Silver Rate Is (Almost) Optimal for Gradient Descent

Yuhan Ye and Kaizhao Liu

Research blog version of Silver Rate Is (Almost) Optimal for Gradient Descent (arXiv:2609.09152). Updated November 4, 2026. In a follow-up work [LY26], we also prove a lower bound matching the silver exponent for smooth strongly convex objectives.

Abstract

Gradient descent (GD) is one of the most fundamental algorithms in optimization. Yet even for smooth convex objectives, the optimal rate among predetermined nonnegative stepsize schedules has remained unresolved. Altschuler and Parrilo showed that their recursive *silver stepsize schedule* achieves an $O(n^{-p_{\mathrm{sil}}})$ rate, where $p_{\mathrm{sil}} = \log_2(1+\sqrt{2}) \approx 1.2716$, improving upon the classical $O(n^{-1})$ rate of constant-step GD. Although the optimality of p_{sil} has been well supported by numerical evidence for small n, a matching lower bound has remained missing.

In this paper, we prove an $\Omega(n^{-p_{\mathrm{sil}}-o(1)})$ lower bound for GD, showing that p_{sil} is the optimal convergence exponent. Additionally, we prove an *anytime* lower bound that matches the exponent $2p_{\mathrm{sil}}/(1+p_{\mathrm{sil}}) \approx 1.1195$ of the known upper bound [ZLDC25].

1. Introduction

We consider the minimization of a smooth convex function $f : \mathbb{R}^d \to \mathbb{R}$. A standard algorithm for this problem is **gradient descent** (GD), with a history dating back to Cauchy [Cau47]. Given a *stepsize schedule* $H = (h_1, \dots, h_n) \in [0,\infty)^n$ fixed before the optimization begins, GD proceeds via the iteration

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

In this paper, we study the limits of this method's convergence rate. To formalize this question, let $\mathcal{F}_L(\mathbb{R}^d)$ be the class of convex L-smooth functions on $\mathbb{R}^d$ with a nonempty set of minimizers. We measure the convergence rate achieved by a schedule H using its worst-case (normalized) function suboptimality

$$\mathcal{R}_n(H) := \sup_{d\in\mathbb{N}} \sup_{f\in\mathcal{F}_L(\mathbb{R}^d)} \sup_{x_*\in\arg\min f} \sup_{x_0\in\mathbb{R}^d\setminus\{x_*\}} \frac{f(x_n)-f(x_*)}{\frac{L}{2}\|x_0-x_*\|^2}. \tag{1}$$

Classical Results.

Classical convergence analyses of gradient methods can be traced to [LP66]. With the constant stepsize of $1/L$, GD achieves the classical $O(n^{-1})$ rate; see the textbooks [Pol87], [Nes04], [Bub15]. This rate is tight, even in one dimension [DT14].

To obtain faster rates, classical acceleration methods introduce momentum or auxiliary sequences, as in Polyak's heavy-ball method [Pol64] and Nesterov's accelerated method [Nes83]. The latter attains an $O(n^{-2})$ rate for smooth convex objectives, matching the classical $\Omega(n^{-2})$ lower bound for deterministic first-order methods established by Nemirovsky and Yudin [NY79].[1] These accelerated methods, however, modify the original GD update. For GD with predetermined stepsizes, the classical results therefore leave a gap between the $O(n^{-1})$ upper bound and the $\Omega(n^{-2})$ lower bound.

Breakthrough in Upper Bounds: Acceleration via Silver Stepsize Schedule.

For many years, the $O(n^{-1})$ rate remained the best general upper bound known for GD, leaving open the question of whether stepsize choice alone could improve it. In recent years, computer-assisted analyses have opened up new opportunities for acceleration by helping identify better stepsize schedules. Drori and Teboulle [DT14] introduced *performance estimation problems* (PEPs), which express a method's worst-case error after a prescribed number of iterations as an optimization problem. *Exact interpolation conditions* characterize when prescribed function values and gradients at finitely many points come from a single function in the specified class [THG17]. These conditions give exact semidefinite formulations of PEPs for *fixed-step methods*.[2] Using PEPs to analyze and optimize GD stepsize schedules over short horizons, Altschuler showed in his thesis [Alt18, Chapter 8] that coordinating short and long steps can improve convergence in the strongly convex setting. Such gains require a joint analysis of multiple steps, since separate bounds on individual steps fail to reflect this advantage.

Altschuler and Parrilo [AP25a] introduced the recursive *silver stepsize schedule* for smooth strongly convex objectives. Their schedule reduces the squared distance to the minimizer by a factor of ε in $O(\kappa^{1/p_{\text{sil}}}\log(1/\varepsilon))$ iterations, where κ is the condition number and $p_{\text{sil}} := \log_2(1+\sqrt{2}) \approx 1.2716$.[3] They later extended their silver schedule to general smooth convex objectives, obtaining an $O(n^{-p_{\text{sil}}})$ rate [AP25b]. In the same work, they conjectured that p_{sil} is the optimal polynomial convergence exponent, which is supported by numerical solutions at small horizons [DGVPR24]. Subsequent work improved the constants in the bound on function suboptimality and established accelerated gradient-norm guarantees [GSW25a]. Further work also gave rules for combining shorter schedules to obtain the silver rate at arbitrary prescribed horizons [GSW25b], [ZJ26]. For a broader introduction to stepsize-based acceleration, see the recent survey [AP26].

The smooth convex silver-rate results above concern the *non-anytime (finite-horizon) setting*, in which the schedule may be chosen separately for each fixed horizon n. In the *anytime setting*, one infinite schedule $H = (h_t)_{t\geq 1}$ is used across all horizons, with $H_n = (h_1, \ldots, h_n)$. By combining silver-schedule blocks, [ZLDC25] construct such a schedule with an $O(n^{-p_{\text{any}}})$

guarantee at every stopping time, where $p_{\mathrm{any}} := 2p_{\mathrm{sil}}/(1+p_{\mathrm{sil}}) \approx 1.1195$. This resolves the open problem [KS24] of whether stepsize choice alone can accelerate GD uniformly over all stopping times.

Recent Progress on Lower Bounds.

Despite these improved upper bounds, the best-known lower bound for GD until recently was the classical $\Omega(n^{-2})$ bound [NY79]. Nesterov [Nes04] provides a simple *quadratic zero-chain* construction for this lower bound. By initializing at the origin, a method satisfying the linear-span assumption can reach at most one new coordinate per iteration, while the minimizer has nonzero entries farther along the chain. Even exact minimization over the reached coordinates leaves an $\Omega(n^{-2})$ error. Using PEP, [KF16] developed the optimized gradient method (OGM), whose bound on function suboptimality was later proved optimal in both rate and constant [Dro17]. The lower bound construction in [Dro17] combines smooth convex interpolation with a zero-chain property. These results sharpened the constants of the classical lower bound, but did not improve the convergence exponent for GD with predetermined stepsizes.

In recent months, a sequence of results brought the lower bounds closer to the silver-rate upper bounds $O(n^{-p_{\mathrm{sil}}})$ and $O(n^{-p_{\mathrm{any}}})$ in the non-anytime and anytime settings, respectively. [MC26] proved an $\Omega(n^{-1.932})$ non-anytime lower bound, and a subsequent blog post [Tsa26] sharpened it to $\Omega(n^{-\sqrt{3}})$. For the anytime setting, [TFZH26] proved an $\Omega(n^{-4/3})$ lower bound.[4] In our previous work [YL26], we proved an $\Omega(n^{-1.6342})$ non-anytime lower bound and an $\Omega(n^{-1.2408})$ anytime lower bound for arbitrary real-valued stepsizes. To the best of our knowledge, these remain the strongest known lower bounds when negative stepsizes are allowed. For nonnegative schedules, [JCY26] further improved the non-anytime and anytime lower bounds to $\Omega(n^{-1.4500})$ and $\Omega(n^{-1.1837})$, respectively. Their hard instances form a chain of local one-sided Huber functions[5] coupling adjacent coordinates, much like Nesterov's quadratic chain.

---

1. A *deterministic first-order method* accesses the objective only through function values and gradients at arbitrary points. Nesterov's simpler quadratic-chain proof assumes $x_t \in x_0 + \mathrm{span}\{\nabla f(x_0), \ldots, \nabla f(x_{t-1})\}$, referred to as the *linear-span assumption* [Nes04, Assumption 2.1.4].↩

2. Fixed-step methods use predetermined coefficients to combine past gradients and may use momentum.↩

3. $1+\sqrt{2}$ is the silver ratio, whose base-2 logarithm is p_{sil}. For smooth strongly convex objectives, constant-step GD has an $O(\kappa \log(1/\varepsilon))$ iteration bound, while Nesterov acceleration achieves $O(\sqrt{\kappa}\log(1/\varepsilon))$ using momentum [Nes83].↩

4. Throughout, we use the phrase '$\Omega(n^{-p})$ anytime lower bound' as shorthand for the statement that no single infinite stepsize schedule achieves an $o(n^{-p})$ rate.↩

5. The one-sided Huber function is defined in Section 2.↩

1.1 Main Results

A key limitation of the local one-sided Huber functions in [JCY26] is that their fixed gradient directions restrict the tightness of the resulting lower bounds. We address this by refining these local functions to *bend the gradient descent trajectory*, which allows us to establish tighter lower bounds with exponents $p_{\mathrm{sil}} + o(1)$ and $p_{\mathrm{any}} + o(1)$.

Theorem 1.1 (Finite-horizon error lower bound).

There is an absolute constant $C > 0$ such that for every sufficiently large n and every schedule $H \in [0, \infty)^n$,

$$\mathcal{R}_n(H) \geq n^{-\left(p_{\mathrm{sil}} + C\sqrt{\frac{\log\log n}{\log n}}\right)}. \tag{2}$$

Theorem 1.2 (Anytime error lower bound).

There is an absolute constant $C > 0$ such that every infinite nonnegative schedule $H = (h_t)_{t \geq 1}$ satisfies

$$\mathcal{R}_n(H_n) \geq n^{-\left(p_{\mathrm{any}} + C\sqrt{\frac{\log\log n}{\log n}}\right)} \tag{3}$$

for infinitely many n, where $p_{\mathrm{any}} := 2p_{\mathrm{sil}}/(1 + p_{\mathrm{sil}})$.

Together with the corresponding upper bounds [AP25b], [ZLDC25], our results identify p_{sil} and p_{any} as the optimal polynomial convergence exponents in the non-anytime and anytime settings, respectively. This resolves the conjecture of Altschuler and Parrilo [AP25b] that the silver exponent is optimal. Sections 2–4 explains the idea behind our hard function construction.

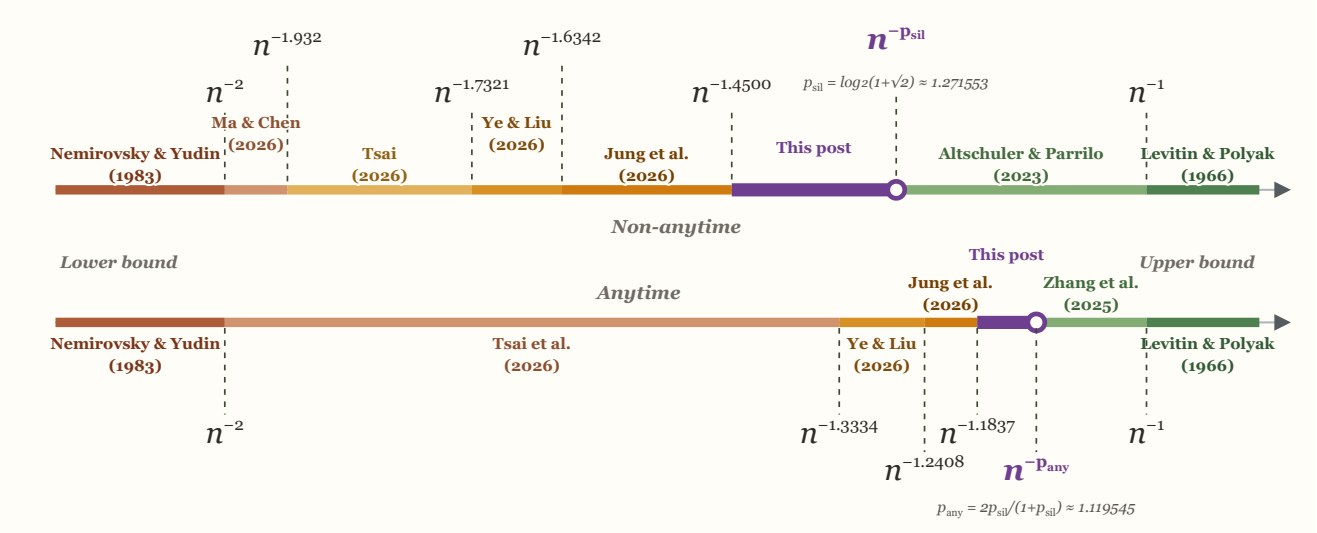


Figure 1. Lower and upper bounds on the optimal polynomial convergence exponents for non-anytime and anytime gradient descent.

1.2 Additional Literature Review

1.2.1 Early work on quadratic objectives.

For a quadratic objective $f(x) = \frac{1}{2}(x - x_*)^\top A(x - x_*)$, the error $x_n - x_*$ is given by a polynomial in A applied to $x_0 - x_*$. When the spectrum of A lies in $[\mu, L]$ with $\mu > 0$, the choice of predetermined stepsizes reduces to a classical polynomial minimax problem [Gav50], [GV61]. Young [You53] used Chebyshev stepsizes to attain the optimal $O(\sqrt{\kappa}\log(1/\varepsilon))$ iteration bound for reducing the squared distance to the minimizer by a factor of ε, where $\kappa = L/\mu$. Conjugate gradients [HS52] and the heavy-ball method [Pol64] achieve acceleration on quadratic objectives through recurrences that also use previous iterates. This polynomial analysis relies on the constant Hessian of quadratic objectives and does not directly extend to general smooth strongly convex functions [Alt18, Chapter 8].

