PAIR ENCODING: APPROXIMATION AND THE COST OF A STRICT BUDGET

Anonymous authors
Paper under double-blind review

ABSTRACT

How well can pair encoding compress a string with a limited number of merge rules? We characterize legal partial executions as reachable creation-ordered binary forests and establish a separation between strict and relaxed budgets. For every string, we construct one binary hierarchy whose prunings approximate both the rule count and final length of every comparison forest within $O(\log^2 n)$. This yields a uniform weighted approximation and an unconditional budget/length bicriteria algorithm. Conversely, unless $\mathrm{P} = \mathrm{NP}$, no deterministic polynomial-time algorithm achieves a sublinear length factor under a strict budget on growing-alphabet inputs, even with sufficiently small fixed budget inflation and complete freedom to redesign the dictionary. The hardness is witnessed by an NP-hard gap between one final token and linearly many tokens; extracting a vertex cover from any output forest makes the conclusion independent of dictionary design. The rule-free n-approximation has the optimal worst-case asymptotic order at strict budget. We also derive computable length certificates and sharp utility-loss identities across snapshot, final-state, and temporal resolutions. Reproducible experiments retain all $9{,}473$ registered records, including $8{,}252$ exact-OPE evaluations and 288 full-window runs, measuring certificate strength, policy sensitivity, and the structural identities on finite inputs. Code and reproducibility materials are available anonymously.[1]

1 INTRODUCTION

Pair encoding turns repeated adjacent tokens into a learned vocabulary. Introduced into neural machine translation by Sennrich et al. (2016), it makes compression at a fixed vocabulary budget a basic algorithmic question (Gallé, 2019; Zouhar et al., 2023). How short a sequence can a polynomial-time algorithm produce with k merge rules? We answer this question for one-string, selected-occurrence pair encoding by exposing the hierarchy behind its local operations.

A rule introduces a fresh symbol and replaces a nonempty disjoint subset of its pair occurrences. On `abababa`, replacing all three ab occurrences gives $XXXa$ using $X \to ab$; partial replacement may select fewer. We prove that these executions are exactly reachable creation-ordered binary forests, preserving the number of rules r and final tokens L. This exact state space exposes the budget distinction summarized in Table 1.

The central separation. Our positive theorem constructs one binary recompression hierarchy for a string and optimizes its dependency-closed prunings exactly. A comparison-grammar argument then proves that this one hierarchy approximates *every* feasible rule/root pair, giving $r \le O(\log^2 n)k$ and $L \le O(\log^2 n)L^*(s,k)$. Our negative theorem shows why a strict budget is fundamentally different: the trivial n-approximation has the optimal worst-case asymptotic order for deterministic polynomial time on growing alphabets, unless $\mathrm{P} = \mathrm{NP}$. The obstruction survives a sufficiently small fixed budget increase and complete dictionary redesign. Its force is visible in a concrete promise gap: a YES instance has a one-token encoding, while every budget-feasible encoding of a NO instance has linearly many tokens (Proposition 5.3). Lemma 5.4 explains why redesign cannot evade the reduction: any output forest yields a vertex cover whose size is controlled by its rules and roots.

[1] https://anonymous.4open.science/r/pair-encoding-iclr2027-9003

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Contribution	Main result
APPROXIMATION	One host approximates every forest's (r, L) within $O(\log^2 n)$, yielding uniform weighted and budget/length algorithms (Theorem 4.2, Corollary 4.3).
HARDNESS	No deterministic strict-budget $o(n)$ length approximation unless P = NP on growing alphabets, even with small fixed budget inflation and redesigned dictionaries (Theorem 5.1); a one-token/linear-length gap and arbitrary-forest extraction expose the mechanism (Proposition 5.3, Lemma 5.4).
STRUCTURE	Exact execution–forest equivalence, computable length certificates, and sharp utility-loss identities at snapshot, final-state, and temporal resolutions (Sections 3, 6, and 7).
EVIDENCE	9,473 registered records, including 8,252 exact-OPE evaluations; certificate improvement in 288/288 windows, policy-sensitive diagnostics, and byte-identical replay (Section 8).

Table 1: The rule budget separates two approximation regimes. All algorithms and hardness statements use the selected-occurrence one-string model.

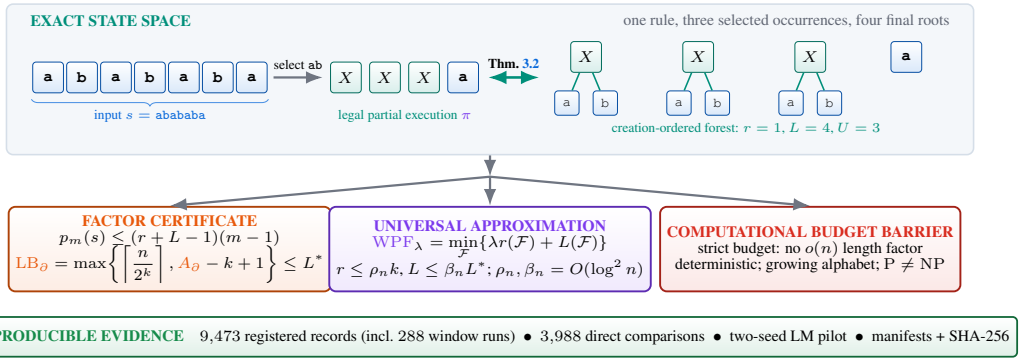

Figure 1: Legal pair replacements and binary forests encode the same state. This exact representation supports approximation, budget hardness, and instance certificates (Sections 4–6).

The prior length APX-hardness excludes sufficiently accurate constant-factor approximations, and a linear lower bound for BPE constrains that particular algorithm (Kozma & Voderholzer, 2026). Our theorem excludes *every* sublinear length factor for *every* deterministic polynomial-time algorithm satisfying the stated budget guarantee. The known graph gadget supplies the input family; arbitrary-forest extraction and the quantitative budget transfer supply this stronger conclusion.

Utility $U = n - L$ remains a separate approximation frontier: a multiplicative utility guarantee does not imply a multiplicative length guarantee. Our exact packing, fragmentation, overlap, and spoiler-chain results identify the hierarchical information that stronger utility arguments must retain. The experiments measure certificate strength and utility geometry on explicit finite domains, retaining nulls and policy differences. The host's worst-case approximation guarantee is established by proof; its practical budget–length tradeoff remains an empirical question. Sections 4 and 5 develop the central approximation and hardness results immediately after the exact representation. Sections 6 and 7 then give instance certificates and the utility laws tested in Section 8.

2 Models and Prior Work

Canonical one-string model. For $s \in \Sigma^n$, rule i creates a fresh $X_i \to Y_i Z_i$, with children terminals or earlier labels, and replaces a nonempty pairwise-disjoint subset of current $Y_i Z_i$ occurrences. We use at most k rules and minimize final length; dictionary storage is excluded. Thus

$$L^*_{\mathrm{OPE}}(s, k) = \min_{\pi:\, r(\pi) \le k} L(\pi), \qquad O = n - L^*_{\mathrm{OPE}}(s, k).$$

Full OMS replaces every nonoverlapping occurrence under the left-to-right convention. Deterministic BPE selects the highest-frequency full rule with first-occurrence tie breaking; EVENODD is the partial policy of Kozma & Voderholzer (2026). We write O, B, E for the utilities of optimal OPE, BPE, and EVENODD. Input symbols, created hierarchy objects, and execution quantities are colored blue, teal, and violet in the diagrams.

Pair encoding and tokenization. Zouhar et al. (2023) give a formal BPE optimization program, a curvature-dependent greedy guarantee, and a forward ordered-forest description for full merge sequences. Our equivalence includes selected occurrences, reverse replay, and every exact (r, L) pair. Kozma & Voderholzer (2026) establish utility APX-completeness, length APX-hardness, $E \geq O/2$, a BPE utility ratio between $1/3$ and $5/8$, and a linear BPE length gap. Theorem 5.1 addresses all deterministic algorithms for length. Corpus direct-vocabulary and bottom-up tokenization have different feasible sets: the NP-completeness results of Whittington et al. (2025), partition-cover formulation of Lim et al. (2025), and binary APX-hard/no-PTAS results of Kastreva et al. (2026) do not transfer automatically to one-string partial replacement. Table 6 in Appendix J compares these feasible sets and objectives; Figure 1 unifies the forest representation, theorem guarantees, and empirical evidence. Appendix C.1 gives the explicit at-most-budget alignment of the earlier hardness reduction, with its source-version arithmetic.

Grammar compression. Binary acyclic grammars with explicit start sequences and our forests represent the same hierarchies. Costs distinguish them: ordinary right-hand-side size (Charikar et al., 2005) is within a factor two of WPF_2, giving a sharper logarithmic approximation at that weight via Jeż (2014); Theorem C.2 in Appendix C.4 gives both conversions, their cost bounds, and normalization details. We use the boundary-popping primitive of Jeż (2013), with separate production and start-sequence accounting, to obtain uniform control of both coordinates. For hardness, we credit the binary-SLP graph gadget and its exact optimum to Kawamoto et al. (2024); our extraction from arbitrary forests and budget/length transfer give the new computational conclusion. String attractors (Prezza, 2017; Kempa & Prezza, 2018) describe occurrence coverage, while the SLP q-gram boundary decompositions (Goto et al., 2011; Bille et al., 2013) transfer through the forest bridge to our computable length certificate.

3 THE ORDERED-FOREST GEOMETRY OF OPE

A creation-ordered binary forest records exactly which hierarchies a partial execution can realize, retaining its rule budget and final token sequence.

Definition 3.1 (Reachable creation-ordered forest)**.** A forest for s has productions $X_i \to Y_i Z_i$ with fresh X_i and children drawn from terminals or $X_1, \ldots, X_{i-1}$, together with ordered roots whose expanded yields concatenate to s. Every production must be reachable from a root. Write $r(\mathcal{F})$ and $L(\mathcal{F})$ for its productions and roots.

Theorem 3.2 (Execution–forest equivalence)**.** *For every finite s and every pair of nonnegative integers (r, L), a legal nonempty-rule OPE execution uses r rules and ends in L tokens iff a reachable creation-ordered forest for s has r productions and L roots.*

Proof. An execution contributes one production per fresh label and takes its final tokens as roots. Freshness gives creation order; reversal recovers s; nonempty use makes every production reachable. Conversely, process a forest in creation order and replace exactly the occurrences represented by nodes labelled X_i. Equal-labelled nodes cannot be ancestor-related without a dependency cycle, hence their leaf intervals are disjoint. They are therefore a legal nonempty partial selection. The roots and the pair (r, L) are preserved. Appendix A gives the replay invariant and edge cases. □

EXACT EQUIVALENCE **The representation loses nothing: feasibility, rule budget, final length, and utility pass in both directions.**

This theorem is an exact change of language:

$$L^*_{\mathrm{OPE}}(s, k) = \min\{L(\mathcal{F}) : \mathcal{F} \text{ reachable},\ r(\mathcal{F}) \leq k\}.$$

Equivalently, it is a binary acyclic grammar with an ordered start sequence; Section 4 uses this bridge while keeping direct tokens, OMS, and corpus merge sequences separate.

4 ONE HIERARCHY APPROXIMATES EVERY COMPARISON FOREST

For a positive dyadic weight λ, let $\mathrm{WPF}_\lambda(s) = \min_F\{\lambda r(F) + L(F)\}$ over reachable forests of s. We construct a single hierarchy that controls the two coordinates separately. The mechanism is recompression followed by exact frequency pruning.

Lemma 4.1 (Exact closed pruning). *Let m_X count occurrences of production X in the fully unfolded host forest H. A label set D is closed if it contains every nonterminal child of each retained label. Expanding unretained roots gives*

$$r_D = |D|, \qquad L_D = n - \sum_{X \in D} m_X.$$

Sorting by decreasing m_X, with creation order breaking ties, yields a closed prefix optimal among all closed prunings at each rule budget. Retaining exactly $m_X > \lambda$ minimizes weighted cost, with value $n - \sum_X \max\{m_X - \lambda, 0\}$.

Every child has at least its parent's multiplicity, so the indicated sets are closed. The identity counts retained internal occurrences in a binary forest; maximizing their sum proves both optimality statements. This optimizes prunings of a host; the next theorem supplies its global approximation power. Appendix K.1 gives the full closure, reachability, and tie-breaking argument.

Algorithm 1 Universal binary host and frequency selection

Input: explicit string s; weight $\lambda > 0$ or budget $k \geq 0$
Output: reachable forest selected by weight or augmented budget

```
w ← s; initialize an empty creation-ordered dictionary
while |w| > 1 do
    build shared binary powers for each maximal-run base
    replace runs using private concatenation chains per distinct base/run-length pair
    choose an alphabet partition (A_L, A_R) by conditional expectations
    replace all A_L A_R pairs, sharing one fresh binary rule per pair type
propagate root multiplicities backwards through the dictionary
return closed pruning by m_X > λ, or the first min{r_H, ρ_n k} rules in frequency order
```

Theorem 4.2 (Universal two-coordinate hierarchy). *For every explicit string s of length $n \geq 2$, Algorithm 1 constructs in deterministic polynomial time a reachable one-root binary host $H(s)$ with at most $n-1$ productions. Put $h = \lceil \log_2 n \rceil$, and let p be its number of phases. For* every *comparison forest F of s, a closed pruning D of this same host satisfies*

$$|D| \leq \rho(s) r(F), \qquad L_D \leq \beta(s) L(F), \qquad \rho(s) = 6h(1+4p), \quad \beta(s) = 1 + h(1+6p). \tag{1}$$

The host depends only on s, and $p = O(\log n)$.

Why one host suffices. We simulate its phases on the comparison grammar using boundary popping (Jeż, 2013). Popping exposes every compressed run and pair without adding comparison references. Counting inserted and deleted atoms separately in secondary productions and in the start sequence pays, respectively, for retained rule types and expanded root occurrences. Shared powers make the retained labels dependency-closed; private run concatenations bound unretained occurrences. This separate accounting proves (1) for each comparator without changing the host.

Appendix K proves the four steps: auxiliary construction and legality (Appendix K.2, Lemma K.1); comparison simulation and atom telescope (Appendices K.3–K.4, (10)); closed marking and its rule bound (Appendix K.5, (11)); and the root bound (Appendix K.6).

Let q be the least positive integer with $3^q(n-1) < 4^q$, and set $\rho_n = 6h(1+4q)$, $\beta_n = 1+h(1+6q)$. Each phase contracts $|w| - 1$ by at least $1/4$, so $p \leq q$ and $\beta_n \leq \rho_n = O(\log^2 n)$.

Corollary 4.3 (Uniform weighted and budgeted approximation). *Algorithm 1 returns legal* OPE *forests satisfying, respectively,*

$$\lambda r + L \leq \rho_n \mathrm{WPF}_\lambda(s), \qquad \textit{or} \qquad \boxed{r \leq \rho_n k, \quad L \leq \beta_n L^*_{\mathrm{OPE}}(s,k)}.$$

Runtime is polynomial in the explicit string, budget, and dyadic encoding lengths, including binary-encoded signed exponents. For $n \le 1$ set both constants to one and return the exact rule-free forest; $k = 0$ and $k \ge n - 1$ are also exact.

Apply Theorem 4.2 to an optimal comparator, then use Lemma 4.1 to select an at-least-as-good pruning. No optimal grammar or multiplier-dependent host enters the algorithm. Appendix K.7 gives both selection arguments; Appendix K.8 proves the encoding-sensitive runtime. The unary, all-distinct, empty-input, and shortest nonexact calibrations in Appendix K.9 distinguish exact pruning within a host from approximation over all forests.

Scalar guarantees and budget repair. At weight two, Theorem C.2 gives the sharper $O(1 + \log(n/\mathrm{WPF}_2))$ approximation. The general scalar conversion (Theorem C.1, Algorithm 3) and its arithmetic are in Appendices C.2–C.3. Three exact mismatches and the diverging completed-phase selector gap (Proposition C.3, Algorithm 4) are in Appendices C.5–C.6; Appendix C collects this scalar analysis. Even arbitrarily small constant slack in weighted optimality does not ensure strict-budget repair: a unary family forces $\Omega(n^\eta)$ roots when $L^*_{\mathrm{OPE}} = 1$ if productions must remain unchanged (Theorem K.2, Appendix K.10). The next section permits complete dictionary redesign and proves a computational barrier.

5 The Linear Cost of a Strict Budget

The universal hierarchy establishes what budget relaxation can achieve. We now prove that preserving the budget changes the approximation problem: even a sufficiently small fixed inflation excludes every sublinear worst-case length factor in the general one-string model.

Theorem 5.1 (Computational budget barrier)**.** *Unless* $\mathrm{P} = \mathrm{NP}$*, there are fixed constants $\varepsilon_0, c_0 > 0$ such that no deterministic polynomial-time algorithm returns legal forests with*

$$r \le (1 + \varepsilon_0)k, \qquad L \le c_0 n L^*_{\mathrm{OPE}}(s, k)$$

for all sufficiently long explicit strings over growing alphabets. This permits arbitrary new productions and arbitrary legal occurrence choices.

Corollary 5.2 (Three consequences)**.** *Under the same assumption and model: (i) strict-budget $o(n)$ approximation, hence polylogarithmic approximation, is impossible; (ii) no scheme works for every fixed $\varepsilon > 0$ with $r \le (1 + \varepsilon)k$ and length factor $\mathrm{poly}(\log n, 1/\varepsilon)$; (iii) redesigning the dictionary does not remove this computational obstruction. The rule-free forest is an n-approximation, matching the strict-budget lower bound in asymptotic order.*

The statements and all three consequences are proved below and in Appendix L.5, within the complete reduction in Appendix L. The key phenomenon is a gap at one-root feasibility, which we make explicit before developing the extraction.

