A near-linear-time $(2-c)$-approximation for binary LCS via string regularity

July 13, 2026

Abstract

We give absolute constants $\delta, c > 0$ and a deterministic algorithm which, on arbitrary binary strings s, t of total length $n = |s| + |t|$, runs in $n \operatorname{polylog} n$ time and outputs an explicit common subsequence of length at least $(1/2 + \delta)\operatorname{LCS}(s, t)$. Equivalently, the algorithm is a $(2 - c)$-approximation, with no assumption that the input lengths or symbol counts agree. He and Li previously showed that, for every fixed $\varepsilon > 0$, some gain $\delta_\varepsilon > 0$ over $1/2$ is attainable in deterministic $n^{1+\varepsilon}$ time for the same regime. Thus the present result achieves a single absolute gain, independent of the exponent slack ε, in $n \operatorname{polylog} n$ time while retaining explicit-subsequence output. Combined with the reduction of Akmal and Vassilevska Williams, the result also gives, for every fixed $q \geq 2$, a deterministic near-linear $(1/q + \delta_q)$-approximation with an explicit witness for arbitrary-length q-ary LCS.

The technical core is a promise algorithm for equal-length strings having exactly half zeros and half ones: it always finds $N/2$ symbols and beats $N/2$ by a constant fraction when the two strings have a near-perfect common subsequence. Its structural input is the string-regularity decomposition of Guruswami–He–Li; physical-position grids, robust flag-transfer lemmas, and an exact diagonal-total-variation dynamic program turn the decomposition into an explicit near-linear-time matching. A trimming lemma first extends this core to nearly balanced pairs. We then give a robust version of the Rubinstein–Song equal-length case analysis, replacing its edit-distance call by the trimming lemma, and finally apply the He–Li reduction from equal to arbitrary input lengths.

Note. The balanced proof underlying this paper was fully AI generated by GPT 5.6 Sol, prompted by Yang Liu. The reduction to arbitrary inputs and the present revisions were generated and audited by GPT-5 in the Codex harness, also prompted by Yang Liu.

1 Introduction

Throughout, strings are binary. For a string x, let $\#_1(x)$ and $\#_0(x)$ denote its numbers of ones and zeros. A common subsequence is always represented by a noncrossing matching, so every lower bound constructed below is algorithmically certifiable.

For strings x and y, insertion–deletion edit distance and LCS are linked by the exact identity

$$\operatorname{ED}_{\text{ins/del}}(x, y) = |x| + |y| - 2\operatorname{LCS}(x, y).$$

The standard dynamic program computes either quantity in $O(|x||y|)$ time. For strings of lengths $p \geq q$ over a fixed alphabet, the Four Russians method of Masek and Paterson computes an optimal alignment in $O(p \max\{1, q/\log p\})$ time on their logarithmic-cost RAM model [MP80]. Counting its $O(\log n)$-bit boundary vectors as unit-cost words gives the familiar $O(n^2/\log^2 n)$ equal-length operation count. For general sequences, later tabulation refinements compute the LCS length of strings

of lengths m and n in $O(mn \log\log n / \log^2 n)$ time on a word RAM [Gra16]. These improvements do not change the quadratic exponent. Under the Strong Exponential Time Hypothesis, LCS on constant-size alphabets has no $O(n^{2-\varepsilon})$ algorithm [ABW15], and this remains true already for binary strings [BK15].

Approximation landscape. We use the maximization convention that an α-approximation outputs a common subsequence of length at least $\alpha\,\mathrm{LCS}(x, y)$; the conventional approximation factor is $1/\alpha$. On an alphabet of size q, the longest unary common subsequence gives a $1/q$-approximation in linear time. For two length-n strings over an unrestricted alphabet, Hajiaghayi, Seddighin, Seddighin, and Sun gave a randomized linear-time $O(n^{0.497956})$-factor approximation in expectation [HSSS19]. Bringmann, Cohen-Addad, and Das improved this to a $\widetilde{O}(n^{0.4})$ factor in linear time with high probability [BCAD23], and Nosatzki obtained a randomized $O(n)$-time $n^{o(1)}$-factor approximation with high probability [Nos21]. On the deterministic side, Boneh, Golan, and Kraus gave the first near-linear-time sublinear-factor guarantee, an $O(n^{3/4} \log n)$ factor, again for two length-n sequences [BGK25]. These large-alphabet results address a different regime from the constant gain over $1/2$ sought here for binary strings. Hajiaghayi et al. explicitly return a common subsequence; Bringmann et al. state their main guarantee for the value and note that a witness can be recovered; Nosatzki's result is a value estimate.

Approximate edit distance developed along a more favorable trajectory. The work cited in this paragraph permits substitutions; its metric differs from insertion–deletion distance by at most a factor of two. Randomized algorithms of Andoni and Onak achieved a $2^{\widetilde{O}(\sqrt{\log n})}$ approximation in $n^{1+o(1)}$ time [AO12], and Andoni, Krauthgamer, and Onak obtained a $(\log n)^{O(1/\varepsilon)}$ approximation in $n^{1+\varepsilon}$ time [AKO10]. Chakraborty, Das, Goldenberg, Koucký, and Saks gave the first randomized constant-factor approximation in $\widetilde{O}(n^{12/7})$ time [CDG+20]; Andoni and Nosatzki then achieved a randomized constant-factor approximation in $n^{1+\varepsilon}$ time for every fixed $\varepsilon > 0$ [AN20].

Rubinstein and Song showed that an edit-distance estimate in $[\mathrm{ED}, c\,\mathrm{ED} + o(n)]$ obtained in time $T(n)$ yields a $(1/2 + \delta(c))$-approximation for equal-length binary LCS in $T(n) + O(n)$ time [RS20]. Their main theorem is phrased as a value estimate; when the edit-distance routine returns a transformation, the reduction also returns an explicit common subsequence. Instantiating the reduction with near-linear edit-distance routines broke the binary $1/2$ barrier in randomized $n^{1+\varepsilon}$ time for equal-length inputs. The equal-length hypothesis is essential to that analysis, in which a near-perfect LCS corresponds to small edit distance.

For every fixed alphabet size q, Akmal and Vassilevska Williams obtained an equal-length $(1/q + \delta_q)$-approximation. They also showed that any arbitrary-length binary improvement transfers to arbitrary-length q-ary LCS [AW21]. He and Li then handled unequal-length binary strings: for every fixed $\varepsilon > 0$, they gave a deterministic $n^{1+\varepsilon}$-time $(1/2 + \delta_\varepsilon)$-approximation for some $\delta_\varepsilon > 0$ [HL23]. More quantitatively, their black-box reduction turns an equal-length gain δ and running time $T(n)$ into an arbitrary-length gain δ^A and running time $O((n + T(n)) \log^A n)$ for an absolute A; it also preserves an explicit subsequence when the oracle returns one. This is the result most directly comparable with the present theorem.

At a different time–accuracy tradeoff, Mao and Rubinstein recently obtained, with high probability, randomized $(1-\varepsilon)$-approximations for LCS on strings of total length n in $n^2/2^{(\log n)^{\Omega(1)}}$ time [MR26]. They also observe that analogous deterministic time–accuracy tradeoffs would imply major new circuit lower bounds. The present paper instead keeps deterministic near-linear time and seeks only an absolute improvement over the unary $1/2$ baseline.

Present contribution. The result below strengthens this runtime–accuracy tradeoff: it obtains a single absolute gain over $1/2$ in $n \operatorname{polylog} n$ time for arbitrary-length binary strings, while preserving explicit-subsequence output. Its new ingredient is the following balanced promise theorem. The structural source is the three-type string-regularity theorem of Guruswami, He, and Li, originally developed to prove limitations for binary deletion codes [GHL22]. Here it is applied to a hidden near-perfect common subsequence and converted into an algorithm by enumerable physical occurrences, robust transfer lemmas, and exact stitching.

Theorem 1.1 (Balanced promise theorem)**.** *There are absolute constants $\eta, \delta > 0$ and a deterministic $N \operatorname{polylog} N$-time algorithm with the following property. If $s, t \in \{0,1\}^N$ satisfy*

$$\#_1(s) = \#_0(s) = \#_1(t) = \#_0(t) = N/2,$$

then the algorithm outputs a common subsequence of s, t of length at least $N/2$. If additionally

$$\mathrm{LCS}(s,t) \geq (1-\eta)N,$$

then the output has length at least $(1/2 + \delta)N$.

The promise theorem is the main technical result. The reductions in Section 9 remove both restrictions on its input.

Theorem 1.2 (Arbitrary binary LCS)**.** *There are absolute constants $\delta_{\mathrm{all}}, c > 0$ and a deterministic algorithm which, given arbitrary strings $s, t \in \{0,1\}^*$ and $n = |s| + |t|$, runs in $n \operatorname{polylog} n$ time and outputs a common subsequence of length at least*

$$\left(\frac{1}{2} + \delta_{\mathrm{all}}\right) \mathrm{LCS}(s,t).$$

Consequently its conventional approximation factor is at most $2 - c$.

Corollary 1.3 (Constant alphabets)**.** *For every fixed integer $q \geq 2$, there is a constant $\delta_q > 0$ and a deterministic algorithm which, on arbitrary q-ary strings x, y of total length n, runs in $n \operatorname{polylog}(n+2)$ time and outputs an explicit common subsequence of length at least $(1/q + \delta_q) \mathrm{LCS}(x,y)$.*

Proof. Apply the arbitrary-length alphabet reduction of Akmal and Vassilevska Williams [AW21] to Theorem 1.2. For fixed q, their construction makes $O(q^2)$ calls to the binary algorithm and otherwise takes linear time; it preserves determinism and lifts the returned binary witnesses to a common subsequence of the original inputs. Their counting argument gives, for example, $\delta_q = 2\delta_{\mathrm{all}}/q$. The stated running time follows. □

The binary result is constructive throughout: every local certificate is stored as a noncrossing matching, trimming retains the original positions, and the two external reductions used in its proof preserve explicit-subsequence output.

