

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

Kaizhao Liu* Yuhao Ye*
MIT MIT
mrzt@mit.edu yyh03@mit.edu

September 23, 2026

Abstract

We study gradient descent with predetermined nonnegative stepsizes on smooth strongly convex functions. Let $p_{\text{sil}} = \log_2(1 + \sqrt{2})$ and κ be the condition number. We prove the iteration lower bound

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

for both relative squared distance and relative function error, uniformly over $0 < \delta < 1$ and sufficiently large κ . This matches the polynomial exponent of κ for the Silver stepsize schedule established in [Altschuler and Parrilo \[2025b\]](#).

*Authors are listed in random order.

Contents

1	Introduction	3
1.1	Our Contributions	3
1.2	Additional Related Work	4
2	Overview of Proof Idea	6
2.1	A Review of Existing Techniques for the Smooth Convex Case	6
2.2	Additional Ingredients for the Strongly Convex Case	8
3	Local Hard Functions: Bending, Repairing, and Bridging	10
3.1	Bending	11
3.2	Repairing	12
3.3	Bridging	12
4	Selecting Checkpoints for Every Schedule	13
4.1	Choosing Checkpoints within a Block	13
4.2	Repairing Accumulated Unselected Steps	14
5	Assembling the Hard Function and Proving the Theorem	16
5.1	Realizing a Maximizing Checkpoint Set	16
5.2	Continuing the Construction for Every Horizon	18
5.3	Choosing the Bending Parameter	22
6	Concluding Remarks	22
A	Proofs for the Local Hard Functions	26
A.1	Proof of Lemma 3.1	26
A.2	Proof of Lemma 3.2	31
A.3	Proof of Lemma 3.3	32
B	Proof of Lemma 4.1	34
B.1	The scalar splitting inequality	34
B.2	Bounding the minimum auxiliary cost	35
B.3	Bounding total mass when the maximum checkpoint score is one	40
B.4	Deriving the checkpoint bounds	41
C	Proof of Lemma 4.2	45

1 Introduction

We study the limits of **gradient descent** (GD) for unconstrained smooth strongly convex minimization when its stepsizes are chosen in advance. For a nonnegative schedule $H = (h_1, \dots, h_n) \in [0, \infty)^n$, the iteration is

$$x_t = x_{t-1} - h_t \nabla F(x_{t-1}), \quad 1 \leq t \leq n.$$

Here, $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is μ -strongly convex with an L -Lipschitz gradient. $\kappa := L/\mu$ is the condition number.¹

Classical guarantees for constant-step GD follow by bounding the progress of each iteration separately [Polyak, 1963]; see also the textbooks by Polyak [1987], Nesterov [2004], and Bubeck [2015]. For quadratic objectives, Young [1953] showed that Chebyshev stepsizes reduce the squared distance to the minimizer by a factor $0 < \delta \leq 1/4$ in $O(\sqrt{\kappa} \log(1/\delta))$ iterations. The matching $\Omega(\sqrt{\kappa} \log(1/\delta))$ lower bound for predetermined stepsizes follows from the minimax property of Chebyshev polynomials. Classical acceleration augments the GD update with momentum or auxiliary sequences, as in Polyak’s heavy-ball method [Polyak, 1964] and Nesterov’s accelerated method [Nesterov, 1983].

Beyond quadratics, however, whether predetermined stepsizes alone could accelerate GD remained open for decades. Separate one-step bounds do not capture the performance of a varying stepsize schedule, leaving room for faster convergence by carefully coordinating short and long steps [Altschuler, 2018]. The recursive *silver stepsize schedule* of Altschuler and Parrilo [2025a] improves the classical function-value convergence rate on general smooth convex objectives from $O(n^{-1})$ to $O(n^{-p_{\text{sil}}})$, where

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

Our previous work [Ye and Liu, 2026b] proves an $\Omega(n^{-p_{\text{sil}} - o(1)})$ lower bound for every predetermined nonnegative schedule, confirming their conjecture that this polynomial exponent is optimal.

For strongly convex objectives with condition number κ , constant-step GD reduces the squared distance to the minimizer by a factor $0 < \delta \leq 1/4$ in $O(\kappa \log(1/\delta))$ iterations, whereas Nesterov’s accelerated method achieves $O(\sqrt{\kappa} \log(1/\delta))$ [Nesterov, 1983]. The latter matches the classical $\Omega(\sqrt{\kappa} \log(1/\delta))$ first-order oracle lower bound [Nemirovsky and Yudin, 1983]. Nesterov [2004] gives a comprehensive treatment of these classical results. For GD with predetermined stepsizes, Altschuler and Parrilo [2025b] introduced a silver schedule for strongly convex objectives that achieves $O(\kappa^{1/p_{\text{sil}}} \log(1/\delta))$ iterations and conjectured its optimality. Here $1/p_{\text{sil}} \approx 0.78644$. The classical oracle lower bound $\Omega(\sqrt{\kappa} \log(1/\delta))$, however, leaves a polynomial gap to the silver upper bound. Hence, the central question is whether we can close this gap by proving a stronger lower bound.

1.1 Our Contributions

In this paper, we prove an iteration lower bound of $\Omega(\kappa^{1/p_{\text{sil}} - o(1)} \log(1/\delta))$ for GD with predetermined nonnegative stepsizes. This matches the silver exponent in the upper bound of Altschuler and Parrilo [2025b].

To state our results, let $\mathcal{F}_{\mu, L}(\mathbb{R}^d)$ denote the class of differentiable μ -strongly convex functions on $\mathbb{R}^d$ with L -Lipschitz gradients. For each F in this class, write x_* for its unique minimizer. For a normalized¹ schedule H , which may depend on κ and the horizon n , the (worst-case) relative squared-distance and function-value errors are

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

¹Dividing the objective by L and multiplying each stepsize by L preserves the iterates. We therefore normalize $L = 1$ and $\mu = 1/\kappa$ throughout.

Both errors satisfy the following lower bound.

Theorem 1.1 (Error 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 $H \in [0, \infty)^n$,*

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

Define the iteration complexities for relative squared distance and relative function error by

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

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

$$T_D(\kappa, \delta), T_E(\kappa, \delta) \geq c \frac{\kappa^{1/p_{\text{sil}}}}{e^{C\sqrt{\log \kappa}} \log \kappa} \log \frac{1}{\delta}. \quad (3)$$

For $0 < \delta \leq 1/4$, the $\Omega(\kappa^{1/p_{\text{sil}} - o(1)} \log(1/\delta))$ iteration lower bound in [Corollary 1.2](#) matches the $O(\kappa^{1/p_{\text{sil}}} \log(1/\delta))$ upper bound of [Altschuler and Parrilo \[2025b\]](#) up to a subpolynomial factor in κ .² The lower and upper bounds contain the same multiplicative factor $\log(1/\delta)$. [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	$\Omega(\kappa \log(1/\delta))$ [Polyak, 1963]	$O(\kappa \log(1/\delta))$ [Polyak, 1963]
Quadratic	Predetermined-step GD	$\Omega(\sqrt{\kappa} \log(1/\delta))$ [Young, 1953]	$O(\sqrt{\kappa} \log(1/\delta))$ [Young, 1953]
General	Deterministic first-order methods	$\Omega(\sqrt{\kappa} \log(1/\delta))$ [Nemirovsky and Yudin, 1983]	$O(\sqrt{\kappa} \log(1/\delta))$ [Nesterov, 1983]
General	Predetermined-step GD	$\Omega\left(\frac{\kappa^{1/p_{\text{sil}}} \log(1/\delta)}{e^{C\sqrt{\log \kappa}} \log \kappa}\right)$ This work	$O(\kappa^{1/p_{\text{sil}}} \log(1/\delta))$ [Altschuler and Parrilo, 2025b]

Table 1: Iteration complexity on smooth strongly convex objectives.³

1.2 Additional Related Work

The special case of quadratic optimization. [Young \[1953\]](#) achieved the optimal $O(\sqrt{\kappa} \log(1/\delta))$ complexity by choosing stepsizes as reciprocals of the roots of a shifted and scaled Chebyshev polynomial. Conjugate gradients use a three-term recurrence, with coefficients that adapt to the particular problem instance [\[Hestenes and Stiefel, 1952\]](#). 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

²The gap to the silver upper bound is the factor $e^{C\sqrt{\log \kappa}} \log \kappa = \kappa^{o(1)}$ in (3).

³All bounds hold for sufficiently large κ and concern reducing squared distance to the minimizer by a factor $0 < \delta \leq 1/4$.

quadratics [Polyak, 1964]. 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 stepsize schedules [Pronzato and Zhigljavsky, 2011, Zhigljavsky et al., 2013, Kalousek, 2017]. Once the stepsizes are chosen, their order does not affect GD’s final iterate on quadratic objectives. However, the order still affects numerical stability, motivating the study of stable Chebyshev orderings [Lebedev and Finogenov, 1971, Agarwal et al., 2021]. For general nonquadratic objectives, the order can affect convergence itself.

General smooth (strongly) convex optimization. Beyond quadratics, the first provable gains from predetermined stepsize schedules were established in the strongly convex setting by Altschuler [2018, Chapter 8], who analyzed two- and three-step schedules. Repeating these short schedules improves constants in the classical $O(\kappa \log(1/\delta))$ iteration complexity. Tools for analyzing and designing finite-horizon schedules include the performance estimation framework [Drori and Teboulle, 2014], smooth strongly convex interpolation [Taylor et al., 2017], and numerical branch-and-bound search [Das Gupta et al., 2024]. The remaining challenge was to improve the exponent of κ .

The silver stepsize schedule of Altschuler and Parrilo [2025b] improves this complexity from $O(\kappa \log(1/\delta))$ to $O(\kappa^{1/p_{\text{sil}}} \log(1/\delta))$. They conjectured that the exponent $1/p_{\text{sil}}$ is optimal among all predetermined stepsize schedules. For separable strongly convex objectives, Altschuler and Parrilo [2024] showed that arcsine randomization achieves the asymptotic distance factor $(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ almost surely. We refer to Altschuler and Parrilo [2026] for a broader overview.

In the general smooth convex setting, early work studied periodic long steps [Grimmer, 2024]. A breakthrough came when Altschuler and Parrilo [2025a] introduced the silver stepsize schedule, achieving the first $O(n^{-p_{\text{sil}}})$ convergence rate for function-value error in gradient descent.⁴ Using Altschuler and Parrilo’s recursive gluing technique, Grimmer et al. [2025b] subsequently improved the constant in the function-value bound and extended the silver rate to squared gradient norm. For $n = 2^k - 1$, Wang et al. [2026] determined the exact worst-case function-value bound for the original silver schedule. Grimmer et al. [2025a] and Zhang and Jiang [2026] independently gave rules for combining shorter schedules to obtain the silver rate for arbitrary prescribed horizons. Under these symmetric composition rules, Liu et al. [2026] proved that recursively joining schedules whose lengths differ by at most one minimizes the worst-case function-value bound at each horizon. Related developments include anytime acceleration [Zhang et al., 2025] and accelerated proximal gradient methods [Bok and Altschuler, 2025].

Lower bounds. Beyond the classical oracle lower bounds, later work has studied specific algorithm classes and exact bounds for a prescribed number of oracle calls. Building on the framework of Arjevani et al. [2016], Arjevani and Shamir [2016] studied lower bounds for linear iterative methods with predetermined coefficients, including momentum methods. For deterministic first-order methods, Drori [2017] established an exact worst-case lower bound on function-value error for smooth convex minimization. In the smooth strongly convex setting, Drori and Taylor [2022] established an exact worst-case lower bound on distance to the minimizer. The method of Taylor and Drori [2023] attains this bound using auxiliary iterates.

The exact oracle bounds also apply to GD, but they do not determine its optimal rate under predetermined stepsizes. For smooth convex objectives, Grimmer et al. [2025a] proved that the exponent p_{sil} is optimal for function-value error within their recursively generated class of schedules. More recent lower bounds apply to arbitrary predetermined nonnegative schedules. Ma and Chen [2026] proved an $\Omega(n^{-1.9319})$ function-value lower bound for predetermined nonnegative stepsizes, giving the first polynomial improvement over

⁴On strongly convex objectives, repeating a suitably chosen silver block of length $O(\kappa^{1/p_{\text{sil}}})$ gives a constant-factor contraction in both squared distance and function error per block, yielding $O(\kappa^{1/p_{\text{sil}}} \log(1/\delta))$ iterations for $0 < \delta \leq 1/4$; see Altschuler and Parrilo [2025a, Theorem 1 and Corollary 2].

the classical $\Omega(n^{-2})$ bound. Tsai [2026] sharpened this to $\Omega(n^{-\sqrt{3}})$ using the same construction. Ye and Liu [2026a] improved this lower bound to $\Omega(n^{-1.6342})$, even for possibly negative stepsizes. Jung et al. [2026] subsequently introduced a family of hard functions that gives an $\Omega(n^{-\log_2(1+\sqrt{3})})$ lower bound for positive schedules. Our previous work [Ye and Liu, 2026b] further refined this construction to obtain an $\Omega(n^{-p_{\text{sil}}-o(1)})$ lower bound. This shows that the polynomial exponent p_{sil} in the $O(n^{-p_{\text{sil}}})$ 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 Section 2.1, we first recall three ingredients from the smooth convex setting, introducing the notation that will also describe our strongly convex construction. In Section 2.2, we explain the new ingredients needed for proving Theorem 1.1.

2.1 A Review of Existing Techniques for the Smooth Convex Case

The Checkpoint Method [Ma and Chen, 2026]. Given a stepsize schedule $H = (h_1, \dots, h_n)$, we select a set of update indices $T = \{t_1 < \dots < t_k\}$ 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 the final error. 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

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

Here, b_i is the stepsize at checkpoint t_i , and s_i is the *mass* (the sum of the stepsizes) of the preceding gap, with s_{k+1} representing the mass after the final checkpoint.

As an example, Ma and Chen [2026] construct a smooth convex hard function designed to keep the gradient constant within each gap. Consequently, the displacements within a gap simply accumulate and depend only on the gap mass s_i . Each checkpoint b_i is then used to force GD to explore a new dimension. This strategy is reminiscent of Nesterov’s lower-bound construction [Ye and Liu, 2026a, Appendix A]: to sustain a strong lower bound, GD cannot be allowed to make continuous progress in the same dimension throughout the run. 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. Ye and Liu [2026a] give 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 [Jung et al., 2026]. The score depends on both the checkpoint set and the hard-function construction. Improving the construction can therefore strengthen the lower bound beyond what checkpoint selection alone achieves.

For instance, the hard function in [Ma and Chen \[2026\]](#) is constructed globally, which introduces complex dependencies between neighboring gaps into the resulting score. [Jung et al. \[2026\]](#) instead propose a decoupled construction that yields a much simpler score by summing two-dimensional local hard components and a final one-dimensional term:

$$F(x) := \frac{1}{2} \sum_{i=1}^k \Phi_i(x^{(i)}, x^{(i+1)}) + \frac{1}{2} H_{\delta_{\text{term}}}(x^{(k+1)} - \ell_{k+1}), \quad x_0 = e_1, \quad x_* = 0.$$

We call each $\Phi_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ a *local hard function*, or simply a *component* of F . It is nonnegative, convex, and 1-smooth, and acts on the adjacent coordinates $(X, Y) = (x^{(i)}, x^{(i+1)})$. We call X and Y the *input* and *output* coordinates, respectively. For $i < k$, Y serves as the input coordinate of Φ_{i+1} ; for $i = k$, it is the coordinate on which the final one-dimensional term acts. 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 the current active component: all later components have zero gradient, and earlier components do not alter the gradient supplied by Φ_i . Each checkpoint activates the next local hard function, or the final one-dimensional term at the last checkpoint.

To fulfill these requirements, each Φ_i is chosen to vanish when $X \leq \ell_i$ and $Y \geq 0$, where the *activation threshold* $\ell_i \geq 0$ is introduced to keep Φ_i inactive until its input reaches the required level. At the beginning of gap i ,

$$(x_{t_{i-1}}^{(i)}, x_{t_{i-1}}^{(i+1)}) = (\ell_i + D_i, 0), \quad (4)$$

where D_i is the *incoming amplitude*, the amount by which the input exceeds its threshold; the next coordinate is initially zero. Initially, $\ell_1 = 0$ and $D_1 = 1$. During this gap, the construction keeps $x^{(i+1)} \leq \ell_{i+1}$, while all further coordinates remain zero. Thus, the next component's input stays at most its activation threshold, so it remains inactive and cannot interfere with Φ_i . The same holds for all later components. At checkpoint t_i , coordinate $i+1$ rises to $\ell_{i+1} + D_{i+1}$, crossing the threshold. For $i < k$, this activates Φ_{i+1} for the next gap. For $i > 1$, the preceding component Φ_{i-1} shares coordinate i with Φ_i . The construction also ensures that the contribution of Φ_{i-1} to coordinate i does not alter the prescribed gradient of Φ_i during the gap or at the checkpoint query. Since no earlier component depends on coordinate $i+1$, these conditions let us determine the increase of coordinate $i+1$ from Φ_i alone. The *transfer ratio*

$$G_i := \frac{D_{i+1}}{D_i} \quad (5)$$

measures the amplitude passed to the next gap.