1.2.2 Accelerating GD via stepsize schedules.

Beyond quadratics, Altschuler's thesis [Alt18, Chapter 8] established improvements over short horizons in the strongly convex setting. Later, Grimmer [Gri24] proved that periodic long steps improve the function suboptimality bound by a constant factor when the initial level set is bounded. The silver schedules discussed above improve the polynomial exponent in both the strongly convex and general smooth convex settings [AP25a], [AP25b]. For the original smooth convex silver schedule, [WMYZ26] determine the exact worst-case bound on function suboptimality at horizons $n = 2^k - 1$. As follow-up work on the silver schedule, [GSW25b], [ZJ26] developed composition and concatenation rules that improve the constant factor and yield the silver rate at arbitrary prescribed horizons. Under the symmetric composition rules in these works, [LCJW26] proved that recursively combining schedules whose lengths differ by at most one minimizes the worst-case bound on function suboptimality at every horizon. [GSW25b] also proved that the silver rate is minimax optimal within a family of schedules

generated by their recursive composition rules. The silver-rate guarantee has also been extended to proximal and projected GD [BA26]. For separable smooth strongly convex objectives, Altschuler and Parrilo [AP24] obtained asymptotic acceleration using random stepsizes.

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 accelerate GD when the Hessian eigenvalues form two clusters [Oym21]. For linearly separable logistic regression, a large constant stepsize chosen based on the horizon length gives an $\widetilde{O}(n^{-2})$ loss bound [WBTY24]. Large constant stepsizes also improve convergence guarantees for certain two-layer networks trained with logistic loss [CWM+24] and for regularized logistic regression on separable data under suitable regularization conditions [WMB25]. More recently, GD with a predetermined stepsize schedule achieves stretched exponential loss decay for separable logistic regression [KMM+26]. Beyond minimization, power-law schedules accelerate the last-iterate convergence of the extragradient method on unconstrained biaffine min-max problems [WZCL26].

1.2.3 Performance estimation.

As mentioned above, PEPs provide a systematic framework for analyzing and designing first-order optimization methods. Optimizing a PEP guarantee over the method's coefficients can identify first-order methods with optimal convergence guarantees within a chosen class [DT20]. Within the class of fixed-step methods, [GJP26] proved that OGM uniquely minimizes worst-case function suboptimality. Branch-and-bound methods have also been used to numerically solve the minimax problem for GD with horizons up to $n = 25$, yielding predetermined stepsizes that minimize the worst-case function suboptimality over smooth convex functions [DGVPR24, Section 6.1]. These numerical solutions closely agree with the recursively constructed schedules in [GSW25b], providing numerical support for the optimality of p_{sil}. For a broader perspective on minimax optimality, see the survey [GSW26].

2. Classical Lower Bound Constructions

We first review classical lower bound constructions and the recent hard functions of [MC26], [JCY26], highlighting the ideas that guide our construction. We then present the idea behind our hard function construction.[1]

---

1. Replacing f by f/L and each h_t by Lh_t preserves the iterates and normalized function suboptimality, so we use $L = 1$ in the constructions and proofs below.↩

2.1 The Role of Huber and Quadratic Functions

To build intuition, consider two basic one-dimensional hard functions first. A one-sided version of the classical Huber function [Hub64] is defined by

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

Its gradient is constant on the affine region $z \ge \delta$, so short steps make slow progress toward the minimum. Next, consider the quadratic objective $f(x) = x^2/2$, where the update is $x_t = (1 - h_t)x_{t-1}$, so a large step ($h_t > 2$) overshoots the minimizer and increases the distance to it. These examples yield the following elementary lower bounds.

Fact 2.1 (Huber and quadratic lower bounds).

Let $n \ge 1$ and $H = (h_1, \dots, h_n) \in [0, \infty)^n$.

1. (see [YL26, Fact A.1]) If every step is short ($h_t \le 2$), a Huber example gives

$$\mathcal{R}_n(H) \ge \frac{1}{1 + 2\sum_{t=1}^n h_t} \ge \frac{1}{4n+1}.$$

2. If every step is long ($h_t > 2$), the quadratic $f(x) = x^2/2$ gives

$$\mathcal{R}_n(H) \ge \prod_{t=1}^n (1 - h_t)^2 > 1.$$

Every constant schedule falls into one of these two cases and therefore has a worst-case error of $\Omega(n^{-1})$. More precisely, the same examples give [DT14, Section 3.2]

$$\mathcal{R}_n(h, \dots, h) \ge \max\left\{ \frac{1}{1 + 2nh}, (1 - h)^{2n} \right\}, \qquad h \ge 0.$$

For all $0 < h < 2$, this lower bound matches the upper bound exactly, including the constant factor [Kim24].

These examples suggest that a hard function for GD should combine affine regions that delay progress during short steps with quadratic curvature that causes large steps to overshoot the minimizer. This insight also appears in PEP analyses [DT14] and motivates hedging between short and long steps in the silver schedule, which yields the accelerated upper bound [AP25a, Section 2.1].

Proof of the Huber and quadratic lower bounds

Set $S := \sum_{t=1}^{n} h_t$ and $\delta := 1/(1+2S)$. Start GD on H_δ at $x_0 = 1$. Since the stepsizes are nonnegative, induction gives

$$x_t = 1 - \delta \sum_{j=1}^{t} h_j \geq 1 - \delta S \geq \delta.$$

Every iterate therefore stays in the affine region, where the gradient is δ. Since 0 minimizes H_δ,

$$\frac{2[H_\delta(x_n) - H_\delta(0)]}{|x_0|^2} = 2\delta(1-\delta S) - \delta^2 = \frac{1}{1+2S}.$$

If every $h_t \leq 2$, then $S \leq 2n$, giving $1/(4n+1)$.

For $f(x) = x^2/2$ and $x_0 = 1$, the recurrence gives $x_n = \prod_{t=1}^{n}(1-h_t)$, so the normalized error is $\prod_{t=1}^{n}(1-h_t)^2$. When every $h_t > 2$, every factor exceeds one. For a constant schedule, the worst-case error is at least the larger of the Huber and quadratic errors, giving the stated maximum.

2.2 The Role of Zero-Chains

As discussed above, using the Huber function alone yields a lower bound that weakens as the sum of stepsizes grows. A different approach uses higher-dimensional hard functions to ensure the minimizer has non-zero components in unexplored directions. Nemirovsky and Yudin [NY79] used this idea to establish the classical $\Omega(n^{-2})$ lower bound for general deterministic first-order methods. Under the linear-span assumption[1], Nesterov [Nes04, Section 2.1.2] presents a simpler quadratic zero-chain that limits how quickly a method can explore new coordinates, regardless of its stepsizes. This chain couples adjacent coordinates $x^{(j)}$ and $x^{(j+1)}$ through quadratic terms, where $x^{(j)}$ denotes the jth coordinate of x.

Fact 2.2 (Nesterov's quadratic chain [Nes04, Section 2.1.2]).

Fix $n \geq 1$ and let $d = 2n + 1$. The function

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

is convex and 1-smooth. Starting from $x_0 = 0$, every method satisfying the linear-span assumption obeys

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

In particular, this holds for GD with any choice of stepsizes.

The bound follows from the *first-order zero-chain* property of f [DT22, Definition 1]. This means that, for every $0 \leq r < d$,

$$x^{(r+1)} = \cdots = x^{(d)} = 0 \quad \Longrightarrow \quad (\nabla f(x))^{(j)} = 0 \quad \text{for } j \geq r + 2.$$

Starting from $x_0 = 0$, each iteration of a method satisfying the linear-span assumption can therefore introduce at most one new coordinate, so $x_t^{(j)} = 0$ for $j > t$. The minimizer, however, has $x_*^{(j)} = 1 - j/(d+1) > 0$ in every coordinate. Minimizing f with $x^{(n+1)} = \cdots = x^{(d)} = 0$ and comparing the result with $f(x_*)$ gives the lower bound in Fact 2.2 [Nes04, Theorem 2.1.7].

Proof of the quadratic zero-chain lower bound

The Hessian of f is positive semidefinite. For any vector z,

$$z^\top \nabla^2 f\, z = \frac{1}{4}\left[(z^{(1)})^2 + \sum_{j=1}^{d-1}(z^{(j)} - z^{(j+1)})^2 + (z^{(d)})^2\right] \leq \|z\|^2,$$

so f is convex and 1-smooth. The zero-chain property implies by induction that $x_t^{(j)} = 0$ for $j > t$ under the linear-span assumption.

The $d+1$ successive differences from 1 to 0 sum to 1. Cauchy–Schwarz shows that their sum of squares is at least $1/(d+1)$, attained when all differences equal $1/(d+1)$. Thus $x_*^{(j)} = 1 - j/(d+1)$ and $f(x_*) = 1/[8(d+1)]$. Restricting to the first n coordinates leaves only $n+1$ differences, so

$$f(x_n) \geq \frac{1}{8(n+1)}, \qquad f(x_n) - f(x_*) \geq \frac{1}{16(n+1)} \quad (d = 2n+1).$$

Finally,

$$\|x_*\|^2 = \frac{d(2d+1)}{6(d+1)} \leq \frac{d+1}{3} = \frac{2(n+1)}{3}, \qquad \frac{2[f(x_n) - f(x_*)]}{\|x_*\|^2} \geq \frac{3}{16(n+1)^2}.$$

3. Recent Lower Bound Breakthrough

The classical $\Omega(n^{-2})$ lower bound discussed above holds for general deterministic first-order methods [NY79]. To obtain tighter lower bounds for GD, we must therefore tailor the hard function to the prescribed stepsize schedule. The construction in [JCY26] combines a coupled coordinate chain with changes in gradient direction after selected steps, which are chosen adversarially for each prescribed schedule.

3.1 Selecting the Checkpoints

To formalize this selection, fix a nonnegative schedule $H = (h_1, \ldots, h_n)$ and select a set of update indices $T = \{t_1 < \cdots < t_k\}$ called *checkpoints*. The updates between successive checkpoints form a *gap*. Set $t_0 = 0$, $t_{k+1} = n+1$, and define

$$b_i := h_{t_i} \quad (1 \leq i \leq k), \qquad s_i := \sum_{t=t_{i-1}+1}^{t_i - 1} h_t \quad (1 \leq i \leq k+1). \tag{4}$$

Here b_i is the stepsize at checkpoint t_i, and s_i is the *mass* of the preceding gap, the sum of its stepsizes. The final gap has mass s_{k+1}. Figure 2 illustrates this notation.

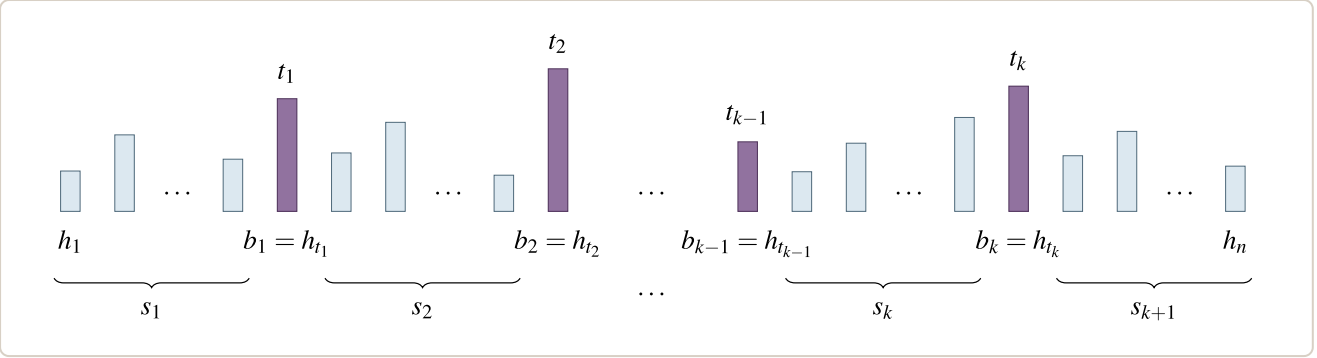


Figure 2. The selected stepsizes are $b_i = h_{t_i}$. Each brace marks the sum s_i of the stepsizes in a gap.

The construction in [MC26] yields an $\Omega(n^{-1.932})$ non-anytime lower bound. For selected long steps, the hard function in [MC26, Theorem 4.1] keeps the gradient constant during each gap and changes it after each selected step. See also [YL26, Appendix A] for a geometric illustration of the trajectory. The hard function in [MC26] is defined globally on all coordinates. The construction in [JCY26, Section 2] instead couples adjacent coordinates through *local hard functions*, which we also call *components*. This chain resembles Nesterov's quadratic construction and leads to the stronger $\Omega(n^{-\log_2(1+\sqrt{3})})$ non-anytime lower bound, where $\log_2(1+\sqrt{3}) \approx 1.4500$. During each gap, the current component contributes a constant gradient, as in the construction of [MC26], while all later components remain inactive.[1] For a fixed checkpoint set, this gives the following bound.

Fact 3.1 ([JCY26, Lemma 2.3]).

