotic upper and lower bounds across different objective and algorithm classes.

Objective class	Algorithm class	Lower bound	Upper bound
General	Constant-step GD	$\Omega(\kappa\log(1/\varepsilon))$ [CC55]	$O(\kappa\log(1/\varepsilon))$ [Pol63]
Quadratic	Predetermined-step GD	$\Omega\big(\sqrt{\kappa}\log(1/\varepsilon)\big)$ [Gav50]	$O\big(\sqrt{\kappa}\log(1/\varepsilon)\big)$ [You53]
General	Deterministic first-order methods	$\Omega\big(\sqrt{\kappa}\log(1/\varepsilon)\big)$ [NY79]	$O\big(\sqrt{\kappa}\log(1/\varepsilon)\big)$ [Nes83]
General	Predetermined-step GD	$\Omega\left(\frac{\kappa^{1/p_{\mathrm{sil}}}\log(1/\varepsilon)}{e^{C\sqrt{\log\kappa}}\log\kappa}\right)$ **This work**	$O\big(\kappa^{1/p_{\mathrm{sil}}}\log(1/\varepsilon)\big)$ [AP25a]

Table 1: Iteration complexity on smooth strongly convex objectives.[5]

[4] The gap to the silver upper bound is the factor $e^{C\sqrt{\log\kappa}}\log\kappa = \kappa^{o(1)}$ in (3).

[5] All bounds hold for sufficiently large κ and concern reducing squared distance to the minimizer by a factor $0 < \varepsilon \leq 1/4$. Some classical bounds are folklore. With limited access to early literature, we cite the earliest relevant sources we could verify.

1.2 Related Work

1.2.1 The special case of quadratic optimization.

On a quadratic objective with Hessian $A = \nabla^2 F$, the error $x_n - x_*$ after n GD steps is a polynomial in A applied to the initial error $x_0 - x_*$. Young [You53] achieved the optimal $O(\sqrt{\kappa}\log(1/\varepsilon))$ complexity by choosing stepsizes as reciprocals of the roots of a shifted and scaled Chebyshev polynomial. Early formulations of the associated Chebyshev minimax problem appear in [Gav50, GV61], with a modern treatment in [Alt18]. Conjugate gradients use a three-term recurrence, with coefficients that adapt to the particular problem instance [HS52]. Polyak's heavy-ball method adds momentum to GD and, with suitable fixed parameters, attains the accelerated asymptotic distance factor $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ on strongly convex quadratics [Pol64]. A three-term momentum recurrence with suitably chosen iteration-dependent coefficients can also generate the normalized Chebyshev polynomials.

Another approach uses the arcsine distribution, the limiting distribution of Chebyshev roots, to design randomized stepsize schedules [PZ11, ZPB13, Kal17]. In exact arithmetic, permuting a fixed set of stepsizes does not change GD's final iterate on a quadratic objective. However, the order still affects numerical stability, motivating the study of stable Chebyshev orderings [LF71, AGZ21]. For general nonquadratic objectives, the order can affect convergence itself. This is one reason why the quadratic analysis does not directly carry over to general smooth strongly convex objectives; see [Alt18, Chapter 8] for a comparison.

1.2.2 General smooth (strongly) convex optimization.

With a suitable constant stepsize, GD reaches normalized function suboptimality at most ε in $O(1/\varepsilon)$ iterations for smooth convex objectives.[6] For smooth strongly convex objectives, the iteration bound is $O(\kappa\log(1/\varepsilon))$ for reducing function suboptimality by a factor ε [Pol87, Nes04, Bub15]. Early analyses cover local convergence [CC55], global convergence for strongly convex objectives [Pol63], and projected gradient methods on bounded convex sets [LP66]. An adaptive alternative is exact line search, which chooses the stepsize that minimizes the objective along the current descent direction. However, its tight worst-case iteration complexity on smooth strongly convex objectives remains $\Theta(\kappa\log(1/\varepsilon))$ for reducing function suboptimality by a factor ε [dKGT17].

The first provable gains from predetermined stepsize schedules beyond quadratic objectives were established in Altschuler's thesis [Alt18, Chapter 8], which analyzed two- and three-step schedules in the strongly convex setting. Repeating these short schedules improves constants in the classical $O(\kappa\log(1/\varepsilon))$ iteration complexity. The remaining challenge was to improve the exponent of κ. A breakthrough came when Altschuler and Parrilo [AP25a] introduced the silver stepsize schedule, improving this complexity from $O(\kappa\log(1/\varepsilon))$ to $O(\kappa^{1/p_{\mathrm{sil}}}\log(1/\varepsilon))$. They conjectured that the exponent $1/p_{\mathrm{sil}}$ is optimal among all predetermined stepsize schedules. For separable smooth strongly convex objectives, they further showed that GD with i.i.d. arcsine-distributed inverse stepsizes satisfies $\left(\|x_n - x_*\|/\|x_0 - x_*\|\right)^{1/n} \to (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ almost surely [AP24]. We refer to their survey [AP26] for a broader overview.

In the general smooth convex setting, Grimmer [Gri24] proved that periodic long steps improve the function suboptimality bound by a constant factor when the initial level set is bounded. Altschuler and Parrilo [AP25b] later extended their silver schedule from strongly convex to general smooth convex objectives, achieving normalized function suboptimality at most ε in $O(\varepsilon^{-1/p_{\mathrm{sil}}})$ iterations. [GSW25a] subsequently improved the constant in the bound on function suboptimality and extended the silver rate to squared gradient norm. [WMYZ26] determined the exact worst-case bound on function suboptimality of the original silver schedule at horizons $n = 2^k - 1$. [GSW25b] and [ZJ26] gave rules for combining shorter schedules to obtain the silver rate for arbitrary prescribed horizons. Under these symmetric composition rules,

[6] For L-smooth convex objectives, this accuracy target is $F(x_n) - F(x_*) \le \varepsilon(L/2)\|x_0 - x_*\|^2$.

[LCJW26] proved that recursively joining schedules whose lengths differ by at most one minimizes the worst-case bound on function suboptimality at each horizon. Related developments include anytime acceleration [ZLDC25] and accelerated proximal gradient methods [BA25].

Acceleration through stepsize choice has also been studied in learning problems and min-max optimization. For linear regression, periodic schedules combining short and long steps exploit clustered Hessian spectra to accelerate GD [Oym21]. For linearly separable logistic regression, a large constant stepsize chosen for the iteration horizon yields an $\widetilde{O}(n^{-2})$ loss bound [WBTY24]. Related guarantees hold for certain two-layer neural networks trained with logistic loss [CWM+24]. Large constant stepsizes also accelerate strongly convex regularized logistic regression on linearly separable data under suitable regularization conditions [WMB25]. More recently, prescribed increasing stepsize schedules yield stretched exponential loss decay for separable logistic regression [KMM+26]. Beyond minimization, power-law stepsize schedules accelerate the last-iterate convergence of extragradient for unconstrained biaffine min-max problems [WZCL26].

1.2.3 Performance estimation.

Performance estimation problems (PEPs) formulate the worst-case performance of a given first-order method after a prescribed number of iterations as an optimization problem [DT14]. For *fixed-step methods*[7], *exact interpolation conditions* characterize which function values and gradients at finitely many points can come from a single function in the specified class [THG17]. These conditions allow the PEP to be reformulated as a semidefinite program. Solving the resulting program numerically or analytically can yield a tight convergence guarantee. Optimizing this guarantee over the method's coefficients then provides a systematic way to design first-order methods with optimal convergence guarantees within a chosen class [DT20].

The PEP framework has led to sharper upper bounds and, in several settings, methods that attain the corresponding lower bounds. In the general smooth convex setting, [KF16] used this approach to develop the optimized gradient method (OGM), whose bound on function suboptimality was later proved optimal in both rate and constant [Dro17]. [GJP26] subsequently established OGM's uniqueness in the corresponding minimax problem over fixed-step methods. Branch-and-bound has also been used to numerically solve the minimax problem for GD with $n \leq 25$ steps, yielding predetermined stepsizes that minimize the worst-case function suboptimality [DGVPR24, Section 6.1]. On smooth strongly convex objectives, the PEP framework also yielded a method attaining the exact lower bound on squared distance to the minimizer for deterministic first-order methods [TD23, DT22]. For function suboptimality in the same setting, [Dro26] derived exact lower bounds and methods attaining them. For a broader perspective on minimax optimality, see the survey [GSW26].

1.2.4 Asymptotic lower bounds for GD.

GD with predetermined stepsizes forms a strict subclass of deterministic first-order methods, so classical first-order lower bounds do not determine GD's optimal rate. For smooth convex objectives, [GSW25b] proved that the exponent p_{sil} is optimal for function suboptimality within their recursively generated class of schedules. More recent lower bounds apply to arbitrary predetermined nonnegative schedules. [MC26] proved an $\Omega(n^{-1.9319})$ lower bound on function suboptimality for predetermined nonnegative stepsizes, giving the first polynomial improvement over the classical $\Omega(n^{-2})$ bound. [Tsa26] sharpened this to $\Omega(n^{-\sqrt{3}})$ using the same construction. [YL26a] improved this lower bound to $\Omega(n^{-1.6342})$, even for possibly negative stepsizes. [JCY26] subsequently introduced a family of hard functions built from chains of one-sided Huber components that gives an $\Omega(n^{-\log_2(1+\sqrt{3})})$ lower bound for positive schedules. Our previous work [YL26b]

[7] These methods use predetermined coefficients to combine past gradients and may use momentum.

further refined this construction to obtain an $\Omega(n^{-p_{\mathrm{sil}}-o(1)})$ lower bound. This shows that the polynomial exponent p_{sil} in the $O(n^{-p_{\mathrm{sil}}})$ upper bound is optimal. Section 2.1 reviews these construction ideas, which form the starting point for our strongly convex hard function.

2 Overview of Proof Idea

In this section, we first review the lower bound constructions for smooth convex objectives. We then explain how to adapt our construction [YL26b] to the strongly convex setting. The main difficulty comes from the quadratic term needed to make our hard function strongly convex.

2.1 A Review of Existing Techniques

Nesterov's quadratic zero-chain. The classical $\Omega(n^{-2})$ lower bound for smooth convex optimization is due to Nemirovsky and Yudin [NY79]. Here, we recall the chain construction for methods satisfying the linear-span assumption[2], as presented in [Nes04, Section 2.1.2]. For example, with $d = 2n+1$, consider the convex, 1-smooth function

$$f(x) = \frac{1}{8}\left[(1-x^{(1)})^2 + \sum_{j=1}^{d-1}(x^{(j)}-x^{(j+1)})^2 + (x^{(d)})^2\right].$$

For $0 \le r < d$, if $x^{(r+1)} = \cdots = x^{(d)} = 0$, then $(\nabla f(x))^{(j)} = 0$ for $j \ge r+2$.[8] Consequently, GD started at $x_0 = 0$ satisfies $x_t^{(j)} = 0$ for $j > t$, regardless of the stepsizes. The minimizer has $x_*^{(j)} = 1 - j/(d+1) > 0$ in every coordinate. Minimizing f over vectors whose coordinates after n are zero gives

$$\frac{2[f(x_n) - f(x_*)]}{\|x_0 - x_*\|^2} \ge \frac{3}{16(n+1)^2}.$$

The strongly convex counterpart yields the classical $\Omega(\sqrt{\kappa}\log(1/\varepsilon))$ iteration lower bound [Nes04, Section 2.1.4].

The Checkpoint Method [MC26]. For GD with predetermined nonnegative stepsizes, the classical $\Omega(n^{-2})$ lower bound also applies. Whether this bound could be improved for such methods was a long-standing question. Ma and Chen [MC26] introduced a *checkpoint method* that improved it to $\Omega(n^{-1.9319})$. This method delays the activation of a new coordinate until a selected step, allowing the lower bound to depend on the given stepsizes. We briefly recall the method. Given a stepsize schedule $H = (h_1, \ldots, h_n)$, we select a set of stepsize indices $T = \{t_1 < \cdots < t_k\}$ called *checkpoints*. The sequence of updates between successive checkpoints forms a *gap*. For each chosen checkpoint set, the construction gives a hard function and an associated *score*; larger scores yield stronger lower bounds on function suboptimality. To obtain the best possible lower bound, the checkpoint method seeks to maximize this score over all possible checkpoint sets T.

To make this maximization tractable, these hard functions are constructed so that the score depends solely on the stepsize at each checkpoint and the sum of the stepsizes within each gap. To formalize this, let $t_0 = 0$ and $t_{k+1} = n+1$, and define the *checkpoint stepsizes* b_i and *sums of gap stepsizes* s_i by

$$b_i := h_{t_i} \quad (1 \le i \le k), \qquad s_i := \sum_{t=t_{i-1}+1}^{t_i-1} h_t \quad (1 \le i \le k+1).$$

[8] This is the *first-order zero-chain* property [DT22, Definition 1]: the gradient can introduce at most one new coordinate.

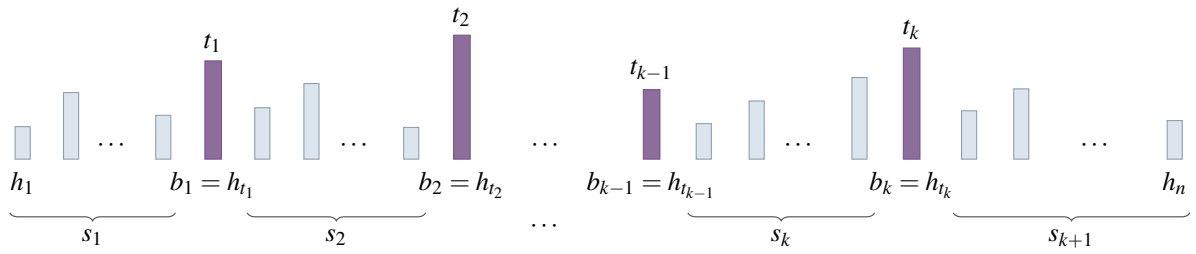


Figure 1: The checkpoint stepsizes are $b_i = h_{t_i}$ (purple). Each brace marks the sum of gap stepsizes s_i.

Ma and Chen [MC26] construct a smooth convex hard function whose gradient remains constant within each gap. Consequently, the displacements within a gap simply accumulate and depend only on the sum of gap stepsizes s_i. At checkpoint t_i, the step of size b_i forces GD to explore a new dimension. This strategy is reminiscent of Nesterov's lower bound construction: to sustain a strong lower bound, GD cannot be allowed to make continuous progress in the same dimension throughout the run. For a geometric explanation of Ma and Chen's construction, see [YL26a, Appendix A]. When identifying a checkpoint set that maximizes the score, this intuition also provides a natural heuristic: updates with larger stepsizes are strong candidates. If an update features a large stepsize while the gradient continues in the same direction, it allows GD to travel too far along that dimension, decreasing the function value too rapidly.

However, while this heuristic is conceptually straightforward, rigorously identifying the checkpoint set that maximizes the score is a highly nontrivial combinatorial problem. Our previous work [YL26a] gives a finer analysis of the factors coupling consecutive checkpoints in the same construction, improving the lower bound from $\Omega(n^{-1.9319})$ to $\Omega(n^{-1.6342})$.

Transfer via Local Hard Functions [JCY26]. [JCY26] further improved the lower bound of [MC26] to $\Omega(n^{-\log_2(1+\sqrt{3})})$, where $\log_2(1+\sqrt{3}) \approx 1.4500$. Note that the hard function in [MC26] is constructed globally, which introduces complex dependencies between neighboring gaps into the resulting score. To simplify these dependencies, [JCY26] construct a chain of local hard functions that couple only adjacent coordinates, much like Nesterov's quadratic zero-chain above. Our construction uses the same chain structure, so we first describe the general framework and notation here. In the smooth convex setting, the objective is the sum of these two-dimensional functions and a final one-dimensional component:

$$F(x) := \frac{1}{2}\sum_{i=1}^{k} \Phi_i(x^{(i)}, x^{(i+1)}) + \frac{1}{2}\Phi_{k+1}(x^{(k+1)}), \qquad x_0 = e_1, \qquad x_* = 0. \tag{4}$$

Here $k = |T| \le n$ is the number of checkpoints. The initial point e_1 is the first coordinate vector. We call each Φ_i a *local hard function*, or simply a *component* of F. Each should be nonnegative, convex, and 1-smooth. For $1 \le i \le k$, $\Phi_i : \mathbb{R}^2 \to \mathbb{R}$ acts on the adjacent coordinates $(X, Y) = (x^{(i)}, x^{(i+1)})$, while $\Phi_{k+1} : \mathbb{R} \to \mathbb{R}$ acts only on $x^{(k+1)}$. We call X and Y the *input* and *output* coordinates respectively, and write $\partial_X \Phi_i$ and $\partial_Y \Phi_i$ for its derivatives with respect to X and Y. The key idea is to make these components active successively, where a component is *active* when its gradient is nonzero. During gap i, Φ_i is active and all later components have zero gradient. At the end of the gap, checkpoint t_i activates Φ_{i+1}.

To activate the components in this order, we choose an *activation threshold* $\ell_i \ge 0$ for Φ_i so that

$$\Phi_i(X, Y) = 0, \qquad \nabla\Phi_i(X, Y) = 0 \quad \text{whenever } X \le \ell_i,\ Y \ge 0.$$

Thus, as long as $Y \ge 0$, the component cannot be active until its input X exceeds ℓ_i. The final component satisfies $\Phi_{k+1}(Y) = \Phi'_{k+1}(Y) = 0$ whenever $Y \le \ell_{k+1}$. At the beginning of gap i, write

$$(x^{(i)}_{t_{i-1}}, x^{(i+1)}_{t_{i-1}}) = (\ell_i + D_i, 0),$$

where $D_i > 0$ is the input's *excess* above the threshold ℓ_i at the start of the gap. Initially, $\ell_1 = 0$ and $D_1 = 1$. During gap i, the construction keeps $x^{(i+1)} \le \ell_{i+1}$ and all further coordinates at zero. At checkpoint t_i, coordinate $i+1$ rises to $\ell_{i+1} + D_{i+1}$, crossing the threshold. Given (ℓ_i, D_i) and the prescribed gap, we construct Φ_i and choose ℓ_{i+1}. The checkpoint update determines D_{i+1}, and the pair (ℓ_{i+1}, D_{i+1}) becomes the starting data for the next component. In our smooth convex construction [YL26b], we also require $\partial_{x^{(i+1)}} \Phi_i = 0$ whenever $x^{(i+1)} \ge \ell_{i+1}$, independently of $x^{(i)}$. The shared coordinate $x^{(i+1)}$ stays above ℓ_{i+1} during the next gap, so Φ_i no longer affects its update. Figure 2 illustrates this construction.

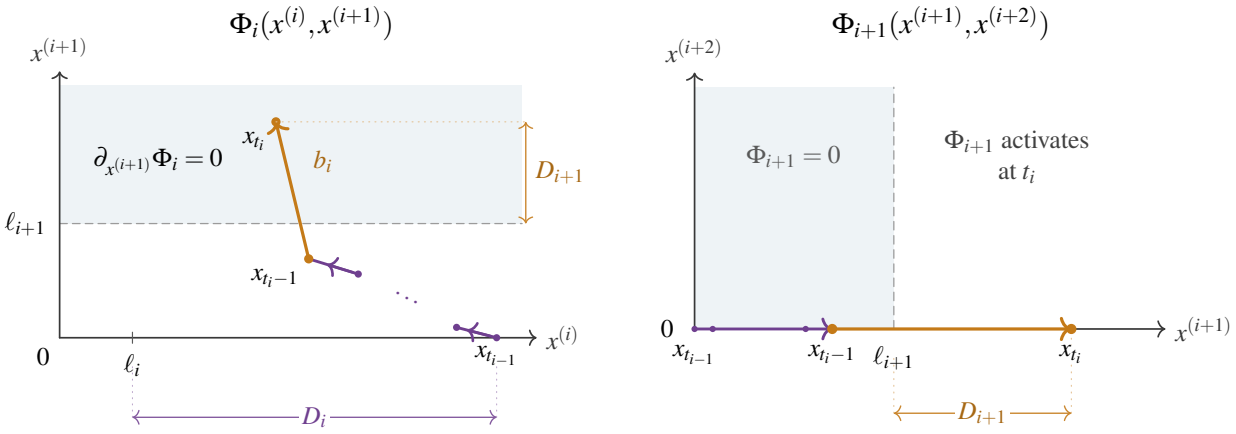


Figure 2: Activation of Φ_{i+1} at checkpoint t_i. Gap steps are purple and the checkpoint step is orange.

