# ANCHORED SELECTIVE STATE SPACES: REUSABLE COMPUTATION AND ANCHOR DECISIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

An exact anchored representation preserves selective recurrences while exposing reusable, length-indexed propagation. This paper quantifies the computation that specialization removes and characterizes when a coordinate-range contract permits its use. On a fixed source schedule, prepared propagation eliminates exactly the consumed dynamic transition products; a separate ledger charges on-demand power generation. Exact feasible-anchor intervals supply retaining anchors or infeasibility certificates, and temporal-order obstructions show why marginal increment statistics are insufficient. A sharp mean-span factor $2 - 2/T$ gives an oracle-relative partition guarantee under budget augmentation, while a same-budget construction can lose the transition-product saving entirely. These partition results also hold for the net local saving after power generation, with explicit boundary-cost tradeoffs. Matched H100 experiments include preparation and compare policies within both affine and source-only executors. On the 48-case primary panel, mean improves affine execution over deployed anchors by $1.658\times$ /$1.298\times$ and over width-two rescue by $1.135\times$ /$1.091\times$ for 130M/370M cores. Source-only adds smaller gains. Strong parallel physical precision controls pass every tested setting and remain substantially faster than anchored execution. The analysis connects exactly removable computation to its coordinate conditions and partition choices, while distinguishing this local saving from a reduction in complete execution cost.

## 1 INTRODUCTION

Selective state-space recurrences retain input-dependent transitions, which generally remove a stationary input–output convolution (Gu & Dao, 2023). An exact anchored representation replaces segment-specific propagation $e^{A\sum_t \Delta_t}$ by length-indexed weights $\Lambda^m = e^{Aam}$ at fixed $A, a$, moving selectivity into source and readout factors. The fixed propagation can eliminate dynamic transition products on a specified source schedule, but its coordinate factors depend on the residual clock. The design question is therefore quantitative: how much of that specialization can a coordinate-range contract retain, and what does subdivision cost?

This paper connects the removed work to the conditions for retaining it. Exact feasible-anchor intervals supply retaining anchors or certificates that further constant-anchor search cannot help. Temporal-order obstructions show why marginal increment statistics cannot settle this decision. On the measured tree, the exact feasible dyadic cut maximizes the transition products that specialization can remove; coarsening also increases local source work, so it need not minimize total execution cost. A sharp mean-span factor $2 - 2/T$ gives a cheap policy an oracle-relative guarantee under budget augmentation. At unchanged budget, the mean can subdivide to width two and lose all of that transition-product saving. The resulting design rule is to establish admissibility first and optimize a declared cost over the admissible choices. The cumulative-product identity and classical approximation tools are building blocks (Katsch, 2023; Yang et al., 2024; Cheney, 1966).

The evidence follows three comparisons. First, exact operation counts isolate specialization on the same source schedule. Second, matched GPU routes test its interaction with anchor policy and required preparation. Third, optimized physical implementations test practical cost at the same output-fidelity contract. In the current H100 primary panel, mean improves affine execution over deployed anchors by $1.658\times$ /$1.298\times$ and over width-two rescue by $1.135\times$ /$1.091\times$ for 130M/370M cores.



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.

On-demand source-only yields additional, smaller gains. Parallel physical precision controls nevertheless remain substantially faster. These comparisons establish a conditional computational benefit and its applicability, while distinguishing it from whole-algorithm or native-runtime superiority.

## 2 EXACT ANCHORED EXECUTION

For one channel with $N$ states, suppress batch/channel indices and write

$$h_t = F_t h_{t-1} + v_t, \quad F_t = e^{A\Delta_t}, \quad v_t = \Delta_t B_t u_t, \quad y_t = C_t h_t + D u_t. \tag{1}$$

Here $A = \mathrm{diag}(A_1, \ldots, A_N)$ is real and strictly negative, $\Delta_t > 0$, and $h_0$ may be nonzero. The source convention and input-dependent $B_t, C_t, \Delta_t$ are preserved. For a finite positive anchor $a$, define

$$\Lambda = e^{Aa}, \qquad r_t = \sum_{i=1}^{t} (\Delta_i - a), \qquad P_t = e^{Ar_t}, \quad P_0 = I. \tag{2}$$

**Proposition 1** (Exact anchored representation)**.** *In exact arithmetic, $q_0 = h_0$ and*

$$q_t = \Lambda q_{t-1} + P_t^{-1} v_t, \qquad h_t = P_t q_t \tag{3}$$

*recover Equation* (1)*, including its readout.*

The identity follows from $P_t \Lambda P_{t-1}^{-1} = e^{A\Delta_t}$. Unrolling gives

$$y_t = C_t P_t \Lambda^t h_0 + D u_t + \sum_{s=1}^{t} \underbrace{C_t P_t}_{\text{selective readout}} \underbrace{\Lambda^{t-s}}_{\text{fixed propagation}} \underbrace{P_s^{-1} \Delta_s B_s}_{\text{selective source}} u_s. \tag{4}$$

Selectivity remains in the outer factors. With cumulative clock $\tau_t = \sum_{i=1}^{t} \Delta_i$ and $\Pi_t = e^{A\tau_t}$,

$$P_t = \Pi_t \Lambda^{-t}, \qquad \Pi_t \Pi_s^{-1} = \Lambda^{(\tau_t - \tau_s)/a} = P_t \Lambda^{t-s} P_s^{-1}. \tag{5}$$

The powered form depends on both endpoints. With generated coefficients held fixed, a one-sided output gate cannot generally reproduce it for arbitrary sources; it remains a structured multiplicative mask (Appendix A.1). Changing $a$ preserves the exact function while changing scaling and prepared powers. Fixed propagation gives $N$ state-wise Toeplitz operators, not generally a stationary I/O convolution. Proposition 6 supplies a fixed-basis contraction condition; Appendix A.1 compares the prior algebra.

**Deriving a range benefit.** Raw prefix normalization fixes internal propagation to the identity but retains the cumulative clock in its factors. For residuals $r_j = c_j - ja$, the log-factor difference is $A_n(r_j - r_i)$; hence the factor's maximum/minimum ratio is $\exp(|A_n| R(a))$. For even $T$ increments alternating $\mu + \varepsilon, \mu - \varepsilon$, with $\mu > \varepsilon > 0$, raw clock span is $T\mu$, whereas $a = \mu$ gives residuals alternating between 0 and $\varepsilon$. Putting the same increments in two homogeneous runs instead gives span $T\varepsilon/2$. Thus anchoring removes drift while retaining sensitivity to temporal order. Appendix E tests this mechanism in FP32 with raw, centered and direct-physical controls; the mean guarantee below compares it with the best anchor.

**Physical boundaries.** A coordinate block starting at $s$ incorporates $F_s h_{s-1} + v_s$ as its first source and forms internal prefixes from $\Delta_{s+1}, \ldots, \Delta_{s+W-1}$. A singleton has zero internal span but still performs its physical transition. Physical summaries $(M_r, b_r)$, with $M_r = e^{A \sum_{t \in r} \Delta_t}$, require an input-dependent affine carry between blocks; coordinates never reset the physical state.

## 3 COORDINATE CONDITIONS AND ANCHOR DECISIONS

For a width-$W$ block starting at $s$, set $T = W - 1$, $d_i = \Delta_{s+i} > 0$, $c_0 = 0$, and $c_j = \sum_{i=1}^{j} d_i$. Internal span is

$$R(a) = \max_{0 \le j \le T} (c_j - ja) - \min_{0 \le j \le T} (c_j - ja). \tag{6}$$

For channel rate $\omega_h = \max_n |A_{h,n}|$, the planner accepts a node when $S_r = \max_h \omega_h R_{rh}(a_h) \le s_{\max}$ and otherwise bisects. For one state,

$$\kappa_2\Big(\mathrm{diag}_{0\le j\le T}\, e^{A_n(c_j - ja)}\Big) = e^{|A_n|R(a)}. \tag{7}$$

This is the conditioning of represented coordinates. It motivates the budget $z = s_{\max}/\omega_h$; complete output accuracy is assessed separately.

### 3.1 Exact feasibility and temporal information

**Proposition 2** (Finite-set feasibility). *For a nonempty finite collection of width-$W$ roots, let $m_k, M_k$ be the minimum/maximum sums of $k$ consecutive internal increments. For $W \ge 2$ and $0 < a_- \le a_+ < \infty$,*

$$\rho(a) = \max_r R_r(a) = \max_{1\le k<W} \{M_k - ka,\ ka - m_k\}. \tag{8}$$

*This continuous convex piecewise-affine function attains its minimum on $[a_-, a_+]$. A budget $z \ge 0$ is feasible exactly when*

$$\max\left(a_-, \max_k \frac{M_k - z}{k}\right) \le \min\left(a_+, \min_k \frac{m_k + z}{k}\right). \tag{9}$$

The result follows by expressing a prefix range through all pairwise differences. It covers shared anchors and restricted domains.

**Corollary 1** (Temporal-order obstruction). *Two internal intervals of lengths $k, \ell$ and mean increments $\mu_1, \mu_2$ imply*

$$\min_{a>0} R(a) \ge \frac{k\ell}{k+\ell}|\mu_1 - \mu_2|. \tag{10}$$

*For $m$ copies each of $\mu + d, \mu - d$ with $\mu > d > 0$, alternating order has optimum $d$, whereas two homogeneous runs have optimum $md$; a final neutral increment $\mu$ preserves both values.*

Thus identical marginal increment statistics can have arbitrarily different irreducible spans. Selected interval pairs give valid but incomplete rejection certificates (Appendix A.3).

**Corollary 2** (Single-root geometry). *For $T \ge 2$, $R^\star = \min_{a>0} R(a)$ has an optimum in $[\min_i d_i, \max_i d_i]$, and*

$$R^\star = 2 \min_{a,b\in\mathbb{R}} \max_{0\le j\le T} |c_j - aj - b| \tag{11}$$

$$= \max_{0\le i<j<k\le T} \left| c_j - \frac{k-j}{k-i}c_i - \frac{j-i}{k-i}c_k \right|. \tag{12}$$

*For $z \ge 0$, define*

$$L_z = \max_{i<j} \frac{c_j - c_i - z}{j-i}, \qquad U_z = \min_{i<j} \frac{c_j - c_i + z}{j-i}. \tag{13}$$

*Then $R(a) \le z$ iff $a \in [L_z, U_z]$; feasibility requires intersection with $(0, \infty)$.*

The optimum is the vertical width of the thinnest affine strip containing the accumulated clock. A finite-set alternation proof appears in Appendix A.4, building on classical uniform approximation (Cheney, 1966). At $T = 0$ every anchor has zero span; at $T = 1$, $R(a) = |d_1 - a|$ and $a = d_1$ gives zero. Figure 1 illustrates the geometry.

### 3.2 Mean policy: guarantee and boundary

For $T \ge 1$, define

$$\bar{a} = c_T/T, \quad B_j = c_j - j\bar{a}, \quad M = \max_j |B_j|, \quad S = \max_j B_j - \min_j B_j. \tag{14}$$

**Theorem 1** (Sharp mean-span bound). *For positive increments and unrestricted positive anchors,*

$$M \le R^\star \le S \le 2M \le 2R^\star. \tag{15}$$

*For every integer $T \ge 2$, the sharp horizon-dependent bound is*

$$R(\bar{a}) \le (2 - 2/T)R^\star. \tag{16}$$

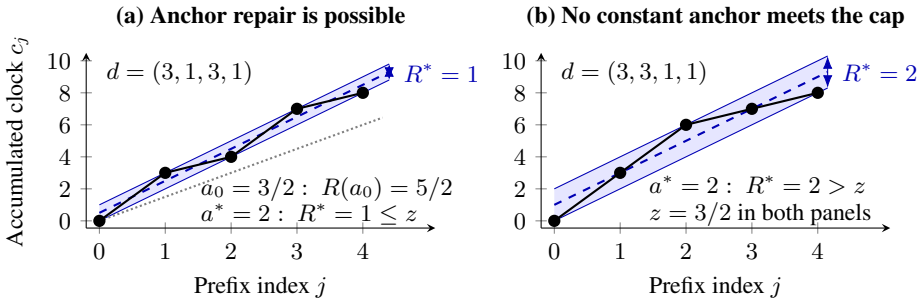


**Figure 1:** Equal increment multisets, different anchor feasibility. At $z = 3/2$, alternating increments admit a strip of width $R^\star = 1$, repairing $a_0 = 3/2$; reordered increments require $R^\star = 2$, exceeding the cap for every positive anchor.

The key observation is that mean residuals subtract the line interpolating the endpoint residuals of an optimal anchor. Both lie in an interval of width $R^\star$, giving $M \le R^\star$; the finite horizon sharpens the two-sided range (Appendix A.5).

**Partition consequence.** Let $K_{\text{mean}}(z), K_\star(z)$ count leaves for exact block mean and independently optimized positive anchors on a fixed complete dyadic tree of root width $W \ge 4$. With $\Gamma_W = 2 - 2/(W-1)$,

$$K_{\text{mean}}(\Gamma_W z) \le K_\star(z), \qquad z \ge 0. \tag{17}$$

Every oracle leaf at budget $z$ passes mean at $\Gamma_W z$, so mean cannot split it. The factor is sharp. This augments a log-range budget: a coordinate-condition allowance $C = e^{\omega z}$ becomes $C^{\Gamma_W}$, not $\Gamma_W C$. At the same budget, equality of partitions is guaranteed if every oracle leaf of width $w \ge 4$ satisfies $\Gamma_w R_B^\star \le z$ (Corollary 4). However, positive increments can give $K_{\text{mean}}(z) = W/2$ and $K_\star(z) = 1$ (Lemma 1). A sharp span approximation therefore does not imply a width-independent leaf-count approximation at unchanged budget.

**Computable decisions.** With division-free numerators $D_j = Tc_j - jc_T$, Equation (15) gives

$$\max_j D_j - \min_j D_j \le Tz \implies \text{mean feasible}, \tag{18}$$

$$\max_j |D_j| > Tz \implies R^\star > z. \tag{19}$$

Both use one scan after $c_T$ is known, assuming exact predicates or controlled rounding. For deployed anchor $a_0 > 0$, the exact intervals make the following exhaustive classification actionable:

$$\mathcal{A}_{\text{keep}} = \{R(a_0) \le z\}, \tag{20}$$

$$\mathcal{A}_{\text{repair}} = \{R(a_0) > z, \ R^\star \le z\}, \tag{21}$$

$$\mathcal{A}_{\text{const}} = \{R^\star > z\}. \tag{22}$$

A nonempty positive interval supplies a retaining anchor; an empty one certifies that further constant-anchor search cannot retain the root at this budget. Subdivision, budget and execution route are recurrence-preserving alternatives. Redesigning the spectrum of $A$ instead changes the model. Restricted/shared anchors and rounded means require their stated variants (Appendix A.6).

### 3.3 From span to a finite-range execution decision

The span characterization also decides whether explicit coordinate factors can fit a numerical range. Let $A_n = -\lambda_n$, $\lambda_n \ge 0$, $\omega = \max_n \lambda_n$, and $r_j = c_j - ja$. A constant statewise gauge gives ideal factors $P_{n,j} = \exp(-\lambda_n r_j + g_n)$. Gauge changes preserve $\Lambda$ and the physical recurrence when applied consistently to source and recovery.

**Proposition 3** (Centered factor range)**.** *Fix $a > 0$ and bounds $0 < m \le 1 \le F < \infty$. Both ideal factors $P_{n,j}$ and $P_{n,j}^{-1}$ can lie in $[m, F]$ for every $n, j$, using constant statewise gauges, if and only if*

$$\omega R(a) \le 2\beta, \qquad \beta = \min\{\log F, -\log m\}. \tag{23}$$

*When feasible, the common clock center $b = (\max_j r_j + \min_j r_j)/2$ with $g_n = \lambda_n b$ suffices.*

Thus Equation (13), with $z = 2\beta/\omega$ for $\omega > 0$, gives an exact anchor decision for this factor-range contract. The proposition is an elementary log-interval translation: its use here is to connect coordinate feasibility to a specified execution requirement.

### 3.4 A DIVISION-FREE ALL-ANCHOR OBSTRUCTION

With $D_j = Tc_j - jc_T$ from Equation (19), the exact predicate

$$\omega \max_j |D_j| > 178T, \qquad T \geq 1, \tag{24}$$

certifies that no constant anchor and no constant statewise gauge can place both ideal factors in the positive finite FP32 range. Indeed, $M \leq R(a)$ for every anchor, whereas twice the log of the largest finite FP32 value is less than 178. Captured FP32 coefficients admit an integer-arithmetic evaluation of this test. It identifies roots requiring a change beyond constant-anchor search; non-triggering roots remain unclassified. The threshold concerns factor range rather than a chosen planning cap. Appendix A.8 proves the statement and separates ideal range, rounded execution, and output accuracy.

### 3.5 USING THE GUARANTEES TO CHOOSE ANCHORS AND PARTITIONS

For one channel of a proposed block in exact arithmetic, let $S = R(\bar{a})$ and $M$ be its mean-residual span and maximum absolute residual from Equation (14). At a fixed budget $z$, the existing bounds give the following exact decision procedure:

$$\begin{array}{lll} S \leq z & \Rightarrow & \text{retain the block at its mean,} \\ M > z & \Rightarrow & \text{no constant anchor can retain the block,} \\ M \leq z < S & \Rightarrow & \text{resolve the feasible interval } [L_z, U_z]. \end{array} \tag{25}$$

A positive point supplies a retaining anchor; an empty positive intersection requires subdivision or another representation. With independent channel anchors, every channel must pass its rate-adjusted cap. The first two branches use a scan to settle the block without the full interval calculation. Resolving every remaining case and traversing a fixed dyadic tree yields the same partition as its unrestricted exact-anchor oracle. Correctness follows from the interval and mean bounds (Appendix A.7), with exact rational checks in Appendix E. Hull-based exact solvers also admit linear-width implementations; no solver-speed advantage is claimed.