For every positive schedule $H = (h_1, \dots, h_n)$ and every checkpoint set T,

$$\mathcal{R}_n(H) \geq \frac{1}{4(1+s_{k+1})} \prod_{i=1}^{k} \left[\frac{b_i}{2(2+s_i)}\right]^2. \tag{5}$$

For $T = \varnothing$, the product is one and $s_1 = \sum_{t=1}^{n} h_t$.

Specifically, their construction uses the one-sided Huber function H_δ defined above, joining adjacent coordinates through

$$F(x) := \frac{1}{4}\sum_{i=1}^{k} H_{\delta_i}\left(x^{(i)} - x^{(i+1)} - \ell_i\right) + \frac{1}{2} H_{\delta_{k+1}}\left(x^{(k+1)} - \ell_{k+1}\right).$$

Here ℓ_i is the *activation threshold* of component i, and $D_i := x^{(i)}_{t_{i-1}} - \ell_i > 0$ is the *excess* above it at the start of gap i. Starting from $x_0 = e_1$, set $\ell_1 = 0$ and $D_1 = 1$. For this Huber construction, the parameters are chosen recursively as $\delta_i = \frac{2D_i}{2+s_i}$, $\ell_{i+1} = \frac{s_i\delta_i}{4}$, and $D_{i+1} = \frac{b_i\delta_i}{4}$ for $1 \le i \le k$. During gap i, coordinate $i+1$ never exceeds ℓ_{i+1} and equals it just before the checkpoint. The checkpoint then raises this coordinate to $\ell_{i+1} + D_{i+1}$ and activates component $i+1$, while all later components remain inactive. Thus each factor $b_i/[2(2+s_i)]$ in (5) is the *transfer ratio* D_{i+1}/D_i. Since $D_1 = 1$, their product is the final excess D_{k+1}. The final one-dimensional component is $\frac{1}{2}H_{\delta_{k+1}}(x^{(k+1)} - \ell_{k+1})$. Choosing $\delta_{k+1} = D_{k+1}/(1+s_{k+1})$ gives a lower bound proportional to $D^2_{k+1}/(1+s_{k+1})$.

For a prescribed schedule H, this Huber construction is available for every checkpoint set T. The checkpoints can therefore be chosen to maximize the right-hand side of (5). The recursive argument in [JCY26, Section 3] bounds this maximum from below by $\Omega\big(n^{-\log_2(1+\sqrt{3})}\big)$, uniformly over schedules H.

---

1. A component is *active* at an iterate if its gradient there is nonzero, and *inactive* otherwise.↩

3.2 The Bottleneck

For fixed H and T, improving the local functions to give larger transfer ratios D_{i+1}/D_i would further strengthen the lower bound. 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 transfer ratio $b_i/[2(2+s_i)]$ in (5) has $c = 2$, whereas reducing c toward 1 would bring the exponent toward p_{sil}. During gap i, the current Huber component contributes equally to the decrease in $x^{(i)}$ and the increase in $x^{(i+1)}$. Both changes reduce the margin $x^{(i)} - x^{(i+1)} - \ell_i$, producing the coefficient 2 of s_i in the denominator $4 + 2s_i$ in (5) [JCY26, Section 2].

4. Our Hard Function: Bending the Local Trajectory

Keeping the chain structure and notation above, we seek local hard functions Φ_i for an objective of the form

$$F(x) := \frac{1}{2}\sum_{i=1}^{k}\Phi_i(x) + \frac{1}{2}\Phi_{k+1}(x).$$

For $1 \le i \le k$, each Φ_i should depend only on $x^{(i)}$ and $x^{(i+1)}$ and be nonnegative, convex, and 1-smooth. The final component Φ_{k+1} depends only on $x^{(k+1)}$. For $1 \le i \le k$, we impose the following *transfer property* so that only the current component Φ_i contributes to the updates of $x^{(i)}$ and $x^{(i+1)}$ during gap i and in its checkpoint update[1]

$$\begin{aligned} \Phi_i(x) = 0, \quad \nabla \Phi_i(x) = 0 \qquad & \text{whenever } x^{(i)} \le \ell_i \text{ and } x^{(i+1)} \ge 0, \\ \partial_{x^{(i+1)}} \Phi_i(x) = 0 \qquad & \text{whenever } x^{(i+1)} \ge \ell_{i+1}. \end{aligned} \tag{6}$$

The first condition keeps later components inactive while their first coordinates stay at or below their thresholds and their second coordinates stay nonnegative. For $i > 1$, applying the second condition to Φ_{i-1} gives $\partial_{x^{(i)}} \Phi_{i-1} = 0$ whenever $x^{(i)} \ge \ell_i$, so the preceding component does not affect the update of $x^{(i)}$. The shaded region in Figure 3 illustrates the second condition for Φ_i. For $i < k$, the iterates remain in this region during gap $i+1$, including just before the checkpoint update at t_{i+1}.

As before, we write $D_{i+1} := x_{t_i}^{(i+1)} - \ell_{i+1} > 0$ for the excess above ℓ_{i+1} immediately after the checkpoint update at t_i. Our goal is to construct Φ_i with transfer ratio

$$\frac{D_{i+1}}{D_i} \approx \frac{b_i}{c s_i}.$$

We aim to make c arbitrarily close to 1.

Notice that growth in $x^{(i+1)}$ during gap i raises the threshold ℓ_{i+1} needed to keep Φ_{i+1} inactive. To leave a larger excess D_{i+1} after checkpoint t_i, we limit this early growth while ensuring a large increase at the checkpoint update. When s_i is sufficiently large, the trajectory in gap i initially moves mainly in coordinate $x^{(i)}$ and turns toward coordinate $x^{(i+1)}$ before checkpoint t_i. We call this change in direction *bending*, the key new ingredient in our construction.

More concretely, we construct the GD trajectory in gap i with gradient vectors on a circular arc. For parameters $\epsilon, \gamma > 0$, consider the circle centered at $(0, \gamma)$ with radius $R := \sqrt{\epsilon^2 + (1+\gamma)^2}$, which passes through $(\epsilon, -1)$. Its intersection with the positive horizontal axis is $(c, 0)$, where $c = \sqrt{1 + 2\gamma + \epsilon^2}$. Up to a common positive scale, we place the gradients on the arc joining these two points, as shown on the left of Figure 3. They turn toward $(\epsilon, -1)$ during gap i. At $x_{t_i - 1}$, just before the checkpoint update, the gradient is proportional to $(\epsilon, -1)$, so the update mainly increases $x^{(i+1)}$ when ϵ is small.

Given the stepsizes in gap i, we construct the trajectory backward from the iterate just before checkpoint t_i; see Lemma 4.2. We use the arc vectors along this trajectory, together with 0 and $(c, 0)$, as slopes in a finite maximum of affine functions. After scaling and translation, the

Moreau envelope of this function gives a convex, 1-smooth local hard function Φ_i whose gradient at each constructed iterate is the corresponding scaled arc vector; see Lemma 4.3 and (10).[2]

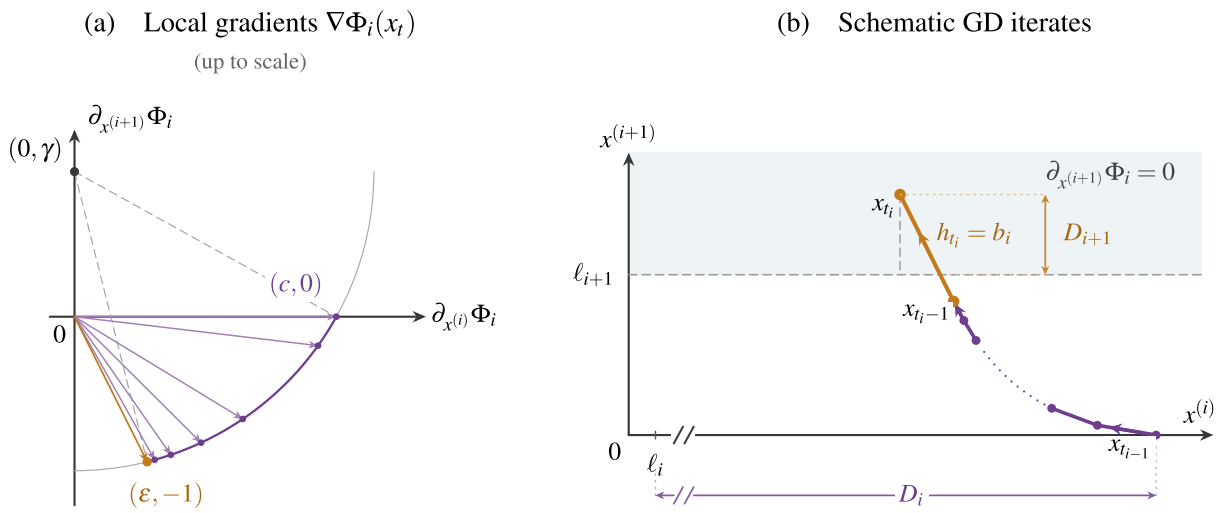


Figure 3. Local gradients $\nabla\Phi_i(x_t)$ (left) and a schematic GD trajectory (right). The step h_{t_i} (orange) activates the next component.

---

1. In [JCY26, Section 2 and Appendix B.2], the preceding component Φ_{i-1} contributes nonnegatively to the update of $x^{(i)}$ for $i > 1$, helping keep Φ_i in its affine region with constant gradient during gap i. Such a contribution can change the gradient direction in our construction, so we impose $\partial_{x^{(i+1)}}\Phi_i(x) = 0$ whenever $x^{(i+1)} \geq \ell_{i+1}$.↩

2. A similar use of Moreau envelopes to realize a prescribed GD trajectory appears in [MC26, Section 4.3].↩

The Improved Hard-Function Bound.

Our local hard functions will give the transfer ratio $D_{i+1}/D_i = (b_i - 2)/[c(a + s_i)]$, where a depends only on ϵ, γ. To leave positive excess, we use checkpoints with $b_i > 2$, which we call *admissible*. A checkpoint set is admissible if all its checkpoints are admissible. The construction below gives the following lower bound for every admissible checkpoint set.

Lemma 4.1 (Lower bound for a checkpoint set).

Fix $\epsilon, \gamma > 0$ and let $c := \sqrt{1 + 2\gamma + \epsilon^2}$ and $R := \sqrt{\epsilon^2 + (1+\gamma)^2}$. For every nonnegative schedule $H = (h_1, \dots, h_n)$ and admissible checkpoint set T,

$$\mathcal{R}_n(H) \geq \frac{1}{4(1+s_{k+1})} \prod_{i=1}^{k} \left[\frac{b_i - 2}{c(a+s_i)} \right]^2, \tag{7}$$

where $a = \max\{2, \frac{2}{c}\left(\epsilon + \epsilon^{-1}[\frac{c(1+\gamma+R)}{\epsilon(\gamma+R)}]^{R/\gamma}\right)\}$.

We prove this lemma in three steps: constructing the gap trajectory in Section 4.1, realizing it with a local hard function in Section 4.2, and combining the local functions into a global objective in Section 4.3.

4.1 Constructing the Gap Trajectory

Using the parameters ϵ, γ, c, R from Lemma 4.1, we first construct a two-dimensional gap trajectory. After scaling and translation, this will give the GD trajectory generated by the local hard function Φ_i during gap i. We choose the gradient vectors on a circular arc with center $(0, \gamma)$ and radius R. Let K be the convex hull of the origin and the shorter arc joining $(c, 0)$ to $(\epsilon, -1)$. Lemma 4.2 constructs points r_j backward from $(\epsilon, -1)$ so that $r_j = r_{j-1} - \alpha_j v_{j-1}$, where $v_j = \Pi_K(r_j)$ is the Euclidean projection of r_j onto K. Each update has the form of a GD step with stepsize α_j and gradient v_{j-1}. The projection condition later lets us realize the scaled vectors as gradients of a convex, 1-smooth function Φ_i.

Lemma 4.2 (Gap trajectory).

For every $m \geq 0$ and nonnegative numbers $\alpha_1, \dots, \alpha_m$, there exist points $r_0, \dots, r_m$ and arc vectors $v_0, \dots, v_m$ satisfying

$$r_m = v_m = (\epsilon, -1), \qquad r_j = r_{j-1} - \alpha_j v_{j-1} \quad (1 \leq j \leq m), \qquad v_j = \Pi_K(r_j) \quad (0 \leq j \leq m).$$

With $S := \sum_{j=1}^{m} \alpha_j$, the initial point has the form

$$r_0 = (\epsilon + P, -1 - Q), \qquad 0 \leq P \leq cS, \qquad 0 \leq Q \leq \left[\frac{c(1+\gamma+R)}{\epsilon(\gamma+R)}\right]^{R/\gamma} - 1.$$

Here P is the decrease in the first coordinate and Q is the increase in the second from r_0 to r_m. When defining Φ_i, we scale and translate these two coordinates to $x^{(i)}$ and $x^{(i+1)}$. The same c in $P \leq cS$ will appear in the denominator $c(a + s_i)$ of the transfer ratio D_{i+1}/D_i.

Proof of Lemma 4.2

We first construct the gap iterates backward, then verify the projection identity $v_j = \Pi_K(r_j)$ and bound the displacements P and Q.

Step 1. Constructing the gap trajectory.