For a fixed checkpoint set, a larger final excess D_{k+1} gives a larger lower bound on function suboptimality. We therefore seek a large *transfer ratio* D_{i+1}/D_i when constructing each local hard function.

In [JCY26], the two-dimensional local hard functions Φ_i are constructed as scaled one-sided Huber functions:

$$\Phi_i(X, Y) = \frac{1}{2} H_{2D_i/(2+s_i)}(X - Y - \ell_i).$$

Here H_δ is the one-sided Huber function defined in Section 3.2. With the threshold $\ell_{i+1} = s_i D_i/[2(2+s_i)]$, the transfer ratio $D_{i+1}/D_i = b_i/[2(2+s_i)]$ depends only on b_i and s_i. A suitable choice of Φ_{k+1} gives

$$\frac{F(x_n) - F(x_*)}{\|x_0 - x_*\|^2} \ge \frac{1}{4(1+s_{k+1})} \prod_{i=1}^{k} \left(\frac{D_{i+1}}{D_i} \right)^2 = \frac{1}{4(1+s_{k+1})} \prod_{i=1}^{k} \left[\frac{b_i}{2(2+s_i)} \right]^2. \tag{5}$$

The expression on the right is the score for this construction. By maximizing this score over all checkpoint sets, [JCY26, Section 3] obtain the lower bound $\Omega(n^{-\log_2(1+\sqrt{3})})$.[9]

Bending the Local Trajectory [YL26b]. The local hard functions in [JCY26] have constant gradients during their respective gaps. These gradients decrease X and increase Y by equal amounts, accounting for the coefficient 2 of s_i in the denominator $2(2+s_i)$ in (5) [YL26b, Section 2]. To reduce this loss, our previous work introduces a construction that bends the GD trajectory within each gap [YL26b, Section 3.1]. For a checkpoint set with $b_i > 2$, the construction gives the score

$$\frac{1}{4(1+s_{k+1})} \prod_{i=1}^{k} \left[\frac{b_i - 2}{c(a+s_i)} \right]^2.$$

[9] In general, the argument in [JCY26, Section 3 and Appendix C] suggests a lower bound of order $n^{-\log_2(1+\sqrt{1+c})}$ when the coefficient of s_i in the denominator of the transfer ratio D_{i+1}/D_i is c. The factor $b_i/[2(2+s_i)]$ in (5) corresponds to $c = 2$.

Here a and $c > 1$ are constants determined by the construction, where c can be made arbitrarily close to 1. A recursive analysis of this score then yields the lower bound $\Omega(n^{-p_{\mathrm{sil}}-o(1)})$ [YL26b, Section 3.2].

2.2 Our Construction: Handling Quadratic Contraction

Using the notation above, we replace the final component in (4) by a quadratic term and scale the remaining sum by $1-1/\kappa$:

$$F(x) := \frac{1}{2\kappa}\|x\|^2 + \frac{1-1/\kappa}{2}\sum_{i=1}^{k}\Phi_i(x^{(i)},x^{(i+1)}), \qquad x_0 = e_1, \quad x_* = 0. \tag{6}$$

The resulting F is 1-smooth and $1/\kappa$-strongly convex. As in the smooth convex setting, we seek a large final excess $D_{k+1} = \prod_{i=1}^{k}(D_{i+1}/D_i)$ and use it to bound the final error.

Quadratic Contraction. However, preserving the excess is harder in the strongly convex setting, since the added quadratic term can reduce it during a gap. For $s_i < \kappa$, the quadratic term alone scales the input coordinate $x^{(i)}$ by χ_i, while the threshold ℓ_i stays fixed. The excess therefore changes from D_i to $D_i\eta_i$, where

$$\chi_i := \prod_{t_{i-1}<t<t_i}\left(1-\frac{h_t}{\kappa}\right), \qquad \chi_i(\ell_i+D_i)-\ell_i = D_i\underbrace{\left[\chi_i-(1-\chi_i)\frac{\ell_i}{D_i}\right]}_{:=\eta_i}. \tag{7}$$

Since $\chi_i \geq 1-s_i/\kappa$, the fraction of the excess lost to contraction satisfies $1-\eta_i \leq (s_i/\kappa)(1+\ell_i/D_i)$. We therefore need to keep s_i small relative to κ and control the *relative threshold* ℓ_i/D_i.

Choosing Checkpoints within Blocks. As in our smooth convex construction, we use *bending components*, adapted to the strongly convex setting (constructed in Section 3.1). The adapted construction depends on a bending parameter β.[10] For each bending component, Lemma 3.1 gives

$$\frac{D_{i+1}}{D_i} \geq \frac{b_i-8}{c_\beta(a_\beta+s_i+8)}\,\eta_i\left(1-\frac{s_i}{\kappa}\right).$$

The constants $a_\beta \geq 8$ and $1 < c_\beta < 2$ depend only on β, with $c_\beta = 1+O(\beta)$ and $\log a_\beta = O(1/\beta)$. Compared with $(b_i-2)/(c(a+s_i))$ in the smooth convex score, the first factor has the same form, with c_β in place of c and different additive constants. Taking β smaller brings c_β closer to one, which improves the exponent of κ in the iteration lower bound, but increases a_β. We choose β at the end of the proof to balance these effects.

The factor $\eta_i(1-s_i/\kappa)$ in this bound accounts for quadratic contraction. Notice that η_i depends on the relative threshold ℓ_i/D_i passed from the preceding component Φ_{i-1}. Keeping η_i in the score would make checkpoint selection depend on the local functions constructed in earlier gaps. To simplify checkpoint selection, we hence omit the whole factor $\eta_i(1-s_i/\kappa)$ and use the score

$$\prod_{i=1}^{k}\frac{b_i-8}{c_\beta(a_\beta+s_i+8)}, \qquad b_i > 8.$$

However, maximizing this score over the whole schedule can leave gaps with very large sums of stepsizes s_i. The resulting quadratic contraction can destroy the lower bound. We therefore divide the schedule into blocks of at most m steps and maximize the score within each block, as formalized in Section 4.1.

[10] Our smooth convex construction [YL26b, Section 3.1] prescribes the gradients of each local hard function along its gap trajectory by scaling vectors on a circular arc centered at $(0,\gamma)$ for a parameter $\gamma > 0$. We adapt this construction to the strongly convex setting with the center (β,β) (see Section A.1 for details).

As in our smooth convex analysis [YL26b, Section 3.2], maximality of the chosen checkpoint set imposes constraints on the checkpoint stepsizes b_i and the sums of gap stepsizes s_i. Specifically, the bounds on the gap sums s_i depend on $a_\beta m^{p_{\mathrm{sil}}+O(\beta)}$ and the sum of unselected stepsizes carried over from earlier blocks (see the first inequality in Lemma 4.1 (ii)). Recall that we need s_i small relative to κ to limit the loss of excess from quadratic contraction. Therefore, m should be chosen such that $a_\beta m^{p_{\mathrm{sil}}+O(\beta)}$ is a sufficiently small fraction of κ. We then use these constraints to account for the omitted contraction factors after checkpoint selection (see Lemma 5.1).

Repairing Accumulated Unselected Steps. If the chosen maximizing checkpoint set is empty, we select no checkpoint in that block. When this happens in several consecutive blocks, their stepsizes all contribute to the same s_i. Quadratic contraction can then exhaust the excess D_i before the next checkpoint. We therefore fix a *repair limit*, a small constant multiple of κ. At the end of such a block, once the sum of stepsizes since the latest checkpoint reaches this limit, we select one or two additional checkpoints, called *repair checkpoints*. At these checkpoints, we use *Huber repair components*, built in Section 3.2 from the one-sided Huber functions of [JCY26]. To control the quadratic contraction discussed above, we choose the repair checkpoints to bound the relative threshold after the repair and keep the remaining stepsize sum below the repair limit. Section 4.2 gives the detailed selection rule, with basic estimates provided by Lemma 4.2. The product of transfer ratios over all repairs is at least $(m+1)^{-O(1+n/m)}$, which is eventually proved in Section 5.2.

Bridging Back to Bending. When constructing the global hard function, we sometimes need to return to bending components after a repair. However, a bending component Φ_{i+1} requires that the preceding component Φ_i contribute nothing to the update of $x^{(i+1)}$. This means $\partial_{x^{(i+1)}}\Phi_i = 0$ during gap $i+1$ and just before its checkpoint update. A Huber repair component Φ_i guarantees only $\partial_{x^{(i+1)}}\Phi_i \le 0$, so its contribution to this update may be positive. We therefore use a *bridging component*, or bridge, for Φ_{i+1}, constructed in Section 3.3. The bridge works under the weaker condition $\partial_{x^{(i+1)}}\Phi_i \le 0$. It also satisfies $\partial_{x^{(i+2)}}\Phi_{i+1} = 0$ whenever $x^{(i+2)} \ge \ell_{i+2}$, so a bending component Φ_{i+2} can follow it. Since a repair may precede any nonempty maximizing set, we use a bridge at its first checkpoint and bending components at the remaining checkpoints. Lemma 5.1 controls the loss from quadratic contraction and shows that the combined transfer ratio of the bridge and bending components is at least $1/8$ of the score. Since the empty checkpoint set has score 1, every nonempty maximizing set therefore retains at least $1/8$ of the incoming excess.

From the Construction to the Main Lower Bound. Combining the $1/8$ factors from the nonempty maximizing sets with the repair bound over at most $1+n/m$ blocks gives

$$D_{k+1} \ge \exp[-O((1+n/m)\log(m+1))]\,.$$

The repair rule ensures that the final coordinate $x_n^{(k+1)}$ remains at least a constant multiple of D_{k+1}. Proposition 5.3 and Lemma 5.4 establish the following lower bounds on squared distance to the minimizer and function suboptimality:

$$\min\{D_{n,\kappa}(H), E_{n,\kappa}(H)\} \ge \exp\left[-O\left(\frac{n\log(m+1)}{m}\right)\right].$$

To strengthen the bound, we choose m as large as possible while keeping $a_\beta m^{p_{\mathrm{sil}}+O(\beta)}$ a sufficiently small fraction of κ, as required in the earlier paragraph *Choosing Checkpoints within Blocks*. We therefore take m of order $(\kappa/a_\beta)^{1/p_{\mathrm{sil}}-O(\beta)}$; see (24) for the precise choice. Using $\log a_\beta = O(1/\beta)$, we obtain

$$\log m \ge \frac{\log\kappa}{p_{\mathrm{sil}}} - O\left(\beta\log\kappa + \frac{1}{\beta}\right).$$

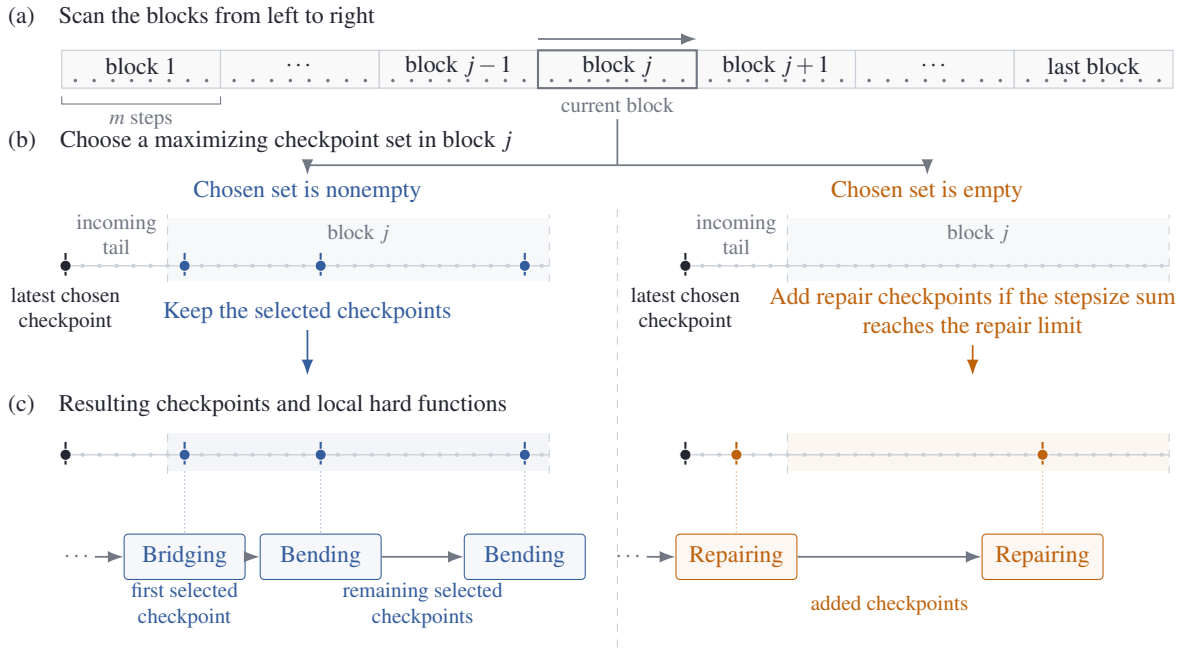


Figure 3: Selecting checkpoints and local hard functions. If the maximizing checkpoint set of a block is nonempty, we use a bridge at its first checkpoint and bending components at the remaining checkpoints. Otherwise, at the end of the block, we add one or two repair checkpoints with Huber repair components when the sum of stepsizes since the latest checkpoint reaches the repair limit. Repair checkpoints may lie in the incoming tail between the latest chosen checkpoint and the current block.

Finally, we balance the two terms by choosing $\beta = (\log \kappa)^{-1/2}$ for sufficiently large κ. This gives $m \geq \kappa^{1/p_{\mathrm{sil}}} e^{-O(\sqrt{\log \kappa})}$ and proves Theorem 1.1.

Figure 3 summarizes how we select checkpoints and local hard functions.

Organization. The rest of this paper is organized as follows. Section 3 constructs the three types of local hard functions and bounds their transfer ratios D_{i+1}/D_i and relative thresholds ℓ_{i+1}/D_{i+1}. Section 4 provides the checkpoint selection rule and estimates for the selected checkpoints. Section 5 constructs a hard function for each prescribed schedule and proves the main theorem.

3 Local Hard Functions: Bending, Repairing, and Bridging

In this section, we present three types of local hard functions $\Phi_i(x^{(i)}, x^{(i+1)})$ on adjacent coordinates used in (6). We first recall the local transfer setup from Section 2.1 and fix the notation used below.

Notation and setup. For the current local hard function Φ_i, write $(X,Y) = (x^{(i)}, x^{(i+1)})$. Fix an input threshold $\ell_i \geq 0$, an incoming excess $D_i > 0$ above ℓ_i, and gap stepsizes $0 \leq h_j < \kappa$ for $1 \leq j \leq m_{\mathrm{gap}}$. Here the stepsize index j starts at one in the current gap. Starting from $(X_0, Y_0) = (\ell_i + D_i, 0)$, let (X_j, Y_j) be the state after j updates. The gap is followed by a checkpoint update with stepsize $h_{m_{\mathrm{gap}}+1} = b_i > 0$, which evaluates the gradient at $(X_{m_{\mathrm{gap}}}, Y_{m_{\mathrm{gap}}})$, the state just before this update. We construct the current component

Φ_i and choose an outgoing threshold $\ell_{i+1} \geq 0$ so that

$$\begin{aligned} X_j \geq \ell_i, \qquad 0 \leq Y_j \leq \ell_{i+1} \quad (0 \leq j \leq m_{\text{gap}}), \\ Y_{m_{\text{gap}}+1} = \ell_{i+1} + D_{i+1}, \qquad D_{i+1} > 0. \end{aligned} \tag{8}$$

For $i < k$, the pair (ℓ_{i+1}, D_{i+1}) becomes the incoming data for the next local hard function Φ_{i+1}.

Local updates with the preceding component. With later components inactive, (6) gives the local updates, for $1 \leq j \leq m_{\text{gap}} + 1$:

$$\begin{pmatrix} X_j \\ Y_j \end{pmatrix} = \left(1 - \frac{h_j}{\kappa}\right) \begin{pmatrix} X_{j-1} \\ Y_{j-1} \end{pmatrix} - \frac{(1 - 1/\kappa) h_j}{2} \nabla \Phi_i(X_{j-1}, Y_{j-1}) + h_j \begin{pmatrix} f_j \\ 0 \end{pmatrix}. \tag{9}$$

For $i > 1$, the term $h_j f_j$ is the contribution of the preceding component Φ_{i-1} to the update of the shared coordinate $x^{(i)}$, where $f_j = -(1 - 1/\kappa)\partial_{x^{(i)}} \Phi_{i-1}/2$. Here $\partial_{x^{(i)}} \Phi_{i-1}$ differentiates $\Phi_{i-1}(x^{(i-1)}, x^{(i)})$ in the shared coordinate $x^{(i)}$ and is evaluated before step j. Each local hard function below satisfies $\partial_Y \Phi_i \leq 0$, so $f_j \geq 0$. For $i = 1$, there is no preceding component and $f_j = 0$.

To bound the loss from quadratic contraction, recall χ_i and η_i from (7). If $s_i < \kappa$, then

$$1 - \frac{s_i}{\kappa} \leq \chi_i \leq 1, \qquad \eta_i \geq 1 - \frac{s_i}{\kappa}(1 + \ell_i / D_i). \tag{10}$$

For an empty gap, $s_i = 0$ and $\chi_i = \eta_i = 1$.

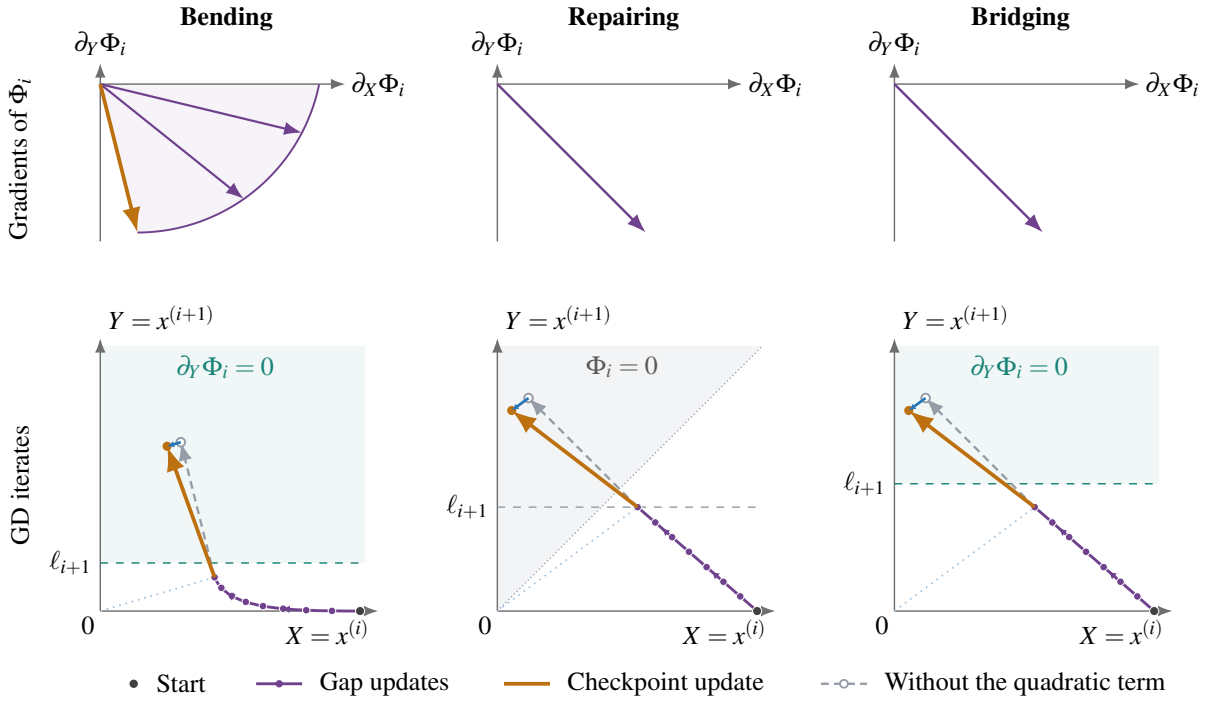


Figure 4: Schematic scaled gradients of Φ_i (top) and GD trajectories (X_j, Y_j) with the quadratic term (bottom). Gray dashed arrows show the checkpoint updates from the same starting points without this term. Blue arrows show the contraction toward the origin.