For the one-sided Huber components of [Jung et al. \[2026\]](#), the construction gives $G_i = b_i / [2(2 + s_i)]$, which depends only on the checkpoint stepsize b_i and the preceding gap mass s_i . The final term $H_{\delta_{\text{term}}}$ is a one-sided Huber function, with parameter $\delta_{\text{term}} = D_{k+1} / (1 + s_{k+1})$, which turns the remaining amplitude in coordinate $k+1$ into a function-value lower bound, yielding

$$\frac{F(x_n) - F(x_*)}{\|x_0 - x_*\|^2} \geq \frac{1}{4(1 + s_{k+1})} \prod_{i=1}^k G_i^2.$$

Ultimately, this structural decoupling simplifies the identification of the maximizing checkpoint set and yields the improved $\Omega(n^{-\log_2(1+\sqrt{3})})$ lower bound.

Bending the Local Trajectory [[Ye and Liu, 2026b](#)]. The constructions of [Ma and Chen \[2026\]](#) and [Jung et al. \[2026\]](#) keep the gradient constant during each gap. In contrast, [Ye and Liu \[2026b\]](#) change its direction to limit the increase of the output coordinate before the checkpoint while retaining a large increase at the

checkpoint update. This improves the coefficient of the gap mass in the transfer bound from 2 toward 1, yielding the lower bound $\Omega(n^{-p_{\text{sil}}-o(1)})$.

A bending transfer prescribes the updates of both $X = x^{(i)}$ and $Y = x^{(i+1)}$. When $i > 1$, the preceding component Φ_{i-1} must therefore contribute zero to the update of X . We require $\partial_2 \Phi_i(X, Y) = 0$ whenever $Y \geq \ell_{i+1}$, independently of X , where ∂_2 denotes the derivative in the output coordinate. During the next gap, the shared coordinate $x^{(i+1)}$ stays above ℓ_{i+1} , so Φ_i contributes no gradient to the coordinates used by Φ_{i+1} . Note that Φ_i may still have a nonzero derivative in its first coordinate, and full inactivity is unnecessary.

2.2 Additional Ingredients for the Strongly Convex Case

For the strongly convex case, we rescale the sum of local hard functions from [Section 2.1](#) and add a quadratic term:

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

Here κ is the condition number, and the k local hard functions $\Phi_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ are nonnegative, convex, 1-smooth, and vanish at zero. Each coordinate appears in at most two of the Φ_i , so the coefficient in (6) makes F 1-smooth and $1/\kappa$ -strongly convex, with minimizer $x_* = 0$. The initial point e_1 is the first coordinate vector. We retain the thresholds ℓ_i , amplitudes D_i , and transfer ratios $G_i = D_{i+1}/D_i$ from [Section 2.1](#). A lower bound on $|x_n^{(k+1)}|$ yields a distance lower bound and, through the quadratic term, a function-value lower bound.

Threshold-to-Amplitude Ratio and Quadratic Decay. The *threshold-to-amplitude ratio* ℓ_i/D_i compares the activation threshold ℓ_i with the input's excess D_i above it immediately after the preceding checkpoint. This ratio determines how much quadratic decay the next transfer can tolerate. Over gap i , quadratic contraction alone multiplies the input by χ_i and leaves excess $D_i \eta_i$ above its fixed threshold, where

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

As this excess shrinks, the threshold becomes larger relative to the available amplitude. When ℓ_i/D_i is large, even a small amount of contraction can exhaust the excess needed for the next transfer. We therefore need to control quadratic decay and, when necessary, introduce checkpoints that restore a bounded outgoing threshold-to-amplitude ratio.

Controlling Decay by Blocks and Additional Checkpoints. To control the quadratic decay, we divide the schedule into blocks of at most m updates and choose checkpoints block by block. Writing s_i for the total stepsize in gap i , the estimate $\chi_i \geq 1 - s_i/\kappa$ for $s_i < \kappa$ suggests keeping the unselected gap mass small relative to κ . Heuristically, [Ye and Liu \[2026b\]](#) gives a gap-mass scale of $m^{p_{\text{sil}}+o(1)}$. Therefore, quadratic decay is controlled when this mass is small relative to κ , suggesting the choice of $m \lesssim \kappa^{1/p_{\text{sil}}-o(1)}$. With the decay controlled in each block, we aim to keep the product of transfer ratios bounded below by an absolute constant, detailed in [Section 5.1](#). Over all blocks with nonempty maximizing checkpoint sets, these bounds contribute a factor of at least $\exp[-O(1+n/m)]$ to the retained amplitude. To obtain a better lower bound, we therefore want to choose m as large as possible, saturating the aforementioned constraint.

However, a block boundary does not end a gap. If the optimal checkpoint selection leaves several blocks without checkpoints, their unselected mass continues accumulating across blocks. If a block has no nonempty maximizing checkpoint set and the accumulated unselected mass reaches the repair threshold, we

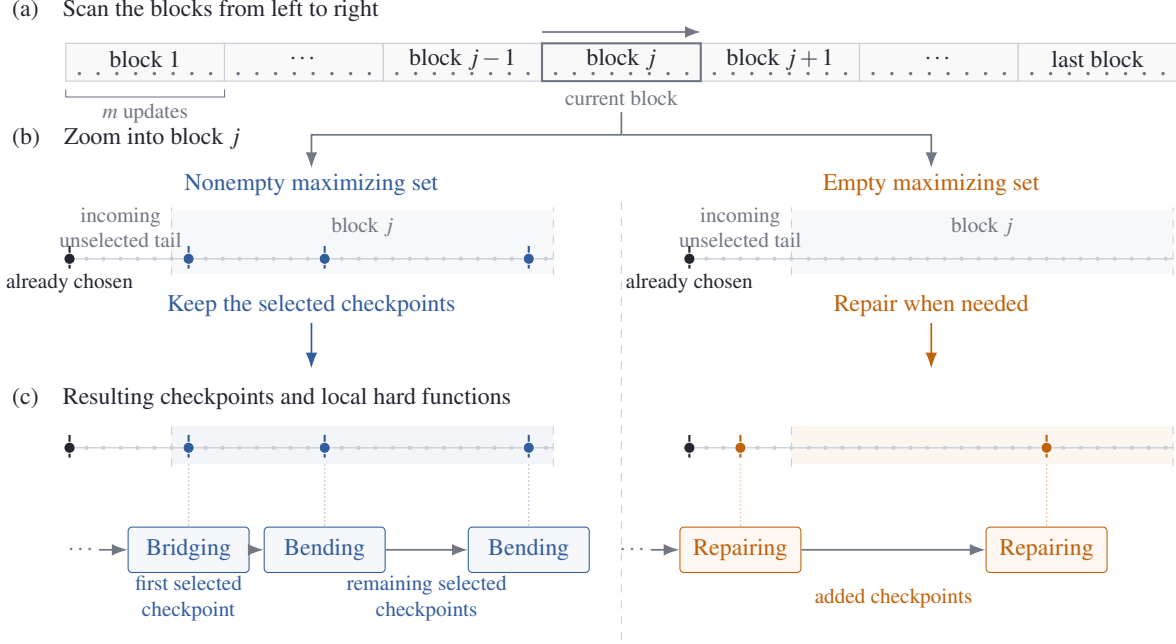


Figure 1: An overview of checkpoint selection and the construction of hard function in one block. Black marks the latest previously chosen checkpoint; blue marks a nonempty maximizing set. For an empty maximizing set, the block’s steps remain unselected if the accumulated unselected mass is below the repair threshold; otherwise, repair adds one or two orange checkpoints. Repair checkpoints may lie in the incoming tail. The two branches show alternative outcomes, and two repair checkpoints are illustrated. The unselected tail is carried over to the next block.

perform a repair by selecting one or two additional checkpoints. Thus, at most one repair is performed per block. The Huber repair components in [Section 3.2](#) restore a bounded threshold-to-amplitude ratio, at a controlled loss of amplitude. [Section 4.2](#) details the precise checkpoint choice and transfer ratio. As this penalty is incurred at most once per block, [Lemma 5.2](#) allows us to control the overall loss of the incoming amplitude across the entire schedule.

Bridging from Repairing to Bending. After repairing the threshold-to-amplitude ratio, we want to resume bending to retain its sharper dependence on κ . However, a Huber repair component may still contribute to the coordinate that serves as the next component’s input, whereas bending requires control of both coordinate updates. We need a component that tolerates this earlier contribution during its own gap and prevents its own output derivative from affecting the following bending component. The Huber bridge supplies this compatibility, allowing repairs and bending to coexist in one fixed hard function. Its construction is given in [Section 3.3](#).

Organization. The rest of this paper is organized as follows. [Section 3](#) gives the three types of local hard functions and bounds their transfer ratios and outgoing threshold-to-amplitude ratios. [Section 4](#) selects the checkpoints and details the repairing of threshold-to-amplitude ratio by adding checkpoints. [Section 5](#) assembles one hard function and proves the main theorem. [Figure 1](#) gives an overview of the checkpoint selection procedure in [Section 4](#) and the construction of the hard function in [Section 5](#).

3 Local Hard Functions: Bending, Repairing, and Bridging

In this section, we present three types of local hard functions 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. We construct the components in checkpoint order, starting from $(\ell_1, D_1) = (0, 1)$. Each construction takes the incoming pair (ℓ_i, D_i) , chooses Φ_i and the next threshold ℓ_{i+1} , and determines D_{i+1} from the value of $x^{(i+1)}$ after the checkpoint update. For the current component Φ_i , write $(X, Y) = (x^{(i)}, x^{(i+1)})$. Within this gap, j indexes the local updates and h_j denotes their stepsizes. Fix an input threshold $\ell_i \geq 0$, an incoming amplitude $D_i > 0$, and gap stepsizes $0 \leq h_j < \kappa$ for $1 \leq j \leq m_{\text{gap}}$. 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 $h_{m_{\text{gap}}+1} = b_i > 0$, with gradient evaluated at the checkpoint query $(X_{m_{\text{gap}}}, Y_{m_{\text{gap}}})$. We construct the current component Φ_i and choose an outgoing threshold $\ell_{i+1} \geq 0$ so that

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

For $i < k$, the resulting pair (ℓ_{i+1}, D_{i+1}) serves as the incoming data for constructing Φ_{i+1} , with activation threshold ℓ_{i+1} . As explained after (4), $Y_j \leq \ell_{i+1}$ gives $\nabla \Phi_{i+1}(Y_j, 0) = 0$. All subsequent components are evaluated at $(0, 0)$ and also have zero gradient, so none affects the checkpoint gradient. The checkpoint update raises Y to $\ell_{i+1} + D_{i+1}$, activating Φ_{i+1} . For $i = k$, this update instead sets the value of the final coordinate $x^{(k+1)}$. We first describe the local updates and quantify the loss caused by quadratic contraction during each gap.

Local updates with the preceding component. With later components inactive, the updates of (X, Y) come from Φ_i , the quadratic term, and possibly the preceding component Φ_{i-1} when $i > 1$. The latter depends on $X = x^{(i)}$, its second coordinate, but not on $Y = x^{(i+1)}$. If its contribution to the X update is zero, (6) gives

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

More generally, Φ_{i-1} contributes $h_j f_j$ to the update of X , where $f_j = -(1 - 1/\kappa)\partial_2 \Phi_{i-1}/2 \geq 0$ and the derivative is evaluated at the query before update j . Our local analysis allows arbitrary $f_j \geq 0$ in these updates:

$$\begin{pmatrix} X_j \\ Y_j \end{pmatrix} = \left(1 - \frac{h_j}{\kappa}\right) \begin{pmatrix} X_{j-1} \\ Y_{j-1} \end{pmatrix} - \frac{(1 - 1/\kappa)h_j}{2} \nabla \Phi_i(X_{j-1}, Y_{j-1}) + h_j \begin{pmatrix} f_j \\ 0 \end{pmatrix}. \quad (10)$$

Both recurrences run over $1 \leq j \leq m_{\text{gap}} + 1$. For the first component Φ_1 , $f_j = 0$. If $\partial_2 \Phi_{i-1} = 0$ whenever its second coordinate $X \geq \ell_i$, then $X_{j-1} \geq \ell_i$ makes $f_j = 0$.

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

$$1 - \frac{s_i}{\kappa} \leq \chi_i \leq 1, \quad \eta_i \geq 1 - \frac{s_i}{\kappa}(1 + \ell_i/D_i). \quad (11)$$

For an empty gap, $s_i = 0$ and $\chi_i = \eta_i = 1$. Each local lemma below assumes $\eta_i > 0$.

We now construct the three types of local hard functions. For each Φ_i , our goal is to bound D_{i+1}/D_i , the transfer ratio in (5), from below to retain amplitude, and ℓ_{i+1}/D_{i+1} from above to limit quadratic contraction loss in the next gap, as quantified in (11).

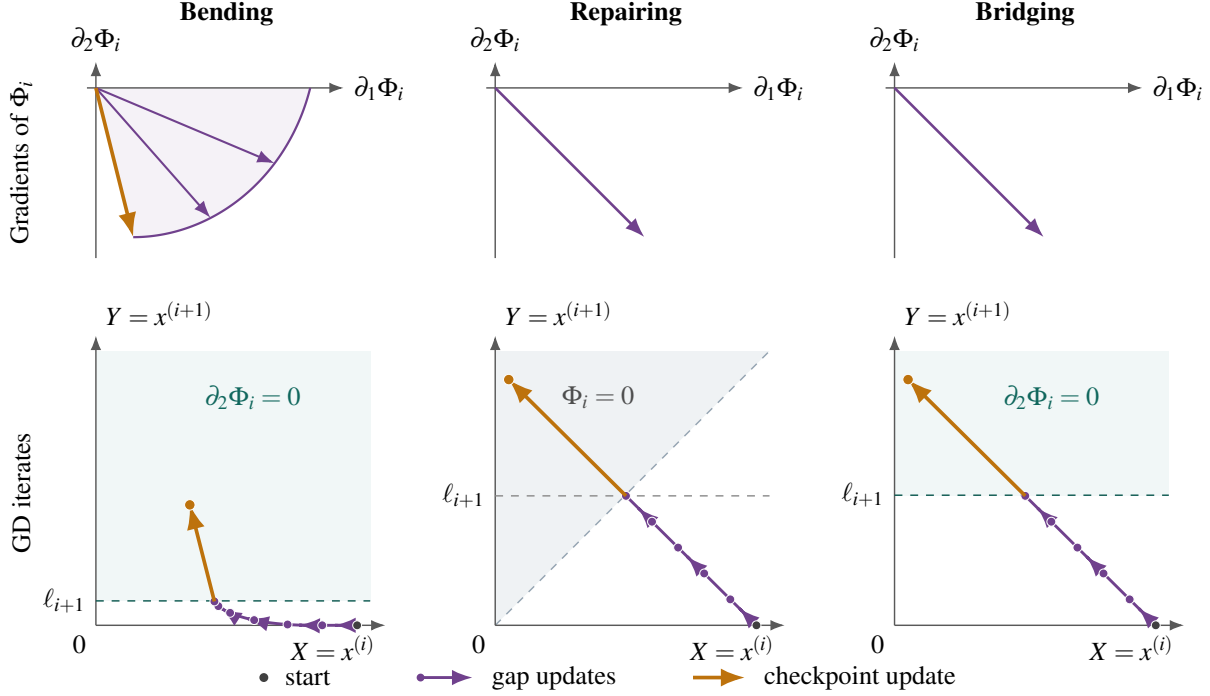


Figure 2: Scaled gradients of Φ_i (top) and example GD trajectories (bottom). For visual clarity, the trajectories omit contributions from the quadratic term and preceding component Φ_{i-1} . Orange marks checkpoints. Gray indicates $\Phi_i = 0$; green shading indicates $\partial_2 \Phi_i = 0$.

3.1 Bending

We adapt the bending idea of [Ye and Liu \[2026b\]](#) reviewed in [Section 2.1](#), accounting for quadratic contraction. Along the constructed gap trajectory, the scaled gradient turns toward the negative Y -direction along a circular arc centered at (β, β) . The construction gives a gap-mass coefficient c_β and an additive parameter $a_\beta \geq 8$ for the denominator in the transfer bound (13) below, satisfying

$$1 < c_\beta < 2, \quad c_\beta = 1 + O(\beta), \quad \log a_\beta = O(1/\beta), \quad (12)$$

uniformly when $0 < \beta \leq 1/4$.

Lemma 3.1 (Bending). *Let $\kappa \geq 4$, $0 < \beta \leq 1/4$, $s_i/\kappa \leq 1/16$, and $\eta_i > 0$. There exist a nonnegative convex 1-smooth local hard function Φ_i and a threshold $\ell_{i+1} \geq 0$, independent of the checkpoint stepsize b_i , such that $\partial_2 \Phi_i \leq 0$ everywhere and*

$$\Phi_i(X, Y) = 0 \quad (X \leq \ell_i, Y \geq 0), \quad \partial_2 \Phi_i(X, Y) = 0 \quad (X \in \mathbb{R}, Y \geq \ell_{i+1}).$$

For every $b_i > 8$, the iterations (9) satisfy (8), and

$$\frac{D_{i+1}}{D_i} \geq \frac{b_i - 8}{c_\beta(a_\beta + s_i + 8)} \eta_i(1 - s_i/\kappa), \quad \frac{\ell_{i+1}}{D_{i+1}} \leq 4 \frac{s_i + 2}{b_i - 8}. \quad (13)$$

For $Y \geq \ell_{i+1}$, the derivative $\partial_2 \Phi_i(X, Y)$ vanishes for every X . Thus, Φ_i does not affect the update of $Y = x^{(i+1)}$ in this region. For $i < k$, this is the input coordinate of Φ_{i+1} . The value $\Phi_i(X, Y)$ and derivative $\partial_1 \Phi_i(X, Y)$ may still be nonzero. Taking β small makes the gap coefficient c_β close to one, at the cost of the additive constant a_β . The construction and proof are given in [Section A.1](#).