5.1 One Token Versus Linearly Many Tokens

Let $\gamma(s)$ be the fewest binary rules in a one-root forest. Joining any forest's roots gives $\gamma(s) \le r + L - 1$, and hence, for nonempty s,

$$\mathrm{WPF}_1(s) = \gamma(s) + 1, \qquad L^*_{\mathrm{OPE}}(s, k) = 1 \iff \gamma(s) \le k.$$

Lemma L.1 and its proof in Appendix L.1 establish these identities and the exact terminal-alias convention. A deficit of available rules therefore forces additional roots.

For cubic vertex cover, there are fixed rational constants $0 \le \beta < \alpha \le 1$ such that distinguishing $\tau(G) \le \beta v$ from $\tau(G) > \alpha v$ is NP-hard, where v counts vertices and τ is minimum cover size (Alimonti & Kann, 2000; Friggstad et al., 2025). Let w_G be the graph encoding defined in the next subsection; for cubic G, its length is $n = 21v$.

Proposition 5.3 (One-root versus linear-length gap)**.** *Set $t = \lfloor \beta v \rfloor$ and $k = \frac{29}{2}v + t - 1$. On the YES instances, $L^*_{\mathrm{OPE}}(w_G, k) = 1$. On the NO instances, every forest with $r \le (1 + \varepsilon)k$ satisfies*

$$L > \big[(\alpha - \beta) - \varepsilon(29/2 + \beta)\big]v + 1 + \varepsilon. \tag{2}$$

For every fixed $0 \le \varepsilon \le (\alpha - \beta)/(2(29/2 + \beta))$, the NO-instance bound is $\Omega(n)$, regardless of the output dictionary. Distinguishing these YES and NO cases is NP-hard.

Thus the computational obstruction separates a complete one-token encoding from linear-length output even after a small budget increase. A single query would distinguish the cases if it respected the enlarged budget and achieved a sublinear length factor. Appendix L.4 gives the complete proof, including the floor in t and integrality of $29v/2$.

5.2 Extraction from any output dictionary

We use the binary-SLP graph construction of Kawamoto et al. (2024). For a simple graph $G = (V, E)$ with $v = |V| \geq 1$ and $e = |E|$, give each vertex its own terminal and add $\#$. Every $\diamond$ occurrence is a different private terminal, including those in the two blocks for the same vertex. Fix vertex and edge orders and one orientation per edge, and form

$$\begin{aligned} X &= \prod_{a \in V} \big[(\#a \diamond a\#\diamond)(\#a \diamond a\#\diamond)\big], \\ Z &= \prod_{(a,b) \in E} (\#a\#b\#\diamond), \qquad w_G = XZ. \end{aligned} \tag{3}$$

The length is $12v + 6e$, and the alphabet has $5v + e + 1$ symbols.

Lemma 5.4 (Arbitrary-forest cover extraction)**.** *Write $C_G = 10v + 3e - 1$. The graph encoding has the known binary optimum $\gamma(w_G) = C_G + \tau(G)$. From* any *reachable forest of w_G with coordinates (r, L), a vertex cover Q can be computed in polynomial time with*

$$|Q| \leq r + L - 1 - C_G = r + L - 10v - 3e. \tag{4}$$

The gadget and its one-root optimum are due to Kawamoto et al. (2024), Theorem 3.1. The arbitrary-forest extraction is the bridge to our budget objective: every algorithmic output carries a cover certificate, even after complete dictionary redesign.

Proof mechanism. Expand labels used once, preserving $r + L$. The remaining yields are vertex pairs $\#a, a\#$ and triples $\#a\#$. Consolidate equal yields, supply missing pairs using their private vertex-block occurrences, and optimally resegment. If S indexes retained triples and $u(S)$ counts uncovered edges, the normalized cost is $10v + 3e + |S| + u(S) \leq r + L$. Adding an endpoint of each uncovered edge gives the required cover. Appendix L.2 proves every normalization step, the exact cost identity (12), and the converse one-root construction. The input forest need not be optimal.

5.3 From rule and length factors to a complexity contradiction

On cubic graphs, $\gamma(w_G) = \frac{29}{2}v + \tau(G) - 1 \leq 30\tau(G)$ and $n \leq 42\tau(G)$; Appendix L.3 derives (13). Suppose a forest algorithm guarantees $r \leq b(n)k$ and $L \leq \ell(n)L^*_{\mathrm{OPE}}(s, k)$. Run it at every budget $0, \ldots, n - 1$ and extract the smallest cover. One query uses the unknown $k = \gamma(w_G)$, where optimum length is one. By Proposition L.2, the cover factor is at most

$$1 + 30(b(n) - 1) + 42\ell(n)/n. \tag{5}$$

This transfer is polynomial and needs no optimum-budget oracle.

Choose a fixed $(1 + \delta)$ inapproximability gap for cubic vertex cover (Alimonti & Kann, 2000). With $\varepsilon_0 = \delta/120$ and $c_0 = \delta/168$, equation (5) becomes $1 + \delta/2$, a contradiction, proving Theorem 5.1. To exclude the scheme in Corollary 5.2(ii), fix any $0 < \varepsilon \leq \varepsilon_0$; its length factor is $o(n)$ regardless of how the polynomial runtime depends on that fixed ε. Appendix L.5 handles sufficiently long inputs, all consequence quantifiers, and the matching rule-free upper bound. Appendix L.6 checks the reduction and extraction with independent finite oracles.

The hard alphabet grows linearly with n. Fixed-alphabet and restricted-instance algorithms, and polylogarithmic length with larger constant budget inflation, remain a concrete frontier. The theorem does not assert that the $O(\log^2 n)$ budget inflation of Corollary 4.3 is necessary. A bounded-error randomized exclusion requires an assumption such as NP $\not\subseteq$ BPP.

6 A COMPUTABLE CERTIFICATE FOR MINIMUM LENGTH

For $2 \le m \le n$, let $p_m(s)$ be the number of distinct length-m factors. The SLP boundary decompositions of Goto et al. (2011) and Bille et al. (2013) charge each factor to a lowest production cut. The forest equivalence makes that fact algorithmically informative for OPE.

Lemma 6.1 (Boundary–substring inequality). *If an* OPE *execution uses r rules and ends in L roots, then for every $2 \le m \le n$,*

$$p_m(s) \le (r + L - 1)(m - 1).$$

Join the L roots with $L - 1$ auxiliary productions. A cut plus its offset in $\{1, \dots, m-1\}$ determines the charged factor, proving the inequality; the full injective charge is in Appendix B.1.

Define, for nonempty s,

$$A_\partial(s) = \max_{2 \le m \le n} \left\lceil \frac{p_m(s)}{m-1} \right\rceil, \qquad \mathrm{LB}_\partial(s,k) = \max\left\{ \left\lceil \frac{n}{2^k} \right\rceil, A_\partial(s) - k + 1 \right\},$$

with $A_\partial = 0$ when $n < 2$ and $\mathrm{LB}_\partial(\varepsilon, k) = 0$.

Theorem 6.2 (Universal length certificate). *For every finite string s and integer $k \ge 0$,*

$$\boxed{\mathrm{LB}_\partial(s,k) \le L^*_{\mathrm{OPE}}(s,k)}.$$

The certificate is deterministically polynomial-time computable.

Proof. In a forest with $r \le k$, every root spans at most 2^r leaves, giving the depth term. Lemma 6.1 gives $L \ge A_\partial - r + 1 \ge A_\partial - k + 1$. Enumerating and deduplicating all factors for every m is polynomial time. Appendix B.1 records the explicit implementation bound; Appendix B.4 proves the baseline's resulting certificate guarantees. □

For pairwise distinct s_N, $N \ge 4$, at $k = 2$ the certificate is exact: $\mathrm{LB}_\partial = L^*_{\mathrm{OPE}} = N - 2 > \lceil N/4 \rceil$. It also strictly improves on depth on fixed-binary de Bruijn families (Appendix B.3). PREFIXFOLD merges one leftmost occurrence per rule and supplies the feasible baseline used in the experiments (Algorithm 2, Appendix B.2). Its a posteriori factor is $L_{\mathrm{PF}}/\mathrm{LB}_\partial$; the decidable $1/\alpha$ class guarantee (6) and exact unary limitation are in Appendix B.4. Appendix B collects the complete computation and calibration details.

7 THE STRUCTURAL ANATOMY OF UTILITY LOSS

Utility $U = n - L$ counts captured original boundaries. The known guarantee is $E \ge O/2$ (Kozma & Voderholzer, 2026); our analysis of their family with $B = E = 5t + 1$ and $O = 8t$ bounds the best-of ratio from above by $5/8$ ((8), Appendix D.2). Whether that bound is attained as a worst-case guarantee and the three related best-of inequalities remain open. Static packing also has unbounded realizability slack even when BPE is optimal: Lemma D.1 and Proposition D.2 give its exact marginals and gap (7) (Appendix D.1).

Our structural results identify the information a stronger utility argument must retain. Figure 3 illustrates the three resolutions; the formal statements and full constructions are in Appendices D–G.

Snapshot fragmentation. After BPE gains b_j, the optimal witness's fragmentation is at most $2b_j$, with exact coefficient two (Theorem D.3, Appendix D.3). A Cycle-3 family has unbounded type fragmentation despite zero packing and EvenOdd slack (Appendix D.4). Thus instantaneous scalar slack cannot account for the lost alignment.

Final overlap and global phase. Common-cut overlap graphs are bipartite caterpillars with exact regret identities (Theorem E.1, Appendix E). Their stars have unbounded local regret. A separate equality family has alternating margins $+2, -2, +2, -2, \dots$, exposing cancellation across components (Proposition F.1, (9), Appendix F). Stronger arguments must preserve that global phase.

Temporal chains. A gain-g rule owns at most $2g$ first-spoiler ancestor chains (Theorem G.1, Appendix G). Each chain can carry unboundedly many missed occurrences; constant event capacity and constant terminal repair both fail. These exact obstructions leave global dynamic and telescoping arguments open and define the structural quantities measured next.

8 Measuring certificates, utility, and model distinctions

The experiments address three questions: how tight is the length certificate, how do merge policies and structural obstructions behave on finite inputs, and what changes when the tokenizer's feasible set changes? Exhaustive grids, stress families, and deterministic public-corpus byte slices provide explicit denominators and replay paths. These data measure the certificate and utility results in Sections 6 and 7; they do not benchmark the universal host's practical approximation tradeoff.

Evidence class	Records	$O > 0$	min B/O	min E/O	min best/O	tie spread
Exact grids	7,819	5,992	5/7	1/2	5/7	253
Stress suite	1,222	289	3/4	1/2	3/4	6
Corpus records	432	144	1	1/2	1	0
Combined registered suite	**9,473**	**6,425**	5/7	1/2	5/7	**259**

Table 2: Registered utility results. Optimum-dependent columns use the 8,252 records with an executed exact-OPE oracle; ratios require $O > 0$ and are observations, not guarantees. Tie spread counts nonzero best-minus-worst BPE utility over all registered records.

Exact grids and policy sensitivity. Across Table 2's 6,425 positive-optimum records, the smallest observed ratios are 5/7 for deterministic BPE, 1/2 for EvenOdd, and 5/7 for their best-of composite. On 8,252 rows with an executed exact OPE oracle, LB_∂/L^* has median 6/7 and is exact in 4,024 rows. The 259 tie-spread records make tie policy measurable. All losing, tied, and zero-utility rows remain in the data. Appendix H.1 specifies the exact oracles and domains (Table 3); Figures 4 and 5 show the cell minima and complete distributions. Figure 6 exposes a one-token tie-policy separation and the measured oracle runtimes. The finite 5/7 best-of minimum is compatible with the asymptotic 5/8 upper family in (8): the declared finite grids and the unbounded family answer different questions.

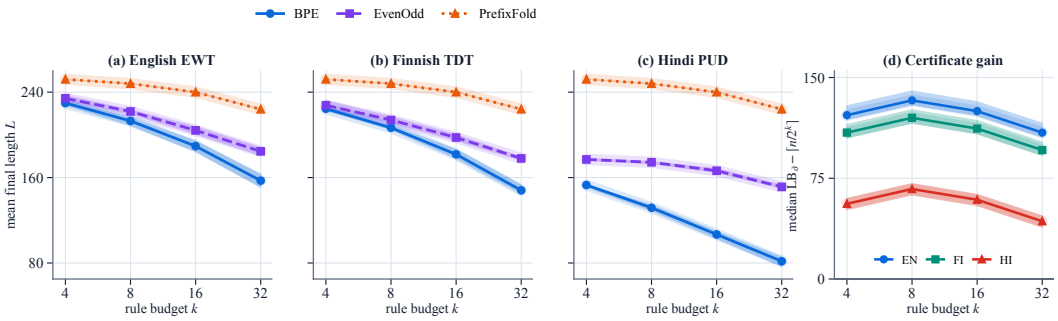

Figure 2: Budget response on deterministic 256-byte windows. Panels (a)–(c) show mean final length with interquartile envelopes over 24 windows per cell; panel (d) shows certificate improvement, positive in all 288 cases. Wide strokes separate curves and encode no uncertainty.

Corpus structure and replay. Figure 2 reports 72 English, Finnish, and Hindi windows at budgets $4, 8, 16, 32$. The certificate strictly improves on depth in all 288 cases (Table 4); Figure 7 separates the language–budget outcomes. Their 12-byte prefixes pass every reconstruction, packing, snapshot, common-cut, and time-expanded identity. All 1,224 final components have zero regret, and snapshot loss and missed-occurrence counts are zero. These nulls locate this sample within the structural theory: the obstruction families prove what can happen, while the checked prefixes show where

these obstructions are absent. Appendix H.2 gives the complete counts and distinguishes exact-prefix checks from full-window measurements.

Model distinctions. Direct vocabulary selection gives strictly shorter length than exact OPE in all 3,988 comparisons under its different feasible set (Appendix H.3); these are not OPE approximation ratios. In the separate two-seed scratch-model pilot, train-only BPE improves validation bits per byte in both seeds and static EvenOdd-derived pairs degrade in both (Table 5, Figure 9, Appendix I). This static adaptation inherits no instance-EVENODD guarantee; the small sample and unequal parameter counts support a diagnostic comparison, not a general downstream-quality conclusion.

Proof checks and replay. The new host and reduction have separate executable finite checks (Appendices K.11 and L.6); their guarantees are established by proof. Independent replay reproduces $18/18$ core empirical artifacts byte-for-byte. Appendix H.4 and Figure 8 describe the manifests and archived evidence map; Appendix H specifies all empirical protocols and denominators.

9 THE FRONTIER EXPOSED BY THE THEORY

The exact forest representation yields a sharp algorithmic distinction: polylogarithmic relaxation of both coordinates is achievable through one universal hierarchy, whereas strict budgets and sufficiently small fixed rule inflation incur linear-order length hardness for deterministic algorithms on growing alphabets unless $\mathrm{P} = \mathrm{NP}$. The one-token/linear-length gap makes this separation explicit, and arbitrary-forest extraction makes it robust to dictionary redesign. The remaining length frontier is fixed alphabets, structured instances, and the gap between small constant and polylogarithmic budget inflation. Our certificate gives instance-level information throughout these regimes.

For utility, improvement beyond $1/2$ and the four best-of conjectures remain open. The sharp fragmentation, overlap, and spoiler-chain laws identify what stronger arguments must aggregate across the hierarchy and time. Exact grids, multilingual corpus windows, and the controlled two-seed pilot provide a reproducible empirical foundation for this structural theory.

REPRODUCIBILITY STATEMENT

The mathematical claims are paired with complete proofs and executable replay artifacts in the supplement. Reproducibility manifests bind the finite evidence to source files, configurations, and SHA-256 digests. The empirical protocols and regeneration records are in Appendix H.4; the algorithm and reduction checks are in Appendices K.11 and L.6.

AI USE STATEMENT

Generative AI assisted related-work retrieval, research and proof development, implementation and analysis, figure preparation, and manuscript drafting and editing. The authors independently verified all sources, mathematical arguments, code, data, and reported results and take full responsibility for the manuscript.

REFERENCES

Paola Alimonti and Viggo Kann. Some APX-Completeness Results for Cubic Graphs. *Theoretical Computer Science*, 237(1–2):123–134, 2000. doi: 10.1016/S0304-3975(98)00158-3.

Philip Bille, Patrick Hagge Cording, and Inge Li Gørtz. Compact q-gram Profiling of Compressed Strings, 2013. URL https://arxiv.org/abs/1304.5373.

Moses Charikar, Eric Lehman, Ding Liu, Rina Panigrahy, Manoj Prabhakaran, Amit Sahai, and Abhi Shelat. The Smallest Grammar Problem. *IEEE Transactions on Information Theory*, 51(7):2554–2576, 2005. doi: 10.1109/TIT.2005.850116.

Jürgen Eckhoff. Extremal Interval Graphs. *Journal of Graph Theory*, 17(1):117–127, 1993. doi: 10.1002/jgt.3190170112.

Zachary Friggstad, Mohammad R. Salavatipour, and Hao Sun. Approximation Algorithms for the Generalized Point-To-Point Problem. In *19th International Symposium on Algorithms and Data Structures (WADS*

2025), volume 349 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 28:1–28:16, 2025. doi: 10.4230/LIPIcs.WADS.2025.28.

Matthias Gallé. Investigating the Effectiveness of BPE: The Power of Shorter Sequences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1375–1381, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1141.