Proof overview. The balanced promise theorem applies the three-type decomposition of Guruswami, He, and Li to a hidden near-perfect common subsequence. We enumerate rounded physical occurrences of its regular blocks, transfer their flags robustly to the observed inputs, and extract local matching gains in the Imbalanced, Green, and Blue–Yellow cases. An exact dynamic program stitches these gains while charging only the total variation of their matching diagonals.

To pass to general inputs, we first trim a nearly balanced equal-length pair to two exactly balanced strings. Deletions in both strings are charged separately. Next we revisit all cases in the

Rubinstein–Song reduction. Their approximate-edit-distance routine occurs only on nearly balanced equal-length pairs, where the trimming algorithm supplies the required gain; all cross-unbalanced cases use unary or greedy matchings. The resulting equal-length black box is then inserted into the He–Li reduction for possibly unequal lengths.

Organization. Section 2 records the regularity input. The hidden common core, flag transfer, local matchings, and stitching are developed in Sections 3 to 6. Sections 7 and 8 prove the balanced promise theorem. Section 9 gives the nearly balanced, equal-length, and unequal-length reductions and proves Theorem 1.2.

Limitations. The result is asymptotic. The fixed regularity parameters force an enormous polylogarithmic exponent and an enormous absolute threshold below which the algorithm falls back to exact dynamic programming. Neither the approximation gain nor these quantitative constants are intended to be practical, and the paper does not improve the near-quadratic regime for near-exact approximation.

The constants are extremely small. We make no attempt to optimize them. We fix the constants used by Guruswami–He–Li,

$$\varepsilon = 10^{-6}, \qquad \gamma = 10^{-3}\varepsilon^2 = 10^{-15}, \tag{1}$$

and set

$$\rho = \varepsilon/100. \tag{2}$$

After all absolute constants in the proof are fixed, choose

$$0 < \eta \ll \sigma \ll \kappa \ll \varepsilon^6. \tag{3}$$

For example, $\kappa = \varepsilon^{20}$, $\sigma = \varepsilon^{30}$, and a sufficiently small constant multiple of ε^{50} for η are more than adequate. Only the hierarchy in (3) is used.

2 One-rank notation and the imported regularity theorem

2.1 One-rank intervals

Let x be any string containing M ones. For $a \in [M]$ and an integer $b > a$, write $x[a, b)$ for the substring beginning at the ath one and ending immediately before the bth one, or at the end of x when $b > M$. Thus $x[a, a + r)$ contains exactly r ones when $a + r \le M + 1$ and is truncated otherwise. We write

$$z_x(a, r) := \#_0(x[a, a + r)).$$

For $a \in [M + 1]$, define $x[a, a)$ to be the empty string. This is the half-open version of the notation in Guruswami–He–Li.

For a positive integer r, the start a is an r-flag of rate $z_x(a, r)/(r - 1)$, with the usual convention at $r = 1$. It is *strong Blue*, *strong Green*, *strong Yellow*, or *strong Red* according as its rate is respectively

$$> \varepsilon^{-1}, \qquad > 1 + 2\varepsilon, \qquad > 0.9, \qquad \le 0.9.$$

For $a \in [M]$, let $b_x(a)$ be the largest power of two r for which a is a strong Blue r-flag, subject to $r \le M$, and set $b_x(a) = 0$ if none exists.

If x has B ones and $\Delta \in [-B, B]$, define

$$\mathrm{Trim}_\Delta(x) := x\big[\max\{1, 1-\Delta\}, \min\{B+1, B+1-\Delta\}\big). \tag{4}$$

It contains exactly $B - |\Delta|$ ones: it is a prefix when $\Delta > 0$ and a suffix when $\Delta < 0$. A start j is *allowable at scale* r in a B-one string if $1 \le j \le B - r + 1$, equivalently if its full r-window is contained in the string.

When x has leading zeros, they are therefore omitted from every one-rank interval. The reversal $\mathrm{rev}(x)$ is the reversal convention of Guruswami–He–Li: keep the initial one fixed and reverse all subsequent symbols. It is defined only for strings beginning with one. It preserves LCS between two such strings and reverses one-rank intervals exactly. We never apply this operation to an arbitrary full input string.

2.2 The three types

Let x contain $B = 2^m$ ones. We use the following definitions.

(i) x is *Imbalanced* if some one-rank interval I of size $|I| > \varepsilon^5 B$ satisfies

$$\#_0(x_I) \notin (1 \pm \varepsilon)|I|.$$

(ii) x is r-*Green* if $r \le \varepsilon^5 B$ and it has at least $\varepsilon^2 B$ strong Green r-flags.

(iii) For an integer $m_0 \in [1, m]$, x is m_0-*Blue–Yellow* if at least $(\varepsilon^2 - \gamma)B$ indices a satisfy

$$2^{m_0} \le b_x(a) \le \gamma B,$$

and, for every integer r with $2^{m_0} < r \le B$, at most $600\varepsilon B$ starts are strong Red r-flags.

When several types are available, we assign one with priority

$$\text{Imbalanced} > \text{Green} > \text{Blue–Yellow}.$$

For a B-one string x, call it 6γ-*flag-balanced* if the ℓ_1 distance between the distributions of $b_x(a)$ in its first and second one-rank halves is at most 6γ. In particular, if x is both flag-balanced and m_0-Blue–Yellow, then its first half contains at least

$$\frac{1}{2}(\varepsilon^2 - 4\gamma)B > 0.49\varepsilon^2 B \tag{5}$$

starts a with $2^{m_0} \le b_x(a) \le \gamma B$.

2.3 Imported theorem

We use the following consequence of Lemmas 3.5, 4.4, and 5.6 of Guruswami–He–Li [GHL22].

Proposition 2.1 (GHL regularity package)**.** *Let w begin with one and contain $L = 2^n$ ones, for sufficiently large n. Put*

$$m_0^{\mathrm{big}} = \left\lfloor n - 200\gamma^{-3}\log_2 n \right\rfloor, \qquad B_0 = 2^{m_0^{\mathrm{big}}}.$$

At least one of the following alternatives holds.

(a) In w or $\mathrm{rev}(w)$, more than $\gamma^2 L$ indices satisfy $b(a) \geq B_0$. Then that orientation contains pairwise disjoint zero-rich Imbalanced one-rank intervals whose total one-rank size is at least $\gamma^2 L/2$ and each of whose sizes is at least B_0.

(b) For every fixed constant R, there is, for all sufficiently large n, an integer

$$m_* \in [n - 150\gamma^{-3} \log_2 n,\ n - R]$$

such that, with $B = 2^{m_*}$ and $K = L/B$, all but $3\gamma K$ of the consecutive B-one blocks of w are 6γ-flag-balanced, and all but $3\gamma K$ of their reversals are 6γ-flag-balanced. Every block and every reversed block has one of the three types above.

Moreover $K \leq (\log L)^{A_0}$ for an absolute constant A_0 depending only on γ.

Proof. If more than $\gamma^2 L$ indices have $b(a) \geq B_0$, greedily choose the leftmost such index not covered by a previously selected interval and select the interval $[a, \min\{a + b(a) - 1, L\}]$, truncated at the end of the string. These intervals are disjoint and cover every exceptional index, so their total one-rank size is at least $\gamma^2 L$. Each is zero-rich Imbalanced by GHL Lemma 3.5. Only the final selected interval can be truncated to size below B_0; it then contains fewer than B_0 one-ranks. For sufficiently large n, $B_0 < \gamma^2 L/2$. Discarding this possible final interval leaves total size at least $\gamma^2 L/2$, proving (a).

Otherwise at most $\gamma^2 L$ indices satisfy $b(a) \geq B_0$. Since $B_0 \leq 2^{n - 200\gamma^{-3} \log_2 n}$, the threshold hypothesis of GHL Lemma 5.6 holds. That lemma, with $\beta = \gamma$, excludes at most $32\gamma^{-3} \log_2 n$ scales for w and the same number for $\mathrm{rev}(w)$ inside an interval of $150\gamma^{-3} \log_2 n$ top scales. After excluding the fixed top R scales, at least one common scale remains for sufficiently large n. Lemma 5.6 gives the asserted 6γ-balance at that scale. GHL Lemma 4.4 supplies a type for every power-of-two-one block. Finally, $n - m_* = O(\log n)$ implies $K = 2^{n - m_*} = (\log L)^{O(1)}$. □

3 The hidden common core and physical-grid candidates

Fix balanced strings $s, t \in \{0,1\}^N$ and a common subsequence matching q of length $N - D$. Each input has exactly D unmatched symbols. Since s has $N/2$ ones and zeros,

$$\#_1(q), \#_0(q) \geq N/2 - D. \tag{6}$$

Let $L = 2^n$ be the largest power of two not exceeding $\#_1(q)$, and let $q' = q[1, L+1)$ (ending with q if q has exactly L ones). Then

$$L > \frac{\#_1(q)}{2} \geq \frac{N}{4} - \frac{D}{2}. \tag{7}$$

At a candidate block scale B, partition q' into consecutive B-one blocks $Q_1, \ldots, Q_K$. Let S_i and T_i be the minimal physical spans of the images of Q_i in s and t. Put

$$e_i^s = |S_i| - |Q_i|, \qquad e_i^t = |T_i| - |Q_i|, \qquad e_i = e_i^s + e_i^t.$$

Lemma 3.1 (Defect accounting)**.** *The spans S_i are ordered and pairwise disjoint, as are the T_i, and*

$$\sum_i e_i \leq 2D.$$

Proof. The blocks Q_i are consecutive pieces of one fixed noncrossing matching. No symbol matched to a different Q_j lies inside the minimal span of Q_i. Hence every symbol of S_i outside the image of Q_i is unmatched in the fixed matching, and the same is true in t. Disjointness charges every unmatched symbol at most once. □

Fix $h = \lceil \sigma B \rceil$ and let the physical grid consist of $0, h, 2h, \ldots, N$, together with N if necessary. Round a physical interval outward to the nearest grid points.

