telecom

MDPI

Article

Empirical Tail-Latency Characterization of GPU-Accelerated 5G NR LDPC Decoding

Sooyoung Jang [1] and Eunkyung Kim [2,*]

[1] Department of Computer Engineering, Hanbat National University, Daejeon 34158, Republic of Korea; syjang@hanbat.ac.kr
[2] Department of Artificial Intelligence Software, Hanbat National University, Daejeon 34158, Republic of Korea
* Correspondence: ekim@hanbat.ac.kr

Abstract

GPU LDPC latency comparisons across implementations are challenging due to variations in decoder schedules, batch granularity, and timing boundaries. Batch completion was characterized for FP32-layered and flooding decoders on a GPU, considering block acquisition, matched outcomes, and three timing boundaries. The schedule study included 600,000 observations. Flooding exhibited higher P50 and P99 in 15 full-boundary matched-cap comparisons, while J_{99} and R_{99} did not display uniform schedule ordering. For words satisfying the syndrome under both schedules, layered decoding achieved first satisfaction earlier in every estimable 2- and 4-dB cell. Layered endpoint levels varied among full, decode-only, and transfer-only boundaries. Over six sessions, full-boundary J_{99} increased by 234.7 µs from batch 64 to 128 (pointwise descriptive 95% percentile interval, 196.1–281.5 µs), which contrasted with two other designs. These results pertain to a single dynamically clocked Windows Subsystem for Linux 2 (WSL2) GPU and do not establish fixed-clock or cross-platform behavior, radio-interface latency, causal attribution, or worst-case bounds.

Keywords: 5G New Radio; LDPC decoding; layered scheduling; flooding scheduling; GPU acceleration; tail latency; timing boundaries; empirical quantiles

1. Introduction

Fifth-generation (5G) New Radio (NR) supports services with different throughput, reliability, and delay requirements. Ultra-reliable low-latency communication (URLLC) is a prominent example, but its service latency includes radio-interface, protocol, transport, and application processing rather than a single decoder call [1,2]. Decoder measurements can therefore inform a physical-layer budget without demonstrating URLLC compliance.

Low-density parity-check (LDPC) codes, introduced by Gallager [3], protect the NR data channel [4]. Their sparse quasi-cyclic graphs expose parallelism that maps naturally to graphics processing units (GPUs). Existing GPU implementations demonstrate standards-oriented decoding, high throughput, and integration with virtualized radio-access software [5–10]. Cross-platform results also show that data movement and workload granularity affect accelerator performance [11]. Reported latency values nevertheless depend on the graph, precision, stopping rule, batch definition, schedule, and timed software boundary.

Real-time GPU research examines scheduling, interference, resource isolation, and analyzable execution. TimeGraph controls GPU work at the driver level, whereas GPUSync

Check for updates

Academic Editor: Minseok Kim

Received: 27 July 2026
Revised: 3 September 2026
Accepted: 11 September 2026
Published: 15 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

coordinates accelerator access in multicore, multi-GPU systems [12,13]; Gheibi Fetrat et al. survey the broader architectural and scheduling constraints [14]. The present study instead characterizes the empirical distribution of one declared decoder-batch boundary without an intentionally co-scheduled GPU workload. The measurements are not response-time bounds or schedulability tests.

The studies reviewed here do not jointly provide three forms of evidence for one controlled decoder setting: whether schedule differences persist in both latency endpoints and matched decoding outcomes, how those endpoints depend on the declared software timing boundary, and whether an observed batch-tail direction recurs across acquisition sessions. This empirical gap is narrower than the problem addressed by real-time schedulability research. It concerns measured decoder-batch distributions rather than worst-case response-time or interference guarantees.

The primary comparisons evaluate schedule latency, matched schedule outcomes, three timing boundaries, and the batch-64-to-128 direction across six acquisition sessions. Sensitivity analyses vary the algorithm, stopping rule, input mode, channel condition, graph, lifting size, and operating state. Block-placement and CUDA Profiling Tools Interface (CUPTI) records provide supplementary diagnostic evidence: they constrain interpretation but do not identify a hardware mechanism.

The study addresses four questions:

RQ1. How do P50, P99, absolute P99–P50 separation J_{99}, and the P99-to-P50 ratio R_{99} differ between the tested layered and flooding implementations at the tested batches and fixed iteration caps?

RQ2. How do the two implementations compare in information-bit error rate (BER), information-word block error rate (BLER), final syndrome success, and first syndrome satisfaction when they decode the same codewords and noise?

RQ3. For the layered implementation, how do these endpoints differ among the full, decode-only, and equal-volume transfer-only boundaries, and does the batch-64-to-128 direction recur across acquisition sessions?

RQ4. How sensitive are the measured endpoints to algorithm, channel, input, graph, lifting size, stopping rule, and starting state?

The study makes three empirical and methodological contributions. Methodologically, it applies established P50, P99, J_{99}, and R_{99} endpoints within explicitly declared software boundaries and uses paired block-level resampling to preserve the acquisition-block structure in contrasts. Comparatively, it relates schedule latency to outcomes from matched codeword/noise pairs rather than assuming that equal iteration caps imply equal decoding behavior. Empirically, it finds under the tested conditions that flooding is higher at P50 and P99 in every same-cap schedule cell, whereas J_{99} and R_{99} have no uniform schedule ordering, and that the layered batch-64-to-128 J_{99} direction differs across experimental designs.

Internal checks establish consistency among the encoded words, compiled graph, CPU reference, and GPU outputs; they are not an independent 3GPP conformance certification. The scope is synchronous decoder-batch completion for the tested FP32 implementations on one dynamically clocked RTX 4070 Ti SUPER under WSL2. Native Linux, a second GPU, fixed clocks, and hardware performance counters were unavailable, so the results do not support platform-wide generalization or causal microarchitectural attribution.

2. Background and Related Work

2.1. NR LDPC Graphs and Min-Sum Decoding

Irregular LDPC constructions can approach channel capacity while retaining sparse-graph decoding [15]. NR defines two quasi-cyclic base graphs for different information-

block and rate regimes [4]. The primary configuration uses Base Graph 1 (BG1), lifting size $Z = 384$, and lifting-set index $i_{\mathrm{LS}} = 1$. We selected $Z = 384$ because it is the largest standardized lifting size, provides a high-workload case on the available device, and matches the baseline scaling configuration. It yields 26,112 internal values and 121,344 lifted edges per codeword. BG1 contains 46 base-graph rows, 68 columns, and 316 nonzero circulant permutation submatrices. Its systematic region has $22Z$ bits, including the first $2Z$ punctured positions. Controls use BG1 at $Z = 192$ and BG2 at $Z = 384$; they test sensitivity to lifting size and topology but do not establish representativeness across all NR configurations.

Let $L_{\mathrm{ch},n}$ denote the channel log-likelihood ratio (LLR), r_{mn} a check-to-variable message, $\mathcal{N}(m)$ the variables adjacent to check m, and $\mathcal{M}(n)$ the checks adjacent to variable n. For extrinsic values q_{mn}, the min-sum check rule is

$$r_{mn} = s_{mn}\, g\Bigg(\min_{n' \in \mathcal{N}(m)\setminus\{n\}} |q_{mn'}|\Bigg), \qquad s_{mn} = \prod_{n' \in \mathcal{N}(m)\setminus\{n\}} \operatorname{sign}(q_{mn'}). \tag{1}$$

Here, $\operatorname{sign}(x) = +1$ for $x \geq 0$ and -1 otherwise, matching the two-valued sign used by both kernels. The corresponding sum–product magnitude is

$$2\tanh^{-1}\Bigg(\prod_{n' \in \mathcal{N}(m)\setminus\{n\}} \tanh(|q_{mn'}|/2)\Bigg).$$

It does not exceed the minimum magnitude used by min-sum, so min-sum tends to overstate check-message reliability. Standard min-sum uses $g(x) = x$; normalized min-sum uses $g(x) = \alpha x$, with $0 < \alpha < 1$ attenuating that overstatement; and offset min-sum uses $g(x) = \max(x - \beta, 0)$ [16]. The algorithmic-control study tests normalized min-sum at $\alpha \in \{0.6, 0.7, 0.75, 0.8, 0.9\}$, standard min-sum, and offset min-sum at the fixed engineering control $\beta = 0.5$.

The primary value $\alpha = 0.75$ was fixed a priori as a workload parameter. It was not derived by density evolution and is not presented as theoretically optimal. An optimum would require a declared code, rate, channel, precision, stopping rule, and objective. Normalization changes message magnitude without changing the fixed kernel's loop bounds, whereas parity-check early stopping can change the executed iteration count. The error/convergence experiment therefore accompanies latency sensitivity so that timing is not mistaken for decoding quality; its finite operating points do not turn $\alpha = 0.75$ into an optimized value.

Layered scheduling makes an updated belief available to the next base-graph row [17]. Initialize $r_{mn}^{(0)} = 0$ and $L_n^{(1,0)} = L_{\mathrm{ch},n}$; after the 46th row, set $L_n^{(\ell+1,0)} = L_n^{(\ell,46)}$. If $L_n^{(\ell,m-1)}$ is the belief immediately before row m in iteration ℓ, the implementation reads the message retained for that row, applies Equation (1), and updates in place:

$$q_{mn}^{(\ell)} = L_n^{(\ell,m-1)} - r_{mn}^{(\ell-1)}, \qquad L_n^{(\ell,m)} = q_{mn}^{(\ell)} + r_{mn}^{(\ell)}. \tag{2}$$

The superscript on the retained message identifies its value before the row update; rows already visited in iteration ℓ have already changed the belief. One BG1 iteration comprises 46 ordered row updates. Each row traverses its incident lifted edges once to form signs and the first and second minima, then again to write messages and beliefs. The layered kernel thus performs $2(316Z)$ lifted-edge traversals and 46 block barriers per codeword per fixed iteration.

https://doi.org/10.3390/telecom7050120

The flooding implementation does not expose within-iteration updates. It initializes $r_{mn}^{(0)} = 0$ and $L_n^{(0)} = L_{\mathrm{ch},n}$ and retains an immutable channel-LLR array and distinct old and new check-message buffers. Every check update in iteration ℓ reads only the old buffer,

$$q_{mn}^{(\ell)} = L_{\mathrm{ch},n} + \sum_{m' \in \mathcal{M}(n) \setminus \{m\}} r_{m'n}^{(\ell-1)}, \qquad L_n^{(\ell)} = L_{\mathrm{ch},n} + \sum_{m \in \mathcal{M}(n)} r_{mn}^{(\ell)}, \tag{3}$$

and writes the new buffer through Equation (1). A complete check phase precedes a column-oriented variable phase; the buffers swap only after both phases finish. The check phase makes two lifted-edge traversals and the variable phase one, for $3(316Z)$ traversals and two phase barriers per fixed iteration. At $Z = 384$, the graph has 121,344 lifted edges, so one iteration performs 242,688 traversals in the layered kernel and 364,032 in the flooding kernel. A syndrome-observed iteration in either schedule adds one 121,344-edge sweep and three barriers. These operation counts are schedule-specific implementation facts. They do not establish an iteration-equivalence factor or a layered speedup; convergence is compared empirically at the same caps and channel conditions.

2.2. GPU LDPC Decoders and Real-Time GPU Systems

Published GPU LDPC decoders differ substantially in graph configuration, numeric precision, schedule, stopping policy, batch granularity, and treatment of data transfers. Li et al. use a scaled parallel layered decoder, Ling and Cautereels report layered integer and floating-point variants, Tarver et al. and Dai et al. study flooding min-sum implementations, and Liu and Zhao use a memory-reduced layered design [6–10]. Their reported operating points range from per-iteration decoder measurements to host boundaries that include transfers. DecodeX likewise separates kernel and orchestration boundaries across several accelerator platforms [11]. These differences preclude a direct ranking of the published latency values.

System-level studies place accelerator performance in a wider radio-access context. Kelkar et al. describe a GPU-hosted 5G radio access network (RAN), Notari et al. review Open RAN deployment and cost, Arabian et al. compare waveform families, and Lin et al. discuss accelerated computing in an AI-native RAN [18–21]. Cellular latency analyses allocate delay across network and protocol stages [2], whereas real-time GPU research examines scheduling and response-time control under contention [12–14]. The present work addresses a narrower question: empirical quantiles of a specified decoder-batch boundary without an intentionally competing GPU workload.

Appendix A provides four detailed comparison matrices (Tables A1–A4). Values are included only when a source reports them; "N/R" denotes not reported. The matrices document configuration and timing-boundary differences rather than imply a performance ranking.

3. Materials and Methods

3.1. Platform and Decoder Implementation

All experiments used an Intel Core i7-14700KF host and an NVIDIA RTX 4070 Ti SUPER with 16 GiB. The software environment was Ubuntu 24.04.4 under WSL2, kernel `6.18.33.1-microsoft-standard-WSL2`, CUDA 12.0 with `nvcc` 12.0.140, and NVIDIA driver 610.62. The GPU has compute capability 8.9 and 66 streaming multiprocessors (SMs), and the device code targets `sm_89`.

FP32 provides a validation-oriented reference rather than a model of commercial decoder efficiency. In this implementation, the active layered state is one $68Z$-value belief array and one $316Z$-value message array, or $384Z$ FP32 values per BG1 codeword. At $Z = 384$, this is 147,456 floats, 0.5625 MiB per codeword, or 288 MiB at batch 512 before

output and ancillary buffers. The implemented flooding state contains the current belief, an immutable $68Z$-value channel array, and two $316Z$-value message buffers: $768Z$ floats, 1.125 MiB per codeword, or 576 MiB at batch 512. These buffer choices are not lower bounds for every implementation of either schedule. The common implementation allocates the $768Z$ flooding superset for both schedules, so 576 MiB—not 288 MiB—is the decoder-state allocation in either batch-512 timing condition. The smaller layered value describes its active payload requirement, not the allocation used for the comparison. INT8 would use one quarter of these payload bytes, but a width change also alters traffic, instruction mix, occupancy, and possibly convergence. We therefore make no FP16 or INT8 timing generalization.

For BG1 $Z = 384$, either decoder kernel launches one CUDA block per codeword with 384 threads (12 warps). Batch size B means `gridDim.x`$= B$ for one decoder-kernel grid: the codewords are submitted together but need not be resident simultaneously. The hard-decision kernel uses a separate grid. Static compiler evidence gives 40 registers per thread for layered decoding and 38 for flooding; both kernels use 4 bytes of shared memory per block and have no spills. With the device limits of 1536 threads and 65,536 registers per SM, either kernel has a static resource-limit ceiling of four decoder blocks per SM, or 264 across 66 SMs. This is not achieved occupancy and does not predict a latency change at batch 264. The schedules have different instruction and storage paths; achieved occupancy and memory traffic require hardware-counter measurements, which were unavailable in this environment.

3.2. Inputs and Experimental Factors

The base-graph tables were generated from the lifting-set-index-1 matrices distributed with NVIDIA Sionna. Runtime validation checks dimensions, index ranges, row uniqueness, and column-adjacency consistency. These checks establish consistency between the compiled tables and both decoder implementations; they are not an independent reconstruction or certification of the standardized shifts.

A systematic quasi-cyclic encoder specialized to the compiled BG1 and BG2 matrices generates random codewords for both graphs. Its four leading parity groups use the sparse structure of those matrices, and lower parity groups follow by sparse circulant multiplication. Every generated codeword is checked against the corresponding expanded parity-check matrix. A zero syndrome establishes internal consistency between this encoder and the compiled matrix; it is not an independent 3GPP conformance test. For a linear code on output-symmetric AWGN, exact-arithmetic min-sum decoding with sign-symmetric updates has a codeword-invariant error distribution under the corresponding sign transformation. That symmetry concerns decoding outcomes; it does not prove execution-time invariance for a finite-precision GPU kernel with data-dependent control flow. The all-zero-to-random timing comparison is therefore an empirical sensitivity check rather than a consequence of the symmetry theorem. Random valid codewords remain the input for the schedule comparison.

The first $2Z$ positions are punctured; every remaining bit is mapped to a unit-energy BPSK symbol and passed through additive white Gaussian noise. The experiments use one fixed transmission rule for each graph configuration: no filler bits, shortening, repetition, additional puncturing, or cyclic-redundancy-check (CRC) bits are applied. Thus, the configurable NR rate-matching chain is not implemented beyond the fixed first-$2Z$ omission, and each remaining mother-code bit is transmitted once. With k_b systematic base-graph columns and n_b total columns, the effective transmitted rate used for noise scaling is $R = k_b/(n_b - 2)$. BG1 $Z = 384$ has $K = 8448$, transmitted $N = 25{,}344$, and an internal 26,112-value lifted vector; BG1 $Z = 192$ has $K = 4224$, transmitted $N = 12{,}672$,

and 13,056 internal values. BG2 $Z = 384$ has $K = 3840$, transmitted $N = 19{,}200$, and 19,968 internal values. Thus, $R = 22/(68-2) = 1/3$ for BG1, and $R = 10/(52-2) = 1/5$ for BG2. For $E_b/N_0 = \gamma_b$,

$$\sigma^2 = \frac{1}{2R10^{\gamma_b/10}}, \qquad L_{\mathrm{ch}} = \frac{2y}{\sigma^2}. \tag{4}$$

The first $2Z$ punctured LLRs are zero. The master seed is 20260828. Each timing input seed is derived from the study, global block, graph, lifting size, batch, E_b/N_0, and input mode. Schedule, timing boundary, clock mode, and iteration cap are omitted. A codeword/noise batch is regenerated outside timing for each visit, is byte-identical across its intended within-study contrasts, and is reused for that visit's 10 warm-ups and 200 retained trials. Each configuration visit runs in a fresh benchmark process and CUDA context. Buffer allocation, context establishment, topology upload, and input generation occur outside the declared timing boundary; process and stream state are not reused across visits.

The schedule-timing study executes exactly 5, 10, or 20 iterations without a syndrome sweep. These levels span a fourfold controlled-work range; they are neither standardized traffic settings nor assumed to be equivalent across schedules.

The matched error/convergence study is untimed and uses no warm-ups. Each 20,000-codeword cell is decoded in chunks of at most 256 codewords (78 full chunks and a final 32-codeword chunk), rather than as one 20,000-block grid. It runs the full cap but evaluates the syndrome after every iteration. It records final-output syndrome validity separately from whether a codeword ever satisfied the syndrome and reports the first-satisfied iteration only for the latter codewords. It does not stop at first success because later min-sum iterations can change the hard decision. The timing-control early-stopping mode remains distinct: it stops a codeword at first syndrome satisfaction, subject to a cap of 20, so the slowest unfinished codeword can govern synchronous batch completion.

For W decoded codewords with K information bits, let $e_w = \sum_{i=1}^{K} \mathbf{1}\{\widehat{u}_{wi} \neq u_{wi}\}$. The reported information-bit and information-word error rates are

$$\mathrm{BER} = \frac{\sum_{w=1}^{W} e_w}{WK}, \qquad \mathrm{BLER}_{\mathrm{info}} = \frac{1}{W}\sum_{w=1}^{W} \mathbf{1}\{e_w > 0\}. \tag{5}$$

Final syndrome success is $W^{-1}\sum_w \mathbf{1}\{H\widehat{\mathbf{c}}_w^{\mathsf{T}} = \mathbf{0}\}$ for the final decoded mother-code vector. Thus, $\mathrm{BLER}_{\mathrm{info}}$ is an information-word error rate, not transport-block BLER, and syndrome failure is a separately reported event.

3.3. Timing Boundaries and Instrumentation

End-to-end batch latency refers only to the full host boundary (Figure 1). It is the host monotonic interval from immediately before the initial CUDA-event enqueue and pinned host-to-device (H2D) submission until the final stream-synchronization call returns after reset, decoding, hard decision, and device-to-host (D2H) copies of information bits and iteration counts. Allocation, graph upload, encoding, noise generation, input validation, clock settling, telemetry setup, event-time queries, and file output are outside this boundary.

The decode-only boundary prepares input and reset state and synchronizes before its start. It times the decoder submission and blocking stream synchronization, then performs hard decision and D2H outside timing. The transfer-only boundary prepares zero output buffers, then times the same H2D and D2H byte volumes as the full path plus final synchronization, without reset, decode, or hard decision. It measures completion of those transfer calls, not a subtraction-derived transfer contribution.