Keisuke Goto, Hideo Bannai, Shunsuke Inenaga, and Masayuki Takeda. Fast q-gram Mining on SLP Compressed Strings, 2011. URL https://arxiv.org/abs/1103.3114.

Artur Jeż. Approximation of Grammar-Based Compression via Recompression, 2013. URL https://arxiv.org/abs/1301.5842.

Artur Jeż. A Really Simple Approximation of Smallest Grammar, 2014. URL https://arxiv.org/abs/1403.4445.

Violeta Kastreva, Philip Whittington, Dennis Komm, and Tiago Pimentel. Tokenisation over Bounded Alphabets is Hard. In *International Conference on Learning Representations*, volume 2026, pp. 36860–36894, 2026. URL https://openreview.net/forum?id=Xhf9YqwlM4.

Akiyoshi Kawamoto, Tomohiro I, Dominik Köppl, and Hideo Bannai. On the Hardness of Smallest RLSLPs and Collage Systems. In *2024 Data Compression Conference (DCC)*, pp. 243–252, 2024. doi: 10.1109/DCC58796.2024.00032.

Dominik Kempa and Nicola Prezza. At the Roots of Dictionary Compression: String Attractors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 827–840, 2018. doi: 10.1145/3188745.3188814.

László Kozma and Johannes Voderholzer. Theoretical Analysis of Byte-Pair Encoding (Full Version), 2024. URL https://arxiv.org/abs/2411.08671v1.

László Kozma and Johannes Voderholzer. Theoretical Analysis of Byte-Pair Encoding. In *34th Annual European Symposium on Algorithms (ESA 2026)*, volume 388 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 80:1–80:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi: 10.4230/LIPIcs.ESA.2026.80. URL https://doi.org/10.4230/LIPIcs.ESA.2026.80.

Jia Peng Lim, Shawn Tan, XianJun Davin Choo, and Hady W. Lauw. A Partition Cover Approach to Tokenization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2237.

Nicola Prezza. String Attractors, 2017. URL https://arxiv.org/abs/1709.05314.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162.

Philip Whittington, Gregor Bachmann, and Tiago Pimentel. Tokenisation is NP-Complete. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28133–28153, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1365.

Vilém Zouhar, Clara Meister, Juan Luis Gastaldi, Li Du, Tim Vieira, Mrinmaya Sachan, and Ryan Cotterell. A Formal Perspective on Byte-Pair Encoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 598–614. Association for Computational Linguistics, July 2023. doi: 10.18653/v1/2023.findings-acl.38.

A FORMAL SEMANTICS AND THE FOREST REPLAY

At time zero the token word is $s_1 \cdots s_n$. Every token occurrence carries a contiguous interval of original positions and a recursively defined yield. A rule chooses an ordered current token type (Y, Z) and a nonempty set of its pairwise nonoverlapping occurrences, replaces precisely that set by a history-wide fresh label X, and records $X \to YZ$. The word order and all unselected occurrences are unchanged. The budget counts rules, not occurrences; the length objective counts final tokens, not the dictionary. Empty selected sets are disallowed.

A creation-ordered forest consists of productions $X_i \to Y_i Z_i$, where each child is terminal or belongs to $\{X_1, \ldots, X_{i-1}\}$, and an ordered multiset of roots. Repeated uses of one production are allowed. Reachability means every production label occurs below at least one root. The decoded frontier is the concatenation of root yields.

Here is a detailed replay proof of Theorem 3.2. For an execution, attach to each current token its derivation tree. A rule adds one parent above each selected sibling pair. The selected intervals are disjoint, so the current token trees remain an ordered forest whose leaves are the original positions in order. Freshness gives creation order. On termination the current tokens are exactly its roots. A used label either is a root or later becomes a descendant of a replacement token, so every production is reachable.

Conversely, fix a reachable creation-ordered forest. After productions $1, \ldots, i-1$ have been replayed, maintain the invariant that the current tokens are the maximal already-created forest nodes together with all uncovered leaves, in frontier order. For $X_i \to Y_i Z_i$, every forest node labelled X_i has its two children as adjacent current tokens. Two nodes with this label cannot be ancestor-related: that would place X_i inside its own expansion, contradicting creation order. They are therefore incomparable and their leaf intervals are disjoint. Selecting precisely these sibling pairs is legal; other current occurrences of (Y_i, Z_i) may be ignored. Reachability makes the set nonempty. Replacing them establishes the invariant for i. After the last production only the prescribed roots remain.

The conventions cover all degenerate cases. The empty string has the unique empty forest with $(r, L) = (0, 0)$. A singleton has no legal nonempty pair rule and has $(0, 1)$. With $k = 0$, the only forest has the n terminal roots. A forest with a positive number of productions cannot decode an empty string. Unreachable productions are never counted: deleting them preserves root yields and creation order.

Finally, each forest occurrence of a production has two adjacent child intervals and therefore a unique original cut. Distinct occurrences created by one replacement are disjoint, so their cuts are distinct. A later replacement only makes an external cut internal and never exposes an internal cut. Induction shows that the union of all production cuts is exactly the set of original boundaries internal to final roots. Hence a forest with L roots has $n - L$ charged cuts, which is its compression utility.

B CERTIFICATE PROOFS, COMPUTATION, AND STRICT FAMILIES

This appendix supplies the algorithm, complete charge, calibration families, and baseline guarantees for Section 6, including Lemma 6.1 and Theorem 6.2.

B.1 COMPLETE BOUNDARY CHARGE

Fix $m \geq 2$ and select the leftmost occurrence of every distinct length-m factor. If such an occurrence crosses a final-root boundary, charge it to the leftmost crossed root boundary and let $a \in \{1, \ldots, m-1\}$ be the number of its characters lying to the left. Otherwise choose the lowest forest node containing the occurrence. Minimality says the occurrence crosses its child cut; charge the production label and the corresponding offset a.

A root boundary and offset determine the occurrence start. A production $X \to YZ$ and offset determine

$$\mathrm{suf}_a(\mathrm{yield}(Y))\,\mathrm{pre}_{m-a}(\mathrm{yield}(Z)),$$

whenever the requested pieces exist. Thus two distinct factors never share a charge. There are $(L-1)(m-1)$ root charges and $r(m-1)$ production-label charges. This proves Lemma 6.1.

Equivalently, joining the roots supplies the $L - 1$ auxiliary SLP cuts used by the known q-gram decomposition (Goto et al., 2011; Bille et al., 2013).

For deterministic computation, enumerate the $n - m + 1$ factors for every $m = 2, \ldots, n$, sort them lexicographically, count distinct adjacent entries, and maximize. A deliberately loose direct representation uses polynomial space and $O(n^3 \log n)$ symbol comparisons; suffix structures are optional.

The remaining edge cases follow directly. At $k = 0$ the depth term equals n, so the certificate is exact. For a singleton it equals one. When $k \geq n - 1$, a left fold reaches one root and both certificate terms are at most one. On a unary string every $p_m = 1$, hence $A_\partial = 1$ when $n \geq 2$.

B.2 Certified feasible baseline

Algorithm 2 PrefixFold: a certified feasible baseline

Input: string $s = s_1 \cdots s_n$ and rule budget $k \geq 0$
Output: legal execution π, final token sequence T, and certified factor γ when $n > 0$

$T \leftarrow (s_1, \ldots, s_n)$; $\pi \leftarrow ()$
for $i = 1, \ldots, \min\{k, n-1\}$ **do**
  let YZ be the leftmost adjacent token occurrence in T
  introduce the fresh production $X_i \to YZ$
  replace *that occurrence only* by X_i in T
  append $(X_i \to YZ,$ selected occurrence$)$ to π
if $n > 0$ **then**
  $\gamma \leftarrow |T|/\mathrm{LB}_\partial(s, k)$ ▷ a posteriori factor
else
  $\gamma \leftarrow 1$ ▷ empty instance is exact
return (π, T, γ)

B.3 All-distinct and de Bruijn families

If $s_N = a_1 \cdots a_N$ has all symbols distinct, then $p_2 = N - 1$, so $A_\partial = N - 1$. At $k = 2$, Theorem 6.2 gives $L \geq N - 2$. Two successive left-prefix merges attain $N - 2$, and $N - 2 > \lceil N/4 \rceil$ for every $N \geq 4$.

For the binary family, an Euler tour in the order-$(q-1)$ de Bruijn graph gives a cyclic binary word containing every order-q word once. Append its first $q - 1$ symbols to obtain d_q with

$$n_q = 2^q + q - 1, \qquad p_q(d_q) = 2^q.$$

Let $k_q = \lceil \log_2(4(q-1)) \rceil$ and $x = 2^q/(q-1)$. Then

$$A_\partial(d_q) \geq \lceil x \rceil, \qquad \left\lceil \frac{n_q}{2^{k_q}} \right\rceil \leq \frac{x}{4} + \frac{5}{4}.$$

For $q \geq 8$, $4(q-1) \leq 2^{q-3}$ gives $k_q \leq q - 3$, while $x > 4q/3$. Therefore

$$\lceil x \rceil - k_q + 1 \geq x - k_q + 1 > \frac{x}{4} + \frac{5}{4},$$

which proves the strict certificate inequality on a fixed binary alphabet.

B.4 PrefixFold and its exact limitation

PrefixFold merges the leftmost adjacent occurrence once per rule (Algorithm 2, Appendix B.2). Its output length is $L_{\mathrm{PF}} = 0$ for $n = 0$ and $L_{\mathrm{PF}} = n - \min\{k, n-1\}$ otherwise. Hence $L_{\mathrm{PF}}/\mathrm{LB}_\partial$ is a valid a posteriori factor. On the decidable class

$$A_\partial(s) \geq k - 1 + \alpha L_{\mathrm{PF}}(s, k), \qquad \alpha \in (0, 1], \tag{6}$$

PrefixFold is a $1/\alpha$ approximation.

Condition (6) is decidable from the input and turns the returned certificate into a formal $1/\alpha$ guarantee. This is an explicit certified class, not a universal approximation theorem.

At every step PREFIXFOLD chooses the single pair at current position zero. The selection is nonempty and nonoverlapping, and the fresh parent becomes the new prefix token. Hence it performs exactly $\min\{k, n-1\}$ rules when $n > 0$, proving the length formula above. Since $\mathrm{LB}_\partial \le L^*_{\mathrm{OPE}}$, division gives its a posteriori factor. If $A_\partial \ge k - 1 + \alpha L_{\mathrm{PF}}$, then $\mathrm{LB}_\partial \ge \alpha L_{\mathrm{PF}}$, proving the $1/\alpha$ class guarantee.

For $s = a^{2^k}$, $k \ge 1$, the rules $X_1 \to aa$ and $X_i \to X_{i-1}X_{i-1}$, applied fully, reach one root in k steps. Thus $\mathrm{LB}_\partial = L^*_{\mathrm{OPE}} = 1$, while the singleton-selection policy reduces length by only one per rule and returns $2^k - k$. This is an exact obstruction to a universal factor for this certificate/algorithm pair.

C HARDNESS ALIGNMENT AND WEIGHTED-ROUTE DETAILS

This appendix supports the prior-work comparison in Section 2 and the scalar and algorithmic distinctions in Section 4. It keeps the general weighted conversion and sharper weight-two bridge separate from the fixed endpoint selector's obstruction.

C.1 KNOWN HARDNESS AND THE AT-MOST-BUDGET ALIGNMENT

The full arXiv version of Kozma & Voderholzer (2024), Section 2 and Appendix A, gives a growing-alphabet construction for a cubic N-vertex MAX-CUT instance with budget $k = N + 1$ and exact optimum length

$$87N - \mathrm{cut}^*(G).$$

The counts in this paragraph refer specifically to arXiv v1: input length $121N$ and optimum utility $34N + \mathrm{cut}^*(G)$. The primary result uses an exact rule count. On this family it equals our at-most optimum. The constructed string contains $35N$ separator symbols, each occurring once. Call a current token marked when its yield contains a separator. A token type containing a fixed unique separator occurs at most once, so one partial rule reduces the number of marked tokens by at most one. After $r \le N$ rules at least $35N - r \ge 34N$ marked tokens remain, and hence at least two tokens remain. If an at-most-k optimum used $r < k$ rules, one more singleton occurrence merge would legally shorten it, a contradiction. Thus every optimum uses all $N + 1$ rules and the published identity transfers. This is a model alignment, not a new reduction; it gives APX-hardness and no PTAS unless P = NP, but not length membership in APX and not fixed-alphabet hardness.

The adjacent corpus results remain separate. Kastreva et al. prove a binary bottom-up corpus gap ruling out ratios below 1.0000001 unless P = NP (Kastreva et al., 2026). The conference version labels its theorem APX-complete, whereas later arXiv v2 states APX-hardness and leaves membership open. We retain only the common stable claim: APX-hardness/no PTAS for that corpus, yield-identified, full-merge model. Concatenating documents creates mergeable cross-document pairs, so it does not produce a one-string fresh-label OPE theorem.

C.2 GENERAL WEIGHTED-TO-BUDGETED CONVERSION

Theorem C.1 (Weighted-to-budgeted conversion)**.** *Suppose one uniform algorithm, on every nonempty length-n string and positive dyadic λ, returns in polynomial time a reachable forest of cost at most $c(n)\mathrm{WPF}_\lambda(s)$, for a known polynomial-time evaluable $c(n) \ge 1$. Then there is a polynomial-time OPE algorithm with*

$$\boxed{r < 3c(n)k, \qquad L < 3c(n)L^*_{\mathrm{OPE}}(s,k)}.$$

After rounding, this is a $(\lceil 3c(n)\rceil, 3c(n))$ bicriteria guarantee; empty, zero-budget, singleton, and $k \ge n - 1$ instances are solved exactly.

Algorithm 3 Dyadic Pareto sweep for the weighted-to-budgeted conversion

Input: nontrivial (s, k), $n = |s|$, and weighted $c(n)$-approximation $\mathcal{A}(\cdot, \lambda)$
Output: reachable forest satisfying the bicriteria bounds of Theorem C.1

$\mathcal{C} \leftarrow \varnothing$
for $j = -\lceil \log_2 n \rceil, \ldots, \lceil \log_2 n \rceil$ **do**
  $\mathcal{F}_j \leftarrow \mathcal{A}(s, 2^j)$
  if $r(\mathcal{F}_j) < 3c(n)k$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathcal{F}_j\}$
return $\arg\min_{\mathcal{F} \in \mathcal{C}} L(\mathcal{F})$

Proof. Let $L^* = L^*_{\mathrm{OPE}}(s, k)$ and $\rho = L^*/k \in [1/n, n]$. The grid contains $\rho \le \lambda < 2\rho$. Comparing with an optimal at-most-k forest gives

$$\lambda r + L \le c(n)(\lambda k + L^*) < 3c(n)L^*.$$

Dropping either nonnegative term yields both bounds and proves the filter nonempty. Theorem 3.2 converts the forest without changing (r, L); Appendix C checks rounding and edge cases. □

The uniform premise is now supplied by Corollary 4.3; its direct closed-prefix argument gives the sharper (ρ_n, β_n) bounds.

C.3 Scalarization arithmetic and edge cases

For a nonempty input, the no-production forest proves $\mathrm{WPF}_\lambda(s) \le n$, while every forest has cost at least one. This is the uniform weighted n-approximation; the analogous rule-free length bound calibrates Section 5. For the conditional conversion, the nontrivial regime $1 \le k \le n - 2$ implies $1 \le L^* \le n$ and hence $L^*/k \in [1/n, n]$. The stated dyadic grid contains the smallest power of two not below this ratio. Its strict upper inequality $\lambda < 2L^*/k$ yields

$$c(n)(\lambda k + L^*) < 3c(n)L^*.$$

After filtering, the integral rule allowance may be written $r \le \lceil 3c(n)k \rceil - 1$, and therefore as augmentation factor $\lceil 3c(n) \rceil$. None of these roundings returns the budget to k.

The exact branches are: $s = \varepsilon$ gives no roots; $k = 0$ leaves all tokens; $n = 1$ needs no rule; and $k \ge n - 1$ permits a left fold to one root, which is optimal. These branches also avoid dividing by zero in L^*/k.

C.4 Exact forest–grammar conversion

Theorem C.2 (Forest–grammar bridge)**.** *Let $g^*(s)$ be the minimum total right-hand-side size of an acyclic grammar generating a nonempty length-n string s. Then*

$$g^*(s) \le \mathrm{WPF}_2(s) \le 2g^*(s), \qquad c_2(s) = O\left(1 + \log \frac{n}{\mathrm{WPF}_2(s)}\right),$$

where $c_2(s)$ is the approximation factor of a polynomial-time algorithm. Every output forest reverse replays to a legal OPE execution with the same (r, L).

We prove Theorem C.2 under the classical size convention of Charikar et al. (2005): an acyclic grammar has one rule per nonterminal and size equal to the total number of symbols on all right-hand sides. Empty and unreachable material can be deleted, unit nonterminal rules can be inlined, and a secondary rule $A \to a$ can be identified with terminal a; none of these operations increases size.

First take a reachable forest F with productions $X_i \to Y_i Z_i$ and ordered roots $R_1 \cdots R_L$. Add these productions to a grammar and introduce a fresh start rule

$$S \to R_1 \cdots R_L.$$

Creation order makes the grammar acyclic, forest reachability removes useless rules, and its size is exactly $2r(F) + L(F)$. Consequently $g^*(s) \le \mathrm{WPF}_2(s)$.