3.2 Repairing

Using the bending component alone for checkpoint selection can leave several consecutive blocks without a checkpoint. We split such long gaps with additional checkpoints and use Huber repair components to control the threshold-to-amplitude ratio ℓ_{i+1}/D_{i+1} . The construction uses the following one-sided Huber function of Jung et al. [2026]:

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

We apply H_δ to the margin $X - Y - \ell_i$ and choose its scale so that every query through the checkpoint lies in the affine region, where the gradient of Φ_i is constant. The extra term $h_j f_j \geq 0$ in the X update helps keep the margin in this region.

Lemma 3.2 (Repairing). *Let $\kappa > 1$ and $\eta_i > 0$. Define*

$$\delta_H = \frac{D_i \eta_i}{2 + (\kappa - 1)(1 - \chi_i)}, \quad \Phi_i(X, Y) = \frac{1}{2} H_{2\delta_H}(X - Y - \ell_i). \quad (14)$$

Set $\ell_{i+1} = (\kappa - 1)(1 - \chi_i)\delta_H/2$. Φ_i is nonnegative, convex, and 1-smooth, with $\partial_2 \Phi_i \leq 0$ everywhere and $\Phi_i(X, Y) = 0$ whenever $X \leq \ell_i$, $Y \geq 0$.

For every $b_i > 0$ and every sequence $f_j \geq 0$, the iterations (10) satisfy (8), and

$$\frac{D_{i+1}}{D_i} = \frac{(1 - 1/\kappa)b_i \chi_i \eta_i}{2[2 + (\kappa - 1)(1 - \chi_i)]} \geq \frac{(1 - 1/\kappa)b_i \chi_i \eta_i}{2(s_i + 2)}, \quad \frac{\ell_{i+1}}{D_{i+1}} = \frac{\kappa(1 - \chi_i)}{b_i \chi_i}. \quad (15)$$

Consequently, if $s_i < \kappa$, then

$$\frac{s_i}{b_i} \leq \frac{\ell_{i+1}}{D_{i+1}} \leq \frac{s_i/b_i}{1 - s_i/\kappa}. \quad (16)$$

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. When s_i is small relative to κ , (16) shows that a checkpoint with b_i comparable to s_i gives a bounded threshold-to-amplitude ratio 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 $\partial_2 \Phi_i(X, Y) = 0$ only when $Y \geq X - \ell_i$. Thus, even after Y crosses ℓ_{i+1} , Φ_i may still affect the update of Y , the input of Φ_{i+1} . The construction in the next section resolves this issue.

3.3 Bridging

After a Huber repair, the preceding component Φ_{i-1} may still add $h_j f_j \geq 0$ to the update of X , whereas Lemma 3.1 requires $f_j = 0$. We therefore use a bridge Φ_i between repairing and bending. It allows $f_j \geq 0$ during its own transfer and satisfies $\partial_2 \Phi_i(X, Y) = 0$ for $Y \geq \ell_{i+1}$. Thus, it does not affect the update of $Y = x^{(i+1)}$ during the next bending gap and at its checkpoint query.

Lemma 3.3 (Bridging). *Let $\kappa \geq 4$, $s_i/\kappa \leq 1/16$, and $\eta_i > 0$. With δ_H from (14), define*

$$\ell_{i+1} = \frac{D_i \eta_i}{2}, \quad K_\Delta = \text{conv}\{0, (\delta_H, -\delta_H), (\delta_H, 0)\},$$

and

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

Φ_i is nonnegative, convex, and 1-smooth, with $\partial_2 \Phi_i \leq 0$ everywhere and $\Phi_i(X, Y) = 0$ whenever $X \leq \ell_i$, $Y \geq 0$. Moreover, $\partial_2 \Phi_i(X, Y) = 0$ for all $X \in \mathbb{R}$, $Y \geq \ell_{i+1}$.

For every $b_i > 4$ and every sequence $f_j \geq 0$, the iterations (10) satisfy (8), and

$$\frac{D_{i+1}}{D_i} \geq \frac{1}{4} \eta_i \frac{b_i - 4}{s_i + 2}, \quad \frac{\ell_{i+1}}{D_{i+1}} \leq 2 \frac{s_i + 2}{b_i - 4}.$$

With the same incoming data, stepsizes, and sequence $f_j \geq 0$, the bridge and the Huber repair component of Section 3.2 have identical gradients through the checkpoint query and hence the same value of Y after the checkpoint update. Unlike the repair component, whose derivative with respect to Y vanishes only when $Y \geq X - \ell_i$, the bridge has $\partial_2 \Phi_i(X, Y) = 0$ for every X whenever $Y \geq \ell_{i+1}$. In this region, it contributes nothing to the update of Y , the input coordinate of Φ_{i+1} . The proof is provided in Section A.3. Figure 2 provides an illustration of the trajectories of these local constructions.

4 Selecting Checkpoints for Every Schedule

Fix a positive integer m . Divide the schedule $H = (h_1, \dots, h_n)$ into blocks of length m , where the last block may contain fewer steps. Note that a block's length counts steps, while its mass is the sum of their stepsizes. The block boundaries are fixed before selecting checkpoints. In contrast, a gap is determined by successive checkpoints and may cross several block boundaries.

We use a chain of bending components as the reference for checkpoint selection. In each block, we maximize the score in (19), formed from the bending transfer bound of Lemma 3.1. The score omits the quadratic decay factors and the dependence on the incoming threshold-to-amplitude ratios; Section 5.1 verifies that the selected transfers remain feasible and retain a constant fraction of this score. Under this score-maximizing selection, Lemma 4.1 gives bounds on the unselected mass and on the relation between checkpoint stepsizes and adjacent gap masses. These bounds will be used to control quadratic decay and the threshold-to-amplitude ratios in Section 5.

After examining a block, the steps after the latest checkpoint through the end of that block form the *unselected tail*; if no checkpoint has been selected, this includes all steps examined so far. If the score maximizer selects no checkpoint, the current block extends this tail. Once its accumulated mass becomes too large, we perform a repair by selecting one or two additional checkpoints in this tail and using a Huber repair component from Section 3.2 for each selected checkpoint. This is described in Section 4.2.

The two selections serve different purposes. Maximizing the bending score preserves the favorable transfer ratios responsible for the exponent in (17). A Huber repair restores a bounded threshold-to-amplitude ratio and leaves little unselected mass, at a controlled loss of amplitude.

4.1 Choosing Checkpoints within a Block

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

$$p = p(c) := \log_2(1 + \sqrt{1 + c}), \quad v = 1/p. \quad (17)$$

Consider one fixed block of $r \leq m$ steps, with stepsizes relabelled as $H = (h_1, \dots, h_r)$. Let $w \geq 0$ be the mass of the unselected tail from preceding blocks; initially $w = 0$.

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

$$s_1 := w + \sum_{j=1}^{t_1-1} h_j, \quad s_i := \sum_{j=t_{i-1}+1}^{t_i-1} h_j \quad (2 \leq i \leq k), \quad s_{\text{tail}} := \sum_{j=t_k+1}^r h_j. \quad (18)$$

Note that the first gap starts immediately after the preceding checkpoint, even if that checkpoint is in an earlier block. Its mass s_1 therefore includes both w and the stepsizes before the first checkpoint selected in the current block. Define the checkpoint score

$$P_w(T; H) := \prod_{i=1}^k \frac{B_i}{d_i}, \quad B_i := b_i - 8, \quad d_i := c(a + s_i + 8), \quad P_w(\emptyset; H) := 1. \quad (19)$$

For the fixed block H and fixed mass w , we maximize this score over the admissible checkpoint sets T . If a nonempty set attains the maximum, we choose one. Otherwise we choose $T = \emptyset$, 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 gives the bounds needed to realize the maximizing checkpoint set using the local hard functions. Parts (i)–(ii) control the unselected mass and the sum of the score denominators $\sum_i d_i$, hence the gap mass that causes quadratic decay. Parts (iii)–(iv) bound the terms used to control contraction loss in the next gap; see (26).

Lemma 4.1 (Bounds for a Maximizing Checkpoint Set). *There is an absolute constant C_b with the following properties, uniformly for $1 \leq c \leq 2$, $a \geq 8$, $1 \leq r \leq m$, and $w \geq 0$. Parts (ii)–(iv) concern a nonempty maximizing checkpoint set $T = \{t_1 < \dots < t_k\}$.*

(i) *If the empty set maximizes (19), then*

$$w + \sum_{j=1}^r h_j \leq C_b(w + am^p).$$

(ii) *If T is a nonempty maximizer, then*

$$\sum_{i=1}^k d_i \leq C_b(w + am^p), \quad s_{\text{tail}} \leq C_b am^p.$$

(iii) *For two consecutive checkpoints, set $x = a + s_i + 8$, $y = a + s_{i+1} + 8$, and $B = B_i$. Then*

$$B(x + y + B - a) \geq cxy, \quad \frac{xy}{B} \leq 2(x + y).$$

In particular,

$$\frac{d_i d_{i+1}}{B_i} \leq 4(d_i + d_{i+1}), \quad \frac{s_{i+1} d_i}{B_i} \leq 2(d_i + d_{i+1}).$$

(iv) *The last checkpoint satisfies $B_k \geq d_k$.*

The full proof is provided in [Appendix B](#).

4.2 Repairing Accumulated Unselected Steps

The checkpoint set maximizing the score may be empty. We then retain the block as part of the unselected tail. Part (i) of [Lemma 4.1](#) bounds the total mass of the enlarged tail when this happens for one block, but several consecutive blocks with no selected checkpoint can still accumulate too much mass. The next lemma selects $q \in \{1, 2\}$ checkpoints in this tail to bound both the threshold-to-amplitude ratio ℓ_{q+1}/D_{q+1} after the last checkpoint and the remaining unselected mass.

Lemma 4.2 (Selecting checkpoints for repair). *Fix $C_0 \geq 1$ and $R = 32C_0$. There are constants $\varepsilon_0 \in (0, 1)$ and $c_0 > 0$, depending only on C_0 , with the following property. Let $\kappa \geq 4$, let $r \geq 1$ be an integer, and let $h_1, \dots, h_r \geq 0$ satisfy*

$$S := \sum_{t=1}^r h_t, \quad S_0 \leq S \leq C_0 S_0, \quad 1 \leq S_0 \leq \frac{\varepsilon_0 \kappa}{C_0}.$$

There exist $q \in \{1, 2\}$ and checkpoint indices $0 = t_0 < t_1 < \dots < t_q \leq r$, with $b_i := h_{t_i} > 0$, having the following properties. For $1 \leq i \leq q$, define the gap sum and contraction by

$$s_i := \sum_{t=t_{i-1}+1}^{t_i-1} h_t, \quad \chi_i := \prod_{t=t_{i-1}+1}^{t_i-1} (1 - h_t/\kappa). \quad (20)$$

Let $\ell \geq 0$, $D > 0$, and $\ell/D \leq R$. Starting from $(\ell_1, D_1) = (\ell, D)$, define successively

$$\begin{aligned} \eta_i &:= \chi_i - (1 - \chi_i) \frac{\ell_i}{D_i}, & \delta_{H,i} &:= \frac{D_i \eta_i}{2 + (\kappa - 1)(1 - \chi_i)}, \\ \ell_{i+1} &:= \frac{\kappa - 1}{2} \delta_{H,i} (1 - \chi_i), & D_{i+1} &:= \frac{1 - 1/\kappa}{2} b_i \delta_{H,i} \chi_i. \end{aligned} \quad (21)$$

Then:

(i) *For every $1 \leq i \leq q$, $\eta_i > 0$ and $D_{i+1} > 0$, and*

$$\prod_{i=1}^q \frac{D_{i+1}}{D_i} = \frac{D_{q+1}}{D_1} \geq c_0 (r+1)^{-4}. \quad (22)$$

(ii) $\ell_{q+1}/D_{q+1} \leq R$.

(iii) $\sum_{t=t_q+1}^r h_t \leq S_0/4$.

(iv) *Define*

$$j_* := \max \left\{ j \in \{1, \dots, r\} : \sum_{t=j}^r h_t > S_0/8 \right\}, \quad (23)$$

then $j_ \leq t_1 < \dots < t_q \leq r$.*

Here, (i) bounds the product of the amplitude ratios, (ii) bounds the threshold-to-amplitude ratio immediately after the last selected checkpoint, and (iii) bounds the remaining tail mass. Assertion (iv) locates the selected indices in $\{j_*, \dots, r\}$. In the application below, $(h_1, \dots, h_r)$ is the entire unselected tail from immediately after the latest chosen checkpoint through the end of the current block, including steps from preceding blocks; hence r may exceed m . Repair checkpoints may therefore lie in the incoming tail, while previously chosen checkpoints remain fixed.

The quantities (21) in the lemma are from the Huber repair components in [Lemma 3.2](#). For a Huber repair component Φ_i , the threshold ℓ_{i+1} is the value of Y just before the checkpoint update, and D_{i+1} is the increase in Y produced by that update. By (16), their ratio is comparable to s_i/b_i when s_i is small relative to κ . Therefore, a step comparable to its preceding gap mass restores a bounded ratio. However, such a step need not exist, so [Lemma 4.2](#) also allows two checkpoints. The checkpoint choice and the proof of the transfer bound are given in [Appendix C](#).

5 Assembling the Hard Function and Proving the Theorem

After selecting the checkpoints as in Section 4, we construct the objective (6) from the local hard functions of Section 3. Each checkpoint added by a repair uses a Huber repair component. Within each nonempty maximizing checkpoint set, the first checkpoint uses a triangular Huber bridge and the remaining checkpoints use bending components.

With the hard function constructed, Lemma 5.1 bounds the product of transfer ratios from below by an absolute constant for each nonempty maximizing set, and Lemma 5.2 controls the accumulated penalty from repairs. Together, they yield the bound in Proposition 5.3 for a fixed bending parameter and every horizon. Lemma 5.4 removes the prefactor and gives the corresponding bound for relative function error. Finally, Section 5.3 chooses the bending parameter to obtain the (almost) silver dependence on κ .

5.1 Realizing a Maximizing Checkpoint Set

Fix $0 < \beta \leq 1/4$ and set $a = a_\beta$, $c = c_\beta$, $p = p(c)$, and $v = 1/p$. We keep β fixed through the construction and repetition argument, and choose its value in Section 5.3.

Fix $\kappa \geq 4$, a block $H = (h_1, \dots, h_r)$, and the unselected tail from preceding blocks, whose mass is w . Let $T = \{t_1 < \dots < t_k\}$, with $k \geq 1$, be an admissible maximizer of $P_w(T; H)$. Throughout this subsection, $i = 1, \dots, k$ numbers the selected checkpoints and their components within this block; t_i is the corresponding index in H . Take $b_i = h_{t_i} > 8$ and the gap masses s_i from (18), and put

$$B_i = b_i - 8, \quad d_i = c(a + s_i + 8) \quad (1 \leq i \leq k).$$

The first gap includes the unselected steps after the preceding checkpoint, including those before this block. For each $1 \leq i \leq k$, let χ_i be the product of $1 - h/\kappa$ over all stepsizes h in gap i .

The local trajectory for Φ_i starts at $(\ell_i + D_i, 0)$ and reaches $Y = \ell_{i+1} + D_{i+1}$ after the update at checkpoint t_i . The parameters (ℓ_1, D_1) are given at the beginning of the first gap; the subsequent pairs are constructed in checkpoint order. Write

$$r_i = \frac{\ell_i}{D_i} \quad (1 \leq i \leq k+1),$$

$$G_i = \frac{D_{i+1}}{D_i}, \quad \eta_i = \chi_i - (1 - \chi_i)r_i \quad (1 \leq i \leq k).$$

We use a bridge for $i = 1$ and bending components for $2 \leq i \leq k$. Whenever $s_i/\kappa \leq 1/16$ and $\eta_i > 0$, the local lemmas give

$$G_1 \geq \frac{1}{4} \frac{B_1}{d_1} \eta_1 \quad (\text{bridge at } t_1), \quad (24)$$

$$G_i \geq \frac{B_i}{d_i} \eta_i (1 - s_i/\kappa) \quad (2 \leq i \leq k), \quad (25)$$

$$r_{i+1} \leq \frac{4(s_i + 2)}{B_i}, \quad \eta_i \geq 1 - \frac{s_i}{\kappa} (1 + r_i) \quad (1 \leq i \leq k). \quad (26)$$

Here the bridge bounds follow from Lemma 3.3 using $b_1 - 4 \geq B_1$ and $d_1 \geq s_1 + 2$; the remaining bounds follow from Lemma 3.1 and (11).

The maximizing score is at least $P_w(\emptyset; H) = 1$. Under the condition in (27) below, the next lemma realizes the selected transfers with combined ratio at least $P_w(T; H)/8 \geq 1/8$, and leaves a threshold-to-amplitude ratio at most 4. Thus the checkpoint selection retains a constant fraction of the incoming amplitude while allowing the construction to continue.