The full schedule comparison is an implementation-level comparison, not a pure decoder-kernel contrast. After H2D has populated the belief array, layered reset clears one

message array, and the iteration counters. Flooding H2D instead populates the immutable channel array; its reset also copies that array device-to-device into the belief array, clears two message arrays, and clears the iteration counters. The full-boundary schedule contrast therefore includes the state-preparation asymmetry of these implementations; it is not an unavoidable cost of every flooding design. Decode-only timing begins after the corresponding reset, but the three-boundary ablation itself is layered only.

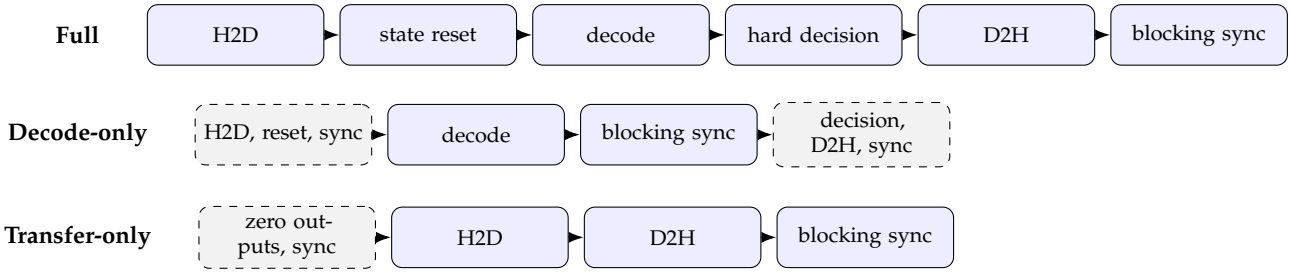


Figure 1. Host timing boundaries. Solid boxes lie inside the host monotonic interval; dashed boxes are excluded. Same-stream CUDA events delimit the named device operations. The final synchronization is host-timed and measures blocking while queued work completes.

Host-call timing ends when an asynchronous CUDA runtime API returns; the final synchronization field instead measures the blocking wait while queued work completes. Same-stream CUDA events separately delimit H2D, reset, decode, hard decision, and D2H activity. The common-clock enclosure check described in Section 4.2 failed under WSL2. Raw event intervals are therefore reported only as descriptions on the CUDA-event clock; they were not cross-calibrated to host elapsed time. Neither a host–event difference nor the synchronization wait are interpreted as an independent execution or synchronization component.

Two perturbative diagnostics provide additional timing context for layered decoding. Block instrumentation records the SM identifier and `%globaltimer` at the start and end of each CUDA block. Because cross-SM timestamp comparability was not validated, analysis is restricted to assignments, within-SM durations, and within-SM overlap. CUPTI activity records correlated CUDA runtime API, synchronization, kernel, and memory-copy intervals for the three layered boundaries. Its inclusive runtime interval overlaps the synchronization call, and host- and device-origin timestamps are not subtracted. Detailed timer fields and diagnostic validation are given in Appendix C.

3.4. Experimental Design and Inclusion Criteria

Table 1 summarizes the experiments. The schedule-latency, matched-outcome, timing-boundary, and session-replication studies provide the primary comparisons. Algorithmic, configuration, and operating-state studies assess sensitivity, while block-placement and CUPTI studies provide diagnostic evidence only.

Within each timing block, a seeded permutation determines visit order. Standard timing visits use 10 warm-ups followed by 200 retained observations; the error study is untimed, and each warmed CUPTI invocation contains 100 traced trials. The schedule and boundary studies contain 100 ordered blocks per configuration. Six additional acquisition sessions, each initiated separately, contain 40 blocks per configuration at batches 64 and 128. As stated above, every configuration visit within a session runs in its own benchmark process and CUDA context. Conditions within a batch share input seeds, whereas the two batches are separately seeded and therefore block-matched rather than input-paired.

Table 1. Experimental design. NMS: normalized min-sum. Timing studies use 10 warm-ups and 200 retained observations per visit; CUPTI trials form a separate perturbative diagnostic sample.

Study	Design	Sample	Purpose
Schedule latency	Layered and flooding; batches 16, 64, 128, 256, and 512; caps 5, 10, and 20	600,000 timings	Same-cap latency comparison
Schedule outcomes	Inputs matched across schedules and caps 5, 10, and 20; 0, 2, and 4 dB	360,000 rows; 60,000 unique word/noise pairs	Error and convergence comparison
Timing boundaries	Layered full, decode-only, and transfer-only; batches 16, 64, 128, 256, and 512	300,000 timings	Software-boundary sensitivity
Session replication	Six acquisition sessions; four conditions; batches 64 and 128	384,000 timings	Between-session consistency of the batch contrast
Operating state	Idle-cooled and preheated starts; two boundaries; 30 cycles	24,000 timings	Starting-state sensitivity
Block placement	Instrumented and uninstrumented decode-only kernels	72,000 timings	Within-SM placement diagnostic
CUPTI Activity	Three layered boundaries; batches 64, 128, 256, and 512	3600 traced trials	Perturbative API and device-timeline diagnostic

The baseline scaling and component-timing studies used a layered-only executable and the smaller layered allocation. The schedule/boundary and session-replication studies retained BG1 $Z = 384$, FP32 normalized min-sum with $\alpha = 0.75$, the 2-dB channel condition, 20-iteration layered kernel geometry, and the full host boundary. Their executable additionally supported flooding and multiple boundaries and allocated the larger flooding-state superset during layered visits. The designs also used different input seeds and either 100 or 40 acquisition blocks. Comparisons among these study groups therefore assess consistency across specified designs; they are not exact software reproductions or estimates of a single isolated design change.

The operating-state study uses 30 matched batch-512 cycles. Each cycle randomizes idle-cooled and preheated starts and then randomizes full and decode-only visits within each state. The idle condition requires 60 s at or below 45 °C with at most 5% GPU utilization; preheating uses untimed decoder executions to reach 58–62 °C. Inputs are matched across the four visits. This contrast measures the combined effect of temperature, clock, power, and recent workload history rather than temperature alone. Continuous NVIDIA Management Library (NVML) telemetry records temperature, clocks, power, P-state, utilization, aggregate PCIe rates, and clock-event masks at a nominal 100-ms interval.

The block-placement study compares instrumented and uninstrumented kernels with matched inputs. The instrumented kernel produced identical decoded outputs, no spills, and the same static occupancy limit. Start and end SM identifiers agreed for every block, but cross-SM timestamp comparability was not established. The resulting evidence is therefore limited to SM assignments and within-SM timing. The CUPTI diagnostic recovered the expected correlated API, synchronization, kernel, and memory-copy records with zero dropped records. Neither diagnostic is part of the unprofiled latency sample or supplies hardware performance counters.

Before acquisition, self-tests validate graph dimensions and adjacency, encoded-codeword syndromes, deterministic inputs, state reset, fixed-versus-early output con-

sistency, and CPU–GPU agreement for both schedules. A timing block enters analysis only when it contains the planned configuration visits and observation counts, finite timings, consistent paired-condition identifiers, stable decoded outputs, and the required telemetry. No observation is trimmed, winsorized, or excluded because of its magnitude. The inferential samples contain the prespecified retained observations from complete experimental units; warm-up observations and the incomplete operating-state cycle are excluded.

Two separately executed 360,000-row outcome acquisitions used the same 60,000 deterministic codeword/noise pairs: 20,000 pairs at each of three channel conditions, each decoded by two schedules at three caps. Corresponding decoded outcomes and convergence fields were identical for every row identifier. The analysis uses one acquisition; pooling both would duplicate every input and decoding condition without adding an independent input realization. The operating-state sample contains 30 complete cycles. One additional cycle ended when a software-power condition appeared; it is excluded from the primary sample and enters only the omission/substitution sensitivity analysis described in Appendix C. Fixed-clock control and privileged performance-counter measurements were unavailable and contribute no inferential observations.

3.5. Statistical Analysis

For sorted observations $T_{(1)} \leq \cdots \leq T_{(N)}$, quantiles use the nearest-rank convention

$$\widehat{Q}_p = T_{(\lceil pN \rceil)}. \tag{6}$$

For even N, Equation (6) therefore defines P50 as the lower of the two central order statistics, rather than their average. The four primary quantitative endpoints are P50, P99, J_{99}, and R_{99}:

$$J_{99} = \widehat{Q}_{0.99} - \widehat{Q}_{0.50}, \qquad R_{99} = \frac{\widehat{Q}_{0.99}}{\widehat{Q}_{0.50}} = 1 + \frac{J_{99}}{\widehat{Q}_{0.50}}. \tag{7}$$

P95, P97.5, P99.5, P99.9, and maximum are used only as finite-sample distribution or survival diagnostics rather than as additional primary endpoints. J_{99} has time units and scales by c if all observations are multiplied by c. R_{99} is the P99-to-P50 ratio; it is dimensionless and unchanged by that pure scaling. The median-relative separation is $R_{99} - 1 = J_{99}/\widehat{Q}_{0.50}$. This identity is algebraic and does not dictate empirical directions when shape also changes. P99 was prespecified as the primary tail level. A 20,000-observation primary cell has 200 order positions in its upper 1%, and an 8000-observation session cell has 80; P99.9 would leave only 20 and 8, respectively. These are finite-sample positions, not independent tail events. The interquartile range describes the central 50% rather than the P99 region, while the coefficient of variation replaces the declared quantile with mean-relative whole-distribution spread. Neither answers the absolute P99-minus-median separation measured by J_{99}; J_{99} itself is not a latency budget. Extreme-value inference requires a declared tail model, threshold diagnostics, and assumptions about the relevant extremal regime [22]. We did not prespecify those elements, so we do not extrapolate beyond the observed quantiles.

For the 100-block control, schedule, and boundary studies, pointwise 95% percentile intervals use 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks. Block resampling follows the dependent-data rationale of Künsch and Lahiri, with circular wrapping as described by Politis and Romano [23–25]. The prespecified length is $L = \lceil 100^{1/3} \rceil = 5$. Each replicate draws circular runs until 100 positions are filled and applies the same block weights $w^{*(r)}$ to every configuration in a study. For endpoint θ, the replicate contrast is $\Delta_\theta^{*(r)} = \theta_{\text{right}}(w^{*(r)}) - \theta_{\text{left}}(w^{*(r)})$. Its variance satisfies

$$\operatorname{Var}(\widehat{\Delta}) = \operatorname{Var}(\widehat{\theta}_{\text{right}}) + \operatorname{Var}(\widehat{\theta}_{\text{left}}) - 2\operatorname{Cov}(\widehat{\theta}_{\text{right}}, \widehat{\theta}_{\text{left}}). \tag{8}$$

Common weights retain block-level covariance induced by shared global block identifiers and, where the design uses them, shared input seeds. In the six-session batch contrast, input seeds are shared among the four conditions within a batch but not between batches 64 and 128; that contrast is block-matched, not input-paired. It estimates a difference between latency distributions generated by separately seeded random batches, not the within-input effect of appending codewords to the same batch. The latter would require a nested-input design. Separately resampling two configurations would remove the retained block-level covariance by construction and could misstate contrast variability. Contrasted visits are nevertheless randomized, nonsimultaneous, and executed in fresh processes, so common weights do not eliminate within-block drift or make the visits simultaneous controls. Circular runs preserve local ordered-block dependence. Resampling individual observations would instead treat the 200 repeated timings within a block as independent.

Sensitivities use $L = 3$, $L = 10$, and paired independent whole-block resampling; the last retains common weights but removes the neighboring-run construction. The value $L = 5$ was prespecified rather than estimated as coverage-optimal. Circular wrapping avoids endpoint truncation, but it proves neither stationarity nor mixing and does not remove nonstationary drift.

Each 40-block replication session is analyzed separately with $L = \lceil 40^{1/3} \rceil = 4$, plus $L = 3$, $L = 8$, and paired whole-block sensitivities. If $\widehat{\Delta}_s$ is a session-specific quantile contrast, the reported estimand is the equal-weight session mean $\overline{\Delta}_6 = 6^{-1} \sum_{s=1}^{6} \widehat{\Delta}_s$, not the contrast between quantiles of pooled observations. Its two-stage bootstrap first resamples sessions and then draws paired circular block runs within each selected session. With only six sessions, session-level resampling is a descriptive stability analysis for this set; its nominal 95% percentiles do not demonstrate calibrated coverage for a population of future sessions. Session ranges and leave-one-session-out values are therefore reported alongside the interval.

The 30 operating-state cycles retain their acquisition order. Their primary intervals use 5000 paired circular moving-block replicates with $L = \lceil 30^{1/3} \rceil = 4$; sensitivities use $L = 3$, $L = 8$, and paired independent whole-cycle resampling. The full-minus-decode treatment response is a difference-in-differences computed with the same cycle weights. The block-placement timing and trace-tail contrasts use paired whole-block resampling; the upper host-latency 1% is selected anew inside each replicate.

Schedule and boundary contrasts use the same four endpoints and resampled block weights; a qualified clock contrast would have used the same method, but no clock acquisition qualified. Contrasts are made only between tested levels. We classify a contrast as an increase when its complete interval is above zero, a decrease when its complete interval is below zero, and uncertain otherwise. Intervals are pointwise descriptions conditional on their acquisition sessions and resampling schemes; they are not multiplicity-adjusted tests, worst-case bounds, equivalence intervals, or evidence about untested levels. Because only discrete levels were tested, the analysis assigns neither hypothesis-test significance nor monotonic behavior over a continuum.

The matched error/convergence analysis uses the codeword/noise pair, rather than an acquisition block or decoded-row count, as its resampling unit and analyzes one of the two 360,000-row acquisitions described above. Each acquisition contains 20,000 unique pairs per E_b/N_0 condition; each pair contributes outcomes from both schedules at all three caps. Within each E_b/N_0 condition, a paired bootstrap draws pair identifiers with replacement and applies the same multiplicity to both schedules and all three caps. Separate point estimates are reported for each schedule's information-bit BER, information-word BLER, final syndrome-success probability, ever-satisfied probability, and conditional first-satisfied iteration. Pointwise 95% percentile intervals are reported for estimable paired flooding-

minus-layered differences. When every matched pair has the same binary outcome under both schedules, the empirical paired bootstrap has zero discordance and cannot estimate population uncertainty; the contrast interval is reported as N/E rather than $[0, 0]$. Treating the generated codeword/noise pairs as independent and identically distributed word-level Bernoulli trials, a zero event count in a 20,000-codeword cell is accompanied by the exact one-sided 95% Clopper–Pearson upper bound. Conditional first-iteration differences and ratios use only matched pairs whose decoded words satisfy the syndrome under both schedules; the ratio is empirical at the tested cap and channel point, not an iteration-equivalence constant.

OpenAI Codex assisted with code review, analysis validation, artifact regeneration, and language editing (GPT-5.6 Sol, OpenAI, San Francisco, CA, USA; accessed 26 July–31 August 2026). The authors directed and verified its use and retain responsibility for all scientific decisions and content.

4. Results

4.1. Schedule Latency and Convergence

At the same batch and fixed iteration cap, the flooding implementation had higher P50 and P99 in all 15 tested cells under the prespecified analysis. The flooding-minus-layered differences were 153.9 µs [150.0, 158.1] for P50 and 170.9 µs [126.9, 201.2] for P99 at batch 16 and five iterations. At batch 512 and 20 iterations, they were 12,812.3 µs [12,777.0, 12,846.7] and 13,102.3 µs [12,288.5, 14,214.7]. These are controlled same-cap comparisons between the tested implementations, not comparisons at equal error performance.

Tail separation did not follow one schedule ordering. For J_{99}, 3 of the 15 flooding-minus-layered intervals indicated an increase, 1 a decrease, and 11 an uncertain direction. For the P99-to-P50 ratio R_{99}, 2 indicated a decrease and 13 were uncertain. Thus, the schedule that took longer at P50 and P99 did not have a consistently wider absolute separation or P99-to-P50 ratio (Figure 2).

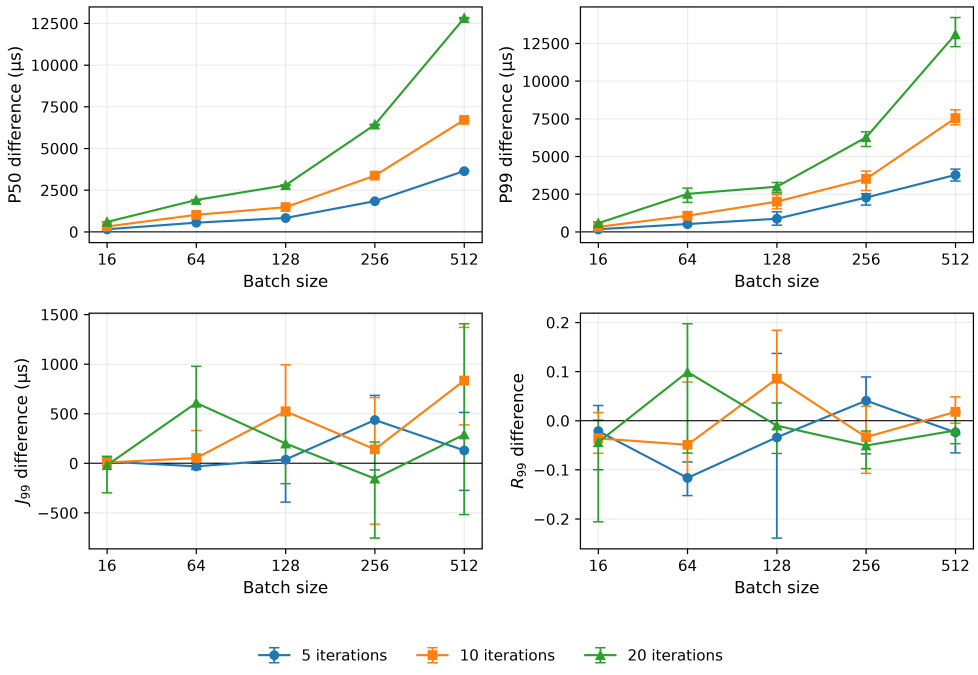


Figure 2. Flooding-minus-layered latency contrasts for BG1 $Z = 384$, FP32 NMS with $\alpha = 0.75$, random valid codewords at 2 dB, and the five tested batches and three fixed iteration caps. Error bars are pointwise descriptive 95% percentile intervals from 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are not multiplicity-adjusted.

Complete schedule endpoint and contrast matrices are reported in Appendix A.

At 0 dB, neither schedule produced a syndrome-satisfying codeword at any cap, so first-satisfaction summaries are marked not applicable (NA). At 2 dB and cap 5, layered decoding reached a valid final syndrome for 23.87% of codewords, whereas flooding reached none; their information-word BLERs were 0.2797 and 1.0000. At cap 10, layered decoding had zero observed erroneous information words and full final-syndrome success. Flooding had two erroneous information words and 93.93% final-syndrome success. At cap 20, both schedules had zero observed information-word errors and full final-syndrome success. Information-word BLER can be lower than syndrome-failure rate because it ignores parity-only errors, whereas the syndrome evaluates the entire decoded mother-code vector. Under the stated word-level Bernoulli model, a zero BLER in 20,000 codewords has a one-sided exact 95% upper bound of 1.50×10^{-4}; it is not evidence of equivalence. Conditional on the 20,000 matched codewords that ever satisfied under both schedules by cap 20, mean first satisfaction was 5.78 iterations for layered and 9.64 for flooding, an empirical ratio of 1.667 [1.666, 1.669].

At 4 dB and cap 5, layered decoding had zero observed information-word errors and full final-syndrome success. Flooding had information-word BLER 0.0002 and final-syndrome success 0.2684. Both schedules had zero observed information-word errors and full final-syndrome success at caps 10 and 20. Conditional on the 20,000 matched codewords that ever satisfied under both schedules by cap 10 or 20, their mean first-satisfaction iterations were 3.13 and 5.74, giving a flooding-to-layered ratio of 1.836 [1.833, 1.839]. These ratios describe only the tested code, channel, cap, and implementation.

The untimed 2-dB outcome sample provides a descriptive error-performance screen for a cross-cap timing overlay. In that sample, layered decoding at cap 10 and flooding decoding at cap 20 both had zero observed information-word errors and full final-syndrome success. In the separate timing acquisition, flooding at cap 20 had higher P50 and P99 than layered at cap 10 at all five batches. Its P50 excess ranged from 1.390 ms at batch 16 to 23.509 ms at batch 512; the corresponding P99 excess ranged from 1.456 to 24.808 ms. The outcome cells were not stratified by the five timing batch sizes, and the timing trials used separately generated words and noise. This overlay therefore does not establish batch-specific equal error performance, an equivalence margin, or an iteration-conversion rule.