With these preparations in hand, we now give the constructions and transfer guarantees for the three types of local hard functions introduced in Section 2.2. For each Φ_i, we bound the transfer ratio D_{i+1}/D_i

from below to limit the loss of excess. A larger final excess D_{k+1} gives a stronger lower bound. We also bound the relative threshold ℓ_{i+1}/D_{i+1} from above to limit contraction loss in the next gap.

3.1 Bending

For our basic local hard function, we adapt the bending construction from our previous work [YL26b] to account for quadratic contraction. Specifically, we choose a parameter $0 < \beta \le 1/4$. Along the constructed gap trajectory, the scaled gradient turns toward the negative Y-direction along a circular arc centered at (β, β). The construction yields constants $a_\beta \ge 8$ and $1 < c_\beta < 2$, depending only on β, with

$$c_\beta = 1 + O(\beta), \qquad \log a_\beta = O(1/\beta), \tag{11}$$

uniformly when $0 < \beta \le 1/4$. Their explicit choices are given in (31) and (34) in Section A.1.

Lemma 3.1 (Bending)**.** *Let $\kappa \ge 4$, $0 < \beta \le 1/4$, $s_i/\kappa \le 1/16$, and $\eta_i > 0$. There exist a nonnegative convex 1-smooth local hard function Φ_i and a threshold $\ell_{i+1} \ge 0$, independent of the checkpoint stepsize b_i, such that $\partial_Y \Phi_i \le 0$ everywhere and*

$$\Phi_i(X,Y) = 0 \quad (X \le \ell_i, Y \ge 0), \qquad \partial_Y \Phi_i(X,Y) = 0 \quad (X \in \mathbb{R}, Y \ge \ell_{i+1}).$$

For every $b_i > 8$, the iterations (9) with $f_j = 0$ satisfy (8), and

$$\frac{D_{i+1}}{D_i} \ge \frac{b_i - 8}{c_\beta(a_\beta + s_i + 8)}\,\eta_i(1 - s_i/\kappa), \qquad \frac{\ell_{i+1}}{D_{i+1}} \le 4\frac{s_i + 2}{b_i - 8}. \tag{12}$$

For these explicit choices, taking β smaller brings c_β closer to one but increases a_β. The construction and proof are given in Section A.1. As in our smooth convex construction [YL26b], the bending component Φ_i satisfies $\partial_Y \Phi_i(X,Y) = 0$ for every X whenever $Y \ge \ell_{i+1}$. For $i < k$, $Y = x^{(i+1)}$ is the input coordinate of the next component Φ_{i+1}. Thus, Φ_i does not affect the update of this coordinate in this region.

3.2 Repairing

As discussed in Section 2.2, we select checkpoint sets by maximizing a product based on the bending transfer bound. This selection can leave many consecutive steps unselected, so the sum of gap stepsizes s_i can become large. If s_i becomes large relative to κ, quadratic contraction can substantially reduce the incoming excess D_i. We therefore split such gaps with suitably chosen additional checkpoints and use *Huber repair components* to bound the relative threshold ℓ_{i+1}/D_{i+1} after the last added checkpoint t_i. The construction uses the following one-sided Huber function of [JCY26]:

$$H_\delta(t) = \begin{cases} 0, & t \le 0, \\ t^2/2, & 0 \le t \le \delta, \\ \delta t - \delta^2/2, & t \ge \delta. \end{cases}$$

We choose the Huber scale and local hard function as follows:

$$\delta_{\mathrm{H}} = \frac{D_i \eta_i}{2 + (\kappa - 1)(1 - \chi_i)}, \qquad \Phi_i(X,Y) = \frac{1}{2} H_{2\delta_{\mathrm{H}}}(X - Y - \ell_i). \tag{13}$$

Set $\ell_{i+1} = (\kappa - 1)(1 - \chi_i)\delta_{\mathrm{H}}/2$.

Lemma 3.2 (Repairing)**.** *Let $\kappa > 1$ and $\eta_i > 0$. The function Φ_i defined in (13) is nonnegative, convex, and 1-smooth, with $\partial_Y \Phi_i \le 0$ everywhere and $\Phi_i(X,Y) = 0$ whenever $X \le \ell_i$, $Y \ge 0$.*

For every $b_i > 0$ and every sequence $f_j \ge 0$, the iterations (9) satisfy (8), and

$$\frac{D_{i+1}}{D_i} = \frac{(1-1/\kappa)b_i\chi_i\eta_i}{2[2+(\kappa-1)(1-\chi_i)]}, \qquad \frac{\ell_{i+1}}{D_{i+1}} = \frac{\kappa(1-\chi_i)}{b_i\chi_i}. \tag{14}$$

For this Huber repair component, ℓ_{i+1} is the value of Y just before the checkpoint update, while D_{i+1} is the net increase in Y produced by that update. If $s_i < \kappa$, the second identity in (14) gives

$$\frac{s_i}{b_i} \le \frac{\ell_{i+1}}{D_{i+1}} \le \frac{s_i/b_i}{1-s_i/\kappa}. \tag{15}$$

When s_i is small relative to κ, (15) shows that a checkpoint with b_i comparable to s_i gives a bounded relative threshold ℓ_{i+1}/D_{i+1}. The proof of the local guarantees is given in Section A.2. One issue of the Huber repair component is that $\partial_Y \Phi_i(X,Y) = 0$ only when $Y \ge X - \ell_i$. Thus, even after the input coordinate $x^{(i+1)}$ of the next component Φ_{i+1} crosses ℓ_{i+1}, Φ_i may still affect its update. The construction in the next section resolves this.

3.3 Bridging

After a Huber repair, the preceding component Φ_{i-1} may still contribute $h_j f_j \ge 0$ to the update of $X = x^{(i)}$, whereas Lemma 3.1 requires $f_j = 0$. We therefore insert a *bridging component* Φ_i between the repair component Φ_{i-1} and the next bending component Φ_{i+1}. We construct Φ_i so that its transfer guarantees hold for every sequence $f_j \ge 0$ and $\partial_Y \Phi_i(X,Y) = 0$ for every X whenever $Y \ge \ell_{i+1}$. The latter property ensures that Φ_i does not affect the update of $x^{(i+1)}$ during the next bending gap or its checkpoint update.

Using δ_{H} from (13), we define the bridging component by

$$\ell_{i+1} = \frac{D_i\eta_i}{2}, \qquad K_\triangle = \mathrm{conv}\{0, (\delta_{\mathrm{H}}, -\delta_{\mathrm{H}}), (\delta_{\mathrm{H}}, 0)\},$$

and

$$\Phi_i(X,Y) = \max_{g \in K_\triangle} \left\{ \langle g, (X - \ell_i - \ell_{i+1}, Y - \ell_{i+1}) \rangle - \frac{1}{2}\|g\|^2 \right\}.$$

By the projection identity in Fact A.1, the horizontal edge of $K_\triangle$ ensures that $\partial_Y \Phi_i(X,Y) = 0$ whenever $Y \ge \ell_{i+1}$.

Lemma 3.3 (Bridging)**.** *Let $\kappa \ge 4$, $s_i/\kappa \le 1/16$, and $\eta_i > 0$. The function Φ_i defined above is nonnegative, convex, and 1-smooth, with $\partial_Y \Phi_i \le 0$ everywhere and $\Phi_i(X,Y) = 0$ whenever $X \le \ell_i$, $Y \ge 0$. Moreover, $\partial_Y \Phi_i(X,Y) = 0$ for all $X \in \mathbb{R}$, $Y \ge \ell_{i+1}$.*

For every $b_i > 4$ and every sequence $f_j \ge 0$, the iterations (9) satisfy (8), and

$$\frac{D_{i+1}}{D_i} \ge \frac{1}{4}\eta_i \frac{b_i - 4}{s_i + 2}, \qquad \frac{\ell_{i+1}}{D_{i+1}} \le 2\frac{s_i + 2}{b_i - 4}.$$

With the same ℓ_i, D_i, stepsizes, and sequence $f_j \ge 0$, the bridge and the Huber repair component of Section 3.2 follow the same trajectory through the checkpoint update. The proof is provided in Section A.3. Figure 4 provides an illustration of the trajectories of these local constructions.

4 Selecting Checkpoints for Every Schedule

In this section, we describe our checkpoint selection rule. As explained in Section 2.2, selecting checkpoints solely by maximizing the bending score, as in the smooth convex setting, can leave gaps in which quadratic contraction exhausts the excess D_i. We therefore divide the schedule into blocks of m consecutive steps and maximize this score within each block sequentially. If the sum of unselected stepsizes since the latest checkpoint becomes too large, we add repair checkpoints.

The rest of this section is organized as follows. Section 4.1 defines the score within each block, states the rule for choosing checkpoints by maximizing this score, and gives estimates for the selected checkpoints and gaps. Section 4.2 states the rule for adding repair checkpoints and gives its guarantees.

4.1 Choosing Checkpoints within a Block

Fix a positive integer m. Divide the schedule $H = (h_1, \dots, h_n)$ into consecutive blocks of m steps, with possibly fewer steps in the last block.[12] We examine the blocks sequentially. At the start of a block, the *incoming tail* consists of the preceding steps after the latest chosen checkpoint, or all preceding steps if none has been chosen. Write $w \geq 0$ for the sum of stepsizes in the incoming tail; $w = 0$ at the start of the first block. For a single block of $1 \leq r \leq m$ steps, relabel its nonnegative stepsizes as $H = (h_1, \dots, h_r)$.

Fix $a \geq 8$ and $1 \leq c \leq 2$. Define

$$p = p(c) := \log_2(1 + \sqrt{1+c}). \tag{16}$$

An *admissible checkpoint set* $T = \{t_1 < \cdots < t_k\} \subseteq \{1, \dots, r\}$ selects only steps with $b_i := h_{t_i} > 8$. For a nonempty checkpoint set, put

$$s_1 := w + \sum_{j=1}^{t_1-1} h_j, \qquad s_i := \sum_{j=t_{i-1}+1}^{t_i-1} h_j \quad (2 \leq i \leq k), \qquad s_{\text{tail}} := \sum_{j=t_k+1}^{r} h_j. \tag{17}$$

In the hard function construction, we take $a = a_\beta$ and $c = c_\beta$. The lower bound on D_{i+1}/D_i in Lemma 3.1 is $(b_i - 8)/(c(a + s_i + 8))$ times the contraction factor $\eta_i(1 - s_i/\kappa)$. As explained in Section 2.2, we omit the contraction factor when choosing checkpoints and account for it in Section 5.1. Following our smooth convex analysis [YL26b, Section 3.2], we define the score as the product of the remaining factors:

$$P_w(T; H) := \prod_{i=1}^{k} \frac{b_i - 8}{c(a + s_i + 8)}, \qquad P_w(\varnothing; H) := 1. \tag{18}$$

For fixed H and w, we maximize this score over the admissible checkpoint sets T. If a nonempty set attains the maximum, we choose one. Otherwise we choose $T = \varnothing$, selecting no checkpoint in this block. There are finitely many sets, so a maximizer exists and its score is at least one.

The next lemma provides properties of a maximizing set that we use to bound the product of transfer ratios in Lemma 5.1.

Lemma 4.1 (Bounds for a maximizing checkpoint set)**.** *In the setting above, there is an absolute constant C_b with the following properties. Parts (ii)–(iv) hold for every nonempty maximizing set $T = \{t_1 < \cdots < t_k\}$.*

(i) If the empty set maximizes (18), then $w + \sum_{j=1}^{r} h_j \leq C_b(w + am^p)$.

(ii) $c\sum_{i=1}^{k}(a + s_i + 8) \leq C_b(w + am^p)$ and $s_{\text{tail}} \leq C_b am^p$.

[12] Blocks are fixed before checkpoint selection. A gap is determined by successive checkpoints and may cross block boundaries.

(iii) For every $1 \le i < k$,

$$\frac{1}{b_i - 8} \le \frac{2}{a + s_i + 8} + \frac{2}{a + s_{i+1} + 8}.$$

(iv) The last checkpoint satisfies $b_k - 8 \ge c(a + s_k + 8)$.

The proofs of parts (i)–(ii) adapt the recursive analysis in [YL26b, Section 3.2]. The full proof is provided in Appendix B.

4.2 Repairing Accumulated Unselected Steps

If the chosen checkpoint set in a block is empty, its steps extend the *unselected tail*.[13] Over consecutive blocks, the sum of these stepsizes can grow large enough for quadratic contraction to exhaust the excess D_i. We therefore fix a *repair limit* S_0, a small constant multiple of κ.[14] At the end of such a block, if the tail stepsizes sum to at least S_0, we add repair checkpoints in this tail and use the Huber repair components of Lemma 3.2 there. By (15), a bounded ratio s_i/b_i gives a bounded relative threshold ℓ_{i+1}/D_{i+1} when s_i is small relative to κ. This requires b_i to be at least a constant fraction of s_i. However, satisfying this condition alone does not restrict the sum of stepsizes remaining after the checkpoint in the current tail. We therefore design the following repair checkpoint selection rule.

Repair checkpoint selection rule. For a repair, let $h_1, \dots, h_r \ge 0$ denote the stepsizes in the unselected tail through the end of the current block, listed in their original order. Suppose $S := h_1 + \cdots + h_r \ge S_0 > 0$. We select $q \in \{1,2\}$ checkpoints from the shortest suffix whose stepsizes sum to more than $S_0/8$, and write $b_i := h_{t_i}$ and $t_0 = 0$.

- If this suffix contains a stepsize $h_t \ge S_0/16$, select one such index as the only checkpoint t_1.
- Otherwise, select $t_1 < t_2$ in the suffix with $b_1, b_2 \ge S_0/(64r^2)$ and $s_2 := \sum_{t=t_1+1}^{t_2-1} h_t \le \min\{b_1, b_2\}$. Such a pair exists by Lemma C.1.

The following lemma gives bounds for repair checkpoints selected by this rule.

Lemma 4.2 (Bounds for repair checkpoints)**.** *Fix* $C_0 \ge 1$. *There are constants* $\xi_0 \in (0,1)$ *and* $c_0 > 0$, *depending only on* C_0, *with the following property. Let* $\kappa \ge 4$, $1 \le S_0 \le \xi_0 \kappa / C_0$, *and* $S_0 \le S \le C_0 S_0$. *For any checkpoints selected by the rule above and every incoming pair* $\ell_1 \ge 0$, $D_1 > 0$ *with* $\ell_1/D_1 \le 32C_0$, *the Huber repair components of Lemma 3.2 can be constructed successively at these checkpoints. They satisfy* $\eta_i > 0$ *and* $D_{i+1} > 0$ *for every* $1 \le i \le q$, *with the following bounds.*

(i) ***Transfer ratio lower bound.***

$$\frac{D_{q+1}}{D_1} \ge c_0 (r+1)^{-4}.$$

(ii) ***Relative threshold upper bound.*** $\ell_{q+1}/D_{q+1} \le 32C_0$.

(iii) ***Remaining tail bound.*** $\displaystyle\sum_{t=t_q+1}^{r} h_t \le S_0/4$.

The proof is given in Appendix C.

[13] The unselected tail consists of the examined steps after the latest chosen checkpoint, or all examined steps if none has been chosen. At the start of a block, it is the incoming tail defined in Section 4.1.

[14] The value of S_0 is chosen in Proposition 5.3, where we construct the global hard function.

5 Assembling the Hard Function and Proving Theorem 1.1

In this section, we assemble the local hard functions into the global objective (6) and prove Theorem 1.1.

5.1 Constructing Components at a Maximizing Checkpoint Set

The score $P_w(T;H)$ used for checkpoint selection in Section 4.1 omits quadratic contraction. The following Lemma 5.1 shows that we can construct components whose combined transfer ratio is at least one eighth of this score, even with quadratic contraction.

We first specify the notation for this construction. Fix $\kappa \geq 4$ and $0 < \beta \leq 1/4$, and set $a = a_\beta$, $c = c_\beta$, and $p = p(c)$. Consider a block $H = (h_1, \ldots, h_r)$ with an incoming tail whose stepsizes sum to w. Let $T = \{t_1 < \cdots < t_k\}$ be a nonempty admissible maximizer of $P_w(T;H)$.[15] Write $b_i = h_{t_i}$ and take the sums of gap stepsizes s_i from (17). The first gap includes the incoming tail. Starting from the given threshold and excess (ℓ_1, D_1), we construct the components in checkpoint order.

Notice that the first component may follow a Huber repair, so its input update in (9) may include a nonzero term $h_j f_j$, where $f_j \geq 0$. As discussed in Section 3.3, we therefore use a bridge for $i = 1$, followed by bending components for $2 \leq i \leq k$, which require $f_j = 0$.

Lemma 5.1. *For the block and checkpoint set above, fix $R \geq 0$, $\ell_1 \geq 0$, and $D_1 > 0$ with $\ell_1/D_1 \leq R$. Suppose that*

$$\frac{c}{\kappa} \sum_{i=1}^{k} (a + s_i + 8) \leq \frac{1}{128(R+1)}. \tag{19}$$

Then the bridge construction in Lemma 3.3 can be applied at t_1, followed by the bending construction in Lemma 3.1 at each t_i, $2 \leq i \leq k$. Their local trajectories satisfy (8), and the resulting pairs satisfy

$$\frac{D_{k+1}}{D_1} \geq \frac{1}{8} P_w(T;H), \qquad \frac{\ell_{k+1}}{D_{k+1}} \leq 4.$$

Proof. Since $s_i \leq c(a + s_i + 8)$, (19) gives $s_i/\kappa \leq 1/128 < 1/16$ for every $1 \leq i \leq k$. Whenever $\eta_i > 0$, Lemma 3.3 for $i = 1$ and Lemma 3.1 for $i \geq 2$ construct the ith component and give

$$\frac{D_2}{D_1} \geq \frac{1}{4} \frac{b_1 - 8}{c(a + s_1 + 8)} \eta_1 \qquad \text{(bridge at } t_1\text{)}, \tag{20}$$

$$\frac{D_{i+1}}{D_i} \geq \frac{b_i - 8}{c(a + s_i + 8)} \eta_i (1 - s_i/\kappa) \qquad (2 \leq i \leq k), \tag{21}$$

$$\frac{\ell_{i+1}}{D_{i+1}} \leq \frac{4(s_i + 2)}{b_i - 8} \qquad (1 \leq i \leq k). \tag{22}$$

We verify $\eta_i > 0$ in checkpoint order using (10). For the first gap,

$$1 - \eta_1 \leq \frac{s_1}{\kappa} \left(1 + \frac{\ell_1}{D_1}\right) \leq \frac{(R+1)c(a + s_1 + 8)}{\kappa} \leq \frac{1}{128},$$

so $\eta_1 > 0$ and the bridge can be constructed. Now fix $2 \leq i \leq k$ and suppose that $\Phi_1, \ldots, \Phi_{i-1}$ have been constructed. Then (22) at $i - 1$ gives $\ell_i/D_i \leq 4(s_{i-1} + 2)/(b_{i-1} - 8)$. Using Lemma 4.1 (iii) and $c \leq 2$, we

[15] Admissibility means $h_{t_i} > 8$ at every selected checkpoint, as defined in Section 4.1.

obtain

$$\begin{aligned}1-\eta_i &\le \frac{s_i}{\kappa}\left(1+\frac{\ell_i}{D_i}\right)\\ &\le \frac{c(a+s_i+8)}{\kappa}+\frac{4c^2(a+s_{i-1}+8)(a+s_i+8)}{\kappa(b_{i-1}-8)}\\ &\le \frac{c}{\kappa}\big[17(a+s_i+8)+16(a+s_{i-1}+8)\big] \le \frac{17}{128(R+1)}<1.\end{aligned}$$

Thus $\eta_i > 0$, so the next bending component can be constructed.