Lemma 3.2 (Physical-grid lemma)**.** *Let $\widehat{S}_i, \widehat{T}_i$ be the outward rounded spans. Then they contain the two images of Q_i and*

$$(|\widehat{S}_i| - |Q_i|) + (|\widehat{T}_i| - |Q_i|) \le e_i + 4h.$$

One of the two index parities contains at least half of any prescribed set of indices and, on either fixed parity, the rounded spans are pairwise disjoint in each input, provided $2h < B$.

Proof. Outward rounding adds fewer than $2h$ physical symbols per string. Between S_i and S_{i+2} lies the entire span S_{i+1}, which contains B matched ones and therefore has physical length at least B. Rounding can reduce this gap by less than $2h$, so it remains positive. The same argument applies in t. □

The algorithm enumerates every physical grid interval. At the successful regular scale, the number of grid points is

$$G = O(N/h) = O(N/B) = \operatorname{polylog} N, \tag{8}$$

so there are only $O(G^2)$ intervals per input and $O(G^4)$ interval pairs. We shall repeatedly use

$$Kh \le \sigma L + K = O(\sigma L). \tag{9}$$

The last equality holds once $B \ge 2/\sigma$; the finitely many smaller instances are included in the exact-algorithm fallback below.

If a physical interval X contains at least B ones, let $F_B(X)$ be its substring beginning at its first one and ending immediately before its $(B+1)$st one, or at the end of X if there is no such one. Thus $F_B(X)$ has exactly B ones. Define $L_B(X)$ analogously from the last B ones, and use $\operatorname{rev}(L_B(X))$ for reverse-oriented certificates.

4 Strong-to-weak flag transfer

Define, for every integer $r \ge 1$,

$$H(r) = r + \lfloor \rho(r-1) \rfloor. \tag{10}$$

The floor is essential: it gives $H(2) = 2$ and, more generally,

$$H(r) \le (1+\rho)r. \tag{11}$$

4.1 A maximal inequality

Lemma 4.1 (One-sided maximal inequality)**.** *Let $f_1, \ldots, f_M \ge 0$, with $\sum_j f_j = E$. For $\lambda > 0$, the set of indices i for which some $r \ge 1$ satisfies*

$$\sum_{j=i}^{i+r-1} f_j > \lambda r$$

has size at most $3E/\lambda$.

Proof. For every bad i, choose an interval $I_i = [i, i + r_i)$ witnessing the inequality. Order these intervals by nonincreasing length and greedily retain an interval when it is disjoint from those already retained. This gives a disjoint subfamily $\mathcal{J}$. Every discarded interval intersects a retained interval of at least its length, so its left endpoint lies in the threefold enlargement of that retained interval. Therefore the number of bad integer left endpoints is at most $3\sum_{J\in\mathcal{J}} |J|$. Since the chosen intervals are disjoint,

$$\lambda \sum_{J\in\mathcal{J}} |J| < \sum_{J\in\mathcal{J}} \sum_{j\in J} f_j \le E.$$

This proves the claim. □

When a core string Q is embedded in a supersequence X, put into f_j the number of extra one symbols in the gap following the jth core one. If the r-one core interval starting at i contains at most $\lfloor \rho(r-1) \rfloor$ extra ones, the $H(r)$-one interval of X starting at the image of i contains the entire core interval. By Lemma 4.1, simultaneously over variable $r \ge 2$, all but $O(E/\rho)$ starts have this property; explicitly, the displayed maximal inequality gives at most $6E/\rho$ after using $r - 1 \ge r/2$.

4.2 Canonical-window transfer

The next lemma records precisely what survives after taking the first B ones of a rounded physical span. It does *not* claim that this canonical window is close in ordinary edit distance to the whole core block.

Lemma 4.2 (Canonical flag stability)**.** *Let Q begin with one and contain exactly B ones. Let X be a physical string containing Q as a subsequence and at most a additional ones, and put $U = F_B(X)$. Assume $a \le \kappa B$.*

(a) If Q has at least $\varepsilon^2 B$ strong Green r-flags, where $r \le \varepsilon^5 B$, then there is a scale r' depending only on r such that U has at least $0.8\varepsilon^2 B$ weak Green r'-flags of rate greater than $1 + 1.5\varepsilon$. The same r' works for every occurrence X satisfying the hypotheses.

(b) Suppose Q is 6γ-flag-balanced and m_0-Blue–Yellow. Then U has at least $0.45\varepsilon^2 B$ starts in its first one-rank half, each equipped with a power $p \in [2^{m_0}, \gamma B]$, such that

$$z_U(i, H(p)) \ge 0.45p/\varepsilon. \tag{12}$$

Moreover, for every power $p \in [2^{m_0}, \gamma B]$, at most $700\varepsilon B$ allowable starts j fail

$$z_U(j, Q(p)) \ge 0.46p/\varepsilon, \qquad Q(p) := H(\lceil 0.55p/\varepsilon \rceil)\,. \tag{13}$$

There is also a right-to-left version: if $\mathrm{rev}(Q)$ has either property, then the corresponding conclusion holds, in right-to-left one-ranks, in $\mathrm{rev}(L_B(X))$.

Proof. Write $c_1, \dots, c_B$ for the one symbols of the displayed copy of Q in X, and let $\phi(k)$ be the one-rank of c_k in X. Since at most a ones of X are outside this copy,

$$k \le \phi(k) \le k + a. \tag{14}$$

In particular $c_1, \dots, c_{B-a}$ occur in U, and at most a mapped starts cross any fixed one-rank boundary.

For $1 \le j < B$, let f_j count the additional ones of X strictly between c_j and c_{j+1}, and let f_B count the additional ones after c_B. Consider first a start $k \le B - r + 1$, so that its full core window is defined. If

$$\sum_{j=k}^{k+r-1} f_j \le \lfloor \rho(r-1) \rfloor, \tag{15}$$

then the $H(r)$-one window of X starting at rank $\phi(k)$ contains the entire r-one core window starting at c_k, including its terminal one-gap. For $r \ge 2$, failure of (15) implies $\sum_{j=k}^{k+r-1} f_j > (\rho/2)r$. Hence Lemma 4.1 shows that, even when r varies with k among full core windows, at most $6a/\rho$ starts fail this containment. We use the round upper bound $8a/\rho$ below.

For (a), first discard the at most $r-1$ core starts without a full r-window and assume $r \ge 2$. Outside a further set of at most $8a/\rho$ starts, the image $H(r)$-window contains the full strong core window. It therefore contains more than $(1+2\varepsilon)(r-1)$ zeros. Put

$$d = H(r) - 1$$

and round d upward to a member g of the integer geometric grid from Lemma 5.2 below. That grid satisfies

$$d \le g \le (1 + 2\varepsilon/100)d.$$

Extending the window cannot remove zeros, and

$$\frac{(1+2\varepsilon)(r-1)}{g} \ge \frac{1+2\varepsilon}{(1+\rho)(1+2\varepsilon/100)} > 1 + 1.5\varepsilon.$$

Thus $r' = g+1$ works. The map $k \mapsto \phi(k)$ is injective. Including the initially discarded core-boundary starts, at most $a + r'$ starts are lost at the terminal boundary of $F_B(X)$. Since $r' = O(\varepsilon^5 B) + O_\varepsilon(1)$ and $a \le \kappa B \ll \rho\varepsilon^2 B$, fewer than $0.2\varepsilon^2 B$ starts are lost. If $r = 1$, a strong Green flag means that the one-gap contains a zero. Each inserted one destroys this property for at most one core one-gap, and the same count follows directly.

For (b), (5) gives more than $0.49\varepsilon^2 B$ strong Blue labeled starts in the core first half. Apply Lemma 4.1 to the variable powers p. Outside at most $8a/\rho$ starts, the candidate $H(p)$-window contains the strong core p-window. Since $p \ge 2$,

$$\varepsilon^{-1}(p-1) \ge p/(2\varepsilon) > 0.45p/\varepsilon.$$

These core windows are full because their starts are in the first half and $p \le \gamma B < B/2$. At most a starts move across the first-half boundary, and the maximum window length is at most $(1+\rho)\gamma B$. The hierarchy of constants therefore leaves at least $0.45\varepsilon^2 B$ starts satisfying (12).

Fix a power p and put

$$q_0 = \lceil 0.55p/\varepsilon \rceil .$$

Discard the last $q_0 - 1$ core starts, which have no full q_0-window. The Blue–Yellow type says that at most $600\varepsilon B$ of the remaining core starts are strong Red q_0-flags, because $q_0 > 2^{m_0}$. Every other core start has more than $0.9(q_0 - 1)$ zeros in its q_0-window. Outside $O(a/\rho)$ insertion-dense starts, the candidate $H(q_0) = Q(p)$ window contains it. As $p \ge 2$ and $\varepsilon \le 10^{-6}$,

$$0.9(q_0 - 1) \ge 0.495p/\varepsilon - 0.9 > 0.46p/\varepsilon.$$

For this fixed scale, double counting shows that at most $2a/\rho$ core starts are insertion-dense. The candidate starts not arising from retained core starts, the shifted terminal starts, and the discarded

q_0-boundary starts account for at most $2a + Q(p)$ further exceptions (the last group is covered because $Q(p) \geq q_0$). Finally

$$Q(p) \leq 0.551p/\varepsilon \leq 0.000551\varepsilon B.$$

Thus the total number of failures is at most

$$600\varepsilon B + 2a/\rho + 2a + Q(p) < 700\varepsilon B$$

under the chosen constant hierarchy.