Conversely, let a reduced acyclic grammar G of size g have start right-hand side of length L. Keep that right-hand side as the ordered root sequence. For every secondary rule $A \to Z_1 \cdots Z_\ell$ with $\ell \geq 2$, replace its right-hand side by a left-associated binary chain of $\ell - 1$ productions, using A as the final label. Topologically order the original and introduced labels. The resulting binary dictionary is reachable from the roots and therefore is a creation-ordered forest; Theorem 3.2 also supplies its selected-occurrence replay. It has

$$r = \sum_{A \neq S} (|\operatorname{rhs}(A)| - 1), \qquad 2r + L \leq 2g - L \leq 2g.$$

Taking G optimal proves $\mathrm{WPF}_2(s) \leq 2g^*(s)$. Under the binary-secondary, explicit-start-sequence normal form, forests and grammars are the same object up to label renaming and the cost is exactly $2r + L$. If a one-root variable is itself designated as the start symbol, the convention changes the displayed cost by at most one.

Finally, Jeż (2014) construct in polynomial time a grammar G_J of size

$$|G_J| = O(g^*(s)(1 + \log(n/g^*(s)))) .$$

Normalizing G_J as above produces a reachable forest of cost at most $2|G_J|$. Since $g^* \leq \mathrm{WPF}_2 \leq 2g^*$, $\log(n/g^*) \leq 1 + \log(n/\mathrm{WPF}_2)$, proving the algorithmic part of Theorem C.2. This transfer uses the output grammar; it does not need Jeż's internal execution to coincide with an OPE trace. Theorem 3.2 constructs that trace after normalization.

C.5 Three exact black-box mismatches

A direct token for `0011` compiles to three binary productions, not one; at budget one direct segmentation has length one but OPE has length three. A size-one attractor for a^{16} describes substring occurrence coverage, not a binary derivation; one OPE rule gives length eight. On an all-distinct length-n word, every merge is used once, so ordinary grammar cost $2r + L$ is minimized by no rules at value n, yet the guarantee of a 2-approximation permits a complete $n - 1$-rule left fold of cost $2n - 1$. The last example is not a reachability separation: it proves that scalar grammar cost, even when it equals WPF_2, does not control the two coordinates of the requested (r, L) Pareto curve or enforce a strict rule budget.

C.6 Fixed $\mathcal{H}_J/\mathcal{A}_J$ construction

The following specification fixes the completed-phase selector whose gap is distinct from the universal host in Algorithm 1.

Proposition C.3 (Diverging endpoint-selector gap)**.** *For $m \geq 2$, set $t = 2^m$, $s_m = a^{t^2} b_1 \cdots b_t$ with all symbols distinct except a, and $\lambda = t$. Then*

$$\frac{\mathrm{cost}(\mathcal{A}_J(s_m, t))}{\log_2(2 + |s_m|)\mathrm{WPF}_t(s_m)} \geq \frac{t}{2(2m+1)(2m+2)} \longrightarrow \infty.$$

The specification is as follows. At the first unparsed input position p, the greedy overlapping LZ parse chooses the largest $\ell \geq 2$ for which $s[p : p + \ell - 1]$ has an equal source $s[j : j + \ell - 1]$ with $j < p$; source–target overlap is allowed, and ties choose the smallest j. If no such factor exists, it emits the single letter $s[p]$. Each phase then runs the published PAIRING and PAIRREPLACEMENT scans of Jeż (2014), including their factor-source and `newpos` updates, with the normalization below.

Algorithm 4 Fixed recompression hierarchy $\mathcal{H}_J$ and selector $\mathcal{A}_J$

Input: explicit nonempty string s and positive dyadic weight λ
Output: a reachable endpoint forest minimizing cost over recorded phases

build the overlapping greedy LZ parse above; set $w_0 \leftarrow s$, $h \leftarrow 0$
record endpoint F_0 with roots w_0 and no productions
while $|w_h| > 1$ **do**
    run published PAIRING fully on the current factor representation
    let I be positions marked `first`; their two-token intervals are disjoint
    group I by ordered *pre-phase* child-label pair
    order groups by leftmost selected occurrence
    for each ordered nonempty group G_u **do**
        create fresh $X_{h+1,u} \to ab$ for the group's pair (a, b)
        replace exactly the occurrences in G_u by $X_{h+1,u}$
    complete published PAIRREPLACEMENT source/position updates
    set $h \leftarrow h + 1$ and let w_h be the complete phase output
    prune productions unreachable from the ordered roots w_h; record F_h
return the earliest F_j minimizing $\lambda r(F_j) + L(F_j)$

Every occurrence of one ordered pre-phase pair receives the same phase-local label, different pairs and phases receive distinct labels, and only complete phase endpoints are candidates. The selector never receives or queries an optimal comparator.

For legality, contracting one marked group cannot destroy the children inside another disjoint marked interval. All children predate the phase, and partial replacement need not select newly created cross-boundary occurrences. Induction preserves the decoded yield; pruning preserves it again. For runtime, a naive exact greedy parse is already polynomial, the pairing scan is linear per current word, and the source shrinkage $N_h - 1 \le (2/3)(N_{h-1} - 1)$ gives $O(\log n)$ phases. Every new rule accounts for at least one contraction, so at most $n - 1$ rules are created. For $\lambda = a2^e$, compare endpoint costs by first checking the bit-length orders of the integer coefficients and $a2^e$; shift only when their orders overlap, as in Appendix K.8. This remains polynomial when the signed exponent is binary encoded.

For Proposition C.3, write $T = t^2$. The exact greedy parse of $a^T b_1 \cdots b_t$ is the free first a, the factor a^{T-1} sourced one position left, and the t new suffix letters. The pairing repair marks positions $1, 2$ and copies an alternating pairing through the even a block. It then marks exactly

$$(b_1, b_2), (b_3, b_4), \ldots, (b_{t-1}, b_t).$$

The suffix symbols are distinct, so phase one creates $t/2$ distinct rules. Each corresponding node either stays a root or becomes a descendant of a later root, so all remain reachable at every positive endpoint. Hence

$$tr_h + L_h \ge t^2/2 + 1 \quad (h \ge 1), \qquad \mathrm{cost}(h = 0) = t^2 + t.$$

The comparator

$$X_1 \to aa, \qquad X_j \to X_{j-1}X_{j-1} \ (2 \le j \le 2m)$$

has roots $(X_{2m}, b_1, \ldots, b_t)$, so $r = 2m$, $L = t + 1$, and cost $(2m + 1)t + 1 \le (2m + 2)t$. Finally $\log_2(t^2+t+2) \le 2m+1$. Combining the bounds gives exactly the divergence ratio in Proposition C.3. The assumption $m \ge 2$ ensures the even suffix pairing and all displayed comparisons in the exact family.

D PACKING, THE COMPOSITE UPPER FAMILY, AND SNAPSHOT FRAGMENTATION

This appendix proves the packing and best-of statements in Section 7, Theorem D.3, and the zero-slack fragmentation construction used to interpret its sharp bound.

Figure 3 gives the common geometric picture for this appendix and Appendices E–G.

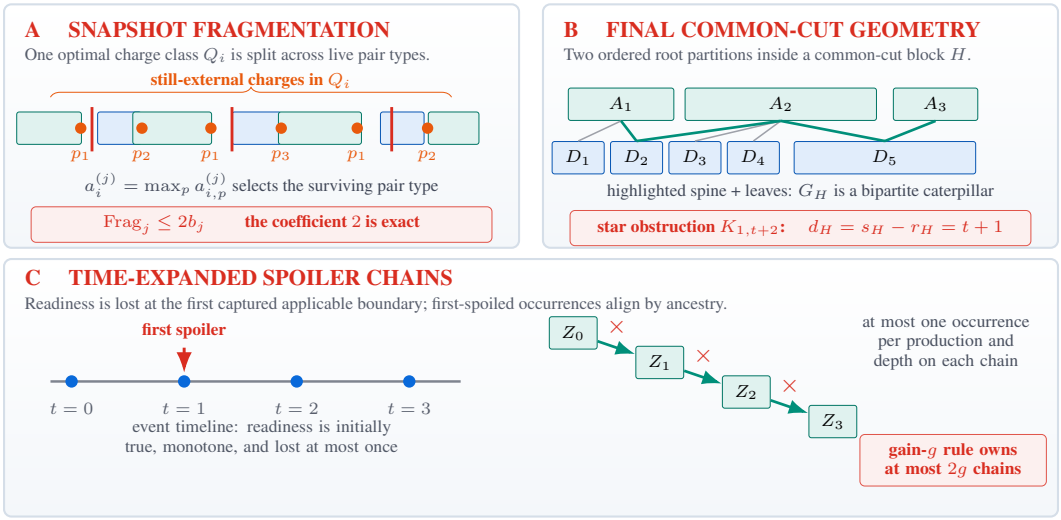

Figure 3: One utility gap viewed at three exact resolutions. Snapshot fragments live charge classes; final common cuts induce overlap caterpillars; time expansion assigns first spoilers to ancestor chains.

D.1 Packing marginals and the hierarchy gap

Index the $n-1$ original boundaries. Each internal occurrence joins its two children across one original production cut; each final pair of roots is separated by a root boundary. A performed replacement makes one previously external boundary internal, permanently. Thus utility $U = n - L$ is exactly the number of distinct charged boundaries.

Let $P_k(s)$ be the maximum number of original boundaries covered by at most k token-disjoint packings, each using one original pair type.

Lemma D.1 (Exact packing marginals). *For nonloop xy, include the marginal $f(xy)$. If $r_1, \dots, r_q$ are maximal runs of occurrence starts of a loop xx, include*

$$A_{xx} = \sum_j \left\lceil \frac{r_j}{2} \right\rceil, \qquad C_{xx} = \sum_j \left\lfloor \frac{r_j}{2} \right\rfloor \quad (C_{xx} > 0).$$

Then P_k is the sum of the k largest positive marginals, or all marginals if fewer than k exist.

The disjoint-path calculation below proves this closed form and exposes the missing constraint: static boundary packings need not share one creation-ordered hierarchy.

Proposition D.2 (Unbounded realizability slack). *For $s_m = (ab)^m a$, $m \ge 2$, and $k = 2$,*

$$P_2 = 2m, \qquad O = B = m + \lfloor m/2 \rfloor, \qquad E = m, \qquad P_2 - O = \lceil m/2 \rceil. \tag{7}$$

The realizability slack is unbounded even though deterministic BPE is optimal on the family. Packing is therefore an exact relaxation, not a surrogate for the forest.

UNBOUNDED SLACK **Static packing overestimates the best reachable hierarchy by $\lceil m/2 \rceil$ even on a family where deterministic BPE itself is optimal.**

Two occurrences of a nonloop pair xy, $x \neq y$, cannot begin at consecutive boundaries; all $f(xy)$ occurrences therefore form one packing. Occurrences of xx form disjoint paths on consecutive start positions. Alternating vertices on every path give two disjoint independent sets of total sizes A_{xx} and C_{xx} from Lemma D.1, covering the path, and no third positive component is useful. Boundary indices of different original pair types are disjoint. Hence the problem is simply to choose the k largest positive component marginals; if a choice included C_{xx} without A_{xx}, swapping in A_{xx} would not decrease its value, so the stated closed form is exact.

For $s_m = (ab)^m a$, the ab and ba packings cover all $2m$ boundaries, giving $P_2 = 2m$. Full $ab \to X$ has gain m and produces $X^m a$; full $XX \to Y$ then gains $\lfloor m/2 \rfloor$. For the converse, by reversal it suffices to let a first rule replace r of the m ab blocks. A second rule can gain at most one of

$$m - r, \quad \lfloor r/2 \rfloor, \quad m - r + 1.$$

For $m \geq 2$ each is at most $m - r + \lfloor m/2 \rfloor$, so total utility is at most $m + \lfloor m/2 \rfloor$. First-occurrence BPE takes the displayed two rules. EvenOdd selects all $2m$ consecutive starts and retains exactly m. This proves Proposition D.2 for its stated domain.

D.2 Exact five-eighths upper family

The known unconditional guarantee is $E \geq O/2$ (Kozma & Voderholzer, 2026). The composite $\max\{B, E\}$ looks stronger. We analyze the BPE upper-bound construction of Kozma & Voderholzer (2026), Section 3: $s_t = (Aq)^t \texttt{aa}$ for integer $t \geq 1$, $k = 4$, with $A = \texttt{abaacaaba}\, p\, \texttt{aca}$ and five distinct symbols. We establish the exact values

$$B = E = 5t + 1, \qquad O = 8t. \tag{8}$$

Thus $2/3$ already fails at $t \geq 4$ and the worst-case ratio is at most $5/8$. The argument below proves the exact utilities, including the optimal upper bound. Whether $5/8$ is exact remains open, as do $B + E \geq 5O/4$, $4(B + E) \geq P + 4O$, and, for $k = 2$, $\max\{B, E\} \geq 3O/4$. We attack this frontier at three resolutions.

In $A = \texttt{abaacaaba}\, p\, \texttt{aca}$, each of ab, ba, aa, ac, ca occurs twice. On $s_t = (Aq)^t \texttt{aa}$, the terminal pair makes $f(aa) = 2t + 1$, while the other four counts are $2t$. Thus the first BPE rule is uniquely $aa \to X$ and has gain $2t + 1$. One block becomes $\texttt{abXcXba}\, p\, \texttt{aca}$; every within-block current pair then occurs t times, and first-occurrence BPE takes three block-prefix pairs of gain t. Therefore $B = 5t + 1$.

For EvenOdd, the selected starts in each length-14 block have offsets $0, 1, 2, 3, 5, 6, 7, 10$, with the terminal aa start after the blocks. Global rank parity retains offsets $0, 2, 5, 7, 10$ in each block and the terminal start, so $E = 5t + 1$.

The four rules

$$ac \to R, \quad Ra \to S, \quad ab \to T, \quad Ta \to V$$

compress two aca and two aba macros per block and achieve utility $8t$. The four largest raw frequencies sum to $8t + 1$. Equality would force one packing to contain every aa boundary and three more to contain every boundary of three types among ab, ba, ac, ca. The rule capturing the terminal aa must be the literal rule $aa \to X$, must capture all $2t + 1$ such boundaries, and must occur first: any preceding full equality-packing rule would consume an endpoint of an internal aa. After that rule the maximum current pair frequency is t, and a partial replacement with r occurrences cannot increase this maximum (all new pair types involve one of the r fresh tokens). Three remaining rules add at most $3t$, contradicting utility $8t + 1$. Integrality yields $O = 8t$. Thus the deterministic best-of ratio is $(5t + 1)/(8t)$ and approaches $5/8$ only from above.

D.3 Proof of the snapshot theorem

Fix an optimal forest with charge classes $Q_1, \ldots, Q_r$ and a deterministic BPE prefix of utility b_j. Let C_j be the captured original boundaries. For current pair type p, let $a_{i,p}^{(j)}$ be the largest token-disjoint subset of still-external charges in Q_i currently realizing p, and put

$$a_i^{(j)} = \max_p a_{i,p}^{(j)}, \qquad \mathrm{Frag}_j = \sum_{i=1}^{r} \left(|Q_i \setminus C_j| - a_i^{(j)} \right).$$

Theorem D.3 (Sharp snapshot bound). *For every fixed optimal witness and deterministic BPE prefix,*

$$\mathrm{Frag}_j \leq 2b_j, \qquad g_{j+1} \geq \max\{0, (O - 3b_j)/k\} \quad (j < k).$$

The coefficient 2 is exact.

This recurrence reaches $(1 - e^{-3})/3$. The separate construction in Appendix D.4 shows that even zero packing and EvenOdd slack permits one optimal rule to split into unboundedly many current pair types, so scalar slack cannot pay fragmentation instantaneously.

For each current type p, let $\alpha_{i,p}^{(j)}$ be the maximum token-disjoint subset of still-external charges in Q_i presently having type p, and $a_i^{(j)} = \max_p \alpha_{i,p}^{(j)}$. Full replacement of a nonloop type takes all its occurrences; on a loop run it takes a maximum matching. Thus the next greedy gain is at least $\max_i a_i^{(j)}$.

Call a remaining charge untouched when its two incident current tokens are singleton original symbols. All untouched charges of Q_i retain the same original pair type and are token-disjoint, so they contribute to $a_i^{(j)}$. Every other remaining charge touches a nonsingleton token. Assign it to one such incident token. The optimal charge classes are globally disjoint, and a current token has only two exterior boundaries, so it receives at most two assignments. Each nonsingleton token contains at least one of the b_j captured boundaries, and distinct current tokens contain disjoint captured sets; hence there are at most b_j of them. This proves $\mathrm{Frag}_j \le 2b_j$.

Let $h_j = |Q \cap C_j| \le b_j$. Since

$$\sum_i a_i^{(j)} = O - h_j - \mathrm{Frag}_j,$$

averaging over $r \le k$ production classes yields

$$g_{j+1} \ge \max\left\{0, \frac{O - h_j - 2b_j}{k}\right\} \ge \max\left\{0, \frac{O - 3b_j}{k}\right\}.$$

To see exactness of coefficient two, take $m \ge 3$, common symbols a, b, pairwise distinct u_i, v_i, six private guards $g_{i,0}, \ldots, g_{i,5}$ per cell, and a final guard. Concatenate

$$[g_{i,0}u_iabv_ig_{i,1}][g_{i,2}u_iag_{i,3}][g_{i,4}bv_ig_{i,5}]$$

for $i = 1, \ldots, m$. Here $n = 14m + 1$ and take $k = n - 1$. Pair ab occurs m times, while each u_ia and bv_i occurs twice and every other type once, so the first greedy rule is uniquely $ab \to X$ with $b_1 = m$. An optimal one-root forest first applies the $2m$ rules u_ia and bv_i to both occurrences, then completes with singleton rules. Each of these $2m$ classes loses one unit of current capacity after $ab \to X$, while later singleton classes lose none. Thus

$$\mathrm{Frag}_1 = 2m = 2b_1, \qquad P = O = B = n - 1, \qquad E = (n - 1)/2.$$

D.4 One class fragments into unboundedly many types

For even $t \ge 2$, set $q = 2t + 3$ and

$$s_t = \prod_{i=0}^{t-1} (0^{q2^i}1), \qquad n = q(2^t - 1) + t, \qquad k = n - 1.$$

A one-root creation-ordered forest begins with

$$A \to 00, \quad C \to A1,$$

builds a unary base N_0 of yield 0^{q-2} by a chain, doubles it via $D_i \to N_iA$ and $N_{i+1} \to D_iN_i$, and makes $H_i \to N_iC$ before left-folding the H_i. It uses $6t - 2 \le k$ rules and decodes s_t. The production $C \to A1$ occurs at every block end and charges all t terminal 01 boundaries.