Lemma 5.1 (Realizing the maximizing checkpoint set). *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*

$$\frac{1}{\kappa} \sum_{i=1}^k d_i \leq \frac{1}{128(R+1)}. \quad (27)$$

There exist a bridge Φ_1 and bending components $\Phi_2, \dots, \Phi_k$, with thresholds $\ell_2, \dots, \ell_{k+1} \geq 0$ and amplitudes $D_2, \dots, D_{k+1} > 0$, such that the local trajectory for each component satisfies (8) on its prescribed gap and checkpoint. The iterations for Φ_1 follow (10) with arbitrary nonnegative input contributions; those for $\Phi_2, \dots, \Phi_k$ follow (9). Moreover,

$$\frac{D_{k+1}}{D_1} = \prod_{i=1}^k G_i \geq \frac{1}{8} P_w(T; H), \quad \frac{\ell_{k+1}}{D_{k+1}} \leq 4.$$

Proof. We first construct the components in checkpoint order. Since $s_i \leq d_i$, (27) gives $s_i/\kappa \leq 1/128 < 1/16$ for every $1 \leq i \leq k$. Thus every stepsize within a gap is less than κ . For the first gap,

$$1 - \eta_1 \leq \frac{s_1}{\kappa} (1 + r_1) \leq \frac{(R+1)d_1}{\kappa} \leq \frac{1}{128}.$$

Hence $\eta_1 > 0$, and Lemma 3.3 constructs Φ_1 and its outgoing parameters, satisfying (24) and (26) for $i = 1$.

Now fix $2 \leq i \leq k$ and suppose that $\Phi_1, \dots, \Phi_{i-1}$ have been constructed. Their last ratio bound gives $r_i \leq 4(s_{i-1} + 2)/B_{i-1}$. Since $s_{i-1} + 2 \leq d_{i-1}$, the deletion comparison in Lemma 4.1 (iii) yields

$$\begin{aligned} 1 - \eta_i &\leq \frac{s_i}{\kappa} (1 + r_i) \leq \frac{d_i}{\kappa} + \frac{4d_{i-1}d_i}{\kappa B_{i-1}} \\ &\leq \frac{17d_i + 16d_{i-1}}{\kappa} \leq \frac{17}{128(R+1)} < 1. \end{aligned}$$

Thus $\eta_i > 0$, so Lemma 3.1 constructs Φ_i and its outgoing parameters, satisfying (25) and (26) at index i . At every gap query and at the checkpoint query, the new trajectory has $x^{(i)} \geq \ell_i$. The preceding bridge or bending component therefore satisfies $\partial_2 \Phi_{i-1} = 0$ and does not affect the update of $x^{(i)}$ in (9). The bounds in (8) also ensure that every later component has value zero at these query points, and hence zero gradient. This completes the induction.

We now bound the total amplitude decrease. Summing the preceding estimates gives

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

All factors η_i and $1 - s_i/\kappa$ lie in $(0, 1]$. Using $\prod_j (1 - u_j) \geq 1 - \sum_j u_j$ for $0 \leq u_j \leq 1$, and multiplying (24)–(25), we obtain

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

Finally, Lemma 4.1 (iv) gives $B_k \geq d_k$. Applying the ratio bound in (26) at $i = k$ yields

$$r_{k+1} \leq \frac{4(s_k + 2)}{B_k} \leq \frac{4d_k}{B_k} \leq 4.$$

□

5.2 Continuing the Construction for Every Horizon

A repair can involve an unselected tail spanning several blocks. We bound the total number of indices considered by all repairs, which will control their total amplitude loss.

Fix integers $n, m \geq 1$, 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 $J \leq B$. Repair j considers the 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 $u_{j+1} \geq p_j + 1$ for $1 \leq j < J$. These indices are illustrated in Figure 3.

Each repair starts with unselected mass at least S_0 and leaves mass at most $S_0/4$ after its last checkpoint:

$$\sum_{t=u_j}^{v_j} h_t \geq S_0, \quad \sum_{t=p_j+1}^{v_j} h_t \leq S_0/4 \quad (1 \leq j \leq J). \quad (28)$$

Lemma 4.2 guarantees that repair j retains at least a fraction $c_0(n_j + 1)^{-4}$ of its incoming amplitude, so we must control $\sum_{j=1}^J \log(n_j + 1)$. The next lemma bounds this sum in terms of the block count B and block length m , even though an individual repair may consider more than m indices. This is what controls the total amplitude loss for arbitrarily long schedules.

Lemma 5.2 (Counting indices used by repairs). *Under the conditions above, each index is considered by at most two repairs. Consequently,*

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

Proof. We first show $p_{j+1} > v_j$ for $1 \leq j < J$. Otherwise, $u_{j+1} \geq p_j + 1$ and (28) give

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

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

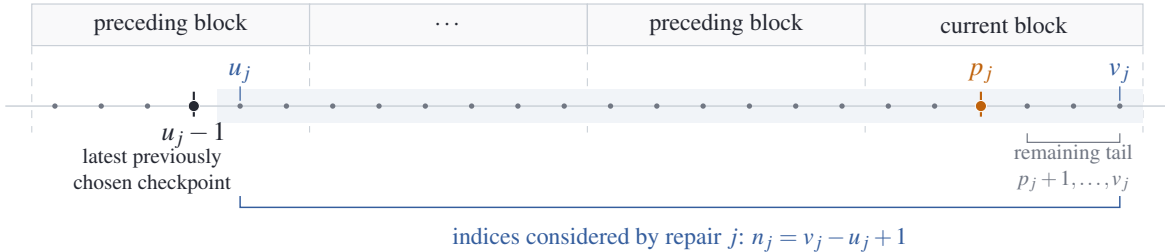


Figure 3: Indices for repair j . The black checkpoint is $u_j - 1$; the repair considers all indices from u_j through v_j , the last index of the current block. Orange marks p_j , the last checkpoint chosen by this repair, leaving indices $p_j + 1, \dots, v_j$ unselected. The location of p_j in the current block is illustrative: it may also lie in a preceding block.

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

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

□

We now combine the transfer bound for nonempty maximizing sets with the aggregate repair bound, assigning the local hard functions as described at the start of this section. For GD on the resulting objective (6), a lower bound on the last coordinate gives the following lower bound on relative squared distance for every horizon.

Proposition 5.3 (Uniform bound for a fixed bending parameter). *There are absolute constants $c_*, C, \varepsilon > 0$ with the following property. For every $0 < \beta \leq 1/4$ and $\kappa \geq 4$, set*

$$S_0 := \varepsilon \kappa, \quad m := \left\lfloor c_*(\kappa/a_\beta)^{v(c_\beta)} \right\rfloor.$$

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

$$D_{n,\kappa}(H) \geq \exp \left[-C \left(1 + \frac{n}{m} \right) \log(m+1) \right]. \quad (29)$$

Proof. Let $C_b \geq 1$ be an absolute constant covering the budget bounds in Lemma 4.1, and set

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

Choose ε small enough for Lemmas 5.1 and 4.2, and then choose c_* so small that

$$C_b a m^p \leq S_0/4. \quad (30)$$

These choices are uniform: $p(c)$ stays within a fixed compact subset of $(1, 2)$ and $a m^p \leq c_*^p \kappa$.

Partition the schedule into $B := \lceil n/m \rceil$ fixed blocks, each containing at most m steps. After examining a block, retain all unselected steps after the most recent checkpoint up to the end of that block. Write w for their total mass, and let $\ell \geq 0$ and $D > 0$ be the threshold and amplitude immediately after the most recent checkpoint. We maintain

$$w < S_0, \quad \frac{\ell}{D} \leq R. \quad (31)$$

Initially, $w = \ell = 0$ and $D = 1$. The unselected tail following the checkpoint is included in the next local transfer.

For the next fixed block, maximize (19) over its checkpoint sets, with w equal to the mass of the unselected tail from preceding blocks. Choose a nonempty maximizing set whenever one exists.

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

$$\sum_i d_i \leq C_b(w + a m^p) \leq (C_b + 1)S_0 \leq C_0 S_0.$$

The choice of ε ensures (27). Lemma 5.1 realizes the transfers with a product at least $1/8$, since the maximizing score is at least the empty score 1. The outgoing ratio is at most 4 and, by (30), the remaining tail mass is at most $S_0/4$. Thus (31) 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 total mass of this enlarged tail is at most

$$C_b(w + am^p) \leq C_0 S_0.$$

If this mass is less than S_0 , retain this tail without selecting a new checkpoint. Otherwise perform the next repair, numbered j . Let u_j be one plus the latest checkpoint index, taking $u_j = 1$ if there is no checkpoint yet, and let v_j be the end of the current block. Put $n_j = v_j - u_j + 1$. Then

$$S_0 \leq \sum_{t=u_j}^{v_j} h_t \leq C_0 S_0, \quad 1 \leq S_0 \leq \varepsilon_0 \kappa / C_0,$$

where the last inequality follows from the choice of ε . The local input immediately after the latest checkpoint, or at initialization, is $(\ell + D, 0)$, with $\ell/D \leq R$ by (31).

Apply Lemma 4.2 with $r = n_j$, initial data $(\ell_1, D_1) = (\ell, D)$, and the relabelled steps $(h_{u_j}, \dots, h_{v_j})$. It supplies $q_j \in \{1, 2\}$ and local checkpoint indices $t_1 < \dots < t_{q_j}$. Their indices in the full schedule are

$$u_j + t_1 - 1, \dots, u_j + t_{q_j} - 1, \quad p_j := u_j + t_{q_j} - 1.$$

Select these indices as checkpoints and use the Huber repair components with the parameters in (21). Part (i) of Lemma 4.2 gives

$$\frac{D_{q_j+1}}{D_1} \geq c_0(n_j + 1)^{-4}.$$

In the full schedule, parts (ii) and (iii) give

$$\frac{\ell_{q_j+1}}{D_{q_j+1}} \leq R, \quad \sum_{t=p_j+1}^{v_j} h_t \leq S_0/4.$$

Thus the new threshold and amplitude are ℓ_{q_j+1} and D_{q_j+1} , and the retained tail consists of the steps at indices $p_j + 1, \dots, v_j$. These bounds restore (31).

One globally defined objective. Write all selected checkpoint indices in the full schedule as $t_1 < \dots < t_k$, and set $t_0 = 0$. Starting from $(\ell_1, D_1) = (0, 1)$, we construct $\Phi_1, \dots, \Phi_k$ in checkpoint order using the local constructions in Section 3. For each i , use the incoming pair (ℓ_i, D_i) , the gap stepsizes $h_{t_{i-1}+1}, \dots, h_{t_i-1}$, and the checkpoint stepsize $b_i = h_{t_i}$ to construct Φ_i and determine the outgoing pair (ℓ_{i+1}, D_{i+1}) . Use a Huber repair component (Lemma 3.2) if t_i was selected by a repair, a bridge (Lemma 3.3) if it is the first checkpoint of a nonempty maximizing set, and a bending component (Lemma 3.1) otherwise.

For $i < k$, use the resulting pair (ℓ_{i+1}, D_{i+1}) as the input for the next construction. Cases 1 and 2 verify the hypotheses needed at every step. For Huber repair and bridge components, D_{i+1} is independent of the nonnegative input contributions f_j , by (15) and (41). Thus the recursion fixes all components from the prescribed schedule before GD is run.

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

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

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

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

Indeed, all components after Φ_i have value zero at these query points and hence zero gradient. If Φ_i is a bending component, then Φ_{i-1} is a bridge or a bending component. Since $x_t^{(i)} \geq \ell_i$, $\partial_2 \Phi_{i-1}(x_t^{(i-1)}, x_t^{(i)}) = 0$, so the update follows (9). If Φ_i is a Huber repair component or a bridge, the preceding component's nonpositive output derivative gives the permitted nonnegative input contribution in (10); for $i = 1$ this contribution is zero. Components with index less than $i - 1$ involve neither coordinate. The local lemmas therefore maintain the displayed bounds through the checkpoint query and give

$$x_{t_i}^{(i+1)} = \ell_{i+1} + D_{i+1}, \quad x_{t_i}^{(j)} = 0 \quad (i+2 \leq j \leq k+1).$$

This proves the next induction hypothesis and realizes all the prescribed transfers along GD on the single objective F .

The aggregate repair cost. The procedure makes at most one repair per block, and the two mass bounds in (28) were verified above. Each repair j retains at least $c_0(n_j + 1)^{-4}$ of its incoming amplitude. By Lemma 5.2 and $J \leq B$, their combined factor is at least

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

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 final amplitude is at least $\exp[-CB \log(m+1)]$.

The final coordinate. If $k \geq 1$, then $x_{t_k}^{(k+1)} = \ell_{k+1} + D_{k+1} \geq D_{k+1}$. Since $\partial_2 \Phi_k \leq 0$, every remaining update $t_k < t \leq n$ satisfies

$$x_t^{(k+1)} \geq \left(1 - \frac{h_t}{\kappa}\right) x_{t-1}^{(k+1)}.$$

The remaining tail has mass less than $S_0 = \varepsilon \kappa$, so

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

Since $\|x_0 - x_*\|^2 = 1$ and $\|x_n - x_*\|^2 \geq (x_n^{(k+1)})^2$, increasing C and using $B \leq 1 + n/m$ proves (29). If $k = 0$, the entire schedule has mass less than S_0 , and the pure quadratic $F(x) = x^2/(2\kappa)$ gives the same conclusion. $\square$

The factor $\exp[-C \log(m+1)]$ in (29) is independent of the horizon. Applying this bound to repetitions of the same schedule removes this factor and also yields a lower bound for relative function error.

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

$$\min\{D_{n,\kappa}(H), E_{n,\kappa}(H)\} \geq \exp\left[-\frac{Cn}{m} \log(m+1)\right]. \quad (32)$$

Proof. Fix n and H . For each integer $K \geq 1$, let $H^{[K]}$ denote K consecutive copies of H , a schedule of length Kn .

The definition of $D_{n,\kappa}(H)$ bounds the squared-distance ratio from every initial point. Applying this bound at the beginning of each copy, on the same objective, and using Proposition 5.3 at horizon Kn gives

$$\exp\left[-C \left(1 + \frac{Kn}{m}\right) \log(m+1)\right] \leq D_{Kn,\kappa}(H^{[K]}) \leq (D_{n,\kappa}(H))^K.$$

Taking K th roots and letting $K \rightarrow \infty$ yields

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

The same argument bounds the function-error ratio of K copies by $(E_{n,\kappa}(H))^K$. Strong convexity and smoothness also give $D_{Kn,\kappa}(H^{[K]}) \leq \kappa E_{Kn,\kappa}(H^{[K]})$. Therefore,

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

Taking K th roots and letting $K \rightarrow \infty$ gives the same lower bound for $E_{n,\kappa}(H)$, since $\kappa^{1/K} \rightarrow 1$. $\square$

The hard function and its dimension may depend on K , as permitted by the supremum in (1); the schedule H and parameters β, m remain fixed.

5.3 Choosing the Bending Parameter

The explicit constants in [Lemma 3.1](#) satisfy

$$c_\beta = 1 + O(\beta), \quad \log a_\beta = O(1/\beta), \quad v(c_\beta) = 1/p_{\text{sil}} - O(\beta).$$

The first two estimates follow from the explicit formulas in [Section A.1](#); the last follows by differentiating (17) near $c = 1$. For sufficiently large κ , choose

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

Then $\kappa/a_\beta \rightarrow \infty$, so the hypotheses of [Proposition 5.3](#) hold. Its block length satisfies

$$\begin{aligned} \log m &\geq v(c_\beta)(\log \kappa - \log a_\beta) - O(1) \\ &\geq \frac{\log \kappa}{p_{\text{sil}}} - C(\beta \log \kappa + 1/\beta) \geq \frac{\log \kappa}{p_{\text{sil}}} - C'\sqrt{\log \kappa}. \end{aligned} \tag{33}$$

The parameter is chosen from κ alone and remains fixed during the repetition argument.

Proof of Theorem 1.1 and Corollary 1.2. For the chosen β , [Lemma 5.4](#) gives (32). By (33) and $\log(m+1) \leq C \log \kappa$, this yields (2). If either worst-case error is at most δ , taking logarithms gives (3). $\square$

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 . Two 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 comes from two parts of the proof: the repair estimates and the choice of the bending parameter. If a repair considers r indices, [Lemma 4.2](#) guarantees that the amplitude after the repair is at least $c_0(r+1)^{-4}$ times the amplitude before the repair. Combining these bounds over all repairs produces the factor $\log(m+1)$ in (32). For bending, taking a smaller β improves the exponent of κ but increases the constant a_β , which limits the block length. Balancing these effects as in (33) gives the factor $e^{C\sqrt{\log \kappa}}$ in (3).

2. *Can we extend our lower bound to predetermined schedules that may contain negative stepsizes?* Our previous work [Ye and Liu, 2026a] 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.

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

- 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.
- Jason Altschuler. Greed, hedging, and acceleration in convex optimization. Master’s thesis, Massachusetts Institute of Technology, 2018.
- 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.
- 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, 2025a.
- 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, 2025b. Article 12.
- 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.
- Yossi Arjevani and Ohad Shamir. On the iteration complexity of oblivious first-order optimization algorithms. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 908–916. PMLR, 2016.
- Yossi Arjevani, Shai Shalev-Shwartz, and Ohad Shamir. On lower and upper bounds in smooth and strongly convex optimization. *Journal of Machine Learning Research*, 17(126):1–51, 2016.
- 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.
- Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.
- 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.