For completeness, the right-to-left assertion is not an appeal to literal subsequence containment under the modified reversal. A B-one string y has a unique representation

$$y = 1\,0^{g_1}1\,0^{g_2}\cdots 1\,0^{g_B}, \qquad \mathrm{rev}(y) = 1\,0^{g_B}1\,0^{g_{B-1}}\cdots 1\,0^{g_1}.$$

Consequently, whenever the displayed indices lie in $[B]$,

$$z_{\mathrm{rev}(y)}(i,r) = \sum_{j=B-i-r+2}^{B-i+1} g_j.$$

Put $W = L_B(X)$ and enumerate the ones of Q and W from the right. If $\psi(k)$ is the right-rank in W of the kth core one from the right, then $k \leq \psi(k) \leq k + a$. For every rank $k \geq 2$, the kth one of $\mathrm{rev}(Q)$ corresponds to the $(k-1)$st core one from the right, and its image in $\mathrm{rev}(W)$ has rank $1 + \psi(k-1) \in [k, k+a]$. The intervening gap sequence is exactly the original gap sequence read right-to-left, with only the additional ones of X inserted. Thus (15), the maximal inequality, and the boundary counts apply verbatim. The exceptional rank $k = 1$ is discarded. This one extra loss is absorbed by the $a + r'$ and $2a + Q(p)$ bounds for sufficiently large B. □

5 Local matching modules

5.1 Imbalanced intervals

Lemma 5.1 (Robust Imbalanced certificate)**.** *Suppose a common core interval has r ones and z zeros, with $r \geq \varepsilon^5 B$, and its two physical occurrence spans acquire at most k additional symbols in total.*

(a) If $z \geq (1+\varepsilon)r$, matching zeros gives, relative to the all-one baseline, adjusted reward at least $\varepsilon r - k/2$.

(b) If $z \leq (1-\varepsilon)r$, matching ones gives, relative to the all-zero baseline, adjusted reward at least $\varepsilon r - k/2$.

In particular the applicable reward is at least $\varepsilon r/2$ whenever $k \leq \varepsilon r$.

Proof. In case (a), both occurrences contain all z common zeros, while their total numbers of ones are at most $r + k_s$ and $r + k_t$, where $k_s + k_t = k$. Hence

$$z - \frac{(r + k_s) + (r + k_t)}{2} \geq \varepsilon r - k/2.$$

Case (b) is identical after exchanging the symbols: both occurrences contain the r core ones, and their total zero counts are at most $2z + k$. □

The algorithm finds these certificates because it enumerates all physical grid intervals and stores both unary matchings for every interval pair. Rounding the two images of a witness outward adds at most $4h$ symbols.

5.2 Green matching

Lemma 5.2 (Integer geometric scales)**.** *There is a set $\mathcal{G}_B$ of $O(\varepsilon^{-1}\log B)$ nonnegative integers such that, for every integer $d \in [0, B]$, its least upper neighbor $g \in \mathcal{G}_B$ satisfies*

$$d \le g \le (1 + 2\varepsilon/100)d$$

when $d > 0$, while $0 \in \mathcal{G}_B$.

Proof. Include every integer up to $100/\varepsilon$. Above that threshold, include the ceilings of successive powers of $1 + \varepsilon/100$, stopping only after including a value at least B. For $d > 100/\varepsilon$, the additive ceiling error is at most $(\varepsilon/100)d$, proving the bound. □

Call a start a weak canonical r-Green flag if it has at least one zero when $r = 1$, and if its rate is greater than $1 + 1.5\varepsilon$ when $r \ge 2$.

Lemma 5.3 (Weak Green matching)**.** *Let U, V each contain exactly B ones. Suppose that for one common scale r they each have at least αB weak canonical r-Green starts whose r-windows are contained in the string, where $\alpha = 0.8\varepsilon^2$. For $\Delta \in [-B, B]$, let $M_r(\Delta)$ be the number of starts that are weak Green in U and whose Δ-shift is weak Green in V, with both full r-windows contained in the corresponding trimmed strings. Then*

$$\mathrm{LCS}(\mathrm{Trim}_{\Delta} U, \mathrm{Trim}_{-\Delta} V) \ge B - |\Delta| + \frac{\varepsilon}{2} M_r(\Delta), \tag{16}$$

and consequently

$$\mathbb{E}_{\Delta}\big[\mathrm{LCS}(\mathrm{Trim}_{\Delta} U, \mathrm{Trim}_{-\Delta} V) + |\Delta|\big] \ge B + 0.05\varepsilon^5 B. \tag{17}$$

Here and below the expectation is uniform over the displayed integer range of Δ.

Proof. Equivalently, $M_r(\Delta)$ counts pairs (i, j) of allowable weak starts with $j - i = \Delta$. The two full windows are then in the trims: if $\Delta \ge 0$, $j \ge 1 + \Delta$ and $i + r - 1 \le B - \Delta$, and the case $\Delta < 0$ is symmetric. The correlation identity gives

$$\sum_{\Delta=-B}^{B} M_r(\Delta) = |G_U||G_V|,$$

so $\mathbb{E}M_r(\Delta) \ge \alpha^2 B/3$. For a fixed Δ, greedily retain r-separated paired starts. At least $M_r(\Delta)/r$ remain. If $r \ge 2$, replacing the following $r-1$ matched ones by zeros gains at least $1.5\varepsilon(r-1) \ge 0.75\varepsilon r$ per retained pair. If $r = 1$, the common zero in the one-gap is inserted without deleting a one match and gives a gain of one. In either case the gain is at least $(\varepsilon/2)M_r(\Delta)$, proving (16). Averaging and using $\alpha^2\varepsilon/6 > 0.1\varepsilon^5$ proves the more conservative (17). □

All $M_r(\Delta)$ for all canonical scales can be computed deterministically in $O(\varepsilon^{-1}B\log^2 B)$ time by binary cross-correlations. After a scale and shift are selected, the greedy matching in the proof is reconstructed in $O(B)$ time.

5.3 A slackened Blue–Yellow lemma

Lemma 5.4 (Directional weak Blue–Yellow matching)**.** *Let U, V each have exactly B ones, with B sufficiently large in terms of ε. Fix an integer $m_0 \in [1, \lfloor \log_2 B \rfloor]$, and assume:*

(i) at least $0.45\varepsilon^2 B$ *starts* $i \le B/2$ *are each assigned a power* $p(i) \in [2^{m_0}, \gamma B]$ *satisfying*

$$z_U(i, H(p(i))) \ge 0.45p(i)/\varepsilon;$$

(ii) for every power $p \in [2^{m_0}, \gamma B]$*, at most* $700\varepsilon B$ *allowable starts* j *fail*

$$z_V(j, Q(p)) \ge 0.46p/\varepsilon, \qquad Q(p) = H(\lceil 0.55p/\varepsilon \rceil).$$

Put

$$S = \lfloor (1/2 + 0.01\varepsilon)B \rfloor, \qquad T = \lfloor (1/2 + 0.27\varepsilon)B \rfloor.$$

For every integer $\Delta \in [0, \lfloor B/4 \rfloor]$*, one can explicitly construct a common subsequence of* $U[1, S+1)$ *and* $V[1+\Delta, 1+\Delta+T)$*, of weight* $W(\Delta)$*, such that*

$$\mathbb{E}_\Delta \left[W(\Delta) - \frac{S+T}{2} \right] \ge 0.04\varepsilon B. \tag{18}$$

Proof. Greedily choose the leftmost remaining qualifying source start i_k, set $p_k = p(i_k)$, and delete all qualifying starts in $[i_k, i_k + H(p_k))$. Continue until

$$P := \sum_k p_k \ge 0.44\varepsilon^2 B.$$

This stopping point exists: the disjoint chosen intervals cover all $0.45\varepsilon^2 B$ qualifying starts, while $H(p) \le (1+\rho)p$. Since the final power is at most $\gamma B = 0.001\varepsilon^2 B$,

$$0.44\varepsilon^2 B \le P \le 0.441\varepsilon^2 B. \tag{19}$$

Every selected source interval lies before S, because its start is at most $B/2$ and $H(p) \le (1+\rho)\gamma B \ll 0.01\varepsilon B$.

Write $h_k = H(p_k)$ and $q_k = Q(p_k)$. For $p \ge 2$,

$$H(p) \le (1+\rho)p, \qquad 0.55p/\varepsilon \le Q(p) \le 0.551p/\varepsilon. \tag{20}$$

The upper bound uses $p \ge 2$ and $\varepsilon \le 10^{-6}$. Therefore

$$\sum_k q_k \le 0.551P/\varepsilon \le 0.242991\varepsilon B. \tag{21}$$

For a shift Δ, define

$$j_k = i_k + \Delta + \sum_{r<k} (q_r - h_r). \tag{22}$$

The source intervals are disjoint, and hence

$$j_{k+1} - (j_k + q_k) = i_{k+1} - (i_k + h_k) \ge 0.$$

The final target endpoint, including the final one-gap, is at most

$$1 + \Delta + S + \sum_k (q_k - h_k) \le 1 + \Delta + (1/2 + 0.252991\varepsilon)B < 1 + \Delta + T.$$

Thus every target interval is allowable.

Call k good if condition (ii) holds at j_k. As Δ ranges over at $\lfloor B/4 \rfloor + 1 \geq B/4$ consecutive integers, j_k does likewise, so

$$\Pr[k \text{ bad}] \leq 2801\varepsilon \tag{23}$$

for sufficiently large B. Put $b_k = \lfloor 0.44 p_k/\varepsilon \rfloor$. Both source and a good target interval contain at least b_k zeros.