Two alternating 00 packings together with the 01 and 10 packings cover all boundaries. With budget $n - 1$, both optimum OPE and BPE reach one root, and EvenOdd retains half the starts; hence

$$(P, O, B, E) = (n - 1, n - 1, n - 1, (n - 1)/2).$$

Put $Z_0 = 0$ and let Z_{j+1} be the result of Z_jZ_j. Before round $j + 1$ the full score of Z_jZ_j is

$$\sum_{i=0}^{t-1} \left\lfloor \frac{q2^i}{2^{j+1}} \right\rfloor > t \quad (j < t - 1),$$

and is at least $t + 1$ at $j = t - 1$; every other current type has score at most t. Thus these t choices are uniquely forced. Since the 2-adic valuation of $q2^i$ is i, the token before block i's final 1 is Z_i. The single class therefore becomes $(Z_0, 1), \ldots, (Z_{t-1}, 1)$, all distinct and of score one, while both slacks $P - O$ and $2E - P$ are zero.

E COMMON-CUT COMPONENTS AND CATERPILLAR REGRET

We prove Theorem E.1 and the unbounded-star construction stated in Section 7.

Let Q, C be the original boundaries captured by optimal OPE and BPE. Split at every common cut. In block H, join intersecting optimal and BPE root intervals; write their counts as r_H, s_H.

Theorem E.1 (Common-cut caterpillar identities)**.** *Every overlap graph G_H is a bipartite caterpillar tree and*

$$|(Q \setminus C) \cap H| = s_H - 1, \quad |(C \setminus Q) \cap H| = r_H - 1,$$
$$d_H = s_H - r_H = 1 + \sum_A (\deg_H A - 2) = -1 - \sum_D (\deg_H D - 2).$$

Each actual merge captures one previously external original boundary and no captured boundary becomes external. Therefore the optimal and BPE final captured sets are precisely Q and C, with cardinalities O and B. A boundary outside $Q \cup C$ is cut by both final partitions; split at all such common cuts.

Fix one block H. Scan it from left to right while recording the current optimal root A and BPE root D. The first nonempty intersection gives the first edge of G_H. At an exclusive cut exactly one root changes. Ordered interval partitions never return to an earlier root, so the next intersection attaches one new vertex to the graph already scanned. The scan visits every root and every nonempty intersection. Hence G_H is connected, and every edge after the first adds one vertex: it is a tree.

During a maximal run of boundaries in $Q \setminus C$, one optimal root persists while two or more BPE roots are crossed, so that optimal root is nonleaf. Conversely, every nonleaf optimal root contains such a run. The symmetric correspondence holds for runs in $C \setminus Q$ and nonleaf BPE roots. The two types of maximal runs alternate, and consecutive run vertices intersect. Thus the subgraph induced by all nonleaves is a possibly empty path; all other vertices are leaves. This is the direct caterpillar proof specialized to the two root partitions.

An optimal root of degree d meets d consecutive BPE roots and contains exactly their $d-1$ intervening cuts, all in $Q \setminus C$. Different optimal roots contain disjoint cuts. Summing proves

$$|(Q \setminus C) \cap H| = \sum_{A \in A_H} (\deg_H(A) - 1) = s_H - 1.$$

Swapping partitions gives the second identity in Theorem E.1. Their difference is $s_H - r_H$. Since a tree on $r_H + s_H$ vertices has $r_H + s_H - 1$ edges,

$$1 + \sum_{A \in A_H} (\deg_H(A) - 2) = 1 + (r_H + s_H - 1) - 2r_H = s_H - r_H,$$

and the symmetric degree formula follows. Finally,

$$\sum_H (s_H - r_H) = |D| - |A| = (n - B) - (n - O) = O - B.$$

The classical generic interval-graph antecedent is Eckhoff's characterization in *Extremal interval graphs* (Eckhoff, 1993); the scan and the exact charge/regret identities above are the pair-encoding specialization.

For the star obstruction, take the exact growing-alphabet length family for $t > 2$:

$$s_t = \left(\prod_{i=1}^{t} x_i a a y_i\right)\left(\prod_{i=1}^{t} \delta_i x_i a\right)\left(\prod_{i=1}^{t} \varepsilon_i a y_i\right), \qquad k = 8t - 1,$$

with all indexed symbols distinct. The rules $x_i a \to X_i$ and $a y_i \to Y_i$, followed by a binary completion, reach one optimal root, so $O = 10t - 1$. The deterministic first-occurrence BPE trace first merges the t copies of aa and then extends a left prefix by $8t - 2$ unit gains, ending in $t + 2$ roots and utility $B = 9t - 2$. Since the optimal partition has one root, there is no common cut and its root intersects all final BPE roots:

$$G_H = K_{1,t+2}, \qquad |Q \setminus C| = t + 1, \qquad |C \setminus Q| = 0, \qquad d_H = t + 1.$$

The alphabet grows with t. This exact star falsifies constant per-component regret but says nothing about global cancellation or a general local-algorithm lower bound.

F THE PHASED EQUALITY FAMILY

We prove Proposition F.1 and its exact alternating local margins from Section 7.

If F is the total original frequency of pair types selected by EVENODD before sparsification, the following exact family captures the global phase effect.

The open global inequalities referred to below are stated in Appendix D.2.

Proposition F.1 (Alternating-margin equality family)**.** *For $s_t = (\texttt{cbbabbbab})^t$, $t \geq 1$, and $k = 3$,*

$$(P, O, B, E, F) = \left(6t, 6t, 4t, \left\lceil \frac{7t}{2} \right\rceil, 7t\right). \tag{9}$$

For even t, both open global inequalities hold with equality, yet the stronger weights $4(O - B) \leq 4E - F$ and $2(O - B) \leq E$ fail. Successive common-cut blocks have exact local margins $+2, -2, +2, -2, \ldots$ under the global parity. The cancellation is not an impossibility theorem; it proves that phase is a global resource (Appendix F).

Let $w = \texttt{bbab}$ and $s_t = (\texttt{c}ww)^t = (\texttt{cbbabbbab})^t$, with $k = 3$. The original frequencies are

$$f(bb) = 3t, \quad f(ba) = f(ab) = 2t, \quad f(cb) = t, \quad f(bc) = t - 1.$$

The bb starts consist in each block of one isolated start and one run of two. Its packing marginals are therefore $2t$ and t. The three largest overall marginals are $2t$ from the first bb component and $2t$ each from ba and ab, proving $P_3 = 6t$.

The creation-ordered rules

$$A \to bb, \qquad D \to Aa, \qquad G \to Db$$

apply to the initial bb in each of the $2t$ copies of w and then to all constructed copies. Their gains are $2t, 2t, 2t$, with final roots $(c, G, G)^t$. Thus $O \geq 6t$, while $O \leq P$ gives $O = 6t$.

Initially bb, ba, ab tie at full gain $2t$; the first occurrence belongs to bb. Full replacement changes each block to c, A, a, A, b, a, b and gains $2t$. Every within-block type thereafter has gain t, while the cross-block type has gain $t - 1$. First-occurrence tie-breaking chooses cA and then the new prefix token followed by a, each with gain t. Hence $B = 4t$.

EVENODD selects exactly bb, ba, ab, whose union is a run of seven starts in each block. Successive runs are separated by an unselected boundary. The global parity filter retains every other selected rank, so

$$F = 7t, \qquad E = \left\lceil \frac{7t}{2} \right\rceil.$$

Substitution proves Proposition F.1. In particular,

$$4(B + E) - P - 4O = 4(B + E) - 5O = \begin{cases} 0, & t \text{ even}, \\ 2, & t \text{ odd}. \end{cases}$$

For even t, however,

$$4(O - B) = 8t > 7t = 4E - F, \qquad 2(O - B) = 4t > 7t/2 = E.$$

With zero-based blocks $j = 0, \ldots, t - 1$, the final optimal charges are $Q_j = 9j + \{1, 2, 3, 5, 6, 7\}$, while BPE captures $C_j = 9j + \{0, 1, 2, 4\}$. Every block is one common-cut component, its exclusive event word is $\texttt{CQCQQQ}$, and its signed regret is two. The optimal packing contributes six starts per block. EVENODD retains four selected starts in even-indexed blocks and three in odd-indexed blocks because its rank parity is global. Consequently the local margin for $4(B + E) \geq P + 4O$

$$4E_j - 6 - 4 \cdot 2$$

is $+2, -2, +2, -2, \ldots$. Consecutive blocks cancel exactly. This disproves an intrinsic componentwise restriction and the two stronger global weights; it neither disproves the two global inequalities in Appendix D.2 nor rules out explicit cross-component transfer.

G READINESS, LABELLED SPOILER CHAINS, AND UNBOUNDED FIBERS

We prove Theorem G.1 and both unbounded-capacity families summarized in Section 7.

An optimal occurrence is partition-ready when both child intervals are unions of current BPE cells. Its first captured outer boundary, if any, is its *first spoiler*.

Theorem G.1 (Readiness and spoiler chains)**.** *Readiness is initially true, monotone, and lost exactly at the first captured applicable boundary. On either side of one replacement event, first-spoiled optimal occurrences form one ancestor chain containing at most one occurrence of each production and depth. A gain-g rule therefore owns at most $2g$ such chains.*

Let D_j be the partition into current BPE token intervals after j rules. For an optimal occurrence v, both child intervals are unions of D_j cells exactly when no current cell crosses either outer endpoint and the child boundary remains a cut. This is equivalent to retaining every applicable member of

$$\{\ell(v) - 1, q(v), r(v) - 1\}.$$

At a string endpoint omit the nonexistent outer boundary. Initially all cells are singletons. Later replacements only coarsen the partition, so partition readiness is monotone. Its first loss is exactly the event that captures the first boundary in this set. For a missed occurrence $q(v)$ is never captured; only its left or right outer boundary can be a first spoiler. If neither is captured, the occurrence stays partition-ready at termination. Exact readiness additionally requires the two children themselves to be adjacent cells and need never occur.

The rule-time exchange graph has missed optimal occurrences on the left and BPE rule times plus a terminal vertex $\perp$ on the right. A finite edge records the first-spoiling time; an unspoiled occurrence goes to $\perp$. The event-expanded graph replaces a rule of gain g_j by its g_j replacement events. Edge labels retain the optimal production, occurrence, derivation depth, charged boundary, BPE time and result depth, crossing side, captured outer boundary, rule gain, and whether exact readiness ever held. These labels distinguish proofs but do not give constant total event capacity.

Fix one event capturing boundary c. An occurrence spoiled on its left has $\ell(v) = c + 1$. Forest intervals are disjoint or nested, and two nonempty intervals with the same left endpoint cannot be disjoint; all such occurrences therefore form one ancestor chain. Right-side spoilers similarly share $r(v) = c+1$ and form one chain. Two occurrences of the same production cannot be ancestor-related, because the production result cannot occur inside its own creation-ordered expansion. Derivation depth strictly increases along an ancestor chain. Each event-side chain therefore has congestion one for a fixed production or a fixed depth. A rule with g_j events and two sides has at most $2g_j$ chains, proving Theorem G.1 including simultaneous left/right spoilers.

For the unbounded constructions, take the $t > 2$ string from Appendix E. Its length is $10t$. First use the common $2t$ productions

$$x_i a \to X_i, \qquad a y_i \to Y_i,$$

at both legal occurrences. Their gain is $4t$, and the current state has the $6t$ cells

$$(X_i Y_i)_{i=1}^{t} (\delta_i X_i)_{i=1}^{t} (\varepsilon_i Y_i)_{i=1}^{t}.$$

The core boundaries in the third original block are

$$\{7t + 3i : 0 \le i < t\} \cup \{7t + 3i + 2 : 0 \le i < t - 1\}.$$

After its initial gain-t aa rule, first-occurrence BPE sees only unique current pair types and extends one left prefix for $8t - 2$ unit-gain rules. It misses the consecutive original boundaries

$$q_0 = 9t - 2, q_0 + 1, \ldots, 10t - 2.$$

Counting residues zero and two modulo three among the displayed core boundaries in this interval gives

$$m_t = \left\lfloor \frac{2(t+1)}{3} \right\rfloor.$$

Any binary tree completing the $6t$ core cells uses $6t - 1$ further one-occurrence productions. Together with the common core it uses exactly $8t - 1 = k$ rules and reaches one root, so both completions below are optimal.

For the event construction, let c be the first core boundary at least q_0, let T be the core cell immediately to its left, and left-fold from T through the string end. Fold the preceding cells separately and finally join the two roots. The suffix fold has m_t internal occurrences; their charges are precisely the missed core boundaries at least q_0, and they share the left endpoint of T. The preceding core boundary e is below q_0 and is captured by BPE. It is the first spoiler of all m_t occurrences. It lies after the aa step, where every pair type is unique, so its rule gain is one.

For the terminal construction, start the suffix fold at the core cell immediately to the right of c, fold the prefix separately, and join the two roots across c. The suffix nodes charge the $m_t - 1$ later missed core boundaries and the final root charges c. Their outer boundaries are missed or are string endpoints, so all m_t occurrences remain partition-ready and have no spoiler. In both variants the untouched-tail geometry prevents exact readiness. Thus a unit-gain event fiber and the terminal fiber each have size $m_t \to \infty$, although the positive theorem continues to bound their chain count and per-label congestion.

H FINITE-EVIDENCE PROTOCOLS AND EXACT DENOMINATORS

This appendix binds the empirical statements in Section 8 to their reproducible evidence classes. No empirical row is used as a theorem premise.

H.1 SPECIFIED EXACT AND SYNTHETIC CLASSES

The exact grids contain 7,819 instances with an executed exact-OPE oracle, of which 5,992 have $O > 0$. The adversarial and stochastic stress families retain 1,222 records; exact OPE was executed on the 289 predeclared feasible records, all of which have $O > 0$. The deterministic public-corpus class retains 432 records: 288 length-256 full-window records without exact OPE, and 144 length-12 exact-prefix records with exact OPE and $O > 0$. Their union produces Table 2; its optimum-dependent columns use the 8,252 records with an executed exact-OPE oracle. Exhaustiveness is only over each declared finite grid. Losing, tied, zero-utility, and tie-sensitive records remain in the raw JSONL files rather than being filtered out of the manifest.

Alphabet	Length	Budgets	Canonical words	Records
Binary	10	$0, \ldots, 4$	511	2,555
Ternary	8	$0, \ldots, 3$	966	3,864
Quaternary	7	$0, \ldots, 3$	350	1,400
Total			**1,827**	**7,819**

Table 3: Complete exact-grid domains. Canonical restricted-growth words use every listed symbol and retain one representative of each alphabet-relabeling class; every displayed budget is included.

Exact solvers. For a token word w, remaining budget b, and next fresh label ν, let $\mathcal{P}(w)$ be its adjacent pair types and let $\mathcal{I}(w, p)$ be the nonempty pairwise-nonoverlapping subsets of occurrences of p. The two oracles evaluate

$$D_{\mathrm{OMS}}(w, b, \nu) = \min\left\{|w|, \min_{p \in \mathcal{P}(w)} D_{\mathrm{OMS}}(M_{\mathrm{full}}(w, p, \nu), b - 1, \nu + 1)\right\},$$

$$D_{\mathrm{OPE}}(w, b, \nu) = \min\left\{|w|, \min_{p \in \mathcal{P}(w)} \min_{I \in \mathcal{I}(w,p)} D_{\mathrm{OPE}}(M_I(w, p, \nu), b - 1, \nu + 1)\right\},$$

with value $|w|$ when $b = 0$ or no adjacent pair exists. M_{full} is the deterministic left-to-right full replacement; M_I simultaneously replaces exactly the selected occurrence intervals. Both searches memoize only the complete state (w, b, ν) and use neither branch-and-bound nor dominance pruning. Thus every legal positive-utility transition is explored.

The stress suite contains the theorem and counterexample families plus controlled i.i.d., Markov, Zipf-block, grammar-reuse, periodic-noise, and unary sources. Exact optima are reported only for pre-specified feasible parameter cells. Every completed enumerated row is retained; ratios are

formed only when $O > 0$; a missing cell is displayed as "–", never as zero; and an incomplete or resource-stopped run cannot enter the combined table.

Certificate tightness. On the 8,252 rows for which the exact OPE oracle was executed, $\mathrm{LB}_\partial / L^*$ has median $6/7$, equals one in 4,024 rows, and has minimum $1/6$. These are finite measurements of the theorem's certificate, not approximation guarantees for an algorithm.