Complete schedule-outcome point estimates and estimable paired contrasts are reported in Appendix A.

4.2. Timing Boundaries and Session Replication

Decode-only P50 and P99 were lower than their full-boundary counterparts at every batch. At batch 512, the paired differences were −2924.5 µs [−2952.5, −2901.9] and −3264.5 µs [−3713.8, −2923.3]. Transfer-only was also lower than full at every batch (Figure 3); at batch 512 the differences were −22,001.2 µs [−22,023.9, −21,979.9] and −24,206.3 µs [−24,730.7, −23,533.5]. These paired contrasts compare separately acquired boundaries. They are not terms in an additive decomposition.

At batch 256, the separately acquired P50 values were 12.283 ms for the full boundary, 10.786 ms for decode-only, and 1.280 ms for transfer-only. These values show that the measured distribution depends on the declared software boundary, but they cannot apportion the full-boundary value among decode, transfer, driver, and synchronization causes.

The batch-64-to-128 direction differed across study designs. The baseline scaling and component-timing studies produced J_{99} decreases of 9016.3 and 8863.3 µs, respectively. In the 100-block boundary study, J_{99} instead increased by 505.8 µs [74.6, 792.0] for the full boundary, 788.3 µs [143.8, 1154.1] for decode-only, and 88.9 µs [33.2, 110.6] for transfer-only. The schedule study also produced a positive layered full-boundary estimate of

340.2 µs [6.3, 565.4] under the primary $L = 5$ analysis, although two sensitivity intervals included zero. Because the study groups differ in executable, allocation footprint, input seeds, and block design, this is evidence of design-dependent direction rather than a direct reproduction comparison.

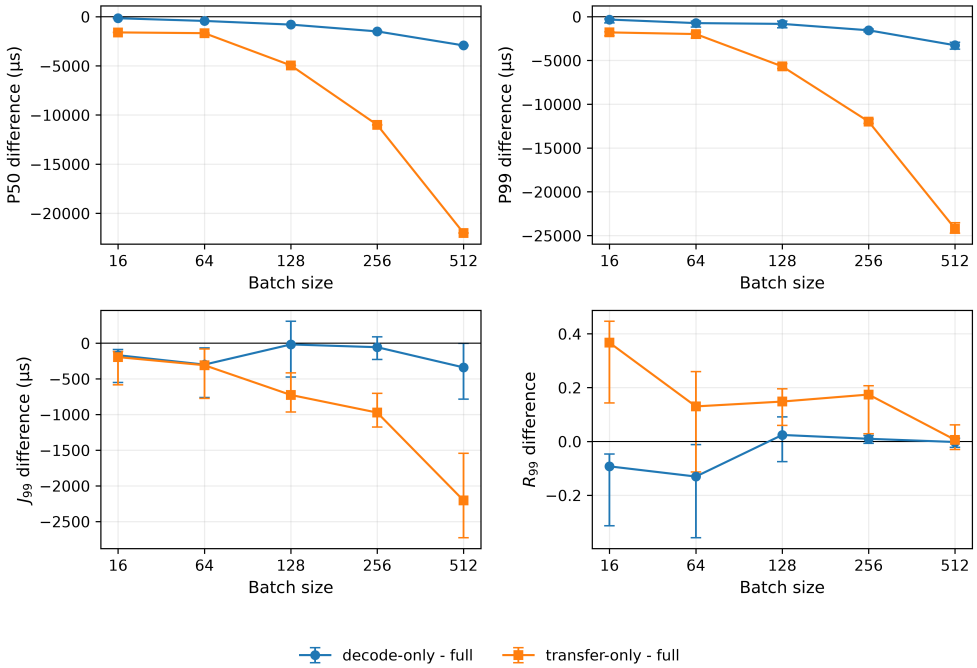


Figure 3. Decode-only-minus-full and transfer-only-minus-full endpoint contrasts for layered BG1 $Z = 384$ decoding at 20 fixed iterations, with random valid codewords at 2 dB. Error bars are pointwise descriptive 95% percentile intervals from 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are not multiplicity-adjusted. Boundaries are compared as separate measurements, not subtracted as a physical component decomposition.

At batch 512, the median raw event-delimited decode interval was 21.809 ms in the full cell and 21.812 ms in decode-only, while transfer-only had a 2.285-ms median raw H2D interval. The corresponding host medians were 24.480, 21.555, and 2.478 ms. These values occupy two timer domains and come from different visits; neither equality nor subtraction identifies a host, transfer, or execution contribution.

An observation-level host-clock calculation replaces any comparison of marginal quantiles. At batch 512, the synchronization-wait/host-total ratio had P50 and P99 values of 0.9973 and 0.9984 for the full boundary, 0.9991 and 0.9994 for decode-only, and 0.9897 and 0.9931 for transfer-only. The P99 of host total minus synchronization wait was 169.4, 72.7, and 74.7 µs, respectively. These paired quantities show that the host spent most of the measured interval blocked while queued work completed; they do not measure an independently separable synchronization overhead. In the finite-sample upper-1% diagnostic, batch-512 full-boundary observations selected by host total had mean raw decode intervals of 24.657 ms, compared with 21.877 ms among the other 99%. Decode-only values were 24.410 and 21.871 ms. The co-occurrence does not distinguish active execution, device scheduling, memory stalls, or timer behavior and has no bootstrap interval.

The synchronized host total was shorter than the raw CUDA-event interval in 447,282 of the 900,000 schedule and boundary observations. In the five batch-512 cells shown in Table A17, the cell frequencies ranged from 42.2% to 98.1%, and mean host-minus-CUDA differences ranged from −478.6 to +18.5 µs. Host and CUDA events therefore did not satisfy the proposed common-clock consistency check. Because no cross-domain calibration was established, raw CUDA-event intervals are used only as event-clock descriptions. Their

absolute scale (Figure 4) is not compared with host elapsed time, and their difference from host time is not interpreted as a component.

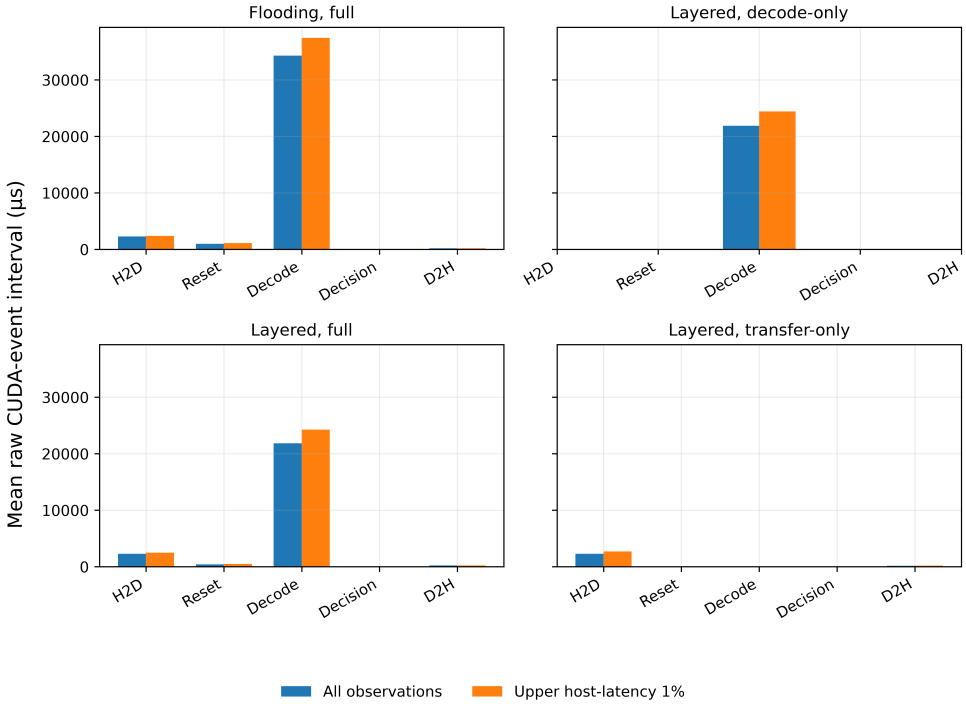


Figure 4. Finite-sample component means for all observations and the 200 batch-512, 20-iteration observations selected into each cell's upper host-total 1%. Operations outside a declared boundary are omitted, not treated as measured zeros. The raw CUDA-event panel was not cross-calibrated to host elapsed time and is not compared in scale with the host panel. This diagnostic has no bootstrap interval and does not identify active execution, synchronization overhead, PCIe contention, or a causal source of tail observations.

Complete boundary endpoint, contrast, event-component, and host-call matrices are reported in Appendix A.

Table A7 reports the prespecified J_{99} batch-64-to-128 contrast in every replication session. All 24 point estimates were positive (Figure 5). Intervals for the layered full boundary, flooding full boundary, and layered decode-only boundary were above zero in all six sessions. The layered transfer-only interval was above zero in sessions 2, 3, and 6 and uncertain in the other three. Across the 240 rows covering batch, schedule, and boundary contrasts for all four endpoints, six classifications changed only between an indicated direction and uncertainty under $L = 3$, $L = 4$, $L = 8$, or paired whole-block resampling; none reversed from an increase to a decrease or vice versa.

The equal-weight two-stage summary in Table 2 preserves session-level variation rather than pooling raw trials. For the layered full boundary, P50 rose by 3660.3 µs and P99 by 3895.0 µs, giving $\Delta J_{99} = +234.7$ µs [196.1, 281.5]. The positive J_{99} contrast coexists with a negative R_{99} contrast because the median increased proportionally more; this is the scale distinction in Equation (7). All four two-stage J_{99} directions remained positive with $L = 3$, $L = 8$, and paired whole-block inner resampling. Because the upper-level sample contains six sessions, the interval summarizes stability within this experiment rather than coverage for future sessions.

The same sessions also provide a repeated schedule comparison at each batch. Flooding-minus-layered J_{99} was +71.4 µs [46.9, 101.0] at batch 64 and +271.4 µs [143.9, 417.8] at batch 128 in the equal-session two-stage analysis. These estimates concern the two tested batches at 20 iterations; they do not define a schedule-wide tail ordering.

Table 2. Equal-weight two-stage batch-64-to-128 contrasts under the schedule-generalized benchmark. Each entry is an estimate with its pointwise 95% percentile interval from 5000 replicates that resample the six acquisition sessions and then paired circular moving-block runs over the 40 ordered blocks in each selected session ($L = 4$). The estimand is the equal-weight mean of six session-specific contrasts, not a contrast between pooled-observation quantiles. With six sessions, the intervals are descriptive stability summaries, not calibrated population-coverage intervals or multiplicity-adjusted tests.

Panel A: Median and 99th Percentile		
Configuration	**ΔP50 (μs)**	**ΔP99 (μs)**
Layered, full	3660.3 [3651.2, 3671.5]	3895.0 [3856.8, 3944.1]
Flooding, full	4434.4 [4423.7, 4447.5]	4869.1 [4758.0, 5001.4]
Layered, decode-only	3291.8 [3281.1, 3301.0]	3471.2 [3437.8, 3493.6]
Layered, transfer-only	308.2 [305.2, 311.1]	332.7 [321.5, 343.3]
Panel B: P99 Separation Measures		
Configuration	**ΔJ_{99} (μs)**	**ΔR_{99}**
Layered, full	234.7 [196.1, 281.5]	−0.02204 [−0.03170, −0.01339]
Flooding, full	434.7 [322.1, 564.4]	0.01578 [0.00124, 0.03137]
Layered, decode-only	179.4 [151.5, 198.1]	−0.01415 [−0.02099, −0.00810]
Layered, transfer-only	24.5 [13.8, 34.5]	−0.07737 [−0.10305, −0.05236]

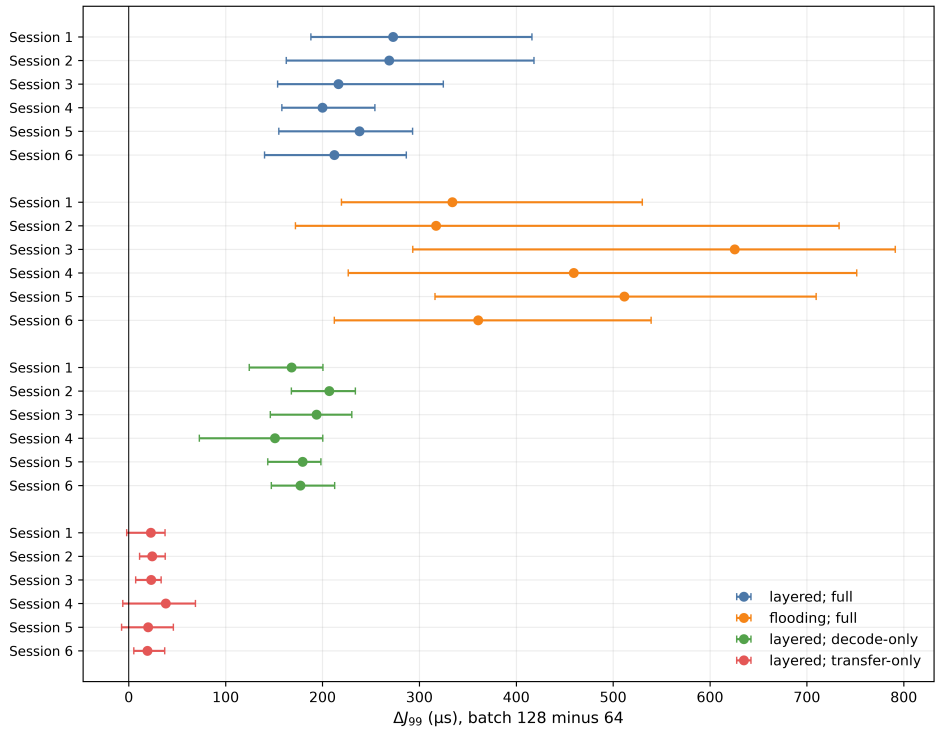


Figure 5. Session-specific estimates of $\Delta J_{99} = J_{99}(128) - J_{99}(64)$ for four block-matched schedule/boundary configurations. Points and horizontal bars show the estimate and pointwise descriptive 95% percentile interval from 5000 paired circular moving-block replicates over the 40 ordered acquisition blocks in each session ($L = 4$). Sessions are discrete experimental replications rather than observations in a time series.

All 24 session-specific point estimates were positive, and no sensitivity analysis reversed a direction. These sessions used the schedule/boundary executable and allocation, unlike the baseline and component-timing designs. Because batches 64 and 128 also used separately seeded inputs, the repeated increase is a contrast between sampled latency distributions rather than the effect of enlarging the same nested batch. The evidence cannot distinguish session variability from effects of the differing executable, allocation, seeds, or block design, and it does not identify a scheduling, memory, clock, or transfer mechanism.

4.3. Operating-State and Algorithmic Sensitivity

Both starting-state conditions were achieved in all 30 analyzed cycles. Idle-cooled visits began at 32–38 °C, and preheated visits began at 58–62 °C. During measurement, mean temperature was 49.4–49.8 °C after idle cooling and 58.0 °C after preheating. Mean SM clocks fell from 2743.8–2744.7 to 2734.9–2735.5 MHz, while mean power rose from 207.2–209.0 to 214.7–215.5 W. The treatment therefore changed temperature, clock, power, and recent workload history together.

Under the prespecified $L = 4$ analysis, the full-boundary P99 and J_{99} contrasts were $+200.7$ µs [7.2, 935.9] and $+186.3$ µs [0.1, 917.6]. The remaining 10 endpoint and difference-in-differences intervals included zero (Figure 6). Whole-cycle resampling made the P99 direction uncertain, and both $L = 3$ and whole-cycle resampling made the J_{99} direction uncertain. The evidence is therefore sensitive to the dependence model and does not identify a temperature-only effect. Appendix A reports the complete endpoint and incomplete-cycle sensitivity results.

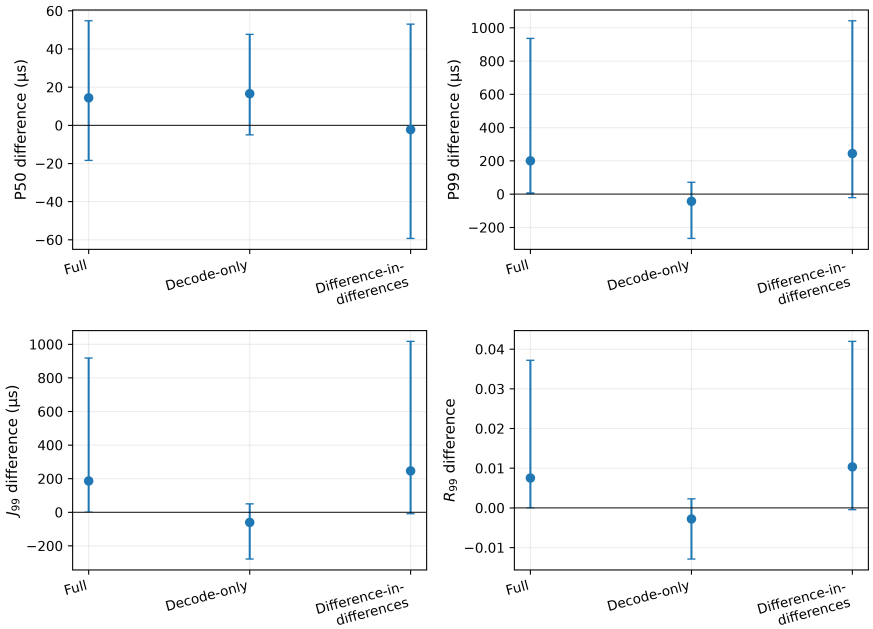


Figure 6. Preheated-minus-idle-cooled contrasts at batch 512. Points and error bars show pointwise descriptive 95% percentile intervals from 5000 paired circular moving-block bootstrap replicates over the 30 ordered acquisition cycles ($L = 4$); they are not multiplicity-adjusted tests or equivalence intervals. The regimen bundles temperature, clock, power, and recent-workload history.

4.3.1. Algorithm, Stopping, Input, and Graph Controls

The algorithmic-control studies provide the stopping, input, channel, graph, and lifting-size comparisons. Across the seven fixed-loop min-sum variants, batch-16 P50 ranged from 1786.4 to 1792.2 µs and P99 from 12,644.9 to 13,265.0 µs. At batch 256, the corresponding ranges were 12,347.7–12,364.0 and 13,047.9–13,192.0 µs. No variant minimized both endpoints at both batches. Decoding quality also differed: at 2 dB, standard min-sum had information-bit BER 0.08128, information-word BLER 0.3230, and syndrome success 0.6764, whereas the tested normalized and offset variants had no observed information-word errors and full syndrome success in their 20,000-word samples. These finite operating points do not establish an optimal normalization.

At 0 dB, every fixed and early observation executed all 20 iterations. For random inputs, early-minus-fixed P50 was $+530.4$ µs [520.9, 539.7] at batch 16 and $+436.7$ µs [402.5, 464.0] at batch 256. P99 moved in opposite directions: -9042.7 µs [-9801.9, -7825.6] and

+347.2 µs [264.6, 450.5]. The batch-16 decrease saved no decoding iterations and incurred syndrome-check work, so it is not latency saved by early termination. The randomized fixed-versus-early-mode contrast combines the altered syndrome-check path with between-visit system variation; this design does not separate those sources. The early path added 20 syndrome sweeps, equal to 2,426,880 lifted-edge visits and 60 barriers per codeword at $Z = 384$. For random inputs at 2 and 4 dB, early stopping reduced executed iterations and both P50 and P99, but R_{99} increased because P50 fell proportionally faster. At batch 256, J_{99} was uncertain at 2 dB and increased by 4883.8 µs [3782.4, 6078.0] at 4 dB.