These guarantees apply to stored and on-demand propagation alike. They distinguish sufficient budget augmentation from the unchanged-budget limitation. Once a partition is chosen, a fixed four-candidate sharing rule can reduce a materialized bank using directed admission and original-mean fallback (Appendix A.10); the choice to create that bank is tested separately.

## 4 EXECUTION CONSEQUENCE: LENGTH-INDEXED PROPAGATION

A general diagonal affine block maps $h \mapsto Dh + b$. Chronological composition is

$$(D_L, b_L) \star (D_R, b_R) = (D_R D_L,\ D_R b_L + b_R). \tag{26}$$

**Proposition 4** (Source-only composition)**.** *Inside one anchored coordinate block, a length-$m$ segment maps $q \mapsto \Lambda^m q + s$, with associative summary law*

$$(m_L, s_L) \star_\Lambda (m_R, s_R) = (m_L + m_R,\ \Lambda^{m_R} s_L + s_R). \tag{27}$$

*On a fixed tree, lengths are structural metadata. Prepared $\Lambda^m$ removes the need for dynamic transition-product vectors inside that tree.*

Expanding two affine maps proves the law; expanding three proves associativity. The dynamic transition vector becomes structural length metadata plus a prepared weight. The exact saving depends on which intermediate results the chosen scan actually consumes.

**Proposition 5** (Consumed transition work)**.** *Fix a source-update schedule with prepared weights at common $A, a$. After removing unused transition products, let $F$ be its scalar source FMAs and*

*$H$ its consumed transition multiplications. Keeping this source schedule, affine and source-only arithmetic per state component are*

$$C_{\text{affine}} = 2F + H, \qquad C_{\text{source}} = 2F, \qquad \Delta C = H. \tag{28}$$

*An FMA counts as two ordinary operations. Replacing transition products by their length-indexed coefficients preserves the exact source outputs.*

The result follows by induction along the source computation: each update receives the same coefficient and predecessors. It is an accounting consequence of specialization on a specified schedule, not a minimum-work claim over scans. Appendix B.1 gives the full derivation.

**Corollary 3** (The measured GPU trees)**.** *For a dyadic leaf $w = 2^k > 1$, the matched GPU executor's readout tree has $F = 3w - 2$ and $H = w - 2$. Consequently,*

$$C_{P,\text{read}} = 7w - 6, \qquad C_{S,\text{read}} = 6w - 4. \tag{29}$$

*Its terminal-source summary tree has $F = w - 1$ and $H = w - 1 - k$, giving $3(w-1) - k$ versus $2(w-1)$. Width one is dispatched directly with zero local-tree work; both counted trees tie at width two.*

Readout uses $w - 1$ upsweep, $w - 1$ downsweep and $w$ inclusive-source updates; $w - 2$ nonroot transition products are consumed. Summary omits one transition product per level. These schedules include identity-prefix operations but exclude unused transition work. Appendix B.3 treats a different Brent–Kung graph separately.

**Charging on-demand powers.** In the current two-tree implementation, each tree forms its own dyadic ladder. For $w = 2^k \geq 2$, the removed transition products total $2w - 3 - k$ per state, while the two ladders require $2(k-1)$ squarings. Thus the net reduction in this transition/power subset is $2w - 1 - 3k$: zero at width two and positive for dyadic widths at least four. This is an exact local count, including these generation costs. Appendix B.2 derives it; Appendix B.4 contrasts full coordinate/preparation ledgers under declared cost models.

**How much specialization survives the range constraint?** For a dyadic partition $\mathcal{P}$ with $K$ leaves of width at least two and total length $L$, summing both trees gives

$$C_S(\mathcal{P}) = 8L - 6K, \qquad H(\mathcal{P}) = 2L - 3K - \sum_{B \in \mathcal{P}} \log_2 w_B. \tag{30}$$

A feasible merge into width $w = 2^k \geq 4$ adds six source operations and removes $k + 1$ more transition products per state. Charging both on-demand ladders gives net saving $G = H - Q$, with $\Delta G = 3k - 5 > 0$. Thus the oracle maximizes both $H$ and $G$ at its cap; mean's budget-augmentation guarantee transfers to both, whereas its same-budget counterexample loses both savings entirely (Appendix B.2). Policy choice also depends on the cost contract: charging $\alpha$ per ordinary operation and $\beta$ per leaf, with other costs fixed, a feasible source-only merge is beneficial when $\beta > 6\alpha$ for prepared powers, but when $\beta > \alpha(12 - 2k)$ with both ladders charged. These exact additive-model thresholds connect feasibility to cost without predicting GPU latency.

**Which cost changes?** The core counts exclude coefficient preparation, coordinate transforms, interleaf physical carry and output contraction. Fresh GPU timing includes those required costs. Physical clock/source summaries can also omit transition vectors by generating $e^{A\sigma_R}$; anchoring makes the weights precomputable by length at common $A, a$. Both counted scan variants retain $O(Nw)$ work and $O(\log w)$ depth. Section 5.2 tests whether removing $H$ benefits the complete measured executor.

**Coefficient reuse and storage.** One dynamic numeric summary shrinks from $2N$ to $N$ scalars, while lengths remain structural metadata. Repeated lengths can share a propagation coefficient at common $A, a$. The appendix's width-2048 example reduces a retained coefficient payload from 261,312 to 704 bytes, but coordinate arrays can outweigh that saving. The full ledger separates coefficient reuse from total storage and physical recomputation. Section 5.2 tests the distinct decisions to share a materialized bank or omit it.

**Table 1:** H100 matched on-demand comparisons, including fresh preparation and SOURCE power generation. $D, R, M, H$ denote deployed, width-two rescue, mean and deployed-first mean fallback. Ratios above one favor the denominator. P130/P370 use $L = 2048$, root 16/cap 32 and three layers; adverse uses root 256/cap 8; long uses $L = 4096$, root 16/cap 32. All contrasts use eight complete documents except [6]adverse, which uses six common passing documents. Intervals and full SOURCE policy contrasts: Appendix C.

| Panel | PAIR $D/M$ | PAIR $R/M$ | PAIR $H/M$ | $M$: PAIR/SOURCE |
|---|---|---|---|---|
| 130M primary | 1.658 | 1.135 | 1.0026 | 1.0058 |
| 370M primary | 1.298 | 1.091 | 0.9998 | 1.0123 |
| 370M adverse[6] | 1.005 | 0.853 | 1.0494 | 1.2568 |
| 370M long | 1.291 | 1.065 | 1.0007 | 1.0187 |

## 5 STRUCTURAL GAINS AND MEASURED EXECUTION EFFECTS

**Panels and matched controls.** The H100 study has 48 primary settings (two checkpoints, three layers and eight documents; $L = 2048$, root width $W = 16$/log-span cap $s_{\max} = 32$), eight adverse settings (370M layer 39, root 256/cap 8) and eight long settings ($L = 4096$, the same layer, root 16/cap 32). There are 56 distinct captured cores, since adverse settings reuse eight primary cores. Batch one, $N = 16$ and all 1,536/2,048 channels are retained. One H100 80GB allocation recaptures the preceding A100 study's documents/checkpoints; it is not a new-document panel or a bit-identical cross-device input comparison. Appendix C gives the selection and full protocol.

Four policies are crossed with on-demand affine (PAIR) and source-only (SOURCE) execution. Within each policy, the pair shares plans, anchor bits and a common preparation contract; SOURCE's required power generation is timed. Physical controls include unchanged native and four parallel variants of its scan family: FP32, FP64 coefficient/source evaluation with FP32 scan/readout (COEFF64), FP32 coefficients with FP64 scan/readout (ACC64), and FP64 coefficient/source plus scan/readout (ALL64). A sequential ALL64 bridge retains the earlier comparator. Same-family contrasts isolate specific interventions; cross-family times compare complete implementations.

**Contract and estimator.** Every readout $C_t h_t$ and final-state element is checked against an FP64 recurrence: finite values, error $\le 2\times10^{-5}+2\times10^{-4}|r_i|$, and relative L2 $\le 2\times10^{-4}$. Fresh includes required planning, validation, packing, generation and execution from resident coefficients; prepared reuses compatible metadata. Capture, compilation, I/O and references are excluded. Qualification failures receive no timing. For paired routes $A, B$,

$$S_{A/B} = \exp\left(\frac{1}{D}\sum_{d=1}^{D}\frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\frac{1}{K}\sum_{k=1}^{K}\log\frac{t_{A,d\ell k}}{t_{B,d\ell k}}\right). \tag{31}$$

There are $K = 28$ balanced repetitions after two warmups. Models/panels remain separate; $\mathcal{L}$ has three primary layers or one diagnostic layer. Comparisons retain only documents passing every required layer in both routes. Pointwise descriptive 95% intervals resample those complete passing documents ($D \le 8$; 10,000 draws); repetitions and channels are not independent inputs.

### 5.1 POLICY GAINS AFTER WIDTH-TWO RESCUE

Width-two rescue preserves deployed acceptances and repairs only rejected width-two nodes. Full mean tries the block mean at every reached node. Deployed-first mean fallback tries deployed, then mean, at every node. The latter can change the partition, unlike fixed-partition coefficient sharing.

Table 1 shows that mean improves primary PAIR time over deployed by $1.658\times$ /$1.298\times$ and over rescue by $1.135\times$ /$1.091\times$. SOURCE gives corresponding gains of $1.665\times$ /$1.301\times$ and $1.138\times$ /$1.092\times$. These effects survive the rescue control and occur in both representations. Fallback and mean remain close; they do not establish a large additional policy gain. The earlier exact census independently found leaf reductions beyond rescue (Appendix D.2).

At length 4096, PAIR mean improves over deployed/rescue by $1.291\times$ /$1.065\times$. Adverse settings differ: mean and fallback each fail two of eight documents in both representations, whereas deployed and rescue pass all eight. Among the six common passing documents, rescue is faster than mean

in PAIR and SOURCE ($R/M = 0.853/0.961$). This supports configuration-specific choices rather than a universal mean ranking. Root and cap changed jointly; no separate threshold is inferred.

### 5.2 Representation and coefficient-storage effects

At a fixed mean plan, fresh PAIR/SOURCE ratios are $1.0058\times$ $/1.0123\times$ in the primary groups and $1.0187\times$ at length 4096. The primary 95% intervals are $[1.0052, 1.0064]$ and $[1.0114, 1.0132]$. All 256 policy–setting pairs have identical recorded output/state hashes. Together with the charged ladder counts, this completes the matched on-demand comparison: the arithmetic specialization survives, while its primary total-time effect is small. The adverse mean ratio is $1.257\times$ on six passing documents and must not be promoted to an eight-document result.

Earlier implementations give useful context rather than interchangeable estimates. The four-core factorial reported fresh source-only gains of $1.163$–$1.172\times$, whereas packed sweeps included reversals (Appendices D.2, D.2). The preceding A100 policy/storage study found on-demand execution faster than materialized banks, with fallback's bank-storage advantage largely disappearing on demand (Appendix D.5). Shared coefficients offer both storage and recomputation choices; neither a smaller bank nor fewer local operations determines total latency alone.

### 5.3 What range certificates decide in execution

The earlier explicit-factor study certifies all-anchor obstructions in $1{,}635{,}971/4{,}718{,}592$ width-16 roots for 130M and $173{,}750/6{,}291{,}456$ for 370M. These witnesses rule out retaining those roots under the positive-finite bidirectional factor contract by anchor tuning or constant statewise centering. All three tested fixed-width-16 routes fail the case-level fidelity gate; uniform width-two and adaptive mean pass all 48 cases. Adaptive subdivision improves fresh time over uniform width two by $1.326\times$ $/1.988\times$ and reduces retained plan-plus-bank storage by 31.15%/65.03% (Appendix D.3). The certificate thus ends futile anchor search and identifies a required change of block or execution contract. Direct physical computation is outside that factor contract. A retained coordinate block still needs the output-fidelity check used here.

### 5.4 Strong physical controls and practical cost

Parallel COEFF64 and ALL64 pass all 64 settings. Native and parallel FP32 pass 54/64, ACC64 passes 55/64, and the eight anchored routes pass 504/512 setting–route combinations. All 37 qualification failures are finite readout-tolerance exceedances in 1–11 elements; final states and relative-L2 checks pass. This is a stricter elementwise failure, not observed divergence or a measured downstream quality loss. COEFF64/ALL64 change coefficient evaluation and exponential implementation, so these contrasts do not identify precision alone as the cause.

Table 2 places the internal gains on an absolute scale. Primary mean SOURCE takes 4.048/3.479 ms, versus 0.188/0.249 ms for parallel COEFF64 and 0.230/0.307 ms for parallel ALL64. Both strong controls pass the same primary documents. Mean SOURCE is therefore $17.60\times$ $/11.33\times$ slower than parallel ALL64. Its advantage over sequential ALL64 is comparator-specific; the parallel bridge is $34.16\times$ $/25.41\times$ faster than that sequential route. At length 4096, parallel ALL64 remains $9.00\times$ faster than mean SOURCE.

The primary median peak tensor increments are 360.40/405.05 MiB for mean and 12.19/16.25 MiB for the parallel physical controls. These are allocator increments above resident inputs, not total device memory or shared-memory usage. They do not support a whole-memory advantage. The completed controls establish internal policy/specialization benefits while leaving native-competitive execution unestablished.

## 6 Related work

GateLoop and GLA share cumulative-product algebra (Katsch, 2023; Yang et al., 2024); Mamba supplies selective recurrences and fused scan (Gu & Dao, 2023). Fixed-transition scans (Smith et al., 2023), LTI contraction (Gu et al., 2022a;b), and semiseparable masks (Dao & Gu, 2024) are related building blocks. Thin-strip geometry uses classical uniform affine approximation (Cheney,

**Table 2:** H100 absolute fresh latency (geometric mean, ms). Primary panels: $L = 2048$, root 16/cap 32; long: 370M layer 39, $L = 4096$. Every value uses eight complete documents unless a superscript gives a smaller count. Native, FP32 and ACC64 counts differ because failed routes are excluded, so those row-wise means are not matched ratios. COEFF64 and ALL64 pass all 64 settings, including adverse. The sequential control is a historical bridge, not the strongest physical baseline.

| Route | 130M primary | 370M primary | 370M long |
|---|---|---|---|
| Mean PAIR | 4.072 | 3.522 | 5.335 |
| Mean SOURCE | 4.048 | 3.479 | 5.237 |
| Official native | 0.162 | $0.210^{6}$ | $0.378^{2}$ |
| Parallel FP32 | 0.153 | $0.210^{6}$ | $0.396^{2}$ |
| Parallel COEFF64 | 0.188 | 0.249 | 0.482 |
| Parallel ACC64 | 0.187 | $0.252^{6}$ | $0.485^{3}$ |
| Parallel ALL64 | 0.230 | 0.307 | 0.582 |
| Sequential ALL64 | 7.858 | 7.803 | 15.851 |

1966), and key cover uses interval stabbing. The contribution concerns exact anchor decisions, sharp policy guarantees and their execution consequences. Appendix A.1 separates these results from the known identities.

## 7 Discussion

**A conditional computational result.** Fixed propagation replaces consumed transition products by length-indexed coefficients. Exact feasibility and sharp policy bounds characterize where that specialization can be retained; the merge ledger states when fewer boundaries can offset greater local source work. The range certificate decides whether further constant-anchor tuning can retain a block at its span budget, independently of the execution implementation.

**From operation counts to implementation.** The prepared-weight theorem, generation-inclusive local ledger and complete runtime are different cost contracts. Appendix B.4 counts transforms and preparation for alternative scans: some reduce ordinary arithmetic relative to a specified native source schedule, but an equally improved physical-clock comparator can be cheaper still. Thus these counts do not establish smaller total native work. The H100 comparisons likewise establish local representation and policy effects, while optimized parallel physical controls win overall. Turning local savings into competitive execution requires further algorithm and implementation work; the measured gap is not attributed solely to missing coding optimization, nor is future superiority guaranteed.

**Applicability and empirical scope.** Propagation is shared at common $A, a$; plans also bind layout and transitions. Materialized bank savings are distinct from full memory. Coordinate span controls factor range, not complete output error. Empirical evidence covers selected layers of two checkpoints, batch one, $N = 16$, nested lengths and particular A100/H100 sessions. H100 reuses the selected documents but recaptures coefficients, so cross-device timings are not controlled on identical arrays. The observations support conditional choices without an evaluated automatic selector. Anchor, gauge and partition changes preserve the recurrence; redesigning $A$'s spectrum changes the model.

## 8 Conclusion

Anchored execution provides a precisely countable specialization and an exact admissibility problem. Feasible intervals, sharp mean guarantees and partition separation connect the available arithmetic reduction to anchor and subdivision choices. Matched measurements establish policy gains beyond width-two rescue and smaller on-demand source-only gains. Strong physical controls delimit practical applicability: the present implementations do not provide competitive total execution cost. The contribution is the quantitative connection between removable computation, its coordinate conditions and tested execution decisions.

## AI USE STATEMENT

The author proposed the core mathematical approach and set the research direction, experimental priorities and manuscript structure. Generative AI tools assisted mathematical formulation and proof exposition, experiment planning and critique, experimental and verification code, construction of synthetic controls, interpretation of recorded results, related-work checking, and manuscript editing and typesetting. Reported observations come from recorded executions and the stated CPU checks; generated observations do not replace missing experiments. Validation includes exact-arithmetic checks, saved-output comparisons with independently recomputed FP64 recurrences, and reconstruction of timing statistics. AI tools also assisted environment diagnosis, archive-integrity checks, protocol and implementation development, failure analysis, and organization of the supplementary material. These activities do not constitute independent-hardware replication; metrics-only exports cannot support a new elementwise audit of excluded arrays. The author is responsible for the final claims, code, references and disclosure. AI tools are not coauthors.