Partition the source window into the gap before the first selected interval, the selected intervals, the gaps between them, and the final gap ending at one-rank S. In the target, use the gap beginning at one-rank $1+\Delta$, the intervals beginning at the j_k, the corresponding intermediate gaps, and the final gap ending at $1 + \Delta + S + \sum_k (q_k - h_k)$. Equation (22) makes every pair of corresponding gaps have equal one counts. All target pieces lie in the declared T-one window by the preceding endpoint bound. Match all gap ones; on a bad flag pair match all h_k source ones to target ones; and on a good pair replace those matches by b_k zero matches. This is noncrossing and has weight

$$W(\Delta) = S + \sum_{k \text{ good}} (b_k - h_k). \tag{24}$$

Every summand is positive. If d is the number of selected flags, then $d \leq P/2$, because every $p_k \geq 2$. Consequently

$$\sum_k b_k \geq 0.44P/\varepsilon - P/2, \qquad \sum_k h_k \leq (1+\rho)P.$$

Using (23) and (19),

$$\begin{aligned} \mathbb{E}W(\Delta) &\geq S + (1 - 2801\varepsilon)\left(\frac{0.44}{\varepsilon} - 1.5 - \rho\right) P \\ &\geq (1/2 + 0.19\varepsilon)B \end{aligned}$$

for sufficiently large B. Since $(S+T)/2 \leq (1/2 + 0.14\varepsilon)B$, the conservative bound (18) follows. □

Lemma 5.5 (Directional cancellation)**.** *Let U_1, U_2, V_1, V_2 each have B ones. Suppose Lemma 5.4 applies to (U_1, V_1) and to $(\mathrm{rev}(V_2), \mathrm{rev}(U_2))$. With the same random Δ in both directions,*

$$\mathbb{E}_\Delta \left[\mathrm{LCS}(\mathrm{Trim}_\Delta(U_1U_2), \mathrm{Trim}_{-\Delta}(V_1V_2)) + \Delta\right] \geq 2B + 0.08\varepsilon B.$$

Proof. The forward matching uses the first S ones of U_1 and a T-one interval of V_1 beginning after Δ ones. The reversed matching maps to a T-one interval of U_2 ending Δ ones before its end and the last S ones of V_2. The middle regions in the two concatenations each contain exactly $2B - S - T - \Delta$ ones. Match those ones and apply (18) twice. Linearity of expectation is sufficient; no independence is used. □

For fixed selected source flags, the bad-weight function of Δ is a sum, over the $O(\log B)$ powers p, of cross-correlations between a sparse weighted source array and the target bad-start indicator at scale $Q(p)$. Thus all values $W(\Delta)$ are computed deterministically in $B \operatorname{polylog} B$ time.

6 Exact stitching by diagonal total variation

Fix a baseline symbol $b \in \{0, 1\}$. A local certificate e consists of ordered source and target physical intervals, an explicit common subsequence of weight $w(e)$, and baseline counts $q_s(e), q_t(e)$ in those intervals. Let $R_b^x(u)$ denote the number of copies of b strictly before physical position u in x. Define

$$z^-(e) = R_b^t(\ell_t(e)) - R_b^s(\ell_s(e)), \qquad z^+(e) = z^-(e) + q_t(e) - q_s(e),$$

and

$$r(e) = w(e) - \frac{q_s(e) + q_t(e)}{2}.$$

Lemma 6.1 (Exact chain identity)**.** *Let M_s, M_t be the total baseline counts in s, t. For any ordered chain $e_1 \prec \cdots \prec e_k$, concatenating the local certificates and matching the baseline symbol greedily in every gap produces a common subsequence of length at least*

$$\frac{M_s + M_t}{2} + \sum_{i=1}^{k} r(e_i) - \frac{1}{2}\Big(|z^-(e_1)| + \sum_{i=1}^{k-1} |z^-(e_{i+1}) - z^+(e_i)| \tag{25}$$

$$+ |(M_t - M_s) - z^+(e_k)|\Big). \tag{26}$$

Proof. If a source gap contains x baseline symbols and its target gap contains y, greedy matching contributes $\min(x, y) = (x + y - |x - y|)/2$. Summing the $x + y$ terms over all gaps gives $M_s + M_t - \sum_i (q_s(e_i) + q_t(e_i))$. The gap differences are exactly the three kinds of diagonal differences in (25). Adding the local weights proves the identity. □

Consequently the optimal chain over a finite certificate set is found by

$$F(e) = r(e) + \max\left\{-\frac{|z^-(e)|}{2}, \max_{f \prec e}\left(F(f) - \frac{|z^-(e) - z^+(f)|}{2}\right)\right\}. \tag{27}$$

After computing these values, the length certified by the best chain is

$$\frac{M_s + M_t}{2} + \max\left\{0, \max_{e}\left(F(e) - \frac{|(M_t - M_s) - z^+(e)|}{2}\right)\right\}. \tag{28}$$

Predecessor pointers recover the chain. The zero in (28) represents the empty chain. There are only polylog N certificates at a fixed scale and shift, so a quadratic implementation is sufficient.

Lemma 6.2 (Hidden offset variation)**.** *For a chain arising from the fixed common subsequence q, the extended total variation of its unshifted base diagonals is*

$$O(D + Kh) = O(D + \sigma N).$$

This assertion applies in either of the following two situations:

(i) the certificates use full outward-rounded physical witness spans and the baseline is either symbol;

(ii) the certificates lie in first-B or last-B canonical windows and the baseline is one.

Proof. At a cut of the fixed matching, the difference between the target and source b-ranks equals the difference between the numbers of unmatched b symbols seen so far. Along increasing cuts, the total variation of this quantity is at most the combined number of unmatched b symbols, at most $2D$.

In case (i), moving a true physical endpoint outward to its grid point changes either symbol-rank by less than h. In case (ii), the one-rank of a first or last canonical boundary differs from the corresponding core one-rank by at most the number of additional ones in that occurrence. On the disjoint parity, the sum of these additional-one errors is at most $2D + 4Kh$ by Lemmas 3.1 and 3.2. The triangle inequality proves the claim, including the initial and terminal diagonals. Notice that the canonical assertion would be false for the zero baseline, which is why case (ii) is explicitly restricted to the one baseline. □

For Green certificates, one common residual shift Δ adds the same quantity to the start and end diagonal of every selected block. It therefore costs only $|\Delta|$ at the two global endpoints. For Blue–Yellow, a forward certificate has

$$(z^-, z^+) = (P_i^F + \Delta,\ P_i^F + \Delta + (T - S)),$$

while a later reversed certificate has

$$(z^-, z^+) = (P_j^R + \Delta + (T - S),\ P_j^R + \Delta).$$

Thus the excursion $T - S$ cancels exactly; the intervening penalty is only $|P_j^R - P_i^F|/2$, already covered by Lemma 6.2.

Corollary 6.3 (Separated directional blocks)**.** *Consider any ordered alternating chain of forward and later reverse Blue–Yellow certificates, all using the same $\Delta \in [0, B/4]$. Relative to the all-one baseline, the chain's adjusted reward is the sum of its local adjusted rewards, minus $O(D + Kh + B)$. In particular the conclusion uses no adjacency assumption on paired directional blocks.*

Proof. For a forward block i the two diagonals are $(P_i^F + \Delta, P_i^F + \Delta + T - S)$; for a later reverse block j they are $(P_j^R + \Delta + T - S, P_j^R + \Delta)$. Thus each $T - S$ excursion cancels at the next forward–reverse interface. All remaining internal penalties are the variation of the unshifted P values and are $O(D + Kh)$ by Lemma 6.2. The unmatched excursion, if the chain has odd length, and the two terminal shifts cost $O(B)$ in total. Apply Lemma 6.1. □

7 The balanced algorithm and its running time

Choose an absolute exponent $A > \max\{A_0, 200\gamma^{-3}\} + 10$ and a sufficiently large absolute constant K_0; $K_0 \geq 10^6\varepsilon^{-6}$ is enough. For all powers of two

$$\frac{N}{(\log N)^A} \leq B \leq \frac{N}{K_0}, \tag{29}$$

the algorithm performs the following operations. Reverse-oriented local tests use $\mathrm{rev}(L_B(X))$; the full inputs are never passed to the modified reversal.

1. Set $h = \lceil \sigma B \rceil$ and enumerate every physical interval with endpoints on the h-grid in each input.
2. For every physical interval pair, store both unary certificates: matching as many zeros as possible and matching as many ones as possible.
3. For every physical interval containing at least B ones, form its first-B and last-B canonical variants.
4. For every pair of canonical variants, construct the weak Green flag arrays at every scale $r = g + 1$, $g \in \mathcal{G}_B$, compute all shift correlations, and store the certified Green score functions. Do this both left-to-right on first canonical windows and right-to-left on reversed last canonical windows.
5. For every canonical pair and every integer $m_0 \in [1, \lfloor \log_2 B \rfloor]$, test the two observable hypotheses of Lemma 5.4. Do this in both ordered directions: $(F_B(X_s), F_B(X_t))$ and $(\mathrm{rev}(L_B(X_t)), \mathrm{rev}(L_B(X_s)))$. When the hypotheses hold, greedily select source flags and compute all directional Blue–Yellow score functions by grouped cross-correlations.

6. For every residual shift Δ in the appropriate range and for each baseline symbol, run (27) on the resulting ordered certificates, evaluate (28), and reconstruct the best explicit chain. Compare it with the two whole-input unary subsequences.

For N below the absolute threshold needed by the structural lemmas, use the quadratic exact LCS algorithm. This does not affect the asymptotic bound, because the threshold is an absolute constant.

Lemma 7.1 (Soundness and running time)**.** *Every reported value is the length of an explicit common subsequence. The algorithm is deterministic and runs in $N \operatorname{polylog} N$ time on a word RAM with $\Theta(\log N)$-bit words.*

Proof. Soundness follows from the explicit local constructions and Lemma 6.1. Prefix counts and select data structures give constant-time symbol-count and one-rank interval queries.

At a scale in (29), the physical grid has $G = O(N/B) = \operatorname{polylog} N$ points. There are $O(G^2)$ intervals per input and $O(G^4)$ interval pairs. Green and Blue–Yellow processing costs $B \operatorname{polylog} B$ per pair, so its total is

$$O(G^4 B \operatorname{polylog} B) = N \operatorname{polylog} N.$$

There are $O(\log \log N)$ candidate scales. At a fixed Δ, the certificate set has size $\operatorname{polylog} N$; running its quadratic DP for every one of $O(B)$ shifts costs $B \operatorname{polylog} N = N \operatorname{polylog} N$. All correlation inputs are nonnegative integer arrays of length $O(B)$ with entries bounded by a fixed polynomial in N. Choose a base 2^w, $w = O(\log N)$, larger than twice the largest possible convolution coefficient and pack the two polynomials as integers (reversing one coefficient sequence for cross-correlation). Kronecker substitution then recovers every coefficient from one exact product of two $O(B \log N)$-bit integers. The deterministic Schönhage–Strassen multiplication algorithm [SS71] therefore gives $B \operatorname{polylog} N$ bit operations, and hence at most that many word operations. No numerical approximation or randomized prime choice is involved. Reconstruction is linear in the total selected span. □

8 Completeness under a near-perfect common subsequence

Lemma 8.1 (The successful scale is tried)**.** *Assume $D \le \eta N$. For all sufficiently large N, the scale B_0 in Proposition 2.1(a), and every scale B furnished by Proposition 2.1(b) after choosing $R = \lceil \log_2 K_0 \rceil$, lie in the range (29).*

Proof. In alternative (a), the floor in the definition of B_0 gives

$$\frac{L}{2(\log_2 L)^{200\gamma^{-3}}} < B_0 \le \frac{L}{(\log_2 L)^{200\gamma^{-3}}}.$$

In alternative (b), $K \le (\log L)^{A_0}$ gives $B = L/K \ge L/(\log L)^{A_0}$, while the choice of R gives $K \ge K_0$. Now use $L > N/4 - D/2$, $L \le N/2$, and the choice of A. The upper bound $B_0 \le N/K_0$ also holds for all sufficiently large N. □

Assume $D \le \eta N$ and use the hidden q' and L from (7). The algorithm does not know them; they are used only to identify a successful candidate chain among the enumerated objects. Lemma 8.1 ensures that the required grid scale is enumerated.