Summing the preceding bounds on $1-\eta_i$ before applying (19), and using $s_i \le c(a+s_i+8)$, gives

$$\sum_{i=1}^{k}(1-\eta_i)+\sum_{i=2}^{k}\frac{s_i}{\kappa} \le \frac{35(R+1)c}{\kappa}\sum_{i=1}^{k}(a+s_i+8) \le \frac{35}{128}<\frac{1}{2}.$$

All factors η_i and $1-s_i/\kappa$ lie in $(0,1]$. Using $\prod_j(1-u_j) \ge 1-\sum_j u_j$ for $0 \le u_j \le 1$, and multiplying (20)–(21), we obtain

$$\frac{D_{k+1}}{D_1} \ge \frac{1}{4}P_w(T;H)\prod_{i=1}^{k}\eta_i\prod_{i=2}^{k}(1-s_i/\kappa) \ge \frac{1}{8}P_w(T;H).$$

Finally, Lemma 4.1 (iv) and (22) at $i=k$ give

$$\frac{\ell_{k+1}}{D_{k+1}} \le \frac{4(s_k+2)}{b_k-8} \le \frac{4(s_k+2)}{c(a+s_k+8)} \le 4.$$

□

5.2 Constructing the Global Hard Function

We now construct the global hard function. Recall that Lemma 5.1 bounds the combined transfer ratio of the bridge and bending components at each nonempty maximizing checkpoint set, while Lemma 4.2 gives the corresponding bound for each repair. To combine these bounds, we first describe which stepsize indices each repair considers.

Fix integers $n,m \ge 1$, a nonnegative schedule $(h_1,\dots,h_n)$, and $S_0 > 0$. Partition the schedule into $B := \lceil n/m \rceil$ consecutive blocks of at most m steps. Consider a checkpoint-selection procedure that examines the blocks in order and performs at most one repair after each block. Number the repairs in the order made, $j = 1,\dots,J$, where $J \le B$.

Repair j considers the stepsize indices $u_j,\dots,v_j$, where u_j is one plus the latest checkpoint index before this repair ($u_j = 1$ if none exists), and v_j is the end of the current block. Let $p_j \in \{u_j,\dots,v_j\}$ be the last checkpoint selected by this repair, and let $n_j := v_j - u_j + 1$. Checkpoints are selected in increasing order, so $u_{j+1} \ge p_j + 1$ for $1 \le j < J$. These indices are illustrated in Figure 5. Suppose that, for each repair, the stepsizes under consideration sum to at least S_0, and those after its last checkpoint sum to at most $S_0/4$:

$$\sum_{t=u_j}^{v_j} h_t \ge S_0, \qquad \sum_{t=p_j+1}^{v_j} h_t \le S_0/4 \qquad (1 \le j \le J). \tag{23}$$

We then bound how often each stepsize index is considered across repairs.

Lemma 5.2 (Counting stepsize indices considered by repairs)**.** *Under the conditions in* (23)*, each stepsize index belongs to at most two of the sets* $\{u_j,\dots,v_j\}$*. Consequently,*

$$\sum_{j=1}^{J} n_j \le 2n, \qquad \sum_{j=1}^{J}\log(n_j+1) \le B\log(2m+1).$$

Proof. We first show $p_{j+1} > v_j$ for $1 \le j < J$. Otherwise, $u_{j+1} \ge p_j + 1$ and (23) give

$$S_0 \le \sum_{t=u_{j+1}}^{v_{j+1}} h_t \le \sum_{t=p_j+1}^{v_j} h_t + \sum_{t=p_{j+1}+1}^{v_{j+1}} h_t \le \frac{S_0}{2},$$

a contradiction. Hence $u_{j+2} \ge p_{j+1} + 1 > v_j$ for $1 \le j \le J-2$, so repairs whose numbers differ by at least two share no stepsize indices. This proves $\sum_{j=1}^{J} n_j \le 2n$.

Pad $n_1, \dots, n_J$ with zeros to obtain B entries. Concavity of the logarithm and $n \le Bm$ give

$$\sum_{j=1}^{J} \log(n_j + 1) \le B \log\left(1 + \frac{\sum_{j=1}^{J} n_j}{B}\right) \le B \log(2m+1).$$

□

Combining Lemma 5.1 with the repair bounds in Lemmas 4.2 and 5.2, we construct the global hard function and obtain the following lower bound on squared distance to the minimizer.

Proposition 5.3 (Global lower bound for fixed β)**.** *There exist absolute constants $c_*, C, \xi > 0$ with the following property. For every $0 < \beta \le 1/4$ and $\kappa \ge 4$, set*

$$S_0 := \xi\kappa, \qquad m := \left\lfloor c_*(\kappa/a_\beta)^{1/p(c_\beta)} \right\rfloor. \tag{24}$$

If $S_0 \ge 1$ and $m \ge 1$, then every horizon $n \ge 1$ and every nonnegative schedule H satisfy

$$D_{n,\kappa}(H) \ge \exp\left[-C\left(1 + \frac{n}{m}\right)\log(m+1)\right]. \tag{25}$$

Proof. **Step 1. Choosing the parameters.** Take the constant C_b in Lemma 4.1 to be at least 1, and set

$$C_0 := C_b + 2, \qquad R := 32C_0.$$

Let ξ_0 and c_0 be the constants of Lemma 4.2 for this C_0, and choose

$$\xi := \min\left\{\frac{\xi_0}{C_0}, \frac{1}{128C_0(R+1)}\right\}, \qquad c_* := \frac{\xi}{4C_b}.$$

Since $c_* \le 1$ and $p > 1$, the definition of m gives $am^p \le c_*^p \kappa \le c_* \kappa$, and hence

$$C_b a m^p \le S_0/4. \tag{26}$$

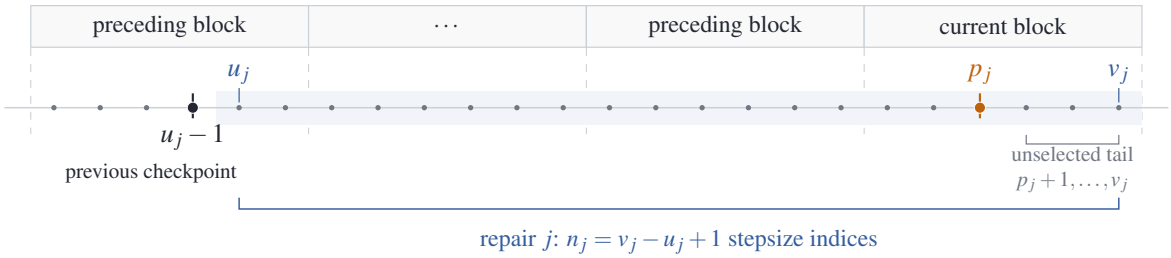


Figure 5: Repair j considers the $n_j = v_j - u_j + 1$ stepsize indices $u_j, \dots, v_j$, from just after the previous checkpoint $u_j - 1$ to the end of the current block. The last selected repair checkpoint p_j is highlighted, and the steps after it remain unselected. Although shown in the current block, p_j may lie in an earlier block.

Step 2. Selecting checkpoints and constructing components. Partition the schedule into $B := \lceil n/m \rceil$ fixed blocks, each containing at most m steps. We examine the blocks in order and construct the components in checkpoint order. Before each block, let w be the sum of the stepsizes in its incoming tail, and let $\ell \geq 0$ and $D > 0$ be the threshold and the excess above it immediately after the most recent checkpoint. We maintain

$$w < S_0, \qquad \frac{\ell}{D} \leq R. \tag{27}$$

Initially, $w = \ell = 0$ and $D = 1$. For the next block, maximize (18) over its checkpoint sets with this w, choosing a nonempty maximizing set whenever one exists.

Case 1: The maximizing checkpoint set is nonempty. By Lemma 4.1 (ii), (26), and (27),

$$c\sum_i (a + s_i + 8) \leq C_b(w + am^p) \leq (C_b + 1)S_0 \leq C_0 S_0 = C_0 \xi \kappa.$$

The choice of ξ ensures (19). Applied with $(\ell_1, D_1) = (\ell, D)$, Lemma 5.1 gives a bridge at the first selected checkpoint and bending components at the others. Their transfer ratios have product at least $1/8$, since the maximizing score is at least the empty score 1. At the last selected checkpoint, the new relative threshold ℓ/D is at most 4. By Lemma 4.1 (ii) and (26), the remaining tail stepsizes sum to $w \leq S_0/4$. Thus (27) is maintained.

Case 2: The maximizing checkpoint set is empty. All steps in this block remain unselected, so they extend the tail after the latest checkpoint. Part (i) of Lemma 4.1 implies that the sum of stepsizes in this enlarged tail is at most $C_b(w + am^p) \leq C_0 S_0$. If this sum is less than S_0, retain this tail and leave (ℓ, D) unchanged. Otherwise perform repair j, using the indices u_j, v_j and length n_j defined above. Then $S_0 \leq \sum_{t=u_j}^{v_j} h_t \leq C_0 S_0$, and $1 \leq S_0 \leq \xi_0 \kappa / C_0$ by the choice of ξ.

Select checkpoints from $(h_{u_j}, \ldots, h_{v_j})$ by the repair rule in Section 4.2. Apply Lemma 4.2 with $r = n_j$ and $(\ell_1, D_1) = (\ell, D)$, where $\ell/D \leq R$ by (27). Shift the selected checkpoint indices by $u_j - 1$ to obtain indices in the full schedule, and let p_j be the last of them. Use the Huber repair components of Lemma 3.2 at these checkpoints. By Lemma 4.2 (i)–(iii), their transfer ratios have product at least $c_0(n_j + 1)^{-4}$, the new relative threshold is at most R, and the new tail $h_{p_j+1}, \ldots, h_{v_j}$ sums to at most $S_0/4$. Thus (27) is maintained, and repair j satisfies (23).

By (14) and (38), the Huber repair and bridge outputs D_{i+1} do not depend on the nonnegative contributions f_j. Thus all incoming pairs and components can be determined in checkpoint order from the schedule before GD is run.

Step 3. Verifying the GD trajectory. Write all selected checkpoint indices in the full schedule as $t_1 < \cdots < t_k$, and set $t_0 = 0$. Define F on $\mathbb{R}^{k+1}$ by (6), with $x_0 = e_1$. Each Φ_i is nonnegative, convex, 1-smooth, and zero at $(0,0)$. Since each coordinate appears in at most two of the Φ_i, F is 1-smooth and $1/\kappa$-strongly convex, with minimizer $x_* = 0$.

We now verify the trajectory by induction over the checkpoints. The induction hypothesis is

$$x^{(i)}_{t_{i-1}} = \ell_i + D_i, \qquad x^{(j)}_{t_{i-1}} = 0 \quad (i+1 \leq j \leq k+1),$$

which holds for $i = 1$. Within gap i, we prove the bounds in (8) by induction over the updates: at every iterate x_t with $t_{i-1} \leq t < t_i$,

$$x^{(i)}_t \geq \ell_i, \qquad 0 \leq x^{(i+1)}_t \leq \ell_{i+1}, \qquad x^{(j)}_t = 0 \quad (i+2 \leq j \leq k+1).$$

The checkpoint hypothesis gives these bounds at $t = t_{i-1}$. Assuming they hold at x_t, all components after Φ_i have value zero and hence zero gradient. If Φ_i is a bending component, then Φ_{i-1} is a bridge or a bending component, since t_{i-1} lies in the same maximizing set. Since $x^{(i)}_t \geq \ell_i$, $\partial_{x^{(i)}} \Phi_{i-1}(x^{(i-1)}_t, x^{(i)}_t) = 0$, so the

update follows (9) with $f_j = 0$. If Φ_i is a Huber repair component or a bridge, then $\partial_Y \Phi_{i-1} \le 0$ gives $f_j \ge 0$ when $i > 1$, and $f_j = 0$ when $i = 1$. Lemmas 3.2 and 3.3 allow every such sequence. Components with index less than $i-1$ involve neither $x^{(i)}$ nor $x^{(i+1)}$. The local lemmas therefore maintain the displayed bounds until just before the checkpoint update and give $x_{t_i}^{(i+1)} = \ell_{i+1} + D_{i+1}$, with all later coordinates still zero. This proves the next induction hypothesis.

Step 4. Bounding the final coordinate. The procedure makes at most one repair per block, and Step 2 verified (23) for each repair. Each repair j has a product of transfer ratios at least $c_0(n_j+1)^{-4}$. By Lemma 5.2 and $J \le B$, their combined factor is at least

$$\prod_{j=1}^{J} c_0(n_j+1)^{-4} \ge \exp[-CB\log(m+1)],$$

where C depends only on c_0. Each block with a nonempty maximizing checkpoint set contributes a factor at least $1/8$ by Lemma 5.1. There are at most B such blocks, so, after increasing C, the excess D_{k+1} after the last checkpoint is at least $\exp[-CB\log(m+1)]$.

If $k \ge 1$, then $x_{t_k}^{(k+1)} = \ell_{k+1} + D_{k+1} \ge D_{k+1}$. Since $\partial_{x^{(k+1)}} \Phi_k \le 0$, every remaining update $t_k < t \le n$ satisfies $x_t^{(k+1)} \ge (1 - h_t/\kappa) x_{t-1}^{(k+1)}$. By (27), these remaining stepsizes sum to less than $S_0 = \xi\kappa$, so

$$x_n^{(k+1)} \ge (1-\xi) D_{k+1} \ge (1-\xi)\exp[-CB\log(m+1)].$$

Since $\|x_0 - x_*\|^2 = 1$ and $\|x_n - x_*\|^2 \ge (x_n^{(k+1)})^2$, increasing C and using $B \le 1 + n/m$ proves (25). If $k = 0$, the entire schedule satisfies $\sum_{t=1}^{n} h_t < S_0$, and the pure quadratic $F(x) = x^2/(2\kappa)$ gives the same conclusion. □

The factor $\exp[-C\log(m+1)]$ in (25) is independent of the horizon n. The next lemma shows that we can remove this factor using a repetition argument. We also include the corresponding lower bound for function suboptimality.

Lemma 5.4. *Under the hypotheses of Proposition 5.3, let m and C be as there. For every integer $n \ge 1$ and every schedule $H \in [0,\infty)^n$,*

$$\min\{D_{n,\kappa}(H), E_{n,\kappa}(H)\} \ge \exp\left[-\frac{Cn}{m}\log(m+1)\right]. \tag{28}$$

Proof. Fix n and H. For each integer $K \ge 1$, let $H^{[K]}$ denote K consecutive copies of H, a schedule of length Kn. By (1), the squared distance to the minimizer after each copy is at most $D_{n,\kappa}(H)$ times its value before that copy. The same holds for function suboptimality with $E_{n,\kappa}(H)$. Applying these bounds on the same objective gives

$$D_{Kn,\kappa}(H^{[K]}) \le D_{n,\kappa}(H)^K, \qquad E_{Kn,\kappa}(H^{[K]}) \le E_{n,\kappa}(H)^K.$$

Strong convexity and smoothness also give $D_{Kn,\kappa}(H^{[K]}) \le \kappa E_{Kn,\kappa}(H^{[K]})$. Combining these inequalities with Proposition 5.3 at horizon Kn yields

$$\begin{aligned}\exp\left[-C\left(1+\frac{Kn}{m}\right)\log(m+1)\right] &\le D_{Kn,\kappa}(H^{[K]}) \\ &\le \min\left\{D_{n,\kappa}(H)^K,\, \kappa E_{n,\kappa}(H)^K\right\}.\end{aligned}$$

Taking Kth roots and letting $K \to \infty$ proves (28), since $\kappa^{1/K} \to 1$. □

Choosing the Bending Parameter. Note that the block length m defined in (24) depends on β, and increasing m strengthens the bound in (28). We therefore choose β as a function of κ to optimize this bound. The explicit constants in Lemma 3.1 satisfy

$$c_\beta = 1 + O(\beta), \qquad \log a_\beta = O(1/\beta), \qquad \frac{1}{p(c_\beta)} = 1/p_{\mathrm{sil}} - O(\beta).$$

The first two estimates are given in (11), and the last follows by differentiating (16) near $c = 1$. For sufficiently large κ, choose

$$\beta := (\log \kappa)^{-1/2} \le 1/4.$$

Since $\log a_\beta = O(\sqrt{\log \kappa}) = o(\log \kappa)$, we have $\kappa/a_\beta \to \infty$. Thus $S_0 \ge 1$ and $m \ge 1$ for sufficiently large κ, as required by Proposition 5.3. The definition of m and the estimates above give

$$\begin{aligned} \log m &\ge \frac{1}{p(c_\beta)}(\log \kappa - \log a_\beta) - O(1) \\ &\ge \frac{\log \kappa}{p_{\mathrm{sil}}} - C\big(\beta \log \kappa + 1/\beta\big) \ge \frac{\log \kappa}{p_{\mathrm{sil}}} - C'\sqrt{\log \kappa}. \end{aligned} \tag{29}$$

The loss $O(\beta \log \kappa)$ comes from the exponent $1/p(c_\beta)$, while $O(1/\beta)$ comes from $\log a_\beta$. Our choice of β balances these two losses.

Proof of Theorem 1.1 and Corollary 1.2. For the chosen β, Lemma 5.4 gives (28). Since $m \le \kappa$ by its definition in Proposition 5.3, we have $\log(m+1) \le C \log \kappa$. Combining this estimate with (28) and (29) yields (2). To prove Corollary 1.2, suppose that a schedule H achieves $D_{n,\kappa}(H) \le \varepsilon$ or $E_{n,\kappa}(H) \le \varepsilon$. The lower bound in (2) must then also be at most ε. Taking logarithms and rearranging gives

$$n \ge \frac{\kappa^{1/p_{\mathrm{sil}}}}{C_1 e^{C_2\sqrt{\log \kappa}} \log \kappa} \log \frac{1}{\varepsilon}.$$

This necessary condition holds for every schedule H and for either error criterion, proving (3). □

6 Concluding Remarks

In this paper, we proved an almost tight iteration lower bound for GD with arbitrary nonnegative predetermined stepsizes on smooth strongly convex objectives, matching the silver upper bound up to a subpolynomial factor in κ. Three open questions remain:

1. *Can we reduce the subpolynomial factor $e^{C\sqrt{\log \kappa}} \log \kappa$, which separates our lower bound in (3) from the silver upper bound?* This factor seems unavoidable in our current proof. Reducing or removing this subpolynomial factor may require more refined local hard functions, or even a new framework for constructing hard instances.

2. *Can we extend our lower bound to predetermined schedules that may contain negative stepsizes?* Our previous work [YL26a] gives an $\Omega(n^{-1.6342})$ lower bound for GD with possibly negative predetermined stepsizes on smooth convex objectives, leaving a polynomial gap to the silver upper bound. We believe that a more refined construction and analysis can yield lower bounds matching the silver upper bounds up to subpolynomial factors in both the smooth convex and smooth strongly convex settings, even when negative stepsizes are allowed.

3. *Can we connect the smooth convex and strongly convex lower bounds?* Nesterov's analysis of accelerated first-order methods gives a single upper bound covering both settings [Nes04, Theorem 2.2.2]. For GD, the smooth convex silver schedule arises as $\mu \to 0$ after omitting the terminal step of the strongly convex schedule. Conversely, repeating suitable smooth convex silver blocks recovers the strongly convex iteration bound up to constant factors [AP25b, Remark 1 and Corollary 2]. It would be interesting to develop a unified lower bound construction that captures the transition between these settings.

Acknowledgments

We thank Benjamin Grimmer for helpful discussions during the *Readings in Optimization* seminar at MIT and for reading and commenting on the manuscript.

AI Disclosure

This work was driven by the human authors' conceptual direction. ChatGPT-6 Astra was utilized as an interactive assistant to execute instructions, work out technical details, and refine arguments through iterative feedback. After digesting the arguments in detail, the authors reorganized and rewrote the presentation. They take full responsibility for the correctness and originality of all content.

References

[AGZ21] Naman Agarwal, Surbhi Goel, and Cyril Zhang. Acceleration via fractal learning rate schedules. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 87–99. PMLR, 2021.

[Alt18] Jason Altschuler. Greed, hedging, and acceleration in convex optimization. Master's thesis, Massachusetts Institute of Technology, 2018.

[AP24] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by random stepsizes: Hedging, equalization, and the arcsine stepsize schedule. *arXiv preprint arXiv:2412.05790v2*, 2024. To appear in Foundations of Computational Mathematics.

[AP25a] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging: Multi-step descent and the Silver stepsize schedule. *Journal of the ACM*, 72(2):1–38, 2025. Article 12.

[AP25b] Jason M. Altschuler and Pablo A. Parrilo. Acceleration by stepsize hedging: Silver Stepsize Schedule for smooth convex optimization. *Mathematical Programming*, 213(1–2):1105–1118, 2025.