- Yoel Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 39:1–16, 2017.
- Yoel Drori and Adrien Taylor. On the oracle complexity of smooth strongly convex minimization. *Journal of Complexity*, 68:101590, 2022.
- 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.
- Benjamin Grimmer. Provably faster gradient descent via long steps. *SIAM Journal on Optimization*, 34(3):2588–2608, 2024.
- Benjamin Grimmer, Kevin Shu, and Alex L. Wang. Composing optimized stepsize schedules for gradient descent. *Mathematics of Operations Research*, 2025a. Articles in Advance.
- 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, 2025b.
- 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.
- Minchan Jung, Hanseul Cho, and Chulhee Yun. Stronger lower bounds for (non-)anytime acceleration of gradient descent, 2026. arXiv preprint arXiv:2609.04032v2.
- Zdeněk Kalousek. Steepest descent method with random step lengths. *Foundations of Computational Mathematics*, 17(2):359–422, 2017.
- 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.
- Yu Liu, Kang Chen, Rujun Jiang, and Tianyu Wang. Optimal recursive composition and dyadic phase laws for gradient descent with predetermined stepsizes, 2026. arXiv preprint arXiv:2609.11788.
- Jianhao Ma and Yuxin Chen. A lower bound for stepsize-based acceleration of gradient descent. *arXiv preprint arXiv:2608.10418*, 2026.
- Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93(3):273–299, 1965.
- Arkadii S. Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Chichester, 1983. ISBN 0471103454. Translated by E. R. Dawson.
- Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.
- 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.
- Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.
- Boris T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.
- Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

- Boris T. Polyak. *Introduction to Optimization*. Optimization Software, Inc., Publications Division, New York, 1987. ISBN 0911575146.
- Luc Pronzato and Anatoly Zhigljavsky. Gradient algorithms for quadratic optimization with fast convergence rates. *Computational Optimization and Applications*, 50(3):597–617, 2011.
- R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, Heidelberg, 1998.
- Adrien Taylor and Yoel Drori. An optimal gradient method for smooth strongly convex minimization. *Mathematical Programming*, 199(1–2):557–594, 2023.
- 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.
- Chung-En Tsai. An improved lower bound for non-anytime gradient descent, 2026. URL <https://chungentsai.github.io/gd-lower-bounds.html>. Blog post.
- 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.
- Yuhan Ye and Kaizhao Liu. Improved gradient descent lower bounds beyond Nesterov. *arXiv preprint arXiv:2609.02855*, 2026a.
- Yuhan Ye and Kaizhao Liu. Silver rate is (almost) optimal for gradient descent, 2026b. URL <https://arxiv.org/abs/2609.09152v2>. arXiv preprint arXiv:2609.09152v2.
- 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.
- Zehao Zhang and Rujun Jiang. Accelerated gradient descent by concatenation of stepsize schedules. *SIAM Journal on Optimization*, 36(2):1182–1210, 2026.
- 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.
- 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 D_{i+1}/D_i from below to retain amplitude and ℓ_{i+1}/D_{i+1} from above to limit quadratic contraction loss in the next gap.

We first recall a property of Moreau envelopes of support functions used in all three constructions.

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*

$$\begin{aligned} (\text{env}_1 \sigma_K)(x) &= \min_{y \in \mathbb{R}^2} \left\{ \sigma_K(y) + \frac{1}{2} \|x - y\|^2 \right\} \\ &= \max_{g \in K} \left\{ \langle g, x \rangle - \frac{1}{2} \|g\|^2 \right\}. \end{aligned} \tag{34}$$

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

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

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

$$\langle x - g, w - g \rangle \leq 0 \quad \text{for every } w \in K. \tag{35}$$

The envelope identities follow from [Moreau \[1965, Propositions 5.b, 7.b, and 7.d\]](#). For background, see [Rockafellar and Wets \[1998, Sections 1.G and 2.D\]](#) and [Parikh and Boyd \[2014, Sections 3.1 and 6.4.2\]](#).

A.1 Proof of [Lemma 3.1](#)

Proof idea. We adapt the construction of [Ye and Liu \[2026b, Lemmas 3.1 and 3.2\]](#) to include quadratic contraction. First, [Lemma A.2](#) constructs the gap trajectory and its gradient vectors. We then match the initial point $(\ell_i + D_i, 0)$ and define Φ_i with these gradients. The checkpoint update gives the two ratio bounds in [Lemma 3.1](#).

Constructing the circular arc for gradient directions. We use a circular arc to turn the gradient toward the negative Y -direction. Fix $\ell_i, D_i, \kappa, \beta$ and the gap stepsizes as in [Lemma 3.1](#). Set

$$R = 1 + \beta, \quad c_\beta = \beta + \sqrt{1 + 2\beta}, \quad p(q) = \beta + \sqrt{R^2 - (q + \beta)^2} \quad (0 \leq q \leq 1),$$

and let $K = \text{conv}(\{0\} \cup \{(p(q), -q) : 0 \leq q \leq 1\})$. The arc has center (β, β) , radius R , and endpoints $(c_\beta, 0)$ and $(\beta, -1)$. As q increases, $p(q)$ decreases from $c_\beta < 2$ to β . The disk contains 0 and hence K , so the radius vector $(p(q) - \beta, -q - \beta)$ is an outward normal to K at $(p(q), -q)$.

Constructing the gap trajectory. For parameters $\rho > 0$ and $y_0 \geq 1$, to be chosen, define the translated and rescaled coordinates by

$$r_j = (u_j, -y_j) = \frac{(X_j, Y_j) - (\ell_i + \rho y_0 / \beta, \rho y_0)}{\rho}.$$

The outgoing threshold will be $\ell_{i+1} = \rho y_0$. We select arc points $v_j = (p_j, -q_j)$, where $p_j = p(q_j)$, as the intended gradients divided by ρ . Substituting these coordinates and gradients ρv_{j-1} into (9) gives the required relation

$$r_j = \left(1 - \frac{h_j}{\kappa}\right) r_{j-1} - \frac{h_j}{\kappa} (\ell_i/\rho + y_0/\beta, y_0) - \frac{(1 - 1/\kappa)h_j}{2} v_{j-1} \quad (1 \leq j \leq m_{\text{gap}}). \quad (36)$$

We prescribe $r_{m_{\text{gap}}} = v_{m_{\text{gap}}} = (\beta, -1)$. This will put $Y_{m_{\text{gap}}}$ a distance ρ below ℓ_{i+1} , with $\partial_2 \Phi_i = -\rho$ at this point for the checkpoint update.

Lemma A.2 (Gap trajectory). *There is $a_\beta \geq 8$, depending only on β , with $\log a_\beta = O(1/\beta)$, such that for every $\rho > 0$ there exist points r_j and arc vectors v_j satisfying (36), with $r_{m_{\text{gap}}} = v_{m_{\text{gap}}} = (\beta, -1)$ and*

$$\Pi_K(r_j) = v_j, \quad u_j \geq \beta, \quad 1 \leq y_j \leq y_0 \leq 1 + \frac{1 - 1/\kappa}{2} s_i \quad (0 \leq j \leq m_{\text{gap}}).$$

Moreover, $\beta + y_0/\beta \leq a_\beta/4$, and y_0, u_0 can be chosen continuously in $\rho > 0$.⁶

Proof. If $h_j = 0$, set $r_{j-1} = r_j$ and $v_{j-1} = v_j$; the point and gradient are unchanged by this update. The maximum rule below is used only when $h_j > 0$. For an empty gap, take $r_0 = v_0 = (\beta, -1)$ and $y_0 = 1$.

The backward construction. For a nonempty gap, fix $\rho > 0$ and temporarily choose $y_0 \geq 1$. Starting from $u_{m_{\text{gap}}} = \beta$ and $y_{m_{\text{gap}}} = q_{m_{\text{gap}}} = 1$, compute $q_{j-1}, u_{j-1}, y_{j-1}$ for $j = m_{\text{gap}}, \dots, 2$. At $j = 1$, compute q_0, u_0 , leaving the second-coordinate equation as a condition on the trial y_0 .

Rearranging (36), we write $r_{j-1} = \hat{r}_j + \alpha_j v_{j-1}$, where

$$\hat{r}_j = (\hat{u}_j, -\hat{y}_j) = \frac{r_j + (h_j/\kappa)(\ell_i/\rho + y_0/\beta, y_0)}{1 - h_j/\kappa}, \quad \alpha_j = \frac{(1 - 1/\kappa)h_j}{2(1 - h_j/\kappa)}.$$

Set $\Delta_j = \hat{y}_j + (1 - \alpha_j)\beta$. We choose v_{j-1} so that $\Pi_K(r_{j-1}) = v_{j-1}$ also holds. For an interior arc point, (35) requires $r_{j-1} - v_{j-1}$ to be parallel to the radius vector of the circle. By the remaining update, this is equivalent to collinearity of the center (β, β) , v_{j-1} , and $\hat{r}_j + \alpha_j(\beta, \beta)$, as shown in Figure 4. Indeed, the update gives

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

The two shaded triangles in Figure 4 are therefore related by scaling by α_j and translation by $\hat{r}_j$. The center's vertical coordinate minus that of $\hat{r}_j + \alpha_j(\beta, \beta)$ is Δ_j . Equating the slopes, and including the endpoint $q_{j-1} = 0$, gives

$$\frac{q_{j-1} + \beta}{p_{j-1} - \beta} = \max \left\{ \frac{\beta}{c_\beta - \beta}, \frac{\Delta_j}{\hat{u}_j - (1 - \alpha_j)\beta} \right\}, \quad r_{j-1} = \hat{r}_j + \alpha_j v_{j-1}. \quad (37)$$

Backward induction gives $u_j \geq \beta$, so the denominator is at least $\alpha_j \beta > 0$. The left-hand side increases continuously from $\beta/(c_\beta - \beta)$ to infinity as q_{j-1} increases from 0 to 1. Thus the rule determines a unique $q_{j-1} \in [0, 1)$, continuously in $\hat{u}_j, \hat{y}_j$.

Choosing y_0 to satisfy the initial update. At $j = 1$, we still need $y_0 = \hat{y}_1 + \alpha_1 q_0$ (or $y_0 = y_1$ if $h_1 = 0$). Expanding the second coordinate of (36) from $y_{m_{\text{gap}}} = 1$ gives the equivalent condition

$$y_0 = 1 + \frac{1 - 1/\kappa}{2} \sum_{j=1}^{m_{\text{gap}}} h_j q_{j-1} \prod_{r=j+1}^{m_{\text{gap}}} (1 - h_r/\kappa).$$

⁶Here ℓ_i, κ, β and the gap stepsizes are fixed. The selected y_0, u_0 are functions of $\rho > 0$: small changes in ρ cause small changes in both values.

Bounding y_0 in terms of β . In addition to the bound in terms of s_i , we need a bound depending only on β to obtain the lower bound on ρ/D_i in (39). For $q_j > 0$, the collinearity of $(\beta, \beta), v_j, r_j$ gives

$$u_j - \beta = \frac{p_j - \beta}{q_j + \beta}(y_j + \beta).$$

Also, $\hat{u}_j \geq u_j$ and $\hat{y}_j \leq y_j$. Using $r_{j-1} = \hat{r}_j + \alpha_j v_{j-1}$ from (37), the gradient terms cancel:

$$p_{j-1}y_{j-1} - q_{j-1}u_{j-1} = p_{j-1}\hat{y}_j - q_{j-1}\hat{u}_j \leq p_{j-1}y_j - q_{j-1}u_j.$$

For $q_{j-1} > 0$, use the displayed formula for $u_j - \beta$ at indices $j-1$ and j to obtain

$$\begin{aligned} \frac{y_{j-1} + \beta}{q_{j-1} + \beta} &\leq \frac{y_j + \beta}{q_j + \beta} \left[1 + \frac{p_{j-1}q_j - q_{j-1}p_j}{\beta(p_{j-1} + q_{j-1})} \right] \\ &\leq \frac{y_j + \beta}{q_j + \beta} \exp \left[\frac{1}{\beta} \int_{q_{j-1}}^{q_j} \max\{1, -p'(t)\} dt \right]. \end{aligned} \quad (38)$$

For $q_{j-1} = 0$, the first inequality follows from $y_{j-1} = \hat{y}_j \leq y_j$. For $h_j = 0$, both inequalities are equalities. For the exponential bound, use $1 + x \leq e^x$ and $p_{j-1}q_j - q_{j-1}p_j = \int_{q_{j-1}}^{q_j} [p_{j-1} - q_{j-1}p'(t)] dt$. The integrand divided by $p_{j-1} + q_{j-1}$ is a weighted average of 1 and $-p'(t)$, so

$$\frac{p_{j-1} - q_{j-1}p'(t)}{p_{j-1} + q_{j-1}} \leq \max\{1, -p'(t)\}.$$

Applying (38) for $j = m_{\text{gap}}, \dots, 1$ multiplies the exponential factors, whose integrals add from q_0 to $q_{m_{\text{gap}}} = 1$. Since $y_{m_{\text{gap}}} = q_{m_{\text{gap}}} = 1$, we obtain

$$\begin{aligned} y_0 + \beta &\leq (q_0 + \beta) \exp \left[\frac{1}{\beta} \int_{q_0}^1 \max\{1, -p'(t)\} dt \right] \\ &\leq \beta \exp \left[\frac{1}{\beta} \int_0^1 \max\{1, -p'(t)\} dt \right]. \end{aligned}$$

The second inequality uses $\log(1 + q_0/\beta) \leq q_0/\beta$ and $\max\{1, -p'(t)\} \geq 1$. Since $p'(t) = -(t + \beta)/(p(t) - \beta)$ for $t < 1$, the two entries of the maximum agree at $t = R/\sqrt{2} - \beta$. Integrating 1 below this point and $-p'(t)$ above it gives

$$\int_0^1 \max\{1, -p'(t)\} dt = \left(\frac{R}{\sqrt{2}} - \beta \right) + p \left(\frac{R}{\sqrt{2}} - \beta \right) - p(1) = \sqrt{2}R - \beta.$$

Hence $y_0 + \beta \leq \beta \exp[(\sqrt{2}R - \beta)/\beta]$. Choose

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

Then $\beta + y_0/\beta \leq a_\beta/4$, $a_\beta \geq 8$, and $\log a_\beta = O(1/\beta)$. □

We now use Lemma A.2 to prove Lemma 3.1.

Proof of Lemma 3.1.

Step 1. Matching the initial point. Since $Y_0 = 0$, it remains to choose ρ so that $\rho(u_0 + y_0/\beta) = D_i$. Expanding the first coordinate of (36) and using $0 \leq p_j \leq c_\beta$ gives

$$\begin{aligned} \rho(y_0/\beta + \beta) - D_i \eta_i &\leq \chi_i[\rho(u_0 + y_0/\beta) - D_i] \\ &\leq \rho \left[y_0/\beta + \beta + \frac{1-1/\kappa}{2} c_\beta s_i \right] - D_i \eta_i. \end{aligned}$$

By Lemma A.2, $\rho(u_0 + y_0/\beta) = X_0 - \ell_i$ is continuous in ρ , and y_0 is uniformly bounded. Since $\chi_i > 0$ and $D_i \eta_i > 0$, these bounds give $X_0 < \ell_i + D_i$ for sufficiently small ρ and $X_0 > \ell_i + D_i$ for sufficiently large ρ . Choose ρ with $X_0 = \ell_i + D_i$, and set $\ell_{i+1} = \rho y_0$. The upper bound above and the two bounds on y_0 give

$$\frac{\rho}{D_i} \geq \frac{2\eta_i}{a_\beta/2 + (1-1/\kappa)c_\beta s_i}, \quad 1 \leq \frac{\ell_{i+1}}{\rho} \leq 1 + \frac{1-1/\kappa}{2} s_i. \quad (39)$$

Step 2. Defining the local hard function. Set

$$\Phi_i(X, Y) = (\text{env}_1 \sigma_{\rho K})(X - \ell_i - \ell_{i+1}/\beta, Y - \ell_{i+1}).$$

By Fact A.1, Φ_i is nonnegative, convex, and 1-smooth, with $\partial_2 \Phi_i \leq 0$ since K lies below the horizontal axis. Moreover, Lemma A.2 gives

$$\nabla \Phi_i(X_j, Y_j) = \rho \Pi_K(r_j) = \rho v_j.$$

The constructed points therefore satisfy (9). The coordinate change and the bounds on u_j, y_j give

$$X_j > \ell_i, \quad 0 \leq Y_j < \ell_{i+1} \quad (0 \leq j \leq m_{\text{gap}}).$$

At the endpoint, $r_{m_{\text{gap}}} = v_{m_{\text{gap}}} = (\beta, -1)$ gives

$$Y_{m_{\text{gap}}} = \ell_{i+1} - \rho, \quad \partial_2 \Phi_i(X_{m_{\text{gap}}}, Y_{m_{\text{gap}}}) = -\rho.$$

None of these choices uses b_i .

We next verify $\Phi_i(X, Y) = 0$ for $X \leq \ell_i, Y \geq 0$, so $\nabla \Phi_i = 0$ there. Use (34) with $g = \rho(p, -q)$, $(p, -q) \in K$. Since $0 \leq q \leq p/\beta$, the linear term is at most $\rho \ell_{i+1}(q - p/\beta) \leq 0$. The choice $g = 0$ attains zero, proving $\Phi_i(X, Y) = 0$.

For $Y \geq \ell_{i+1}$, we show $\partial_2 \Phi_i(X, Y) = 0$, so Φ_i does not contribute to the update of Y in this region. Since $(p, 0) \in K$ whenever $(p, -q) \in K$, replacing $(p, -q)$ by $(p, 0)$ increases the maximized expression by $\rho q(Y - \ell_{i+1}) + \rho^2 q^2/2 > 0$ if $q > 0$. Every maximizer therefore has $q = 0$, giving $\partial_2 \Phi_i(X, Y) = 0$ for every X .