First suppose Proposition 2.1(a) holds. Round the extracted zero-rich interval occurrences on the physical grid at scale B_0 and take the heavier parity. Their total one-rank size remains at least $\gamma^2 L/4$. Indeed, order the extracted intervals. Between intervals j and $j+2$ lies interval $j+1$, whose

two physical occurrence spans each have length at least B_0. Since $2h < B_0$, the outward-rounded spans of either fixed parity are disjoint in both inputs; weighted parity pigeonholing preserves at least half the total one-rank mass. If there are M selected intervals, then $M \le L/B_0$ before parity selection. Their total additional-symbol count after rounding is $O(D + Mh) = O(D + \sigma L)$ by defect charging and (9). Summing Lemma 5.1(a) over the selected intervals and using the global all-one baseline therefore gives

$$\mathrm{ALG}(s,t) \ge \frac{N}{2} + \Omega(\varepsilon\gamma^2 L) - O(D + \sigma N) = \frac{N}{2} + \Omega(\varepsilon^6 L) - O(D + \sigma N). \tag{30}$$

We may therefore assume Proposition 2.1(b). Choose its scale with R large enough that $K = L/B \ge K_0$. A block is *defective* if $e_i + 4h > \kappa B$. By Lemma 3.1, the defective fraction is

$$O(D/(\kappa L) + \sigma/\kappa + 1/(\kappa B)) = o(1).$$

All nondefective rounded occurrences satisfy the hypotheses of Lemma 4.2.

Assign priority types independently to every block and its reversal. We consider three exhaustive cases.

Case 1: many Imbalanced blocks. Suppose at least $K/10$ blocks are assigned Imbalanced in one orientation. Discard defective blocks, choose a parity with at least half the survivors, and pigeonhole the sign of imbalance. At least $K/50$ blocks remain for small enough η, σ. Every witness has one-rank size $r > \varepsilon^5 B$. Use the all-one baseline for zero-rich witnesses and the all-zero baseline for zero-poor witnesses; because the sign was pigeonholed, one single baseline is used by the entire chain. By Lemma 5.1, every selected block has adjusted reward at least $\varepsilon^6 B/3$. The chain identity and Lemma 6.2 yield

$$\mathrm{ALG}(s,t) \ge \frac{N}{2} + \frac{\varepsilon^6}{150} L - O(D + \sigma N). \tag{31}$$

Case 2: many Green blocks. Suppose Case 1 does not hold and at least $K/10$ blocks are assigned Green in one orientation. After defect removal and parity selection, at least $K/30$ remain. Their Green scales may differ. By Lemmas 4.2 and 5.3, each has expected adjusted reward at least $0.05\varepsilon^5 B$ under the same uniform residual shift $\Delta \in [-B, B]$. Linearity of expectation gives one common Δ for which the sum of local rewards has this lower bound. The residual shift is paid only once, so

$$\mathrm{ALG}(s,t) \ge \frac{N}{2} + \frac{\varepsilon^5}{600} L - O(D + \sigma N + B). \tag{32}$$

Case 3: Blue–Yellow in both orientations. If neither of the first two cases occurs in either orientation, more than $4K/5$ blocks are assigned Blue–Yellow forward and more than $4K/5$ are assigned Blue–Yellow after reversal. Intersect these sets with the two regular sets and the nondefective set, then choose a parity. At least $K/4$ survivors remain for large enough N. Order them and use them alternately as forward and reverse blocks, producing at least $K/10$ directional pairs.

By Lemmas 4.2 and 5.4, each direction has expected adjusted reward at least $0.04\varepsilon B$ for one common $\Delta \in [0, B/4]$. The forward–reverse excursions cancel, and Corollary 6.3 controls the separated blocks. Hence some common Δ satisfies

$$\mathrm{ALG}(s,t) \ge \frac{N}{2} + 0.008\varepsilon L - O(D + \sigma N + B). \tag{33}$$

When a dense Imbalanced set occurs in the reverse orientation, map each one-rank witness interval back to its original occurrence; symbol counts and order are preserved. When a dense Green set occurs in the reverse orientation, use the right-to-left arrays of $\mathrm{rev}(L_B(X))$ and map the resulting local matching back. Thus neither argument requires a reversal of an arbitrary full input.

Since $K \geq K_0$, the one-time B loss in (32) and (33) is smaller than a fixed small fraction of the corresponding gain. Combining (30)–(33), there is an absolute $a_* > 0$ such that

$$\mathrm{ALG}(s,t) \geq \frac{N}{2} + a_* L - C(D + \sigma N) \tag{34}$$

for an absolute C. By (7),

$$a_* L \geq a_* \left(N/4 - D/2\right).$$

Choose σ and then η sufficiently small relative to a_*/C. For $D \leq \eta N$, (34) is at least $(1/2 + \delta)N$ for some absolute $\delta > 0$. This proves Theorem 1.1.

9 Reductions to arbitrary inputs

Let $\eta_0, \delta_0 > 0$ be constants for which Theorem 1.1 holds. We may assume $\delta_0 \leq 1/2$. Fix

$$r = \min\left\{\frac{1}{40}, \frac{\eta_0}{20}, \frac{\delta_0}{11}\right\}. \tag{35}$$

For arbitrary binary strings x, y, write

$$\mathrm{BestMatch}(x,y) = \max_{a \in \{0,1\}} \min\{\#_a(x), \#_a(y)\}. \tag{36}$$

This is the length of the longer of the two unary common subsequences and is computable, with its matching, in linear time.

9.1 Nearly balanced equal-length strings

Call a length-m string *r-nearly balanced* if each of its symbol counts lies in $[(1/2 - r)m, (1/2 + r)m]$.

Lemma 9.1 (Trimming lemma)**.** *There is a deterministic $m\,\mathrm{polylog}(m+2)$-time algorithm which, on two r-nearly-balanced strings $x, y \in \{0,1\}^m$, outputs an explicit common subsequence of length at least*

$$\left(\frac{1}{2} + r\right) \mathrm{LCS}(x,y).$$

Proof. If $m = 0$, return the empty subsequence. Henceforth $m > 0$. Put

$$k = \min\{\#_0(x), \#_1(x), \#_0(y), \#_1(y)\}, \qquad M = 2k.$$

Retain exactly k occurrences of each symbol in each string, and let x', y' be the resulting strings. They are exactly balanced and have common length M. Near balance gives

$$M \geq (1 - 2r)m, \qquad m - M \leq 2rm. \tag{37}$$

Let $L = \mathrm{LCS}(x,y)$ and $L' = \mathrm{LCS}(x', y')$. Fix an optimal matching for x, y. Deleting $m - M$ positions from each input destroys at most the sum of those two numbers of matched pairs. Thus

$$L' \geq L - 2(m - M). \tag{38}$$

The factor two is important: the deletions made in the two inputs need not destroy the same matched pairs.

The algorithm returns the better of $\mathrm{BestMatch}(x, y)$ and the output of the balanced promise algorithm on x', y'. Since every symbol count in both inputs is at least k, $\mathrm{BestMatch}(x, y) \geq k = M/2$. If this candidate is shorter than $(1/2 + r)L$, then

$$L > \frac{M}{1 + 2r}. \tag{39}$$

Combining (38), (39), and (37),

$$\begin{aligned} M - L' &\leq M - L + 2(m - M) \\ &< \frac{2r}{1 + 2r} M + 2(m - M) \leq 6rm. \end{aligned} \tag{40}$$

Moreover,

$$\frac{6rm}{M} \leq \frac{6r}{1 - 2r} < \eta_0$$

by (35). Hence the promise in Theorem 1.1 holds for x', y', and that algorithm returns at least $(1/2 + \delta_0)M$ symbols. Finally,

$$\begin{aligned} \left(\frac{1}{2} + \delta_0\right) M &\geq \left(\frac{1}{2} + \delta_0\right) (1 - 2r)m \\ &\geq \left(\frac{1}{2} + r\right) m \geq \left(\frac{1}{2} + r\right) L. \end{aligned}$$

Indeed, the difference between the second line's two coefficients is $\delta_0 - 2r(1 + \delta_0) > 0$ by $r \leq \delta_0/11$ and $\delta_0 \leq 1/2$.

One pass through each input performs the trimming while retaining a map to the original positions. The balanced matching therefore lifts directly to x, y. This pass and reconstruction are linear, and the only nontrivial call takes $M \,\mathrm{polylog}(M + 2)$ time. □

9.2 Arbitrary equal-length strings

We next modify the equal-length reduction of Rubinstein and Song [RS20]. Their proof normalizes all lengths and symbol counts by the common input length. We retain that convention in this subsection. Thus $\#_a(u)$ below means the number of a's in u divided by the original length n, and $\mathrm{LCS}(u, v)$ is normalized by n as well.