Of the 48 all-zero-to-random input endpoint contrasts, 45 primary intervals included zero. The three exceptions occurred together in the batch-256, 0-dB early cell. Expressed as random minus all-zero, P99, J_{99}, and R_{99} were −129.3 µs [−213.8, −41.5], −104.7 µs [−176.6, −22.4], and −0.006 [−0.011, −0.001], respectively. Input mode enters the seed derivation, so the two modes are aligned by acquisition block but do not share codewords or noise samples. The contrasts therefore compare separately sampled latency distributions rather than isolate a within-noise payload effect. The 48 pointwise intervals were not multiplicity-adjusted, and channel symmetry constrains error distributions rather than GPU execution time. The three exceptions cannot be attributed to payload values or to noise realization from this design. The data show no uniform input-mode direction and do not establish equivalence. A direct payload-timing test would require all-zero and random codewords constructed with matched sign-transformed noise; the complete 12-cell contrast matrix is reported in Appendix B.

At batch 256, BG1 $Z = 192$ and BG2 $Z = 384$ had P50 values of 5.702 and 7.626 ms, compared with 12.264 ms for BG1 $Z = 384$. These are tested-configuration differences. BG1 and BG2 change topology, rate, transmitted length, and work jointly, so the table does not isolate a causal base-graph effect or establish generality across NR codes.

4.3.2. Operating-State Monitoring

The schedule and boundary studies contained 66,334 and 30,585 valid telemetry samples. Temperatures spanned 34–63 and 39–63 °C, respectively, and the sampled SM and memory clocks varied. Only the informational GPU-idle clock-event bit was observed. Across 107,538 numeric-valid samples from the nine session-replication telemetry sets, temperature ranged from 32 to 64 °C, SM clock from 210 to 2760 MHz, memory clock from 405 to 10,501 MHz, and power from 7.528 to 229.898 W. The clock-event mask was zero in 56,962 samples and GPU-idle only in 50,575; the remaining sample carried the software-power-cap bit that terminated the incomplete operating-state cycle.

For the baseline layered batch-256, 20-iteration acquisition, the block-P99 Spearman associations with temperature and acquisition order were −0.037 and 0.142; memory clock was constant. In the algorithmic-control acquisitions, the corresponding temperature associations were 0.457 at batch 256 and 0.642 at batch 512, while the power associations were 0.470 and 0.717. These acquisition-specific correlations differ in magnitude and were computed from operating variables that covaried. They neither establish nor exclude a thermal or power mechanism. The telemetry is also too coarse to exclude changes between samples. Appendix A Table A22 reports the block-mean and block-P99 associations, and Tables A21 and A23 report operating ranges and clock-event counts.

4.4. Diagnostic Timing Evidence

The block-placement trace was informative only within an SM. Batch 64 assigned one block to each of 64 SMs, whereas batches 128 and above used all 66 observed SMs. The inferred maximum overlap within one SM rose from one block at batch 64 to four blocks at batches 256–512. Within each batch, the unique-SM count and inferred maximum overlap

were constant across trials, so neither quantity explains which observations entered that cell's host-latency tail. Instrumentation changed P50 by +18.4 µs [15.6, 21.2] at batch 64 and by −41.8 to −19.1 µs at batches 128–512; all P99, J_{99}, and R_{99} perturbation intervals included zero. In the instrumented sample, mean within-SM block duration in the upper host-latency 1% exceeded the remaining observations by 1.069 ms [0.689, 1.301] at batch 64 and 1.573 ms [1.342, 1.746] at batch 128. These associations do not measure achieved occupancy or identify a scheduling mechanism.

CUPTI recovered correlated records with no dropped activities in 3600 traced layered trials. At batch 512 under the layered full boundary, P50/P99 durations were 24.951/26.235 ms for the inclusive runtime API, 24.881/26.168 ms for the overlapping synchronization call, 21.894/23.218 ms for summed kernels, and 2.423/2.514 ms for summed memory copies. The records locate activity within the diagnostic execution, but their overlapping intervals and separate host/device origins prevent an additive decomposition. Figures A2 and A3 in Appendix C provide the complete diagnostic views. Fixed-clock control and privileged counters remained unavailable, so driver delay, PCIe contention, memory stalls, and active-SM execution cannot be separated causally.

5. Discussion

5.1. Interpretation of Schedule and Repeated-Session Results

The flooding implementation required more time at P50 and P99 in every same-cap cell. Its kernel makes three lifted-edge traversals per iteration rather than two, maintains old and new message buffers, and uses a different reset path. These implementation differences are consistent with the observed ordering, although the experiment does not estimate their individual contributions. The matched outcome study adds an important qualification: among words that satisfied the syndrome under both schedules by the tested cap, the layered implementation had a lower conditional first-satisfaction iteration in every estimable 2- and 4-dB cell. This comparison is cap-dependent and does not include words censored by failure to satisfy the syndrome. Both schedules used the same prespecified $\alpha = 0.75$; schedule-specific parameter tuning could alter the convergence comparison. The result therefore compares these implementations and settings, not intrinsically optimal versions of the two schedules.

The batch-64-to-128 result was not invariant to experimental design. The baseline scaling and component-timing designs produced large J_{99} contractions, whereas the schedule/boundary design and all six replication sessions produced positive layered full-boundary estimates. The positive session-specific direction persisted under the resampling sensitivities and leave-one-session-out summaries, but six sessions do not support population-level coverage. Moreover, separately seeded batches do not estimate the effect of appending codewords to a fixed input set, and the executable and allocation footprint differ between the two design groups. The evidence therefore establishes design sensitivity, not the reason for it. The theoretical 264-block resource ceiling cannot explain either direction without an achieved-residency measurement.

For layered decoding, endpoint levels differed across the separately acquired full, decode-only, and transfer-only boundaries. The comparison documents boundary-specific distributions, not additive transfer or non-decoder contributions. In instrumented runs, upper host-latency observations were associated with longer within-SM block durations; in separate traced runs, CUPTI located overlapping API, kernel, memory-copy, and synchronization intervals. Neither diagnostic maps additively to the unprofiled tail sample or distinguishes driver scheduling, dynamic clocks, memory behavior, PCIe contention, or active execution. Preheating changed several variables together, and its two narrowly positive full-boundary intervals were not stable across all resampling schemes.

5.2. Practical Meaning of the Metrics

J_{99} expresses the absolute separation between P50 and P99, whereas R_{99} is the P99-to-P50 ratio; the median-relative separation is $R_{99} - 1$. Equation (7) shows why their empirical directions can differ when the median changes. They are two interpretations of the same quantiles rather than independent outcomes. P99 remains the latency endpoint; neither derived metric defines a budget or supports interpolation between the tested batches.

The full-boundary measurement is synchronous decoder-batch completion, not per-codeword or receiver latency. A batch of B codewords is submitted as one decoder-kernel grid with B CUDA blocks, one per codeword; the reported time ends when the entire batch completes. A physical-layer budget would also include CRC processing and hybrid automatic repeat request (HARQ) control:

$$T_{\mathrm{PHY}} = T_{\text{front end}} + T_{\text{demod}} + T_{\text{rate recovery}} + T_{\text{decoder batch}} + T_{\mathrm{CRC}} + T_{\mathrm{HARQ}} + T_{\text{other}}. \quad (9)$$

Percentiles of dependent stages cannot generally be added to obtain the percentile of their sum. The measured P99 values therefore characterize one decoder term in Equation (9), not full-PHY, radio-interface, application, or URLLC latency.

At 2 and 4 dB, early stopping reduced latency when it reduced the iteration count, but R_{99} rose because P50 fell proportionally faster. At 0 dB, it saved no iterations, added 20 syndrome sweeps, and moved P50 and P99 in opposite directions at batch 16. Evaluation therefore depends on channel condition, iteration cap, batch size, and syndrome-check schedule.

5.3. Limitations and Future Work

All measurements use one consumer GPU under dynamically clocked WSL2. Six separately initiated acquisition sessions provide repeated measurements within this installation but do not substitute for operating-system restarts, native Linux, or independent device replication. No intentionally competing GPU workload was launched, but the experiment did not establish exclusive control of host, desktop, driver, or virtualization activity. With only six sessions, the two-stage bootstrap is a descriptive stability analysis rather than a calibrated population inference. Unavailable fixed clocks and privileged counters leave achieved occupancy, active warps, cache and memory traffic, per-trial PCIe use, and virtualization effects unresolved.

The benchmark excludes rate recovery, traffic arrivals, multi-stream overlap, other physical-layer kernels, and application integration. FP32 is a validation reference. Across the graph and schedule studies, only three graph/lifting configurations, one implemented rate per base graph, and three channel conditions were tested. The schedules share one normalization value rather than separately optimized parameters, and the experiment does not factor schedule from every implementation choice or establish equal optimization maturity. The all-zero and random input modes use separate codeword and noise seeds, so that control compares distributions rather than matched-noise payload effects; codeword symmetry does not remove this timing-design limitation. Within each timing visit, 200 repetitions reuse one input batch; they measure execution variability but are not 200 independent channel realizations. The block bootstrap resamples 100 visit blocks per primary cell and 40 per replication session, so its intervals remain conditional on those blocks and the dependence model. P99.9 and maxima are finite-sample descriptions, not worst-case bounds.

The next platform experiments should test native Linux, a second GPU architecture, and fixed clocks with performance-counter access. Counter-enabled profiling should measure achieved occupancy, active warps, cache and memory traffic, and per-trial PCIe activity. Reduced-precision comparisons should evaluate FP16 and INT8 at matched information-bit BER and information-word BLER. Multi-stream studies should report transfer–computation overlap, and fuller-receiver studies should include rate recovery. Matched CPU and FPGA comparisons require the same code configuration, precision, stopping rule, error target, and timing boundary.

6. Conclusions

GPU LDPC latency comparisons depend on decoder schedule, batch granularity, convergence behavior, and timing boundary. We used a boundary-explicit empirical framework that combined P50, P99, J_{99}, R_{99}, paired-block resampling, matched decoding outcomes, three timing boundaries, and session replication. Flooding had higher P50 and P99 in 15 same-cap cells, although J_{99} and R_{99} had no uniform ordering. At batch 512 and 20 iterations, flooding exceeded layered decoding by 12.812 ms at P50 and 13.102 ms at P99. At 2 dB and cap 20, the mean first-satisfaction iteration among the 20,000 matched words satisfying the syndrome under both schedules was 5.78 for layered decoding and 9.64 for flooding. At batch 512, layered decode-only P50 and P99 were 2.925 and 3.265 ms below the separately acquired full-boundary endpoints; these differences are not additive component estimates. Across six sessions, full-boundary J_{99} increased by 234.7 µs from batch 64 to 128 (pointwise descriptive 95% interval, 196.1–281.5 µs), whereas the baseline scaling and component-timing designs produced decreases. Together, these findings show that schedule, decoding-outcome context, timing boundary, and acquisition design must be reported together. They apply to the tested FP32 implementations with $\alpha = 0.75$ on one dynamically clocked GPU under WSL2 and do not establish microarchitectural causation, worst-case or cross-platform behavior, or URLLC compliance.

Author Contributions: Conceptualization, S.J. and E.K.; methodology, S.J.; software, S.J.; validation, S.J. and E.K.; formal analysis, S.J.; investigation, S.J.; resources, S.J.; data curation, S.J.; writing—original draft preparation, S.J.; writing—review and editing, S.J. and E.K.; visualization, S.J.; supervision, E.K.; project administration, E.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code and baseline timing dataset are publicly available at https://github.com/pz1004/5G-NR-BG1-LDPC-Decoding (accessed on 10 September 2026). The additional raw datasets supporting the algorithmic-control, schedule, timing-boundary, operating-state, and diagnostic analyses are available from the corresponding author upon reasonable request.

Acknowledgments: The authors acknowledge the use of OpenAI Codex as described in Materials and Methods. The authors retain full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Detailed Comparison Matrices

The following tables provide the complete schedule, outcome, boundary, component, and operating-state matrices summarized in the main text. They are derived from the experimental observations described in Methods and Appendix C.

Appendix A.1. Literature Comparison Matrices

Table A1. Configuration and algorithm fields reported for cited GPU LDPC operating points. Length and rate follow each source's convention. N/R means not reported; no missing value is inferred.

Study	GPU	Graph and Z	Reported n,k and Rate	Algorithm
Aronov et al. [5]	Model N/R	5G NR; base graph and Z N/R	N/R	Standards-compliant decoder; detail N/R
Li et al. [6]	RTX 2080 Ti	BG1-derived; $Z = 80$	$n = 2080$, $k = 1760$, rate 5/6	Scaled parallel layered min-sum
Ling and Cautereels [7]	GTX 1660 (TU116)	BG1 and BG2; up to $Z = 384$	BG1: $n = 26{,}112$, k N/R; rates 1/3 (latency) and 1/2 (throughput)	Layered min-sum, integer and floating-point (FP) variants
Tarver et al. [8]	980 Ti, 1080 Ti, Titan RTX	BG1; $Z = 384$ selected point	$n = 25{,}344$, $k = 8448$, rate 1/3	Flooding scaled min-sum
Dai et al. [9]	RTX 2080 Ti; Xavier NX separately	BG1; $Z = 384$ matched point	$n = 25{,}344$, $k = 8448$, rate 1/3	Flooding offset min-sum, $\beta = 1$
Liu and Zhao [10]	RTX 3070, RTX 4090	BG1; $Z = 384$	$n = 26{,}112$, $k = 8448$, rate 1/3	Memory-reduced layered min-sum
DecodeX: Aerial [11]	H200, RTX 3090, RTX 6000 Ada	BG1 and BG2; varies by modulation and coding scheme (MCS); Z N/R	Length N/R; rate varies by MCS	Implementation-specific iterative belief propagation
DecodeX: Sionna-RK [11]	NVIDIA Jetson Orin AGX	Base graph and Z N/R for the reported points	n N/R; $k = 1936, 4224, 8440$; rates 0.33 and 0.88 reported	Implementation-specific iterative belief propagation
This work	RTX 4070 Ti SUPER	BG1 $Z = 192, 384$; BG2 $Z = 384$	BG1: rate 1/3, $k = 4224$ or 8448; BG2: rate 1/5, $k = 3840$; transmitted and internal lengths stated in Methods	Layered and flooding NMS; layered standard and offset min-sum controls

Table A2. Stopping, precision, and software fields reported for the same GPU LDPC operating points. N/R means not reported; no missing value is inferred.

Study	Stopping and Iterations	Precision	CUDA, OS, and Software
Aronov et al. [5]	N/R	N/R	N/R
Li et al. [6]	Fixed, 10	4-bit input LLR; 8-bit messages	N/R
Ling and Cautereels [7]	Valid-codeword or maximum stopping; maximum N/R	8-bit integer; FP16	CUDA intrinsics; CUDA and OS versions N/R
Tarver et al. [8]	Fixed, 5	INT8, FP16, FP32	CUDA C++ library and OpenAirInterface integration; versions N/R
Dai et al. [9]	Fixed cap 10 at tabulated point; early termination elsewhere	8-bit	CUDA 11.3, GCC 11.1, GNU/Linux 5.12.15
Liu and Zhao [10]	Fixed 5 at tabulated point; early cap 5 separately	INT8	CUDA 11.4, Windows 10
DecodeX: Aerial [11]	Stopping N/R; workload-dependent iteration count	N/R	Aerial v25.2-cuBB, Ubuntu 24.04, kernel 6.8
DecodeX: Sionna-RK [11]	Stopping N/R; iterations 2–16 at reported points	N/R	Sionna-RK 1.2.1, Ubuntu 22.04, kernel 5.15
This work	Fixed caps 5, 10, and 20; syndrome stopping at cap 20 in the sensitivity study	FP32	CUDA 12.0, driver 610.62, Ubuntu 24.04.4 under WSL2, kernel 6.18.33.1

Table A3. Reported execution granularity, performance, software, and timing boundaries for the same operating points. Latency and throughput values may come from different points within a source and are not ranked against this work. DTP: data-transfer process.

Study	Batch or Granularity	Latency	Throughput	Software Environment	Timed Boundary
Aronov et al. [5]	N/R	GPU latency qualitatively above FPGA	N/R	N/R	Decoder comparison; exact boundary N/R
Li et al. [6]	800 codewords; two codewords per CUDA thread block at the reported optimum	N/R	1.385 Gbit/s	Environment N/R	Exact transfer treatment N/R
Ling and Cautereels [7]	Up to 24 integer or 12 FP instances	BG1 35 µs (integer) and 40 µs (FP) per iteration; BG2 27 and 30 µs, respectively	1.8 Gcoded-bit/s (integer) and 0.8 Gcoded-bit/s (FP)	CUDA intrinsics; CUDA and OS versions N/R	Per-iteration decoder latency; transfer times excluded (inline boundary)
Tarver et al. [8]	Throughput point: 240 codewords across six streams; latency-point batch N/R	Best latency point 87 µs; throughput point about 700 µs	3.964 Gbit/s at throughput point; 290 Mbit/s at latency point	CUDA C++ library and OpenAirInterface integration; versions N/R	H2D and D2H included

Table A4. Reported execution granularity, performance, software, and timing boundaries (continued). Values may come from different operating points within a source and are not ranked against this work. DTP: data-transfer process.

Study	Batch or Granularity	Latency	Throughput	Software Environment	Timed Boundary
Dai et al. [9]	512 codewords at matched point	1.969 ms including transfer; 1.190 ms excluding transfer	6.590 Gbit/s	CUDA 11.3, GCC 11.1, GNU/Linux 5.12.15	Transfer-included and decoder-only values separated
Liu and Zhao [10]	RTX 4090 long-code point: 512 codewords	0.484 ms excluding DTP; 1.667 ms including one DTP stream	27.63 Gbit/s excluding DTP; 8.02 Gbit/s including it	CUDA 11.4, Windows 10	DTP included and excluded values separated
DecodeX: Aerial [11]	Parallel transport blocks; 1–50 user equipments (UEs)	Workload-dependent	No single reported Gbit/s value	Aerial v25.2-cuBB, Ubuntu 24.04, kernel 6.8	Kernel-only and total orchestration/transfer boundaries separated
DecodeX: Sionna-RK [11]	N/R	LDPC processing latency; value depends on the reported point	N/R	Sionna-RK 1.2.1, Ubuntu 22.04, kernel 5.15	Exact host/kernel and transfer sub-boundaries N/R
This work	Scaling: 4, 16, 64, 128, 192, 256, 320, 512 codewords; schedule and boundary studies: 16, 64, 128, 256, 512	Matched-schedule P50 0.534–37.291 ms; P99 0.643–39.548 ms	N/R	CUDA 12.0 (nvcc 12.0.140), driver 610.62, Ubuntu 24.04.4 under WSL2, kernel 6.18.33.1	Layered full, decode-only, and equal-volume transfer-only host boundaries; flooding full boundary; host API and blocking-wait durations; separate raw CUDA-event intervals

Appendix A.2. Experimental and Statistical Matrices

Table A5. Zero-observed-error cross-cap timing contrasts: flooding at cap 20 minus layered at cap 10, at 2 dB. Pointwise 95% percentile intervals use 5000 paired circular moving-block replicates over acquisition blocks ($L = 5$). The zero observed bit and information-word errors and full final-syndrome success come from separate, untimed, non-batch-stratified 20,000-word outcome cells; the latency contrasts come from the listed timing-batch cells with separately generated inputs. This overlay is not a joint equal-performance or equivalence analysis.

Panel A: Median and P99 Contrasts		
Batch	**P50 [95% Interval, µs]**	**P99 [95% Interval, µs]**
16	1389.9 [1376.8, 1404.0]	1456.3 [1402.3, 1516.2]
64	2714.2 [2697.8, 2726.7]	3359.9 [2822.2, 3731.0]
128	5178.1 [5155.9, 5200.3]	5685.0 [5328.3, 5806.8]
256	11,774.9 [11,755.4, 11,797.2]	12,135.8 [11,832.2, 12,580.2]
512	23,508.7 [23,473.2, 23,546.1]	24,807.5 [24,210.4, 25,810.8]
Panel B: P99-Separation Contrasts		
Batch	**J_{99} [95% Interval, µs]**	**R_{99} [95% Interval]**
16	66.4 [14.9, 121.9]	−0.071 [−0.105, −0.045]
64	645.6 [110.0, 1016.7]	0.054 [−0.078, 0.156]
128	506.9 [134.9, 641.0]	0.010 [−0.033, 0.026]
256	360.9 [56.2, 812.5]	−0.071 [−0.107, −0.007]
512	1298.8 [695.1, 2308.1]	−0.009 [−0.026, 0.020]

Table A6. Acquisition-specific batch-64-to-128 J_{99} evidence. Pointwise 95% percentile intervals use 5000 paired circular moving-block replicates over acquisition blocks ($L = 5$). Acquisitions are displayed separately and are not pooled or meta-analyzed.