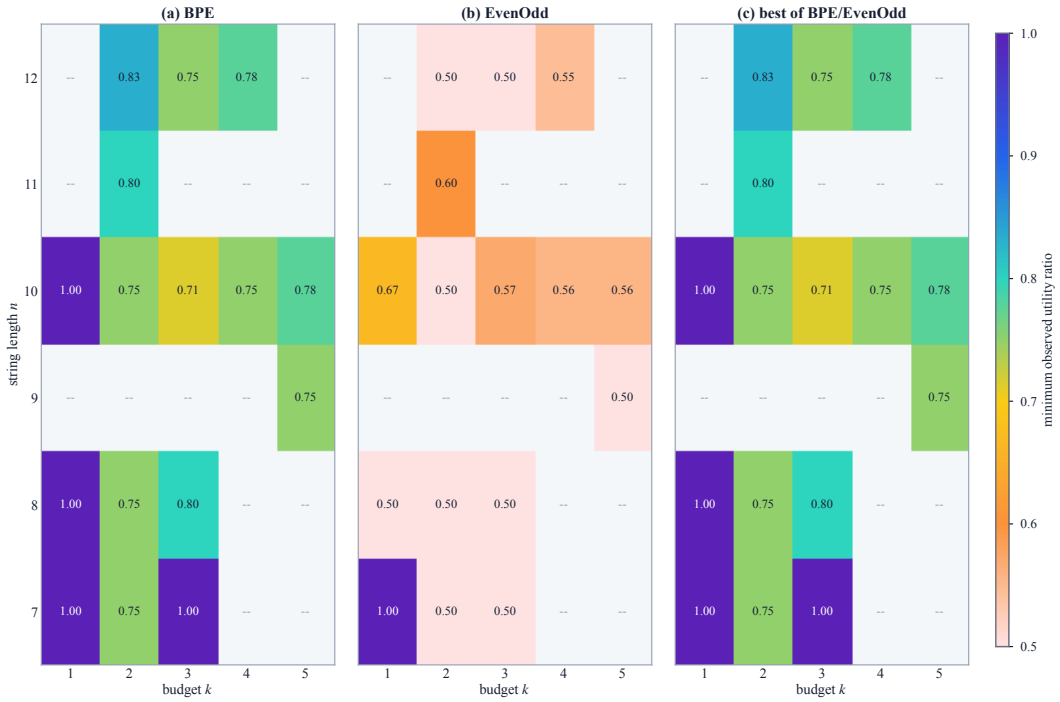

Figure 4: Finite exact utility landscape. Each cell is the minimum observed utility ratio at one specified (n, k) cell; "–" marks an unevaluated cell, not a zero. The three aligned panels expose the distinct BPE, EvenOdd, and best-of profiles while preserving every finite domain boundary.

The heatmap gives the compact cellwise view. The complete distribution and ordered cell sequence below retain the complementary information carried by the original exact-search diagnostics.

Figure 5(a) uses a logarithmic vertical scale so the rare 0.031% and 0.265% bins remain visible beside the dominant exact-optimum bin; a cross on the plotting floor means an empty bin. Panel (b) gives one small multiple for every evaluated string length, retaining every (n, k) cell minimum. Slight horizontal marker offsets reveal coincident BPE and best-of values; they do not alter the budgets. Wide translucent strokes visually separate coincident curves and encode no uncertainty.

Figure 6(a) makes the one-token separation on `0001001`, $k = 2$ explicit: first and worst-tie BPE traces finish at $L = 4$, while exact OMS/OPE and the best tie finish at $L = 3$. Panels (b)–(c) show every evaluated oracle-runtime cell with a shared logarithmic color scale and its numerical mean in milliseconds.

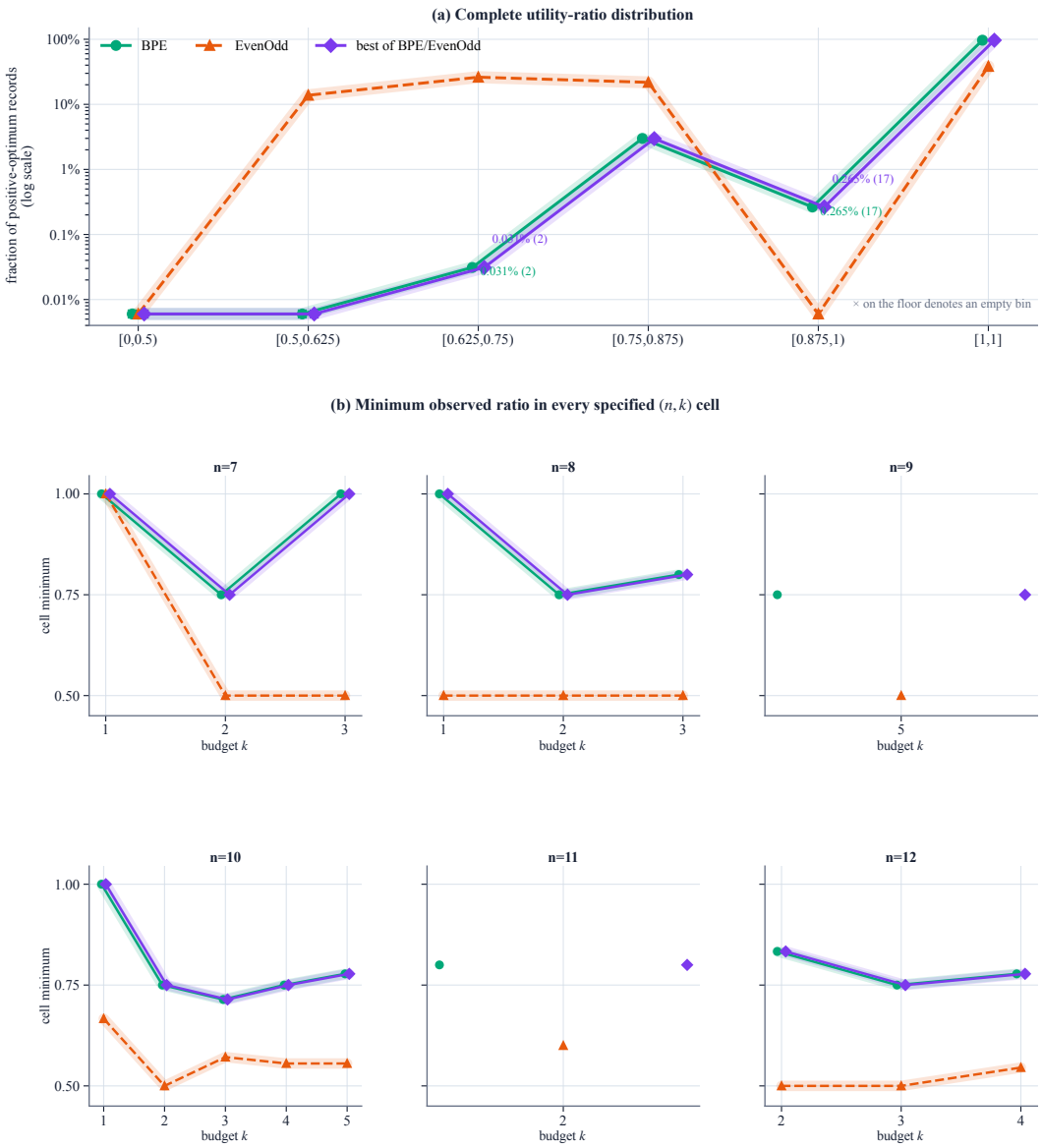

Figure 5: Complete finite exact-ratio diagnostics, redrawn directly from the archived result tables: (a) ratio distribution; (b) cell minima by string length. These are finite observations, not universal guarantees.

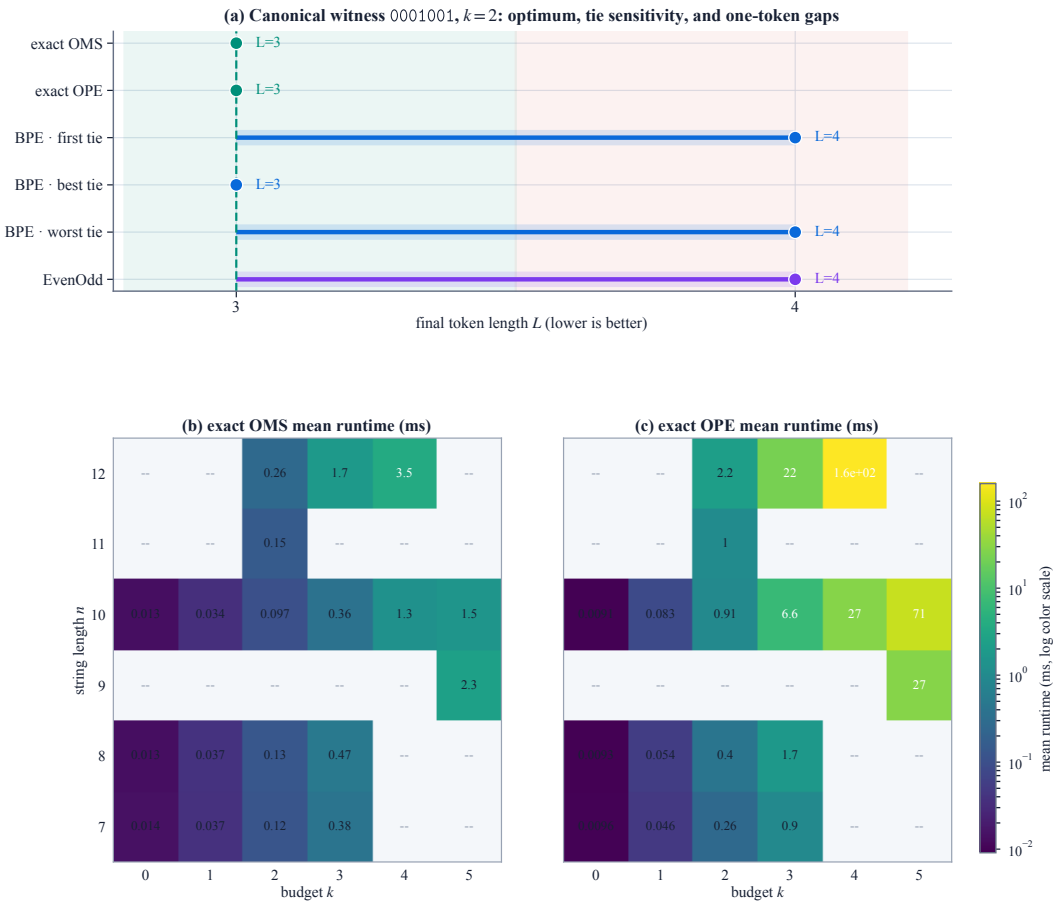

Figure 6: Exact-search audit views, redrawn from the archived witness and runtime CSVs: (a) a one-token separation; (b)–(c) oracle runtimes in milliseconds. Runtime is environment-specific evidence, not a complexity claim.

H.2 Full-window groups

Each row below contains 24 deterministic 256-byte windows. "Strict" means LB_∂ is strictly larger than the depth term; "spread" means positive best-minus-worst utility among the saved named BPE policies.

Source	Budget	Records	Strict	Spread	Exact fields
English EWT	4 / 8 / 16 / 32	96	96	45	not computed
Finnish TDT	4 / 8 / 16 / 32	96	96	32	not computed
Hindi PUD	4 / 8 / 16 / 32	96	96	22	not computed
Total		**288**	**288**	**99**	not computed

Table 4: Full-window outcomes. Exact snapshot/final/time quantities were not computed on full windows; the corresponding fields are absent, not zero.

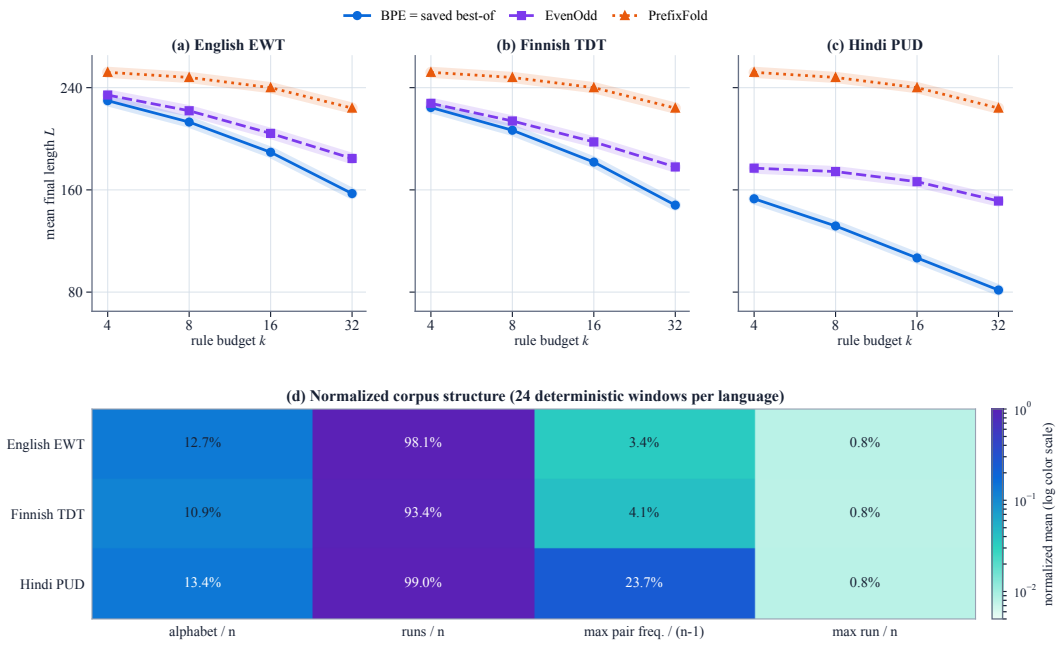

Figure 7: Multilingual corpus diagnostics, redrawn from the archived result tables. Panels (a)–(c) expose every language–budget mean without joining unrelated languages on one line; the saved best-of policy coincides with deterministic BPE in all twelve cells. Panel (d) replaces visually compressed bars with a logarithmic-color matrix whose annotations give every normalized structural mean explicitly. Same-color halos in panels (a)–(c) improve separation and do not encode uncertainty.

The 12-byte prefixes use budgets $2, 3$, yielding 144 exact records. All checked reconstruction, utility/root, packing, snapshot, common-cut, and time-expanded identities pass. Snapshot loss and missed-occurrence counts are zero. Across $1{,}224$ final components, the counts of negative or positive regret, nonpaths, and nonalternation are all zero; no record exhibits cross-component sign cancellation. These nulls describe this deterministic finite sample and do not negate the exact obstruction families.

H.3 Direct-vocabulary diagnostic

The direct experiment uses the same saved records but a different feasible set: arbitrary added substrings followed by optimal segmentation. Over $3{,}988$ budgeted comparisons, direct length is strictly below exact OPE length in every record. The aggregate denominator comprises $3{,}654$ exact grid comparisons, 190 stochastic comparisons, and 144 corpus-prefix comparisons. We report absolute lengths only; no direct/OPE approximation ratio is defined.

H.4 Artifact integrity

The archived experimental bundle contains 432 primary records, $5{,}256$ aggregate rows, and 30 allowlisted deliverable files. A clean replay regenerated 18 core artifacts byte-for-byte. The manifest stores paths, byte counts, and SHA-256 digests; the verification report checks the allowlist, hashes, schemas, identity assertions, and regeneration equality. The anonymous code repository provides solvers, independent tests, frozen configurations, and regeneration instructions. Generated raw records, aggregate tables, and verification reports are not bundled in that code-only release.

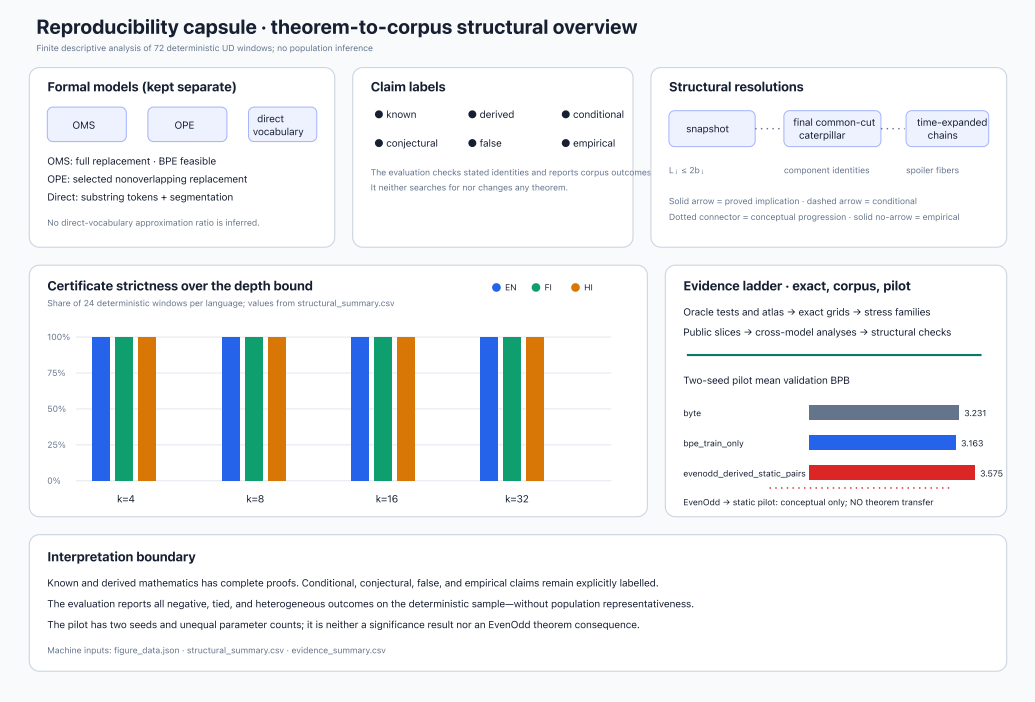

Figure 8: Theorem-to-corpus synthesis. The overview keeps the formal models separate, labels known, derived, conditional, conjectural, false, and empirical claims, aligns the three structural resolutions, and displays the complete evidence ladder from exact grids through the two-seed pilot. The rendered derivative shortens only the three proportional pilot bars so their unchanged numeric labels remain inside the panel. Its archived notation L_j denotes Frag_j in this paper, not token length.