[AP26] Jason M. Altschuler and Pablo A. Parrilo. Stepsize hedging: An alternative mechanism for accelerating gradient descent. *INFORMS Computing Society Newsletter*, pages 19–27, June 2026. Research Highlight.

[AZH16] Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[BA25] Jinho Bok and Jason M. Altschuler. Accelerating proximal gradient descent via Silver stepsizes. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 421–453. PMLR, 2025.

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.

[CC55] Jean Bronfenbrenner Crockett and Herman Chernoff. Gradient methods of maximization. *Pacific Journal of Mathematics*, 5(1):33–50, 1955.

[CWM+24] Yuhang Cai, Jingfeng Wu, Song Mei, Michael Lindsey, and Peter L. Bartlett. Large stepsize gradient descent for non-homogeneous two-layer networks: Margin improvement and fast optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[DGVPR24] Shuvomoy Das Gupta, Bart P. G. Van Parys, and Ernest K. Ryu. Branch-and-Bound Performance Estimation Programming: A unified methodology for constructing optimal optimization methods. *Mathematical Programming*, 204(1–2):567–639, 2024.

[dKGT17] Etienne de Klerk, François Glineur, and Adrien B. Taylor. On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions. *Optimization Letters*, 11(7):1185–1199, 2017.

[Dro17] Yoel Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 39:1–16, 2017.

[Dro26] Yoel Drori. Exact oracle complexity for function-value suboptimality under weighted initial conditions. *arXiv preprint arXiv:2609.33360*, 2026.

[DT14] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: A novel approach. *Mathematical Programming*, 145(1–2):451–482, 2014.

[DT20] Yoel Drori and Adrien B. Taylor. Efficient first-order methods for convex minimization: A constructive approach. *Mathematical Programming*, 184:183–220, 2020.

[DT22] Yoel Drori and Adrien Taylor. On the oracle complexity of smooth strongly convex minimization. *Journal of Complexity*, 68:101590, 2022.

[Gav50] Mark K. Gavurin. The use of polynomials of best approximation for improving the convergence of iterative processes. *Uspekhi Matematicheskikh Nauk*, 5(3(37)):156–160, 1950. In Russian. English translation in NBS Report 2007 (1952), pp. 44–50.

[GJP26] Benjamin Grimmer, Sunghyeon Jo, and Chanwoo Park. On minimax optimality and uniqueness of fixed-step first-order methods for smooth convex optimization. *arXiv preprint arXiv:2609.07671*, 2026.

[Gri24] Benjamin Grimmer. Provably faster gradient descent via long steps. *SIAM Journal on Optimization*, 34(3):2588–2608, 2024.

[GSW25a] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Accelerated objective gap and gradient norm convergence for gradient descent via long steps. *INFORMS Journal on Optimization*, 7(2):156–169, 2025.

[GSW25b] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Composing optimized stepsize schedules for gradient descent. *Mathematics of Operations Research*, 2025. Articles in Advance.

[GSW26] Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Subgame perfection and dynamic optimality in optimization. *SIAG on Optimization Views and News*, 34(1):2–8, 2026.

[GV61] Gene H. Golub and Richard S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. Part I. *Numerische Mathematik*, 3:147–156, 1961.

[HS52] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[JCY26] Minchan Jung, Hanseul Cho, and Chulhee Yun. Stronger lower bounds for (non-)anytime acceleration of gradient descent. *arXiv preprint arXiv:2609.04032v2*, 2026.

[Kal17] Zdeněk Kalousek. Steepest descent method with random step lengths. *Foundations of Computational Mathematics*, 17(2):359–422, 2017.

[KF16] Donghwan Kim and Jeffrey A. Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical Programming*, 159:81–107, 2016.

[KMM+26] Sacchit Kale, Piyushi Manupriya, Pierre Marion, Francis Bach, and Anant Raj. Stretched exponential convergence of (stochastic) gradient descent for separable logistic regression. *Transactions on Machine Learning Research*, 2026.

[LCJW26] Yu Liu, Kang Chen, Rujun Jiang, and Tianyu Wang. Optimal recursive composition and dyadic phase laws for gradient descent with predetermined stepsizes. *arXiv preprint arXiv:2609.11788*, 2026.

[LF71] V. I. Lebedev and S. A. Finogenov. Ordering of the iterative parameters in the cyclical Chebyshev iterative method. *USSR Computational Mathematics and Mathematical Physics*, 11(2):155–170, 1971.

[LP66] E. S. Levitin and Boris T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.

[MC26] Jianhao Ma and Yuxin Chen. A lower bound for stepsize-based acceleration of gradient descent. *arXiv preprint arXiv:2608.10418*, 2026.

[Mor65] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93(3):273–299, 1965.

[Nes83] Yurii E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nes04] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[NY79] Arkadii S. Nemirovsky and David B. Yudin. *Slozhnost' Zadach i Effektivnost' Metodov Optimizatsii*. Nauka, Moscow, 1979. In Russian. English translation: Problem Complexity and Method Efficiency in Optimization, Wiley, 1983.

[Oym21] Samet Oymak. Provable super-convergence with a large cyclical learning rate. *IEEE Signal Processing Letters*, 28:1645–1649, 2021.

[PB14] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[Pol63] Boris T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

[Pol64] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[Pol87] Boris T. Polyak. *Introduction to Optimization*. Optimization Software, Inc., Publications Division, New York, 1987.

[PZ11] Luc Pronzato and Anatoly Zhigljavsky. Gradient algorithms for quadratic optimization with fast convergence rates. *Computational Optimization and Applications*, 50(3):597–617, 2011.

[RW98] R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, Heidelberg, 1998.

[TD23] Adrien Taylor and Yoel Drori. An optimal gradient method for smooth strongly convex minimization. *Mathematical Programming*, 199(1–2):557–594, 2023.

[THG17] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161(1–2):307–345, 2017.

[Tsa26] Chung-En Tsai. An improved lower bound for non-anytime gradient descent. Blog post, `https://chungentsai.github.io/gd-lower-bounds.html`, 2026.

[WBTY24] Jingfeng Wu, Peter L. Bartlett, Matus Telgarsky, and Bin Yu. Large stepsize gradient descent for logistic loss: Non-monotonicity of the loss improves optimization efficiency. In *Proceedings of the Thirty-Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 5019–5073. PMLR, 2024.

[WMB25] Jingfeng Wu, Pierre Marion, and Peter L. Bartlett. Large stepsizes accelerate gradient descent for regularized logistic regression. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[WMYZ26] Bofan Wang, Shiqian Ma, Junfeng Yang, and Danqing Zhou. Relaxed proximal point algorithm: Tight complexity bounds and acceleration without momentum. *INFORMS Journal on Optimization*, 8(2):141–162, 2026.

[WZCL26] Yue Wu, Weiqiang Zheng, Yang Cai, and Haipeng Luo. Accelerating min-max optimization via power-law stepsizes. *arXiv preprint arXiv:2606.01764*, 2026. Accepted at NeurIPS 2026.

[YL26a] Yuhan Ye and Kaizhao Liu. Improved gradient descent lower bounds beyond Nesterov. *arXiv preprint arXiv:2609.02855*, 2026.

[YL26b] Yuhan Ye and Kaizhao Liu. Silver rate is (almost) optimal for gradient descent. *arXiv preprint arXiv:2609.09152*, 2026.

[You53] David Young. On Richardson's method for solving linear systems with positive definite matrices. *Journal of Mathematics and Physics*, 32(1–4):243–255, 1953.

[ZJ26] Zehao Zhang and Rujun Jiang. Accelerated gradient descent by concatenation of stepsize schedules. *SIAM Journal on Optimization*, 36(2):1182–1210, 2026.

[ZLDC25] Zihan Zhang, Jason D. Lee, Simon S. Du, and Yuxin Chen. Anytime acceleration of gradient descent. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 5991–6013. PMLR, 2025.

[ZPB13] Anatoly Zhigljavsky, Luc Pronzato, and Elena Bukina. An asymptotically optimal gradient algorithm for quadratic optimization with low computational cost. *Optimization Letters*, 7(6):1047–1059, 2013.

A Proofs for the Local Hard Functions

In this section, we prove Lemmas 3.1 to 3.3 by constructing Φ_i and the next threshold ℓ_{i+1} from the given ℓ_i, D_i and gap stepsizes. The construction ensures (8), with D_{i+1} determined by the checkpoint update. We bound the transfer ratio D_{i+1}/D_i from below to limit the loss of excess, and the relative threshold ℓ_{i+1}/D_{i+1} from above to limit contraction loss in the next gap.

All three constructions use the following property of Moreau envelopes of support functions.

Fact A.1 (Moreau envelopes of support functions)**.** *Let $K \subset \mathbb{R}^2$ be compact and convex, with $0 \in K$. Write $\sigma_K(u) = \max_{g\in K}\langle g,u\rangle$ for its support function. Its Moreau envelope with parameter 1 satisfies*

$$\begin{aligned}(\mathrm{env}_1\,\sigma_K)(x) &= \min_{y\in\mathbb{R}^2}\left\{\sigma_K(y)+\frac{1}{2}\|x-y\|^2\right\}\\ &= \max_{g\in K}\left\{\langle g,x\rangle-\frac{1}{2}\|g\|^2\right\}.\end{aligned}$$

This envelope is nonnegative, convex, and 1-smooth, and

$$\nabla(\mathrm{env}_1\,\sigma_K)(x) = \Pi_K(x).$$

Here Π_K denotes Euclidean projection onto K. For $g \in K$, the equality $g = \Pi_K(x)$ holds if and only if

$$\langle x-g, w-g\rangle \le 0 \qquad \text{for every } w \in K. \tag{30}$$

The envelope identities follow from [Mor65, Propositions 5.b, 7.b, and 7.d]. For background, see [RW98, Sections 1.G and 2.D] and [PB14, Sections 3.1 and 6.4.2].

A.1 Proof of Lemma 3.1

Constructing the gap trajectory. Fix $\kappa \ge 4$, $0 < \beta \le 1/4$, $\ell_i \ge 0$, and the nonnegative gap stepsizes with $s_i/\kappa \le 1/16$. As in our smooth convex construction [YL26b, Section 3.1], we first construct a two-dimensional gap trajectory. After scaling and translation, this will give the GD trajectory generated by the bending component Φ_i during gap i. We obtain the gradients by scaling vectors on a circular arc with center $C := (\beta,\beta)$ and radius R, where

$$R = 1+\beta, \qquad c_\beta = \beta + \sqrt{1+2\beta}. \tag{31}$$

Let K be the convex hull of the origin and the shorter arc of this circle joining $(c_\beta, 0)$ to $(\beta,-1)$.

The quadratic term in (6) also contracts the iterates toward the origin. Its effect in the new coordinates depends on the scaling and translation, so we specify these first. For a scale $\rho > 0$ and a parameter $\tau \ge 1$, both chosen later, define

$$\Phi_i(X,Y) := (\mathrm{env}_1\,\sigma_{\rho K})\big((X,Y)-z^\circ\big), \qquad z^\circ := \big(\ell_i + \ell_{i+1}/\beta, \ell_{i+1}\big), \qquad \ell_{i+1} := \rho\tau.$$

By Fact A.1, $\nabla\Phi_i(z^\circ+\rho r) = \rho\Pi_K(r)$ for every $r \in \mathbb{R}^2$. We therefore write the gap trajectory as $z^\circ + \rho r_j$ and use gradients ρv_j, where $v_j = \Pi_K(r_j)$. Dividing (9) with $f_j = 0$ by ρ gives

$$r_j + \frac{z^\circ}{\rho} = \left(1-\frac{h_j}{\kappa}\right)\left(r_{j-1}+\frac{z^\circ}{\rho}\right) - \frac{(1-1/\kappa)h_j}{2}v_{j-1} \quad (1 \le j \le m_{\mathrm{gap}}), \tag{32}$$

where $z^\circ/\rho = (\ell_i/\rho + \tau/\beta, \tau)$. The quadratic part contracts toward $-z^\circ/\rho$, the point corresponding to the original origin $(X,Y) = (0,0)$.

Lemma A.2 constructs points r_j backward from $r_{m_{\mathrm{gap}}} = (\beta,-1)$ so that (32) holds with $v_j = \Pi_K(r_j)$. Since the center of contraction depends on τ, the lemma also chooses τ so that the trajectory in (X,Y) starts at $Y_0 = 0$.

Lemma A.2 (Gap trajectory). *For every $\rho > 0$, there exist $\tau \geq 1$, points $r_0, \ldots, r_{m_{\mathrm{gap}}}$, and vectors $v_0, \ldots, v_{m_{\mathrm{gap}}}$ on the arc satisfying (32), such that $r_{m_{\mathrm{gap}}} = v_{m_{\mathrm{gap}}} = (\beta, -1)$, r_0 has second coordinate $-\tau$, and*

$$v_j = \Pi_K(r_j), \qquad r_j \in [\beta, \infty) \times [-\tau, -1] \quad (0 \leq j \leq m_{\mathrm{gap}}).$$

Let P be the first coordinate of the weighted sum

$$\frac{1 - 1/\kappa}{2} \sum_{j=1}^{m_{\mathrm{gap}}} h_j \left[\prod_{l=j+1}^{m_{\mathrm{gap}}} \left(1 - \frac{h_l}{\kappa}\right) \right] v_{j-1}. \tag{33}$$

Then

$$P \leq \frac{1 - 1/\kappa}{2} c_\beta s_i, \qquad \tau \leq 1 + \frac{1 - 1/\kappa}{2} s_i, \qquad \tau \leq \beta \exp\left[\frac{\sqrt{2}R - \beta}{\beta}\right].$$

Moreover, τ and P can be chosen as continuous functions of ρ.

The term ρP is the contribution of the gradient steps to the decrease in X over the gap, after accounting for later contractions. Every arc vector has first coordinate at most c_β, giving the bound on P. This accounts for the coefficient c_β in (12). The bound on τ in terms of s_i controls $\ell_{i+1} = \rho\tau$, and hence the relative threshold. The last bound gives the exponential term in the constant

$$a_\beta := 8 + 4\beta + 4\exp\left[\frac{\sqrt{2}R - \beta}{\beta}\right]. \tag{34}$$

It satisfies $a_\beta \geq 8$ and $\log a_\beta = O(1/\beta)$. The lemma is proved in Section A.4, and Fig. 6 illustrates one backward step.

Choosing the scale and verifying the local hard function. We now choose ρ in the function Φ_i defined above so that $z^\circ + \rho r_0 = (\ell_i + D_i, 0)$. We then verify the zero conditions and compute the excess after the checkpoint update.

Proof of Lemma 3.1. For each $\rho > 0$, choose the quantities in Lemma A.2 with τ and P continuous in ρ, and write $(X_j, Y_j) := z^\circ + \rho r_j$ with $\ell_{i+1} = \rho\tau$. Since r_0 has second coordinate $-\tau$, we have $Y_0 = 0$. The bounds on P, τ and (34) give

$$\beta + P + \frac{\tau}{\beta} \leq \frac{a_\beta}{4} + \frac{1 - 1/\kappa}{2} c_\beta s_i. \tag{35}$$

Step 1. Choosing the scale ρ. We first choose ρ so that the scaled trajectory starts at $(\ell_i + D_i, 0)$. Multiplying (32) by ρ gives

$$(X_j, Y_j) = \left(1 - \frac{h_j}{\kappa}\right)(X_{j-1}, Y_{j-1}) - \frac{(1 - 1/\kappa)h_j}{2} \rho v_{j-1} \quad (1 \leq j \leq m_{\mathrm{gap}}).$$

Expanding the first coordinate and using (33), we obtain $X_{m_{\mathrm{gap}}} = \chi_i X_0 - \rho P$. Since $X_{m_{\mathrm{gap}}} = \ell_i + \rho(\beta + \tau/\beta)$ and $\chi_i(\ell_i + D_i) - \ell_i = D_i\eta_i$, the condition $X_0 = \ell_i + D_i$ is equivalent to

$$\rho \left[\beta + P + \frac{\tau}{\beta}\right] = D_i\eta_i. \tag{36}$$

Here P, τ depend on ρ. By Lemma A.2, the coefficient $\beta + P + \tau/\beta$ is continuous in ρ; it is bounded above by (35) and below by $\beta + 1/\beta > 0$. Thus the left side tends to zero as $\rho \downarrow 0$ and to infinity as $\rho \to \infty$. Since $D_i\eta_i > 0$, choose a solution and use this ρ in the function Φ_i defined above. By Fact A.1 and Lemma A.2,

$$\nabla\Phi_i(X_j, Y_j) = \Pi_{\rho K}(\rho r_j) = \rho\Pi_K(r_j) = \rho v_j.$$

The displayed recurrence therefore coincides with (9) with $f_j = 0$. Starting from the matched initial point, GD visits (X_j, Y_j) by induction on j. Since $r_j \in [\beta, \infty) \times [-\tau, -1]$, we have $X_j \geq \ell_i + \rho(\beta + \tau/\beta) > \ell_i$ and $0 \leq Y_j \leq \rho(\tau - 1) < \ell_{i+1}$ throughout the gap. Moreover, (35) and (36) imply

$$\frac{\rho}{D_i} \geq \frac{2\eta_i}{a_\beta/2 + (1 - 1/\kappa)c_\beta s_i}. \tag{37}$$

The function and threshold do not depend on b_i.

Step 2. Verifying $\Phi_i = 0$ and $\partial_Y \Phi_i = 0$. On the arc, $p \geq \beta$ and $0 \leq q \leq 1$, so $p, q \geq 0$ and $q \leq p/\beta$ throughout K by convexity. Moreover, $p \leq c_\beta$ throughout K, so $(p, 0) \in [0, c_\beta] \times \{0\} \subset K$ whenever $(p, -q) \in K$. Apply Lemma A.3, proved in Section A.5, to ρK with $\theta = \beta$ and translation z°. It follows that Φ_i is nonnegative, convex, and 1-smooth, with $\partial_Y \Phi_i \leq 0$ everywhere and

$$\Phi_i(X, Y) = 0 \quad (X \leq \ell_i, Y \geq 0), \qquad \partial_Y \Phi_i(X, Y) = 0 \quad (X \in \mathbb{R}, Y \geq \ell_{i+1}).$$

Step 3. The checkpoint update. Just before the checkpoint update, $r_{m_{\text{gap}}} = v_{m_{\text{gap}}} = (\beta, -1)$ gives $Y_{m_{\text{gap}}} = \rho(\tau - 1)$ and $\partial_Y \Phi_i = -\rho$. For any $b_i > 8$, the step of size b_i yields

$$D_{i+1} = Y_{m_{\text{gap}}+1} - \ell_{i+1} = \rho \left[\frac{1 - 1/\kappa}{2} b_i - \frac{b_i}{\kappa}(\tau - 1) - 1 \right].$$

Using the bound on τ in Lemma A.2 and $4(1 - 1/\kappa)(1 - s_i/\kappa) > 1$, we obtain

$$D_{i+1} \geq \frac{1 - 1/\kappa}{2} \rho (1 - s_i/\kappa)(b_i - 8) > 0.$$

This proves (8). Dividing by D_i, using (37) and $(1 - 1/\kappa)c_\beta \geq 3/4$, gives the first bound in (12):

$$\frac{D_{i+1}}{D_i} \geq \frac{(1 - 1/\kappa)(1 - s_i/\kappa)(b_i - 8)\eta_i}{a_\beta/2 + (1 - 1/\kappa)c_\beta s_i} \geq \frac{b_i - 8}{c_\beta(a_\beta + s_i + 8)} \eta_i (1 - s_i/\kappa).$$

For the second bound, $\ell_{i+1} = \rho\tau$, the bound on τ, and $(1 - 1/\kappa)(1 - s_i/\kappa) \geq 45/64$ give

$$\frac{\ell_{i+1}}{D_{i+1}} \leq \frac{2 + s_i}{(1 - 1/\kappa)(1 - s_i/\kappa)(b_i - 8)} \leq 4 \frac{s_i + 2}{b_i - 8}.$$

Finally, (31) and (34) give the parameter estimates in (11). □

A.2 Proof of Lemma 3.2

Proof. **Step 1. Choosing the Huber scale.** For a positive trial scale δ_{H}, define points (X_j, Y_j) from $(X_0, Y_0) = (\ell_i + D_i, 0)$ by (9), using the constant vector $(\delta_{\text{H}}, -\delta_{\text{H}})$ in place of $\nabla \Phi_i$. For every sequence $f_j \geq 0$, solving this linear recurrence gives, for $0 \leq j \leq m_{\text{gap}}$,

$$Y_j = \frac{\kappa - 1}{2} \delta_{\text{H}} \left[1 - \prod_{l=1}^{j} (1 - h_l/\kappa) \right],$$

$$\begin{aligned} X_j - Y_j - \ell_i &\geq [\ell_i + D_i + (\kappa - 1)\delta_{\text{H}}] \prod_{l=1}^{j} (1 - h_l/\kappa) - \ell_i - (\kappa - 1)\delta_{\text{H}} \\ &\geq D_i \eta_i - (\kappa - 1)(1 - \chi_i)\delta_{\text{H}}. \end{aligned}$$

The first inequality uses $f_j \geq 0$ and $0 \leq h_j < \kappa$, and the second uses $\prod_{l=1}^{j}(1-h_l/\kappa) \geq \chi_i$. The Huber gradient equals $(\delta_{\mathrm{H}}, -\delta_{\mathrm{H}})$ when $X - Y - \ell_i \geq 2\delta_{\mathrm{H}}$. To keep every gap point in this region, choose the scale so that the last lower bound equals $2\delta_{\mathrm{H}}$:

$$\delta_{\mathrm{H}} = \frac{D_i \eta_i}{2 + (\kappa - 1)(1 - \chi_i)} > 0.$$

This is the choice in (13), and it gives $X_j - Y_j - \ell_i \geq 2\delta_{\mathrm{H}}$ throughout the gap. Since Y_j is nondecreasing in j, set

$$\ell_{i+1} = Y_{m_{\mathrm{gap}}} = \frac{\kappa - 1}{2}(1 - \chi_i)\delta_{\mathrm{H}}.$$

The resulting function Φ_i and threshold ℓ_{i+1} depend only on the incoming data and gap stepsizes, so neither changes with the sequence f_j or the checkpoint stepsize b_i. The margin bound and monotonicity of Y_j give $X_j \geq \ell_i$ and $0 \leq Y_j \leq \ell_{i+1}$ for $0 \leq j \leq m_{\mathrm{gap}}$.