## REPRODUCIBILITY STATEMENT

Appendices A and B.1 give the assumptions, proofs and arithmetic contracts supporting the main claims. Appendix C specifies the H100 selection, hardware, compiler, timing and fidelity contracts. Appendices D and E retain complementary controls; Appendix F maps the evidence to the separate *Detailed Supplement*. The anonymous code supplement provides compact numerical records and CPU builders for 34 retained table files from P3R–P3W, finite exact/arithmetic checks, and the preserved P3W GPU computational source with a captured-input adapter. Its README distinguishes raw-record reaggregation from retained intervals and summary-level reconstruction. Pretrained weights, corpus assets and raw paper tensors are excluded. Metrics-only records cannot support a new elementwise audit of omitted outputs; the adapter does not recreate the historical input panel or memory protocol. No new GPU experiment is reported by this packaging.

## REFERENCES


Albert Gu and Tri Dao. *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. arXiv:2312.00752v2, 2023; revised 2024, Sections 3.2–3.3.

Albert Gu, Karan Goel, and Christopher Ré. *Efficiently Modeling Long Sequences with Structured State Spaces*. ICLR, 2022. arXiv:2111.00396.

Tri Dao and Albert Gu. *Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality*. ICML, 2024. arXiv:2405.21060.

Tobias Katsch. *GateLoop: Fully Data-Controlled Linear Recurrence for Sequence Modeling*. arXiv:2311.01927v1, 2023, Sections 3.1–3.3.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. *Gated Linear Attention Transformers with Hardware-Efficient Training*. ICML, 2024. arXiv:2312.06635v6, Sections 4.1–4.3.

Guy E. Blelloch. *Prefix Sums and Their Applications*. Technical Report CMU-CS-90-190, Carnegie Mellon University, 1990.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second edition, SIAM, 2002.

Daniel Y. Fu, Hermann Kumbong, Eric Nguyen, and Christopher Ré. *FlashFFTConv: Efficient Convolutions for Long Sequences with Tensor Cores*. arXiv:2311.05908, 2023.

Leo Gao et al. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. arXiv:2101.00027, 2020.

EleutherAI. *Language Model Evaluation Harness*, release v0.4.5. Software release; the full-quality protocol in the supplement specifies exact task and input bindings.

Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. *Simplified State Space Layers for Sequence Modeling*. ICLR, 2023. arXiv:2208.04933.

Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. *On the Parameterization and Initialization of Diagonal State Space Models*. NeurIPS, 2022. arXiv:2206.11893.

E. W. Cheney. *Introduction to Approximation Theory*. McGraw–Hill, 1966. Classical background on uniform approximation and alternation.

Richard P. Brent and H. T. Kung. *A Regular Layout for Parallel Adders*. IEEE Transactions on Computers, C-31(3):260–264, 1982. Corrected author reprint: `https://maths-people.anu.edu.au/~brent/pd/rpb060.pdf`.

**Reading the supporting material.** The appendices in this PDF contain the proofs, cost contracts, current H100 protocol and results, and the complementary controls needed for the main claims. The separate *Detailed Supplement* retains the full earlier experimental tables and protocols, oracle-leaf audit, and study-specific provenance. It introduces no new observations. All central proofs and the current strong physical-control results are retained here; the supplement is for inspecting the underlying detail. The study identifiers denote different protocols, not additional independent datasets.

# A Essential proofs and coordinate decisions

This appendix contains the proofs and qualifications needed for the main-text theory. It consolidates the representation, feasibility, policy, and finite-range arguments. The work and storage counts for specific execution graphs are derived separately in Appendix B.1.

## A.1 Exact coordinates, physical boundaries, and prior algebra

*Proof of Proposition 1.* Fix finite positive $a, \Delta_t$, a finite horizon, and real diagonal $A$. The exponentials are invertible and commute. With the definitions in Equation (2),

$$P_t \Lambda P_{t-1}^{-1} = \exp\left( A \left[ \sum_{i=1}^{t} (\Delta_i - a) + a - \sum_{i=1}^{t-1} (\Delta_i - a) \right] \right) = e^{A\Delta_t}.$$

Multiplying Equation (3) by $P_t$ therefore gives $h_t = e^{A\Delta_t} h_{t-1} + v_t$. Since $P_0 = I$ and $q_0 = h_0$, induction proves equality at every time, including readout and final physical state. The identity allows any fixed physical source $v_t$; the experiments retain $v_t = \Delta_t B_t u_t$. □

For a block starting at $s$, absorb its first physical step into $z_0 = F_s h_{s-1} + v_s$, and put $z_j = v_{s+j}$, $c_j = \sum_{i=1}^{j} \Delta_{s+i}$ for $j \geq 1$, and $c_0 = 0$. With $P_j = e^{A(c_j - ja)}$,

$$h_{s+j} = P_j \sum_{i=0}^{j} \Lambda^{j-i} P_i^{-1} z_i. \tag{32}$$

At $j = 0$ this includes the complete incoming-state contribution. A singleton has no internal drift but is not a state reset. The physical block map remains $h_{\text{out}} = M_r h_{\text{in}} + b_r$, where $M_r = e^{A \sum_{t \in r} \Delta_t}$ is input dependent; adjacent block maps require affine physical carry.

A constant invertible diagonal gauge $\widetilde{P}_t = P_t D$ preserves the recurrence only with $\widetilde{q}_0 = \widetilde{P}_0^{-1} h_0$, consistently transformed sources $\widetilde{P}_t^{-1} v_t$, and recovery $h_t = \widetilde{P}_t \widetilde{q}_t$. In particular, centering the factors generally makes $\widetilde{P}_0 \neq I$. When the homogeneous incoming-state response is computed separately, its physical value is retained and the paired gauge factors apply to the inhomogeneous contribution.

**Correspondence with existing algebra.** For $\Pi_t = F_t \cdots F_1$ and $\Phi_{t,s} = F_t \cdots F_{s+1}$, invertibility gives $\Phi_{t,s} = \Pi_t \Pi_s^{-1}$. This is the prefix-product separation appearing in GateLoop (Katsch, 2023, Section 3.3, Equations 14–19). Writing $z_t = \Pi_t^{-1} h_t$ gives $z_t = z_{t-1} + \Pi_t^{-1} v_t$. Introducing $\Lambda = e^{Aa}$, $P_t = \Pi_t \Lambda^{-t}$, and $q_t = \Lambda^t z_t$ yields the anchored recurrence. Unrolling it gives

$$h_t = P_t \Lambda^t h_0 + P_t \sum_{s=1}^{t} \Lambda^{t-s} P_s^{-1} v_s. \tag{33}$$

The non-unit anchor retains reference decay inside propagation instead of putting the entire cumulative decay in the outer factors. This identifies the overlap with known algebra; the identity is not claimed as a first discovery.

Mamba's selective coefficients generally remove a stationary input–output kernel (Gu & Dao, 2023, Sections 3.2–3.3). Equation (33) retains selective outer factors and generally $N$ state-wise operators, so it does not restore one stationary I/O convolution. GLA also reorganizes cumulative diagonal products and develops log-space and chunkwise evaluation (Yang et al., 2024, Sections 4.1–4.3); a comparison with raw prefix factors is not a stability or speed comparison against that stabilized method. The analysis here concerns constant-anchor range, conditional specialization, and the named implementations.

**Powered propagation and the scope of a mask.** Write $\kappa_n = (\Lambda)_{nn} = e^{A_n a}$ for propagation eigenvalues, distinct from decay rates $\lambda_n = -A_n$; use $\kappa$ in the scalar case. The positive diagonal $\Lambda$ has an unambiguous real logarithm. For $\tau_t = \sum_{i=1}^{t} \Delta_i$ and $G_i = e^{A(\Delta_i - a)}$, commutation proves

$$\Phi_{t,s} = \Lambda^{(\tau_t - \tau_s)/a} = \Lambda^{t-s} \prod_{i=s+1}^{t} G_i = P_t \Lambda^{t-s} P_s^{-1}. \tag{34}$$

The empty product is $I$; unrolling includes $\Phi_{t,0} h_0$. For constant scalar $G_i = g$, coefficients become $(\kappa g)^{t-s}$; for varying $g_i$, replacing the interval product by $g_t^{t-s}$ is generally wrong. For fixed input-generated coefficients and arbitrary injected sources, a one-sided output gate has entries $m_t \kappa^{t-s}$, whereas the anchored operator has entries $\kappa^{t-s} p_t / p_s$. Equality for every source first forces $m_t = 1$ on the diagonal, then fails for nonconstant $p_t/p_s$. The two-sided factorization $D_p T_\kappa D_p^{-1}$ is exact. Its multiplier satisfies $M_{t,t} = 1$ and $M_{t,s} M_{s,r} = M_{t,r}$, so it is a structured multiplicative mask in the broader sense also used by SSD (Dao & Gu, 2024), not an arbitrary mask or solely an output-amplitude gate. Altering the auxiliary anchor changes $\kappa_n$ and $G_{i,n}$ together while preserving their product. Learning an unconstrained replacement for $G_i$ instead changes the model.

### A.2 Source composition and conditional I/O contraction

*Proof of Proposition 4.* Composition of two consecutive segments gives

$$\Lambda^{m_R}(\Lambda^{m_L} q + s_L) + s_R = \Lambda^{m_L + m_R} q + \Lambda^{m_R} s_L + s_R.$$

For three segments the source is $\Lambda^{m_3+m_2} s_1 + \Lambda^{m_3} s_2 + s_3$, independent of parenthesization; $(0, 0)$ is the identity. On a fixed tree, lengths are topology, not dynamic state. Upsweep stores zero-state subtree sources; downsweep gives a right child the incoming value $\Lambda^{m_L} q_{\text{in}} + s_L$. Thus every prefix is recovered with arbitrary incoming state. □

This removes transition-product vectors on the stated source graph after supplying its coefficients. The physical family also has a clock/source summary $(\sigma_L, b_L) \star (\sigma_R, b_R) = (\sigma_L + \sigma_R, e^{A\sigma_R} b_L + b_R)$. Two dynamic vectors are therefore not necessary for every physical algorithm; anchoring makes the weights depend on length at shared $A, a$. The measured work graphs and their generation charges are separate from this algebraic proof.

For $\widetilde{b}_t = P_t^{-1} \Delta_t B_t$ and $\widetilde{c}_t = C_t P_t$, the selective coefficient $\sum_n \widetilde{c}_{t,n} \kappa_n^{t-s} \widetilde{b}_{s,n}$ generally depends on both endpoints. State-wise block FFT evaluation costs $O(N \sum_B w_B \log(2 w_B))$ before transformations, planning, and carry. Removing the state axis requires additional structure.

**Proposition 6** (Sufficient fixed-basis contraction)**.** *For fixed $A, a$, suppose $h_0 = 0$ and $\widetilde{b}_{t,n} = \sum_{p=1}^{r_b} \alpha_p(t, x) b_{p,n}$, $\widetilde{c}_{t,n} = \sum_{q=1}^{r_c} \gamma_q(t, x) c_{q,n}$, with $b, c$ independent of input and time. Then*

$$K_{qp}[\ell] = \sum_n c_{q,n} b_{p,n} \kappa_n^{\ell}, \qquad y_t - D u_t = \sum_{q,p} \gamma_q(t, x) [K_{qp} * (\alpha_p u)]_t. \tag{35}$$

*Proof.* Substitute both expansions into $\sum_{s \le t} \sum_n \widetilde{c}_{t,n} \kappa_n^{t-s} \widetilde{b}_{s,n} u_s$. Interchanging finite sums leaves, for each $q, p$, the convolution kernel displayed above, multiplied by the corresponding input and output gates. The kernel is reusable at common $A, a$ and input-independent bases $b, c$. □

With spectra and gates available, transform each of $r_b$ gated inputs once, multiply and sum the $r_b r_c$ frequency-domain pairs by output basis, then inverse-transform $r_c$ outputs. Application work is $O((r_b + r_c) L \log L + r_b r_c L)$; direct kernel construction costs $O(N r_b r_c L)$. Gate construction and the nonzero-state response $C_t P_t \Lambda^t h_0$ are additional. Factoring each test input independently would not establish a fixed reusable bank. The LTI structural control has a fixed physical kernel whose anchor factors cancel; its gain is established state contraction and reuse. An efficient frequency-reduced comparator can already perform that contraction. Changing the coefficients invalidates the bank. Finally, $c^\top F b = \ell c^\top b$ for all $b, c$ implies $F = \ell I$ by testing basis vectors; general diagonal spectra do not satisfy that scalar-identity factorization with unchanged I/O factors. This is a comparison to that layer form, not the general semiseparable framework.

### A.3 FINITE-SET FEASIBILITY, TEMPORAL OBSTRUCTION, AND EXACT WITNESSES

*Proof of Proposition 2.* A finite residual range is the maximum of its ordered pairwise differences. A pair separated by $k$ internal increments contributes $(c_j - c_i) - ka$ or its negative. Taking extrema over pairs and roots gives Equation (8). A finite maximum of affine functions is continuous, convex, and piecewise affine, so it attains its minimum on the stated nonempty compact domain. Requiring each term to be at most $z$ yields $a \geq (M_k - z)/k$ and $a \leq (m_k + z)/k$; intersecting these inequalities with $[a_-, a_+]$ proves Equation (9). □

Equivalently, requiring every lower endpoint to be below every upper endpoint gives

$$\rho^\star = \max\left\{0, \max_k(M_k - ka_+), \max_k(ka_- - m_k), \max_{k,\ell}\frac{\ell M_k - km_\ell}{k+\ell}\right\}. \tag{36}$$

For a single positive-increment root, an unrestricted positive optimum lies in $[\min d_i, \max d_i]$: outside this interval all residual increments have one sign and moving toward it decreases the range. For independent channel domains, minimizing $\max_h \omega_h R_h(a_h)$ therefore gives $\max_h \omega_h \min_{a_h} R_h(a_h)$ by combining individual minimizers. A shared anchor or coupled domain does not permit that step.

*Proof of Corollary 1.* For internal intervals $I, J$ of lengths $k, \ell$ and sums $S_I, S_J$, prefix differences give $|S_I - ka|, |S_J - \ell a| \leq R(a)$. Hence

$$R(a) \geq \frac{|\ell S_I - kS_J|}{k+\ell} = \frac{k\ell}{k+\ell}|\bar{d}_I - \bar{d}_J|.$$

Alternating $\mu + d, \mu - d$ yields residuals $0, d, 0, d, \ldots$ at $a = \mu$, and the two singleton values give the matching lower bound $d$. Two homogeneous runs of length $m$ have span $md$ at that anchor, with the two runs supplying the same lower bound. A trailing neutral increment has zero residual and leaves both bounds unchanged. □

If $|c_j - j\mu| \leq \epsilon$, endpoints and the range triangle inequality also give $(T|\mu - a| - \epsilon)_+ \leq R(a) \leq 2\epsilon + T|\mu - a|$. An anchor can remove systematic drift without eliminating the temporal obstruction.

**Restricted and exact certificates.** For any fixed subset $\mathcal{P}_0$ of interval pairs,

$$B_T = \max_{(I,J)\in\mathcal{P}_0}\frac{|I||J|}{|I|+|J|}|\bar{d}_I - \bar{d}_J|, \qquad B_0 = \frac{\max d_i - \min d_i}{2} \tag{37}$$

are lower bounds for $R^\star$. Thus $\omega \max(B_0, B_T) > s_{\max}$ is an all-anchor rejection; equality or a smaller bound is undecided. Maximizing over all interval pairs recovers the unrestricted optimum: feasibility requires every lower endpoint in the pairwise interval to be below every upper endpoint, giving precisely that maximum. This is another oracle formulation, not an independent predictor. The empirical restricted screen fixes seven sibling pairs before capture, retaining odd interval lengths. Even all dyadic sibling pairs can be incomplete: on $(1, 1, 1, 2)$ they give at most $1/2$, while $R^\star = 3/4$ at $a = 5/4$ by comparing the length-three mean-one interval to the final singleton.

Exact stored FP32 coefficients are dyadic data. With orientation $\ell S_I - kS_J \geq 0$, a proposed optimum certificate is

$$a^\star = (S_I + S_J)/(k+\ell), \qquad z = (\ell S_I - kS_J)/(k+\ell).$$

The interval bound proves $R^\star \geq z$; checking the complete prefix range $R(a^\star) = z$ in rational arithmetic proves attainment. Constant increments use their common value and zero range. A screen followed by a verified oracle on every undecided root gives the same binary verdict as the oracle, but need not return optimum anchors after early rejection. For reversed increments, $c'_j = c_T - c_{T-j}$, so residuals are a translated sign reversal and $R_{\mathrm{rev}}(a) = R(a)$. Arbitrary permutation has no such invariance. These operations concern captured internal increments and do not reset or permute the first physical transition.

### A.4 THIN-STRIP GEOMETRY AND THREE-PREFIX CHARACTERIZATION

*Proof of Corollary 2.* For fixed $a$ and $z \geq 0$, $R(a) \leq z$ iff $-z \leq (c_j - c_i) - (j-i)a \leq z$ for every $i < j$. Dividing by $j - i > 0$ and intersecting gives $a \in [L_z, U_z]$. Intersecting with $(0, \infty)$ imposes positive anchors. The compact positive optimum interval was proved above.