Step 3. The checkpoint update. Substituting $Y_{m_{\text{gap}}} = \ell_{i+1} - \rho$ and $\partial_2 \Phi_i(X_{m_{\text{gap}}}, Y_{m_{\text{gap}}}) = -\rho$ into (9) gives

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

The last term subtracts the distance ρ still needed to reach ℓ_{i+1} . Using (39) and $4(1-1/\kappa)(1-s_i/\kappa) > 1$, we obtain for $b_i > 8$

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

Combining this with (39), using $\kappa \geq 4$ and $s_i/\kappa \leq 1/16$, gives the two ratio bounds (13). Finally, $c_\beta = \beta + \sqrt{1+2\beta} = 1 + O(\beta)$ and Lemma A.2 gives $\log a_\beta = O(1/\beta)$, proving (12). $\square$

A.2 Proof of Lemma 3.2

Before the formal proof, we first explain the choice of δ_H in (14). The intended trajectory starts at $(\ell_i + D_i, 0)$ and stays in the region $X - Y - \ell_i \geq 2\delta_H$ through the checkpoint query, where the Huber gradient is $(\delta_H, -\delta_H)$. Consider $f_j = 0$ first. The Huber component decreases X and increases Y by equal amounts, while the quadratic term contracts both coordinates toward zero. In particular, Y moves toward $(\kappa - 1)\delta_H/2$: each gap update multiplies the remaining distance to this height by $1 - h_j/\kappa$. Starting from $Y_0 = 0$, this gives

$$Y_{m_{\text{gap}}} = \frac{(\kappa - 1)(1 - \chi_i)}{2} \delta_H.$$

The Huber contributions cancel in $X + Y$, so quadratic contraction also gives $X_{m_{\text{gap}}} + Y_{m_{\text{gap}}} - \ell_i = \chi_i(\ell_i + D_i) - \ell_i = D_i\eta_i$. Placing the checkpoint query on the boundary $X - Y - \ell_i = 2\delta_H$ of the constant-gradient region gives $D_i\eta_i = 2\delta_H + 2Y_{m_{\text{gap}}} = [2 + (\kappa - 1)(1 - \chi_i)]\delta_H$, which yields (14). Since Y is nondecreasing during the gap, setting $\ell_{i+1} = Y_{m_{\text{gap}}}$ keeps the next component inactive through the checkpoint query; the checkpoint update then raises Y above this threshold. A nonnegative contribution f_j from the preceding component only moves X farther to the right during the gap, preserving the constant gradient and the same output trajectory. The proof below verifies these properties without assuming in advance that the queries stay in the required region.

Proof of Lemma 3.2. Let $K = \text{conv}\{0, (\delta_H, -\delta_H)\}$. By (34), the Huber repair component equals $(\text{env}_1 \sigma_K)(X - \ell_i, Y)$. It is therefore nonnegative, convex, and 1-smooth. It vanishes when $X \leq \ell_i$, $Y \geq 0$, and its gradient is

$$\nabla \Phi_i(X, Y) = \min\{\delta_H, \frac{1}{2} \max\{X - Y - \ell_i, 0\}\}(1, -1).$$

In particular, $\partial_2 \Phi_i \leq 0$ everywhere.

The gap trajectory. For $1 \leq j \leq m_{\text{gap}}$, each gradient coordinate has magnitude at most δ_H . Together with $f_j \geq 0$, this gives

$$X_j - Y_j - \ell_i \geq \left(1 - \frac{h_j}{\kappa}\right)(X_{j-1} - Y_{j-1} - \ell_i) - \frac{h_j \ell_i}{\kappa} - \left(1 - \frac{1}{\kappa}\right)h_j \delta_H.$$

Expanding this recurrence from $X_0 - Y_0 - \ell_i = D_i$, we use the telescoping identity

$$\sum_{r=1}^j h_r \prod_{l=r+1}^j (1 - h_l/\kappa) = \kappa \left[1 - \prod_{l=1}^j (1 - h_l/\kappa) \right] \leq \kappa(1 - \chi_i).$$

The contraction product over the first j updates is at least χ_i , so, for $0 \leq j \leq m_{\text{gap}}$,

$$\begin{aligned} X_j - Y_j - \ell_i &\geq \chi_i(\ell_i + D_i) - \ell_i - (\kappa - 1)\delta_H(1 - \chi_i) \\ &= D_i\eta_i - (\kappa - 1)\delta_H(1 - \chi_i) = 2\delta_H, \end{aligned}$$

where the last equality follows from (14). Thus every gap query and the checkpoint query has gradient $(\delta_H, -\delta_H)$. Solving the recurrence for Y_j in (10) gives

$$\begin{aligned} X_j - Y_j - \ell_i &\geq 2\delta_H, \\ Y_j &= \frac{\kappa - 1}{2} \delta_H \left[1 - \prod_{l=1}^j (1 - h_l/\kappa) \right], \quad 0 \leq j \leq m_{\text{gap}}. \end{aligned} \tag{40}$$

At $j = m_{\text{gap}}$, the product equals χ_i and $Y_{m_{\text{gap}}} = \ell_{i+1}$. In particular, $X_j \geq \ell_i$ and $0 \leq Y_j \leq \ell_{i+1}$ throughout the gap.

The checkpoint transfer. At the checkpoint query, $Y_{m_{\text{gap}}} = \ell_{i+1}$ and the gradient is still $(\delta_H, -\delta_H)$. The checkpoint update gives

$$\begin{aligned} D_{i+1} &= Y_{m_{\text{gap}}+1} - \ell_{i+1} = b_i \left(\frac{1 - 1/\kappa}{2} \delta_H - \frac{\ell_{i+1}}{\kappa} \right) \\ &= \frac{1 - 1/\kappa}{2} b_i \delta_H \chi_i > 0. \end{aligned}$$

This holds for every $b_i > 0$, including $b_i \geq \kappa$, and every $f_{m_{\text{gap}}+1} \geq 0$. This proves (8). Substituting δ_H gives the two equalities in (15). At $j = m_{\text{gap}}$, each product in the telescoping sum lies between χ_i and 1, so $\chi_i s_i \leq \kappa(1 - \chi_i) \leq s_i$. These bounds give the lower bound in (15) and, together with $\chi_i \geq 1 - s_i/\kappa$ for $s_i < \kappa$, the ratio bounds in (16). $\square$

A.3 Proof of Lemma 3.3

The bridge follows the same trajectory as repairing through the checkpoint update and has $\partial_2 \Phi_i(X, Y) = 0$ whenever $Y \geq \ell_{i+1}$, independently of X . To obtain this horizontal boundary, recall that the Huber gradient is the projection of $(X - \ell_i, Y)$ onto the segment $\text{conv}\{0, (\delta_H, -\delta_H)\}$. Translating both coordinates by $-\ell_{i+1}$ leaves their difference $X - Y - \ell_i$ unchanged: this translation is parallel to the boundary of the Huber constant-gradient region and changes neither the Huber function nor its gradient. We therefore use the translated query $(X - \ell_i - \ell_{i+1}, Y - \ell_{i+1})$ and enlarge the segment to K_Δ by adding $(\delta_H, 0)$. When $Y \geq \ell_{i+1}$, the translated query lies on or above the horizontal axis, so its projection onto the triangle lies on the horizontal edge, giving the desired zero output derivative.

To preserve the repairing trajectory, every query through the checkpoint must still project onto the lower vertex $(\delta_H, -\delta_H)$. Besides the Huber condition $X - Y - \ell_i \geq 2\delta_H$, the triangle requires $Y - \ell_{i+1} \leq -\delta_H$: the translated query must lie at or below that vertex. Since the repairing output is nondecreasing during the gap, the smallest threshold satisfying this condition is

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

This explains the choice in Lemma 3.3.

The translation above is used only in the projection formula; GD still evolves the original coordinates (X, Y) . By Section A.2, the two conditions above remain valid for nonnegative contributions f_j , so both components have gradient $(\delta_H, -\delta_H)$ at the same iterates through the checkpoint query. With the same initial point, stepsizes, and f_j , their GD trajectories therefore agree through the checkpoint update. The bridge's higher threshold reduces the outgoing amplitude by δ_H ; the resulting amplitude remains positive under the lemma's assumptions.

Proof of Lemma 3.3. By (34), the bridge component equals $(\text{env}_1 \sigma_{K_\Delta})(X - \ell_i - \ell_{i+1}, Y - \ell_{i+1})$, where $\ell_{i+1} = D_i \eta_i / 2 > 0$. By Fact A.1, it is nonnegative, convex, and 1-smooth, and its gradient lies in K_Δ . In particular, $\partial_2 \Phi_i \leq 0$ everywhere.

If $X \leq \ell_i$ and $Y \geq 0$, the linear term in the maximum defining Φ_i has values

$$\delta_H(X - Y - \ell_i) \leq 0, \quad \delta_H(X - \ell_i - \ell_{i+1}) \leq 0$$

at the two nonzero vertices of K_Δ . It is therefore nonpositive throughout K_Δ , and the maximum is zero, attained at $g = 0$. Thus $\Phi_i(X, Y) = 0$. For $Y \geq \ell_{i+1}$, if $(p, -q) \in K_\Delta$ with $q > 0$, replacing it by $(p, 0) \in K_\Delta$ increases the maximized expression by $q(Y - \ell_{i+1}) + q^2/2 > 0$. The maximizing gradient consequently has second coordinate zero, proving $\partial_2 \Phi_i(X, Y) = 0$ for every X .

The gap trajectory. Fix any sequence $f_j \geq 0$ for $1 \leq j \leq m_{\text{gap}} + 1$, and consider the Huber trajectory (X_j, Y_j) from [Section A.2](#) with the same incoming data and stepsizes. We show that these states also satisfy the bridge updates. The bridge threshold is

$$\ell_{i+1} = \frac{D_i \eta_i}{2} = \delta_H + \frac{\kappa - 1}{2} \delta_H (1 - \chi_i).$$

Consequently, (40) implies, for $0 \leq j \leq m_{\text{gap}}$,

$$X_j - Y_j - \ell_i \geq 2\delta_H, \quad 0 \leq Y_j \leq Y_{m_{\text{gap}}} = \ell_{i+1} - \delta_H.$$

Restricting the maximum defining Φ_i to the segment $\text{conv}\{0, (\delta_H, -\delta_H)\}$ gives the Huber function $\frac{1}{2}H_{2\delta_H}(X - Y - \ell_i)$. At each of the displayed states, its gradient is $(\delta_H, -\delta_H)$, so (35) holds on this segment with $x = (X_j - \ell_i - \ell_{i+1}, Y_j - \ell_{i+1})$. At the remaining vertex $(\delta_H, 0)$, the left-hand side of that condition is

$$\delta_H(Y_j - \ell_{i+1} + \delta_H) \leq 0.$$

The condition therefore holds on the convex hull K_Δ . Hence [Fact A.1](#) gives

$$\nabla \Phi_i(X_j, Y_j) = (\delta_H, -\delta_H) \quad (0 \leq j \leq m_{\text{gap}}).$$

Induction in (10), starting from $(\ell_i + D_i, 0)$ and using the same f_j , now shows that the Huber and bridge trajectories agree through update $m_{\text{gap}} + 1$. In particular, the bridge satisfies $X_j \geq \ell_i$ and $0 \leq Y_j \leq \ell_{i+1} - \delta_H < \ell_{i+1}$ through the checkpoint query.

The checkpoint transfer. At the checkpoint query, $Y_{m_{\text{gap}}} = \ell_{i+1} - \delta_H$ and the gradient is $(\delta_H, -\delta_H)$. The checkpoint update gives

$$\begin{aligned} D_{i+1} &= Y_{m_{\text{gap}}+1} - \ell_{i+1} = b_i \left(\frac{1 - 1/\kappa}{2} \delta_H - \frac{Y_{m_{\text{gap}}}}{\kappa} \right) - \delta_H \\ &= \delta_H \left(\frac{(1 - 1/\kappa)b_i \chi_i}{2} - 1 \right). \end{aligned} \tag{41}$$

Since $s_i/\kappa \leq 1/16$, we have $\chi_i \geq 15/16$. Together with $\kappa \geq 4$, this gives $(1 - 1/\kappa)\chi_i \geq 45/64 > 1/2$. Thus, for every $b_i > 4$,

$$\frac{1 - 1/\kappa}{2} b_i \chi_i - 1 \geq \frac{1 - 1/\kappa}{2} \chi_i (b_i - 4) > 0.$$

Hence $D_{i+1} > 0$, which, together with the gap bounds, proves (8). Substituting δ_H into (41) and using $(\kappa - 1)(1 - \chi_i) \leq s_i$ gives

$$\begin{aligned} \frac{D_{i+1}}{D_i} &= \frac{\eta_i[(1 - 1/\kappa)b_i \chi_i - 2]}{2[2 + (\kappa - 1)(1 - \chi_i)]} \\ &\geq \frac{\eta_i(1 - 1/\kappa)\chi_i(b_i - 4)}{2(s_i + 2)} \geq \frac{1}{4} \eta_i \frac{b_i - 4}{s_i + 2}. \end{aligned}$$

Since $\ell_{i+1} = D_i \eta_i / 2$, the same estimates give

$$\begin{aligned} \frac{\ell_{i+1}}{D_{i+1}} &= \frac{2 + (\kappa - 1)(1 - \chi_i)}{(1 - 1/\kappa)b_i \chi_i - 2} \\ &\leq \frac{s_i + 2}{(1 - 1/\kappa)\chi_i(b_i - 4)} \leq 2 \frac{s_i + 2}{b_i - 4}. \end{aligned}$$

□

B Proof of Lemma 4.1

Let $1 \leq c \leq 2$ and $a \geq 8$ be fixed, and define

$$p = \log_2(1 + \sqrt{1+c}), \quad v = 1/p.$$

Consequently, $1 < p < 2$, and v is bounded within the fixed interval $[\log_{1+\sqrt{3}} 2, \log_{1+\sqrt{2}} 2] \subset (0, 1)$. Throughout this section, all numerical constants remain independent of c and a .

We first prove a scalar splitting inequality in Section B.1, then use it to bound an auxiliary cost introduced in Section B.2. These bounds control the unselected mass when the empty checkpoint set is optimal (Section B.3). We then derive all four checkpoint bounds in Section B.4.

B.1 The scalar splitting inequality

Lemma B.1 (Scalar splitting). *If $x, y \geq a$, $B > 0$, and*

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

then $W^v \leq x^v + y^v$.

Proof. Our goal is to show $W \leq M$, where $M := (x^v + y^v)^{1/v}$.

First, we characterize W by eliminating B . From $B = W - x - y + a$ and $WB = cxy$, we have

$$W(W - x - y + a) = cxy \implies W^2 - (x + y)W - cxy = -aW.$$

Since $x, y \geq a \geq 8$ and $B > 0$, it is clear that $W > 0$. Thus, the right-hand side is strictly negative, meaning W evaluates to a negative value in the quadratic polynomial $Q(X) = X^2 - (x + y)X - cxy$. Consequently, W must be strictly less than the unique positive root R of $Q(X) = 0$. To establish $W \leq M$, it therefore suffices to show $R \leq M$. Since $Q(X)$ opens upwards, $R \leq M$ holds if and only if $Q(M) \geq 0$, which is equivalent to

$$M^2 - (x + y)M - cxy \geq 0.$$

Dividing by M^2 , our target inequality becomes

$$\frac{x}{M} + \frac{y}{M} + c \left(\frac{x}{M} \right) \left(\frac{y}{M} \right) \leq 1.$$

By the definition of M , we know $(x/M)^v + (y/M)^v = 1$. This naturally suggests the substitution $z = (x/M)^v$, which forces $1 - z = (y/M)^v$. Since $x, y > 0$, we have $z \in (0, 1)$. Recalling that $p = 1/v$, we can rewrite $x/M = z^p$ and $y/M = (1 - z)^p$. The inequality then reduces to demonstrating that for all $z \in [0, 1]$,

$$f(z) := z^p + (1 - z)^p + c[z(1 - z)]^p \leq 1.$$

We now analyze $f(z)$. By the definition of p , we have $2^p - 1 = \sqrt{1+c}$, which yields $c = 2^{2p} - 2^{p+1}$. It is easy to verify that $f(0) = 1$ and, using this exact value of c , that $f(1/2) = 1$. To determine the behavior of $f(z)$ on $(0, 1/2)$, we examine its derivative:

$$f'(z) = p[z(1 - z)]^{p-1} \underbrace{\left((1 - z)^{1-p} - z^{1-p} + c(1 - 2z) \right)}_{g(z)}.$$

Because $p > 0$ and $z \in (0, 1/2)$, the sign of $f'(z)$ is identical to the sign of $g(z)$. The second derivative $g''(z) = p(p-1)[(1-z)^{-p-1} - z^{-p-1}]$ is strictly negative since $p > 1$ and $z < 1 - z$. Thus, g is strictly concave. Observing that $g(0^+) = -\infty$, $g(1/2) = 0$, and $g'(1/2) = 2^{p+1}(p+1-2^p) < 0$, we see that g crosses zero exactly once in $(0, 1/2)$, transitioning from negative to positive. Consequently, $f'(z)$ starts negative and becomes positive, meaning $f(z)$ decreases from 1 and then increases back to 1 at $z = 1/2$. By symmetry, $f(z) \leq 1$ for every $z \in [0, 1]$. This confirms $Q(M) \geq 0$, yielding $W < R \leq M$, completing the proof. $\square$

Analogous inequalities form the foundation of prior analyses in the smooth convex setting; see, for example, [Jung et al. \[2026, Lemma C.4\]](#).