Step 2. Verifying the local hard function. For $K = \mathrm{conv}\{0, (\delta_{\mathrm{H}}, -\delta_{\mathrm{H}})\}$, the Huber repair function in (13) has the Moreau representation

$$\Phi_i(X, Y) = (\mathrm{env}_1 \sigma_K)(X - \ell_i, Y) = \tfrac{1}{2} H_{2\delta_{\mathrm{H}}}(X - Y - \ell_i).$$

By Lemma A.3 with $\theta = 1$ and translation $(\ell_i, 0)$, this function is nonnegative, convex, and 1-smooth, with $\partial_Y \Phi_i \leq 0$ everywhere and $\Phi_i(X, Y) = 0$ when $X \leq \ell_i$, $Y \geq 0$. At each constructed point, the condition $X_j - Y_j - \ell_i \geq 2\delta_{\mathrm{H}}$ makes the projection onto K equal to $(\delta_{\mathrm{H}}, -\delta_{\mathrm{H}})$. Indeed, the left side of (30) at the other endpoint 0 is

$$-\delta_{\mathrm{H}}(X_j - Y_j - \ell_i - 2\delta_{\mathrm{H}}) \leq 0.$$

Thus $\nabla \Phi_i(X_j, Y_j) = (\delta_{\mathrm{H}}, -\delta_{\mathrm{H}})$ for $0 \leq j \leq m_{\mathrm{gap}}$. Induction in (9) confirms that the constructed points are the GD iterates for every nonnegative sequence f_j.

Step 3. The checkpoint update. For any $b_i > 0$ and $f_{m_{\mathrm{gap}}+1} \geq 0$, this update uses $Y_{m_{\mathrm{gap}}} = \ell_{i+1}$ and $\partial_Y \Phi_i = -\delta_{\mathrm{H}}$, giving

$$D_{i+1} = b_i \left(\frac{1 - 1/\kappa}{2} \delta_{\mathrm{H}} - \frac{\ell_{i+1}}{\kappa} \right) = \frac{1 - 1/\kappa}{2} b_i \chi_i \delta_{\mathrm{H}} > 0.$$

Here $\chi_i > 0$ because every gap stepsize is strictly less than κ. This proves (8); substituting the chosen δ_{H} and ℓ_{i+1} gives the exact ratios in (14). Finally, telescoping gives

$$\kappa(1 - \chi_i) = \sum_{j=1}^{m_{\mathrm{gap}}} h_j \prod_{l=j+1}^{m_{\mathrm{gap}}} (1 - h_l/\kappa), \qquad \chi_i s_i \leq \kappa(1 - \chi_i) \leq s_i.$$

Together with $\chi_i \geq 1 - s_i/\kappa$, these bounds prove (15) when $s_i < \kappa$. □

A.3 Proof of Lemma 3.3

Proof. **Step 1. Raising the threshold.** Use the Huber repair trajectory from Section A.2, with the same ℓ_i, D_i and gap stepsizes. For every nonnegative sequence f_j, it satisfies

$$X_j - Y_j - \ell_i \geq 2\delta_{\mathrm{H}}, \qquad 0 \leq Y_j \leq Y_{m_{\mathrm{gap}}} \quad (0 \leq j \leq m_{\mathrm{gap}}).$$

The repair threshold is $Y_{m_{\mathrm{gap}}}$. For the bridge, raise it by δ_{H} and denote the new threshold by ℓ_{i+1}:

$$\ell_{i+1} = Y_{m_{\mathrm{gap}}} + \delta_{\mathrm{H}} = \left[1 + \frac{\kappa - 1}{2}(1 - \chi_i) \right] \delta_{\mathrm{H}} = \frac{D_i \eta_i}{2}.$$

Every gap point is now at least δ_{H} below this threshold, so $Y_j - \ell_{i+1} \leq -\delta_{\mathrm{H}}$. Since $Y_{m_{\mathrm{gap}}}$ and δ_{H} do not depend on f_j or b_i, neither does this threshold.

Step 2. Verifying the local hard function. The bridge construction in Section 3.3 enlarges the segment used for Huber repair to the triangle

$$K_\triangle = \mathrm{conv}\{0, (\delta_{\mathrm{H}}, -\delta_{\mathrm{H}}), (\delta_{\mathrm{H}}, 0)\}.$$

By Fact A.1, the bridging function Φ_i can be written as

$$\Phi_i(X,Y) = (\mathrm{env}_1\, \sigma_{K_\triangle})(X - \ell_i - \ell_{i+1}, Y - \ell_{i+1}).$$

Every $(p,-q) \in K_\triangle$ satisfies $0 \le q \le p$, and $(p,0)$ lies on its horizontal edge. Applying Lemma A.3 with $\theta = 1$ and translation $(\ell_i + \ell_{i+1}, \ell_{i+1})$ shows that Φ_i is nonnegative, convex, and 1-smooth, with $\partial_Y \Phi_i \le 0$ everywhere and

$$\Phi_i(X,Y) = 0 \quad (X \le \ell_i, Y \ge 0), \qquad \partial_Y \Phi_i(X,Y) = 0 \quad (X \in \mathbb{R}, Y \ge \ell_{i+1}).$$

At each repair iterate, the translated point $(X_j - \ell_i - \ell_{i+1}, Y_j - \ell_{i+1})$ projects onto $(\delta_{\mathrm{H}}, -\delta_{\mathrm{H}})$. The expression in (30) is affine in w, so it suffices to check the other vertices 0 and $(\delta_{\mathrm{H}}, 0)$. The corresponding inner products are

$$-\delta_{\mathrm{H}}(X_j - Y_j - \ell_i - 2\delta_{\mathrm{H}}) \le 0, \qquad \delta_{\mathrm{H}}(Y_j - \ell_{i+1} + \delta_{\mathrm{H}}) \le 0.$$

Both inequalities follow from Step 1, so the criterion holds throughout $K_\triangle$. Thus the bridge and repair gradients agree for $0 \le j \le m_{\mathrm{gap}}$. Induction in (9), using the same initial point and sequence f_j, shows that their trajectories agree through the checkpoint update. In particular, $X_j \ge \ell_i$ and $0 \le Y_j < \ell_{i+1}$ throughout the gap.

Step 3. The checkpoint update. For every $b_i > 4$ and every nonnegative sequence f_j, the bridge has the same value of Y after this update as the repair, but its threshold is δ_{H} larger. Subtracting this amount from the repair excess gives

$$D_{i+1} = Y_{m_{\mathrm{gap}}+1} - \ell_{i+1} = \delta_{\mathrm{H}} \left[\frac{(1 - 1/\kappa) b_i \chi_i}{2} - 1 \right]. \tag{38}$$

Since $s_i/\kappa \le 1/16$ and $\kappa \ge 4$, we have $(1 - 1/\kappa)\chi_i \ge 45/64 > 1/2$, and hence

$$(1 - 1/\kappa) b_i \chi_i - 2 \ge (1 - 1/\kappa)\chi_i (b_i - 4) > 0.$$

Thus $D_{i+1} > 0$, completing (8). Substituting (13) into (38) and using $(\kappa - 1)(1 - \chi_i) \le s_i$ gives

$$\frac{D_{i+1}}{D_i} \ge \frac{\eta_i (1 - 1/\kappa)\chi_i (b_i - 4)}{2(s_i + 2)} \ge \frac{1}{4} \eta_i \frac{b_i - 4}{s_i + 2},$$
$$\frac{\ell_{i+1}}{D_{i+1}} \le \frac{s_i + 2}{(1 - 1/\kappa)\chi_i (b_i - 4)} \le 2 \frac{s_i + 2}{b_i - 4}.$$

□

A.4 Proof of Lemma A.2

We use the notation of Section A.1.

Proof. We construct the points backward for a trial value of τ, then choose τ so that r_0 has second coordinate $-\tau$. We verify $v_j = \Pi_K(r_j)$ and finally bound τ. Write $r_j = (u_j, -y_j)$ and $v_j = (p_j, -q_j)$, where $p_j = p(q_j)$ and

$$p(q) := \beta + \sqrt{R^2 - (q+\beta)^2} \quad (0 \le q \le 1).$$

We must show $y_0 = \tau$, $u_j \ge \beta$, and $1 \le y_j \le \tau$. Let Q be the weighted sum of the q_{j-1} with the weights in (33), so that the vector in that display is $(P, -Q)$.

To choose the arc vectors, we measure directions from the center C. For a point $z = (z_1, z_2)$ with $z_1 > \beta$, let $\omega(z) := (\beta - z_2)/(z_1 - \beta)$ be the downward slope from C to z, and set $\omega(z) = +\infty$ if $z_1 = \beta > z_2$. Along the arc, $\omega(p(q), -q) = (q+\beta)/(p(q) - \beta)$ increases from $\omega_c := \beta/(c_\beta - \beta)$ at $(c_\beta, 0)$ to $+\infty$ at $(\beta, -1)$. Thus each slope in $[\omega_c, +\infty]$ determines exactly one arc point.

For an empty gap, take $r_0 = v_0 = (\beta, -1)$, so $P = Q = 0$ and $\tau = 1$. Each zero stepsize leaves the point and gradient unchanged. We therefore set $r_{j-1} = r_j$, $v_{j-1} = v_j$ when $h_j = 0$ and apply the following rule only when $h_j > 0$.

Fix $\rho > 0$ and a trial value $\tau \ge 1$, and start from $r_{m_{\mathrm{gap}}} = v_{m_{\mathrm{gap}}} = (\beta, -1)$. Given r_j with $u_j \ge \beta$, rearrange (32) as $r_{j-1} = \widehat{r}_j + \alpha_j v_{j-1}$, where

$$\widehat{r}_j := r_j + \frac{h_j/\kappa}{1 - h_j/\kappa}\left(r_j + \frac{z^\circ}{\rho}\right), \qquad \alpha_j := \frac{(1 - 1/\kappa)h_j}{2(1 - h_j/\kappa)}.$$

Here $\widehat{r}_j$ undoes the quadratic contraction in step j. Choose v_{j-1} where the ray from C through $\widehat{r}_j + \alpha_j C$ meets the arc, or choose $(c_\beta, 0)$ if this ray passes above the corner. Equivalently,

$$\omega(v_{j-1}) = \max\{\omega_c, \omega(\widehat{r}_j + \alpha_j C)\}. \tag{39}$$

Then set $r_{j-1} := \widehat{r}_j + \alpha_j v_{j-1}$. The first coordinate of $r_j + z^\circ/\rho$ is positive, so $\widehat{r}_j$ has first coordinate at least $u_j \ge \beta$, and $\widehat{r}_j + \alpha_j C$ has first coordinate greater than β. Thus the rule is well defined and depends continuously on $\widehat{r}_j$. It also gives $u_{j-1} \ge \beta$, so it applies at every index $j = m_{\mathrm{gap}}, \dots, 1$. This is the rule of our smooth convex construction [YL26b, Appendix A], with center C and with $\widehat{r}_j$ in place of r_j.

Figure 6 shows the geometry of (39). When $q_{j-1} > 0$, the center (β, β), v_{j-1}, and $\widehat{r}_j + \alpha_j(\beta, \beta)$ are collinear, and

$$r_{j-1} - [\widehat{r}_j + \alpha_j(\beta, \beta)] = \alpha_j[v_{j-1} - (\beta, \beta)].$$

Thus the two shaded triangles are similar, and r_{j-1} lies on the same ray from C. When $v_{j-1} = (c_\beta, 0)$, the vector $r_{j-1} - C$ is the sum of $\widehat{r}_j + \alpha_j C - C$ and $\alpha_j((c_\beta, 0) - C)$. Both have positive first coordinates and downward slopes at most ω_c, so their sum does too. Applying this observation at each backward step, with $r_{m_{\mathrm{gap}}} = v_{m_{\mathrm{gap}}}$ as the base case, proves

$$\omega(v_j) = \max\{\omega_c, \omega(r_j)\} \quad (0 \le j \le m_{\mathrm{gap}}). \tag{40}$$

Thus each v_j is where the ray from C through r_j meets the arc, or the corner if this ray passes above it. To prove that $v_j = \Pi_K(r_j)$, we will also need the lower bound on y_j established after choosing τ. First, we show that the gradients turn toward the corner as we work backward:

$$q_{j-1} \le q_j \quad (1 \le j \le m_{\mathrm{gap}}). \tag{41}$$

For $h_j > 0$, the definition of $\widehat{r}_j$ gives

$$\left(1 - \frac{h_j}{\kappa}\right)(\widehat{r}_j + \alpha_j C - C) = (r_j - C) + \frac{h_j}{\kappa}\left(C + \frac{z^\circ}{\rho}\right) + \frac{(1 - 1/\kappa)h_j}{2} C.$$

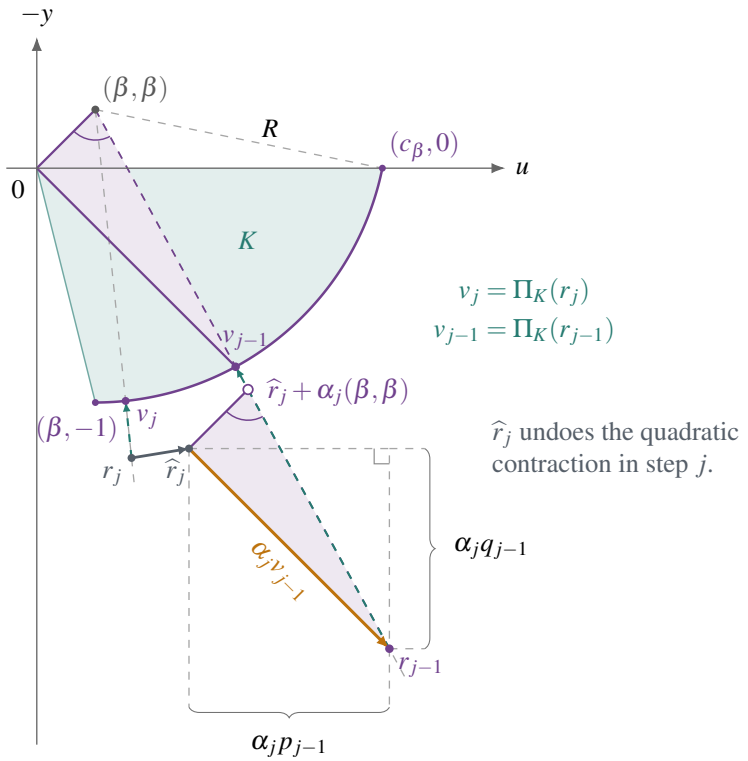


Figure 6: A backward step with $q_{j-1} > 0$. Pale green denotes K. Green dashed arrows show projection onto K, giving the scaled gradients $v_j = \Pi_K(r_j)$ and $v_{j-1} = \Pi_K(r_{j-1})$.

The last two vectors have nonnegative coordinates. Adding such a vector cannot increase $\max\{\omega_c, \omega(\cdot)\}$. Indeed, if $a = (a_1, a_2)$ has nonnegative coordinates and $\omega(z+a) > \omega_c$, then $0 < \beta - z_2 - a_2 \le \beta - z_2$ and $z_1 + a_1 - \beta \ge z_1 - \beta$, so $\omega(z+a) \le \omega(z)$. Since rescaling about C leaves the slope unchanged, (39) and (40) give

$$\omega(v_{j-1}) \le \max\{\omega_c, \omega(r_j)\} = \omega(v_j).$$

This proves (41); zero steps give equality.

Choosing the threshold. The second coordinate of (32) gives

$$\tau - y_j = \left(1 - \frac{h_j}{\kappa}\right)(\tau - y_{j-1}) + \frac{h_j}{\kappa}\frac{\kappa - 1}{2} q_{j-1}.$$

Expanding over the gap and using $y_{m_{\text{gap}}} = 1$ gives $\tau - 1 = \chi_i(\tau - y_0) + Q$. Thus $y_0 = \tau$ if and only if $\tau = 1 + Q$.

As the trial value τ increases, both coordinates of $z^\circ/\rho = (\ell_i/\rho + \tau/\beta, \tau)$ increase. Backward induction from the common terminal point shows that each r_j is coordinatewise nondecreasing in τ. Indeed, if r_j is, then so is $\widehat{r}_j$. The slope comparison above shows that q_{j-1} is nonincreasing in τ, so $v_{j-1} = (p(q_{j-1}), -q_{j-1})$ and $r_{j-1} = \widehat{r}_j + \alpha_j v_{j-1}$ are coordinatewise nondecreasing. Since Q is a nonnegative combination of the q_{j-1}, it is continuous and nonincreasing in τ, with values in $[0, (1 - 1/\kappa)s_i/2]$. On the interval $[1, 1 + (1 - 1/\kappa)s_i/2]$, $\tau - 1 - Q$ is continuous and strictly increasing, nonpositive at the left endpoint and nonnegative at the right endpoint. It therefore has exactly one zero; fix this value of τ. The recursion is jointly continuous in τ and $\rho > 0$, and all these roots lie in the same compact interval. When ρ converges to a positive value, any limit point of τ solves the limiting equation, whose solution is unique. Thus τ, the points, and P are continuous in ρ.

The second-coordinate recurrence is a convex combination of $\tau - y_{j-1}$ and $(\kappa-1)q_{j-1}/2$. Starting from $\tau - y_0 = 0$, induction using (41) gives

$$\tau - y_{j-1} \le \tau - y_j \le \frac{\kappa-1}{2} q_{j-1} \le \frac{\kappa-1}{2} q_j.$$

Since $y_{m_{\mathrm{gap}}} = 1$, this proves $1 \le y_j \le \tau$.

The projection identity. Since $\|C\| = \sqrt{2}\beta < R$, the disk centered at C with radius R contains 0 and hence K. If $0 < q_j < 1$, then (40) puts r_j on the ray from C through v_j, and $y_j \ge 1 \ge q_j$ puts r_j at or beyond v_j. Thus $r_j - v_j$ is an outward normal to K at v_j. If $q_j = 0$, then (40) gives $y_j \le \beta(u_j - c_\beta)/(c_\beta - \beta)$, and $y_j \ge 1$ forces $u_j > c_\beta$. So $r_j - (c_\beta, 0)$ is a nonnegative combination of the outward normals $(c_\beta - \beta, -\beta)$ and $(0,1)$. If $q_j = 1$, only zero steps have been undone, so $r_j = v_j = (\beta, -1)$. The projection criterion (30) gives $\Pi_K(r_j) = v_j$ in every case.