For fixed real $a$, centering the residual interval gives $\min_b \max_j |c_j - aj - b| = R(a)/2$. The optimum slope may be chosen positive, so $E^\star = \min_{a,b} \max_j |c_j - aj - b| = R^\star/2$ and attainment follows. Define $\delta_{ijk} = c_j - \frac{k-j}{k-i}c_i - \frac{j-i}{k-i}c_k$. The same expression in residuals $r_j = c_j - ja$ is a residual minus a convex combination of two others. Thus

$$\max_{i<j<k} |\delta_{ijk}| \leq R^\star. \tag{38}$$

Let $\ell(j) = a^\star j + b^\star$ attain $E^\star$, with $e_j = c_j - \ell(j)$. If $E^\star = 0$, every chord deviation is zero. Otherwise active residuals $|e_j| = E^\star$ must contain three ordered indices with alternating signs. If they do not, their signs have at most one change. A nonzero constant, or an affine function with its zero strictly between the two sign groups, then supplies $p$ satisfying

$$\operatorname{sign} p(j) = \operatorname{sign} e_j \quad \text{at every active index.} \tag{39}$$

For sufficiently small $\varepsilon > 0$, every active residual $e_j - \varepsilon p(j)$ decreases in magnitude, while all inactive ones stay strictly below $E^\star$ because the index set is finite. This contradicts optimality. At an alternating triple $e_i = e_k = E^\star$, $e_j = -E^\star$, the chord has value $\ell(j) + E^\star$ but $c_j = \ell(j) - E^\star$, so $|\delta_{ijk}| = 2E^\star = R^\star$. The opposite orientation is identical. This proves both equations by the finite-set degree-one alternation argument (Cheney, 1966). □

For $T = 0$, all residuals are zero. For $T = 1$, $R(a) = |d_1 - a|$ and the positive anchor $d_1$ attains zero; no empty triple maximum is used. These coordinate edge cases preserve the physical first step.

### A.5 SHARP MEAN-SPAN GUARANTEE

*Proof of Theorem 1.* Let $a^\star$ attain $R^\star$ and put $r_j = c_j - ja^\star$. Since $r_0 = 0$,

$$B_j = r_j - \frac{j}{T} r_T. \tag{40}$$

Both $r_j$ and the interpolation $(j/T)r_T$ belong to the residual interval of width $R^\star$. Hence $M \leq R^\star$. The positive mean is an allowed anchor, giving $R^\star \leq S$; all $B_j$ lie in $[-M, M]$, giving $S \leq 2M$. This proves the chain in Equation (15).

For the sharper factor, $B_0 = B_T = 0$. If an endpoint attains a maximum or minimum, all mean residuals have one sign and $S = M \leq R^\star$. Otherwise the extrema occur at distinct interior indices $p, q$, so

$$S = |B_p - B_q| \leq |r_p - r_q| + \frac{|p-q|}{T}|r_T| \leq \left(2 - \frac{2}{T}\right) R^\star, \tag{41}$$

since $|p - q| \leq T - 2$ and $|r_T - r_0| \leq R^\star$. For $T = 2$, $(3, 1)$ attains $R(\bar{a}) = R^\star = 1$. For every $T \geq 3$, take

$$d = (3, 1, \underbrace{2, \ldots, 2}_{T-3 \text{ terms}}, 3). \tag{42}$$

At anchor two its residuals are $0, 1, 0, \ldots, 0, 1$, giving span one; the singleton values three and one give the matching lower bound. The mean $2 + 1/T$ gives $B_1 = 1 - 1/T$, $B_j = -j/T$ for $2 \leq j \leq T - 1$, and $B_T = 0$, hence $R(\bar{a}) = 2 - 2/T$. The factor is sharp for every stated horizon. □

Because $D_j = TB_j$, Equations (18)–(19) are exactly $S \leq z$ and $M > z$. They concern the current exact mean. A frozen calibration mean, restricted/shared domain, or rounded anchor needs its own admission test.

### A.6 DYADIC PARTITIONS: EXACT OPTIMUM, SEPARATION, AND AUGMENTED BUDGETS

Fix a complete dyadic tree of width $W = 2^q$. Every leaf may choose an independent positive constant anchor; its first physical increment is excluded only from internal span. Let $\mathcal{O}(z)$ be the stop-at-first-oracle-feasible cut, and $\mathcal{M}(z)$ the cut testing each current exact mean; singletons are accepted directly. At a feasible node, one leaf is minimal. At an infeasible node, every admissible cut must cover both prescribed children, whose optimal cuts are obtained inductively. Thus $\mathcal{O}(z)$ minimizes leaf count over this tree. This is not optimization over arbitrary segmentations, shared anchors, fidelity, or latency.

Recomputed child means need not inherit parent acceptance. At cap one, the width-eight physical increments $(2, 3, 1, 3, 1, 2, 2, 2)$ have root internal mean two and span one, but their left width-four child has internal increments $(3, 1, 3)$ with mean $7/3$ and span $4/3$. Retaining a parent's unchanged feasible anchor on a child is a different operation.

**Lemma 1** (No uniform leaf-count approximation for the mean)**.** *For every dyadic $W \geq 4$, strictly positive physical increments and a budget exist for which $K_{\mathrm{mean}} = W/2$ and $K_\star = 1$ on the complete dyadic tree.*

*Proof.* Choose $\mu > d > 0$, $d < z < 4d/3$, any $\Delta_0 > 0$, and $\Delta_i = \mu + (-1)^{i+1}d$ for $1 \leq i < W$. Every aligned width-$w \geq 4$ node starts at an even index; its $T = w - 1$ internal increments alternate from $\mu + d$ to $\mu + d$, with $c_j = \mu j + d1_{\{j \text{ odd}\}}$. Anchor $\mu$ attains span $d$, and the two increment values give the matching lower bound. Its mean is $\mu + d/T$, whose residuals are $d(1_{\{j \text{ odd}\}} - j/T)$. The maximum at $j = 1$ and minimum at $j = T - 1$ give $R(\bar{a}) = 2d(w - 2)/(w - 1) \geq 4d/3 > z$. Mean rejects every node of width at least four and admits every width-two node, whereas the oracle admits the root. The ratio is unbounded as $W$ grows in exact arithmetic. □

The offset $d/T$ can round away at sufficiently long fixed-precision horizons; this is an exact-arithmetic separation.

For dyadic widths define $\Gamma_1 = \Gamma_2 = 1$ and

$$\Gamma_w = 2 - \frac{2}{w - 1} \quad (w \geq 4). \tag{43}$$

**Corollary 4** (Budget augmentation and oracle-leaf margins)**.** *For $z \geq 0$, $\mathcal{M}(\Gamma_W z)$ coarsens $\mathcal{O}(z)$, so*

$$K_{\mathrm{mean}}(\Gamma_W z) \leq K_\star(z). \tag{44}$$

*For each fixed dyadic $W \geq 4$, the multiplier cannot be uniformly decreased. At unchanged budget, $\mathcal{M}(z)$ refines $\mathcal{O}(z)$, with equality iff every oracle leaf passes its own mean. The sufficient condition for an oracle leaf $B$ of width $w_B \geq 4$ is*

$$\Gamma_{w_B} R^\star_B \leq z. \tag{45}$$

*Writing $\mathcal{Q}(z) = \{B \in \mathcal{O}(z) : w_B \geq 4,\ \Gamma_{w_B} R^\star_B > z\}$,*

$$0 \leq K_{\mathrm{mean}}(z) - K_\star(z) \leq \sum_{B \in \mathcal{Q}(z)} (w_B/2 - 1). \tag{46}$$

*Proof.* On an oracle leaf of width at least four, the mean bound gives $R_B(\bar{a}_B) \leq \Gamma_{w_B} R^\star_B \leq \Gamma_W z$, since $\Gamma_w$ increases over dyadic widths. Width two has zero mean span, and singletons have zero span. The enlarged-budget mean traversal stops at each oracle leaf or an ancestor, proving coarsening. For sharpness take $T = W - 1$, the construction in Equation (42), and $z = 1$. Its oracle retains the root, but any multiplier below $\Gamma_W$ rejects its mean and produces at least two leaves.

At unchanged budget, oracle rejection implies mean rejection, because a passing mean is a positive feasible anchor. Mean therefore reaches every oracle leaf and can only refine it; equality is exactly acceptance at all those leaves. The stated margin suffices by the span bound. Every uncertified oracle leaf generates at most $w_B/2$ mean leaves, since width two always passes, giving the sum bound. At $z = 0$, an oracle-feasible block has zero optimal and mean span and both retain it. □

Membership in $\mathcal{Q}$ is not a rejection. Budget augmentation enlarges the coordinate-condition allowance exponentially as stated in the main text, without promising unchanged accuracy or latency. If a rounded anchor obeys $|\widehat{a}_B - \bar{a}_B| \le \varepsilon_B$, the range triangle inequality supplies instead

$$R_B(\widehat{a}_B) \le R_B(\bar{a}_B) + (w_B - 1)\varepsilon_B \le \Gamma_{w_B} R_B^\star + (w_B - 1)\varepsilon_B. \tag{47}$$

The extra margin must be allowed or the actual anchor checked.

### A.7 Exact retain, repair, and subdivide decisions

For $T \ge 1$, compute the positive mean and $S, M$ from Equation (14). If $S \le z$, the mean retains the block. If $M > z$, every constant anchor fails. Otherwise $M \le z < S$ and the exact answer is

$$I_z = (0, \infty) \cap [L_z, U_z]. \tag{48}$$

A member supplies a retaining anchor, while an empty set certifies rejection. Since increments are positive and $z \ge 0$, every upper endpoint is positive; unrestricted infeasibility is exactly $L_z > U_z$, certified by the conflicting lower and upper prefix-pair constraints. Width one is retained directly; width two has $S = M = 0$ even at $z = 0$. Equality $M = z$ is not rejection and equality $S = z$ is acceptance.

The mean and interval proofs show that each branch gives the exact positive-anchor verdict. Induction on the fixed tree then yields precisely $\mathcal{O}(z)$ and its minimum leaf count. This is a consequence of the existing characterizations, not a separately measured faster planner. Mean screens use linear work; direct interval formation is quadratic, but exact hull-based solvers also admit linear-width implementations. No solver-speed or asymptotic advantage is asserted for screen-and-resolve.

An independent finite reference can evaluate all positive slopes $(c_j - c_i)/(j - i)$. Between consecutive slopes the residual order is fixed and $R(a)$ is affine; below the smallest it decreases, and as $a \to \infty$ it grows. Thus an optimum occurs at one of those slopes. Exact rational agreement checks an implementation, rather than replacing the proof.

Testing a deployed anchor first separates already-retained roots from repairable roots and all-anchor failures. For $(3, 1, 3)$, $S = 4/3$ and $M = 2/3$: mean passes at cap $4/3$, anchor two repairs at cap one, and every constant anchor fails at cap $1/2$. With independent channel anchors, all rate-adjusted channel budgets must pass. On an allowed representable grid $\mathcal{F}$, acceptance requires $[L_z, U_z] \cap \mathcal{F} \neq \varnothing$; a real mean or feasible real interval alone is insufficient. Actual rounded means are checked in GPU execution. The oracle-leaf margins above also localize possible mean-only refinement; they do not predict its execution time.

### A.8 Centered factor range and an all-anchor FP32 obstruction

Let $A_n = -\lambda_n$, $\lambda_n \ge 0$, $\omega = \max_n \lambda_n$, and $r_j = c_j - ja$. A constant gauge gives

$$P_{n,j}(g_n) = \exp[-\lambda_n r_j + g_n], \qquad P_{n,j}^{-1} = \exp[\lambda_n r_j - g_n]. \tag{49}$$

*Proof of Proposition 3.* Set $\beta = \min\{\log F, -\log m\}$ and $x_j = -\lambda_n r_j$. Simultaneous $m \le P, P^{-1} \le F$ is equivalent to $-\beta \le x_j + g_n \le \beta$. The complete feasible gauge interval is

$$g_n \in [-\beta - \min_j x_j,\ \beta - \max_j x_j], \tag{50}$$

which is nonempty iff $\lambda_n R(a) \le 2\beta$. Requiring this for every state gives $\omega R(a) \le 2\beta$. Its midpoint $g_n = \lambda_n(\min r + \max r)/2$ uses the same clock center for every state and has margin $\beta - \lambda_n R(a)/2$. This includes zero rates and equality at the boundary. □

Consequently, for $\omega > 0$, existence in an anchor domain $\mathcal{D}$ is exactly

$$[L_{2\beta/\omega}, U_{2\beta/\omega}] \cap \mathcal{D} \neq \varnothing. \tag{51}$$

At $T = 0$ or $\omega = 0$, unit factors meet the contract directly. At $T = 1$, $a = d_1$ gives unit factors. Unequal factor ranges reduce to the common log interval $[\ell, u]$, where $\ell = \max\{\log m_P, -\log F_I\}$ and $u = \min\{\log F_P, -\log m_I\}$; infeasibility occurs if $\ell > u$, otherwise the same argument replaces $2\beta$ by $u - \ell$.

For any constant center, the exact ratio identity is

$$\log_{10} \frac{\max_j P_{n,j}}{\min_j P_{n,j}} = \frac{\lambda_n R(a)}{\log 10}. \tag{52}$$

For even $T$ and $\mu > \varepsilon > 0$, take $T$ full transitions alternating $\mu+\varepsilon, \mu-\varepsilon$. Raw prefix normalization has clock range $T\mu$, while $a = \mu$ has range $\varepsilon$. Two homogeneous runs of the same increments give

$$r_t = \begin{cases} t\varepsilon, & 0 \le t \le T/2, \\ (T-t)\varepsilon, & T/2 \le t \le T, \end{cases} \qquad R(\mu) = T\varepsilon/2. \tag{53}$$

Here $T$ counts full transitions rather than the $W-1$ transitions after block-source absorption. Centering changes neither range nor the ratio; consistent initial coordinates preserve nonzero physical state as shown above.

**Proof of the division-free obstruction.** For $T \ge 1$ and $D_j = Tc_j - jc_T$, Equation (40) holds for any anchor in place of $a^\star$, hence

$$\max_j |D_j|/T \le R(a) \quad \text{for every real } a. \tag{54}$$

If $\omega \max_j |D_j| > 178T$, a state attaining $\omega$ has log-factor range greater than 178 under every constant anchor and gauge. Requiring both factors to be at most the largest finite FP32 value $F_{\max}$ restricts all log factors to $[-\log F_{\max}, \log F_{\max}]$. Its width is strictly below 178 because

$$\log 2 = 2\sum_{k=0}^{\infty} \frac{1}{(2k+1)3^{2k+1}} < \frac{25}{36}, \qquad 2\log F_{\max} < 256 \log 2 < \frac{1600}{9} < 178. \tag{55}$$

For the first bound keep $2/3$ and bound remaining denominators by three, then sum the geometric tail. Thus Equation (24) proves impossibility even allowing subnormal factors: the reciprocal upper bound already supplies a necessary lower bound. No witness is reported at $T = 0$; at $T = 1$ all $D_j = 0$. Zero rate or equality at the strict threshold does not reject.

For dyadic data write $c_j = p_j 2^e$ and $\omega = m_\omega 2^{e_\omega}$. The exact predicate becomes

$$m_\omega \max_j |Tp_j - jp_T|\, 2^{e+e_\omega} > 178T. \tag{56}$$

Moving negative powers of two to the right gives an integer comparison; sufficient integer capacity or arbitrary precision avoids floating threshold ambiguity. The maximizing prefix and represented inputs are checkable evidence. Non-triggering roots remain unclassified.

**What finite range does and does not certify.** These statements bound ideal exponential magnitudes, not exact representability of transcendental values. FP32 normal-only factors allow $2 \cdot 126 \log 2 \approx 174.673$; allowing subnormals gives $2 \log F_{\max} \approx 177.446$. The conservative witness 178 is distinct from planning cap 32. Exponent evaluation, reciprocal formation, and subnormal handling still require implementation checks. With independently established exponent and factor-log error bounds $\eta, \varepsilon$, the sufficient margin is $\lambda_n R(a)/2 + \eta + \varepsilon \le \beta$; these bounds are not assumed for measured kernels. Finite factors do not bound transformed sources, intermediate states, readout cancellation, or complete output error. Conversely, this coordinate obstruction does not make physical execution impossible: zero forcing and zero initial state give zero physical outputs. Subdivision or a different precision, scaling, or execution contract can remain feasible.

### A.9 Residual roundoff and coordinate-error scope

Assume exact stored inputs $d_j, a$, unit roundoff $u$, the usual relative-error model for subtraction and addition, no overflow, and $R(a) \le z$. Put $x_j = d_j - a$, $r_j = \sum_{i=1}^{j} x_i$, $\widehat{x}_j = \mathrm{fl}(d_j - a)$, $\widehat{r}_j = \mathrm{fl}(\widehat{r}_{j-1} + \widehat{x}_j)$, and $E_j = |\widehat{r}_j - r_j|$, starting from zero. Since $x_j$ is a difference of two residual prefixes and $r_0 = 0$, $|x_j| \le z$ and $|r_j| \le z$. Thus $\widehat{x}_j = x_j + \delta_j$ with $|\delta_j| \le uz$. The exact adder input is $r_j + e_{j-1} + \delta_j$; its rounding error is bounded by $u(z + E_{j-1} + uz)$. Therefore

$$E_j \le (1+u)E_{j-1} + (2+u)uz, \tag{57}$$

$$E_j \le (2+u)uz \sum_{m=0}^{j-1} (1+u)^m = (2+u)z[(1+u)^j - 1] \tag{58}$$

$$= 2juz + O(j^2u^2z) \quad (ju \ll 1). \tag{59}$$

At $u = 0$ the bound is zero directly. Perturbing stored inputs from their targets adds the separate term $\sum_{i=1}^{j} |\delta d_i| + j|\delta a|$. The relative-error premise does not silently cover all subnormal behavior or certify complete output accuracy.