Set

$$v(q) := (p(q), -q), \qquad p(q) := \sqrt{R^2 - (q+\gamma)^2}, \qquad 0 \leq q \leq 1. \tag{21}$$

The vectors $v(q)$ lie on the circle of radius R centered at $(0, \gamma)$, and

$$v(0) = (c, 0), \qquad v(1) = (\epsilon, -1), \qquad \epsilon \leq p(q) \leq c, \qquad \frac{q}{p(q)} \leq \frac{1}{\epsilon}. \tag{22}$$

Let

$$K := \operatorname{conv}\big(\{0\} \cup \{v(q) : 0 \leq q \leq 1\}\big).$$

The set K lies in the closed disk centered at $(0, \gamma)$ with radius R, and every $w \in K$ satisfies $w^{(1)} \le c$. Fix stepsizes $\alpha_1, \dots, \alpha_m \ge 0$ and set $S = \sum_{j=1}^{m} \alpha_j$. Write $r_j = (u_j, -w_j)$ and $v_j = (p_j, -q_j)$ for $0 \le j \le m$. Start with $r_m = v_m = (\epsilon, -1)$. For $j = m, m-1, \dots, 1$, define

$$\Delta_j := w_j + \gamma - \alpha_j \gamma.$$

If $\Delta_j \le u_j \gamma / c$, choose $v_{j-1} = (p_{j-1}, -q_{j-1}) := (c, 0)$. Otherwise, choose

$$d_j := \sqrt{u_j^2 + \Delta_j^2}, \qquad q_{j-1} := \frac{R\Delta_j}{d_j} - \gamma, \qquad p_{j-1} := p(q_{j-1}) = \frac{Ru_j}{d_j}. \tag{23}$$

In both cases, set $r_{j-1} := r_j + \alpha_j v_{j-1}$, so

$$u_{j-1} = u_j + \alpha_j p_{j-1}, \qquad w_{j-1} = w_j + \alpha_j q_{j-1}.$$

Geometrically, when $\Delta_j > u_j \gamma / c$, we choose v_{j-1} where the ray from $(0, \gamma)$ through $r_j + \alpha_j (0, \gamma)$ meets the circle, as shown in Figure 4. If the ray passes at or above the corner $(c, 0)$, we choose the corner instead.

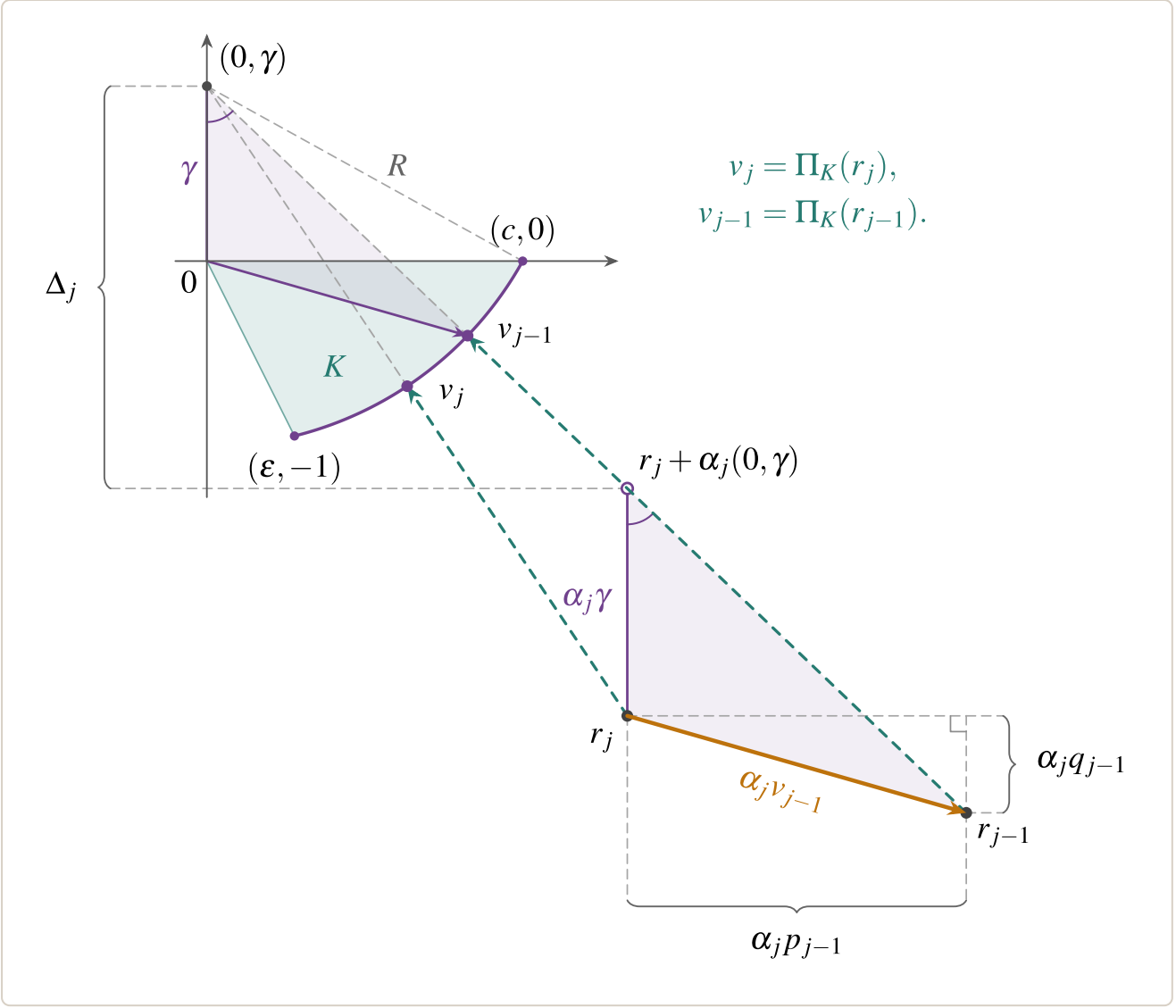


Figure 4. The backward construction in (23). Pale green denotes K, and green dashed arrows show $v_j = \Pi_K(r_j)$ and $v_{j-1} = \Pi_K(r_{j-1})$. The purple triangles are similar, with scale factor α_j.

Step 2. Verifying the projection identity.

We prove by backward induction that, for $0 \le j \le m$,

$$v_j = \Pi_K(r_j), \qquad u_j \ge \epsilon, \qquad w_j \ge 1, \qquad \frac{w_j}{u_j} \le \frac{1}{\epsilon}. \tag{24}$$

The statements in (24) hold at $r_m = v_m = (\epsilon, -1)$. Assume they hold at r_j. We first check that the chosen vector lies on the arc and that the coordinate bounds hold at r_{j-1}. If $\Delta_j > u_j\gamma/c$, then

$$\frac{\gamma}{c} < \frac{\Delta_j}{u_j} \le \frac{w_j + \gamma}{u_j} \le \frac{1+\gamma}{\epsilon}.$$

By (23), $q_{j-1} = R(\Delta_j/u_j)/\sqrt{1+(\Delta_j/u_j)^2} - \gamma$. The map $s \mapsto Rs/\sqrt{1+s^2} - \gamma$ is increasing and takes the values 0 and 1 at $s = \gamma/c$ and $s = (1+\gamma)/\epsilon$, respectively. Hence $q_{j-1} \in (0, 1]$. If $\Delta_j \le u_j\gamma/c$, then $(p_{j-1}, q_{j-1}) = (c, 0)$. In both cases, $p_{j-1} \ge \epsilon$ and $q_{j-1} \ge 0$, giving $u_{j-1} \ge \epsilon$ and $w_{j-1} \ge 1$. The inductive ratio bound and (22) also give

$$w_{j-1} = w_j + \alpha_j q_{j-1} \le \frac{u_j + \alpha_j p_{j-1}}{\epsilon} = \frac{u_{j-1}}{\epsilon}.$$

It remains to verify the projection. For $v \in K$, the identity $v = \Pi_K(r)$ is equivalent to $\langle r - v, w - v\rangle \le 0$ for every $w \in K$. If $\Delta_j > u_j\gamma/c$, we show that $r_{j-1} - v_{j-1}$ points along the outward radius at v_{j-1}. By (23),

$$r_j + \alpha_j(0,\gamma) - (0,\gamma) = (u_j, -\Delta_j) = \frac{d_j}{R}\big(v_{j-1} - (0,\gamma)\big).$$

Substituting this into $r_{j-1} = r_j + \alpha_j v_{j-1}$ gives

$$r_{j-1} - v_{j-1} = \left(\frac{d_j}{R} + \alpha_j - 1\right)\big(v_{j-1} - (0,\gamma)\big). \tag{25}$$

The reverse triangle inequality and the coordinate bounds give

$$d_j \ge \sqrt{u_j^2 + (w_j + \gamma)^2} - \alpha_j\gamma \ge R - \alpha_j\gamma.$$

Thus the coefficient in (25) is nonnegative, since

$$\frac{d_j}{R} + \alpha_j - 1 \ge \alpha_j\left(1 - \frac{\gamma}{R}\right) \ge 0.$$

Because K lies inside the disk, every $w \in K$ satisfies

$$\langle v_{j-1} - (0,\gamma), w - v_{j-1}\rangle \le R\|w - (0,\gamma)\| - R^2 \le 0.$$

This proves $v_{j-1} = \Pi_K(r_{j-1})$.

If $\Delta_j \le u_j\gamma/c$, then $v_{j-1} = (c, 0)$ and $w_{j-1} = w_j$. Rearranging this condition gives

$$u_{j-1} - c = u_j + \alpha_j c - c \geq \frac{c w_j}{\gamma}.$$

For $w \in K$, we have $w^{(1)} \leq c$ and $\langle (c, -\gamma), w - (c, 0) \rangle \leq 0$, since $(c, -\gamma)$ is the outward radius at $(c, 0)$. Hence

$$\begin{aligned} \langle r_{j-1} - (c, 0), w - (c, 0) \rangle &= (u_{j-1} - c)(w^{(1)} - c) - w_j w^{(2)} \\ &\leq \frac{w_j}{\gamma} \langle (c, -\gamma), w - (c, 0) \rangle \leq 0. \end{aligned} \tag{26}$$

By (26), $v_{j-1} = \Pi_K(r_{j-1})$, completing the induction.

Step 3. Bounding P and Q.

We first identify the indices at which $q_{j-1} = 0$. If $\Delta_j \leq u_j \gamma / c$ and $j \geq 2$, the choice $v_{j-1} = (c, 0)$ gives $w_{j-1} = w_j$ and $u_{j-1} = u_j + \alpha_j c$. Hence

$$\Delta_{j-1} = w_{j-1} + \gamma - \alpha_{j-1}\gamma \leq w_j + \gamma \leq \frac{u_j \gamma}{c} + \alpha_j \gamma = \frac{u_{j-1}\gamma}{c}.$$

Thus the backward recursion also chooses $v_{j-2} = (c, 0)$. Repeating this argument shows that once $(c, 0)$ is chosen, all earlier gradient vectors equal $(c, 0)$. Thus there is an index $j_\star \in \{0, \ldots, m\}$ such that, for $1 \leq j \leq m$,

$$\Delta_j > \frac{u_j \gamma}{c} \quad \Longleftrightarrow \quad j_\star < j \leq m, \tag{27}$$

where $j_\star = m$ if the strict inequality never holds.

The coordinate recurrences give

$$P = \sum_{j=1}^{m} \alpha_j p_{j-1}, \qquad Q = \sum_{j=1}^{m} \alpha_j q_{j-1}.$$

Since $p_{j-1} \leq c$, we have $P \leq cS$. To bound Q independently of S, use $q_{j-1} = 0$ for $j \leq j_\star$ and $q_{j-1} \leq p_{j-1}/\epsilon$ from (22). These give

$$0 \le Q = \sum_{j=j_\star+1}^{m} \alpha_j q_{j-1} \le \frac{1}{\epsilon} \sum_{j=j_\star+1}^{m} \alpha_j p_{j-1} = \frac{u_{j_\star}}{\epsilon} - 1.$$

The last equality follows from $\alpha_j p_{j-1} = u_{j-1} - u_j$ and $u_m = \epsilon$. It therefore suffices to bound $u_{j_\star}/\epsilon$.