Bounding τ. Since $p_j \le c_\beta$, (33) gives the bound on P. The bound $\tau \le 1 + (1 - 1/\kappa)s_i/2$ was proved above, so it remains to bound τ in terms of β alone. Set $\lambda_j := (y_j + \beta)/(q_j + \beta)$. When $q_j > 0$, (40) gives $r_j - C = \lambda_j(v_j - C)$, so $\lambda_j = \|r_j - C\|/R$. Write $[r, v] := r_1 v_2 - r_2 v_1$ for the two-dimensional determinant. For $v = (p, -q)$ with $p, q \ge 0$, this determinant does not change when a multiple of v is added to r, and does not increase when a vector with nonnegative coordinates is added to r. Since $y_j \le \tau$, the vector $r_j + z^\circ/\rho$, and hence $\widehat{r}_j - r_j$, has nonnegative coordinates. Therefore

$$[r_{j-1}, v_{j-1}] = [\widehat{r}_j, v_{j-1}] \le [r_j, v_{j-1}].$$

If $q_{j-1} > 0$, then also $q_j > 0$. Substituting $r_{j-1} = C + \lambda_{j-1}(v_{j-1} - C)$ and $r_j = C + \lambda_j(v_j - C)$ gives

$$(1 - \lambda_{j-1})[C, v_{j-1}] \le (1 - \lambda_j)[C, v_{j-1}] + \lambda_j [v_j, v_{j-1}].$$

Using $[C, v] = -\beta(p + q)$ and $[v_j, v_{j-1}] = p_{j-1}q_j - p_j q_{j-1}$, we obtain

$$\begin{aligned} \lambda_{j-1} &\le \lambda_j \left[1 + \frac{p_{j-1}(q_j - q_{j-1}) + q_{j-1}(p_{j-1} - p_j)}{\beta(p_{j-1} + q_{j-1})} \right] \\ &\le \lambda_j \exp\left[\frac{\max\{q_j - q_{j-1}, p_{j-1} - p_j\}}{\beta} \right]. \end{aligned} \tag{42}$$

When $q_{j-1} = 0$, the first inequality follows from $y_{j-1} \le y_j$, because

$$\lambda_{j-1} = \frac{y_{j-1} + \beta}{\beta} \le \frac{y_j + \beta}{\beta} = \lambda_j \left(1 + \frac{q_j}{\beta} \right).$$

The second bounds a weighted average by its larger entry and uses $1 + x \le e^x$.

Since q_j is nondecreasing and $p(q) + q$ is concave, the sequence $p_j + q_j$ first increases and then decreases. Moreover, $q_j - q_{j-1} \ge p_{j-1} - p_j$ exactly when $p_j + q_j \ge p_{j-1} + q_{j-1}$. Thus the larger coordinate change is $q_j - q_{j-1}$ before the maximum and $p_{j-1} - p_j$ afterward. These changes telescope:

$$\sum_{j=1}^{m_{\mathrm{gap}}} \max\{q_j - q_{j-1}, p_{j-1} - p_j\} = \max_{0 \le j \le m_{\mathrm{gap}}} (p_j + q_j) - \beta - q_0 \le \sqrt{2}R - \beta - q_0.$$

The last inequality follows from $(p_j - \beta)^2 + (q_j + \beta)^2 = R^2$. Multiplying (42) and using $\lambda_{m_{\mathrm{gap}}} = 1$ gives

$$y_0 + \beta \le (q_0 + \beta) \exp\left[\frac{\sqrt{2}R - \beta - q_0}{\beta} \right] \le \beta \exp\left[\frac{\sqrt{2}R - \beta}{\beta} \right],$$

where the last inequality uses $1 + q_0/\beta \le e^{q_0/\beta}$. Since $y_0 = \tau$, this proves the last bound in Lemma A.2. □

A.5 Projection properties

The following lemma gives sufficient conditions for $\Phi = 0$ and for $\partial_Y \Phi = 0$ in the regions used above.

Lemma A.3 (Projection properties)**.** *Let $K \subset \mathbb{R}^2$ be compact and convex, with $0 \in K$, and let $z^\circ = (z_1, z_2) \in \mathbb{R}^2$. The function*

$$\Phi(X,Y) = (\mathrm{env}_1 \sigma_K)((X,Y) - z^\circ) = \max_{g \in K} \left\{ \langle g, (X,Y) - z^\circ \rangle - \tfrac{1}{2}\|g\|^2 \right\}$$

is nonnegative, convex, and 1-smooth, with $\nabla\Phi(X,Y) = \Pi_K((X,Y) - z^\circ)$. Suppose $K \subset \{(p,-q) : p,q \geq 0\}$. Then:

(i) $\partial_Y \Phi(X,Y) \leq 0$ for all (X,Y).

(ii) If, for some $\theta > 0$ and $\ell_i \geq 0$, every $(p,-q) \in K$ satisfies $q \leq p/\theta$, and $z_2 \geq 0$, $z_1 \geq \ell_i + z_2/\theta$, then

$$\Phi(X,Y) = 0 \qquad \text{whenever } X \leq \ell_i,\ Y \geq 0.$$

(iii) If $(p,0) \in K$ whenever $(p,-q) \in K$, then

$$\partial_Y \Phi(X,Y) = 0 \qquad \text{whenever } Y \geq z_2.$$

Proof. The representation and gradient formula follow from Fact A.1. Since the gradient belongs to K, its second coordinate is nonpositive, proving (i). For (ii), if $X \leq \ell_i$ and $Y \geq 0$, then every $g = (p,-q) \in K$ satisfies

$$\langle g, (X,Y) - z^\circ \rangle = p(X - z_1) - q(Y - z_2) \leq p(\ell_i - z_1) + qz_2 \leq z_2(q - p/\theta) \leq 0.$$

Thus the maximum defining Φ is zero, attained at $g = 0$. For (iii), suppose $Y \geq z_2$ and $g = (p,-q) \in K$ has $q > 0$. Replacing g by $g' = (p,0) \in K$ reduces the squared distance to $(X,Y) - z^\circ$ by

$$\|(X,Y) - z^\circ - g\|^2 - \|(X,Y) - z^\circ - g'\|^2 = 2q(Y - z_2) + q^2 > 0.$$

The projection therefore has second coordinate zero, proving (iii). □

B Proof of Lemma 4.1

We prove Lemma 4.1 here. Throughout, fix $1 \leq c \leq 2$ and $a \geq 8$, and let $p = p(c)$ be as in (16).

B.1 Bounding the minimum checkpoint cost

In the score (18), only the part of a selected stepsize above 8 appears in the numerator. We therefore separate each stepsize into a fixed part $g_j = \min\{h_j, 8\}$ and a selectable part $B_j = (h_j - 8)_+$. The entry B_j is positive exactly when index j can be selected. For the proof, allow arbitrary nonnegative vectors $g = (g_1, \ldots, g_r)$ and $B = (B_1, \ldots, B_r)$ and arrange their entries as

$$g_1, B_1, g_2, B_2, \ldots, g_r, B_r.$$

Only entries $B_j > 0$ may be selected as checkpoints. The entries g_j are included in gap sums and are never selected. For a selected B_j, let $\widetilde{s}_j$ be the sum of all entries after the previous selected B and before B_j. For

the first selected position, the sum starts at the beginning of the list. Let s_{tail} be the sum of the entries after the final selected B. Define the checkpoint score $P(T)$ and *cost* $Q_\Lambda(T;g,B)$ by

$$P(T)=\prod_{j\in T}\frac{B_j}{c(a+\widetilde{s}_j)},\qquad Q_\Lambda(T;g,B)=\frac{c(\Lambda+s_{\text{tail}})}{P(T)},\qquad \Lambda\ge a.$$

Here the *terminal constant* Λ appears in the factor $c(\Lambda+s_{\text{tail}})$ for the gap after the last checkpoint. For the empty set, $P(\varnothing)=1$ and $Q_\Lambda(\varnothing;g,B)=c[\Lambda+\sum_j(g_j+B_j)]$. Write

$$V_\Lambda(g,B):=\min_{T\text{ admissible}}Q_\Lambda(T;g,B),\qquad U_\Lambda(g):=\sup_{B\in[0,\infty)^r}V_\Lambda(g,B).$$

The entries g remain fixed in this supremum. Taking the supremum over B lets us bound the minimum cost using only the fixed entries g. A checkpoint set attaining $V_\Lambda(g,B)$ is called *tight*. We abbreviate $Q_\Lambda(T;g,B)$ to $Q_\Lambda(T)$ when g,B are fixed.

To see why the cost includes a terminal factor, fix an endpoint checkpoint t. The score contributed by checkpoints before and including t is B_t/Q_{a+g_t}, with the cost evaluated on the positions before t. Thus optimizing additional checkpoints before this endpoint amounts to minimizing that cost.

For the original stepsizes, we add the incoming sum w to the first entry, using the list $g_1+w,B_1,g_2,B_2,\dots,g_r,B_r$. Each selected step contributes its fixed part 8 to the preceding sum, so $\widetilde{s}_{t_i}=s_i+8$ in the notation of (17). This gives $P(T)=P_w(T;H)$.

As in [YL26b, Lemma 3.3], we decompose the minimum cost at a selected checkpoint. The entry g_t immediately before B_t stays on the left, so the constant in the left subproblem is $a+g_t$. This is why we allow any $\Lambda\ge a$.

Lemma B.1. *For every $g,B\in[0,\infty)^r$, $r\ge 1$, and $\Lambda\ge a$,*

$$V_\Lambda(g,B)=\min\left\{c\left(\Lambda+\sum_{j=1}^{r}(g_j+B_j)\right),\ \min_{\substack{1\le t\le r\\ B_t>0}}\frac{V_{a+g_t}(g_{1:t-1},B_{1:t-1})V_\Lambda(g_{t+1:r},B_{t+1:r})}{B_t}\right\}.\tag{43}$$

Here $g_{u:v}:=(g_u,\dots,g_v)$ and likewise for $B_{u:v}$. For empty vectors, $V_\mu(\varnothing,\varnothing)=U_\mu(\varnothing)=c\mu$. The inner minimum is $+\infty$ if no B_t is positive.

Proof. For a checkpoint set T containing t, let T_L,T_R denote its selected indices before and after t, reindexed within the two shorter lists. Then

$$Q_\Lambda(T;g,B)=\frac{Q_{a+g_t}(T_L;g_{1:t-1},B_{1:t-1})Q_\Lambda(T_R;g_{t+1:r},B_{t+1:r})}{B_t}.$$

Minimizing over T_L,T_R and t accounts for every nonempty checkpoint set. Including the empty set gives (43). □

To bound $U_\Lambda(g)$ using this recursion, we need a maximizing B at which both the empty set and a singleton are tight. We first show that $\log Q_\Lambda(T)$ is submodular in the checkpoint set T, following [YL26b, Lemma B.1].

Lemma B.2 (Log-submodularity)**.** *For any two admissible checkpoint sets S,T,*

$$Q_\Lambda(S)Q_\Lambda(T)\ge Q_\Lambda(S\cap T)Q_\Lambda(S\cup T).\tag{44}$$

If S and T are disjoint and both nonempty, the inequality is strict. Consequently, tight sets are closed under unions and intersections, and two nonempty tight sets cannot be disjoint.

Proof. Inserting a checkpoint. Insert a checkpoint of size $B > 0$ into a gap. Let $\ell \geq 0$ be the sum of entries before it in this gap, and let K be the sum after it plus the constant a if the gap ends at an existing checkpoint, or Λ if it is the final gap. Thus $K \geq a$. Before insertion, this gap contributes $c(\ell + B + K)$ to Q_Λ. After insertion, it contributes $c(a+\ell)cK/B$, so the cost is multiplied by

$$\frac{c(a+\ell)K}{B(\ell+B+K)}.$$

Its logarithmic derivatives with respect to ℓ and K are $1/(a+\ell) - 1/(\ell+B+K)$ and $1/K - 1/(\ell+B+K)$, respectively. Both are positive because $B > 0$ and $K \geq a$, so the ratio is strictly increasing in both variables.

Comparing checkpoint sets. Insert the checkpoints in $S \setminus T$, one at a time in the same order, into $S \cap T$ and into T. At each insertion, the set built from T contains the set built from $S \cap T$. Its gap containing the new checkpoint therefore starts no earlier and ends no later, so its values of ℓ and K are no larger. If this gap ends at a checkpoint while the corresponding gap for the smaller set is final, the constant in K also decreases from Λ to $a \leq \Lambda$. Multiplying the insertion ratios gives

$$\frac{Q_\Lambda(S)}{Q_\Lambda(S\cap T)} \geq \frac{Q_\Lambda(S\cup T)}{Q_\Lambda(T)},$$

which proves (44). If S and T are disjoint and nonempty, the first insertion compares the empty set with T. Every selected entry in T is positive, so at least one of ℓ, K is strictly smaller when inserting into T. The first comparison, and hence the final inequality, is strict.

Finally, if S, T are tight and $V = V_\Lambda(g,B)$, then

$$V^2 \geq Q_\Lambda(S\cap T)Q_\Lambda(S\cup T) \geq V^2.$$

Each factor in the middle is at least $V > 0$, so both equal V. Thus $S \cap T$ and $S \cup T$ are tight. If S, T were also nonempty and disjoint, the first inequality would be strict, a contradiction. □

We now prove the analogue of [YL26b, Lemma 3.4], keeping the fixed entries g unchanged.

Lemma B.3. *For fixed $g \in [0,\infty)^r$, $r \geq 1$, and $\Lambda \geq a$, the supremum defining $U_\Lambda(g)$ is finite and attained. At every maximizing B, the empty set and a singleton are tight.*

Proof.

Step 1. Finiteness of $U_\Lambda(g)$. We induct on the length r of g, simultaneously for all $g \geq 0$ and $\Lambda \geq a$. The case $r = 0$ is $U_\Lambda(\varnothing) = c\Lambda$. For an arbitrary B, let $B_{\max} = \max_j B_j$. If $B_{\max} \leq 1$, the empty set gives

$$V_\Lambda(g,B) \leq c\left(\Lambda + \sum_j g_j + r\right).$$

Otherwise, choose j with $B_j = B_{\max}$ and split at j in (43). Using the empty set on the left gives

$$V_\Lambda(g,B) \leq \frac{c\left(a + g_j + \sum_{i<j}(g_i + B_i)\right)}{B_j} U_\Lambda(g_{j+1:r}) \leq c\left(r + a + \sum_i g_i\right) U_\Lambda(g_{j+1:r}).$$

The finitely many possible suffixes have finite U_Λ by induction, so these bounds are uniform over B.

Step 2. Existence of a maximizer. For fixed g, the function $V_\Lambda(g,B)$ is continuous at every finite $B \geq 0$. Indeed, costs selecting an entry that tends to zero diverge, since all numerator factors stay positive. Such sets cannot minimize the cost nearby because the empty-set cost stays bounded. The remaining costs are continuous, and their finite minimum is therefore continuous.

Choose $B^{(k)}$ with $V_\Lambda(g,B^{(k)}) \to U_\Lambda(g)$. If this sequence were unbounded, pass to a subsequence along which each coordinate has a limit in $[0,\infty]$ and at least one tends to infinity. Let j be the first such coordinate, so the preceding entries stay bounded. Splitting at j and using the empty set on the left yields

$$V_\Lambda(g,B^{(k)}) \leq \frac{c\left(a+g_j+\sum_{i<j}(g_i+B_i^{(k)})\right)}{B_j^{(k)}} U_\Lambda(g_{j+1:r}) \longrightarrow 0.$$

This contradicts $U_\Lambda(g) \geq V_\Lambda(g,0) = c(\Lambda+\sum_j g_j) > 0$. Thus a subsequence converges to a finite B, and continuity gives $V_\Lambda(g,B) = U_\Lambda(g)$.

Step 3. Tightness of the empty set. Fix a maximizing B and write $V = V_\Lambda(g,B)$. If a perturbation strictly increased every tight cost, all other costs would remain above V for sufficiently small perturbations, by Step 2. Since there are finitely many checkpoint sets, their new minimum would exceed $U_\Lambda(g)$, a contradiction.

Suppose that the empty set were not tight. By Lemma B.2, the intersection of all tight sets is tight and hence nonempty. Choose j in this intersection and slightly decrease B_j, keeping it positive. Every tight set selects j, so its cost depends on B_j only through $1/B_j$ and strictly increases. The observation above gives a contradiction. Thus the empty set is tight.

Step 4. Existence of a tight singleton. There must also be a nonempty tight set. Otherwise, increasing any B_j slightly would increase the only tight cost, that of the empty set, contradicting Step 3.

Let T_* be the intersection of all nonempty tight sets. By Lemma B.2, T_* is nonempty and tight. Suppose it contains distinct indices i, j, and put $u = B_i$, $v = B_j$, with $u \leq v$. If $u < v$, choose $1 < \theta < v/u$ and replace (u,v) by $(u-t, v+\theta t)$ for sufficiently small $t > 0$. If $u = v$, replace them by $(u-t, u+t+t^2/(2u))$. In both cases the sum increases and the product decreases. The empty-set cost therefore increases. Every nonempty tight set selects both entries, so its cost depends on them only through $1/(uv)$ and also increases. This again contradicts Step 3. Hence T_* is a tight singleton.

□

Splitting at this singleton gives two shorter lists whose empty sets are both tight. As in [YL26b, Lemma 3.5 and Appendix B.3], we apply induction to these lists to bound $U_\Lambda(g)$. The induction step uses the following scalar inequality, proved as in [YL26b, Lemma B.2]. The case $c = 2$ also follows by rescaling [JCY26, Lemma C.4].

Lemma B.4. *If $x, y \geq a$, $B > 0$, and*

$$W = x+y+B-a, \qquad WB = cxy,$$

then $W^{1/p} \leq x^{1/p} + y^{1/p}$.

Proof. Set $\overline{W} = (x^{1/p}+y^{1/p})^p$. Our goal is to show $W \leq \overline{W}$. Define $\psi(X) = X^2 - (x+y)X - cxy$. Eliminating B gives

$$\psi(W) = -aW < 0.$$

Since ψ has exactly one positive root and $W, \overline{W} > 0$, it suffices to prove $\psi(\overline{W}) \geq 0$. Set $z = (x/\overline{W})^{1/p}$, so $1 - z = (y/\overline{W})^{1/p}$ and $z \in (0,1)$. Then

$$\frac{\psi(\overline{W})}{\overline{W}^2} = 1 - f(z), \qquad f(z) := z^p + (1-z)^p + c[z(1-z)]^p.$$

Thus it remains to show $f(z) \leq 1$ on $[0,1]$. The choice $p = \log_2(1+\sqrt{1+c})$ gives $c = 2^{2p} - 2^{p+1}$, so $f(0) = f(1/2) = 1$. On $(0,1/2)$,

$$f'(z) = p[z(1-z)]^{p-1} g(z), \qquad g(z) = (1-z)^{1-p} - z^{1-p} + c(1-2z).$$

Moreover,

$$g''(z) = p(p-1)[(1-z)^{-p-1} - z^{-p-1}] < 0, \qquad g(0+) = -\infty, \qquad g(1/2) = 0.$$

Concavity implies that g either stays negative or changes sign once from negative to positive. Thus f either decreases or first decreases and then increases, so its maximum on $[0,1/2]$ is at an endpoint. By symmetry, $f(z) \leq 1$ on $[0,1]$, so $\psi(\overline{W}) \geq 0$ and $W \leq \overline{W}$, as required. □

We now use this inequality to bound $U_\Lambda(g)$ by induction.

Lemma B.5. *For every $g \in [0,\infty)^r$ and $\Lambda \geq a$,*

$$\left[\frac{U_\Lambda(g)}{c}\right]^{1/p} \leq \Lambda^{1/p} + \sum_{j=1}^{r} (a + g_j)^{1/p}.$$

Proof. We induct on r, simultaneously for all $g \geq 0$ and $\Lambda \geq a$. The case $r = 0$ is $U_\Lambda(\varnothing) = c\Lambda$. For $r \geq 1$, choose a maximizing B. By Lemma B.3, the empty set and a singleton $\{t_*\}$ are tight. Set

$$W := U_\Lambda(g)/c, \qquad x := a + g_{t_*} + \sum_{j<t_*} (g_j + B_j), \qquad y := \Lambda + \sum_{j>t_*} (g_j + B_j).$$

Then $x, y \geq a$ and $B_{t_*} > 0$. Tightness of the empty set and singleton gives

$$W = x + y + B_{t_*} - a, \qquad W B_{t_*} = cxy.$$

Thus Lemma B.4 gives $W^{1/p} \leq x^{1/p} + y^{1/p}$.