For prefix perturbations $\|\widetilde{c} - c\|_\infty \le \epsilon$, each fixed-anchor range changes by at most $2\epsilon$. Taking infima over the same domain proves

$$|R^\star(\widetilde{c}) - R^\star(c)| \le 2\epsilon. \tag{60}$$

Each interval endpoint also changes by at most $2\epsilon$, since individual pair constraints change by at most $2\epsilon/(j-i)$. Strict margins can therefore protect represented-data certificates.

For one state with $\kappa = e^{A_n a}$, write the zero-state source response as $h = \mathcal{D}T_\kappa \mathcal{D}^{-1} v$ and $\widehat{h} = \mathcal{D}(T_\kappa + E)\mathcal{D}^{-1} v$, with exact $\mathcal{D} = \operatorname{diag}_j e^{A_n(c_j - ja)}$. Then

$$\|\widehat{h} - h\|_2 \le \|\mathcal{D}\|_2 \|\mathcal{D}^{-1}\|_2 \|E\|_2 \|v\|_2 = e^{|A_n| R(a)} \|E\|_2 \|v\|_2. \tag{61}$$

Actual exponentials, source construction, scales, and readout also round, and $E$ may depend on the anchor. For exact readout coefficients and componentwise relative state error $\epsilon$, the relative readout bound is $\epsilon \sum_n |C_n h_n| / |\sum_n C_n h_n|$ when the denominator is nonzero. Cancellation can make it large; the experiments retain absolute as well as relative screens. A physical residual $r_t^{\mathrm{phys}} = \widehat{h}_t - F_t \widehat{h}_{t-1} - v_t$ satisfies $e_t = F_t e_{t-1} + r_t^{\mathrm{phys}}$. Although $\|F_t\|_2 \le 1$, residual evaluation also rounds; no universal certified output selector is inferred.

### A.10 Discrete anchor sharing and bounded candidate admission

Fix one channel, its spectrum and precision, a partition, and a PAIR bank with $N$ entries per anchor key regardless of non-singleton width. For a finite ordered positive-anchor set $\mathcal{F}$ and exact leaf interval $I_B = [L_B, U_B]$, let

$$\ell_B = \min(\mathcal{F} \cap I_B), \qquad u_B = \max(\mathcal{F} \cap I_B). \tag{62}$$

An empty intersection proves infeasibility for that fixed partition/domain. Directed endpoint rounding on the FP32 grid must verify membership; nearest rounding is insufficient. Sparse dictionaries require their own predecessor/successor queries. Singletons need no internal bank; at zero rate every allowed anchor passes a nonnegative budget.

**Proposition 7** (Minimum anchor-key cover)**.** *If all intersections are nonempty, sort the leaves by $u_B$ and select the rightmost allowed point of each first uncovered interval. This gives a minimum-cardinality anchor-key cover. Its triggering intervals certify the optimum; replacing each key by any allowed point in its assigned group's common feasible intersection preserves it.*

*Proof.* The selected point lies in the current interval. If the last point $p$ misses the next interval, sorted upper endpoints imply $p \le u_B$, hence $p < \ell_B$; all earlier points miss it too. The grid hull $[\ell_B, u_B]$ of each triggering interval ends at its selected point, and the next trigger starts strictly above it. These grid hulls, hence their allowed-point sets, are pairwise disjoint, so every cover requires at least one allowed point per trigger. Greedy uses exactly that number. For an assigned group $\mathcal{G}_k$, its allowed intersection

$$J_k = \mathcal{F} \cap \bigcap_{B \in \mathcal{G}_k} I_B \tag{63}$$

contains the selected point. Any replacement from $J_k$ preserves assignments. Each group contains a triggering allowed-point set disjoint from the others, so replacement keys remain distinct and attain the same minimum. □

This is classical interval stabbing applied to an execution-defined grid. Direct all-pairs interval formation and sorting cost $O(\sum_B w_B^2 + q \log q)$ for $q$ bank-requiring leaves, plus domain queries and exact-arithmetic bit costs. The theorem minimizes key count, not solver work, total VRAM, or latency. Different keys can generate identical rounded vectors; measured implementations deduplicate complete words. A SOURCE bank with width-dependent powers needs separate same-width groups or an explicit common-capacity layout.

**The fixed four-candidate policy.** Keep the mean partition and its stored positive finite FP32 anchor $a_{\mathrm{m}}$. For normal $a_{\mathrm{m}}$ choose $e$ with $2^e \le a_{\mathrm{m}} < 2^{e+1}$; for subnormals use $e = -126$. In order $q = 0, 2, 4, 8$, form

$$a_q = 2^{e-q}\,\mathrm{nint}_{\mathrm{even}}(a_{\mathrm{m}}/2^{e-q}). \tag{64}$$

Skip zero or nonfinite results and select the first conservatively admitted candidate. Exponent carry is allowed. At $q = 0$, normal midpoint ties round upward (three maps to four), while half the smallest normal rounds to zero. If none passes, keep the original mean bits under their original planner predicate. This fallback obtains no new exact certificate. Singletons retain their metadata without a scan. Rebind both source and recovery consistently; the same selected anchor can be materialized or evaluated on demand.

Interpret all FP32 inputs as exact dyadic values and let $\mathrm{RD}_{64}, \mathrm{RU}_{64}$ denote directed FP64 rounding. Starting at zero, form

$$\begin{aligned} \underline{\delta}_j &= \mathrm{RD}_{64}(d_j - a), & \overline{\delta}_j &= \mathrm{RU}_{64}(d_j - a), \\ \underline{r}_j &= \mathrm{RD}_{64}(\underline{r}_{j-1} + \underline{\delta}_j), & \overline{r}_j &= \mathrm{RU}_{64}(\overline{r}_{j-1} + \overline{\delta}_j). \end{aligned} \tag{65}$$

**Proposition 8** (Conservative candidate admission)**.** *Let the increments, candidate $a > 0$, and rate $\omega \ge 0$ be finite FP32 values. Including prefix zero, define*

$$\widehat{S}(a) = \mathrm{RU}_{64}\left(\omega\ \mathrm{RU}_{64}[\max_j \overline{r}_j - \min_j \underline{r}_j]\right). \tag{66}$$

*If these operations are finite, $\widehat{S}(a) \le s_{\max}$ implies $\omega R(a) \le s_{\max}$ for the exact represented inputs.*

*Proof.* Directed subtraction encloses each exact increment residual. Given prefix enclosure at $j-1$, monotonicity of addition and directed rounding prove $\underline{r}_j \le c_j - ja \le \overline{r}_j$ by induction. Thus $R(a) \le \max \overline{r} - \min \underline{r}$. The bound is nonnegative because zero is included. Upward subtraction and multiplication by $\omega \ge 0$ preserve the inequality, proving admission. □

A single final rounding adjustment cannot replace directed enclosure at every operation. Widely separated FP32 values need not have an exactly representable FP64 difference. In the measured width-at-most-16 setting all these bounds fit FP64, including extreme FP32 inputs; outward rounding can conservatively reject a feasible boundary candidate. Selection costs one $N$-state rate reduction and at most four scans, $O(N+4(w-1))$ work, without all-pairs interval construction. This heuristic has no minimum-key or approximation guarantee. Preparation includes planning, rebinding, packing and, where used, bank generation/deduplication. Candidate span admission and complete output fidelity remain separate checks.

## B Computational consequences and complete cost comparisons

This appendix supplies the computational proofs used in the main text. All counts are exact-real arithmetic on the stated schedules: addition, subtraction and multiplication cost one ordinary operation, and an FMA costs two. Coefficient preparation is charged where specified. Neither a local count nor a logical array payload is a claim about native latency or peak GPU memory.

### B.1 Consumed work on the measured trees

Fix a source-update DAG, its segment lengths and requested source outputs. Prune transition products with no dependency path to a consumed coefficient or requested transition output, retaining the source schedule. If $F$ source updates $b_R \leftarrow f_R b_L + b_R$ and $H$ transition products $f \leftarrow f_R f_L$ remain, their affine cost is $2F + H$ per state component. At common $A, a$, a length-$m$ anchored segment has transfer $\Lambda^m$. Substituting its prepared length-indexed coefficient preserves every source update by induction: the leaf sources, coefficients and predecessor values agree. This proves Proposition 5:

$$C_P = 2F + H, \qquad C_S = 2F. \tag{67}$$

Length arithmetic, lookup and coefficient preparation are outside this prepared-coefficient ledger; generating powers or changing the source schedule changes the comparison.

For the measured inclusive readout tree at $w = 2^k > 1$, there are $w - 1$ upsweep, $w - 1$ downsweep and $w$ inclusive-source updates. Each nonroot internal subtree transition feeds a source update: right-subtree transitions feed ancestor updates and left-subtree transitions feed the source-only downsweep. The root transition is unused, and no transition-prefix output is requested. Thus $F_{\mathrm{read}} = 3w - 2$, $H_{\mathrm{read}} = w - 2$, and the costs are $7w - 6$ versus $6w - 4$. This schedule includes identity-prefix source operations. The historical $8w - 6$ full-pair count additionally charges transition-prefix work that the measured comparator does not perform.

The terminal-summary tree needs only its $w-1$ upsweep source updates. At each of the $k$ levels, the first subtree is a whole prefix whose transition only feeds further prefix-transition products; those $k$ products are dead. Every other internal transition reaches a right-subtree coefficient consumed by an ancestor source update. Hence $F_{\mathrm{sum}} = w-1$, $H_{\mathrm{sum}} = w-1-k$, and the costs are $3(w-1)-k$ versus $2(w-1)$. This proves Corollary 3. Width one is dispatched directly with zero local-tree work, and both trees tie at width two.

The complete executor also constructs physical endpoint transfers, carries the incoming physical state between leaves, restores coordinates and contracts with $C_t$. These operations and each route's preparation are included in fresh GPU timing, but not in the local-tree counts above.

### B.2 Partition guarantees with prepared and on-demand powers

Fix a complete dyadic tree on $L = 2^q \geq 4$ tokens, positive internal increments, independent positive constant anchors and span cap $z$ at every node. For a cut $\mathcal{P}$, write $K = |\mathcal{P}|$, $K_1$ for its singleton count and $J = \sum_{B\in\mathcal{P}} \log_2 w_B$. Summing the measured summary and readout trees gives, per state,

$$C_S = 8L - 6K - 2K_1, \tag{68}$$

$$H = 2L - 3K - J + K_1, \tag{69}$$

$$C_P = 10L - 9K - J - K_1. \tag{70}$$

The singleton terms enforce zero local work there. Exact mean, exact-mean rescue/fallback and unrestricted-anchor oracle cuts have $K_1 = 0$, because width two is feasible at its own mean. $H$ is work removed from the matched affine tree, not work saved against native execution.

**Corollary 5** (Feasible specialization capacity)**.** *A merge of two width-$w/2$ leaves into $w = 2^k \geq 4$ changes the ledger by*

$$\Delta K = -1, \quad \Delta C_S = 6, \quad \Delta H = k + 1, \quad \Delta C_P = k + 7. \tag{71}$$

*Let $\mathcal{O}(z)$ be the stop-at-first-feasible unrestricted-anchor cut, $\mathcal{M}(z)$ the exact-mean cut and $\Gamma_L = 2 - 2/(L-1)$. Then*

$$H(\mathcal{O}(z)) = \max_{\mathcal{P} \text{ feasible at } z} H(\mathcal{P}), \tag{72}$$

$$H(\mathcal{M}(z)) \leq H(\mathcal{O}(z)) \leq H(\mathcal{M}(\Gamma_L z)). \tag{73}$$

*The same-budget construction of Lemma 1 gives*

$$H(\mathcal{M}(z)) = 0, \qquad H(\mathcal{O}(z)) = 2L - 3 - \log_2 L > 0. \tag{74}$$

*Proof.* Subtracting the two children from their parent proves the merge identities. Every feasible cut refines $\mathcal{O}(z)$: traversal cannot reject an ancestor yet pass through a feasible leaf of another cut without stopping. Each split at width at least four reduces $H$ by $k + 1$; splitting width two leaves $H = 0$. Thus the oracle maximizes $H$, including over cuts with singletons. The mean cut is feasible and refines the oracle cut, while Corollary 4 shows that $\mathcal{M}(\Gamma_L z)$ coarsens it. Monotonicity proves the bounds. In the same-budget fixture, mean has only width-two leaves and oracle keeps the root, giving the displayed separation. □

The multiplier is sharp: an oracle root realizing the sharp mean-span bound is rejected by mean at any smaller uniform multiplier, and any proper refinement of a root of width at least four strictly reduces $H$. Budget augmentation changes the coordinate allowance from $C$ to $C^{\Gamma_L}$; it supplies no unchanged output-fidelity guarantee. Rounded anchors must pass their actual admission tests.

**Power generation.** The on-demand summary and readout kernels separately form their dyadic ladders. Each width-$2^k \geq 2$ leaf therefore adds $2(k-1)$ SOURCE squarings; a singleton adds zero. For $N$ states, the local removed work and generation charge are

$$H_w = N(2w - 3 - k), \quad Q_w = 2N(k-1), \quad H_w - Q_w = N(2w - 1 - 3\log_2 w). \tag{75}$$

The net is zero at width two and positive for every dyadic width at least four (the increment from $w$ to $2w$ is $2w-3$). At widths $4, 16, 256$ it is $1, 19, 487$ per state. For a full cut, in per-state units,

$$Q = 2J - 2K + 2K_1, \qquad G = H - Q = 2L - K - 3J - K_1, \tag{76}$$

$$C_S + Q = 8L - 8K + 2J. \tag{77}$$

**Corollary 6** (Generation-inclusive specialization capacity)**.** *A merge into width $2^k \geq 4$ gives*

$$\Delta G = 3k - 5 > 0, \qquad \Delta(C_S + Q) = 12 - 2k. \tag{78}$$

*A singleton-pair merge gives $\Delta G = 0$ and $\Delta(C_S + Q) = 10$. Consequently,*

$$G(\mathcal{O}(z)) = \max_{\mathcal{P} \text{ feasible at } z} G(\mathcal{P}), \tag{79}$$

$$G(\mathcal{M}(z)) \leq G(\mathcal{O}(z)) \leq G(\mathcal{M}(\Gamma_L z)). \tag{80}$$

*The multiplier remains sharp; the same-budget fixture gives*

$$G(\mathcal{M}(z)) = 0, \qquad G(\mathcal{O}(z)) = 2L - 1 - 3\log_2 L > 0. \tag{81}$$

*Proof.* For non-singleton children, $\Delta J = 2 - k$. Substitution proves the merge identities; width two is checked directly. The preceding refinement proof then applies to $G$, which is nondecreasing under every merge and strictly increases above width two. The sharp root and same-budget constructions therefore give the stated sharpness and separation. □

**When is coarsening cheaper?** For prepared powers and cuts without singletons, an additive model with $\alpha \geq 0$ per local ordinary operation, $\beta \geq 0$ per leaf and all other costs fixed is

$$T_S = \alpha C_S + \beta K + T_0 = 8\alpha L + (\beta - 6\alpha)K + T_0. \tag{82}$$

Oracle minimizes this model when $\beta \geq 6\alpha$; uniform width two minimizes it when $\beta \leq 6\alpha$, with all cuts tied at equality. For the matched affine ledger, a width-$2^k$ merge strictly reduces cost precisely when $\beta > \alpha(k+7)$. Charging both SOURCE ladders instead gives

$$T_{\mathrm{OD}} = \alpha(C_S + Q) + \beta K + T_0, \qquad \Delta T_{\mathrm{OD}} = \alpha(12 - 2k) - \beta \quad (k \geq 2). \tag{83}$$

Thus a merge is strictly beneficial precisely when $\beta > \alpha(12 - 2k)$. At widths $4, 16, 64, 256$, its local arithmetic change is $8, 4, 0, -4$. The sufficient uniform oracle-optimality condition is $\beta \geq 8\alpha$ for minimum width two, or $\beta \geq 10\alpha$ when singletons are allowed. These conditions need not be necessary on a given tree; maximizing $G$ alone need not minimize time. For arbitrary known additive node costs, feasible cuts can instead be optimized by

$$V(B) = \min\{c(B) \text{ if } I_z(B) \neq \varnothing,\ V(B_L) + V(B_R)\}. \tag{84}$$

Permitted terminal widths provide base cases. This is standard tree dynamic programming, not a new or benchmarked optimizer. Shared anchors or nonadditive costs need a different state/problem. Neither additive model has been validated as a GPU-latency predictor: preparation, padding, occupancy, scheduling and physical carry may violate its assumptions.

### B.3 A DIFFERENT SCAN AND ITS STORAGE LEDGER

The following Brent–Kung DAG (Brent & Kung, 1982) is distinct from the measured two-tree schedule. Let $W = 2^k \geq 2$, positions $0, \ldots, W-1$, and absorb the incoming state into $b_0 = F_s h_{s-1} + v_s$. Request only the source component of each inclusive prefix. For distances $d = 1, 2, \ldots, W/2$ in powers of two, the upsweep merges $r - d$ into $r$ at $r = 2d-1, 4d-1, \ldots, W-1$. The downsweep uses $d = W/4, \ldots, 1$ and $r = 3d-1, 5d-1, \ldots, W-d-1$. Prune transition products not consumed by requested source outputs.