Panel A: Endpoint Estimates and Primary Contrast			
Acquisition and Configuration	**$J_{99,64}$**	**$J_{99,128}$**	**ΔJ_{99} [95% Interval]**
Baseline scaling, layered/full	9880.5	864.2	−9016.3 [−9851.8, −7258.7]
Component timing, layered/full	10,567.0	1703.7	−8863.3 [−10,653.5, −5721.6]
Schedule study, layered/full	231.5	571.7	340.2 [6.3, 565.4]
Schedule study, flooding/full	840.2	770.9	−69.3 [−554.4, 465.4]
Boundary study, layered/full	431.0	936.8	505.8 [74.6, 792.0]
Boundary study, layered/decode−only	130.4	918.7	788.3 [143.8, 1154.1]
Boundary study, layered/transfer−only	121.9	210.8	88.9 [33.2, 110.6]

Panel B: Interval Direction under Resampling Sensitivities				
Acquisition and Configuration	**$L = 5$**	**$L = 3$**	**$L = 10$**	**Whole Block**
Baseline scaling, layered/full	decrease	decrease	decrease	decrease
Component timing, layered/full	decrease	decrease	decrease	decrease
Schedule study, layered/full	increase	uncertain	increase	uncertain
Schedule study, flooding/full	uncertain	uncertain	uncertain	uncertain
Boundary study, layered/full	increase	increase	increase	increase
Boundary study, layered/decode−only	increase	increase	increase	increase
Boundary study, layered/transfer−only	increase	increase	increase	increase

Table A7. Session-specific batch-64-to-128 J_{99} differences in µs. Entries are estimates with pointwise 95% percentile intervals from 5000 paired circular moving-block bootstrap replicates over the 40 ordered blocks in each session ($L = 4$).

Panel A: Full-Boundary Contrasts		
Session	**Layered, Full Boundary**	**Flooding, Full Boundary**
1	272.9 [188.1, 416.2]	334.1 [219.6, 530.1]
2	268.8 [162.7, 418.2]	317.2 [172.0, 733.2]
3	216.4 [153.6, 324.8]	625.4 [293.2, 791.2]
4	200.0 [158.1, 254.2]	459.4 [226.5, 751.5]
5	238.1 [154.9, 292.9]	511.6 [316.1, 709.6]
6	212.2 [140.1, 286.4]	360.7 [212.3, 539.2]

Table A7. *Cont.*

Panel B: Decode-Only and Transfer-Only Contrasts		
Session	**Layered, Decode-Only Boundary**	**Layered, Transfer-Only Boundary**
1	168.1 [124.5, 200.5]	22.7 [−2.1, 37.4]
2	207.0 [167.9, 233.9]	24.2 [11.1, 37.6]
3	193.8 [146.2, 230.2]	23.1 [7.2, 33.4]
4	150.8 [72.8, 200.2]	38.1 [−6.2, 68.9]
5	179.5 [143.5, 198.3]	19.9 [−7.3, 46.1]
6	177.2 [147.2, 212.5]	19.2 [5.2, 37.0]

Table A8. Preheated-minus-idle-cooled endpoint contrasts and the full-minus-decode difference-in-differences at batch 512. Entries are estimates with pointwise 95% percentile intervals from 5000 paired circular moving-block bootstrap replicates over 30 ordered cycles ($L = 4$).

Panel A: Median and 99th Percentile		
Contrast	**ΔP50 (µs)**	**ΔP99 (µs)**
Full boundary	14.4 [−18.5, 54.8]	200.7 [7.2, 935.9]
Decode-only boundary	16.6 [−5.0, 47.6]	−43.2 [−266.7, 70.7]
Difference-in-differences	−2.2 [−59.3, 53.0]	243.9 [−21.8, 1041.8]
Panel B: P99 Separation Measures		
Contrast	**ΔJ_{99} (µs)**	**ΔR_{99}**
Full boundary	186.3 [0.1, 917.6]	0.00754 [−0.00002, 0.03717]
Decode-only boundary	−59.8 [−278.6, 49.8]	−0.00280 [−0.01289, 0.00226]
Difference-in-differences	246.1 [−8.9, 1017.4]	0.01034 [−0.00046, 0.04196]

Table A9. Schedule latency endpoints. Pointwise 95% percentile intervals use 5000 paired circular moving-block replicates with the acquisition block as the resampling unit and ordered circular runs of length $L = 5$; they are descriptive and not multiplicity-adjusted.

Full timing boundary; dynamic clock mode.

Panel A: Flooding Schedule—Median and 99th Percentile			
Batch Size	**Iteration Cap**	**Median (P50) Estimate [95% Interval], µs**	**99th Percentile (P99) Estimate [95% Interval], µs**
16	5	688.0 [685.8, 691.8]	813.6 [798.0, 829.7]
16	10	1256.1 [1250.4, 1264.3]	1419.7 [1401.1, 1478.7]
16	20	2325.9 [2316.4, 2338.0]	2547.9 [2502.6, 2603.4]
64	5	1368.8 [1355.4, 1383.1]	1529.1 [1514.0, 1539.9]
64	10	2237.4 [2226.2, 2246.6]	2484.4 [2419.6, 2760.8]
64	20	3934.4 [3925.6, 3944.3]	4774.6 [4228.1, 5140.6]
128	5	2855.3 [2846.4, 2864.7]	3219.6 [3100.5, 3640.5]
128	10	4715.5 [4706.9, 4723.8]	5503.6 [5001.9, 5981.1]
128	20	8414.9 [8399.5, 8430.8]	9185.8 [8812.0, 9325.5]
256	5	6124.4 [6115.2, 6133.7]	6993.7 [6625.0, 7146.0]
256	10	10,319.2 [10,306.4, 10,331.9]	11,461.9 [10,668.7, 11,856.8]
256	20	18,726.9 [18,705.7, 18,751.2]	20,089.9 [19,814.4, 20,293.0]
512	5	12,055.0 [12,037.1, 12,072.8]	13,145.3 [12,911.4, 13,371.4]
512	10	20,495.4 [20,475.3, 20,517.5]	22,288.4 [21,835.6, 22,730.3]
512	20	37,290.9 [37,257.1, 37,328.3]	39,548.1 [38,873.6, 40,545.0]
Panel B: Flooding Schedule—P99 Separation Measures			
Batch Size	**Iteration Cap**	**P99–Median Gap (J_{99}) Estimate [95% Interval], µs**	**P99-to-P50 Ratio (R_{99}) Estimate [95% Interval]**
16	5	125.6 [110.3, 140.8]	1.183 [1.160, 1.205]
16	10	163.6 [144.4, 223.4]	1.130 [1.115, 1.178]
16	20	221.9 [177.0, 276.3]	1.095 [1.076, 1.119]
64	5	160.3 [144.0, 175.7]	1.117 [1.104, 1.129]
64	10	247.0 [188.2, 522.0]	1.110 [1.084, 1.233]
64	20	840.2 [295.6, 1205.9]	1.214 [1.075, 1.307]
128	5	364.3 [248.2, 785.8]	1.128 [1.087, 1.275]
128	10	788.1 [286.8, 1264.8]	1.167 [1.061, 1.268]
128	20	770.9 [389.0, 917.9]	1.092 [1.046, 1.109]
256	5	869.3 [503.8, 1021.7]	1.142 [1.082, 1.167]
256	10	1142.8 [352.4, 1539.3]	1.111 [1.034, 1.149]
256	20	1363.0 [1092.3, 1567.3]	1.073 [1.058, 1.084]
512	5	1090.2 [856.8, 1311.2]	1.090 [1.071, 1.109]
512	10	1793.0 [1346.1, 2235.1]	1.087 [1.066, 1.109]
512	20	2257.2 [1579.9, 3259.7]	1.061 [1.042, 1.087]

Table A9. *Cont.*

Full timing boundary; dynamic clock mode.			
Panel C: Layered Schedule—Median and 99th Percentile			
Batch Size	**Iteration Cap**	**Median (P50) Estimate [95% Interval], µs**	**99th Percentile (P99) Estimate [95% Interval], µs**
16	5	534.1 [531.5, 537.3]	642.7 [617.4, 684.4]
16	10	936.0 [932.2, 942.2]	1091.5 [1076.8, 1120.5]
16	20	1734.0 [1725.8, 1749.6]	1975.7 [1918.2, 2269.0]
64	5	819.9 [811.0, 835.1]	1011.4 [991.9, 1035.6]
64	10	1220.2 [1208.6, 1237.9]	1414.7 [1386.2, 1498.5]
64	20	2024.4 [2008.0, 2039.8]	2255.9 [2201.4, 2494.1]
128	5	2024.6 [2013.9, 2034.4]	2351.4 [2262.1, 2762.0]
128	10	3236.8 [3228.6, 3245.8]	3500.8 [3461.1, 3531.8]
128	20	5621.5 [5610.1, 5633.2]	6193.2 [5928.3, 6411.8]
256	5	4285.3 [4276.6, 4293.2]	4718.7 [4574.6, 5143.6]
256	10	6952.0 [6942.2, 6963.3]	7954.1 [7492.5, 8205.2]
256	20	12,303.7 [12,289.9, 12,318.1]	13,821.7 [13,430.4, 14,384.8]
512	5	8410.9 [8400.0, 8422.8]	9371.5 [9020.5, 9645.5]
512	10	13,782.2 [13,767.6, 13,798.6]	14,740.6 [14,428.0, 14,884.0]
512	20	24,478.5 [24,456.6, 24,500.5]	26,445.8 [25,738.6, 27,001.1]
Panel D: Layered Schedule—P99 Separation Measures			
Batch Size	**Iteration Cap**	**P99–Median Gap (J_{99}) Estimate [95% Interval], µs**	**P99-to-P50 Ratio (R_{99}) Estimate [95% Interval]**
16	5	108.7 [83.4, 149.6]	1.203 [1.156, 1.280]
16	10	155.5 [142.4, 181.6]	1.166 [1.152, 1.194]
16	20	241.7 [180.7, 531.7]	1.139 [1.104, 1.307]
64	5	191.5 [170.6, 217.1]	1.234 [1.206, 1.265]
64	10	194.6 [164.1, 272.9]	1.159 [1.134, 1.223]
64	20	231.5 [176.9, 472.8]	1.114 [1.087, 1.232]
128	5	326.8 [239.3, 733.6]	1.161 [1.118, 1.362]
128	10	264.0 [228.1, 291.2]	1.082 [1.070, 1.090]
128	20	571.7 [314.6, 788.6]	1.102 [1.056, 1.140]
256	5	433.4 [290.6, 859.2]	1.101 [1.068, 1.200]
256	10	1002.1 [535.6, 1252.5]	1.144 [1.077, 1.180]
256	20	1518.0 [1127.1, 2080.7]	1.123 [1.092, 1.169]
512	5	960.6 [611.7, 1237.1]	1.114 [1.073, 1.147]
512	10	958.4 [646.7, 1096.2]	1.070 [1.047, 1.080]
512	20	1967.2 [1259.0, 2518.1]	1.080 [1.051, 1.103]

Table A10. Schedule paired endpoint contrasts (right minus left). Pointwise 95% percentile intervals use 5000 paired circular moving-block replicates with the acquisition block as the resampling unit and ordered circular runs of length $L = 5$; they are descriptive and not multiplicity-adjusted.

Flooding minus layered; full timing boundary; dynamic clock mode.
Positive values denote a larger flooding endpoint.

Panel A: Median and 99th Percentile

Batch Size	**Iteration Cap**	**Median (P50) Estimate [95% Interval], µs**	**99th Percentile (P99) Estimate [95% Interval], µs**
16	5	153.9 [150.0, 158.1]	170.9 [126.9, 201.2]
16	10	320.0 [310.8, 329.9]	328.1 [294.0, 392.6]
16	20	591.9 [578.6, 603.1]	572.2 [292.8, 655.1]
64	5	548.9 [525.8, 566.5]	517.7 [487.5, 542.0]
64	10	1017.3 [999.1, 1030.7]	1069.7 [973.3, 1346.4]
64	20	1910.1 [1893.7, 1927.5]	2518.7 [1948.2, 2896.1]
128	5	830.6 [820.0, 843.1]	868.2 [439.3, 1335.1]
128	10	1478.7 [1464.9, 1491.5]	2002.8 [1515.3, 2471.2]
128	20	2793.4 [2770.0, 2816.9]	2992.6 [2587.2, 3277.3]
256	5	1839.1 [1827.2, 1851.4]	2275.0 [1767.9, 2526.1]
256	10	3367.2 [3353.8, 3379.6]	3507.9 [2749.2, 4030.4]
256	20	6423.2 [6398.6, 6449.6]	6268.2 [5666.2, 6636.6]
512	5	3644.1 [3629.9, 3659.1]	3773.8 [3367.7, 4160.3]
512	10	6713.2 [6689.2, 6738.1]	7547.8 [7097.8, 8094.2]
512	20	12,812.3 [12,777.0, 12,846.7]	13,102.3 [12,288.5, 14,214.7]

Table A10. *Cont.*

Panel B: P99 Separation Measures			
Batch Size	**Iteration Cap**	**P99−Median Gap (J_{99}) Estimate [95% Interval], µs**	**P99-to-P50 Ratio (R_{99}) Estimate [95% Interval]**
16	5	16.9 [−26.7, 46.4]	−0.021 [−0.100, 0.031]
16	10	8.1 [−23.6, 71.9]	−0.036 [−0.066, 0.017]
16	20	−19.7 [−299.2, 65.7]	−0.044 [−0.206, 0.001]
64	5	−31.2 [−63.4, −1.8]	−0.116 [−0.152, −0.084]
64	10	52.5 [−42.7, 330.3]	−0.049 [−0.115, 0.079]
64	20	608.7 [41.5, 978.0]	0.099 [−0.066, 0.197]
128	5	37.6 [−391.2, 502.7]	−0.034 [−0.239, 0.137]
128	10	524.0 [34.3, 992.0]	0.086 [−0.018, 0.184]
128	20	199.2 [−206.4, 480.5]	−0.010 [−0.067, 0.036]
256	5	436.0 [−67.6, 683.7]	0.041 [−0.068, 0.089]
256	10	140.7 [−614.1, 662.3]	−0.033 [−0.107, 0.030]
256	20	−155.0 [−754.5, 213.1]	−0.051 [−0.098, −0.021]
512	5	129.7 [−272.7, 513.4]	−0.024 [−0.066, 0.018]
512	10	834.6 [386.2, 1370.8]	0.018 [−0.005, 0.049]
512	20	290.0 [−518.1, 1406.0]	−0.020 [−0.047, 0.014]

Table A11. Paired random-valid-codeword decoding outcomes for the layered and flooding implementations. Each 20,000-word cell was decoded in chunks of at most 256 words (78 full chunks and one 32-word chunk). "Final syndrome" tests the decoder output after the stated fixed cap; "ever satisfied" records whether the syndrome was satisfied at least once by that cap. Because fixed decoding continues after a first success, these indicators can differ. Under an independent and identically distributed word-level Bernoulli model, parenthetical limits are one-sided 95% exact Clopper–Pearson bounds for a zero event rate or, when the observed event rate is one, for the zero complementary-failure rate. Zero BER denotes no observed information-bit errors; no independent-bit confidence bound is claimed. First-satisfied-iteration summaries are conditional on ever-satisfied words. These word-level results are not organized as acquisition blocks and do not use the acquisition-block bootstrap.

Panel A: Error and Syndrome Outcomes						
Schedule	**E_b/N_0**	**Cap**	**BER**	**BLER**	**Final Syndrome**	**Ever Satisfied**
flooding	0	5	2.251×10^{-1}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
layered	0	5	2.406×10^{-1}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
flooding	0	10	2.398×10^{-1}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
layered	0	10	2.420×10^{-1}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
flooding	0	20	2.420×10^{-1}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
layered	0	20	2.420×10^{-1}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
flooding	2	5	1.657×10^{-2}	1 (non-error $< 1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)	0 ($<1.50 \times 10^{-4}$)
layered	2	5	6.489×10^{-5}	0.2797	0.2387	0.2387
flooding	2	10	1.184×10^{-8}	0.0001	0.9393	0.9395
layered	2	10	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
flooding	2	20	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
layered	2	20	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
flooding	4	5	2.367×10^{-8}	0.0002	0.2684	0.2684
layered	4	5	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
flooding	4	10	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
layered	4	10	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
flooding	4	20	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)
layered	4	20	0	0 ($<1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)	1 (failure $< 1.50 \times 10^{-4}$)

Table A11. *Cont.*

Panel B: Conditional First-Satisfied-Iteration Summaries					
Schedule	E_b/N_0	**Cap**	**Conditioning Words**	**Mean First Iteration**	**Median First Iteration**
flooding	0	5	0	NA	NA
layered	0	5	0	NA	NA
flooding	0	10	0	NA	NA
layered	0	10	0	NA	NA
flooding	0	20	0	NA	NA
layered	0	20	0	NA	NA
flooding	2	5	0	NA	NA
layered	2	5	4773	5.00	5
flooding	2	10	18,789	9.55	10
layered	2	10	20,000	5.78	6
flooding	2	20	20,000	9.64	10
layered	2	20	20,000	5.78	6
flooding	4	5	5368	5.00	5
layered	4	5	20,000	3.13	3
flooding	4	10	20,000	5.74	6
layered	4	10	20,000	3.13	3
flooding	4	20	20,000	5.74	6
layered	4	20	20,000	3.13	3

Table A12. Paired schedule outcome contrasts (flooding minus layered). Pointwise 95% percentile intervals use 5000 paired ordinary bootstrap replicates of matched words; they are descriptive and not multiplicity-adjusted. A binary contrast with no discordant paired-word outcomes is marked N/E because an empirical bootstrap point mass at zero does not quantify population uncertainty. First-iteration differences and the empirical flooding/layered convergence ratio are conditional on ever satisfying the syndrome under both schedules. The ratio is specific to each tested SNR/cap cell and is not an iteration-equivalence factor. This word-level bootstrap is separate from the ordered-block latency bootstrap.

Panel A: Information-Bit and Information-Word Outcomes			
E_b/N_0	**Cap**	**ΔBER [95% Interval]**	**ΔBLER [95% Interval]**
0	5	-1.547×10^{-2} [-1.559×10^{-2}, -1.536×10^{-2}]	N/E (0 discordant pairs)
0	10	-2.190×10^{-3} [-2.297×10^{-3}, -2.081×10^{-3}]	N/E (0 discordant pairs)
0	20	5.839×10^{-5} [-4.840×10^{-5}, 1.640×10^{-4}]	N/E (0 discordant pairs)
2	5	1.651×10^{-2} [1.644×10^{-2}, 1.658×10^{-2}]	0.7202 [0.7140, 0.7268]
2	10	1.184×10^{-8} [0, 2.959×10^{-8}]	0.0001 [0.0000, 0.0003]
2	20	N/E (0 discordant pairs)	N/E (0 discordant pairs)
4	5	2.367×10^{-8} [5.919×10^{-9}, 4.735×10^{-8}]	0.0002 [0.0001, 0.0004]
4	10	N/E (0 discordant pairs)	N/E (0 discordant pairs)
4	20	N/E (0 discordant pairs)	N/E (0 discordant pairs)
Panel B: Syndrome Outcomes			
E_b/N_0	**Cap**	**Δ Final-Syndrome Success [95% Interval]**	**Δ Ever-Satisfied Success [95% Interval]**
0	5	N/E (0 discordant pairs)	N/E (0 discordant pairs)
0	10	N/E (0 discordant pairs)	N/E (0 discordant pairs)
0	20	N/E (0 discordant pairs)	N/E (0 discordant pairs)
2	5	−0.2387 [−0.2447, −0.2327]	−0.2387 [−0.2447, −0.2327]
2	10	−0.0607 [−0.0640, −0.0575]	−0.0605 [−0.0638, −0.0573]
2	20	N/E (0 discordant pairs)	N/E (0 discordant pairs)
4	5	−0.7316 [−0.7379, −0.7254]	−0.7316 [−0.7379, −0.7254]
4	10	N/E (0 discordant pairs)	N/E (0 discordant pairs)
4	20	N/E (0 discordant pairs)	N/E (0 discordant pairs)

Table A12. *Cont.*

Panel C: Iteration Differences Conditional on Joint Success			
E_b/N_0	**Cap**	**Both Ever Satisfied**	**Δ Mean First Iteration [95% Interval]**
0	5	0	NA
0	10	0	NA
0	20	0	NA
2	5	0	NA
2	10	18,789	3.79 [3.78, 3.80]
2	20	20,000	3.86 [3.85, 3.87]
4	5	5368	1.97 [1.96, 1.97]
4	10	20,000	2.61 [2.61, 2.62]
4	20	20,000	2.61 [2.61, 2.62]
Panel D: Empirical Convergence Ratios Conditional on Joint Success			
E_b/N_0	**Cap**		**Empirical First-Iteration Ratio [95% Interval]**
0	5		NA
0	10		NA
0	20		NA
2	5		NA
2	10		1.659 [1.657, 1.661]
2	20		1.667 [1.666, 1.669]
4	5		1.648 [1.646, 1.651]
4	10		1.836 [1.833, 1.839]
4	20		1.836 [1.833, 1.839]

Table A13. Boundary latency endpoints. Pointwise 95% percentile intervals use 5000 paired circular moving-block replicates with the acquisition block as the resampling unit and ordered circular runs of length $L = 5$; they are descriptive and not multiplicity-adjusted.