The recursive decomposition (43) and the empty-set costs on the two sides give

$$cW \leq \frac{V_{a+g_{t_*}}(g_{1:t_*-1}, B_{1:t_*-1}) V_\Lambda(g_{t_*+1:r}, B_{t_*+1:r})}{B_{t_*}} \leq \frac{c^2 xy}{B_{t_*}} = cW.$$

Both numerator factors therefore equal their empty-set bounds cx and cy, respectively. Applying the induction hypothesis to the two shorter lists yields

$$x^{1/p} \leq (a + g_{t_*})^{1/p} + \sum_{j<t_*} (a + g_j)^{1/p},$$
$$y^{1/p} \leq \Lambda^{1/p} + \sum_{j>t_*} (a + g_j)^{1/p}.$$

Adding these bounds proves the claim.

□

B.2 Bounding the sum of entries when the maximum checkpoint score is one

Since $V_\Lambda(g,B) \le U_\Lambda(g)$, Lemma B.5 bounds the minimum cost for every B. We use it to bound $\sum_j (g_j + B_j)$ when $\max_T P(T) = 1$. For the original stepsizes, this gives part (i) of Lemma 4.1 and the bound on s_{tail} in part (ii). First we bound the number of entries B_j above each level $q > 0$, then integrate this bound to obtain their sum. For this subsection, write $M = \sum_{j=1}^{r} (a + g_j)^{1/p}$ for the sum over the fixed entries in Lemma B.5.

Lemma B.6 (Level-set bound)**.** *If* $\max_T P(T) = 1$*, then for every* $q > 0$*,*

$$\#\{j : B_j > q\} \le c^{1/p} M q^{-1/p}.$$

Proof. Fix $q > 0$ and list the positions with $B_j > q$ as $j_1 < \cdots < j_k$. The claim is immediate if $k = 0$. Otherwise set $j_0 = 0$ and

$$M_i = \sum_{t=j_{i-1}+1}^{j_i} (a + g_t)^{1/p} \quad (1 \le i \le k), \qquad \sum_{i=1}^{k} M_i \le M.$$

Select every j_i. Between j_{i-1} and j_i, choose additional checkpoints to minimize the cost with terminal constant $a + g_{j_i}$. By Lemma B.5, this minimum is at most cM_i^p, so this part of the score is at least $B_{j_i}/(cM_i^p)$. Multiplying these factors and using the arithmetic–geometric mean inequality gives

$$1 = \max_T P(T) \ge \prod_{i=1}^{k} \frac{q}{cM_i^p} \ge \left(\frac{q}{c}\right)^k \left(\frac{k}{M}\right)^{pk}.$$

Taking kth roots yields $k \le c^{1/p} M q^{-1/p}$, as required. □

We now convert this counting estimate into a bound on the sum of entries.

Lemma B.7. *If* $\max_T P(T) = 1$*, then*

$$\sum_j (g_j + B_j) \le \left(1 + \frac{cp}{p-1}\right) M^p.$$

Proof. If the list is empty, both sides of the claimed bound are zero. Otherwise, Lemma B.6 gives

$$N(q) := \#\{j : B_j > q\} \le c^{1/p} M q^{-1/p}.$$

If $\max_j B_j > 0$, take $0 < q < \max_j B_j$. Then $N(q) \ge 1$, and hence

$$1 \le c^{1/p} M q^{-1/p}.$$

Letting $q \uparrow \max_j B_j$ gives

$$\max_j B_j \le cM^p.$$

This bound also holds when all $B_j = 0$. Therefore,

$$\sum_j B_j = \int_0^{cM^p} N(q)\,\mathrm{d}q \le c^{1/p} M \int_0^{cM^p} q^{-1/p}\,\mathrm{d}q = \frac{cp}{p-1} M^p.$$

It remains to control the sum of g_j. Since $p > 1$, subadditivity gives

$$\left(\sum_j g_j\right)^{1/p} \le \sum_j g_j^{1/p} \le \sum_j (a + g_j)^{1/p} = M.$$

Combining the two estimates gives

$$\sum_j (g_j + B_j) \le M^p + \frac{cp}{p-1} M^p = \left(1 + \frac{cp}{p-1}\right) M^p.$$

□

B.3 Deriving the checkpoint bounds

We now return to the original stepsizes and the checkpoint score $P_w(T;H)$ to prove Lemma 4.1.

Proof of Lemma 4.1. Let $h_1,\dots,h_r$ be the stepsizes in the current block, where $r \le m$, and let w be the sum of stepsizes in the incoming tail from preceding blocks. Take $g_j = \min\{h_j, 8\}$ and $B_j = (h_j - 8)_+$. As explained in Section B.1, the list $g_1 + w, B_1, g_2, B_2, \dots, g_r, B_r$ has score $P(T) = P_w(T;H)$.

For this list, write

$$M = (a + g_1 + w)^{1/p} + \sum_{j=2}^{r} (a + g_j)^{1/p}.$$

Since $p > 1$, subadditivity of $t^{1/p}$ gives

$$(a + g_1 + w)^{1/p} \le w^{1/p} + (a + g_1)^{1/p}.$$

Since $g_j \le 8$ for every j, it follows that

$$M \le w^{1/p} + r(a+8)^{1/p} \le w^{1/p} + 2^{1/p} r a^{1/p}.$$

Using $(u+v)^p \le 2^{p-1}(u^p + v^p)$, we obtain

$$M^p \le C\big(w + a r^p\big) \le C\big(w + a m^p\big), \tag{45}$$

where C is an absolute constant as the range of p is bounded.

Proof of (i). If the empty set maximizes P_w, its maximum is one. Applying Lemma B.7 to this list and using (45) gives

$$w + \sum_{j=1}^{r} h_j \le \left(1 + \frac{cp}{p-1}\right) M^p \le C_b(w + a m^p).$$

The constant is uniform because c and $\frac{p}{p-1}$ are bounded.

Proof of (ii). Now let $T = \{t_1 < \cdots < t_k\}$ be a nonempty maximizing checkpoint set, and put $t_0 = 0$. Group the positions from $t_{i-1} + 1$ through t_i, including the selected endpoint, and define

$$M_1 = (a + g_1 + w)^{1/p} + \sum_{j=2}^{t_1} (a + g_j)^{1/p}, \qquad M_i = \sum_{j=t_{i-1}+1}^{t_i} (a + g_j)^{1/p} \quad (2 \le i \le k).$$

We claim that

$$a + s_i + 8 \le M_i^p. \tag{46}$$

Keep t_i fixed. Inserting checkpoints before it cannot increase the score, so the empty set minimizes the cost of this gap. Its cost is

$$c(a + s_i + 8).$$

If $i = 1$ and $t_1 = 1$, the incoming sum w is attached to the selected endpoint itself, and $a + 8 + w = M_1^p$ directly. Otherwise, apply Lemma B.5 to the unselected positions before t_i, with terminal constant $\Lambda = a + g_{t_i} = a + 8$. The term $\Lambda^{1/p}$ is exactly the endpoint contribution in M_i. In the first gap, use $g_1 + w$ as the first fixed entry. This gives (46) in every case. The groups defining the M_i's are disjoint, so

$$\sum_{i=1}^{k} M_i \le M.$$

Since $p \geq 1$,

$$c\sum_i (a+s_i+8) \leq c\sum_i M_i^p \leq c\left(\sum_i M_i\right)^p \leq cM^p \leq C_b(w+ar^p).$$

For the tail after t_k, the incoming sum is zero. Any checkpoint set in this tail with score greater than one could be appended to T to increase its score. Hence the maximum score in the tail is one, and Lemma B.7 gives

$$s_{\text{tail}} \leq \left(1+\frac{cp}{p-1}\right)\left[\sum_{j=t_k+1}^{r}(a+g_j)^{1/p}\right]^p \leq C_b a m^p.$$

Here we used $g_j \leq 8$, $a \geq 8$, and $r \leq m$. The same bound holds for an empty tail.

Proof of (iii). For consecutive checkpoints $t_i < t_{i+1}$, write $x = a+s_i+8$, $y = a+s_{i+1}+8$, and $B = b_i - 8$. Delete the checkpoint at t_i, while retaining all the other checkpoints. The two gaps adjacent to t_i then merge, and the step at t_i, whose full stepsize is $h_{t_i} = B+8$, becomes part of the merged gap. The denominator at the next checkpoint is therefore

$$c\big(a+s_i+(B+8)+s_{i+1}+8\big) = c(x+y+B-a).$$

All other factors remain unchanged. Comparing the score before and after deletion gives

$$\frac{B}{cx}\frac{b_{i+1}-8}{cy} \geq \frac{b_{i+1}-8}{c(x+y+B-a)}.$$

Cancelling the positive common factors gives

$$B(x+y+B-a) \geq cxy. \tag{47}$$

We next prove

$$\frac{xy}{B} \leq 2(x+y). \tag{48}$$

If $B \leq x+y$, then (47) gives

$$\frac{xy}{B} \leq \frac{x+y+B-a}{c} \leq x+y+B \leq 2(x+y),$$

where we used $c \geq 1$ and $a \geq 0$. If $B > x+y$, then by the arithmetic–geometric mean inequality,

$$\frac{xy}{B} < \frac{xy}{x+y} \leq \frac{x+y}{4} \leq 2(x+y).$$

Dividing (48) by xy gives part (iii).

Proof of (iv). Deleting the final checkpoint cannot increase the score. Since

$$P_w(T;H) = P_w(T\setminus\{t_k\};H)\frac{b_k-8}{c(a+s_k+8)},$$

the last factor is at least one, proving part (iv). □

C Proof of Lemma 4.2

We first prove the auxiliary lemma used to justify the second case of the repair rule in Section 4.2.

Lemma C.1. *Let $h_1,\dots,h_r \ge 0$, where $r \ge 1$, and let $S' := \sum_{t=1}^{r} h_t > 0$. At least one of the following alternatives holds:*

1. *There exist $1 \le i < j \le r$ such that*

$$h_i, h_j \ge \frac{S'}{8r^2}, \qquad \sum_{t=i+1}^{j-1} h_t \le \min\{h_i, h_j\}.$$

2. *There exists $1 \le i \le r$ such that $h_i \ge 3S'/4$.*

Proof of Lemma C.1.

Suppose the first alternative fails and set $\tau = S'/(8r^2)$. Then

$$\sum_{t=i+1}^{j-1} h_t > \min\{h_i, h_j\} \qquad \text{whenever } i < j \text{ and } h_i, h_j \ge \tau. \tag{49}$$

We will show that the sum of all terms other than the largest is at most $S'/4$.

Two inductive estimates. Consider a nonnegative sequence $a_1,\dots,a_k$ satisfying (49) with a_i in place of h_i. We prove the following two estimates together by induction on k:

(i) If $a_i < \sum_{t<i} a_t$ whenever $a_i \ge \tau$, then $\sum_{i=1}^{k} a_i \le 2\tau k^2$.

(ii) If, in addition, $a_i < \sum_{t>i} a_t$ whenever $a_i \ge \tau$, then $\sum_{i=1}^{k} a_i \le \tau k^2$.

The stronger estimate (ii) is needed to apply the induction to the terms before the largest element.

Both estimates hold for $k = 0$, and for $k \ge 1$ when every $a_i < \tau$, since then $\sum_i a_i < k\tau \le \tau k^2$. Otherwise, choose $a_z = \max_i a_i \ge \tau$ and write $S_L := \sum_{i<z} a_i$ and $S_R := \sum_{i>z} a_i$. Applying (49) between a_z and each $a_i \ge \tau$ gives

$$a_i < \sum_{t=i+1}^{z-1} a_t \quad (i < z), \qquad a_i < \sum_{t=z+1}^{i-1} a_t \quad (i > z).$$

Under either hypothesis, the left subsequence inherits the bound by preceding terms from the full sequence, so it satisfies both conditions in (ii). The right subsequence satisfies the condition in (i). If the full sequence satisfies (ii), the right subsequence also inherits the bound by following terms. Both subsequences retain (49).

Under the hypothesis of (i), we have $a_z < S_L$. Applying (ii) to the left subsequence and (i) to the right gives

$$\sum_{i=1}^{k} a_i < 2S_L + S_R \le 2\tau\big((z-1)^2 + (k-z)^2\big) \le 2\tau k^2.$$

Under the hypothesis of (ii), we have $a_z < \min\{S_L, S_R\}$, and both subsequences satisfy (ii). Thus

$$\begin{aligned}\sum_{i=1}^{k} a_i &< S_L + \min\{S_L, S_R\} + S_R \\ &\le \tau\big((z-1)^2 + \min\{(z-1)^2, (k-z)^2\} + (k-z)^2\big) \\ &\le \tau(k-1)^2 \le \tau k^2,\end{aligned}$$

where the last line uses $\min\{(z-1)^2, (k-z)^2\} \le 2(z-1)(k-z)$. This proves both estimates.

Applying the estimates. Choose $h_z = \max_t h_t \geq S'/r > \tau$. By (49), the reversed left subsequence $(h_{z-1}, \dots, h_1)$ and the right subsequence $(h_{z+1}, \dots, h_r)$ satisfy the hypothesis of (i). Applying that estimate gives

$$S' - h_z \leq 2\tau\big((z-1)^2 + (r-z)^2\big) \leq 2\tau r^2 = \frac{S'}{4}.$$

Hence $h_z \geq 3S'/4$, as required by the second alternative. □

We now prove Lemma 4.2, first verifying the repair rule and then establishing its guarantees.

Proof of Lemma 4.2.

Fix $C_0 \geq 1$ and $R = 32C_0$, and choose

$$\xi_0 = \frac{1}{8(R+1)}, \qquad c_0 = \frac{1}{[4096(1+C_0)]^2}.$$

Let the data satisfy the hypotheses of the lemma. Then

$$\frac{S}{\kappa} \leq \frac{C_0 S_0}{\kappa} \leq \xi_0 < \frac{1}{8}.$$

For candidate checkpoint indices chosen by the rule, set $t_0 = 0$, write $b_i := h_{t_i}$, and define

$$s_i := \sum_{t=t_{i-1}+1}^{t_i-1} h_t, \qquad \chi_i := \prod_{t=t_{i-1}+1}^{t_i-1} (1 - h_t/\kappa) \qquad (1 \leq i \leq q). \tag{50}$$

Thus every repair gap satisfies $s_i \leq S < \kappa$, and every stepsize lies in $[0, \kappa)$.

Step 1. Verifying the repair rule. Define

$$j_* := \max\left\{ j \in \{1, \dots, r\} : \sum_{t=j}^{r} h_t > S_0/8 \right\}.$$

This index exists because $S \geq S_0$. Set $S' := \sum_{t=j_*}^{r} h_t$. By maximality of j_*,

$$S' > \frac{S_0}{8}, \qquad \sum_{t=j_*+1}^{r} h_t \leq \frac{S_0}{8}. \tag{51}$$

The set $\{j_*, \dots, r\}$ is the shortest suffix used in the repair rule. If it contains a stepsize at least $S_0/16$, the rule selects $q = 1$ and gives $b_1 \geq S_0/16$. Otherwise, every stepsize in this suffix is less than $S_0/16$. Apply Lemma C.1 to this list. Its second alternative is impossible, since it would give a stepsize at least $3S'/4 > 3S_0/32 > S_0/16$. Since the list has at most r entries, the first alternative supplies two stepsizes at least $S'/(8r^2) > S_0/(64r^2)$, with the intervening stepsizes summing to at most the smaller of the two. Thus the repair rule can always be carried out.

Fix any checkpoint selection permitted by the rule. If $q = 2$, then

$$b_1, b_2 \geq \frac{S_0}{64r^2}, \qquad s_2 = \sum_{t=t_1+1}^{t_2-1} h_t \leq \min\{b_1, b_2\}. \tag{52}$$

In both cases, the selected stepsizes are positive and all selected checkpoints belong to this suffix. Moreover, $t_q \geq j_*$ and (51) give

$$\sum_{t=t_q+1}^{r} h_t \leq \frac{S_0}{8} \leq \frac{S_0}{4},$$

which proves (iii).

Step 2. Positivity and bounds on ℓ_{i+1}/D_{i+1}. With these checkpoint indices fixed, let $\ell_1 \ge 0$ and $D_1 > 0$ be arbitrary with $\ell_1/D_1 \le R$. We first check that the components can be constructed, then bound the final relative threshold. For s_i and χ_i from (50), the estimate (10) gives

$$\chi_i \ge 1 - \frac{s_i}{\kappa} \ge 1 - \frac{S}{\kappa} \ge \frac{1}{2}, \qquad 0 \le \kappa(1-\chi_i) \le s_i \qquad (1 \le i \le q).$$

We construct the components successively from (ℓ_1, D_1). Whenever $\ell_i \ge 0$, $D_i > 0$, and $\eta_i > 0$, Lemma 3.2 applies at t_i and gives $\ell_{i+1} \ge 0$, $D_{i+1} > 0$, and

$$\begin{aligned} \frac{D_{i+1}}{D_i} &= \frac{(1-1/\kappa) b_i \chi_i \eta_i}{2[2+(\kappa-1)(1-\chi_i)]}, \\ \frac{\ell_{i+1}}{D_{i+1}} &= \frac{\kappa(1-\chi_i)}{b_i \chi_i} \le \frac{2s_i}{b_i}. \end{aligned} \tag{53}$$

For the first checkpoint, the initial bound $\ell_1/D_1 \le R$ implies

$$\eta_1 = 1 - (1-\chi_1)\left(1 + \frac{\ell_1}{D_1}\right) \ge 1 - (R+1)\frac{S}{\kappa} \ge \frac{7}{8}.$$

Thus $D_2 > 0$, and (53) applies to the first transfer.

If $q = 2$, the relative threshold ℓ_2/D_2 may be large, but $s_2 \le b_1$ in (52) controls the contraction loss in the second gap. Using also $s_1 + s_2 \le S$, we obtain

$$\begin{aligned} 1 - \eta_2 &= (1-\chi_2)\left(1 + \frac{\ell_2}{D_2}\right) \le \frac{s_2}{\kappa}\left(1 + \frac{2s_1}{b_1}\right) \\ &\le \frac{s_2 + 2s_1}{\kappa} \le \frac{2S}{\kappa} \le \frac{1}{4}. \end{aligned}$$

Hence $\eta_2 \ge 3/4$ and $D_3 > 0$. All the repair components can therefore be constructed, with $\eta_i \ge 1/2$ for every $1 \le i \le q$.

Finally, (53) and the checkpoint choices give

$$\begin{aligned} \frac{\ell_2}{D_2} &\le \frac{2s_1}{b_1} \le \frac{2C_0 S_0}{S_0/16} = R \quad \text{if } q = 1, \\ \frac{\ell_3}{D_3} &\le \frac{2s_2}{b_2} \le 2 \le R \qquad \text{if } q = 2, \end{aligned}$$

where we used the bound $s_2 \le b_2$ in (52). This proves (ii).

Step 3. Bounding D_{q+1}/D_1 from below. Since $\kappa \ge 4$, $\chi_i \ge 1/2$, and $\eta_i \ge 1/2$, the first identity in (53) gives

$$\frac{D_{i+1}}{D_i} \ge \frac{3b_i}{64(1+S)} \ge \frac{b_i}{64(1+C_0)S_0} \qquad (1 \le i \le q). \tag{54}$$

We used $2 + (\kappa-1)(1-\chi_i) \le 2 + s_i \le 2(1+S)$, followed by $S \le C_0 S_0$ and $S_0 \ge 1$.

If $q = 1$, then $b_1 \ge S_0/16$, so

$$\frac{D_2}{D_1} \ge \frac{1}{1024(1+C_0)} \ge c_0 (r+1)^{-4}.$$

If $q = 2$, then (52) and (54) give

$$\frac{D_3}{D_1} = \frac{D_2}{D_1}\frac{D_3}{D_2} \ge \frac{1}{[4096(1+C_0)r^2]^2} = c_0 r^{-4} \ge c_0 (r+1)^{-4}.$$

Together with the positivity of η_i and D_{i+1} established in Step 2, these bounds prove (i). □