Set $\sigma_j := (w_j + \gamma)/u_j$, the magnitude of the slope from the circle center $(0, \gamma)$ to $r_j = (u_j, -w_j)$. For $j_\star < j \le m$, the recurrence and (23) give

$$(u_{j-1}, w_{j-1} + \gamma) = \left(1 + \frac{\alpha_j R}{d_j}\right)(u_j, \Delta_j),$$

so

$$\sigma_{j-1} = \frac{\Delta_j}{u_j} = \sigma_j - \frac{\alpha_j \gamma}{u_j}, \qquad p_{j-1} = \frac{R}{\sqrt{1 + \sigma_{j-1}^2}}.$$

Thus the change in slope determines the increase in u_j under the backward recursion,

$$\frac{u_{j-1}}{u_j} = 1 + \frac{\alpha_j}{u_j} p_{j-1} = 1 + \frac{R}{\gamma} \frac{\sigma_j - \sigma_{j-1}}{\sqrt{1 + \sigma_{j-1}^2}}. \tag{28}$$

To obtain an inequality that telescopes when multiplied over j, set $A(\sigma) := \sigma + \sqrt{1 + \sigma^2}$. Its derivative satisfies $A'(\sigma)/A(\sigma) = 1/\sqrt{1 + \sigma^2}$, which matches the denominator in (28). Convexity of A gives

$$\frac{A(\sigma_j)}{A(\sigma_{j-1})} \ge 1 + \frac{\sigma_j - \sigma_{j-1}}{\sqrt{1 + \sigma_{j-1}^2}}.$$

Since $\sigma_j \ge \sigma_{j-1}$ and $R/\gamma > 1$, Bernoulli's inequality gives

$$\frac{u_{j-1}}{u_j} \le \left(1 + \frac{\sigma_j - \sigma_{j-1}}{\sqrt{1 + \sigma_{j-1}^2}}\right)^{R/\gamma} \le \left(\frac{A(\sigma_j)}{A(\sigma_{j-1})}\right)^{R/\gamma}. \tag{29}$$

Multiplying (29) over $j = j_\star + 1, \dots, m$ cancels all intermediate terms, leaving

$$\frac{u_{j_\star}}{\epsilon} \le \left[\frac{A(\sigma_m)}{A(\sigma_{j_\star})}\right]^{R/\gamma}.$$

Here $u_m = \epsilon$ and $\sigma_m = (1+\gamma)/\epsilon$. If $j_\star < m$, then $\sigma_{j_\star} = \Delta_{j_\star+1}/u_{j_\star+1} > \gamma/c$ by (27); if $j_\star = m$, then $\sigma_{j_\star} = \sigma_m \ge \gamma/c$. Since A is increasing,

$$A(\sigma_m) = \frac{1+\gamma+R}{\epsilon}, \qquad A(\sigma_{j_\star}) \ge A(\gamma/c) = \frac{\gamma+R}{c}.$$

Combining these estimates yields

$$0 \le Q \le \frac{u_{j_\star}}{\epsilon} - 1 \le H_0 := \left[\frac{c(1+\gamma+R)}{\epsilon(\gamma+R)}\right]^{R/\gamma} - 1. \tag{30}$$

□

4.2 Constructing the Local Hard Function

For gap stepsizes $h_1, \dots, h_m$, we apply Lemma 4.2 with $\alpha_j = h_j/2$, then scale and translate its points so that the trajectory starts at $(1, 0)$. We then construct a nonnegative convex, 1-smooth function f such that GD on $f/2$ follows the scaled points. The following lemma ensures the transfer property (6) and gives the excess after the checkpoint update.

Lemma 4.3 (Local hard function).

Let $m \geq 0$, $h_1, \ldots, h_m \geq 0$, $s = \sum_{j=1}^{m} h_j$, and $b > 2$. There exist a nonnegative convex 1-smooth function $f : \mathbb{R}^2 \to \mathbb{R}$ and a threshold $\tau > 0$ satisfying

$$f(x, y) = 0 \quad (x \leq 0,\ y \geq 0), \qquad \partial_2 f(x, y) = 0 \quad (x \in \mathbb{R},\ y \geq \tau). \tag{8}$$

Starting from $(x_0, y_0) = (1, 0)$, GD on $f/2$ with stepsizes $h_1, \ldots, h_m, b$ satisfies

$$x_j > 0, \qquad 0 \leq y_j \leq y_m = \tau - \frac{2}{c(a+s)} \quad (0 \leq j \leq m),$$
$$y_{m+1} = \tau + \frac{b-2}{c(a+s)},$$

where a is defined in Lemma 4.1.

During the gap, the second coordinate stays below τ. The checkpoint update leaves excess $\frac{b-2}{c(a+s)}$, which becomes the transfer ratio D_{i+1}/D_i in the chain.

Moreau envelopes and proof of Lemma 4.3

We use the following properties of Moreau envelopes in the proof of Lemma 4.3.

Fact 4.4 (Moreau envelopes).

Let $g : \mathbb{R}^d \to (-\infty, +\infty]$ be proper, closed, and convex. Its Moreau envelope with parameter 1 and its proximal map are

$$\operatorname{env}_1 g(x) := \min_{y \in \mathbb{R}^d} \left\{ g(y) + \frac{1}{2}\|x - y\|^2 \right\}, \qquad \operatorname{prox}_g(x) := \arg\min_{y \in \mathbb{R}^d} \left\{ g(y) + \frac{1}{2}\|x - y\|^2 \right\}.$$

The function $\operatorname{env}_1 g$ is finite, convex, and continuously differentiable, and its gradient is 1-Lipschitz and satisfies $\nabla \operatorname{env}_1 g(x) = x - \operatorname{prox}_g(x)$ [Mor65, Propositions 5.b, 7.b, and 7.d].

Moreover, if $K \subset \mathbb{R}^d$ is nonempty, closed, and convex and $\sigma_K(x) := \sup_{v \in K} \langle v, x \rangle$ is its support function, then

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

where Π_K denotes Euclidean projection onto K. If $g \geq 0$, then $\operatorname{env}_1 g \geq 0$ and the two functions have the same zero set. For background, see [RW98, Sections 1.G and 2.D] and [PB14, Sections 3.1 and 6.4.2].

Proof of Lemma 4.3

Apply Lemma 4.2 with $\alpha_j = h_j/2$, so $S = s/2$. Using H_0 defined in (30), set

$$C_0 := \epsilon + \frac{H_0 + 1}{\epsilon}.$$

The resulting initial point and displacements satisfy

$$r_0 = (\epsilon + P, -1 - Q), \qquad \epsilon + P + \frac{1 + Q}{\epsilon} \leq \frac{cs}{2} + C_0. \tag{31}$$

Step 1. Defining the local hard function.

Set $\rho = 1/[c(a + s)]$ and $z_\circ = (1, 0) - 2\rho r_0$. Then the scaled trajectory $z_\circ + 2\rho r_j$ starts at $(1, 0)$. Let $\mathcal{V} = \{(0, 0), v_0, \ldots, v_m, (c, 0)\}$ and define

$$\phi(z) := 2\rho \max_{v \in \mathcal{V}} \langle v, z - z_\circ \rangle, \qquad f := \operatorname{env}_1 \phi. \tag{32}$$

Since $0 \in \mathcal{V}$, we have $\phi \geq 0$, so Fact 4.4 shows that f is nonnegative, convex, and 1-smooth. Let $K_{\mathcal{V}} = \operatorname{conv}(\mathcal{V})$. Since $v_j = \Pi_K(r_j) \in K_{\mathcal{V}} \subseteq K$, we also have $\Pi_{K_{\mathcal{V}}}(r_j) = v_j$. Also, $\phi(z) = \sigma_{2\rho K_{\mathcal{V}}}(z - z_\circ)$. The projection formula in Fact 4.4 gives $\nabla f(z) = \Pi_{2\rho K_{\mathcal{V}}}(z - z_\circ)$ and hence

$$\nabla f(z_\circ + 2\rho r_j) = 2\rho v_j \quad (0 \leq j \leq m). \tag{33}$$

Write $(x_j, y_j) := z_\circ + 2\rho r_j$ for the scaled and translated points. Since $r_j = r_{j-1} - (h_j/2)v_{j-1}$, (33) shows that GD on $f/2$ with stepsizes $h_1, \ldots, h_m$ visits (x_j, y_j) for $0 \leq j \leq m$. Their first coordinate is nonincreasing and at least

$$1 - 2\rho P \geq 1 - \frac{s}{a+s} > 0,$$

and their second coordinate is nondecreasing from 0 to $2\rho Q$.

Step 2. Verifying the zero region of f and $\partial_2 f$.

We next verify (8), which gives the transfer property of Φ_i after the scaling and translation in (10). For $v = (p, -q) \in \mathcal{V}$, we have $p, q \geq 0$ and $q \leq p/\epsilon$. The definition of a in Lemma 4.1 gives $ca \geq 2C_0$, so (31) implies

$$\langle v, -z_\circ \rangle = -p + 2\rho\big[p(\epsilon + P) + q(1 + Q)\big] \leq -p + \rho p(cs + 2C_0) \leq 0.$$

When $z_1 \leq 0$ and $z_2 \geq 0$, we also have $\langle v, z \rangle = pz_1 - qz_2 \leq 0$, so every affine piece $2\rho\langle v, z - z_\circ\rangle$ of ϕ is nonpositive. Thus $\phi = 0$ in this region, and Fact 4.4 gives $f = 0$ there. Since $f \geq 0$, its gradient is also zero there.

Set $\tau = z_\circ^{(2)} = 2\rho(Q+1) > 0$, so $0 \leq y_j \leq 2\rho Q = \tau - 2\rho$ for $0 \leq j \leq m$. For every $(w_1, -w_2) \in 2\rho K_{\mathcal{V}}$, the point $(w_1, 0)$ also belongs to $2\rho K_{\mathcal{V}}$, since $0 \leq w_1 \leq 2\rho c$ and $[0, 2\rho c] \times \{0\} \subset 2\rho K_{\mathcal{V}}$. If $z = (x, y)$ with $y \geq \tau$ and $w_2 > 0$, then

$$\|z - z_\circ - (w_1, -w_2)\|^2 - \|z - z_\circ - (w_1, 0)\|^2 = 2(y - \tau)w_2 + w_2^2 > 0.$$

Thus any point of $2\rho K_{\mathcal{V}}$ with negative second coordinate is farther from $z - z_\circ$ than a point with second coordinate zero. The projection, and hence $\nabla f(x, y)$, must have second coordinate zero, proving

$$\partial_2 f(x, y) = 0 \qquad (x \in \mathbb{R},\ y \geq \tau). \tag{34}$$

Step 3. The checkpoint update.

Just before the checkpoint update,

$$(x_m, y_m) = z_\circ + 2\rho r_m = (1 - 2\rho P, \tau - 2\rho), \qquad \nabla(f/2)(x_m, y_m) = \rho(\epsilon, -1).$$

The step of size b on $f/2$ gives

$$y_{m+1} = y_m + b\rho = \tau + \rho(b - 2).$$

The second coordinate therefore exceeds τ by $\rho(b - 2) > 0$.

□

4.3 Constructing the Global Hard Function

We now combine the local hard functions along the coordinate chain.

For gap i, apply Lemma 4.3 to $h_{t_{i-1}+1}, \ldots, h_{t_i - 1}$ and the checkpoint stepsize b_i. Write f_i, τ_i for the resulting function and threshold. Set

$$D_1 := 1, \quad \ell_1 := 0, \qquad D_{i+1} := D_i \frac{b_i - 2}{c(a + s_i)}, \quad \ell_{i+1} := D_i \tau_i \quad (1 \leq i \leq k). \tag{9}$$

Define the local hard function

$$\Phi_i(x) := D_i^2 f_i\left(\frac{x^{(i)} - \ell_i}{D_i}, \frac{x^{(i+1)}}{D_i}\right). \tag{10}$$

Thus the local initial point $(1, 0)$ corresponds to $(\ell_i + D_i, 0)$ in the chain. Under the scaling in (10), a GD step on $\Phi_i/2$ becomes a step of the same size on $f_i/2$ in the local coordinates $\big((x^{(i)} - \ell_i)/D_i,\, x^{(i+1)}/D_i\big)$. The two conditions in (8) give the transfer property (6), since a nonnegative differentiable function has zero gradient wherever its value is zero. Set $\delta = D_{k+1}/(1 + s_{k+1})$ and define the final component by $\Phi_{k+1}(x) := H_\delta(x^{(k+1)} - \ell_{k+1})$. The full objective is

$$F(x) := \frac{1}{2}\sum_{i=1}^{k} \Phi_i(x) + \frac{1}{2}\Phi_{k+1}(x). \tag{11}$$

The final Huber term also has zero gradient when $x^{(k+1)} \le \ell_{k+1}$. Each component is 1-smooth and each coordinate appears in at most two terms, so F is 1-smooth (see the smoothness and final-value calculation).

For gap i, set $m = t_i - t_{i-1} - 1$ and let (x_j, y_j) be the GD iterates on $f_i/2$ given by Lemma 4.3. Starting from the initial point e_1, we verify by induction over the gaps and their updates that, for $0 \le j \le m + 1$, all coordinates of $x_{t_{i-1}+j}$ beyond $i + 1$ are zero and

$$\big(x^{(i)}_{t_{i-1}+j}, x^{(i+1)}_{t_{i-1}+j}\big) = \big(\ell_i + D_i x_j, D_i y_j\big). \tag{12}$$

In each gap, we check the coordinate bounds needed to apply the transfer property (6) to the preceding and later components. Suppose gap i starts with $x^{(i)} = \ell_i + D_i$ and all later coordinates zero, as holds for $i = 1$. By Lemma 4.3, $\ell_i + D_i x_j > \ell_i$ and $0 \le D_i y_j < \ell_{i+1}$ for $0 \le j \le m$.

Before an update with $0 \le j \le m$, suppose (12) holds and all coordinates beyond $i + 1$ are zero. The second condition in (6), applied to Φ_{i-1}, then gives $\partial_{x^{(i)}} \Phi_{i-1} = 0$ when $i > 1$. Since $x^{(i+1)} < \ell_{i+1}$ and all coordinates beyond $i + 1$ are zero, the first condition in (6) and the analogous property of the final Huber term keep the next and all later components inactive. Thus only Φ_i contributes to the updates of $x^{(i)}$ and $x^{(i+1)}$, and no component changes the coordinates beyond $i + 1$. By the scaling in (10), the next GD step gives (12) at $j + 1$ and leaves all coordinates beyond $i + 1$ zero. Induction over the updates, including the checkpoint update, gives (12) at $j = m + 1$. By Lemma 4.3 and (9),

$$x^{(i+1)}_{t_i} = D_i y_{m+1} = D_i \tau_i + D_i \frac{b_i - 2}{c(a + s_i)} = \ell_{i+1} + D_{i+1},$$

with all coordinates beyond $i + 1$ still zero. This is the starting state for gap $i + 1$.

At x_{t_k}, the last coordinate has excess D_{k+1} above ℓ_{k+1}. During the remaining steps it stays above this threshold, so the preceding component contributes zero to its update when $k > 0$. Since all other components are nonnegative, the final Huber calculation gives

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

Since $x_0 = e_1$ and the origin minimizes F, the initial distance is 1. The normalization in (1) gives $\mathcal{R}_n(H) \geq 2F(x_n)$, which proves Lemma 4.1.