**Corollary 7** (Specified Brent–Kung scan)**.** *The live source FMA and transition-product counts are*

$$F_W = 2W - k - 2, \qquad T_W = W - 1 - k, \tag{85}$$

*so its affine and prepared source cores cost, per state,*

$$C_{\text{affine}} = 5W - 3k - 5, \qquad C_{\text{source}} = 4W - 2k - 4. \tag{86}$$

*Source FMAs request $F_W$ physical segment keys, versus $k$ anchored length keys $\{2^r : 0 \le r < k\}$ at fixed $A, a$.*

*Proof.* Upsweep has $W - 1$ merges; downsweep has $W/(2d) - 1$ at distance $d$, summing to $W - k - 1$. Every source result reaches an output, giving $F_W$. Among upsweep transitions, exactly the $k$ prefixes $[0, 2^r)$ are dead: their transition values only feed further prefix-transition products. Each other upsweep transition lies in the right half of an aligned ancestor, reaching a coefficient used in that ancestor's source FMA. Every downsweep transition result is a prefix and is dead. Hence $T_W = W - 1 - k$. At distance $d = 2^r$, upsweep requests right segments starting at $d, 3d, \ldots$ and downsweep those at $2d, 4d, \ldots$. Together they request all aligned length-$d$ segments except the first, or $W/d - 1$ keys. Summing gives $F_W$. Physical coefficients are $e^{A\sum \Delta_i}$; anchored coefficients $e^{Aad}$ depend only on the $k$ lengths. □

The prepared core reduction $T_W/(2F_W + T_W)$ tends to $1/5$ and is zero at $W = 2$. Distinct physical keys need not have distinct numerical values. Coordinate transforms, coefficient generation and physical carry remain outside these core counts.

**Storage including coordinates.** With $N$ states and $b$ bytes per scalar, an affine summary has numeric payload $2bN$; source-only has $bN$ plus structural length metadata. Materializing every coefficient key in this DAG gives

$$B_{\text{physical}} = bNF_W, \qquad B_{\text{bank}} = bNk. \tag{87}$$

The bank is reusable only at shared $A, a$. A physical algorithm can generate or discard coefficients instead, so $B_{\text{physical}}$ is not its minimum memory requirement. For the exact mean of the $W - 1$ internal increments, $P_0 = P_{W-1} = I$; omitting these endpoints leaves $N(W - 2)$ coordinate scalars. Thus

$$\begin{aligned} B_{\text{bank+pair}} &= bN\{k + 2(W - 2)\}, & B_{\text{bank+pair}} - B_{\text{physical}} &= bN(2k - 2), \\ B_{\text{bank+forward}} &= bN\{k + W - 2\}, & B_{\text{bank+forward}} - B_{\text{physical}} &= bN(-W + 2k). \end{aligned} \tag{88}$$

Forward-only storage requires inverse regeneration. Separately storing transformed sources and readouts adds $2bNW$ bytes. Rounded means may not make the endpoint exactly one. Allocator overhead and schedule-dependent liveness are excluded; the smaller bank alone does not prove smaller total or peak memory.

### B.4 Fresh arithmetic under a common primitive model

Now fix one channel with given $A, \Delta_t, B_t, C_t, u_t$, requesting every $y_t = C_t^\top h_t$ and the final physical state. EXP, reciprocal RCP, division DIV and channel-shared clock operations are recorded separately. EXP denotes the exact-real primitive $x \mapsto e^x$. The native code's $\exp_2(x \log_2 e)$ is normalized to this primitive; base-conversion scaling and its implementation-specific repetition are outside this common model. The common projections/softplus and external skip/gate are excluded from all routes. Source formation and contraction cost exactly $L + NL + NL + L(N - 1) = 3NL$ ordinary operations; there is no dense $N^2$ charge. Metadata, traffic, synchronization and finite-precision fidelity are separate contracts.

**A complete boundary-algorithm comparison.** Set $h_0 = 0$, $L = 2048q$, group width $B = 16$ and $J = L/B = 2^j$. Each group first computes a zero-state source summary; a macro scan gives boundary states, then a local second pass returns all token states. Both boundary candidates use the same pruned Brent–Kung macro graph. Physical-clock generates macro transfers $e^{A\sigma_i}$ from group clocks $\sigma_i$. Boundary mean instead uses

$$\mu = \frac{\sum_{i=2}^{J} \sigma_i}{J - 1}, \qquad r_i = \sum_{s=2}^{i} \sigma_s - (i - 1)\mu, \qquad r_1 = r_J = 0.$$

It forms $e^{Ar_i}$ and reciprocals at the $J-2$ nontrivial positions, transforms/restores macro sources and builds the $e^{A\mu}$ ladder. This fresh comparison has a fixed layout, no span planner and no assumption that one global anchor passes a production range budget. It is not the H100 policy executor.

The macro counts are $F_J = 2J - j - 2$, $T_J = J - j - 1$. Both routes pay the common per-state ordinary count $U = 3L + (L-1) + 2[(L-J) + (L-1)]$ for source/readout contraction, elementary exponential arguments and local summary/readout passes. Their full counts are

$$\begin{aligned} C_{\mathrm{pc}} &= U + 2F_J + T_J + (J-1) = 8L + 4J - 3j - 9, \\ C_{\mathrm{bm}} &= U + 2F_J + 2(J-2) + (J-1) + (j-1) = 8L + 5J - j - 13. \end{aligned} \tag{89}$$

Mean's last three added terms are coordinate multiplications, macro exponential arguments and ladder squarings. Reusing local endpoints removes the same $J$ FMAs from each candidate and leaves the difference unchanged.

For comparison, the pinned Mamba source model with a conditional CUB 2.0.1 schedule uses 128 threads $\times 16$ items and full 2048-token chunks. Backward slicing from requested outputs, with zero/one identities, leaves $4804q - 129$ source FMAs and $2628q - 238$ multiplications: the first zero-state chunk contributes $4675, 2390$ and each further chunk $4804, 2628$. Including the same front end gives

$$C_{\mathrm{ns}} = 3L + (L-1) + 2(4804q - 129) + (2628q - 238) = 20428q - 497. \tag{90}$$

This is a conditional source-DAG model, not a compiled-binary instruction count. The native and boundary algorithms use different scan schedules.

| $L$ | Native $C_{\mathrm{ns}}$ | Mean $C_{\mathrm{bm}}$ | Physical $C_{\mathrm{pc}}$ | Mean − physical |
|---|---|---|---|---|
| 2048 | 19,931 | 17,004 | 16,866 | +138 |
| 4096 | 40,359 | 34,027 | 33,759 | +268 |
| 8192 | 81,215 | 68,074 | 67,548 | +526 |
| 16384 | 162,927 | 136,169 | 135,129 | +1040 |

**Table 3:** Fresh ordinary operations per state for the declared zero-state algorithms. EXP/RCP and shared clock costs are additional, as given below.

Native uses $L - 1$ EXPs per state; both boundary routes use $L + J - 2$. Mean additionally uses $J - 2$ RCPs per state and one channel-level DIV. Physical clock formation costs $15(J-1)$ channel operations; the balanced ragged clock prefix plus mean residuals costs $19J - 24 - j$. Thus

$$C_{\mathrm{bm}} - C_{\mathrm{pc}} = J + 2j - 4 > 0 \tag{91}$$

in this range, with equal EXP counts and additional mean RCP/clock/DIV costs. This fresh mean algorithm is therefore more expensive in every nonidentical primitive category than its matched physical-clock candidate. Its saving over the pinned native graph cannot all be attributed to anchoring.

For ordinary-operation unit weight and EXP, RCP, channel-clock and DIV weights $c_E, c_R, c_C, c_D$, mean is cheaper than the native source model at $L = 4096$ exactly when

$$6332N > 255Nc_E + 254Nc_R + 4832c_C + c_D. \tag{92}$$

This is an algebraic condition on declared weights, not a universal choice of them. All three algorithms have linear work; length changes constants and lower-order terms. These full counts and H100 timings concern different algorithms, so they cannot explain one another's numerical ratios.

**Verification and retained detail.** Exact CPU checks enumerate the declared DAGs and feasible cuts, including singletons, sharpness and same-budget fixtures. The supplementary reproducibility material records their finite scopes and outputs under `analysis/four_benefits/`, `analysis/ondemand_v40/` and `analysis/arithmetic_v39/`. The arithmetic audit checks the native DAG for $q = 1, \ldots, 8$, both boundary and ragged-clock graphs at the table's lengths, and direct-recurrence state agreement. Nonzero-state identity fixtures are included, but zero-state count cancellations require recounting for that contract. These checks complement the proofs; they add no GPU observations and do not certify FP32 fidelity or global-anchor admissibility.

# C H100 MATCHED SOURCE SPECIALIZATION AND PARALLEL PHYSICAL CONTROLS

This experiment supplies two comparisons on a common input panel: PAIR versus SOURCE in the on-demand executor, and anchored execution versus native-derived parallel physical precision controls. The first tests how an identified arithmetic saving affects elapsed time within a matched implementation. The second measures complete implementations with the same output contract; it does not isolate arithmetic counts or equalize all optimizations. Tables 1 and 2 report these distinct questions in the main text.

## C.1 FROZEN SELECTION, HARDWARE AND EXECUTION CONTRACT

The documents, in slot order, are 75489, 74688, 78271, 74557, 72784, 76176, 70557 and 66993, the eight reused documents of Appendices D.4 and D.5. The artifact fixes their 4096-token arrays; shorter cases use the first 2048 tokens. The 130M layers are 3, 11 and 19; the 370M layers are 7, 23 and 39. Batch size is one, $N = 16$, and all channels are evaluated (1536 and 2048, respectively). P130 and P370 each have 24 settings at $L = 2048$, root 16/cap 32. A370 uses root 256/cap 8 on the eight 370M layer-39 inputs; L370 uses root 16/cap 32 and $L = 4096$ on that same selected layer and document set. Thus 64 settings comprise 56 unique cores and eight reused documents; A370 repeats eight P370 physical inputs. No new-document generalization claim or pooling across nested lengths is made.

One H100 80GB HBM3 allocation (SM90) runs Python 3.10.20, PyTorch 2.5.1+cu121 and CUDA 12.1. The frozen checkpoints and tokens are recaptured on H100. H100 core identities are sealed before timing and checked on repeats: historical A100 $A$, deployed anchors and $h_0$ match in all 56 cores, but $u, \Delta, B, C$ differ in all 56. Consequently A100/H100 timing ratios would mix hardware and captured-word changes and are not reported. Within this experiment, all routes receive the same H100 input words. There are no training updates. The measured panel has zero incoming state; qualification additionally exercises nonzero states.

The policies $D, R, M, H$ are deployed, width-two rescue, mean and deployed-first mean fallback, with the definitions in Appendix D.5. Each reached node is tested independently using ordered FP64 clock sums; a computed mean is rounded once to FP32 and the span predicate is rechecked at that anchor. All policies use one planner, width-packed leaves and physical boundary carry. Each is crossed with PAIR and SOURCE, yielding eight anchored routes. Both representations cache the same locally generated FP32 $\lambda$ per state/leaf/kernel; SOURCE additionally constructs the required dyadic powers by rounded squaring, inside execution. Neither stores a global propagation bank. The local source arithmetic, node map, synchronization, carry, readout and thread counts are shared. PAIR computes only consumed transition products: unused readout-root and prefix-alpha work is omitted. The remaining six routes are unchanged official native, parallel FP32, COEFF64, ACC64 and ALL64, and sequential ALL64 as a bridge to the earlier ordered comparator. No fastest-passing precision mode is selected after measurement.

For every policy/setting, PAIR and SOURCE have identical plan topology, execution descriptors and anchor bits (256 matched pairs). Their recorded full-output and final-state hashes also agree in all 256 pairs. Official native and parallel FP32 hashes agree in all 64 settings, including those that fail the reference gate. These observed identities do not assert bitwise equality for arbitrary inputs or between real and finite-precision recurrences.

## C.2 PARALLEL PRECISION MODES AND REPRODUCIBLE IMPLEMENTATION

The parallel controls derive from the native CUB affine-scan structure: native trait tilings, vector/scalar I/O, warp scans, state-index loops and chunk carry are retained in a separately compiled bounded specialization. The archived provenance identifies upstream commit `a07faffa36a7b89e754b5de972418475bcdd77b6`, the two input headers and the Apache-2.0 license. The API specializes batch one, real FP32 inputs, dynamic $B, C$, $N \leq 16$, $L \leq 4096$, and no skip, gate, bias or softplus inside this recurrence stage. Official native is retained unchanged as a separate route. The authored wrapper, explicit composition/readout FMAs, shared-region alignment and FP64 resource requirements are recorded departures, so equal optimality to official native is not assumed.

FP32 preserves native coefficient/source words, including $\exp2f(\mathrm{RN}_{32}(\Delta\,\mathrm{RN}_{32}(A\log_2 e)))$, and uses FP32 scan/readout. COEFF64 evaluates coefficient/source expressions in FP64 and rounds their results to FP32 before scan/readout. ACC64 widens scan/readout while preserving native FP32 coefficient/source words. ALL64 evaluates both stages in FP64. All original inputs and public outputs remain FP32. COEFF64/ALL64 also replace the native exponential formulation by a double-precision exponential; their success is not an isolated causal estimate of precision alone. The sequential bridge uses an ordered time loop with FP64 coefficient/source generation, state and readout.

Parallel and leaf-execution modules compile for SM90 with `-O3 -fmad=true`; the ordered planner and sequential bridge use `-fmad=false`, with explicit FMAs retained where declared. No fast-math or autotuning is used in these newly compiled modules. The build receipt preserves compiler flags, source/module hashes, resource attributes and installed CUB header hashes (CUB version 200001). H100 compilation and mandatory GPU qualifications complete before measurement: 128 policy configurations, 28 parallel-control cases and ten sequential-control cases pass. These qualifications include nonzero incoming states, multiple streams, extreme supported widths/lengths and cache rejection checks as appropriate.

### C.3 Timing, reference gates and uncertainty

Fresh time starts from GPU-resident coefficients and includes policy planning, validation, metadata packing, local coefficient/power generation, full readout and final physical state, including required host synchronizations. Prepared time reuses compatible metadata on the same input; local powers and physical coefficients are still generated during execution. Capture, compilation, reference construction, I/O and separate audits are excluded. After two warmups, each route receives 28 synchronized host-time observations in each mode: fourteen cyclic and fourteen reversed rotations of a seeded route permutation (seed 10593011). Every route occupies each position twice; failed routes are skipped without replacement. There are 48,104 valid timing observations, not 48,104 independent inputs.

Both full readout and final physical state must be finite and satisfy $|\widehat{r}_i - r_i| \le 2\cdot 10^{-5} + 2\cdot 10^{-4}|r_i|$ for every element, as well as relative L2 error at most $2\cdot 10^{-4}$ against FP64 recurrence. Initial failures receive no timings. Post-mode gates and route-local bitwise repeatability pass in all 1718 checks; no route is invalidated after timing. A paired comparison uses only documents for which every required layer passes both routes. Paired log ratios are averaged over rounds, then equally over selected layers within document, then equally over complete documents. Absolute times use the analogous route-specific complete-document hierarchy. Pointwise descriptive 95% intervals resample documents 10,000 times (seed 10593031); they are not multiplicity-adjusted and do not measure independent-session uncertainty.

### C.4 Policy and representation effects

**Table 4:** All fresh policy contrasts with pointwise descriptive 95% document-bootstrap intervals. P130/P370 are primary; A370 is adverse; L370 is long. $n$ counts common passing documents (all required layers). Above one favors mean.

| Panel | Rep. | $D/M$ | $R/M$ | $H/M$ | $n$ |
|---|---|---|---|---|---|
| P130 | PAIR | 1.658 [1.631, 1.687] | 1.135 [1.117, 1.153] | 1.0026 [1.0018, 1.0034] | 8 |
| P130 | SOURCE | 1.665 [1.637, 1.693] | 1.138 [1.120, 1.157] | 1.0017 [1.0011, 1.0023] | 8 |
| P370 | PAIR | 1.298 [1.285, 1.315] | 1.091 [1.082, 1.100] | 0.9998 [0.9980, 1.0014] | 8 |
| P370 | SOURCE | 1.301 [1.288, 1.320] | 1.092 [1.084, 1.101] | 1.0006 [0.9990, 1.0021] | 8 |
| A370 | PAIR | 1.005 [0.971, 1.053] | 0.853 [0.828, 0.885] | 1.0494 [1.0464, 1.0521] | 6 |
| A370 | SOURCE | 1.153 [1.137, 1.177] | 0.961 [0.951, 0.972] | 1.0588 [1.0577, 1.0599] | 6 |
| L370 | PAIR | 1.291 [1.276, 1.311] | 1.065 [1.054, 1.077] | 1.0007 [0.9990, 1.0035] | 8 |
| L370 | SOURCE | 1.298 [1.284, 1.318] | 1.063 [1.053, 1.073] | 0.9997 [0.9979, 1.0017] | 8 |

On both primary panels, mean improves on deployed and width-two rescue under both representations. PAIR deployed/mean ratios are $1.658/1.298$ and rescue/mean ratios are $1.135/1.091$ (130M/370M), including preparation. Thus the policy effect extends beyond singleton rescue and does not depend on SOURCE. Mean and fallback are very close: fresh PAIR fallback/mean is $1.0026/0.9998$; no practically large universal mean advantage over fallback is established.