Layered schedule; 20 fixed iterations; dynamic clock mode.

Panel A: Median and 99th Percentile

		Median (P50)	**99th Percentile (P99)**
Timing Boundary	**Batch Size**	**Estimate [95% Interval], µs**	**Estimate [95% Interval], µs**
Full boundary	16	1725.9 [1717.0, 1743.1]	1986.0 [1907.8, 2376.7]
Full boundary	64	2033.1 [2014.5, 2049.0]	2464.1 [2242.8, 2927.6]
Full boundary	128	5623.3 [5608.3, 5639.2]	6560.1 [6237.4, 6800.2]
Full boundary	256	12,282.5 [12,267.0, 12,298.2]	13,618.3 [13,320.9, 13,735.5]
Full boundary	512	24,479.5 [24,455.7, 24,504.4]	26,949.6 [26,240.5, 27,513.3]
Decode-only boundary	16	1566.3 [1561.3, 1568.7]	1657.3 [1648.3, 1679.1]
Decode-only boundary	64	1599.5 [1596.2, 1601.9]	1729.9 [1709.6, 1834.9]
Decode-only boundary	128	4817.3 [4810.2, 4824.5]	5736.0 [5083.6, 6107.8]
Decode-only boundary	256	10,785.8 [10,773.2, 10,797.5]	12,064.7 [11,748.4, 12,216.0]
Decode-only boundary	512	21,555.0 [21,539.1, 21,568.0]	23,685.0 [23,057.9, 24,021.6]
Transfer-only boundary	16	124.9 [122.6, 127.3]	189.6 [182.6, 197.8]
Transfer-only boundary	64	356.2 [354.4, 358.1]	478.1 [456.7, 495.2]
Transfer-only boundary	128	669.4 [662.6, 677.5]	880.2 [819.4, 896.3]
Transfer-only boundary	256	1280.4 [1273.3, 1287.3]	1642.8 [1454.3, 1677.1]
Transfer-only boundary	512	2478.2 [2470.1, 2486.9]	2743.3 [2667.0, 2883.8]

Panel B: P99 Separation Measures

		P99–Median Gap (J_{99})	**P99-to-P50 Ratio (R_{99})**
Timing Boundary	**Batch Size**	**Estimate [95% Interval], µs**	**Estimate [95% Interval]**
Full boundary	16	260.1 [180.5, 650.6]	1.151 [1.104, 1.377]
Full boundary	64	431.0 [206.1, 896.4]	1.212 [1.101, 1.441]
Full boundary	128	936.8 [619.6, 1172.0]	1.167 [1.110, 1.208]
Full boundary	256	1335.7 [1042.3, 1447.7]	1.109 [1.085, 1.118]
Full boundary	512	2470.1 [1764.0, 3014.9]	1.101 [1.072, 1.123]
Decode-only boundary	16	91.0 [82.8, 113.1]	1.058 [1.053, 1.072]
Decode-only boundary	64	130.4 [109.7, 237.3]	1.082 [1.069, 1.149]
Decode-only boundary	128	918.7 [265.7, 1290.8]	1.191 [1.055, 1.268]
Decode-only boundary	256	1279.0 [962.6, 1426.7]	1.119 [1.089, 1.132]
Decode-only boundary	512	2130.0 [1506.0, 2464.1]	1.099 [1.070, 1.114]
Transfer-only boundary	16	64.7 [58.2, 72.6]	1.518 [1.464, 1.583]
Transfer-only boundary	64	121.9 [100.9, 138.2]	1.342 [1.283, 1.388]
Transfer-only boundary	128	210.8 [153.1, 225.3]	1.315 [1.230, 1.337]
Transfer-only boundary	256	362.4 [175.7, 397.5]	1.283 [1.137, 1.311]
Transfer-only boundary	512	265.0 [189.9, 403.7]	1.107 [1.077, 1.163]

Table A14. Boundary paired endpoint contrasts (right minus left). Pointwise 95% percentile intervals use 5000 paired circular moving-block replicates with the acquisition block as the resampling unit and ordered circular runs of length $L = 5$; they are descriptive and not multiplicity-adjusted.

Layered schedule; 20 fixed iterations; dynamic clock mode. Each publication label gives the subtraction order.

Panel A: Median and 99th Percentile

Contrast	Batch Size	Median (P50) Estimate [95% Interval], µs	99th Percentile (P99) Estimate [95% Interval], µs
Decode-only minus full	16	−159.6 [−177.5, −151.0]	−328.7 [−711.5, −251.5]
Decode-only minus full	64	−433.6 [−449.3, −415.5]	−734.3 [−1192.8, −500.8]
Decode-only minus full	128	−805.9 [−819.4, −793.4]	−824.1 [−1277.6, −498.9]
Decode-only minus full	256	−1496.8 [−1515.3, −1478.0]	−1553.5 [−1726.9, −1405.6]
Decode-only minus full	512	−2924.5 [−2952.5, −2901.9]	−3264.5 [−3713.8, −2923.3]
Transfer-only minus full	16	−1601.0 [−1617.3, −1593.0]	−1796.4 [−2184.1, −1718.7]
Transfer-only minus full	64	−1676.9 [−1692.5, −1658.5]	−1986.1 [−2450.0, −1761.0]
Transfer-only minus full	128	−4953.9 [−4968.7, −4939.8]	−5679.9 [−5920.0, −5370.4]
Transfer-only minus full	256	−11,002.1 [−11,017.4, −10,986.1]	−11,975.4 [−12,183.2, −11,699.4]
Transfer-only minus full	512	−22,001.2 [−22,023.9, −21,979.9]	−24,206.3 [−24,730.7, −23,533.5]

Panel B: P99 Separation Measures

Contrast	Batch Size	P99–Median Gap (J_{99}) Estimate [95% Interval], µs	P99-to-P50 Ratio (R_{99}) Estimate [95% Interval]
Decode-only minus full	16	−169.1 [−550.9, −90.0]	−0.093 [−0.314, −0.047]
Decode-only minus full	64	−300.6 [−761.1, −65.8]	−0.130 [−0.358, −0.012]
Decode-only minus full	128	−18.1 [−475.4, 307.3]	0.024 [−0.075, 0.091]
Decode-only minus full	256	−56.7 [−229.2, 89.0]	0.010 [−0.007, 0.022]
Decode-only minus full	512	−340.0 [−784.5, −1.9]	−0.002 [−0.022, 0.012]
Transfer-only minus full	16	−195.4 [−585.1, −116.2]	0.367 [0.143, 0.447]
Transfer-only minus full	64	−309.1 [−775.8, −82.2]	0.130 [−0.113, 0.260]
Transfer-only minus full	128	−726.1 [−965.1, −416.8]	0.148 [0.060, 0.196]
Transfer-only minus full	256	−973.3 [−1176.7, −703.2]	0.174 [0.029, 0.207]
Transfer-only minus full	512	−2205.1 [−2728.2, −1543.6]	0.006 [−0.030, 0.062]

Table A15. Boundary-study component medians. The synchronized host total uses the host steady clock. The raw components, signed remainder, and event-zero-to-event-five total use the CUDA-event clock, which was not cross-calibrated to host elapsed time. Observation by observation, the five named raw components plus the signed remainder equal the separately queried total by construction; this is an arithmetic within-stream identity check, not evidence of an independent decomposition. Separately computed medians need not obey the observation-level identity.

Boundary	Batch	Host Total	CUDA Total	H2D	Reset	Decode	Decision	D2H	Remainder
full	16	1725.9	1698.8	77.6	26.8	1554.4	4.1	28.0	−0.000017
full	64	2033.1	1986.0	296.6	42.2	1589.2	5.1	46.1	0.000009
full	128	5623.3	5606.9	581.6	93.5	4828.2	10.2	74.2	0.000003
full	256	12,282.5	12,373.6	1144.6	201.6	10,867.7	17.4	116.9	0.000006
full	512	24,479.5	24,765.1	2270.3	404.3	21,809.2	35.8	197.7	0.000019
decode-only	16	1566.3	1564.3	NA	NA	1554.7	NA	NA	8.319974
decode-only	64	1599.5	1598.5	NA	NA	1589.2	NA	NA	8.383989
decode-only	128	4817.3	4842.5	NA	NA	4831.2	NA	NA	8.192062
decode-only	256	10,785.8	10,882.0	NA	NA	10,870.8	NA	NA	7.904053
decode-only	512	21,555.0	21,824.5	NA	NA	21,812.2	NA	NA	8.127213
transfer-only	16	124.9	107.3	93.1	NA	NA	NA	10.1	4.128000
transfer-only	64	356.2	339.1	308.3	NA	NA	NA	25.8	4.127998
transfer-only	128	669.4	645.0	594.2	NA	NA	NA	46.1	4.127994
transfer-only	256	1280.4	1248.2	1154.9	NA	NA	NA	86.8	4.127972
transfer-only	512	2478.2	2459.9	2284.7	NA	NA	NA	168.8	4.127935

Table A16. Component P99 values for the tested batch-512, 20-iteration cells. Panel A uses the synchronized host steady-clock domain; Panel B gives raw CUDA-event-clock values that were not cross-calibrated to host elapsed time. Values across panels are not compared or combined. The CUDA total is separately queried using the same event-zero and event-five endpoint events; the signed remainder is that total minus the five named component intervals. This arithmetic within-stream identity check does not constitute independent decomposition evidence.

Panel A: Host Steady-Clock P99 (µs)							
Configuration	**Total**	**H2D Submit**	**Reset Submit**	**Decode Submit**	**Decision Submit**	**D2H Submit**	**Synchronize Wait**
schedule layered	26,445.8	25.9	64.1	20.4	14.8	35.9	26,347.2
schedule flooding	39,548.1	27.7	81.9	24.2	13.1	38.1	39,427.7
boundary full	26,949.6	26.7	67.1	20.4	15.0	33.8	26,854.2
boundary decode-only	23,685.0	NA	NA	28.8	NA	NA	23,659.2
boundary transfer-only	2743.3	33.2	NA	NA	NA	28.3	2708.1
Panel B: Raw CUDA-Event P99 (µs)							
Configuration	**Event Total**	**H2D**	**Reset**	**Decode**	**Decision**	**D2H**	**Signed Remainder**
schedule layered	26,678.1	2626.9	565.2	23,509.0	41.0	299.6	0.001658
schedule flooding	40,024.3	2562.8	1179.6	36,486.1	41.0	292.1	0.003297
boundary full	27,208.1	2637.4	568.8	24,196.1	39.9	297.4	0.001643
boundary decode-only	23,938.9	NA	NA	23,917.6	NA	NA	60.415268
boundary transfer-only	2699.1	2490.5	NA	NA	NA	205.9	4.448146

Table A17. Observation-level timer consistency for the five batch-512, 20-iteration dynamic-clock cells. The synchronization ratios and host-total-minus-wait P99 values use paired host-clock observations. The host-shorter count and host-minus-CUDA mean test cross-domain enclosure; because that check failed, the CUDA-event values are not treated as calibrated to host elapsed time. These diagnostics are not an additive timing decomposition.

Panel A: Cross-Domain Enclosure Diagnostic			
Configuration	***N***	**Host Shorter, *n* (%)**	**Host−CUDA Mean (µs)**
boundary: decode-only	20,000	19,619 (98.1)	−279.7
boundary: full	20,000	19,341 (96.7)	−287.7
boundary: transfer-only	20,000	8448 (42.2)	18.5
schedule: flooding	20,000	19,347 (96.7)	−478.6
schedule: layered	20,000	18,826 (94.1)	−250.6
Panel B: Paired Host-Clock Summaries			
Configuration	**Sync/Total P50**	**Sync/Total P99**	**(Total−Sync) P99 (µs)**
boundary: decode-only	0.9991	0.9994	72.7
boundary: full	0.9973	0.9984	169.4
boundary: transfer-only	0.9897	0.9931	74.7
schedule: flooding	0.9980	0.9987	191.3
schedule: layered	0.9974	0.9984	170.2

Table A18. Six-session stability summary for the batch-64-to-128 J_{99} contrasts under the schedule-generalized benchmark. The estimand is the equal-weight mean of the six session-specific contrasts, not a pooled-observation quantile contrast. Intervals are pointwise descriptive 95% percentile intervals from a two-stage bootstrap that resamples sessions and then paired ordered blocks within the selected session (5000 replicates; inner primary $L = 4$); they are neither calibrated population-coverage intervals nor multiplicity-adjusted tests.

Configuration	**Mean ΔJ_{99}**	**Session Min/Max**	**Leave-One-Session-Out Min/Max**	**95% Interval**
flooding; full	434.7	317.2/625.4	396.6/458.2	[322.1, 564.4]
layered; decode-only	179.4	150.8/207.0	173.9/185.1	[151.5, 198.1]
layered; full	234.7	200.0/272.9	227.1/241.7	[196.1, 281.5]
layered; transfer-only	24.5	19.2/38.1	21.8/25.6	[13.8, 34.5]

Table A19. Operating-state resampling sensitivity. Direction labels compare the primary paired circular moving-block result at $L = 4$ with $L = 3$, $L = 8$, and paired independent whole-cycle resampling. Each method uses 5000 replicates and pointwise descriptive 95% percentile intervals; no interval is multiplicity-adjusted.

Contrast	Endpoint	$L = 4$	$L = 3$	$L = 8$	Whole Cycle	Disagree
Preheated—idle; full	P_{50} (µs)	uncertain	uncertain	uncertain	uncertain	No
Preheated—idle; full	P_{99} (µs)	increase	increase	increase	uncertain	Yes
Preheated—idle; full	J_{99} (µs)	increase	uncertain	increase	uncertain	Yes
Preheated—idle; full	R_{99}	uncertain	uncertain	uncertain	uncertain	No
Preheated—idle; decode-only	P_{50} (µs)	uncertain	uncertain	uncertain	uncertain	No
Preheated—idle; decode-only	P_{99} (µs)	uncertain	uncertain	uncertain	uncertain	No
Preheated—idle; decode-only	J_{99} (µs)	uncertain	uncertain	uncertain	uncertain	No
Preheated—idle; decode-only	R_{99}	uncertain	uncertain	uncertain	uncertain	No
Treatment response: full, decode-only	P_{50} (µs)	uncertain	uncertain	uncertain	uncertain	No
Treatment response: full, decode-only	P_{99} (µs)	uncertain	uncertain	uncertain	uncertain	No
Treatment response: full, decode-only	J_{99} (µs)	uncertain	uncertain	uncertain	uncertain	No
Treatment response: full, decode-only	R_{99}	uncertain	uncertain	uncertain	uncertain	No

Table A20. Timer-consistency diagnostics for selected session-replication cells. Synchronization fractions and host-total-minus-wait values use paired host-clock observations. Host-shorter counts test enclosure across distinct clock domains; failure of that check precludes treating the raw CUDA-event values as calibrated to host elapsed time. Ranges span the listed cells or sessions.

Panel A: Cross-Domain Enclosure Diagnostic

Condition	Cells	*N*	Host Shorter, *n* (%)
Tail sessions: layered, full, batch 64	6	48,000	2494 (5.2)
Tail sessions: layered, full, batch 128	6	48,000	9075 (18.9)
Operating state: idle-cooled, full	1	6000	5838 (97.3)
Operating state: idle-cooled, decode-only	1	6000	5844 (97.4)
Operating state: preheated, full	1	6000	5393 (89.9)
Operating state: preheated, decode-only	1	6000	5473 (91.2)
Placement: none, batch 512	1	6000	5808 (96.8)
Placement: block-trace, batch 512	1	6000	5929 (98.8)

Panel B: Paired Host-Clock Summaries

Condition	Sync/Total P50 Range	(Total−Sync) P99 Range (µs)
Tail sessions: layered, full, batch 64	0.9790–0.9814	104.5–142.6
Tail sessions: layered, full, batch 128	0.9909–0.9924	114.5–146.7
Operating state: idle-cooled, full	0.9976–0.9976	160.1–160.1
Operating state: idle-cooled, decode-only	0.9992–0.9992	72.8–72.8
Operating state: preheated, full	0.9976–0.9976	160.2–160.2
Operating state: preheated, decode-only	0.9992–0.9992	74.5–74.5
Placement: none, batch 512	0.9992–0.9992	69.4–69.4
Placement: block-trace, batch 512	0.9992–0.9992	79.3–79.3

Table A21. Operating ranges from continuous 100-ms telemetry. Statistics use samples with a valid metric-specific status; valid and total sample counts are stated in each panel heading.

Schedule comparison (66,334 valid of 66,334 samples)

Metric	Min.	P5	Median	P95	Max.
Temperature (°C)	34.0	50.0	57.0	60.0	63.0
SM clock (MHz)	210.0	2730.0	2730.0	2745.0	2760.0
Memory clock (MHz)	405.0	10,251.0	10,251.0	10,501.0	10,501.0
Power (W)	9.0	91.8	175.2	226.9	229.8
GPU utilization (%)	0.0	1.0	82.0	92.0	94.0
Memory utilization (%)	0.0	0.0	74.0	91.0	93.0

Table A21. *Cont.*

Boundary comparison (30,585 valid of 30,585 samples)					
Metric	**Min.**	**P5**	**Median**	**P95**	**Max.**
Temperature (°C)	39.0	49.0	59.0	61.0	63.0
SM clock (MHz)	210.0	2730.0	2730.0	2745.0	2745.0
Memory clock (MHz)	405.0	10,251.0	10,251.0	10,501.0	10,501.0
Power (W)	8.1	56.2	203.9	227.6	230.0
GPU utilization (%)	0.0	1.0	86.0	88.0	91.0
Memory utilization (%)	0.0	0.0	85.0	87.0	88.0

Table A22. Descriptive Spearman associations between block-level latency summaries and operating-condition summaries for the baseline and algorithmic-control layered BG1 $Z = 384$, 20-iteration acquisitions. Each row contains 100 acquisition blocks. Baseline telemetry comprises one before/after pair per complete block; algorithmic-control telemetry uses visit-level samples. Temperature, clocks, power, utilization, and order covary, so these coefficients neither isolate a causal effect nor provide a fixed-clock comparison. N/E denotes a coefficient that is not estimable because memory clock was constant.

Panel A: Block-Mean Latency						
Acquisition and Batch	**Temperature**	**SM Clock**	**Memory Clock**	**Power**	**GPU Utilization**	**Acquisition Order**
Baseline layered, batch 256	−0.037	0.092	N/E	−0.130	−0.248	0.169
Algorithmic control, batch 256	0.014	−0.068	0.004	0.061	0.023	−0.066
Algorithmic control, batch 512	0.363	0.128	0.124	0.451	0.431	−0.090
Panel B: Block-P99 Latency						
Acquisition and Batch	**Temperature**	**SM Clock**	**Memory Clock**	**Power**	**GPU Utilization**	**Acquisition Order**
Baseline layered, batch 256	−0.037	0.042	N/E	−0.079	−0.213	0.142
Algorithmic control, batch 256	0.457	0.187	0.204	0.470	0.521	−0.035
Algorithmic control, batch 512	0.642	0.326	0.307	0.717	0.694	−0.252

Table A23. NVML clock-event counts among valid samples. The GPU-idle bit is informational rather than a thermal or power-throttling event. Every other documented bit—application clock setting, software power cap, hardware slowdown, synchronized boost, software or hardware thermal slowdown, power brake, and display clock setting—was zero in both studies.