□

Smoothness and the final function value

For the objective F in (11), nonnegativity and (8) give $F(0) = \min F = 0$, since every $\ell_i \geq 0$. For $d = y - x$, the component smoothness inequalities give

$$\begin{aligned} F(y) &\leq F(x) + \langle \nabla F(x), d\rangle + \frac{1}{4}\left[\sum_{i=1}^{k}\left((d^{(i)})^2 + (d^{(i+1)})^2\right) + (d^{(k+1)})^2\right] \\ &\leq F(x) + \langle \nabla F(x), d\rangle + \frac{1}{2}\|d\|^2. \end{aligned}$$

Each coordinate appears at most twice in the bracketed sum, so F is 1-smooth.

For the final function value, use $\delta = D_{k+1}/(1+s_{k+1})$ as chosen in Section 4. For $t_k \leq t < n$, if $x_t^{(k+1)} - \ell_{k+1} \geq \delta$, the Huber term is affine and the transfer property (6) gives $\partial_{x^{(k+1)}} \Phi_k(x_t) = 0$ when $k > 0$. Thus the last coordinate decreases by $h_{t+1}\delta/2$. Starting from $x_{t_k}^{(k+1)} - \ell_{k+1} = D_{k+1} \geq \delta$, induction shows that the final iterates remain in this affine region and satisfy

$$\begin{aligned} x_t^{(k+1)} - \ell_{k+1} &= D_{k+1} - \frac{\delta}{2}\sum_{t'=t_k+1}^{t} h_{t'} \\ &\geq D_{k+1} - \frac{\delta s_{k+1}}{2} = \delta\left(1 + \frac{s_{k+1}}{2}\right) \geq \delta \qquad (t_k \leq t \leq n). \end{aligned}$$

All other components are nonnegative, so

$$F(x_n) \geq \frac{1}{2}\left[\delta\left(D_{k+1} - \frac{\delta s_{k+1}}{2}\right) - \frac{\delta^2}{2}\right] = \frac{D_{k+1}^2}{4(1+s_{k+1})}.$$

The same calculation includes $k = 0$, with $t_0 = \ell_1 = 0$ and $D_1 = 1$.

5. Proving (Almost) Tight Lower Bounds

It remains to choose checkpoints that make (7) give a uniform lower bound over all schedules. Following [JCY26, Section 3 and Appendix C], we derive a checkpoint cost from (7) and minimize it over admissible checkpoint sets for each schedule. The structure of a schedule that maximizes this minimum allows us to split at a checkpoint and apply induction to the two shorter sequences. The resulting recursion gives an $\Omega(n^{-p})$ lower bound for every $p > p_{\mathrm{sil}}$ by taking c sufficiently close to 1. The explicit constant a in Lemma 4.1 tends to infinity as $c \downarrow 1$. Balancing this growth against the improvement in p gives Theorem 1.1, as shown in Section 5.1.

For the anytime result, bounded stepsizes already give an $\Omega(n^{-1})$ Huber lower bound. Otherwise, we examine horizons at which the last stepsize exceeds all earlier ones, following [TFZH26, Section 4.1] and [JCY26, Section 4]. The Huber bound is strong when the total stepsize is small, while selecting a large last step as a checkpoint gives a complementary bound from (7). Together with the estimate on the total stepsize at these horizons in Lemma 5.6, these two bounds yield $\Omega(n^{-2p/(1+p)})$ infinitely often. Choosing p to approach p_{sil} as the horizon grows yields Theorem 1.2; see Section 5.2.

5.1 Proofs for the Non-anytime Lower Bound

With these preparations in hand, we now prove Theorem 1.1. We first give a high-level overview of the argument. For a fixed stepsize schedule H, each admissible checkpoint set gives a hard function through Lemma 4.1, and we choose checkpoints to make the resulting lower bound as large as possible. We seek a lower bound on this maximum that holds for every schedule H.

Fix $0 < \epsilon < 1/2$ and $\gamma > 0$, set $c := \sqrt{1 + 2\gamma + \epsilon^2}$, and take a as in Lemma 4.1. For a schedule $H = (h_1, \dots, h_n)$ and an admissible checkpoint set T (so $b_i > 2$ at every checkpoint), choose a parameter Λ and, following [JCY26, Section 3 and Appendix C], define the *checkpoint cost* by

$$Q_\Lambda(T; H) := c(\Lambda + s_{k+1}) \prod_{i=1}^{k} \frac{c(a + s_i)}{b_i - 2}, \qquad \Lambda \geq a. \tag{13}$$

Here Λ replaces the constant a in the factor for the gap after the last checkpoint. The lower bound (7) implies

$$\mathcal{R}_n(H) \geq \frac{c^2(\Lambda - 1)}{Q_\Lambda(T; H)^2}. \tag{14}$$

Proof of the checkpoint-cost lower bound

Applying the transfer bound (7) to T gives

$$\mathcal{R}_n(H) \geq \frac{1}{4(1 + s_{k+1})} \prod_{i=1}^{k} \left[\frac{b_i - 2}{c(a + s_i)}\right]^2 = \frac{c^2(\Lambda + s_{k+1})^2}{4(1 + s_{k+1}) Q_\Lambda(T; H)^2}.$$

Finally, the inequality

$$(\Lambda + s)^2 - 4(\Lambda - 1)(1 + s) = (s - \Lambda + 2)^2 \geq 0$$

with $s = s_{k+1}$ proves the claim.

For fixed H and Λ, maximizing the right-hand side of (14) is equivalent to minimizing $Q_\Lambda(T; H)$ over checkpoint sets. We therefore define

$$V_\Lambda(H) := \min_{T \text{ admissible}} Q_\Lambda(T; H).$$

A checkpoint set attaining $V_\Lambda(H)$ is called *tight*. For fixed H, this minimum admits the following recursive decomposition.

Lemma 5.1.

For every $H \in [0,\infty)^n$, $n \geq 1$, and $\Lambda \geq a$,

$$V_\Lambda(H) = \min\left\{ c\left(\Lambda + \sum_{j=1}^{n} h_j\right), \min_{\substack{1\leq t\leq n \\ t \text{ admissible}}} \frac{V_a(H_{1:t-1})V_\Lambda(H_{t+1:n})}{h_t - 2} \right\}. \quad (15)$$

Here $H_{i:j} := (h_i, \dots, h_j)$, $V_\mu(\varnothing) = c\mu$, and the inner minimum is $+\infty$ if there is no admissible checkpoint.

Proof.

For an admissible checkpoint set T and $t \in T$, let T_L and T_R be the selected indices before and after t, reindexed within the two subsequences. Then

$$Q_\Lambda(T; H) = \frac{Q_a(T_L; H_{1:t-1})Q_\Lambda(T_R; H_{t+1:n})}{h_t - 2}.$$

Minimizing over T_L, T_R and t accounts for every nonempty checkpoint set. Including the empty set gives (15).

□

To make the lower bound uniform over schedules of length n, define

$$U_n(\Lambda) := \sup_{H\in[0,\infty)^n} V_\Lambda(H).$$

Since $V_\Lambda(H) \leq U_n(\Lambda)$, we have

$$\mathcal{R}_n(H) \geq \frac{c^2(\Lambda - 1)}{V_\Lambda(H)^2} \geq \frac{c^2(\Lambda - 1)}{U_n(\Lambda)^2}.$$

Thus it suffices to bound $U_n(\Lambda)$ from above. We call a schedule attaining $U_n(\Lambda)$ a *maximizing schedule*.

Lemma 5.2 shows that, for every maximizing schedule, there exists a tight singleton checkpoint set. This structure lets us turn the recursive formula for $V_\Lambda(H)$ in Lemma 5.1 into recursive bounds on $U_n(\Lambda)$, which is the key step in the proof.

> **Lemma 5.2 (Maximizing schedules).**
>
> For every $n \geq 1$ and $\Lambda \geq a$, the supremum defining $U_n(\Lambda)$ is finite and attained. At every maximizing schedule, the empty set and a singleton are tight.

Log-submodularity and proof of Lemma 5.2

Notation.

We use the constants and conventions from Section 5.1, with $0 < \epsilon < 1/2$, $c = \sqrt{1 + 2\gamma + \epsilon^2}$, a as in Lemma 4.1, and $\Lambda \geq a$. For an empty subsequence, set $V_\mu(\varnothing) = c\mu$. A minimum over no indices is $+\infty$. An inadmissible checkpoint set has $Q_\Lambda(T; H) = +\infty$.

The recursion (15) describes $V_\Lambda(H)$ for a fixed schedule H, whereas we need to bound its largest possible value $U_n(\Lambda)$ over all schedules of length n. Lemma 5.2 gives a maximizing schedule H at which both the empty set and a singleton $\{t_*\}$ are tight. Splitting at t_* gives $U_n(\Lambda) = V_a(H_{1:t_*-1})V_\Lambda(H_{t_*+1:n})/(h_{t_*} - 2)$. The two numerator factors concern shorter schedules and are bounded by $U_{t_*-1}(a)$ and $U_{n-t_*}(\Lambda)$. Together with tightness of the empty set, these bounds let us apply induction on n, as carried out in the proof of Lemma 5.4.

To prove the lemma, we first show that $\log Q_\Lambda(T; H)$ is submodular in the checkpoint set T, following [JCY26, Lemma C.2].

> **Lemma 5.3 (Log-submodularity).**
>
> For any schedule $H \in [0, \infty)^n$ and any two admissible checkpoint sets S, T,
>
> $$Q_\Lambda(S; H)Q_\Lambda(T; H) \geq Q_\Lambda(S \cap T; H)Q_\Lambda(S \cup T; H). \tag{35}$$
>
> If S and T are disjoint and 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 stepsize $b > 2$ into a gap, and let $\ell, r \geq 0$ be the sums of stepsizes before and after it within that gap, respectively. Put

$$K := \begin{cases} a + r, & \text{if the gap ends at an existing checkpoint,} \\ \Lambda + r, & \text{if it is the final gap.} \end{cases}$$

Before insertion, this gap contributes $c(\ell + b + K)$ to Q_Λ. After insertion, it contributes $c(a + \ell)cK/(b - 2)$, so the cost is multiplied by

$$\frac{c(a + \ell)K}{(b - 2)(\ell + b + K)}.$$

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 $b > 2$ 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 set built from $S \cap T$ is final, the constant in K also decreases from Λ to $a \leq \Lambda$. Multiplying the insertion ratios gives

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

which proves (35). If S and T are disjoint and nonempty, the first step inserts the same checkpoint into the empty set and into T. Since every checkpoint already in T has stepsize greater than 2, at least one of ℓ and K is strictly smaller when inserting into T. The first comparison, and hence the final inequality, is therefore strict.

Finally, suppose that S and T are tight, so both attain $V := V_\Lambda(H)$. Every checkpoint set has value at least V, and (35) gives

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

Both factors in the middle must therefore equal V, so $S \cap T$ and $S \cup T$ are tight.[1] If S and T were nonempty and disjoint, the first inequality would be strict, giving $V^2 > V^2$, a contradiction.

□

We now prove Lemma 5.2. After establishing existence of a maximizing schedule, we show that it has a nonempty tight checkpoint set. By Lemma 5.3, the intersection of all such sets is again nonempty and tight. We then perturb the stepsizes at checkpoints in this intersection to show that it contains only one checkpoint, following [JCY26, Appendix C.6].

Proof of Lemma 5.2

Appending zero steps does not change V_Λ, so $U_m(\Lambda) \le U_j(\Lambda)$ whenever $m \le j$.

Step 1. Finiteness of $U_n(\Lambda)$.

We induct on n, starting from $U_0(\Lambda) = c\Lambda$. For an arbitrary schedule H, let $M = \max_t h_t$. If $M \le 2a + 4$, the empty checkpoint set gives

$$V_\Lambda(H) \le c\big[\Lambda + n(2a + 4)\big].$$

If $M > 2a + 4$, choose an index j with $h_j = M$ and split at j in (15). Using the empty set on the left, $M - 2 \ge M/2$, and $a < M$, we obtain

$$V_\Lambda(H) \le \frac{c\big(a + \sum_{t<j} h_t\big)}{M - 2} U_{n-j}(\Lambda) \le 2cn\, U_{n-1}(\Lambda).$$

Both bounds are independent of H, and $U_{n-1}(\Lambda)$ is finite by the induction hypothesis. Thus $U_n(\Lambda)$ is finite.

Step 2. Existence of a maximizing schedule.

We first show that V_Λ is continuous on $[0, \infty)^n$. For a fixed checkpoint set T, its cost is continuous wherever all selected stepsizes exceed 2. Near a finite schedule with a selected stepsize equal to 2, its cost tends to $+\infty$ along admissible schedules, since the numerator factors stay positive while a denominator factor tends to zero. Such a set cannot minimize the cost nearby, because the empty-set cost $c(\Lambda + \sum_t h_t)$ stays bounded. A set with a selected stepsize below 2 remains inadmissible in a neighborhood. Thus, near any fixed schedule, V_Λ is the minimum of the finitely many continuous costs of sets admissible at that schedule, proving continuity.