**Table 5:** Matched PAIR/SOURCE ratios for every policy. Above one favors SOURCE. SOURCE powers are generated inside both fresh and prepared execution; prepared reuses only compatible metadata on the same input. Intervals are descriptive and pointwise.

| Panel | Policy | Fresh [95% interval] | Prepared [95% interval] | $n$ |
|---|---|---|---|---|
| P130 | D | 1.0020 [1.0017, 1.0024] | 1.0040 [1.0032, 1.0050] | 8 |
| P130 | R | 1.0033 [1.0024, 1.0039] | 1.0065 [1.0054, 1.0078] | 8 |
| P130 | M | 1.0058 [1.0052, 1.0064] | 1.0121 [1.0109, 1.0132] | 8 |
| P130 | H | 1.0067 [1.0057, 1.0078] | 1.0125 [1.0117, 1.0133] | 8 |
| P370 | D | 1.0096 [1.0081, 1.0107] | 1.0147 [1.0141, 1.0152] | 8 |
| P370 | R | 1.0113 [1.0099, 1.0125] | 1.0235 [1.0213, 1.0252] | 8 |
| P370 | M | 1.0123 [1.0114, 1.0132] | 1.0231 [1.0221, 1.0241] | 8 |
| P370 | H | 1.0114 [1.0097, 1.0131] | 1.0237 [1.0227, 1.0249] | 8 |
| A370 | D | 1.0954 [1.0881, 1.1026] | 1.1618 [1.1512, 1.1718] | 8 |
| A370 | R | 1.1165 [1.1103, 1.1231] | 1.2097 [1.1975, 1.2221] | 8 |
| A370 | M | 1.2568 [1.2141, 1.2857] | 1.4545 [1.3813, 1.5058] | 6 |
| A370 | H | 1.2456 [1.2053, 1.2738] | 1.4551 [1.3810, 1.5048] | 6 |
| L370 | D | 1.0130 [1.0117, 1.0139] | 1.0194 [1.0176, 1.0208] | 8 |
| L370 | R | 1.0213 [1.0188, 1.0236] | 1.0330 [1.0290, 1.0365] | 8 |
| L370 | M | 1.0187 [1.0165, 1.0210] | 1.0286 [1.0261, 1.0318] | 8 |
| L370 | H | 1.0197 [1.0179, 1.0219] | 1.0302 [1.0287, 1.0321] | 8 |

Holding the mean plan fixed, fresh PAIR/SOURCE is $1.0058/1.0123$ in P130/P370 and $1.0187$ in L370; prepared ratios are $1.0121/1.0231/1.0286$. The arithmetic saving survives as a small positive total-time effect on these panels. The leaf-level diagnostic counts both PAIR's consumed transition products and SOURCE power construction; all 64 mean-plan ledgers agree with their actual width histograms. For a dyadic leaf $w \geq 2$, the counted subset is $N(2w - 3 - \log_2 w)$ PAIR products versus $2N(\log_2 w - 1)$ SOURCE squares, a net $N(2w - 1 - 3\log_2 w)$; $w = 1$ contributes zero. These are the transition/power subset of two local passes, not the whole recurrence's FLOPs or a native comparison. Small local savings need not dominate planning, conversion, boundary work or memory traffic.

In A370, the mean SOURCE specialization ratio is larger ($1.2568$), but only six common passing documents remain. Rescue passes all eight; on the six common passing documents rescue/mean is $0.8533$ for PAIR and $0.9613$ for SOURCE, favoring rescue. This jointly changed root/cap configuration identifies an observed policy reversal, not a universal threshold rule or a guarantee obtainable from span alone.

### C.5 Parallel physical comparison, fidelity and memory

**Table 6:** Full-output and final-state gate passes, setting counts. Each named route has the displayed count; a grouped row is not a sum. Counts do not certify equal error patterns. All 37 failures are initial finite readout elementwise exceedances; none is a nonfinite, final-state or relative-L2 failure.

| Route(s) | P130 | P370 | A370 | L370 |
|---|---|---|---|---|
| Native / parallel FP32 | 24/24 | 22/24 | 6/8 | 2/8 |
| Parallel ACC64 | 24/24 | 22/24 | 6/8 | 3/8 |
| Parallel COEFF64 / ALL64 | 24/24 | 24/24 | 8/8 | 8/8 |
| PAIR/SOURCE: deployed / rescue | 24/24 | 24/24 | 8/8 | 8/8 |
| PAIR/SOURCE: mean / fallback | 24/24 | 24/24 | 6/8 | 8/8 |
| Sequential ALL64 | 24/24 | 24/24 | 8/8 | 8/8 |

The run completes all 896 setting–route combinations: 859 pass the reference gate and 37 fail at initial qualification. Every failure is a finite readout elementwise exceedance, with 1–11 failing elements per combination; no final-state, nonfinite or relative-L2 gate fails. Failed readout relative-L2 errors range from $2.94 \cdot 10^{-6}$ to $1.31 \cdot 10^{-5}$. These are strict-gate failures rather than crashes or wholesale divergence; the prespecified threshold is not relaxed. The repeated adverse physical inputs mean 37 invalid combinations represent 31 unique core–route combinations across nine unique cores.

Parallel COEFF64 and ALL64 pass every setting. On P130/P370, SOURCE mean is $21.59/13.98$ times slower than COEFF64 and $17.60/11.33$ times slower than ALL64. In L370 it is $10.86/9.00$

**Table 7:** Median PyTorch tensor increments in MiB, above resident input/capture tensors. Peak includes fresh preparation; live includes outputs. Metadata is retained plan/bank tensor storage. PAIR/SOURCE mean and native/parallel routes respectively have identical reported allocations. Medians include every recorded setting, with adverse fidelity failures retained. These are not whole-device VRAM, shared-memory or external-workspace measurements.

| Panel | Route | Peak | Live | Metadata |
|---|---|---|---|---|
| P130 | Mean PAIR/SOURCE | 360.40 | 72.40 | 59.91 |
| P130 | Native/parallel | 12.19 | 12.19 | 0.00 |
| P370 | Mean PAIR/SOURCE | 405.05 | 61.63 | 45.10 |
| P370 | Native/parallel | 16.25 | 16.25 | 0.00 |
| A370 | Mean PAIR/SOURCE | 417.93 | 54.99 | 38.86 |
| A370 | Native/parallel | 16.25 | 16.25 | 0.00 |
| L370 | Mean PAIR/SOURCE | 763.57 | 120.82 | 88.20 |
| L370 | Native/parallel | 32.50 | 32.50 | 0.00 |

times slower, respectively. These contrasts include all eight documents and are unaffected by the smaller native/FP32/ACC64 valid sets in Table 2. Sequential ALL64 is itself $34.16/25.41/27.25$ times slower than parallel ALL64 in P130/P370/L370. Consequently the anchored advantage over the sequential bridge ($1.94/2.24/3.03$ for mean SOURCE) does not establish an accuracy–cost advantage over the stronger parallel physical controls. The earlier ordered-control results in Appendix D.5 retain their historical comparison scope; they are not pooled with these changed H100 captures and kernels.

Memory also favors the native/parallel implementations in this experiment. Mean peak tensor increments are 360.40/405.05 MiB in P130/P370 versus 12.19/16.25 MiB for native and the parallel modes. Table 7 separates retained metadata, live outputs and preparation peaks. These allocator increments omit resident inputs, inactive capture-model storage, external workspaces and kernel shared memory; they are not full VRAM. The matched PAIR/SOURCE tensor allocations agree even where kernel-local resource use differs. No whole-execution memory reduction is inferred from elimination of transition summaries.

Separate phase profiles cover document slot zero only, with three repeats per route and selected layer. They record planning, preparation and execution costs but are not summed into the primary estimate and do not causally allocate the gap to one optimization. The experiment therefore supports exact conditional arithmetic savings and observed internal policy improvements, while measuring a substantial complete-implementation gap. That gap cannot be attributed solely to insufficient coding optimization, nor is eventual superiority after optimization established.

### C.6 Compact audit and table reproduction

The compact evidence package retains all 64 committed records, protocol, selection, H100 core bindings, source provenance, build/qualification receipts and recorded source code. The original review archive passes CRC; its 535 included files match size and SHA256 receipts. The archived 2.23 GiB of raw arrays/binaries are omitted from the review export with an exclusion inventory. The present audit verifies recorded fidelity metrics and output hashes; it does not claim independent elementwise recalculation from those omitted arrays or a local GPU rerun.

Full leaf geometry, coverage and policy metadata are validated, while independent numerical replanning checks 128 fixed sampled roots plus boundary roots (seed 10593041); exact numerical certification of every production predicate is not claimed. Running `analysis/p3w/build_tables.py` with Python and NumPy verifies the compact evidence manifest, independently recomputes all 216 paired estimates/intervals, 112 absolute estimates/intervals and 168 memory medians from retained records, checks the experimental summary and regenerates the six tables. This separation makes the arithmetic claim, implementation-specific timing and raw-array verification scope explicit.

## D Supporting controls: policy, range and storage

This appendix retains the earlier experiments needed to interpret the main claims. The matched H100 representation and parallel-precision experiment is reported in Appendix C. Supplement Sections

S1–S11 preserve the complete earlier tables, protocols, profiles and audit scopes; historical timings are not pooled with the H100 experiment.

### D.1 Panels and common measurement contract

The first A100 factorial and packed studies use four captured cores: documents 16385 and 32770 at zero-based layers 12 (130M) and 24 (370M). The later P3O–P3T panel has eight documents, 59964, 52636, 62661, 50284, 61729, 50750, 50995 and 51601, and three fixed layers per model: $(3, 11, 19)$ for 130M and $(7, 23, 39)$ for 370M. P3O selected these documents before capture/timing from IDs $[49152, 65536)$, excluding recorded prior document IDs and content-prefix hashes. Of 16,384 candidates, 2,127 satisfy the fixed length rule. Eligible EOS-plus-2,047-token inputs are ranked by SHA256 of `10585017:` followed by the input-token hash, with document-ID tie breaking and distinct 128-token prefixes. P3P–P3T reuse these captures; they are additional controls, not additional input samples.

The frozen-policy P3U panel instead selects eight documents from $[65536, 81920)$ by seeded SHA256 ranking of canonical 4096-token prefixes (seed 10591017), excluding recorded prior IDs and prefix hashes: 75489, 74688, 78271, 74557, 72784, 76176, 70557 and 66993. The layers are unchanged. P3U measures both 2048 and 4096 coefficients from each 4096-token capture, giving 96 new-document cases, plus 12 bridge cases on old documents 59964 and 52636. Nested prefixes are not independent inputs. P3V reuses the authenticated P3U arrays on a separate A100 allocation; the H100 panel recaptures these documents and is reported separately. Neither corpus panel is claimed to be training-disjoint.

All production cores have batch one, $N = 16$, FP32 core arrays and all 1,536/2,048 channels; their captured initial state is zero. Synthetic qualification additionally exercises nonzero physical incoming states. The contract is $(u, \Delta, A, B, C, h_0) \mapsto (y_{1:L}, h_L)$, including continuous physical carry. Every complete readout and final state must be finite and satisfy both

$$|\widehat{x}_i - x_i^{64}| \le 2 \cdot 10^{-5} + 2 \cdot 10^{-4}|x_i^{64}|, \qquad \|\widehat{x} - x^{64}\|_2 / \|x^{64}\|_2 \le 2 \cdot 10^{-4},$$

against a sequential FP64 physical recurrence. Failed routes retain their diagnostics and do not supply qualified timings. A span admission or finite factor check is not a substitute for this output gate.

The A100 studies use 80GB SM80 GPUs, Python 3.10, PyTorch 2.5.1+cu121 and CUDA 12.1. Fresh synchronized host timing starts from GPU-resident core coefficients and includes planning, policy work, packing, required banks or on-demand powers, execution and allocations. Capture/model shell, I/O, compilation, references and independent audits are excluded. Prepared timing uses compatible case-specific metadata. These are core times, not model inference times. For multiple-layer panels, paired log-time ratios are averaged equally over rounds, layers within document and documents, with models separate. Intervals resample document clusters (10,000 draws for P3O/P3P/P3Q/P3V; 2,000 for P3R/P3S/P3T). P3U also reports document-level intervals. They keep the selected layers/device fixed; channels and repeated timings are not independent documents. Retained storage counts unique plan, bank and offset allocations; peak is an isolated allocated-byte increment above resident inputs, not whole-device memory.

### D.2 Representation and policy controls

The four-core factorial at root 256/cap 8 crosses output-sensitive affine (PAIR) and source-only (SOURCE) with deployed ($D$) and mean ($M$) anchors. Within a policy it holds plan, anchor bits, source schedule and physical carry fixed, charging each representation its required preparation. Its affine readout omits unused transitions: the counts are $7w - 6$ versus $6w - 4$, not the historical full-prefix $8w - 6$ comparator. Fresh PAIR/SOURCE under mean is $1.1633$–$1.1719\times$; deployed-to-mean gains are $1.5369$–$1.8229\times$ in PAIR and $1.3482$–$1.5804\times$ in SOURCE. Mean SOURCE remains $56.16$–$69.32\times$ slower than native.

The packed follow-up adapts launch/shared-memory resources to width, processing up to eight leaves per block and eliminating unused affine scratch at widths one/two. All routes are remeasured. Its primary fresh PAIR/SOURCE ratios are $1.0654, 1.0601, 0.9904, 0.9646$ in the four cases; thus the 370M cases reverse the earlier gain. Packed mean SOURCE remains $25.44$–$28.55\times$ slower

than native. The matched profile records SOURCE bank-preparation spans longer by $0.247$–$0.287$ ms, while its summary/readout savings range from $0.013$ to $0.725$ ms. These locate implementation costs without attributing the entire effect to arithmetic, occupancy or any one operation. All 36 primary route outputs and all 60 root/cap sweep settings pass the declared gate. The sweep does not locate an accuracy-failure threshold.

**New inputs and the rescue control.** P3O tests the eight-document, three-layer panel at root 16/cap 32 and root 256/cap 8 with five routes and 20 balanced rounds. P3P reuses those inputs, comparing $D$, width-two rescue ($R$) and $M$ using packed PAIR, 18 balanced rounds and a different allocation. Rescue preserves deployed acceptances and repairs only rejected width-two parents. Its hybrid bank shares deployed entries and adds transitions only for newly rescued leaves. All P3O and P3P route/setting gates pass. Table 8 retains both policy benefit and representation reversals.

**Table 8:** Earlier fresh ratios on 24 document–layer cases/model. Values above one favor the denominator; brackets are 95% document intervals. P3O representation ratios compare the same mean plan. P3P/P3Q use PAIR. Each row uses its own experiment's paired timings.

| Experiment | Contrast | 130M | 370M |
|---|---|---|---|
| P3O, root 16/cap 32 | $D/M$, PAIR | 1.455 [1.434,1.476] | 1.242 [1.231,1.255] |
| | $D/M$, SOURCE | 1.422 [1.401,1.444] | 1.221 [1.209,1.234] |
| | PAIR/SOURCE, $M$ | 0.978 [0.977,0.979] | 0.997 [0.996,0.998] |
| P3P, root 16/cap 32 | $R/M$ | 1.283 [1.266,1.300] | 1.174 [1.165,1.184] |
| P3P, root 256/cap 8 | $R/M$ | 1.102 [1.096,1.108] | 1.063 [1.051,1.075] |
| P3Q, root 16/cap 32 | $M/H$ | 0.9183 [0.9172,0.9194] | 0.9270 [0.9252,0.9291] |

Mean improves P3O's primary PAIR and SOURCE times in all 48 cases, and P3P's primary rescue comparison in all 48. The latter's secondary aggregate conceals a reversal: in all eight 370M layer-39 cases, mean is slower than rescue (layer-level $R/M = 0.840123$). Root and cap change jointly; no universal threshold is inferred. P3O shares the earlier allocation; P3P's two sessions complete disjoint portions of one new allocation. Neither supplies multiple independent full-panel replications.

**Structural rescue census.** The exact census is retrospective analysis of two sealed eight-document sets: old IDs $(16385, 17414, 18441, 19465, 20483, 21508, 22528, 23556)$ and new IDs $(32770, 33801, 34816, 35869, 36869, 37890, 38943, 39948)$. Old/new layers are $(0, 7, 15, 23)/(2, 8, 14, 20)$ for 130M and $(0, 15, 31, 47)/(5, 17, 29, 41)$ for 370M. Each layer supplies channels $\lfloor jH/16 \rfloor$, $j = 0, \ldots, 15$, and each trace supplies width-256 roots starting at $0, 512, 1024, 1536$, covering only 1,024 of its 2,048 positions. The four old/new document-by-layer combinations per model therefore give $8 \cdot 4 \cdot 16 \cdot 4 = 2,048$ roots each, at cap eight. The same dyadic tree is used for all policies. Oracle chooses a positive real constant independently at each node, minimizing leaf count in that tree; it does not optimize arbitrary boundaries or runtime. Stored FP32 coefficients are exact dyadic rationals, with integer admission predicates and no boundary epsilon. The audit checks all 4,796,158 union visited nodes and supplements shared solver predicates with separate certificate checks and a Fraction sample of 256 roots/72,610 nodes.

For aligned dyadic roots of width at least two, deployed singleton leaves occur in sibling pairs. If their count is $n_1$, rescue has exactly

$$K_R = K_D - n_1/2. \tag{93}$$

An exact CPU audit of 16,384 roots verifies this identity and evaluates deployed, rescue, mean and oracle partitions. Each of four document/layer panels per model contains 2,048 roots. Full mean reduces rescue leaf count by 12.244–16.938% for 130M and 34.462–41.493% for 370M. Mean has fewer, equal and more leaves than rescue on 14,044, 2,337 and three roots, respectively: this is an aggregate observation, not rootwise dominance. The oracle and unrounded real mean use exact rational arithmetic on the represented FP32 inputs; GPU wider-node means are rounded and rechecked. The independent rescue timing above tests a distinct empirical quantity.

**Deployed-first fallback and caps.** P3Q tests $H$, which tries deployed then rounded mean before splitting. Its partition coarsens mean's under identical predicates, and its hybrid bank shares deployed entries. The 120 units comprise 48 primary cases, eight adverse cases and 64 additional

cap units; 20 balanced rounds are used for four-policy units and ten for cap units. All gates pass. Table 8 gives the prespecified $M/H$ comparison: fallback is slower in every primary case despite 22.66%/69.41% smaller banks and 6.71%/13.25% smaller retained storage. Required-bank profile spans increase by 0.609/0.429 ms with the hybrid bookkeeping. Rescue is fastest in all eight adverse cases ($R/M = 0.8276$); fallback does not repair that reversal. Across caps $2, 4, 8, 16, 32$, observed time and resident storage decrease in all 32 document/model/policy sequences, but output error does not decrease monotonically with a smaller cap. Every cap passes: this is a cost map, not an accuracy-calibrated cap-selection rule. The panel spans two allocations and four disjoint completion sessions.