I TWO-SEED SCRATCH-MODEL PILOT

This appendix supplies the learned-representation diagnostic discussed in Section 8; it is separate from the instance utility theorem.

The pilot compares three fixed tokenization conditions in the same small training pipeline: raw bytes, train-only BPE, and a static set of pair tokens derived conceptually from EvenOdd. The static condition is not the instance EVENODD algorithm and inherits none of its $1/2$ theorem.

Condition	Mean validation BPB	Difference vs. byte by seed
Byte	3.2312995296	—
Train-only BPE	3.1632920451	-0.0743553997, -0.0616595692
Static EvenOdd-derived pairs	3.5748389993	$+0.3733108962$, $+0.3137680433$

Table 5: Two-seed pilot. Lower BPB is better; differences are condition minus byte, so both BPE values improve and both static-condition values degrade.

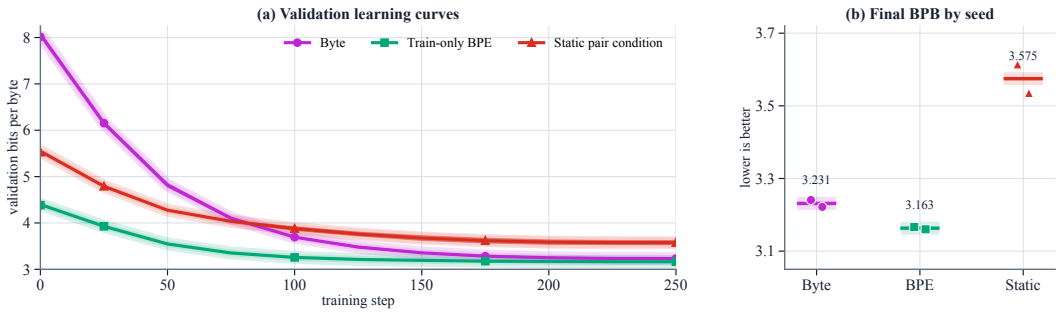

Figure 9: Pilot trajectories. In panel (a), thin lines are the two individual seeds, thick marked lines are their means, and each translucent same-color envelope spans the two observed seed values at that step; it is not a confidence interval. Panel (b) shows both final seed values and the corresponding mean. Train-only BPE improves on bytes in both seeds, whereas the static pair condition degrades in both. Same-color halos improve visual separation and encode no additional statistical quantity.

This pilot is intentionally diagnostic: two seeds, a scratch model, aggregate BPB, unequal tokenizer-dependent parameter counts, and no per-language, significance, large-model, or downstream analysis. The symmetric positive and negative outcomes are reported because they answer whether this particular static adaptation should be treated as a drop-in consequence of the theorem: it should not.

J Model and source ledger

This ledger expands the model comparison in Section 2 and records the conventions used by the approximation and hardness results in Sections 4 and 5.

Model	Feasible set and objective	Proved frontier used here
One-string OPE	selected pair occurrences; minimize roots of one string	utility APX-complete; length APX-hard; EvenOdd is $1/2$
One-string OMS	full left-to-right pair replacement; minimize roots	BPE utility in $[1/3, 5/8]$; length gap $\Omega(n)$
Corpus bottom-up	exhaustive surface-token merge sequence; total tokens	binary APX-hard, no PTAS
Direct vocabulary	arbitrary substrings plus optimal segmentation; total tokens	binary APX-hard, no PTAS

Table 6: Formulations are intentionally separated: a source theorem in one row does not transfer to another row. The paragraphs below record the exact source/version bindings.

One-string models. Kozma & Voderholzer (2026) prove APX-completeness for utility and APX-hardness/no PTAS for final length in the source one-string setting. Compressed-length APX membership is not asserted. Their BPE utility ratio lies between $1/3$ and $5/8$, where $5/8$ is an upper endpoint from a family, not an exact ratio theorem; EvenOdd has the unconditional $1/2$ utility guarantee. The local exact-to-at-most alignment uses their same growing-alphabet hard family and changes no hardness conclusion.

Ordered forests and grammars. Zouhar et al. (2023) show that applying a deterministic full merge sequence yields an ordered forest/partial bracketing. They do not prove the selected-occurrence iff characterization of Theorem 3.2, its reverse replay, or preservation of every exact (r, L) pair. Pair packing and boundary charging originate in Kozma & Voderholzer (2026); our claims are the closed-form snapshot marginals, the unbounded realizability slack, and the three exact resolution theorems. The smallest-grammar objective (Charikar et al., 2005) has the same feasible hierarchies after unit elimination, binarization, and an explicit start sequence. The remaining distinction is objective: grammar size is within a factor two of WPF_2, yielding the logarithmic transfer of Theorem C.2,

while the separate uniform theorem (Theorem 4.2) controls both coordinates and all positive dyadic weights.

Corpus models. Kastreva et al. (2026) prove binary hardness for direct tokenization and standard bottom-up corpus tokenization. The conference version calls the latter APX-complete, whereas arXiv:2511.15709v2 states APX-hardness and leaves constant-factor approximability/APX membership open. We therefore cite only the stable common statement: APX-hardness with the source gap, hence no PTAS unless $P = NP$. It does not transfer to canonical one-string partial replacement.

Boundary substrings and interval graphs. The relevant-substring facts of Goto et al. (2011) and Bille et al. (2013) supply the SLP antecedent of Lemma 6.1. The generic caterpillar fact used in Theorem E.1 is classical (Eckhoff, 1993); the ordered-partition scan and regret identities are the pair-encoding specialization developed here.

New approximation and hardness transfers. The boundary-popping operation is due to Jeż (2013). Theorem 4.2 adds a binary host with shared powers and private run concatenations, exact closed pruning, and separate comparison production/root accounting. Its $O(\log^2 n)$ factors are not claimed tight. The graph gadget and one-root binary optimum in Lemma 5.4 are due to Kawamoto et al. (2024), Theorem 3.1. We establish extraction from arbitrary forests and the budget/length transfer, using the exactly cubic vertex-cover gap of Alimonti & Kann (2000), Theorem 3.1 and the end of Section 3. Theorem 5.1 concerns deterministic algorithms, explicit one-string inputs, and growing alphabets. It neither transfers to the corpus rows nor establishes fixed-alphabet hardness or a utility ratio.

K Universal binary recompression and exact frequency pruning

This appendix proves Lemma 4.1, Theorem 4.2, and Corollary 4.3, and gives the unchanged-dictionary obstruction. All output rules are binary; terminal aliases, run-length rules, and proof-only residual variables incur no unreported output operations.

K.1 Exact pruning of an arbitrary host

Let H be any reachable forest decoding s, not necessarily one-root. Initialize one count for each root occurrence and propagate counts from later productions to both children, twice when the children coincide. The resulting m_X equals the number of X nodes in the fully unfolded occurrence forest. Each occurrence of a parent supplies a distinct occurrence of its child, so $m_Y \geq m_X$ for each nonterminal child Y of X.

For a dependency-closed label set D, recursively expand every unretained root until only terminals and retained labels appear, keeping the definitions in D. Each retained label remains reachable: trace one of its original occurrences upwards until reaching a retained ancestor or the expanded root frontier. Precisely the internal occurrences labeled in D survive. A binary forest with n leaves and L_D roots has $n - L_D$ internal occurrences, proving

$$L_D = n - \sum_{X \in D} m_X .$$

Creation order breaks multiplicity ties with dependencies first. Thus every prefix of descending multiplicity order is closed, and no at-most-k subset has greater total multiplicity than its first $\min\{k, r_H\}$ entries. This proves exact budget optimality within the host. Similarly,

$$\lambda|D| + L_D = n - \sum_{X \in D} (m_X - \lambda)$$

is minimized by retaining exactly $m_X > \lambda$. This set is closed; all zero-coefficient labels can be omitted. These arguments include empty sums. Counts need no exponential unfolding, and explicit frontier expansion visits at most $2n - L_H$ occurrence nodes. This proves Lemma 4.1 in full.

K.2 THE BINARY HOST, INCLUDING EVERY AUXILIARY RULE

Normalize terminal names and use globally fresh names for all created labels. Reuse is within a phase; equal terminal yields do not identify distinct labels. In each phase, execute the following operations.

Runs. For every base token a appearing in a maximal run of length at least two, let M_a be its maximum run length. Create shared powers

$$P_{a,0} = a, \qquad P_{a,j} \to P_{a,j-1}P_{a,j-1} \quad (1 \le j \le \lfloor \log_2 M_a \rfloor).$$

For each distinct length ℓ of base a, list its set bits in descending order and concatenate the corresponding powers in a left-associated chain. The chain's fresh concatenation auxiliaries are *private to* (a, ℓ): never share them between different lengths, even if their right-hand sides coincide. For a power-of-two length, use the power itself; for length one use a. Replace every maximal run by its designated output. Let u be the result.

Pairs. Adjacent names in u are distinct. Assign each name to A_L or A_R so that at least $(|u| - 1)/4$ adjacent occurrences lie in $A_L A_R$. A random independent partition has this expectation. Derandomize by assigning names in increasing order, choosing the larger conditional expectation and choosing left on ties. All scores multiplied by four are integers. Selected occurrences are disjoint because $A_L \cap A_R = \varnothing$. Replace them, using one fresh binary label $C_{a,b} \to ab$ for every selected ordered pair type in this phase.

Lemma K.1 (Host legality, contraction, and one-time run bases)**.** *For $n \ge 2$ the host is reachable, binary, creation-ordered, one-root, and has at most $n - 1$ productions. If a phase starts at length N and ends at N', then $N' - 1 \le 3(N - 1)/4$. A fixed token name is a base of new power rules in at most one phase. With $h = \lceil \log_2 n \rceil$, each base uses at most h powers and each distinct run length at most h concatenation auxiliaries.*

Proof. The replacements define an ordered binary occurrence tree over the original leaves. Every auxiliary is used in its designated run; the largest power of each base is used in a maximum-length run and reaches all smaller powers. Every pair label is used in the current text. Subsequent replacements preserve these occurrences as descendants. Hence all rules remain reachable. The final binary tree has $n - 1$ internal occurrences, bounding distinct productions. Theorem 3.2 gives legal selected-occurrence replay. If $M = |u| \le N$, pair selection gives $N' - 1 \le 3(M - 1)/4 \le 3(N - 1)/4$. Therefore at most q phases suffice for the least positive integer q with $3^q(n - 1) < 4^q$.

After all nonsingleton runs of an old name a have been replaced, surviving a tokens are pairwise nonadjacent. A later operation replaces a nonempty interval by a fresh name; it neither introduces a nor removes an intervening interval without replacement. Two surviving a tokens never become adjacent. New runs of newly created names use new bases. The binary-expansion bounds follow because every run length is at most n. □

K.3 COMPARISON SIMULATION BY BOUNDARY POPPING

Fix any comparison forest with r binary secondary rows and a distinguished start row of L roots. Original comparison-variable references and current host-token atoms are separate alphabets. Transform these rows in parallel with the current text; the start expansion must equal that text. Comparison rows may become empty or unit rows, but they are proof devices, not output rules. Remove references to empty variables. Never add a comparison row or insert or copy a comparison-variable reference. We use the boundary-popping principle of Jeż (2013) with the following explicit simulation.

Block popping. Before a run step, process comparison variables in original creation order. Pop the entire maximal constant-token prefix and suffix of each nonempty variable, with a pure run popped as a prefix only. Delete these pieces from its row and insert them around each reference to the variable; remove the reference if the residual is empty. Store a popped run as one temporary atom (a, ℓ) regardless of its length. Processing children first exposes a parent's maximal prefix and suffix: each nonempty child has exposed its old boundary runs, and its residual starts and ends with a different token. Each pop preserves the expansion of every other row, including the start row.

After popping, no maximal run crosses a reference boundary. An exposed a before a residual cannot meet an a at its front: either the variable did not start with a, or its entire a prefix has been popped. A newly exposed residual boundary is separated from its old context by the popped run of a different token. The same reasoning applies to right boundaries and nested references. Every expanded maximal run is therefore an explicit block in some row. Replace it with exactly the host's designated run-output label. Temporary run atoms disappear, leaving ordinary tokens and references.

Pair popping. Before a pair step, in creation order pop a variable's first token if it is in A_R and its last token if it is in A_L, removing empty residual references. An A_R token before a residual cannot be the left member of a selected pair; an A_L token after a residual cannot be the right member. An unpopped boundary token is on the wrong side to cross its boundary. With dependencies processed first, every selected pair is explicit inside a row. Replace it using the same pair label as the host.

All nonempty comparison variables remain reachable: only empty references and explicit boundary pieces are removed. A nonempty reference is retained either directly or within a retained containing reference. Thus every nontrivial run or selected pair in a secondary row also occurs in the current text and has a host label. This establishes both the simulation invariant and valid label lookups without a common-frontier assumption.

K.4 Whole-execution atom accounting

An atom is one reference, one current-token occurrence, or one temporary run. Count secondary rows and the start row separately. Initial counts are $2r$ and L; reference-occurrence counts never exceed $2r$ and L, respectively. For either collection, write a for its initial atom count and e for this reference bound. Across all p phases, let I_b, I_p be atom insertions around references by block and pair popping, and D_b, D_p the atom-count decreases from block and pair replacement. Each popping procedure inserts at most two atoms per reference per phase. Popping from a variable's own row and removing empty references only decrease counts. Telescoping gives

$$I_b \le 2ep, \quad I_p \le 2ep, \qquad D_b + D_p \le a + I_b + I_p \le a + 4ep. \tag{10}$$

A nontrivial run represented by at least two atoms pays one unit of D_b. One represented by one atom must be a newly inserted temporary run (a, ℓ), $\ell \ge 2$, and pays one insertion. Hence there are at most $D_b + I_b$ nontrivial run-replacement occurrences and exactly D_p pair replacements. Distinct output types are no more numerous. Empty rows and singleton residuals are included in this accounting.

K.5 Closed marking and the production bound

Begin with $D = \varnothing$. Maintain the invariant that every host token explicitly in a secondary row is terminal or already marked. Before block popping, collect all base names explicit anywhere in secondary rows. Mark *all* host power rules created for these bases in that phase, even powers used only by start-row runs. Every popped run has such a base: descend its references before popping to an explicit occurrence in a secondary row. Thus popping cannot import a start-only base into the secondary grammar. Mark the full private concatenation chain for each nontrivial run replaced in a secondary row, and mark each pair label created there.

Closure follows inductively. Marked powers depend on their already marked or terminal bases and smaller powers; run concatenations depend on these powers and earlier auxiliaries of their own chain; pairs depend on already available secondary tokens. Popping moves existing tokens through reference edges. Every new explicit secondary token is marked, maintaining the invariant.

Now use only secondary-row counts in (10). The number J of nonterminal host tokens that ever first appear explicitly in secondary rows satisfies $J \le D_b + D_p + I_b$: first appearance occurs at replacement, and popping cannot introduce a type absent from all secondary rows. There are at most $2r$ initial terminal types. Each paid base creates powers once, so marked powers number at most $h(2r + D_b + D_p + I_b)$. Marked concatenations number at most $h(D_b + I_b)$ and pairs at most D_p. With $h \ge 1$, $a = e = 2r$,

$$\begin{aligned} |D| &\le 2hr + 2h(D_b + D_p) + 2hI_b \\ &\le 2hr + 2h(2r + 8rp) + 2h(4rp) = 6h(1 + 4p)r. \end{aligned} \tag{11}$$

These counts never charge start-row events as secondary productions.

K.6 Unmarked occurrences and the root bound

A rule's final unfolded multiplicity equals the number of corresponding internal nodes made at creation: later substitutions nest those nodes. We charge unmarked occurrences to start-row atom events in two run cases.

If a base is absent from all secondary rows before popping, none of its current occurrences lies inside a comparison-variable expansion. Popping cannot introduce it there. Its runs consist of ordinary explicit start tokens. A run of length ℓ creates exactly $\ell - 1$ binary internal occurrences and removes $\ell - 1$ start atoms. Charge all its new internal occurrences to D_b.

If a base is present in a secondary row, all its power labels are marked. An unmarked run concatenation must have a designated length with no secondary-row replacement. By uncrossing, all occurrences of that full run are explicit start-row replacements. The auxiliary is private to its length, so it occurs *once per occurrence of that designated run*, and in no other run-length representation. At most h auxiliaries per run cost at most $h(D_b + I_b)$. The two cases are disjoint; multiplying all block-decrease charges by $h \geq 1$ pays both. In particular, a huge run exponent never incurs an unmarked power charge when its base appears in the secondary grammar.

An unmarked pair label has no secondary-row replacement, so each occurrence is a start-row pair replacement and costs one D_p. Using now $a = e = L$ gives

$$\sum_{X \notin D} m_X \leq h(D_b + D_p + I_b) \leq h(L + 4Lp + 2Lp) = h(1 + 6p)L.$$

The host has one root, so

$$L_D = 1 + \sum_{X \notin D} m_X \leq [1 + h(1 + 6p)]L,$$

where $L \geq 1$. Together with (11), this proves Theorem 4.2.