Choose schedules $H^{(k)}$ such that $V_\Lambda(H^{(k)}) \to U_n(\Lambda)$. We show that this sequence is bounded. Otherwise, pass to a subsequence along which every coordinate has a finite or infinite limit and at least one tends to $+\infty$. Let j be the first coordinate tending to $+\infty$, so the preceding coordinates stay bounded. For all sufficiently large k, splitting at j as in Step 1 gives

$$V_\Lambda(H^{(k)}) \le \frac{c\big(a + \sum_{t<j} h_t^{(k)}\big)}{h_j^{(k)} - 2} U_{n-j}(\Lambda) \longrightarrow 0.$$

This contradicts $V_\Lambda(H^{(k)}) \to U_n(\Lambda) \ge V_\Lambda(0, \dots, 0) = c\Lambda > 0$. Hence a subsequence converges to some $\bar{H} \in [0, \infty)^n$, and continuity gives $V_\Lambda(\bar{H}) = U_n(\Lambda)$.

Step 3. Tightness of the empty set.

Fix a maximizing schedule H, and write $V := V_\Lambda(H) = U_n(\Lambda)$. The remaining argument uses the following observation. If a perturbation strictly increased the cost of every tight set, all other costs would remain above V for sufficiently small perturbations, by the continuity and divergence arguments in Step 2. Since there are finitely many checkpoint sets, their new minimum would then exceed $V = U_n(\Lambda)$, a contradiction.

Suppose that the empty set were not tight. By Lemma 5.3, the intersection of all tight sets is tight, and hence nonempty. Choose an index j in this intersection and slightly decrease h_j, keeping $h_j > 2$. Every tight set selects j, so its cost depends on h_j only through the factor $1/(h_j - 2)$ and strictly increases. The observation above gives a contradiction, proving that the empty set is tight.

Step 4. Existence of a tight singleton.

There must also be a nonempty tight set. Otherwise, increasing any step slightly would increase the only tight cost, that of the empty set, contradicting the observation in Step 3.

Following [JCY26, Appendix C.6], let T_* be the intersection of all nonempty tight sets. By Lemma 5.3, the intersection of two nonempty tight sets is nonempty and tight, so the same holds for T_*. Suppose that T_* contains distinct indices i, j, and put $u = h_i - 2$ and $v = h_j - 2$, with $u \le v$. 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 i and j, so its cost depends on these stepsizes only through $1/(uv)$ and also increases. This contradicts the observation in Step 3. Thus T_* is a tight singleton.

□

---

1. Closure under unions and intersections is a standard property of submodular minimizers [Top78], applied here to $\log Q_\Lambda(\cdot; H)$.↩

For a tight singleton $\{t_*\}$ at a maximizing schedule, write $cW = U_n(\Lambda)$ and let cx, cy be the empty-set costs of the left and right subproblems. With $B = h_{t_*} - 2$, equality of the empty-set and singleton costs gives

$$W = x + y + B - (a - 2), \qquad WB = cxy. \tag{16}$$

To see heuristically how these identities give rise to the silver exponent, consider a balanced split $x = y$. Neglect the fixed term $a - 2$ in (16), eliminating B gives

$$W^2 - 2xW - cx^2 = 0, \qquad W = (1 + \sqrt{1 + c})x.$$

Combining two equal subproblems therefore multiplies the cost by $1 + \sqrt{1 + c}$. This suggests growth of order $n^{p(c)}$, where $p(c) = \log_2(1 + \sqrt{1 + c})$, tending to p_{sil} as $c \downarrow 1$.[1]

A rigorous argument in the proof of Lemma 5.4 based on (16) gives the following bound on $U_n(\Lambda)$.

Lemma 5.4.

Let $1 < c \le 2$, $p \in [p_{\mathrm{sil}} + c - 1, 3/2]$, and $\nu = 1/p$. Then, for every $n \ge 0$ and $\Lambda \ge a$,

$$\left[\frac{U_n(\Lambda)}{c}\right]^{\nu} \le a^{\nu} n + \Lambda^{\nu}. \tag{17}$$

A scalar inequality and proof of Lemma 5.4

We first prove the following scalar inequality, which will be used in the inductive proof of Lemma 5.4.

Lemma 5.5.

Let $1 < c \le 2$, $p \in [p_{\mathrm{sil}} + c - 1, 3/2]$, and $\nu = 1/p$. If $x, y \ge a$ and $B > 0$ satisfy (16), i.e.

$$W = x + y + B - (a - 2), \qquad WB = cxy,$$

then $W^{\nu} \le x^{\nu} + y^{\nu}$.

Proof.

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

$$\psi(W) = -(a-2)W \le 0.$$

Since ψ has exactly one positive root and $W, \overline{W} > 0$, it suffices to prove $\psi(\overline{W}) \ge 0$.

Set $z = (x/\overline{W})^\nu$, so that $1 - z = (y/\overline{W})^\nu$ and $z \in (0,1)$. Since $p = 1/\nu$, we have

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

Thus it remains to show $f(z) \le 1$ on $[0,1]$. Clearly $f(0) = 1$. Our choice of p gives

$$c \le 1 + p - p_{\text{sil}} \le 2^{2p} - 2^{p+1},$$

where the second inequality holds because the rightmost expression equals 1 at $p = p_{\text{sil}}$ and has derivative $2^{p+1}(\ln 2)(2^p - 1) > 1$ for $p > 1$. Consequently, $f(1/2) = 2^{1-p} + c2^{-2p} \le 1$.

On $(0, 1/2)$, the derivative is

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

Moreover,

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

Strict concavity shows that g either stays negative on $(0, 1/2)$ or changes sign once from negative to positive. Thus f either decreases throughout or first decreases and then increases, so its maximum on $[0, 1/2]$ occurs at an endpoint. By symmetry, $f(z) \le 1$ for every $z \in [0,1]$. Hence $\psi(\overline{W}) \ge 0$, so $W \le \overline{W}$, as required.

□

Proof of Lemma 5.4

We induct on n, simultaneously for all $\Lambda \geq a$. The case $n = 0$ is $U_0(\Lambda) = c\Lambda$. For $n \geq 1$, choose a maximizing schedule H. By Lemma 5.2, the empty set and a singleton $\{t_*\}$ are tight. Put

$$H_L := H_{1:t_*-1}, \qquad H_R := H_{t_*+1:n}, \qquad W := U_n(\Lambda)/c,$$

and

$$x := a + \sum_{t<t_*} h_t, \qquad y := \Lambda + \sum_{t>t_*} h_t, \qquad B := h_{t_*} - 2.$$

Then $x, y \geq a$ and $B > 0$. Tightness of $\varnothing$ and $\{t_*\}$ gives

$$W = x + y + B - (a - 2), \qquad WB = cxy,$$

so Lemma 5.5 gives $W^\nu \leq x^\nu + y^\nu$. The recursive decomposition (15) and the empty-set costs on the two sides give

$$cW \leq \frac{V_a(H_L)V_\Lambda(H_R)}{B} \leq \frac{c^2xy}{B} = cW.$$

Hence $V_a(H_L) = cx$ and $V_\Lambda(H_R) = cy$. Applying the induction hypothesis to the two shorter schedules gives

$$x^\nu \leq a^\nu(t_* - 1) + a^\nu, \qquad y^\nu \leq a^\nu(n - t_*) + \Lambda^\nu.$$

Combining these bounds proves $W^\nu \leq a^\nu n + \Lambda^\nu$, as required.

□

Now we are ready to prove Theorem 1.1.

Proof of Theorem 1.1

Fix $0 < \eta \leq 1/10$ and choose

$$p := p_{\mathrm{sil}} + \eta, \qquad \gamma := \frac{\eta}{4}, \qquad \epsilon := \left(\frac{\eta}{16}\right)^{p}, \qquad c := \sqrt{1 + 2\gamma + \epsilon^{2}}. \tag{18}$$

With a as defined in Lemma 4.1, set $\nu = 1/p$ and $\Lambda = an^{p}$. The parameter calculation verifies the parameter conditions of Lemma 5.4 and proves (20) below. By Lemma 5.4 and $\Lambda^{\nu} = a^{\nu} n$, $V_{\Lambda}(H) \le U_{n}(\Lambda) \le c(2a^{\nu}n)^{p} = c2^{p} a n^{p}$. Since $\Lambda - 1 \ge an^{p}/2$, (14) gives

$$\mathcal{R}_{n}(H) \ge \frac{1}{2^{2p+1} a}\, n^{-p} \qquad \text{for every } H \in [0,\infty)^{n}. \tag{19}$$

Under the choice (18),

$$\log a = O\big(\eta^{-1} \log(1/\eta)\big), \qquad 0 < \eta \le 1/10. \tag{20}$$

Consequently,

$$\mathcal{R}_{n}(H) \ge n^{-p_{\mathrm{sil}}} \exp\big[-\eta \log n - O\big(\eta^{-1}\log(1/\eta)\big)\big].$$

For n large enough, take $\eta = \sqrt{\log\log n / \log n} \le 1/10$. Then $\eta \log n = \sqrt{\log n \, \log\log n}$ and $\eta^{-1}\log(1/\eta) \le \sqrt{\log n \, \log\log n}$, so the preceding bound is at least $n^{-p_{\mathrm{sil}} - C\sqrt{\log\log n / \log n}}$ for an absolute constant C. This proves Theorem 1.1.

□

---

1. The one-sided Huber construction in (5) has $c = 2$, and its lower bound $\Omega(n^{-\log_2(1+\sqrt{3})})$ [JCY26] has exponent $p(2)$.↩

Verifying the parameter choice

Fix $0 < \eta \le 1/10$ and use the parameter choice (18). Then $p \in [p_{\mathrm{sil}}, 3/2]$ and $0 < \epsilon < 1/2$. Moreover, $\epsilon^{2} \le (\eta/16)^{2} \le \eta/6$, so

$$c^{2} = 1 + \frac{\eta}{2} + \epsilon^{2} \le 1 + \frac{2\eta}{3} \le \left(1 + \frac{\eta}{3}\right)^{2}.$$

Consequently $1 < c < 2$ and

$$c - 1 \le \frac{\eta}{3} < \eta = p - p_{\mathrm{sil}}.$$

Thus Lemma 5.4 applies.

The definition of a in Lemma 4.1 can be written as

$$R := \sqrt{\epsilon^2 + (1+\gamma)^2}, \qquad H_0 := \left[\frac{c(1+\gamma+R)}{\epsilon(\gamma+R)}\right]^{R/\gamma} - 1,$$
$$C_0 := \epsilon + \frac{H_0 + 1}{\epsilon}, \qquad a := \max\{2, 2C_0/c\}.$$

Under (18), c and R are bounded above and bounded away from zero by absolute constants, while

$$\log(1/\epsilon) = p\log(16/\eta) = O(\log(1/\eta)), \qquad \frac{R}{\gamma} = O(1/\eta).$$

Since $\log\left[\frac{c(1+\gamma+R)}{\epsilon(\gamma+R)}\right] = \log(1/\epsilon) + O(1) = O(\log(1/\eta))$, we obtain

$$\log(1+H_0) = \frac{R}{\gamma}\log\left(\frac{c(1+\gamma+R)}{\epsilon(\gamma+R)}\right) = O\big(\eta^{-1}\log(1/\eta)\big).$$

Since $C_0 = \epsilon + (H_0+1)/\epsilon$, the stated choice of a proves (20) with an absolute implied constant.

5.2 Proofs for the Anytime Lower Bound

To prove Theorem 1.2, we combine two consequences of Lemma 4.1. Selecting no checkpoints gives the one-dimensional Huber bound, which forces the sum of stepsizes to be large when the error is small. Selecting only the last step as a checkpoint, when its stepsize exceeds 2, gives a second lower bound in terms of that stepsize and the preceding sum. Following [TFZH26, Section 4.1] and [JCY26, Section 4], we apply these bounds at iterations when the stepsize is the largest so far.

To make this argument precise, we first introduce some notation. Fix an infinite nonnegative schedule $H = (h_t)_{t\ge 1}$. For each prefix $H_n = (h_1, \dots, h_n)$, denote its total stepsize, largest stepsize, and worst-case error by

$$S_n := \sum_{t=1}^{n} h_t, \qquad M_n := \max_{1\le t\le n} h_t, \qquad r_n := \mathcal{R}_n(H_n).$$

We call n a *record time* if $h_n = M_n$, and a *strict record time* if $n \ge 2$ and $h_n > M_{n-1}$.

For $p \in (p_{\mathrm{sil}}, p_{\mathrm{sil}} + 1/10]$, set $\nu := 1/p \in (0,1)$ and choose a, c as in the parameter calculation, with $\eta = p - p_{\mathrm{sil}}$. The two lower bounds above still depend on the schedule through S_n and M_n. To obtain a bound in terms of n, the next lemma uses (17) to bound S_n in terms of n and M_n when r_n is small.

Lemma 5.6 (Total stepsize at a record time).

For every $p \in (p_{\mathrm{sil}}, p_{\mathrm{sil}} + 1/10]$, there are constants $C_p, r_{0,p} > 0$ such that every record time n with $M_n \ge 4$ and $r_n \le r_{0,p}$ satisfies

$$S_n \le C_p n M_n^{1-\nu}. \tag{36}$$

Proof of Lemma 5.6: total stepsize at a record time

Take $r_{0,p} := 1/8$ and fix a record time n satisfying the lemma's assumptions. Write

$$M := M_n = h_n, \qquad H^- := H_{n-1}, \qquad V := V_a(H^-).$$

The prefix cost.

We first use the last checkpoint to bound the prefix cost V from below. Take an admissible checkpoint set T attaining $V_a(H^-)$ and append the final checkpoint n. The factorization in the proof of Lemma 5.1 gives

$$Q_a(T \cup \{n\}; H_n) = \frac{caV}{M-2}.$$

Since n is the last step, the final gap is empty. By (13), the product in (7) equals $[(M-2)/V]^2$. Hence Lemma 4.1 gives

$$r_n \geq \frac{(M-2)^2}{4V^2}, \qquad V \geq \frac{M-2}{2\sqrt{r_n}}. \tag{37}$$

Counting large steps.