For ordered strings U, V, W, let

$$\mathrm{Greedy}(U, V; W) = \max_{W = W_1 W_2} \big(\mathrm{BestMatch}(U, W_1) + \mathrm{BestMatch}(V, W_2)\big),$$

where the maximum is over contiguous prefix–suffix decompositions. A single sweep computes an optimizing split and the associated matching. The notation $\mathrm{Match}(U, V, a)$ denotes the unary matching of symbol a between U and V.

Theorem 9.2 (Equal-length reduction)**.** *There is an absolute constant $\delta_{\mathrm{eq}} > 0$ and a deterministic $n \,\mathrm{polylog}(n + 2)$-time algorithm which, for arbitrary $A, B \in \{0, 1\}^n$, outputs an explicit common subsequence of length at least*

$$\left(\frac{1}{2} + \delta_{\mathrm{eq}}\right) \mathrm{LCS}(A, B).$$

Proof. If n is below a fixed absolute threshold, compute an exact LCS by dynamic programming and return it. Thus $n > 0$, so the normalized quantities used in this subsection are well defined. Set

$$b = \frac{r}{100}, \qquad \tau = \frac{b}{2}, \qquad \delta_{\mathrm{eq}} = \frac{b}{10}. \tag{41}$$

Run BestMatch(A, B) at the outset. By interchanging the inputs and complementing both strings if necessary, assume that

$$\alpha = \min\{\#_0(A), \#_1(A), \#_0(B), \#_1(B)\} = \#_1(A).$$

These symmetries preserve both LCS and explicit matchings. If $\alpha = 0$, A has only one symbol and BestMatch is optimal, so assume $\alpha > 0$.

We first dispose of the cross-unbalanced case. If

$$|\#_1(A) - \#_0(B)| > \tau\alpha, \tag{42}$$

then minimality of α implies $\#_0(B) > (1 + \tau)\alpha$. Also BestMatch$(A, B) = \#_0(B)$ and

$$\mathrm{LCS}(A, B) \le \alpha + \#_0(B).$$

Consequently

$$\frac{\mathrm{BestMatch}(A, B)}{\mathrm{LCS}(A, B)} \ge \frac{1 + \tau}{2 + \tau} = \frac{1}{2} + \frac{\tau}{4 + 2\tau} \ge \frac{1}{2} + \delta_{\mathrm{eq}}. \tag{43}$$

Thus the cross-count argument contributes the gain $\tau/(4 + 2\tau)$ used below.

We may now assume

$$\alpha \le \#_0(B) \le (1 + \tau)\alpha. \tag{44}$$

If $\alpha > 1/2 - 10b\alpha$, then every symbol count of both full inputs is within $(10b + \tau)\alpha < r$ of $1/2$. The pair is therefore r-nearly balanced, and Lemma 9.1 applies. Henceforth assume

$$\alpha \le \frac{1}{2} - 10\beta, \qquad \beta = b\alpha. \tag{45}$$

Partition both inputs into three contiguous pieces,

$$A = L_A M_A R_A, \qquad B = L_B M_B R_B,$$

with all four end pieces of normalized length α. Put

$$a_L = \#_0(L_A), \quad a_R = \#_0(R_A), \qquad b_L = \#_1(L_B), \quad b_R = \#_1(R_B).$$

The following six cases are exhaustive:

(1) $b_R, a_R \le \alpha/2 + 2\beta$;

(2) $b_L, a_L \le \alpha/2 + 2\beta$;

(3) $b_L, b_R \le \alpha/2 + \beta$ and $a_L, a_R > \alpha/2 + 2\beta$;

(4) $b_L, b_R > \alpha/2 + 2\beta$ and $a_L, a_R \le \alpha/2 + \beta$;

(5) $b_R, a_L > \alpha/2 + \beta$;

(6) $b_L, a_R > \alpha/2 + \beta$.

For completeness, suppose neither (5) nor (6) holds. If (1) fails because $b_R > \alpha/2 + 2\beta$, then $a_L \le \alpha/2 + \beta$; failure of (2) forces $b_L > \alpha/2 + 2\beta$, which in turn forces $a_R \le \alpha/2 + \beta$, giving (4). If instead (1) fails because $a_R > \alpha/2 + 2\beta$, the symmetric argument gives (3). Thus failure of (1), (2), (5), and (6) implies (3) or (4).

All six conditions depend only on observed symbol counts. The algorithm computes every candidate prescribed by every condition that holds, including the reversed copies below, and returns the longest candidate together with the initial $\mathrm{BestMatch}(A, B)$ output.

We verify the certificate in every case. In Case (1), first suppose

$$a_R, b_R \in [\alpha/2 - 4\beta, \alpha/2 + 4\beta]. \tag{46}$$

The pair (R_A, R_B) is $4b$-nearly balanced relative to its common length αn, and hence is within the range of Lemma 9.1 because $4b < r$.

The algorithm computes all three candidates

$$\begin{aligned} G_{AB} &= \mathrm{Greedy}(L_A M_A, R_A; B), \\ G_{BA} &= \mathrm{Greedy}(L_B M_B, R_B; A), \\ P &= \mathrm{BestMatch}(L_A M_A, L_B M_B) + \mathrm{NB}(R_A, R_B), \end{aligned} \tag{47}$$

where NB is the algorithm from Lemma 9.1. All three candidates are needed because the proof may branch on a partition induced by an unknown optimal matching, while the algorithm has access only to the explicitly computed greedy splits.

Fix an optimal matching. If it has no match from R_A into $L_B M_B$, it induces a decomposition $B = \widehat{L}_B \widehat{R}_B$ for which

$$\mathrm{LCS}(A, B) = \mathrm{LCS}(L_A M_A, \widehat{L}_B) + \mathrm{LCS}(R_A, \widehat{R}_B).$$

Otherwise, noncrossing implies that it has no match from $L_A M_A$ into R_B, and the transposed decomposition used by G_{BA} applies. Consider the applicable direction. Let X, Y be the symbol-count upper bounds for the left and right LCS terms, and let Z be the larger of the two unary contributions to X. Then

$$\mathrm{LCS}(A, B) \le X + Y, \qquad Z \ge X/2, \qquad G_{AB} \text{ or } G_{BA} \ge Z + Y/2. \tag{48}$$

Condition (46) and (44) give $X - Z \le \alpha/2 + 4\beta + \tau\alpha$. Indeed, in the G_{AB} direction the smaller unary contribution to X is at most $\#_1(L_A M_A) = a_R \le \alpha/2 + 4\beta$. In the G_{BA} direction it is at most

$$\#_0(L_B M_B) = \#_0(B) - \alpha + b_R \le \alpha/2 + 4\beta + \tau\alpha.$$

Hence, if $Z > \alpha/2 + 10\beta$, the applicable greedy candidate is at least

$$\frac{1}{2}\,\mathrm{LCS}(A, B) + 2\beta.$$

If $Z \le \alpha/2 + 10\beta$, then

$$\mathrm{LCS}(A, B) \le \alpha + 20\beta + \mathrm{LCS}(R_A, R_B). \tag{49}$$

Furthermore

$$\mathrm{BestMatch}(L_A M_A, L_B M_B) \ge \alpha/2 - 4\beta$$

and $\mathrm{LCS}(R_A, R_B) \geq \alpha/2 - 4\beta$. Thus the candidate P exceeds half the right side of (49) by at least

$$r\,\mathrm{LCS}(R_A, R_B) - 14\beta \geq \big(r(1/2 - 4b) - 14b\big)\alpha \geq 30\beta.$$

Here we used $b = r/100$ and $r \leq 1/40$.

It remains in Case (1) to treat the complement of (46). Then either

$$b_R < \alpha/2 - 4\beta, \quad a_R \leq \alpha/2 + 2\beta,$$

or the same statement holds with a_R, b_R interchanged. In either alternative,

$$a_R + b_R \leq \alpha - 2\beta. \tag{50}$$

Splitting an optimal noncrossing matching at the right blocks, the two possible matching directions give the bounds

$$a_R + \#_0(B) + b_R \leq 2\alpha - 2\beta + \tau\alpha$$

and

$$(\#_0(B) - \alpha + b_R) + \alpha + a_R \leq 2\alpha - 2\beta + \tau\alpha.$$

The latter charges zeros in the left B-piece, all ones of A, and zeros in the right A-piece; these three classes cover both matching subproblems. The two displayed expressions are equal to $\#_0(B)+a_R+b_R$. Thus (50) and (44) prove, in both alternatives,

$$\mathrm{LCS}(A, B) \leq 2\alpha - 2\beta + \tau\alpha. \tag{51}$$

Since $\mathrm{BestMatch}(A, B) \geq \alpha$, that upper bound gives an additive advantage over half the LCS of at least

$$\beta - \frac{\tau\alpha}{2} = \frac{3}{4}\beta. \tag{52}$$

Case (2) is obtained by reversing both strings; the algorithm runs the corresponding candidates.

In Case (3), concatenate

$$\mathrm{Match}(L_A, L_B, 0), \quad \mathrm{Match}(M_A, M_B, 1), \quad \mathrm{Match}(R_A, R_B, 0). \tag{53}$$

The two outer terms each have length at least $\alpha/2 - \beta$. The middle of A has more than 4β ones. The middle of B has at least

$$1 - (1 + \tau)\alpha - (\alpha + 2\beta) > 4\beta$$

ones by (45). Thus (53) has length at least $\alpha + 2\beta$. In Case (4), concatenate the one matchings on the two end pairs and the zero matching on the middle pair. Each outer term has length at least $\alpha/2 - \beta$. The middle of B has

$$\#_0(M_B) = \#_0(B) - (\alpha - b_L) - (\alpha - b_R) > 4\beta,$$

where we used $\#_0(B) \geq \alpha$ and the case assumptions. The middle of A has

$$\#_0(M_A) = 1 - \alpha - a_L - a_R \geq 1 - 2\alpha - 2\beta > 4\beta$$

by (45). This gives the same total lower bound $\alpha + 2\beta$ without invoking an exact symmetry.

In Case (5), concatenate

$$\mathrm{Match}(L_A, L_B M_B, 0), \qquad \mathrm{Match}(M_A R_A, R_B, 1). \tag{54}$$

Each term has length greater than $\alpha/2+\beta$: for instance,

$$\#_0(L_B M_B) = \#_0(B) - (\alpha - b_R) \geq b_R > \alpha/2 + \beta,$$

and the other three required counts follow directly from the case assumptions. Hence (54) again has length greater than $\alpha + 2\beta$. Case (6) follows by reversal.