Study	**Valid Samples**	**GPU-Idle Bit**	**Other Documented Bits (Each)**	**Unrecognized Bits**
Schedule	66,334	264	0 for each	0
Boundary	30,585	11	0 for each	0

Appendix B. Algorithmic and Configuration Sensitivity

The following tables and Figure A1 report the complete workload, algorithm, convergence, stopping, channel, input, graph, and lifting-size controls summarized in the main text.

Table A24. Primary BG1, $Z = 384$ end-to-end batch-latency results. Panel A reports observation accounting. Pointwise 95% percentile intervals in Panels B and C use 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are descriptive and not multiplicity-adjusted. Ten planned warm-ups preceded retention at every visit; no retained observation was excluded after acquisition.

Panel A: Observation Accounting

Batch	Iterations	Retained Observations	Post-Acquisition Exclusions
4	5	20,000	0
16	5	20,000	0
64	5	20,000	0
128	5	20,000	0
192	5	20,000	0
256	5	20,000	0
320	5	20,000	0
512	5	20,000	0
4	10	20,000	0
16	10	20,000	0
64	10	20,000	0
128	10	20,000	0
192	10	20,000	0
256	10	20,000	0
320	10	20,000	0
512	10	20,000	0
4	20	20,000	0
16	20	20,000	0
64	20	20,000	0
128	20	20,000	0
192	20	20,000	0
256	20	20,000	0
320	20	20,000	0
512	20	20,000	0

Panel B: Median and 99th Percentile

Batch	Iterations	P50 (µs) Estimate [95% Interval]	P99 (µs) Estimate [95% Interval]
4	5	493.5 [492.3, 496.3]	5104.6 [5046.3, 5350.2]
16	5	555.7 [554.0, 560.6]	5644.9 [5331.5, 5746.1]
64	5	888.7 [878.2, 894.7]	7637.8 [6108.5, 7975.6]
128	5	2051.8 [2045.0, 2057.5]	6518.4 [6200.5, 7492.8]
192	5	3137.2 [3131.7, 3143.3]	8611.7 [8001.6, 9262.4]
256	5	4282.3 [4276.2, 4288.6]	9722.8 [9209.9, 10,260.6]
320	5	5456.1 [5446.8, 5466.1]	9776.8 [9496.0, 10,264.2]
512	5	8448.8 [8438.4, 8461.9]	10,350.3 [9187.4, 14,115.1]
4	10	920.0 [916.7, 924.6]	9181.5 [9008.1, 9411.8]
16	10	987.3 [978.5, 999.1]	8474.8 [7919.9, 9169.0]
64	10	1274.6 [1266.0, 1283.9]	10,205.0 [9494.5, 11,254.0]
128	10	3243.1 [3235.9, 3250.2]	10,230.9 [9554.9, 10,810.8]
192	10	5031.3 [5023.5, 5040.0]	9054.2 [7820.1, 11,252.8]
256	10	6949.1 [6937.1, 6963.0]	9505.6 [7811.1, 12,375.7]
320	10	8921.7 [8909.3, 8937.2]	9484.8 [9370.5, 9659.6]
512	10	13,805.4 [13,790.8, 13,820.3]	14,525.8 [14,366.1, 14,665.9]
4	20	1722.5 [1718.1, 1728.1]	12,825.2 [12,014.2, 13,460.8]
16	20	1786.0 [1779.5, 1792.2]	12,555.5 [11,328.7, 13,219.0]
64	20	2047.7 [2042.9, 2052.5]	11,928.3 [11,366.6, 12,460.7]
128	20	5650.3 [5641.7, 5660.6]	6514.5 [6071.9, 8093.9]
192	20	8817.3 [8804.0, 8832.8]	9419.7 [9247.4, 9681.6]
256	20	12,263.5 [12,251.1, 12,276.7]	12,905.9 [12,741.9, 13,050.8]
320	20	15,801.7 [15,785.2, 15,816.3]	16,680.9 [16,522.2, 16,767.2]
512	20	24,485.8 [24,468.3, 24,504.5]	25,785.1 [25,618.6, 25,930.6]

Table A24. *Cont.*

Panel C: P99 Separation Measures			
Batch	**Iterations**	J_{99} (µs) **Estimate [95% Interval]**	R_{99} **Estimate [95% Interval]**
4	5	4611.1 [4553.0, 4855.1]	10.344 [10.221, 10.822]
16	5	5089.2 [4776.3, 5189.7]	10.158 [9.611, 10.341]
64	5	6749.1 [5218.6, 7091.3]	8.595 [6.869, 9.019]
128	5	4466.6 [4151.0, 5442.8]	3.177 [3.024, 3.655]
192	5	5474.5 [4866.2, 6126.3]	2.745 [2.551, 2.954]
256	5	5440.5 [4924.0, 5980.3]	2.270 [2.149, 2.397]
320	5	4320.8 [4039.4, 4802.7]	1.792 [1.741, 1.879]
512	5	1901.5 [736.5, 5671.7]	1.225 [1.087, 1.672]
4	10	8261.5 [8088.9, 8491.1]	9.980 [9.785, 10.220]
16	10	7487.5 [6920.2, 8175.1]	8.584 [7.964, 9.277]
64	10	8930.3 [8219.7, 9980.0]	8.006 [7.459, 8.845]
128	10	6987.7 [6308.8, 7567.7]	3.155 [2.944, 3.335]
192	10	4022.9 [2788.8, 6224.7]	1.800 [1.554, 2.238]
256	10	2556.5 [866.3, 5420.3]	1.368 [1.125, 1.779]
320	10	563.0 [452.5, 738.9]	1.063 [1.051, 1.083]
512	10	720.4 [565.8, 860.2]	1.052 [1.041, 1.062]
4	20	11,102.7 [10,292.3, 11,735.6]	7.446 [6.971, 7.816]
16	20	10,769.5 [9551.3, 11,438.9]	7.030 [6.357, 7.412]
64	20	9880.5 [9315.1, 10,413.9]	5.825 [5.543, 6.091]
128	20	864.2 [424.9, 2443.2]	1.153 [1.075, 1.432]
192	20	602.4 [428.4, 865.4]	1.068 [1.049, 1.098]
256	20	642.5 [480.3, 787.3]	1.052 [1.039, 1.064]
320	20	879.2 [724.6, 966.1]	1.056 [1.046, 1.061]
512	20	1299.2 [1141.2, 1434.3]	1.053 [1.047, 1.059]

Table A25. Finite-sample primary tail descriptions in µs from 20,000 retained observations per cell; no retained observation was excluded. P99.9 and the observed maximum have no bootstrap intervals and are not used to classify configuration contrasts or as worst-case bounds.

	5 Iterations		10 Iterations		20 Iterations	
Batch	**P99.9**	**Maximum**	**P99.9**	**Maximum**	**P99.9**	**Maximum**
4	6982.1	9763.9	9915.9	13,109.0	18,020.4	30,703.8
16	6919.2	10,834.7	10,269.8	16,893.9	17,948.3	21,538.4
64	9803.2	101,653.1	14,099.8	24,428.8	18,433.6	26,561.0
128	21,471.6	27,033.3	13,988.7	15,123.0	15,256.1	22,572.8
192	11,353.2	33,849.8	14,587.8	21,835.8	12,192.7	21,893.0
256	13,724.4	18,989.1	16,127.8	21,298.7	14,049.7	22,895.1
320	14,397.1	18,509.0	16,844.5	19,842.2	17,820.3	20,871.2
512	15,057.6	23,752.0	16,366.0	26,919.3	27,021.6	29,026.9

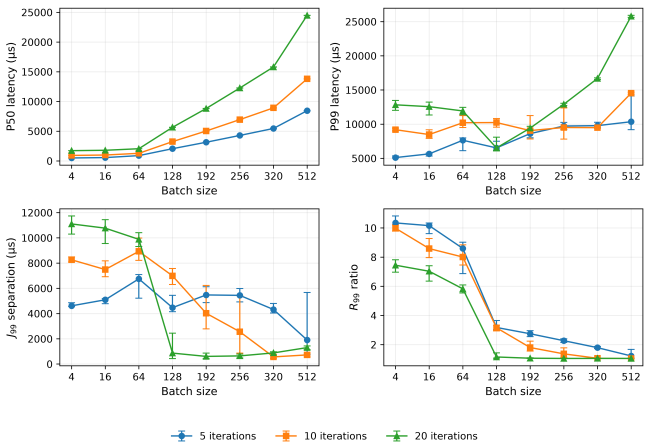


Figure A1. BG1 $Z = 384$ scaling endpoints at batches 4, 16, 64, 128, 192, 256, 320, and 512 and fixed iteration counts 5, 10, and 20. Error bars are pointwise descriptive 95% percentile intervals from 5000 paired

circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are not multiplicity-adjusted. The categorical horizontal axis displays every tested batch.

Table A26. Algorithm sensitivity at 20 fixed iterations, $E_b/N_0 = 2$ dB, and BG1 $Z = 384$. Panel A reports algorithm parameters and observation counts. Pointwise 95% percentile intervals in Panels B and C use 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are descriptive and not multiplicity-adjusted. Each cell has zero post-acquisition exclusions.

Panel A: Algorithm Parameters and Observation Counts				
Algorithm	α	β	**Batch**	**Retained Observations**
ms	1	0	16	20,000
nms-0.6	0.6	0	16	20,000
nms-0.7	0.7	0	16	20,000
nms-0.75	0.75	0	16	20,000
nms-0.8	0.8	0	16	20,000
nms-0.9	0.9	0	16	20,000
oms-0.5	1	0.5	16	20,000
ms	1	0	256	20,000
nms-0.6	0.6	0	256	20,000
nms-0.7	0.7	0	256	20,000
nms-0.75	0.75	0	256	20,000
nms-0.8	0.8	0	256	20,000
nms-0.9	0.9	0	256	20,000
oms-0.5	1	0.5	256	20,000

Panel B: Median and 99th Percentile			
		P50 (µs)	**P99 (µs)**
Algorithm	**Batch**	**Estimate [95% Interval]**	**Estimate [95% Interval]**
ms	16	1787.7 [1781.9, 1794.4]	12,893.9 [11,687.5, 13,381.4]
nms-0.6	16	1786.4 [1781.6, 1791.8]	12,767.0 [12,508.2, 13,363.4]
nms-0.7	16	1789.0 [1783.3, 1796.2]	13,199.5 [11,967.8, 14,121.2]
nms-0.75	16	1792.2 [1786.4, 1797.1]	12,702.6 [12,053.6, 13,287.1]
nms-0.8	16	1789.6 [1784.2, 1794.8]	12,644.9 [11,701.7, 14,287.4]
nms-0.9	16	1786.5 [1779.8, 1792.5]	13,265.0 [12,640.5, 14,122.0]
oms-0.5	16	1791.1 [1786.1, 1797.2]	12,802.6 [11,564.8, 13,641.8]
ms	256	12,347.7 [12,336.9, 12,358.9]	13,102.6 [12,922.1, 13,205.8]
nms-0.6	256	12,357.8 [12,345.4, 12,369.8]	13,161.3 [12,962.3, 13,287.7]
nms-0.7	256	12,364.0 [12,351.8, 12,378.1]	13,158.7 [12,991.8, 13,272.3]
nms-0.75	256	12,362.1 [12,351.5, 12,372.0]	13,192.0 [13,031.0, 13,285.1]
nms-0.8	256	12,353.2 [12,342.0, 12,366.1]	13,047.9 [12,923.8, 13,183.3]
nms-0.9	256	12,352.4 [12,341.5, 12,362.1]	13,075.5 [12,945.0, 13,198.6]
oms-0.5	256	12,356.3 [12,342.3, 12,370.0]	13,165.0 [12,953.9, 13,296.3]

Panel C: P99 Separation Measures			
		J_{99} **(µs)**	R_{99}
Algorithm	**Batch**	**Estimate [95% Interval]**	**Estimate [95% Interval]**
ms	16	11,106.2 [9896.8, 11,593.4]	7.213 [6.537, 7.489]
nms-0.6	16	10,980.6 [10,720.6, 11,575.2]	7.147 [6.997, 7.476]
nms-0.7	16	11,410.6 [10,176.3, 12,329.6]	7.378 [6.683, 7.879]
nms-0.75	16	10,910.4 [10,260.6, 11,492.5]	7.088 [6.727, 7.420]
nms-0.8	16	10,855.3 [9914.4, 12,496.2]	7.066 [6.537, 7.973]
nms-0.9	16	11,478.6 [10,849.7, 12,332.1]	7.425 [7.081, 7.889]
oms-0.5	16	11,011.4 [9775.6, 11,847.1]	7.148 [6.467, 7.600]
ms	256	754.9 [577.6, 849.1]	1.061 [1.047, 1.069]
nms-0.6	256	803.5 [608.7, 924.9]	1.065 [1.049, 1.075]
nms-0.7	256	794.8 [625.5, 909.9]	1.064 [1.051, 1.074]
nms-0.75	256	829.9 [672.1, 918.7]	1.067 [1.054, 1.074]
nms-0.8	256	694.7 [574.7, 828.9]	1.056 [1.047, 1.067]
nms-0.9	256	723.0 [592.0, 845.8]	1.059 [1.048, 1.068]
oms-0.5	256	808.7 [597.7, 938.3]	1.065 [1.048, 1.076]

Table A27. Untimed error and convergence study for 20,000 random valid BG1 $Z = 384$ codewords per cell with a 20-iteration cap. Panel A reports information-bit BER, information-word BLER, and syndrome-success rate. Panel B reports first-satisfied-iteration summaries conditional on words that satisfied the syndrome within the cap; NA denotes no successful word. The OMS 0-dB convergence summaries are conditional on one successful word.

Panel A: Error and Syndrome Outcomes				
Algorithm	E_b/N_0 **(dB)**	**Information-Bit BER**	**Information-Word BLER**	**Syndrome-Success Rate**
ms	0	3.120×10^{-1}	1.0000	0.0000
ms	2	8.128×10^{-2}	0.3230	0.6764
ms	4	0	0.0000	1.0000
nms-0.6	0	1.875×10^{-1}	1.0000	0.0000
nms-0.6	2	0	0.0000	1.0000
nms-0.6	4	0	0.0000	1.0000
nms-0.7	0	2.247×10^{-1}	1.0000	0.0000
nms-0.7	2	0	0.0000	1.0000
nms-0.7	4	0	0.0000	1.0000
nms-0.75	0	2.420×10^{-1}	1.0000	0.0000
nms-0.75	2	0	0.0000	1.0000
nms-0.75	4	0	0.0000	1.0000
nms-0.8	0	2.580×10^{-1}	1.0000	0.0000
nms-0.8	2	0	0.0000	1.0000
nms-0.8	4	0	0.0000	1.0000
nms-0.9	0	2.870×10^{-1}	1.0000	0.0000
nms-0.9	2	0	0.0000	1.0000
nms-0.9	4	0	0.0000	1.0000
oms-0.5	0	1.623×10^{-1}	0.9999	0.0001
oms-0.5	2	0	0.0000	1.0000
oms-0.5	4	0	0.0000	1.0000

Panel B: Conditional First-Satisfied-Iteration Summaries			
Algorithm	E_b/N_0 **(dB)**	**Conditional Mean First Iteration**	**Conditional Median First Iteration**
ms	0	NA	NA
ms	2	10.40	10
ms	4	3.60	4
nms-0.6	0	NA	NA
nms-0.6	2	6.31	6
nms-0.6	4	3.58	4
nms-0.7	0	NA	NA
nms-0.7	2	5.79	6
nms-0.7	4	3.18	3
nms-0.75	0	NA	NA
nms-0.75	2	5.77	6
nms-0.75	4	3.12	3
nms-0.8	0	NA	NA
nms-0.8	2	5.90	6
nms-0.8	4	3.10	3
nms-0.9	0	NA	NA
nms-0.9	2	6.66	7
nms-0.9	4	3.17	3
oms-0.5	0	20.00	20
oms-0.5	2	5.20	5
oms-0.5	4	3.03	3

Table A28. Channel, input, and stopping-policy sensitivity (20-iteration cap, BG1 $Z = 384$). Panel A reports the configuration and mean executed iterations. Pointwise 95% percentile intervals in Panels B–D use 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are descriptive and not multiplicity-adjusted. Random-minus-all-zero contrasts are reported separately.

Panel A: Configuration and Mean Executed Iterations

Batch	E_b/N_0 (dB)	Input	Stopping	Mean Executed Iterations
16	0	all-zero	early	20.00
16	0	all-zero	fixed	20.00
16	0	random	early	20.00
16	0	random	fixed	20.00
16	2	all-zero	early	5.77
16	2	all-zero	fixed	20.00
16	2	random	early	5.75
16	2	random	fixed	20.00
16	4	all-zero	early	3.13
16	4	all-zero	fixed	20.00
16	4	random	early	3.12
16	4	random	fixed	20.00
256	0	all-zero	early	20.00
256	0	all-zero	fixed	20.00
256	0	random	early	20.00
256	0	random	fixed	20.00
256	2	all-zero	early	5.77
256	2	all-zero	fixed	20.00
256	2	random	early	5.78
256	2	random	fixed	20.00
256	4	all-zero	early	3.12
256	4	all-zero	fixed	20.00
256	4	random	early	3.12
256	4	random	fixed	20.00

Panel B: Median (P50)

Batch	E_b/N_0 (dB)	Input	Stopping	P50 (µs) Estimate [95% Interval]
16	0	all-zero	early	2322.5 [2317.6, 2326.9]
16	0	all-zero	fixed	1779.3 [1773.9, 1785.7]
16	0	random	early	2316.0 [2309.9, 2322.5]
16	0	random	fixed	1785.6 [1778.8, 1792.5]
16	2	all-zero	early	856.4 [829.3, 879.5]
16	2	all-zero	fixed	1778.8 [1774.1, 1784.0]
16	2	random	early	861.8 [832.7, 885.6]
16	2	random	fixed	1783.5 [1776.5, 1790.3]
16	4	all-zero	early	579.7 [573.5, 588.7]
16	4	all-zero	fixed	1783.6 [1777.0, 1790.8]
16	4	random	early	578.5 [570.6, 589.4]
16	4	random	fixed	1784.4 [1778.7, 1790.9]
256	0	all-zero	early	16,502.1 [16,396.9, 16,549.4]
256	0	all-zero	fixed	16,049.3 [15,925.1, 16,096.4]
256	0	random	early	16,477.5 [16,349.0, 16,520.4]
256	0	random	fixed	16,040.8 [15,921.7, 16,079.9]
256	2	all-zero	early	5962.5 [5891.6, 5996.6]
256	2	all-zero	fixed	16,033.8 [15,915.5, 16,082.1]
256	2	random	early	5954.4 [5875.9, 5987.6]
256	2	random	fixed	16,042.3 [15,940.2, 16,083.4]
256	4	all-zero	early	4012.6 [3952.8, 4040.7]
256	4	all-zero	fixed	16,037.3 [15,909.6, 16,079.5]
256	4	random	early	4012.9 [3957.3, 4040.7]
256	4	random	fixed	16,042.4 [15,933.5, 16,083.1]

Table A28. *Cont.*

Panel C: 99th Percentile (P99)				
				P99 (µs)
Batch	**E_b/N_0 (dB)**	**Input**	**Stopping**	**Estimate [95% Interval]**
16	0	all-zero	early	6702.4 [3711.2, 9139.0]
16	0	all-zero	fixed	11,726.9 [8644.0, 12,398.2]
16	0	random	early	3596.4 [2999.7, 4894.4]
16	0	random	fixed	12,639.2 [11,663.9, 13,300.7]
16	2	all-zero	early	7576.4 [6800.4, 8359.8]
16	2	all-zero	fixed	12,026.0 [11,016.0, 12,772.2]
16	2	random	early	7645.3 [6981.6, 8393.3]
16	2	random	fixed	11,985.8 [10,577.3, 13,251.2]
16	4	all-zero	early	6023.8 [5557.0, 6152.6]
16	4	all-zero	fixed	11,780.7 [9637.0, 13,264.5]
16	4	random	early	5669.6 [4893.5, 5953.0]
16	4	random	fixed	12,372.7 [10,491.8, 13,266.5]
256	0	all-zero	early	17,036.3 [16,939.4, 17,083.2]
256	0	all-zero	fixed	16,674.1 [16,485.7, 16,975.8]
256	0	random	early	16,907.0 [16,830.1, 16,964.2]
256	0	random	fixed	16,559.8 [16,446.5, 16,600.1]
256	2	all-zero	early	8415.8 [7300.4, 9831.2]
256	2	all-zero	fixed	16,555.7 [16,392.2, 16,619.9]
256	2	random	early	7817.2 [6388.9, 10,201.4]
256	2	random	fixed	16,542.1 [16,386.4, 16,614.9]
256	4	all-zero	early	8786.0 [7931.4, 10,019.8]
256	4	all-zero	fixed	16,522.4 [16,401.3, 16,639.3]
256	4	random	early	9516.3 [8441.9, 10,672.3]
256	4	random	fixed	16,662.1 [16,407.9, 16,704.2]