To bound the sum of stepsizes in H^-, we count how many exceed each level. For $u \geq 0$, define

$$N_{H^-}(u) := \big|\{t < n : h_t > u\}\big|.$$

Now fix $u \in (2, M)$ and set $k := N_{H^-}(u)$. Use all stepsize indices $t < n$ with $h_t > u$ as checkpoints; they are admissible because $u > 2$. Write $b_1, \dots, b_k$ for their stepsizes in time order and let $H^{(0)}, \dots, H^{(k)}$ be the blocks between them, including the initial and final blocks. Write m_j for the length of $H^{(j)}$. Since the prefix has $n-1$ steps,

$$\sum_{j=0}^{k} (m_j + 1) = n. \tag{38}$$

Combine these checkpoints with admissible checkpoint sets attaining $V_a(H^{(j)})$ in all $k+1$ blocks. The decomposition in Lemma 5.1 gives

$$V_a(H^-) \leq V_a(H^{(k)}) \prod_{i=1}^{k} \frac{V_a(H^{(i-1)})}{b_i - 2}. \tag{39}$$

Since $V_a(H^{(j)}) \leq U_{m_j}(a)$, (17) gives

$$V_a(H^{(j)})^\nu \leq (ca)^\nu (m_j + 1) \qquad (0 \leq j \leq k).$$

Taking the νth power of (39), using $b_i - 2 > u - 2$, and then applying AM–GM with (38), we obtain

$$
\begin{aligned}
V^{\nu} &\le (ca)^{\nu}\left(\frac{(ca)^{\nu}}{(u-2)^{\nu}}\right)^{k}\prod_{j=0}^{k}(m_j+1)\\
&\le (ca)^{\nu}\left(\frac{(ca)^{\nu}}{(u-2)^{\nu}}\right)^{k}\left(\frac{n}{k+1}\right)^{k+1}.
\end{aligned}
\tag{40}
$$

Since $M-2 \ge u-2$, comparison of (37) and (40) gives

$$
(2\sqrt{r_n})^{-\nu} \le \left[\frac{n}{k+1}\left(\frac{ca}{u-2}\right)^{\nu}\right]^{k+1}.
$$

Since $r_n \le 1/8$, the left-hand side exceeds one. The quantity in brackets must therefore exceed one, giving

$$
N_{H^-}(u)+1 \le n\left(\frac{ca}{u-2}\right)^{\nu} \qquad (2<u<M). \tag{41}
$$

At a record time, every stepsize in H^- is at most M, so integrating N_{H^-} over $[0,M]$ recovers their sum. Using (41) for $2<u<M$ and $N_{H^-}(u)\le n$ for $0\le u\le 2$, we obtain

$$
\begin{aligned}
\sum_{t=1}^{n-1} h_t &= \int_0^M N_{H^-}(u)\,du\\
&\le 2n + n(ca)^{\nu}\int_2^M (u-2)^{-\nu}\,du\\
&\le \left(2+\frac{(ca)^{\nu}}{1-\nu}\right)nM^{1-\nu}.
\end{aligned}
\tag{42}
$$

Adding the last step.

It remains to bound the final stepsize $h_n = M$. Applying (17) to the $n-1$ steps of H^- and using (37) gives

$$
\frac{(M-2)^{\nu}}{(2\sqrt{r_n})^{\nu}} \le V^{\nu} \le (ca)^{\nu} n.
$$

Since $M \ge 4$ and $r_n \le 1/8 < 1/4$, it follows that

$$M^\nu \le (2ca)^\nu n, \qquad M \le (2ca)^\nu n M^{1-\nu}.$$

Adding this to (42) proves the lemma with

$$C_p := 2 + \frac{(ca)^\nu}{1-\nu} + (2ca)^\nu. \tag{43}$$

□

Proof of Theorem 1.2

Selecting no checkpoints in Lemma 4.1 gives

$$r_n \ge \frac{1}{4(1+S_n)}. \tag{44}$$

If H is bounded, then $S_n = O(n)$ and (44) proves the theorem. Suppose now that H is unbounded, and fix a strict record time n with $M_n = h_n \ge 4$. There are infinitely many such times.

At each such record time, applying Lemma 4.1 with the admissible checkpoint set $\{n\}$ gives

$$r_n \ge \frac{1}{4}\left[\frac{M_n - 2}{c(a + S_{n-1})}\right]^2. \tag{45}$$

Since $M_n - 2 \ge M_n/2$ and $S_n \ge M_n \ge 4$, we obtain

$$M_n \le 4c(a + S_{n-1})\sqrt{r_n} \le c(a+4)S_n\sqrt{r_n}.$$

If $r_n \le 1/8$, substituting this bound on M_n into Lemma 5.6 gives

$$S_n^\nu \le C_p[c(a+4)]^{1-\nu} n r_n^{(1-\nu)/2}.$$

The Huber bound (44) also gives $S_n \ge 1/(8r_n)$, so

$$1 \le 8^{\nu} C_p [c(a+4)]^{1-\nu} n r_n^{(1+\nu)/2}.$$

Consequently, at every strict record time with $M_n \ge 4$ and $r_n \le 1/8$,

$$r_n \ge c_p n^{-2p/(1+p)}, \qquad c_p := \left[8^{\nu} C_p [c(a+4)]^{1-\nu}\right]^{-2/(1+\nu)}. \tag{46}$$

To let p approach p_{sil}, we need to control the dependence of c_p on p. Recall that $\eta = p - p_{\mathrm{sil}}$. As $\eta \downarrow 0$, the quantities ν, $1-\nu$, and c remain bounded above and away from zero. The choices (43) and (46), together with (20), give

$$\begin{aligned} \log C_p,\ \log(1/c_p) &= O\big(\eta^{-1} \log(1/\eta)\big), \\ \frac{2p}{1+p} &= p_{\mathrm{any}} + O(\eta). \end{aligned} \tag{47}$$

At each sufficiently large strict record time, choose

$$\eta = \sqrt{\frac{\log\log n}{\log n}}.$$

If $r_n > r_{0,p} = 1/8$, the claimed bound holds for sufficiently large n. Otherwise, (46) and (47) give

$$r_n \ge n^{-p_{\mathrm{any}}} \exp\left\{-C\left(\eta \log n + \eta^{-1} \log(1/\eta)\right)\right\} \ge n^{-\left(p_{\mathrm{any}} + C'\sqrt{\log\log n / \log n}\right)}.$$

There are infinitely many strict record times, which proves Theorem 1.2.

6. Concluding Remarks

We proved lower bounds for GD with predetermined nonnegative stepsizes that match the silver exponents in both the finite-horizon and anytime settings. Bending the gradient during each gap improves the transfer between local hard functions and closes the polynomial gap to the upper bounds.

The remaining open questions are:

1. Can the lower bounds match the silver exponents for schedules that may include negative stepsizes? To our knowledge, the best known lower bounds for such schedules are $\Omega(n^{-1.6342})$ in the non-anytime case and $\Omega(n^{-1.2408})$ in the anytime case, as established in our earlier paper [YL26]. Extending the lower bounds proved here to such schedules remains open, although we believe that our approach can be adapted to handle negative stepsizes.

2. Can the $\sqrt{\log\log n/\log n}$ losses in the exponents of (2) and (3) be reduced or removed?[1] In the circular construction illustrated in Figure 4, setting $\gamma = 0$ would make every gradient vector v_j equal to $(\epsilon, -1)$. A local calculation similar to the Huber analysis in Section 3 gives the coefficient $\epsilon + \epsilon^{-1} \geq 2$ for s_i. This yields a suboptimal lower bound as in [JCY26]. We therefore need $\gamma > 0$, so $c = \sqrt{1 + 2\gamma + \epsilon^2}$ can approach 1 but cannot equal 1. For this construction, balancing the improvement as $c \downarrow 1$ against the growth of a leaves the $o(1)$ loss in the exponent $p_{\text{sil}} + o(1)$. Removing this loss may require a more refined local construction, or even an entirely new approach to constructing the global hard function.

---

1. We have obtained preliminary improvements by modifying our existing construction to choose gradient vectors along a logarithmic spiral rather than the circular arc in (21).↩

Bibliographic Information

```
@misc{ye2026silver,
  author = {Yuhan Ye and Kaizhao Liu},
  title  = {Silver Rate Is (Almost) Optimal for Gradient Descent},
  year   = {2026},
  note   = {Blog post},
  url    = {https://yeyuhanyyh.github.io/gd-silver-rate/}
}
```

For the full paper, cite the arXiv preprint:

```
@misc{ye2026silverrate,
  author        = {Yuhan Ye and Kaizhao Liu},
  title         = {Silver Rate Is (Almost) Optimal for Gradient Descent},
  year          = {2026},
  eprint        = {2609.09152},
  archivePrefix = {arXiv},
  primaryClass  = {math.OC},
  doi           = {10.48550/arXiv.2609.09152},
  url           = {https://arxiv.org/abs/2609.09152}
}
```

AI Disclosure

We carefully read Jung et al.'s paper [JCY26] and traced the difference between their exponent $\log_2(1+\sqrt{3})$ and the silver exponent $\log_2(1+\sqrt{2})$ to the factor 2 in their local transfer estimate. Their local Huber component has a fixed gradient direction. It changes both adjacent coordinates, so both changes reduce the margin that keeps the component active. We wondered whether we could reduce this loss by bending the local trajectory, gradually turning the descent direction toward $x^{(i+1)}$ as the remaining amplitude passes from $x^{(i)}$ to $x^{(i+1)}$. We shared this intuition with ChatGPT-6 Astra. Through several rounds of substantive interaction and detailed calculations by ChatGPT-6 Astra Ultra, we designed a smooth convex hard function whose local gradient rotates along a circular arc and whose gap coefficient approaches 1. Combining this construction with a refinement of Jung et al.'s recursive analysis [JCY26, Section 3] gave the present result. The paper is available as an arXiv preprint.

References

1. Jason M. Altschuler and Pablo A. Parrilo. Acceleration by Stepsize Hedging II: Silver Stepsize Schedule for Smooth Convex Optimization. 2023; Mathematical Programming version, DOI: 10.1007/s10107-024-02164-2.
2. Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Composing Optimized Stepsize Schedules for Gradient Descent. 2024, revised 2025. arXiv:2410.16249.
3. Jason M. Altschuler and Pablo A. Parrilo. Stepsize Hedging: an Alternative Mechanism for Accelerating Gradient Descent. 2026. arXiv:2605.31386.
4. Jianhao Ma and Yuxin Chen. A lower bound for stepsize-based acceleration of gradient descent. 2026. arXiv:2608.10418.
5. Chung-En Tsai. An Improved Lower Bound for Non-Anytime Gradient Descent. 2026. Blog post, first version August 21.
6. Yuhan Ye. An Omega(n^-1.6342) Lower Bound for (Non-Anytime) Gradient Descent. 2026. Blog post, first announced August 23; based on discussions with Kaizhao Liu.
7. Yuhan Ye and Kaizhao Liu. Improved Gradient Descent Lower Bounds Beyond Nesterov. 2026. arXiv:2609.02855.
8. Minchan Jung, Hanseul Cho, and Chulhee Yun. Stronger Lower Bounds for (Non-)Anytime Acceleration of Gradient Descent. 2026. arXiv:2609.04032.
9. Zihan Zhang, Jason Lee, Simon Du, and Yuxin Chen. Anytime Acceleration of Gradient Descent. 2025. Proceedings of the Thirty Eighth Conference on Learning Theory, PMLR 291:5991-6013.
10. Chung-En Tsai, Ilyas Fatkhullin, Liang Zhang, and Niao He. Lower Bounds for Anytime Acceleration of Gradient Descent. 2026. arXiv:2607.02053.
11. Yuhan Ye. An Omega(n^-1.2408) Lower Bound for Anytime Gradient Descent. 2026. Blog post, first announced August 23.
12. Peter J. Huber. Robust Estimation of a Location Parameter. The Annals of Mathematical Statistics 35(1):73–101, 1964.

13. Augustin-Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences 25:536–538, 1847.

14. 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.

15. Yurii Nesterov. Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, 2004. See Section 2.1.2 for the quadratic chain.

16. Yoel Drori and Marc Teboulle. Performance of First-Order Methods for Smooth Convex Minimization: A Novel Approach. Mathematical Programming 145:451–482, 2014.

17. Jungbin Kim. A Proof of the Exact Convergence Rate of Gradient Descent. 2024. arXiv:2412.04427.

18. 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), Article 12, 2025; first posted as arXiv:2309.07879 in 2023.

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

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

21. Jinho Bok and Jason M. Altschuler. Optimized methods for composite optimization: A reduction perspective. *Mathematical Programming*, 2026.

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

23. 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.

24. 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.

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

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

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

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

33. 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.

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

35. 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.

36. Guy Kornowski and Ohad Shamir. Open problem: Anytime convergence rate of gradient descent. In *Proceedings of the Thirty-Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 5335–5339. PMLR, 2024.

37. 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.

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

39. 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.

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

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

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

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

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

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

51. Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

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

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

54. Donald M. Topkis. Minimizing a submodular function on a lattice. *Operations Research*, 26(2):305–321, 1978.

55. Kaizhao Liu and Yuhan Ye. Silver Rate Is (Almost) Optimal for Gradient Descent: The Strongly Convex Case. 2026. arXiv:2609.26873.