B.2 Bounding the minimum auxiliary cost

To study the score P_w in (19), we introduce an auxiliary list. Let $g_1, \dots, g_r, B_1, \dots, B_r$ be nonnegative numbers, arranged in the ordered auxiliary list

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

We use this auxiliary list to calculate checkpoint scores. Only positions with $B_j > 0$ may be selected, and the g_j entries are never selected. For a selected B_j , let 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 the auxiliary cost $Q_\Lambda(T)$ by

$$P(T) = \prod_{j \in T} \frac{B_j}{c(a + s_j)}, \quad Q_\Lambda(T) = \frac{c(\Lambda + s_{\text{tail}})}{P(T)}, \quad \Lambda \geq a.$$

Here, following the idea of dynamic programming, we introduce a terminal parameter Λ so that the auxiliary problem is closed when splitting into subproblems. For the empty set, $P(\emptyset) = 1$ and $Q_\Lambda(\emptyset) = c[\Lambda + \sum_j (g_j + B_j)]$. Write $V_\Lambda(g, B) = \min_T Q_\Lambda(T)$ for the minimum auxiliary cost.

For the original problem, $B_j = (h_j - 8)_+$ and $g_j = \min\{h_j, 8\}$, with w added to g_1 , give $P(T) = P_w(T; H)$; Λ records the fixed contribution to the next fixed checkpoint's denominator, so including that checkpoint makes the score proportional to $1/Q_\Lambda(T)$.

The next lemma describes how inserting a checkpoint into a gap changes Q_Λ .

Lemma B.2 (Insertion ratio). *Suppose that a new checkpoint of size $B > 0$ is inserted into a gap. Let $l \geq 0$ be the sum of the entries in the gap before this checkpoint, and let $K \geq a$ be the sum of the entries after it plus the terminal offset of the gap (either a or $\Lambda \geq a$). Then inserting the checkpoint multiplies the auxiliary cost Q_Λ by*

$$I(l, K; B) = \frac{c(a + l)K}{B(l + B + K)}.$$

Moreover, for fixed B , the function $I(l, K; B)$ is strictly increasing in both l and K .

Proof. Before the new checkpoint is inserted, the contribution of the gap to the auxiliary cost Q_Λ is $c(l + B + K)$. After insertion, the checkpoint contributes the reciprocal factor $c(a + l)/B$, while the part of the list after the checkpoint contributes cK . Thus the ratio between the new and old costs is

$$I(l, K; B) = \frac{c(a + l)K}{B(l + B + K)}.$$

Its logarithmic derivatives are

$$\frac{\partial}{\partial l} \log I = \frac{1}{a + l} - \frac{1}{l + B + K} = \frac{B + K - a}{(a + l)(l + B + K)} > 0$$

and

$$\frac{\partial}{\partial K} \log I = \frac{1}{K} - \frac{1}{l + B + K} = \frac{l + B}{K(l + B + K)} > 0.$$

Here we used $B > 0$ and $K \geq a$. Hence I is strictly increasing in each of l and K . □

With the above lemma, we can prove the log-submodularity of the auxiliary cost Q_Λ . In other words, the change in $\log Q_\Lambda$ upon inserting a checkpoint is nonincreasing as more checkpoints are selected.

Lemma B.3 (Log-submodularity). *For any two admissible checkpoint sets S, T ,*

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

If S and T are disjoint and both nonempty, the inequality is strict.

Proof. Start with the checkpoint set $S \cap T$. Insert the checkpoints in $S \setminus T$ and $T \setminus S$, one at a time. The insertion ratio from Lemma B.2 is increasing in the sums l and K on either side of the inserted checkpoint. Adding other checkpoints can only split the containing gap and hence decrease these two sums. Consequently, the factor by which Q_Λ changes upon inserting the checkpoints in $S \setminus T$ is no smaller when they are inserted into $S \cap T$ than when they are inserted into T . Therefore

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

which is the desired inequality.

If S and T are disjoint and nonempty, at least one insertion takes place after another checkpoint has strictly shortened the relevant gap. At that insertion, at least one of l or K decreases strictly. The strict monotonicity in Lemma B.2 then gives strict inequality. $\square$

We seek a bound on $V_\Lambda(g, B)$ independent of B . The next lemma treats the case where the empty set minimizes Q_Λ , showing that the total mass of the auxiliary list is then controlled by Λ , a , and the nonselectable entries g_j . This case will suffice: we subsequently show that, for fixed g and Λ , $V_\Lambda(g, B)$ attains its maximum over B where the empty set is a minimizer.

Lemma B.4. *Suppose that $Q_\Lambda(T) \geq Q_\Lambda(\emptyset)$ for every admissible T . Then*

$$\left[\Lambda + \sum_{j=1}^r (g_j + B_j) \right]^v \leq \Lambda^v + \sum_{j=1}^r (a + g_j)^v.$$

Proof. We argue by induction on r . The assertion is immediate when $r = 0$. Now let $r \geq 1$, and assume that the statement holds for every auxiliary list with fewer than r pairs. We prove the statement for a list with r pairs. Write

$$W = \Lambda + \sum_{j=1}^r (g_j + B_j).$$

Treat g_j 's and Λ as fixed and B_j 's as variable. Among all $B \geq 0$ for which the empty set minimizes Q_Λ , choose one that maximizes W .

We first justify the existence of a maximizer by proving that the feasible set is compact. It is nonempty because $B = 0$ is feasible. For boundedness, the singleton constraint $Q_\Lambda(\{j\}) \geq Q_\Lambda(\emptyset)$ gives, whenever $B_j > 0$,

$$B_j \leq c \left(a + \sum_{i < j} (g_i + B_i) + g_j \right).$$

The same inequality holds when $B_j = 0$. Since the g_i are fixed, these inequalities bound $B_1, \dots, B_r$ successively. For closedness, consider a convergent sequence of feasible vectors. Every checkpoint set admissible at the limit remains admissible for all sufficiently large indices, since its selected entries have positive limits.

Its cost therefore converges, and the constraint $Q_\Lambda(T) \geq Q_\Lambda(\emptyset)$ passes to the limit. Thus the limit is feasible, proving closedness. The feasible set is consequently compact, so the continuous function W attains its maximum.

Next, suppose that $B_j = 0$ at a maximizer. Since this position cannot be selected, it may be removed from the list. If $j < r$, replace g_{j+1} by $g_j + g_{j+1}$. This leaves both W and all auxiliary costs $Q_\Lambda(T)$ unchanged. Moreover,

$$(a + g_j + g_{j+1})^v \leq (a + g_j)^v + (a + g_{j+1})^v,$$

because $t \mapsto t^v$ is subadditive. If $j = r$, remove the last pair and replace Λ by $\Lambda + g_r$. Again the auxiliary costs and W are unchanged, while

$$(\Lambda + g_r)^v \leq \Lambda^v + (a + g_r)^v.$$

The induction hypothesis therefore proves the result. Hence it remains to consider a maximizer for which every $B_j > 0$.

Now, call a nonempty checkpoint set T *tight* if

$$Q_\Lambda(T) = Q_\Lambda(\emptyset).$$

At least one nonempty set is tight: otherwise every constraint $Q_\Lambda(T) \geq Q_\Lambda(\emptyset)$ would have positive slack, and by continuity, one could slightly increase one of the B_j while preserving all constraints, thereby increasing W . By log-submodularity, the intersection and union of two tight sets are again tight. Indeed,

$$Q_\Lambda(S)Q_\Lambda(T) \geq Q_\Lambda(S \cap T)Q_\Lambda(S \cup T) \geq Q_\Lambda(\emptyset)^2,$$

and equality at the two ends forces equality throughout. Moreover, two tight sets cannot be disjoint, by the strict part of [Lemma B.3](#). It follows that the intersection of all nonempty tight sets is itself a nonempty tight set. Denote this minimal tight set by T_* .

We claim that T_* consists of a single checkpoint. Suppose instead that it contains two entries with values u and v . Under a perturbation of u and v , the only changing factors in the costs of tight nonempty sets are the denominator factors uv . If $u \neq v$, interchange the labels if necessary so that $u < v$, and perturb

$$u \mapsto u - t, \quad v \mapsto v + qt,$$

where $q > 1$ is chosen sufficiently close to 1. The empty-set cost increases only by $c(q-1)t$. On the other hand, for a tight nonempty set, the reciprocal product changes to first order by

$$\frac{1}{u-t} \frac{1}{v+qt} = \frac{1}{uv} \left[1 + \left(\frac{1}{u} - \frac{q}{v} \right) t + O(t^2) \right].$$

Since $u < v$, we may choose $q > 1$ close enough to 1 that $\frac{1}{u} - \frac{q}{v} > 0$, and then choose it so that the increase of every tight cost is larger than the increase of the empty cost. All nontight constraints have positive slack and therefore remain valid for sufficiently small t . This increases W , contradicting its maximality. If $u = v = b$, use instead

$$u \mapsto b - t, \quad v \mapsto b + t + \eta t^2.$$

Then

$$\frac{1}{(b-t)(b+t+\eta t^2)} = \frac{1}{b^2} \left[1 + \left(\frac{1}{b^2} - \frac{\eta}{b} \right) t^2 + O(t^3) \right],$$

whereas the empty-set cost increases only by $c\eta t^2$. Choosing $\eta > 0$ sufficiently small again yields a feasible perturbation that increases W , a contradiction. Thus T_* is a singleton. In particular, there is a tight singleton checkpoint, say at position j .

Finally, set

$$x = a + g_j + \sum_{i < j} (g_i + B_i), \quad y = \Lambda + \sum_{i > j} (g_i + B_i).$$

Then

$$W = x + y + B_j - a.$$

The equality $Q_\Lambda(\{j\}) = Q_\Lambda(\emptyset)$ is equivalent to

$$WB_j = cxy.$$

The empty set must minimize the auxiliary cost in the part of the list lying strictly to the left of j , regarded as a shorter problem with terminal offset $a + g_j$. Otherwise, an improving collection of checkpoints in that subproblem could be inserted before j , producing a checkpoint set in the original problem with cost below $Q_\Lambda(\{j\})$, and hence below $Q_\Lambda(\emptyset)$. The same argument shows that the empty set minimizes the auxiliary cost in the right subproblem, whose terminal offset is Λ . The induction hypothesis applied to the two shorter lists therefore bounds x^v and y^v by the corresponding portions of

$$\Lambda^v + \sum_{i=1}^r (a + g_i)^v.$$

Using $WB_j = cxy$ and applying [Lemma B.1](#), we obtain

$$W^v \leq x^v + y^v \leq \Lambda^v + \sum_{i=1}^r (a + g_i)^v.$$

This completes the induction. □

Next, we establish the attainment of the maximum of $V_\Lambda(g, B)$ over B .

Lemma B.5. *For fixed g and Λ , the function*

$$B \mapsto V_\Lambda(g, B)$$

attains its supremum on $[0, \infty)^r$.

Proof. We first show, by induction on r , that $V_\Lambda(g, B)$ is uniformly bounded in B . The assertion is immediate for $r = 0$. Suppose $r \geq 1$, and let

$$B_{\max} = \max_j B_j.$$

If $B_{\max} \leq 1$, then choosing no checkpoint gives

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

Now suppose $B_{\max} \geq 1$, and choose a position j for which $B_j = B_{\max}$. Use j as the first checkpoint. The factor contributed by the portion preceding B_j is at most

$$c \left[r + \frac{a + \sum_i g_i}{B_{\max}} \right].$$

The remaining part of the list contains fewer than r pairs, so its minimum auxiliary cost is uniformly bounded by the induction hypothesis. This gives a uniform bound for $V_\Lambda(g, B)$.

Next consider a sequence $B^{(n)}$ escaping every bounded subset of $[0, \infty)^r$. After passing to a subsequence, each coordinate either converges to a finite limit or tends to $+\infty$. Select the first coordinate that tends to $+\infty$ as the first checkpoint. The factor contributed before that checkpoint tends to zero, while the minimum auxiliary cost of the shorter list after it remains uniformly bounded. Consequently, $V_\Lambda(g, B^{(n)}) \rightarrow 0$. Since $V_\Lambda(g, 0) = c(\Lambda + \sum_j g_j) > 0$, no maximizing sequence can escape to infinity.

Finally, V_Λ is continuous at finite B , including points with zero coordinates. Indeed, checkpoint sets selecting a coordinate with $B_j \rightarrow 0$ have costs tending to $+\infty$, while checkpoint sets omitting that coordinate vary continuously and include the finite empty-set cost. Hence the minimum is locally determined by checkpoint sets that omit the vanishing coordinate. A maximizing sequence is therefore bounded, and continuity yields a finite maximizer. $\square$

At a maximizer of $V_\Lambda(g, B)$ over B , the empty set minimizes Q_Λ .

Lemma B.6. *For fixed g and Λ , let B^* maximize $V_\Lambda(g, B)$ over $B \in [0, \infty)^r$. Then*

$$V_\Lambda(g, B^*) = Q_\Lambda(\emptyset).$$

Proof. Suppose, to the contrary, that the empty set is not a minimizer at B^* . Let

$$m = V_\Lambda(g, B^*),$$

and let $\mathcal{M}$ be the family of checkpoint sets T satisfying $Q_\Lambda(T) = m$. Every set in $\mathcal{M}$ is then nonempty. Log-submodularity implies that the intersection of two minimizing sets is again minimizing. Furthermore, the intersection cannot be empty by the strict part of [Lemma B.3](#). Since $\mathcal{M}$ is finite, its total intersection is therefore nonempty.

Choose j belonging to every minimizing set. Slightly decrease B_j^* . For every $T \in \mathcal{M}$, the selected entry B_j^* appears only in the denominator of $Q_\Lambda(T)$. Thus every minimizing cost strictly increases. All other checkpoint costs that have positive slack remain above the new minimum for a sufficiently small perturbation. It follows that $V_\Lambda(g, B)$ increases, contradicting the maximality of B^* . Hence the empty set is a minimizer at B^* . $\square$

The following lemma gives an upper bound of the minimum auxiliary cost $V_\Lambda(g, B)$ that is independent of the B_j 's.

Lemma B.7 (Bound on the minimum auxiliary cost). *For arbitrary $g_1, \dots, g_r \geq 0$, arbitrary $B_1, \dots, B_r \geq 0$, and $\Lambda \geq a$,*

$$V_\Lambda(g, B) \leq c \left[\Lambda^v + \sum_{j=1}^r (a + g_j)^v \right]^p.$$

Proof. Fix g and Λ , and let B^* maximize $V_\Lambda(g, B)$ over $B \in [0, \infty)^r$. Such a maximizer exists by [Lemma B.5](#). By [Lemma B.6](#), the empty set minimizes Q_Λ at B^* . Therefore [Lemma B.4](#) gives

$$\left[\Lambda + \sum_j (g_j + B_j^*) \right]^v \leq \Lambda^v + \sum_j (a + g_j)^v.$$

Since $p = 1/v$,

$$V_\Lambda(g, B) \leq V_\Lambda(g, B^*) = Q_\Lambda(\emptyset) = c \left[\Lambda + \sum_j (g_j + B_j^*) \right] \leq c \left[\Lambda^v + \sum_j (a + g_j)^v \right]^p$$

for every B , which proves the claim. $\square$

B.3 Bounding total mass when the maximum checkpoint score is one

We now consider the checkpoint score P without a fixed checkpoint to the right of the list, and assume $\max_T P(T) = 1$. Here the empty set maximizes P , whereas the empty-set condition in the preceding subsection concerned minimizing Q_Λ .

To bound the total mass under the condition $\max_T P(T) = 1$, we first control how many entries B_j can exceed a given threshold. The auxiliary-cost bound implies that too many such entries would allow a checkpoint set with score greater than one.

Lemma B.8 (A level-set bound). *Let $M = \sum_{j=1}^r (a + g_j)^\nu$. If $\max_T P(T) = 1$, then for every $q > 0$,*

$$\#\{j : B_j > q\} \leq c^\nu M q^{-\nu}.$$

Proof. Fix $q > 0$, and let

$$j_1 < \cdots < j_k$$

be the positions for which $B_{j_m} > q$. If $k = 0$, the claim is immediate. Assume henceforth that $k \geq 1$. We construct a checkpoint set that contains all these k positions. Before each selected position j_m , independently optimize the checkpoints lying in the gap after j_{m-1} and before j_m , with the convention that the first gap begins at the start of the list. Appending B_{j_m} to the optimized checkpoints in its preceding gap contributes a factor

$$\frac{B_{j_m}}{V_m},$$

where V_m is the minimum auxiliary cost of that gap with terminal offset $a + g_{j_m}$. By Lemma B.7,

$$V_m \leq c M_m^p,$$

where M_m is the sum of $(a + g_i)^\nu$ over the positions assigned to that gap, including its right endpoint j_m . The k groups are disjoint, and hence

$$\sum_{m=1}^k M_m \leq M.$$

Since $B_{j_m} > q$, the constructed checkpoint score satisfies

$$\max_T P(T) \geq \prod_{m=1}^k \frac{q}{c M_m^p}.$$

By the arithmetic–geometric mean inequality,

$$\prod_{m=1}^k M_m \leq \left(\frac{1}{k} \sum_{m=1}^k M_m \right)^k \leq \left(\frac{M}{k} \right)^k.$$

Therefore

$$1 = \max_T P(T) \geq \left(\frac{q}{c} \right)^k \left(\frac{k}{M} \right)^{pk}.$$

Taking k -th roots and using $p = 1/\nu$, we obtain

$$1 \geq \frac{q}{c} \left(\frac{k}{M} \right)^p,$$

or equivalently

$$k \leq c^\nu M q^{-\nu}.$$

□

We now convert the preceding counting estimate into a bound on the total mass of the auxiliary list. Applied to the original stepsizes, this controls the mass of a block together with its incoming unselected tail when the empty set maximizes the score, and the remaining tail after the last checkpoint of a nonempty maximizing set.