Finally, throughout the current branch,

$$\mathrm{LCS}(A,B) \leq \min\{\#_0(A),\#_0(B)\} + \min\{\#_1(A),\#_1(B)\} \leq (2+\tau)\alpha. \tag{55}$$

The smallest additive gain established above is $3\beta/4$. Together with (41) and (55), this is more than $\delta_{\mathrm{eq}}\,\mathrm{LCS}(A,B)$. The cross-unbalanced and nearly balanced branches already have the same guarantee. This completes the case analysis.

Every certificate used above is unary or is returned explicitly by Lemma 9.1; concatenated certificates lie in ordered, disjoint subproblems. There are only constantly many oracle calls, and all counts, greedy splits, symmetries, and reconstruction maps take linear time. The total running time is therefore $n\,\mathrm{polylog}(n+2)$. □

Only the following uses of ApproxED from Rubinstein–Song [RS20] enter the preceding proof: its approximately balanced Lemma 3.2, the Case 1(a) call on (R_A, R_B), and the reversed copy of that case. All other cases are precisely the unary and greedy constructions verified above. Thus no edit-distance guarantee remains hidden in Theorem 9.2.

9.3 Unequal lengths and the final factor

We use the following quantitative form of the He–Li reduction. It is stated in the discussion immediately following their main theorem and is implemented by their construction [HL23].

Proposition 9.3 (He–Li equal-to-unequal reduction)**.** *There is an absolute constant $A \geq 1$ with the following property. Suppose that, for every k, a deterministic algorithm on equal-length binary strings of length at most k runs in $k\,\mathrm{polylog}(k+2)$ time and outputs an explicit common subsequence of length at least $(1/2+\zeta)$ times optimal. Then arbitrary binary strings x, y of maximum length N admit a deterministic explicit $(1/2+\zeta^A)$-approximation in $N\,\mathrm{polylog}(N+2)$ time.*

We record the bookkeeping bounds needed to apply the He–Li construction in this form. They also determine how its hierarchy of constants is chosen.

First consider a near-equal oracle call on lengths $p \leq q$, where $q - p \leq \lambda p$ and completeness supplies an optimum $L \geq (1-\chi)p$. Delete $q-p$ symbols from the longer local string. The equal-length instance has optimum L' satisfying

$$L' \geq L - (q-p) \geq \left(1 - \frac{\lambda}{1-\chi}\right) L. \tag{56}$$

Choose the scale parameters so that

$$\frac{\lambda}{1-\chi} \leq \frac{\zeta}{2}. \tag{57}$$

Then an equal-length $(1/2+\zeta)$-approximation on the trimmed pair returns at least $(1/2+\zeta/2)L$ symbols, because $\zeta/2 \leq \zeta/(1+2\zeta)$. He and Li choose λ and χ as fixed powers of ζ, so this requirement only changes absolute exponents in their hierarchy.

Second, write

$$u = \min\{\#_0(x),\#_0(y)\}, \qquad v = \min\{\#_1(x),\#_1(y)\},$$

and suppose $u \geq v$. If $u \geq (1+\theta)v$, the unary candidate has ratio at least

$$\frac{1+\theta}{2+\theta} > \frac{1}{2} + \frac{\theta}{5}.$$

Otherwise $d = u - v < \theta v \leq \theta L$, where $L = \mathrm{LCS}(x, y)$, because the v copies of the minority symbol form a common subsequence. A balancing step may delete d symbols from each input, so the safe bound is

$$L' \geq L - 2d \geq (1 - 2\theta)L. \tag{58}$$

If the internal algorithm has gain Δ, then for $\theta \leq \Delta/10$ its output on the trimmed instance is at least

$$\left(\frac{1}{2} + \Delta\right)(1 - 2\theta)L \geq \left(\frac{1}{2} + \frac{7\Delta}{10}\right)L,$$

using $\Delta \leq 1/2$.

Finally, an earlier one-input balancing step assumes $\#_1(x) \geq (1/2 - \rho)|x|$. The number deleted there is at most $2\rho|x|$, rather than $\rho|x|$, while $L \geq (1/2 - \rho)|x|$. Thus its surviving optimum obeys

$$L' \geq \left(1 - \frac{4\rho}{1 - 2\rho}\right)L. \tag{59}$$

Taking $\rho \leq \Delta/20$ in (59) preserves at least half of the internal gain. The hierarchy in the He–Li construction takes all three tolerances to be fixed powers of ζ; after the displayed losses, increasing the absolute exponent A preserves the form of Proposition 9.3.

One oracle call is implicit in the imbalanced-input reduction used by He and Li: the balanced end-block subcase of their cited linear-time lemma invokes an equal-length approximate-edit-distance routine. In the present instantiation, replace that invocation by the assumed equal-length LCS algorithm and retain its explicit certificate. At a block length m, the construction makes $O((N/m)\,\mathrm{polylog}(N+2))$ calls on strings of length $O(m/\log(m+2))$. Since every such call costs its input length times a polylogarithmic factor, the sum over all calls and all logarithmically many scales is $N\,\mathrm{polylog}(N+2)$.

For explicit output, store the certificate attached to every accepted rectangle and the backpointer of every dynamic-programming state. Unary and structural certificates are embedded greedily, and deletion position maps lift all oracle certificates to the original strings. Fix lexicographically first choices in ties, so determinism is preserved. Inputs below the absolute block-scale threshold, including empty strings, are handled by exact dynamic programming. This completes the verification of Proposition 9.3 for the stated near-linear oracle.

Proof of Theorem 1.2. If $|s| + |t|$ is below the absolute threshold in Proposition 9.3, compute an exact LCS and return it. Henceforth $N = \max\{|s|, |t|\} > 0$. Apply Proposition 9.3 to the algorithm of Theorem 9.2, with

$$\zeta = \delta_{\mathrm{eq}}, \qquad \delta_{\mathrm{all}} = \delta_{\mathrm{eq}}^{A}.$$

Its running time is $T(N) = N\,\mathrm{polylog}(N+2)$, so the resulting running time is still $N\,\mathrm{polylog}(N+2)$. Since

$$N = \max\{|s|, |t|\} \leq |s| + |t| \leq 2N,$$

this is also $n\,\mathrm{polylog}(n+2)$ for $n = |s| + |t|$.

The returned subsequence has length at least $(1/2+\delta_{\text{all}})\,\text{LCS}(s,t)$. Its conventional approximation factor is therefore at most

$$\begin{aligned}\frac{1}{1/2+\delta_{\text{all}}} &= \frac{2}{1+2\delta_{\text{all}}} \\ &= 2 - \frac{4\delta_{\text{all}}}{1+2\delta_{\text{all}}}.\end{aligned}$$

Taking

$$c = \frac{4\delta_{\text{eq}}^{A}}{1+2\delta_{\text{eq}}^{A}} > 0$$

proves the theorem. □

References

[ABW15] Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 59–78. IEEE, 2015.

[AKO10] Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 377–386. IEEE, 2010.

[AN20] Alexandr Andoni and Negev Shekel Nosatzki. Edit distance in near-linear time: It's a constant factor. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 990–1001. IEEE, 2020.

[AO12] Alexandr Andoni and Krzysztof Onak. Approximating edit distance in near-linear time. *SIAM Journal on Computing*, 41(6):1635–1648, 2012.

[AW21] Shyan Akmal and Virginia Vassilevska Williams. Improved approximation for longest common subsequence over small alphabets. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics*, pages 13:1–13:18. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021.

[BCAD23] Karl Bringmann, Vincent Cohen-Addad, and Debarati Das. A linear-time $n^{0.4}$-approximation for longest common subsequence. *ACM Transactions on Algorithms*, 19(1):9:1–9:24, 2023.

[BGK25] Itai Boneh, Shay Golan, and Matan Kraus. Deterministic longest common subsequence approximation in near-linear time, 2025. arXiv:2507.22486.

[BK15] Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 79–97. IEEE, 2015.

[CDG+20] Diptarka Chakraborty, Debarati Das, Elazar Goldenberg, Michal Koucký, and Michael E. Saks. Approximating edit distance within constant factor in truly sub-quadratic time. *Journal of the ACM*, 67(6):36:1–36:22, 2020.

[GHL22] Venkatesan Guruswami, Xiaoyu He, and Ray Li. The zero-rate threshold for adversarial bit-deletions is less than 1/2. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 727–738. IEEE, 2022. Expanded version, arXiv:2106.05250v2.

[Gra16] Szymon Grabowski. New tabulation and sparse dynamic programming based techniques for sequence similarity problems. *Discrete Applied Mathematics*, 212:96–103, 2016.

[HL23] Xiaoyu He and Ray Li. Approximating binary longest common subsequence in almost-linear time. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1145–1158. ACM, 2023. Expanded version, arXiv:2211.16660v2.

[HSSS19] MohammadTaghi Hajiaghayi, Masoud Seddighin, Saeed Seddighin, and Xiaorui Sun. Approximating LCS in linear time: Beating the $\sqrt{n}$ barrier. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1181–1200. SIAM, 2019.

[MP80] William J. Masek and Michael S. Paterson. A faster algorithm computing string edit distances. *Journal of Computer and System Sciences*, 20(1):18–31, 1980.

[MR26] Xiao Mao and Aviad Rubinstein. Approximation schemes for edit distance and LCS in quasi-strongly subquadratic time, 2026. arXiv:2603.29702.

[Nos21] Negev Shekel Nosatzki. Approximating the longest common subsequence problem within a sub-polynomial factor in linear time, 2021. arXiv:2112.08454.

[RS20] Aviad Rubinstein and Zhao Song. Reducing approximate longest common subsequence to approximate edit distance. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1591–1600. SIAM, 2020.

[SS71] Arnold Schönhage and Volker Strassen. Schnelle multiplikation großer zahlen. *Computing*, 7:281–292, 1971.