### D.3 When subdivision is needed under an explicit-factor contract

P3S compares fixed width 16 with mean (uncentered/centered), exact positive-FP32-grid minimax (centered), centered uniform width two, root-16/cap-32 adaptive centered mean, and an authored sequential FP32 physical recurrence. The three fixed-width routes receive no span-based subdivision. Exact minimax search uses adjacent objective comparisons and neighbor checks on the full positive FP32 anchor grid; all 11,010,048 width-16 roots are checked, charging CPU planning and transfers in fresh timing. The packed PAIR executor and physical boundary carry are common to the anchored routes. There are two warmups, eight fresh and 24 prepared rounds per valid route, with deterministically randomized paired orders.

The exact sufficient all-anchor certificate rules out retaining $1{,}635{,}971/4{,}718{,}592$ 130M roots and $173{,}750/6{,}291{,}456$ 370M roots under the positive-finite bidirectional FP32 factor contract, even with constant statewise centering. This is a coordinate-contract obstruction; physical recurrence lies outside it. Non-triggering roots are not certified safe. Actual zero/nonfinite-factor root counts decline from 2,343,805/304,642 (uncentered mean), to 1,917,852/208,819 (centered mean), to 1,817,616/195,156 (centered minimax), but all three fixed-width routes still fail every full-case gate.

**Table 9:** P3S complete-case fidelity and adaptive/uniform2 comparisons. Storage includes plan, bank and offsets. Fresh absolute times are medians of case timing medians; ratios use paired log times and document intervals.

| Quantity | 130M | 370M |
|---|---|---|
| Fixed16: each of three controls passes | 0/24 | 0/24 |
| Uniform2 / adaptive passes | 24/24 each | 24/24 each |
| Direct physical FP32 passes | 24/24 | 22/24 |
| Uniform2 / adaptive fresh time (ms) | 14.316 / 10.831 | 18.407 / 9.052 |
| Uniform2/adaptive fresh ratio | 1.326 [1.313,1.339] | 1.988 [1.978,1.997] |
| Uniform2/adaptive prepared ratio | 1.316 [1.304,1.327] | 1.806 [1.800,1.811] |
| Adaptive leaf / retained-storage reduction | 45.85% / 31.15% | 82.88% / 65.03% |

Both passing anchored choices have no zero, subnormal or nonfinite coordinate factors. Adaptive improves both time modes and retained/peak storage in every case. Direct physical FP32 is faster where it passes; its two failures are sparse finite readout-tolerance violations, distinct from the fixed-width nonfinite failures. Conversely, some fixed-width channel instances pass the factor check but fail fidelity. Range certification therefore informs a necessary execution change for affected roots without certifying complete output accuracy.

### D.4 Sharing a bank versus omitting it

P3R fixes the root-16/cap-32 mean partition and computes exact feasible positive-FP32-anchor intervals. Per-channel greedy interval cover gives a minimum anchor-key count; the primary policy selects the rounded, clamped, exactly rechecked interior midpoint of each group's common interval. Every materialized comparator deduplicates both channel–anchor-bit keys and complete generated FP32 transition vectors, including cross-channel word matches. Thus sharing is already available to the mean baseline. All 192 policy–case routes pass before and after timing. The exact CPU planner reduces bank storage but is prohibitively slow for fresh calls.

P3T replaces exact interval optimization by the fixed four-candidate GPU cascade $q = 0, 2, 4, 8$, using directed admission and original-mean fallback on the same partition. Its grid is developed

on the existing panel. P3U freezes the rule before its new-document captures and compares single-$q$ alternatives, mean/cascade banks, mean/cascade on demand and physical controls. Candidate admission certifies only accepted candidates; fallback inherits the original rounded-mean predicate. No minimum-key, total-cost or general output-error theorem is asserted for the cascade.

**Table 10:** Same-partition storage contrasts, separately by experiment. Bank/retained reductions use ratios of summed matched-case bytes; the fresh ratio is mean-bank/candidate time. P3U ranges span the separately analyzed model/length groups, not uncertainty intervals.

| Study | Model | Bank reduction | Retained reduction | Fresh ratio |
|---|---|---|---|---|
| P3R exact interior | 130M | 95.07% | 36.71% | $1/1787.3$ |
| | 370M | 96.57% | 28.99% | $1/6022.5$ |
| P3T cascade | 130M | 90.94% | 35.12% | 1.1570 |
| | 370M | 92.19% | 27.67% | 1.0539 |
| P3U cascade | Model/length range | 91.02–95.14% | 28.23–35.20% | 1.0523–1.1917 |

P3R's prepared gains are only $1.0151\times$ / $1.0067\times$; its continuation spans two allocations. P3T uses eight fresh and 24 prepared balanced rounds and all 96 routes pass; peak increments fall only 9.24%/4.49%, much less than bank bytes. P3U uses 20 counterbalanced rounds per mode and shares P3T's allocation. All 864 anchored route–cases pass. A single $q = 2$ candidate already captures most timing benefit: the cascade's extra fresh reduction is only 0.27–0.78%. More importantly, mean on demand is the fastest anchored fresh route in all 108 cases, $1.30$–$1.49\times$ faster than cascade bank on the new groups. Cascade selection adds 2.33–4.04% fresh time once the bank is omitted, with no resolved prepared advantage. Materialized/on-demand outputs and anchors match for all 216 case-policy pairs. The direct/native controls fail ten/eight cases; native is fastest wherever it passes. Thus these tests support a storage decision inside the family, not native competitiveness or a general stability advantage.

### D.5 A100 POLICY-BY-STORAGE BRIDGE AND PRECISION SCOPE

P3V crosses $D, R, M, H$ with bank/on-demand PAIR on 48 primary settings (root 16/cap 32, $L = 2048$), eight adverse settings (370M layer 39, root 256/cap 8), and eight nested long settings ($L = 4096$, root 16/cap 32). Its 64 settings represent 56 distinct captured cores on eight reused documents, measured on one further A100 allocation. A unified planner and complete-vector deduplication replace the earlier hybrid-bank construction. Within policy, bank/on-demand descriptors, anchor bits and outputs agree. All 512 anchored route–cases pass. Each mode has two warmups and 28 counterbalanced rounds over 14 routes, skipping failed routes without replacement. This study measures PAIR only; source-count ledgers do not constitute SOURCE timings.

**Table 11:** P3V fresh policy/storage contrasts. OD means on demand. All rows contain eight documents; primary strata also average three layers. Ratios above one favor the denominator. Brackets give descriptive 95% document intervals for the prespecified OD fallback/mean comparison.

| Panel | OD $D/M$ | OD $R/M$ | OD $H/M$ [95% CI] | $M$ bank/OD |
|---|---|---|---|---|
| 130M primary | 1.632 | 1.160 | 1.0022 [1.0014,1.0030] | 1.650 |
| 370M primary | 1.282 | 1.077 | 1.0004 [0.9998,1.0009] | 1.434 |
| 370M adverse | 0.958 | 0.814 | 1.0457 [1.0427,1.0497] | 1.242 |
| 370M long | 1.274 | 1.053 | 0.9990 [0.9983,0.9998] | 1.393 |

On demand is the fastest anchored storage choice in every case, while fallback is the fastest materialized policy in all primary and long cases. Its bank-byte savings over mean are 25.18%/61.81% in the primary groups, but the corresponding retained-plan savings on demand are only 0.0446%/0.0352%. The primary mean/fallback time differences are likewise small. Rescue wins every adverse case with either storage choice. Mean reduces primary rescue leaf count by 26.32%/31.16%, while declared source local work increases by 10.42%/3.44% and affine work by 15.15%/4.81%. Fewer leaves, fewer transition products and lower latency are distinct quantities.

The physical bridge includes unchanged native, four sequential controls crossing FP32/FP64 state and readout accumulation while retaining FP32 generated coefficients, and sequential A64, which

also widens coefficient and source generation. Native passes 48/56 distinct cores; each FP32-coefficient control passes 46/56. Failures are finite readout coordinate exceedances, with final states and relative-$\ell_2$ checks passing. A64 and anchored routes pass all 56. A64/mean-OD fresh ratios are $1.404, 1.512, 0.862, 2.101$ for the four table strata; native is faster whenever it passes. This is a comparison with a sequential template. The H100 native-family parallel precision controls in Appendix C close that comparator gap and do not support an anchored accuracy–cost advantage.

**Evidence and verification scope.** Compact evidence records preserve per-case timings, gates, plans, memory descriptors and protocol identities; the supplement provides the experiment index and regeneration paths. Audits verify export/source hashes and recompute table estimates. Saved full arrays are rechecked where retained; other passing cases retain complete-array metrics and hashes, which are not a later GPU replay. In P3V, saved input/reference cores are independently replayed in CPU FP64 on eleven cores, and retained failure/paired-mean arrays are rechecked. Neither these audits nor the multiple historical continuation sessions establish independent full-panel GPU replication.

# E INDEPENDENT STRUCTURAL AND FINITE-PRECISION CHECKS

These CPU checks accompany the proofs; they are not new GPU experiments or substitutes for general mathematical arguments. The full constructed panels and tables are retained in Detailed Supplement Section S12, and the checkpoint oracle-leaf audit in Section S13.

**Work, storage and exact-output checks.** Backward dependency tracing from all source outputs on the specified Brent–Kung graph agrees with $F = 2W - \log_2 W - 2$ and $H = W - 1 - \log_2 W$ at all 14 dyadic widths $W = 2, \ldots, 16384$. This is the alternative graph of Appendix B.3, not the measured readout tree with $7w-6$ versus $6w-4$ operations. At $W = 2048$, $N = 16$ and four bytes per scalar, materializing the consumed physical segment coefficients costs 261,312 bytes versus 704 bytes for the length bank. Including both compact coordinate directions raises the latter to 262,592 bytes; retaining only forward coordinates gives 131,648 bytes and requires inverse generation. A physical on-demand implementation need not store the materialized coefficient table. These are declared array payloads, not whole-execution memory minima.

For 96 binary64 fixtures spanning widths 2–256, $N \in \{3, 8\}$, varying selective inputs and zero/nonzero incoming states, all readouts and final states agree with direct recurrence to maximum absolute errors $1.25 \cdot 10^{-14}$ and $1.38 \cdot 10^{-14}$. Separate exact enumeration checks the generation-inclusive partition formulas on 711 dyadic cuts and 2,278 positive-increment inputs with 13,668 budget cases. These finite checks supplement the induction and partition proofs; they do not certify every floating-point execution.

**Range control with unchanged physical dynamics.** The construction uses equal numbers of increments $1 \pm 1/4$, a mean anchor of one, and includes the zero prefix. Raw clock span is $T$, alternating mean-residual span is $1/4$, and two-run mean-residual span is $T/8$. These patterns, constant increments, and three fixed balanced permutations give 78 exact checks over 13 dyadic lengths 16–65,536. At $T = 65,536$ and $A_n = -1$, alternation reduces the $\log_{10}$ factor range from 28,461.923 to 0.108574. Temporal ordering remains decisive: the two-run residual span is 8,192.

CPU FP32 execution uses six patterns at $T \in \{16, 128, 1024, 8192, 65536\}$, $N = 4$, $A = (-0.1, -0.25, -0.5, -1)$, bounded sources/readouts and nonzero incoming state. Centering also transforms that incoming state. All elements are compared with FP64 direct recurrence on the same FP32-rounded inputs, at tolerance $5 \cdot 10^{-5} + 5 \cdot 10^{-4}|r|$; this synthetic gate differs from the checkpoint gate.

**Table 12:** Constructed FP32 range control. Every route retains all 30 inputs. All failures are nonfinite readouts or final states; none is a finite tolerance failure.

| Route | Pass/total | Nonfinite |
|---|---|---|
| Direct physical | 30/30 | 0 |
| Raw prefix | 6/30 | 24 |
| Centered raw prefix | 12/30 | 18 |
| Mean anchored | 27/30 | 3 |
| Centered mean anchored | 28/30 | 2 |

Mean failures are two-run inputs of lengths 1024, 8192 and 65,536; centering repairs the first. The improvement is relative to explicit raw cumulative factors. Physical recurrence passes every input and constructs no such factors. The test supports drift removal and the temporal obstruction, not an accuracy advantage over native or an output-error theorem from span alone.

**Exact anchor decisions and partition margins.** For all 3,279 sequences of lengths 1–7 over $\{1, 2, 3\}$ and seven rational budgets, an independent rational reference evaluates the residual envelope at every positive pairwise-intersection slope. All 22,953 keep/repair/impossible decisions agree: 10,549 are mean-feasible, 1,056 repairable and 11,348 impossible. The mean/lower-bound screens resolve 20,773; the other 2,180 use the interval solve. Exact checks also agree for 3,276 nontrivial mean inequalities, 31 sharpness constructions, 15,526 dyadic partition cases, and 1,940 finite-grid sharing families. These are constructed-case counts, not checkpoint prevalence or timing measurements.

The checkpoint margin audit reuses the 16,384 width-256 CPU roots of Appendix D.2, with rate-scaled cap 8 and 1,120,399 oracle leaves. Exact integer-hull evaluation recomputes every oracle optimum and maximizing three-prefix witness from the captured FP32 words. The sufficient margin certifies 79.08–96.26% of oracle leaves, including automatic width-two cases; coverage among leaves of width at least four is 59.0–67.5%. Only 17.47–20.13% of uncertified leaves actually fail their mean. Mean's excess leaf count is 0.67–4.06%, while the worst-case bound is 10.9–38.3 times that observed excess. Of rejected oracle leaves, 95.85–98.74% require only one bisection. Thus the bound is valid but conservative; the observed near-oracle counts additionally depend on the input and shallow refinement. No latency or held-out prediction follows from this post-hoc audit.

# F Evidence map and reproduction scope

The main paper retains the proofs and result summaries needed for its claims. The separate Detailed Supplement preserves earlier full protocols and tables by question, as mapped below. Each study keeps its own input selection, fidelity gate, timing denominator and preparation boundary; observations are never pooled as independent cases merely because a later implementation reuses them.

**Table 13:** Reading map. S-prefixed sections belong to the separate Detailed Supplement. Numerical studies are supporting evidence, not substitutes for the mathematical proofs.

| Main-paper question | Retained location and detailed evidence |
|---|---|
| Exact identity, feasibility, mean/partition guarantees | Appendix A; finite checks here and Supplement S12–S13. |
| Work removed, on-demand generation, whole-algorithm cost | Appendix B; distinct source schedules and all primitive categories remain explicit. |
| Current matched policy/representation and strong physical controls | Appendix C: complete H100 protocol, qualified denominators and unfavorable physical comparison. |
| Policy-by-storage interaction on A100 | Appendix D.5; complete P3V tables and retained-array audit in S1. |
| Range choices and materialized sharing | Appendices D.3–D.4; complete protocols/results in S2, S9–S10. |
| Representation, packing, rescue and fallback controls | Appendix D; S3–S8 retain complete earlier tables and settings. |
| Document/layer coverage | Appendix D.2; S11 selection and layer diagnostics. |
| Earlier integration and numerical counterexamples | S14, with the conclusions retained below; S15 inventories study-specific evidence. |

**Adverse evidence retained.** Earlier full-model integrations of pair/source/FFT cores were all slower than their matched native path. Nine continuation cases passed their screens, whereas all four public BF16 cases failed a tensor screen. Source-only was 2.29–2.50% slower than pair in the continuation cases; its 3.2–3.4% reduction on public 130M accompanied a failed screen. A separate readout-cancellation fixture reached relative output L2 error 0.252 while its maximum absolute error remained below $7.6 \cdot 10^{-7}$. Earlier restricted-witness tests improved the optimizing oracle by 2.46%/12.85%, but direct fixed-cap decision baselines reduced that benefit to approximately zero–1.17%; a candidate planner was 6.503%/5.953% slower despite fewer checks. These outcomes limit runtime and accuracy interpretations of the structural results and are not displaced by newer favorable comparisons.

**Recorded gates versus independent re-evaluation.** P3W retains all 64 committed records and the protocol, input/source bindings and qualification receipts. Independent aggregation reproduces 216 paired comparisons, 112 absolute-time estimates and 168 memory medians. Its compact export excludes raw arrays/binaries, so recorded whole-array gates and output hashes are verified without a new local elementwise reconstruction of those outputs. P3V retains 63 failure-related readout/final-state pairs from 11 cores; CPU FP64 regeneration and those complete saved-output checks are separate from metrics-only checks on other routes. P3U's local full-array re-evaluation covers two diagnostic cores, not every case. Earlier P3O/P3P/P3Q full-array audits do not upgrade the validation scope of later metrics-only exports. No audit constitutes a new GPU run.

**Input and allocation dependence.** P3O extends document/layer coverage; P3P–P3T reuse that selection. P3U adds eight policy-performance-held-out documents and old-input bridges on P3T's allocation. P3V uses a later A100 allocation on eight reused documents. P3W recaptures those documents on one H100 allocation: $A$, deployed anchors and initial states agree with A100, while $u, \Delta, B, C$ do not. Nested lengths, resumed sessions and repeated timings do not supply additional independent documents. The corpus is not claimed disjoint from model training.

**Reproduction artifacts.** The experiment records were frozen before table generation. The anonymous code supplement supplies builders for 34 retained table files with compact P3R–P3W inputs, finite exact/arithmetic checks, and preserved P3W GPU computational source with a captured-input adapter. Its README identifies retained intervals, summary-level inputs and omitted-array analyses. Supplement S15 preserves the original research-artifact identifiers. Weights, corpus assets and raw paper tensors are excluded. Table reconstruction, GPU execution and independent output-array audits remain distinct reproducibility claims.