K.7 Selecting the weighted and budgeted outputs

Define ρ_n, β_n using the integer phase bound q in the main text. For a weighted-optimal comparison forest F, the theorem supplies a closed pruning of cost at most

$$\lambda\rho_n r(F) + \beta_n L(F) \leq \rho_n \mathrm{WPF}_\lambda(s).$$

A weighted optimum exists because there are finitely many partial executions with at most $n - 1$ nonempty steps. Threshold pruning is at least as good as this witness. For an optimal at-most-k comparison, the witness uses at most $\rho_n k$ rules. The frequency prefix with that budget minimizes roots among closed prunings, proving the direct (ρ_n, β_n) guarantee without a scalarization sweep. The constants are integers; no hidden rounding is needed. For $k = 0$ take no productions; for $k \geq n - 1$ and $n > 0$ take the entire host. For $n = 0, 1$ return the exact zero-rule forest. This proves both consequences in Corollary 4.3.

K.8 Polynomial runtime with compact dyadic encodings

The host has $O(n)$ productions and tokens and $O(\log(n+2))$ phases. A direct conditional-expectation implementation scans all adjacent positions for the two trial assignments of each name, taking $O(N^2)$ operations on a phase of length N. Scores times four are integers at most $4N$. Names and run lengths have $O(\log n)$ bits, and all yields expand within the explicit input. Sorting, tables, and scans suffice; this is a polynomial implementation, not a claim of the source algorithm's linear runtime.

Write $\lambda = a2^e$ with positive integer a and signed binary integer e, and let B be their total encoding length. For multiplicity $m \geq 1$, compute $b = \mathrm{bitlength}(a) - 1 + e$. If $b < 0$, then $\lambda < 1$; if $b \geq \mathrm{bitlength}(m)$, then $\lambda > m$. Otherwise compare m with $a \ll e$ for $e \geq 0$, or $m \ll (-e)$ with a for $e < 0$. In this remaining case shifts are at most $O(\log n)$ or $O(\mathrm{bitlength}(a) + \log n)$, respectively. All integers have $O(B + \log(n + 2))$ bits. Ties are discarded. Enormous binary-encoded exponents never require materializing enormous powers or denominators.

With elementary integer arithmetic, a conservative bit bound is

$$O\big(N_{\mathrm{in}}^2 + n^2 \log^2(n + 2) + n(B + \log(n + 2))^2\big),$$

where N_{in} is the explicit terminal encoding length. For a budget query, first compare k with $n-1$; otherwise k has $O(\log n)$ bits. Include the input bit length of k in the runtime. Comparison-grammar simulation and exponential exact validators are proof instrumentation, never algorithmic subroutines. For an execution trace, unfold the output forest and contract equal labels in creation order. At most $n-1$ contractions are emitted; recomputing indices by scans takes $O(n^2)$ word operations.

K.9 Edge cases and exact calibrations

Empty input returns $(0,0)$ and a singleton $(0,1)$. Rule-free comparison forests are allowed: the empty marking has $L=n$ to pay every unmarked node. No proof divides by r. For nonempty input and $\lambda<1$, all host rules survive; for $\lambda \geq n-1$, none survive and the zero-rule forest is weighted-optimal. Exact multiplicity ties may all be omitted for weights; budget ties require creation order to preserve closure. Repeated children count twice, and equal terminal yields never identify distinct host labels.

For an all-distinct string of length n, each rule has multiplicity one. Consequently, for every $n, k \geq 0$,

$$L^*_{\text{OPE}}(s,k) = n - \min\{k, \max\{0, n-1\}\}.$$

For $n>0$, $\text{WPF}_\lambda(s) = \min\{n, 1+\lambda(n-1)\}$; it is zero for $n=0$. Both frequency selectors attain these optima.

For $s = a^{2^h}$, the host is the doubling chain. Any r-rule forest has root yield at most 2^r, so for $0 \leq r \leq h$ its length is at least 2^{h-r}, attained by the chain. Larger budgets cannot improve one root. Host pruning is therefore exact for every strict budget and weight on this family. In particular, for $h \geq 1$,

$$\text{WPF}_2(a^{2^h}) = 2h: \quad 2r + 2^{h-r} = 2h + (2^j - 2j) \geq 2h \quad (j = h-r \geq 1),$$

with equality at $j=1$, while $r \geq h$ costs at least $2h+1$.

Exact optimization within a host need not be global exactness. On $s=00101$, $\lambda=1$, the host has cost five while rules $A \to 01$, $B \to 0A$ and roots BA give cost four, which is optimal. On $s=010010$, $\lambda=1$, the host's best cost is six versus optimum four. These finite examples do not establish tightness of the asymptotic $O(\log^2 n)$ factor.

K.10 A sharp obstruction to unchanged-dictionary repair

Theorem K.2 (Deletion repair under arbitrarily small constant scalar slack)**.** *For every fixed $c>1$, there exist $\eta>0$ and infinitely many unary strings $s=a^n$, budgets $k=\log_2 n$, and reachable one-root forests G with*

$$2r(G)+1 \leq c\,\text{WPF}_2(s), \qquad r(G) \leq (1+\eta)k, \qquad L^*_{\text{OPE}}(s,k)=1,$$

and height $O(\log n)$, such that every legal forest using at most k of G's unchanged productions, with arbitrary resegmentation and no new productions, has at least $\frac{1}{2}n^\eta$ roots.

Proof. Choose $\eta = \min\{(c-1)/2, 1/4\}$, take sufficiently large integer h, and put $d = \lfloor \eta h \rfloor \geq 1$, $2d+1 \leq h$, $n=2^h$, and $k=h$. Starting with $A_0=a$, create

$$B_i \to A_{i-1}A_{i-1}, \quad A_i \to B_i a \quad (1 \leq i \leq d), \qquad P \to A_d a,$$

followed by $h-d-1$ doublings of P and its successors. Induction gives $|A_i| = 2^{i+1}-1$ and $|P| = 2^{d+1}$, so the final root has length 2^h. There are $h+d$ rules, each depending on its immediate predecessor. Hence the dictionary is reachable with dependency order a chain and height at most $h+d$. The exact unary optimum gives

$$\frac{2r(G)+1}{\text{WPF}_2(s)} = 1 + \frac{d}{h} + \frac{1}{2h} \leq 1 + 2\eta \leq c$$

for sufficiently large h, and $r(G) \leq (1+\eta)h$. The ordinary h-rule doubling dictionary achieves one root.

Every legal retained dictionary is a prefix of the chain: a later rule needs all earlier definitions. The first h rules include P and $h-2d-1$ further doublings, with longest yield 2^{h-d}. Any resegmentation over at most these rules therefore needs at least 2^d roots. Expanding the final d doublings attains exactly 2^d, and $2^d \geq \frac{1}{2}n^\eta$. □

For example, $h = 3, d = 1$ gives $B \to aa$, $A \to Ba$, $P \to Aa$, $Q \to PP$. The legal partial trace is $a^8 \to BaaBaa \to AaAa \to PP \to Q$. Budget three retains at most B, A, P and needs exactly two roots. The result isolates a restriction on dictionary repair, whereas Theorem 5.1 allows arbitrary redesign.

K.11 Executable finite validation

The supplementary host validator checks 935 atlas inputs, 4,221 comparison forests, and 11,263 closed subsets. It includes 13 dyadic weights on 934 nonempty inputs, 32 structured cases through length 4,160, 400 random comparison forests, and 56,406 exact fraction comparisons including 68 large-exponent cases. Separate exact oracles and occurrence accounting check legality, pruning, closure, and the two comparison bounds. These are finite implementation checks supporting the proved statements, not estimates of an asymptotic approximation constant.

L Completion, graph extraction, and the computational budget barrier

We prove the full hardness transfer underlying Theorem 5.1 and Corollary 5.2. Inputs are explicit strings; rules are fresh binary concatenations with no charged terminal aliases. The alphabet may grow. The parameter γ below counts one-root binary rules and differs from the ordinary right-hand-side size g^* in Theorem C.2.

L.1 Completion and the exact weight-one identity

Lemma L.1 (Root completion)**.** *For nonempty s, let $\gamma(s)$ be the minimum number of binary productions in a reachable one-root forest. Every forest (r, L) of s satisfies*

$$\gamma(s) \le r + L - 1, \qquad \mathrm{WPF}_1(s) = \gamma(s) + 1, \qquad L^*_{\mathrm{OPE}}(s, k) = 1 \iff \gamma(s) \le k.$$

Proof. Join the ordered roots using $L - 1$ fresh binary productions. Each join has available children and all old rules remain reachable, giving a one-root forest with $r + L - 1$ rules. Conversely, a smallest one-root forest has weighted cost $\gamma(s) + 1$. Nonempty input has at least one root, proving the final equivalence. The joins are proof or extraction operations, not a claim to preserve the original rule budget. Empty input has cost zero and is excluded from the identity with the added one. □

Under a Chomsky-SLP convention charging one terminal rule per distinct symbol, the optimum number of rules is exactly $\gamma(s) + |\Sigma(s)|$: remove aliases or introduce one per terminal. This conversion connects the next gadget to Kawamoto et al. (2024) without confusing binary-rule count with unrestricted right-hand-side size.

L.2 The known graph gadget and extraction from any forest

This subsection proves Lemma 5.4 from Section 5.2. Use the graph encoding (3): each displayed separator is globally distinct, including across the two vertex blocks. It has $n = 12v + 6e$ and alphabet size $5v + e + 1$. We establish extraction from every reachable forest and recover the known one-root optimum with the cited binary-rule convention.

Proof. **Repeated yields.** A substring of length at least two containing a separator occurs once. Separator-free parts of X have length at most two. In Z, the only possibly repeated nontrivial substrings are $\#a$, $a\#$, and $\#a\#$ for $a \in V$. Every longer separator-free substring contains consecutive vertex names along a unique oriented edge of the simple graph, so it occurs once. Some listed triples may also occur once; we need only the containment of repeated types.

Normalize an arbitrary forest without increasing $r + L$. In its fully unfolded occurrence forest, child multiplicity is at least parent multiplicity. Thus labels of multiplicity at least two form a closed set. Expand all other rules. Each deleted label occurs once and changes (r, L) by $(-1, +1)$, preserving $r + L$. All remaining nonterminal yields are among the listed pairs and triples.

Consolidate equal-yield labels in increasing yield length, retaining a single definition and remapping references. Children have strictly shorter yields, so no cycle appears and neither rule count nor root count increases. Each triple uses an available pair child. This normalization is an operation within the reduction; it neither identifies labels for free in the original model nor assumes a special input dictionary.

Add each missing pair $\#a$ and $a\#$ with one binary rule. It has two disjoint private occurrences in the two vertex blocks, exposed as adjacent roots: separators prevent other surviving tokens from covering them. Replacing these two occurrences adds one rule and removes two roots, so $r + L$ decreases. Existing triple definitions remain valid. Now optimally resegment over this pair/triple dictionary by ordinary substring dynamic programming, without increasing L. Unused rules may be removed, or counted for an upper bound.

Read off a cover. Let S be the vertices whose triple was retained, and $u(S)$ the number of edges uncovered by S. The dictionary has $2v + |S|$ rules. Shortest segmentation of X uses $8v$ roots. An edge block uses three roots if covered and four otherwise: the five-symbol core uses a triple and a pair exactly when covered, and otherwise two pairs and a terminal; the private final separator contributes one. Hence normalized cost is

$$(2v + |S|) + 8v + 3(e - u(S)) + 4u(S) = 10v + 3e + |S| + u(S) \le r + L. \tag{12}$$

Add an arbitrary endpoint of each uncovered edge to S, producing a cover Q with $|Q| \le |S| + u(S)$. This proves (4); uncovered edges need not form a matching, and the input need not be optimal.

Attain the one-root optimum. Take a minimum cover S, build all pair rules and its triple rules, parse the blocks as above, and join the roots. The binary-rule count is $10v + 3e + \tau(G) - 1$. The reverse inequality follows from (4) at $L = 1$. Creation order and reverse replay give legal OPE witnesses throughout. □

L.3 A QUANTITATIVE APPROXIMATION TRANSFER WITHOUT AN OPTIMUM ORACLE

For a simple cubic graph, $e = 3v/2$, $n = 21v$, and $\tau(G) \ge v/2$, because a cover vertex meets only three of the $3v/2$ edges. Thus

$$\gamma(w_G) = \tfrac{29}{2}v + \tau(G) - 1 \le 30\tau(G), \qquad n \le 42\tau(G). \tag{13}$$

Proposition L.2 (Budget/length approximation transfer)**.** *Suppose a deterministic polynomial-time algorithm returns forests with $r \le b(n)k$ and $L \le \ell(n)L^*_{\mathrm{OPE}}(s,k)$, where $b(n) \ge 1$. Polynomially many calls give, on every cubic graph, a cover with*

$$|Q| \le \left[1 + 30(b(n) - 1) + \frac{42\ell(n)}{n}\right]\tau(G), \qquad n = 21v.$$

Proof. Construct w_G, run the forest algorithm at each integer $k = 0, \dots, n-1$, apply Lemma 5.4 to each output, and return the smallest cover. The unknown $\gamma(w_G)$ is among these budgets. At that budget the optimum length is one, so one extracted cover satisfies

$$\begin{aligned} |Q| &\le b(n)\gamma(w_G) + \ell(n) - 1 - C_G \\ &= \tau(G) + (b(n) - 1)\gamma(w_G) + \ell(n) - 1. \end{aligned}$$

Apply (13) and drop the final -1. The procedure needs neither the correct budget nor a minimum-cover oracle. It makes n queries with $O(\log n)$ budget bits. All reachable explicit-string forests have polynomial size; expansion and literal segmentation use the explicit text, with no exponential or compressed-input subroutine. □

L.4 A SEPARATE ONE-QUERY GAP WITH EXACT ROUNDING

We prove Proposition 5.3 from Section 5.1. Use the rational source-gap constants $0 \le \beta < \alpha \le 1$, the budget $k = 29v/2 + \lfloor \beta v \rfloor - 1$, and the graph encoding (3). The following argument yields the precise NO-instance bound in (2).

Proof. In the YES case, integrality gives $\tau \le t$, so the gadget identity gives $\gamma \le k$. In the NO case, completion and the gadget identity imply

$$\begin{aligned} L &\ge \gamma + 1 - r \\ &> (29/2 + \alpha)v - (1 + \varepsilon)(29v/2 + t - 1) \\ &\ge \big[(\alpha - \beta) - \varepsilon(29/2 + \beta)\big]v + 1 + \varepsilon. \end{aligned}$$

The last inequality uses $t \le \beta v$ with the subtraction's correct sign; v is even in a cubic graph. A length guarantee below this gap on YES instances would distinguish the promise cases with one query. □

L.5 Linear-order hardness and all three consequences

By Alimonti & Kann (2000), choose a fixed $\delta > 0$ such that a polynomial-time $(1+\delta)$ approximation for cubic vertex cover implies P = NP, decreasing δ to at most one if needed. Set

$$\varepsilon_0 = \delta/120, \qquad c_0 = \delta/168.$$

An algorithm forbidden by Theorem 5.1, used in Proposition L.2, would give a cover factor

$$1 + 30\varepsilon_0 + 42c_0 = 1 + \delta/2 < 1 + \delta.$$

A guarantee for sufficiently long strings suffices: finitely many smaller graph orders can be solved by exhaustive search in constant time with respect to asymptotic input length. This proves Theorem 5.1.

For completeness, each consequence follows with its own quantifiers. First, $\ell(n) = o(n)$ is eventually at most $c_0 n$, and a strict rule budget also satisfies the relaxed bound. Second, if a scheme worked for every fixed $\varepsilon > 0$, fix one $0 < \varepsilon \le \varepsilon_0$. Its $\mathrm{poly}(\log n, 1/\varepsilon)$ factor is $o(n)$. There is no substitution $\varepsilon = 1/k$ and no runtime-uniformity assumption in ε. Third, every step of extraction allowed arbitrary productions and arbitrary legal occurrence choices, so the conclusion is computational even with full dictionary redesign. Finally the zero-rule forest has $L = n \le nL^*_{\mathrm{OPE}}(s, k)$ on every nonempty input. This matches the lower bound in asymptotic order, not in an optimal coefficient multiplying n. These arguments prove Corollary 5.2.

The hard family has alphabet size linear in n with integer labels of $O(\log n)$ bits, so explicit encoding is polynomial and n counts input symbols. The theorem makes no fixed-alphabet, corpus, utility, or BPE-ratio claim. It leaves larger constant budget inflation open and does not prove $\Omega(\log^2 n)$ necessary inflation. The deterministic assumption is material: a bounded-error randomized exclusion needs an assumption such as NP $\not\subseteq$ BPP, not merely P $\ne$ NP.

L.6 Independent finite validation of the reduction

The supplementary validator uses a selected-occurrence mask dynamic program, a separate literal-substring dictionary oracle, exact cover enumeration, shortest segmentation, and fresh random forests. It verifies the completion identity on all 278 nonempty equality patterns through length six, plus the empty case. It checks 80 graph gadgets: all 75 labeled simple graphs on one through four vertices and five seeded five-vertex graphs, enumerating 87,113 feasible dictionaries and testing 400 random forests. Every vertex subset is checked against (12).

Another 105 forests on cubic graphs of $4, 6, 8, 10, 12$ vertices include random outputs and the universal recompression host, with input lengths through 252. There are 15,444 exact rational tradeoff checks, and the seed is 20260905. For K_4, the check gives $n = 84$, $\tau = 3$, $\gamma = 60$. Saved JSON fields match read-only regeneration. The dictionary oracle uses the proved multiplicity-one normalization to enumerate its search space; the independent mask oracle cross-checks that normalization on short strings. These finite checks complement the proof and never substitute for the imported complexity gap or the asymptotic argument.