Lemma B.9. *Let $M = \sum_{j=1}^r (a + g_j)^\nu$. If $\max_T P(T) = 1$, then*

$$\sum_j (g_j + B_j) \leq \left(1 + \frac{c}{1-\nu}\right) M^p.$$

Proof. If the auxiliary list is empty, both sides of the claimed bound are zero. Otherwise, Lemma B.8 gives

$$N(q) := \#\{j : B_j > q\} \leq c^\nu M q^{-\nu}.$$

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

$$1 \leq c^\nu M q^{-\nu}.$$

Letting $q \uparrow \max_j B_j$ gives

$$\max_j B_j \leq c M^{1/\nu} = c M^p.$$

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

$$\sum_j B_j = \int_0^{c M^p} N(q) dq \leq c^\nu M \int_0^{c M^p} q^{-\nu} dq = \frac{c}{1-\nu} M^p.$$

It remains to control the sum of g_j . Since $0 < \nu < 1$, subadditivity gives

$$\left(\sum_j g_j\right)^\nu \leq \sum_j g_j^\nu \leq \sum_j (a + g_j)^\nu = M.$$

Combining the two estimates gives

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

□

B.4 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 $r \leq m$, and let w be the mass of the unselected tail from preceding blocks. Write

$$B_j = (h_j - 8)_+, \quad g_j = \min\{h_j, 8\},$$

so that $h_j = g_j + B_j$. To incorporate the carried mass w , define the auxiliary list

$$g_1 + w, B_1, g_2, B_2, \dots, g_r, B_r.$$

A position is selectable in the auxiliary list exactly when $B_j > 0$, or equivalently when $h_j > 8$. The auxiliary checkpoint score is exactly $P_w(T; H)$.

For this auxiliary list, write

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

Since $0 < v < 1$, subadditivity of t^v gives

$$(a + g_1 + w)^v \leq w^v + (a + g_1)^v.$$

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

$$M \leq w^v + r(a + 8)^v \leq w^v + 2^v r a^v.$$

Using $(u + v)^p \leq 2^{p-1}(u^p + v^p)$ and $pv = 1$, we obtain

$$M^p \leq C(w + ar^p) \leq C(w + am^p), \quad (42)$$

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

Proof of (i). Suppose that the empty set maximizes P_w . Since $P_w(\emptyset; H) = 1$, this means

$$\max_T P_w(T; H) = 1.$$

We may therefore apply [Lemma B.9](#) to the auxiliary list. Its total mass is

$$w + \sum_{j=1}^r (g_j + B_j) = w + \sum_{j=1}^r h_j.$$

Thus

$$w + \sum_{j=1}^r h_j \leq \left(1 + \frac{c}{1-v}\right) M^p.$$

Both c and $(1-v)^{-1}$ are uniformly bounded over the stated range of parameters. Combining this estimate with (42) gives

$$w + \sum_{j=1}^r h_j \leq C_b(w + am^p).$$

This proves part (i).

Proof of (ii). Now let $T = \{t_1 < \dots < t_k\}$ be a nonempty maximizing checkpoint set, and put $t_0 = 0$. For each i , consider the positions after t_{i-1} and up to t_i . In this group, the selected position t_i serves as the right endpoint. Including the endpoint term in each group, define

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

We claim that

$$d_i \leq cM_i^p. \quad (43)$$

To see this, temporarily keep the selected endpoint t_i fixed and consider inserting additional checkpoints into the gap preceding it. If some such insertion, or collection of insertions, increased the product, then adding those checkpoints to T would produce a checkpoint set with product strictly larger than $P_w(T; H)$. This

would contradict the maximality of T . Thus the empty checkpoint set minimizes the auxiliary cost of this gap, and the minimum auxiliary cost is

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

If $i = 1$ and $t_1 = 1$, there are no unselected positions and $d_1 = c(a + 8 + w) = cM_1^p$ directly. Otherwise, apply [Lemma B.7](#) to the unselected positions before t_i , with terminal offset $\Lambda = a + g_{t_i} = a + 8$. The term Λ^v is exactly the endpoint contribution in M_i ; in the first gap, use $g_1 + w$ as the first nonselectable entry. This gives (43) in every case. The groups defining the M_i 's are disjoint, so

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

Since $p \geq 1$,

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

It remains to control the tail after t_k . Consider the sublist

$$h_{t_k+1}, \dots, h_r.$$

Its carried mass is zero, because the gap begins immediately after the selected checkpoint t_k . If some checkpoint set in this tail had product greater than 1, adjoining it to T would multiply $P_w(T; H)$ by a factor greater than 1, contradicting maximality. Therefore the maximum checkpoint score in the tail equals 1. If the tail is empty, its mass is zero. Otherwise apply [Lemma B.9](#) to the tail. Since its corresponding quantity satisfies

$$M_{\text{tail}} := \sum_{j=t_k+1}^r (a + g_j)^v \leq r(a + 8)^v \leq 2^v r a^v,$$

we have

$$s_{\text{tail}} = \sum_{j=t_k+1}^r h_j \leq \left(1 + \frac{c}{1-v} \right) M_{\text{tail}}^p \leq C_b a m^p.$$

This proves part (ii).

Proof of (iii). In parts (iii)–(iv), we return to the checkpoint-indexed notation $B_i := h_{t_i} - 8 > 0$ from (19). Consider two consecutive checkpoints $t_i < t_{i+1}$, and write

$$x = a + s_i + 8, \quad y = a + s_{i+1} + 8, \quad B = B_i.$$

Thus $d_i = cx$ and $d_{i+1} = cy$. 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

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

All factors outside these two checkpoints remain unchanged. Before deletion, the relevant part of the product is

$$\frac{B}{cx} \frac{B_{i+1}}{cy},$$

whereas after deletion it is

$$\frac{B_{i+1}}{c(x + y + B - a)}.$$

Since T is a maximizer, deleting a checkpoint cannot increase its product. Therefore

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

Cancelling the positive common factors gives

$$B(x+y+B-a) \geq cxy. \quad (44)$$

We next prove

$$\frac{xy}{B} \leq 2(x+y). \quad (45)$$

If $B \leq x+y$, then (44) 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,

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

This proves (45). Since $d_i = cx$ and $d_{i+1} = cy$, we now have

$$\frac{d_i d_{i+1}}{B_i} = c^2 \frac{xy}{B} \leq 2c^2(x+y) = 2c(d_i + d_{i+1}) \leq 4(d_i + d_{i+1}),$$

because $c \leq 2$. Moreover, $s_{i+1} \leq y$, so

$$\frac{s_{i+1} d_i}{B_i} = c \frac{s_{i+1} x}{B} \leq c \frac{xy}{B} \leq 2c(x+y) = 2(d_i + d_{i+1}).$$

This proves all the inequalities in part (iii).

Proof of (iv). Delete only the final selected checkpoint t_k . None of the factors associated with the earlier selected checkpoints changes. Therefore

$$P_w(T; H) = P_w(T \setminus \{t_k\}; H) \frac{B_k}{d_k}.$$

Because T is a maximizer,

$$P_w(T; H) \geq P_w(T \setminus \{t_k\}; H).$$

Both products are positive, and hence

$$\frac{B_k}{d_k} \geq 1,$$

which proves part (iv). □

C Proof of Lemma 4.2

We first prove the combinatorial fact used to choose one or two checkpoint indices. We then verify the recurrences and all four conclusions of Lemma 4.2.

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

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

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

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

Proof. Suppose Alternative 1 is false. Then setting $\tau = \frac{S'}{8r^2}$, the array must satisfy the following condition:

$$\text{For any } 1 \leq i < j \leq r, \text{ if } h_i, h_j \geq \tau, \text{ then } \sum_{t=i+1}^{j-1} h_t > \min\{h_i, h_j\}. \quad (46)$$

Claim. Consider a sequence $a_1, \dots, a_k \geq 0$ satisfying (46) with a_i in place of h_i , k in place of r , and the same threshold τ . We use two additional conditions:

- *one-sided:* For all i , if $a_i \geq \tau$, then $a_i < \sum_{v=1}^{i-1} a_v$;
- *two-sided:* For all i , if $a_i \geq \tau$, then $a_i < \min(\sum_{v=1}^{i-1} a_v, \sum_{v=i+1}^k a_v)$.

We claim that any sequence of length k satisfying (46) has a total sum $\sum_{i=1}^k a_i$ bounded by $2\tau k^2$ if it is one-sided, and by τk^2 if it is two-sided.

Proof of Claim. We prove by strong induction on k . The base case $k = 0$ holds trivially since the sum is 0. For $k \geq 1$, if every $a_i < \tau$, the total sum is strictly bounded by $\sum_{i=1}^k a_i < k\tau \leq \tau k^2$, satisfying both cases.

Otherwise, choose a maximum element $b = a_z \geq \tau$. We partition the sequence into a left part $L = (a_1, \dots, a_{z-1})$ and a right part $R = (a_{z+1}, \dots, a_k)$ with lengths k_L and k_R , and let $S_L = \sum_{v=1}^{z-1} a_v$ and $S_R = \sum_{v=z+1}^k a_v$ be their respective sums. Both inherit (46). Because b is the maximum, applying (46) between b and any element $a_i \geq \tau$ imposes strict new bounds directed toward b :

- For $i < z$, the inequality $a_i < \sum_{v=i+1}^{z-1} a_v$ gives L a right-sided bound.
- For $i > z$, the inequality $a_i < \sum_{v=z+1}^{i-1} a_v$ gives R a left-sided bound.

Now we evaluate the sequence (L, b, R) based on whether it is one-sided or two-sided:

- *If two-sided:* L is two-sided by inheriting its left-sided bound. Similarly, R is two-sided by inheriting its right-sided bound. The two-sided property of the full sequence also requires $b < \min(S_L, S_R)$. Using the inductive bounds for L and R :

$$S_L + b + S_R < S_L + \min(S_L, S_R) + S_R \leq \tau k_L^2 + \tau \min(k_L^2, k_R^2) + \tau k_R^2 \leq \tau(k_L + k_R)^2 < \tau k^2.$$

- *If one-sided:* L is two-sided by inheriting its left bound. R is one-sided. The one-sided property of the full sequence requires $b < S_L$. Thus:

$$S_L + b + S_R < 2S_L + S_R \leq 2\tau k_L^2 + 2\tau k_R^2 \leq 2\tau(k_L + k_R)^2 < 2\tau k^2.$$

This completes the induction for the claim.

Applying the Claim. Return to the original array $h_1, \dots, h_r$. If all $h_i < \tau$, then $S' < r\tau = S'/(8r) \leq S'/8$, which contradicts $S' > 0$. Thus, there exists a maximum element $b = h_z \geq \tau$. By (46), for any $i < z$ with $h_i \geq \tau$, we have $h_i < \sum_{t=i+1}^{z-1} h_t$. Thus the reversed left tail $(h_{z-1}, \dots, h_1)$ is one-sided. Similarly, for $i > z$, $h_i < \sum_{t=z+1}^{i-1} h_t$, meaning the right tail $(h_{z+1}, \dots, h_r)$ is also one-sided. Applying the claim to both tails:

$$S' - b = S_L + S_R \leq 2\tau k_L^2 + 2\tau k_R^2 \leq 2\tau(k_L + k_R)^2 \leq 2\tau r^2.$$

Substituting $\tau = \frac{S'}{8r^2}$ yields $S' - b \leq 2\left(\frac{S'}{8r^2}\right)r^2 = \frac{S'}{4}$. Therefore, the maximum element $b \geq \frac{3S'}{4}$, proving Alternative 2. $\square$

We apply Lemma C.1 to the final portion $h_{j_*}, \dots, h_r$ specified in Lemma 4.2. A stepsize at least $S_0/16$ gives a single Huber checkpoint. Otherwise, we obtain two stepsizes $b_1, b_2 \geq S_0/(64r^2)$ whose intervening gap mass satisfies $s_2 \leq \min\{b_1, b_2\}$. In this case, the first checkpoint may leave a large threshold-to-amplitude ratio. The inequality $s_2 \leq b_1$ ensures $\eta_2 > 0$, so the second transfer remains feasible, while $s_2 \leq b_2$ bounds its outgoing ratio ℓ_3/D_3 . The lower bounds on b_1 and b_2 yield the amplitude bound in (22).

Proof of Lemma 4.2. Fix $C_0 \geq 1$ and $R = 32C_0$, and choose

$$\varepsilon_0 = \frac{1}{8(R+1)}, \quad c_0 = \frac{1}{[4096(1+C_0)]^2}.$$

Let the data satisfy the hypotheses of the lemma. Then

$$\frac{S}{\kappa} \leq \frac{C_0 S_0}{\kappa} \leq \varepsilon_0 < \frac{1}{8},$$

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

Step 1. Choosing the checkpoints. Let j_* be the index in (23), which exists because $S \geq S_0$. Set $S' := \sum_{t=j_*}^r h_t$. By maximality of j_* ,

$$S' > \frac{S_0}{8}, \quad \sum_{t=j_*+1}^r h_t \leq \frac{S_0}{8}. \quad (47)$$

We choose the checkpoints from $\{j_*, \dots, r\}$ as follows.

If this list contains a stepsize at least $S_0/16$, choose its index as t_1 and set $q = 1$. Thus $b_1 = h_{t_1} \geq S_0/16$. Otherwise, every stepsize in the list 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 $3S'/4 > 3S_0/32 > S_0/16$. The first alternative therefore supplies indices $j_* \leq t_1 < t_2 \leq r$. Set $q = 2$ and $b_i = h_{t_i}$ for $i = 1, 2$. Since the list has at most r entries,

$$b_1, b_2 \geq \frac{S'}{8r^2} > \frac{S_0}{64r^2}, \quad s_2 = \sum_{t=t_1+1}^{t_2-1} h_t \leq \min\{b_1, b_2\}. \quad (48)$$

In both cases, the selected stepsizes are positive and conclusion (iv) holds. Moreover, $t_q \geq j_*$ and (47) 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 threshold-to-amplitude ratios. For the chosen checkpoints, use s_i and χ_i from (20). The product inequality $\prod_t (1 - a_t) \geq 1 - \sum_t a_t$ for $a_t \in [0, 1]$ gives, for every $1 \leq i \leq q$,

$$\chi_i \geq 1 - \frac{s_i}{\kappa} \geq 1 - \frac{S}{\kappa} \geq \frac{1}{2}, \quad 0 \leq \kappa(1 - \chi_i) \leq s_i.$$

Starting from $(\ell_1, D_1) = (\ell, D)$, define the remaining parameters successively by (21). Whenever $\ell_i \geq 0$, $D_i > 0$, and $\eta_i > 0$, these recurrences give $\delta_{H,i} > 0$, $\ell_{i+1} \geq 0$, $D_{i+1} > 0$, and

$$\begin{aligned} \frac{D_{i+1}}{D_i} &= \frac{(1 - 1/\kappa)b_i\chi_i\eta_i}{2[2 + (\kappa - 1)(1 - \chi_i)]}, \\ \frac{\ell_{i+1}}{D_{i+1}} &= \frac{\kappa(1 - \chi_i)}{b_i\chi_i} \leq \frac{2s_i}{b_i}. \end{aligned} \tag{49}$$

For the first checkpoint, the initial bound $\ell/D \leq R$ implies

$$\eta_1 = 1 - (1 - \chi_1) \left(1 + \frac{\ell}{D}\right) \geq 1 - (R + 1) \frac{S}{\kappa} \geq \frac{7}{8}.$$

Thus $D_2 > 0$, and (49) applies to the first transfer.

If $q = 2$, the first outgoing ratio may be large. The bound $s_2 \leq b_1$ in (48) nevertheless gives

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

Here $s_1 + s_2 \leq S$ because the two gap sums run over disjoint index sets. Hence $\eta_2 \geq 3/4$ and $D_3 > 0$. All the recurrences are therefore well defined, with $\eta_i \geq 1/2$ for every $1 \leq i \leq q$.

Finally, (49) and the checkpoint choices give

$$\begin{aligned} \frac{\ell_2}{D_2} &\leq \frac{2s_1}{b_1} \leq \frac{2C_0S_0}{S_0/16} = R \quad \text{if } q = 1, \\ \frac{\ell_3}{D_3} &\leq \frac{2s_2}{b_2} \leq 2 \leq R \quad \text{if } q = 2, \end{aligned}$$

where we used the bound $s_2 \leq b_2$ in (48). This proves (ii).

Step 3. Bounding the amplitude loss. Since $\kappa \geq 4$, $\chi_i \geq 1/2$, and $\eta_i \geq 1/2$, the first identity in (49) gives

$$\frac{D_{i+1}}{D_i} \geq \frac{3b_i}{64(1 + S)} \geq \frac{b_i}{64(1 + C_0)S_0} \quad (1 \leq i \leq q). \tag{50}$$

We used $2 + (\kappa - 1)(1 - \chi_i) \leq 2 + s_i \leq 2(1 + S)$, followed by $S \leq C_0S_0$ and $S_0 \geq 1$.

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

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

If $q = 2$, then (48) and (50) give

$$\frac{D_3}{D_1} = \frac{D_2}{D_1} \frac{D_3}{D_2} \geq \frac{1}{[4096(1 + C_0)r^2]^2} = c_0r^{-4} \geq c_0(r + 1)^{-4}.$$

Together with the positivity established in Step 2, these bounds prove (i). $\square$