Panel D: P99 Separation Measures					
				J_{99} (µs)	**R_{99}**
Batch	**E_b/N_0 (dB)**	**Input**	**Stopping**	**Estimate [95% Interval]**	**Estimate [95% Interval]**
16	0	all-zero	early	4379.9 [1385.1, 6815.1]	2.886 [1.595, 3.934]
16	0	all-zero	fixed	9947.6 [6864.4, 10,623.1]	6.591 [4.858, 6.984]
16	0	random	early	1280.4 [681.1, 2575.7]	1.553 [1.294, 2.111]
16	0	random	fixed	10,853.6 [9878.2, 11,514.2]	7.079 [6.533, 7.445]
16	2	all-zero	early	6720.0 [5948.0, 7505.5]	8.846 [7.952, 9.847]
16	2	all-zero	fixed	10,247.2 [9232.8, 10,996.9]	6.761 [6.186, 7.193]
16	2	random	early	6783.5 [6124.4, 7514.6]	8.871 [8.165, 9.640]
16	2	random	fixed	10,202.3 [8794.0, 11,469.3]	6.720 [5.930, 7.437]
16	4	all-zero	early	5444.0 [4978.4, 5573.9]	10.391 [9.610, 10.648]
16	4	all-zero	fixed	9997.0 [7853.5, 11,477.4]	6.605 [5.405, 7.422]
16	4	random	early	5091.1 [4313.5, 5375.3]	9.801 [8.471, 10.330]
16	4	random	fixed	10,588.3 [8706.2, 11,480.6]	6.934 [5.873, 7.430]
256	0	all-zero	early	534.2 [456.5, 621.5]	1.032 [1.028, 1.038]
256	0	all-zero	fixed	624.8 [448.5, 936.4]	1.039 [1.028, 1.058]
256	0	random	early	429.5 [363.8, 526.3]	1.026 [1.022, 1.032]
256	0	random	fixed	519.0 [427.9, 614.7]	1.032 [1.027, 1.039]
256	2	all-zero	early	2453.3 [1321.9, 3861.0]	1.411 [1.221, 1.648]
256	2	all-zero	fixed	521.9 [389.4, 626.7]	1.033 [1.024, 1.039]
256	2	random	early	1862.8 [470.6, 4238.2]	1.313 [1.079, 1.711]
256	2	random	fixed	499.9 [356.5, 593.6]	1.031 [1.022, 1.037]
256	4	all-zero	early	4773.5 [3935.0, 6015.7]	2.190 [1.982, 2.502]
256	4	all-zero	fixed	485.1 [383.6, 611.3]	1.030 [1.024, 1.038]
256	4	random	early	5503.5 [4437.5, 6658.6]	2.371 [2.103, 2.661]
256	4	random	fixed	619.7 [395.3, 703.7]	1.039 [1.025, 1.044]

Table A29. Random-minus-all-zero input contrasts. Panel A reports mean executed-iteration differences. Pointwise 95% percentile intervals in Panels B and C use 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are descriptive and not multiplicity-adjusted. Mean-iteration differences are point estimates.

Panel A: Mean Executed-Iteration Differences			
Batch	**E_b/N_0 (dB)**	**Stopping**	**Δ Mean Executed Iterations**
16	0	early	0.000
16	0	fixed	0.000
16	2	early	−0.013
16	2	fixed	0.000
16	4	early	−0.012
16	4	fixed	0.000
256	0	early	0.000
256	0	fixed	0.000
256	2	early	0.001
256	2	fixed	0.000
256	4	early	0.004
256	4	fixed	0.000

Panel B: Median and 99th Percentile				
			ΔP50 (µs)	**ΔP99 (µs)**
Batch	**E_b/N_0 (dB)**	**Stopping**	**Estimate [95% Interval]**	**Estimate [95% Interval]**
16	0	early	−6.5 [−14.1, 1.9]	−3106.0 [−5334.9, 78.7]
16	0	fixed	6.3 [−3.1, 15.9]	912.3 [−11.5, 3472.2]
16	2	early	5.4 [−27.2, 37.3]	68.9 [−924.0, 786.6]
16	2	fixed	4.7 [−2.0, 11.6]	−40.2 [−1436.6, 1188.9]
16	4	early	−1.3 [−9.0, 7.7]	−354.2 [−1152.8, 146.6]
16	4	fixed	0.8 [−7.1, 8.8]	592.1 [−1167.8, 2480.6]
256	0	early	−24.6 [−63.1, 1.5]	−129.3 [−213.8, −41.5]
256	0	fixed	−8.5 [−33.2, 17.3]	−114.4 [−456.3, 82.3]
256	2	early	−8.1 [−26.1, 3.9]	−598.6 [−2653.2, 2016.8]
256	2	fixed	8.5 [−17.1, 42.0]	−13.6 [−173.1, 118.6]
256	4	early	0.3 [−14.9, 17.2]	730.3 [−793.9, 2309.0]
256	4	fixed	5.1 [−19.1, 40.1]	139.7 [−121.9, 239.9]

Panel C: P99 Separation Measures				
			ΔJ_{99} (µs)	**ΔR_{99}**
Batch	**E_b/N_0 (dB)**	**Stopping**	**Estimate [95% Interval]**	**Estimate [95% Interval]**
16	0	early	−3099.4 [−5329.7, 86.1]	−1.333 [−2.298, 0.042]
16	0	fixed	906.0 [−18.2, 3469.2]	0.488 [−0.029, 1.938]
16	2	early	63.5 [−928.3, 772.6]	0.025 [−1.140, 0.865]
16	2	fixed	−44.9 [−1441.5, 1186.3]	−0.040 [−0.819, 0.648]
16	4	early	−352.9 [−1147.9, 148.4]	−0.589 [−1.940, 0.252]
16	4	fixed	591.3 [−1168.2, 2480.6]	0.329 [−0.660, 1.375]
256	0	early	−104.7 [−176.6, −22.4]	−0.006 [−0.011, −0.001]
256	0	fixed	−105.9 [−443.8, 84.9]	−0.007 [−0.028, 0.005]
256	2	early	−590.5 [−2643.2, 2026.3]	−0.099 [−0.442, 0.340]
256	2	fixed	−22.1 [−176.5, 105.0]	−0.001 [−0.011, 0.007]
256	4	early	730.0 [−797.8, 2306.7]	0.182 [−0.199, 0.573]
256	4	fixed	134.6 [−117.2, 225.8]	0.008 [−0.007, 0.014]

Table A30. Finite-sample 0-dB tail diagnostics. Panel A reports executed iterations, P97.5, and P99. Every observation reached the 20-iteration cap. P97.5 is descriptive and is not an additional primary endpoint. Panel B reports the upper tail quantiles and the probability of exceeding the fixed-cap P99.

Panel A: Executed Iterations, P97.5, and P99					
Batch	**Input**	**Stopping**	**Mean Executed Iterations**	**P97.5 (µs)**	**P99 (µs)**
16	all-zero	early	20.00	2533.4	6702.4
16	all-zero	fixed	20.00	3680.6	11,726.9
16	random	early	20.00	2514.0	3596.4
16	random	fixed	20.00	6951.9	12,639.2
256	all-zero	early	20.00	16,931.8	17,036.3
256	all-zero	fixed	20.00	16,471.7	16,674.1
256	random	early	20.00	16,812.5	16,907.0
256	random	fixed	20.00	16,462.3	16,559.8
Panel B: Upper Tail Quantiles and Fixed-Cap Exceedance					
Batch	**Input**	**Stopping**	**P99.5 (µs)**	**P99.9 (µs)**	**Pr($T > \mathrm{P99}_{\mathrm{fixed}}$)**
16	all-zero	early	11,237.7	17,542.6	0.0048
16	all-zero	fixed	13,447.5	18,957.6	0.0100
16	random	early	9785.3	18,395.2	0.0043
16	random	fixed	14,617.6	18,545.2	0.0100
256	all-zero	early	17,127.6	18,006.0	0.1933
256	all-zero	fixed	16,982.8	18,501.3	0.0100
256	random	early	17,066.1	18,145.3	0.3179
256	random	fixed	16,677.1	17,935.1	0.0100

Table A31. Base-graph and lifting-size controls at 20 fixed iterations. Panel A reports location and tail quantiles. Pointwise 95% percentile intervals use 5000 paired circular moving-block bootstrap replicates over ordered acquisition blocks ($L = 5$); they are descriptive and not multiplicity-adjusted. Panel B reports the corresponding J_{99} and R_{99} separation measures.

Panel A: Location and Tail Quantiles				
Base Graph	Z	**Batch**	**P50 (µs) Estimate [95% Interval]**	**P99 (µs) Estimate [95% Interval]**
1	192	16	1568.4 [1558.2, 1577.7]	11,595.7 [11,082.4, 12,291.7]
1	192	256	5701.8 [5687.0, 5719.3]	6416.2 [6061.4, 10,464.7]
1	384	16	1777.7 [1766.8, 1787.8]	11,969.8 [11,039.0, 13,226.8]
1	384	256	12,264.2 [12,250.2, 12,279.4]	12,733.9 [12,685.0, 12,805.0]
2	384	16	1240.1 [1218.0, 1256.7]	10,057.0 [9560.9, 10,735.7]
2	384	256	7625.7 [7610.5, 7642.3]	8044.3 [7970.8, 8339.1]
Panel B: P99 Separation Measures				
Base Graph	Z	**Batch**	J_{99} **(µs) Estimate [95% Interval]**	R_{99} **Estimate [95% Interval]**
1	192	16	10,027.3 [9512.4, 10,724.7]	7.394 [7.076, 7.838]
1	192	256	714.4 [362.1, 4766.4]	1.125 [1.064, 1.836]
1	384	16	10,192.1 [9261.8, 11,450.8]	6.733 [6.210, 7.435]
1	384	256	469.6 [419.3, 539.3]	1.038 [1.034, 1.044]
2	384	16	8817.0 [8319.7, 9502.0]	8.110 [7.689, 8.735]
2	384	256	418.5 [349.9, 709.6]	1.055 [1.046, 1.093]

Appendix C. Measurement Validation and Supplementary Methods

The scientific samples comprise 1,840,000 algorithmic-control timings, 600,000 schedule timings, 300,000 boundary timings, 384,000 session-replication timings, 24,000 operating-state timings, and 72,000 block-placement timings. The matched outcome analysis contains 360,000 decoded rows representing 60,000 unique codeword/noise pairs. No timing ob-

servation was excluded because of its magnitude. CUPTI supplied a separate diagnostic sample of 3600 traced trials with no dropped activity records.

The operating-state sample contains 30 complete cycles. In one additional cycle, a sampled software-power-cap event appeared after the preheated visits and before the idle-cooled visits. That incomplete cycle contains 400 timing records, 500 untimed preconditioning executions, and 4856 telemetry samples; none enters the primary analysis. Table A19 reports analyses that omit the corresponding complete cycle or substitute the available preheated records. The 41,000 preconditioning executions reported for the primary dataset are the total across its 30 complete cycles.

The two matched outcome acquisitions used identical codeword/noise pairs and produced identical decoded outcomes and convergence records for every word. One acquisition is analyzed, and the second establishes deterministic repeatability; they are not pooled. Their latency and visit-level telemetry are not used for inference because the experiment was designed as an untimed outcome comparison.

Four timer sources appear in the supplementary analyses: the host monotonic clock, same-stream CUDA events, `%globaltimer`, and CUPTI activity timestamps. The WSL2 host/CUDA enclosure check failed, and cross-SM `%globaltimer` comparability was not established. Cross-domain differences and cross-SM spans are therefore not interpreted as physical components. Fixed application clocks, achieved occupancy, active-warps measurements, privileged DRAM/L2 traffic counters, and per-trial PCIe attribution were unavailable. CUPTI supplied runtime, synchronization, kernel, and memory-copy intervals but no performance counters.

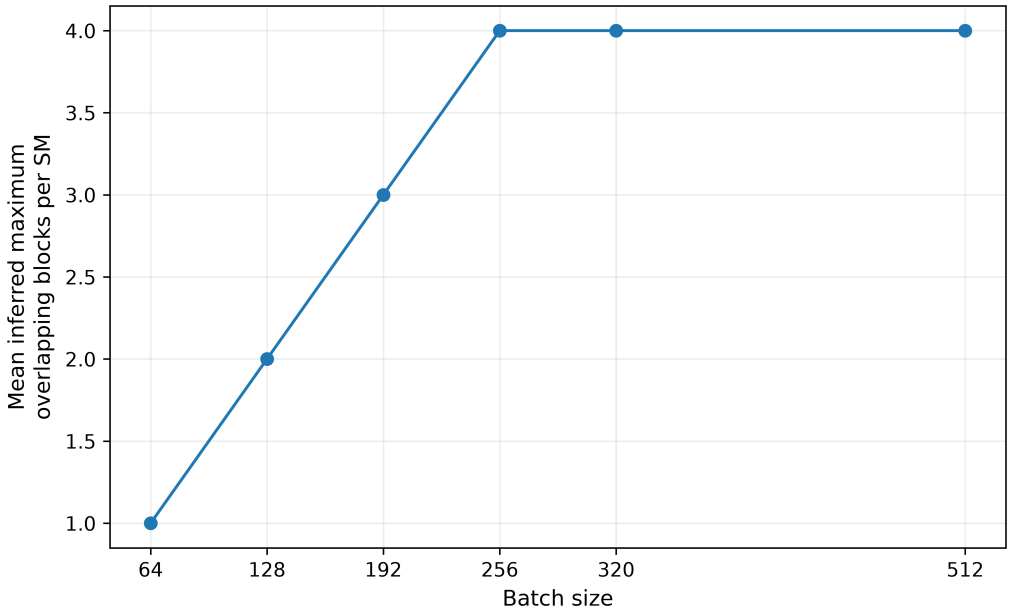


Figure A2. Mean inferred maximum overlapping blocks per SM in the instrumented layered-decoder diagnostic. Values use within-SM `%globaltimer` comparisons only. Cross-SM timing was not validated, and the trace changed P50; the figure is not an achieved-occupancy or device-wide concurrency measurement.

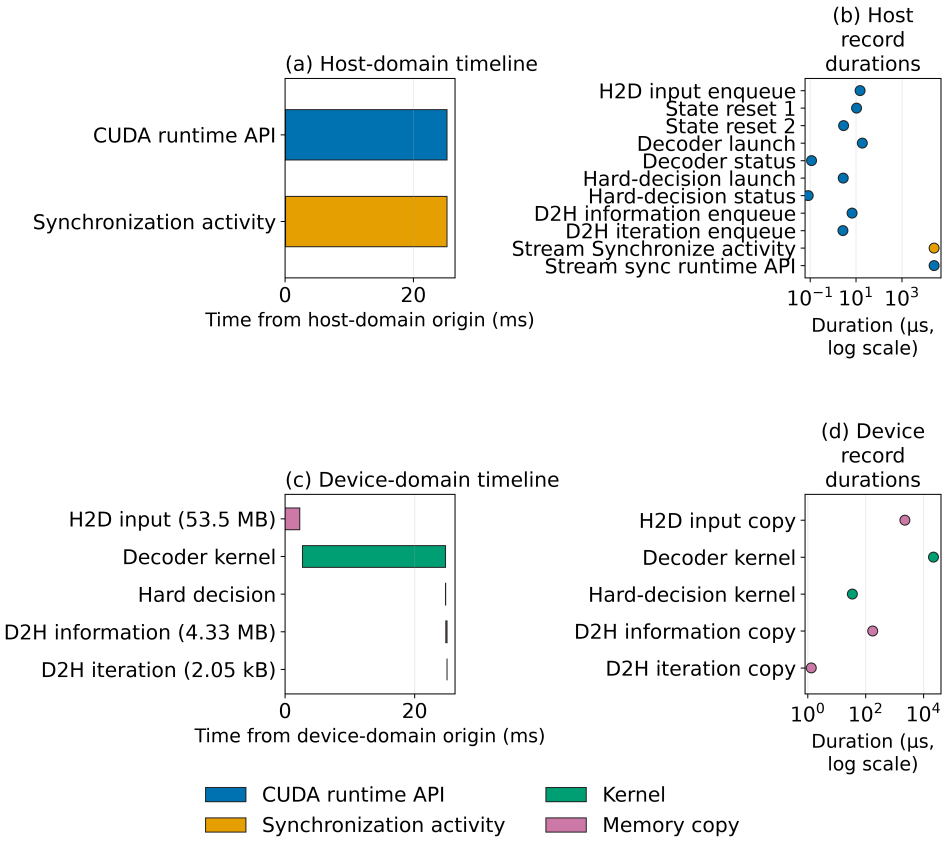


Figure A3. CUPTI Activity records for the first retained traced trial of invocation 1 at batch 512 under the layered full boundary (external correlation identifier 901). Panels (**a**,**c**) show the host- and device-domain timelines, respectively, using independent timestamp origins. Panels (**b**,**d**) report the duration of every corresponding record on logarithmic axes. Blue, orange, green, and purple identify CUDA runtime API, synchronization activity, kernel, and memory-copy records; lane and operation labels provide the same distinctions without reliance on color. The `cudaStreamSynchronize` runtime-API interval and the stream-synchronization activity interval cover the same blocking period and must not be added. CUPTI-traced trials are excluded from the unprofiled latency sample.

References

1. 5G Americas. *New Services & Applications with 5G Ultra-Reliable Low Latency Communications*; Technical Report; 5G Americas: Bellevue, WA, USA, 2018. Available online: https://www.5gamericas.org/new-services-applications-with-5g-ultra-reliable-low-latency-communications/ (accessed on 27 July 2026).
2. Liang, B.; Gregory, M.A.; Li, S. Latency Analysis for Mobile Cellular Network uRLLC Services. *J. Telecommun. Digit. Econ.* **2022**, *10*, 39–57. [CrossRef]
3. Gallager, R.G. Low-Density Parity-Check Codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [CrossRef]
4. 3rd Generation Partnership Project (3GPP). NR; Multiplexing and Channel Coding. 3GPP Technical Specification 38.212, European Telecommunications Standards Institute, 2026. Release 18; ETSI TS 138 212 V18.8.0. Available online: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/18.08.00_60/ts_138212v180800p.pdf (accessed on 27 July 2026).
5. Aronov, A.; Kazakevich, L.; Mack, J.; Schreider, F.; Newton, S. 5G NR LDPC Decoding Performance Comparison Between GPU & FPGA Platforms. In *Proceedings of the 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*; IEEE: New York, NY, USA, 2019; pp. 1–6. [CrossRef]
6. Li, R.; Zhou, X.; Pan, H.; Su, H.; Dou, Y. A High-Throughput LDPC Decoder Based on GPUs for 5G New Radio. In *Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC)*; IEEE: New York, NY, USA, 2020; pp. 1–7. [CrossRef]
7. Ling, J.; Cautereels, P. Fast LDPC GPU Decoder for Cloud RAN. *IEEE Embed. Syst. Lett.* **2021**, *13*, 170–173. [CrossRef]
8. Tarver, C.; Tonnemacher, M.; Chen, H.; Zhang, J.; Cavallaro, J.R. GPU-Based, LDPC Decoding for 5G and Beyond. *IEEE Open J. Circuits Syst.* **2021**, *2*, 278–290. [CrossRef]
9. Dai, J.; Yin, H.; Lv, Y.; Xu, W.; Yang, Z. Multi-Gbps LDPC Decoder on GPU Devices. *Electronics* **2022**, *11*, 3447. [CrossRef]
10. Liu, Z.; Zhao, L. High-Throughput Software LDPC Decoder on GPU. *EURASIP J. Adv. Signal Process.* **2024**, *2024*, 91. [CrossRef]

11. Qi, Z.; Yao, Y.; Li, Y.; Tung, C.H.; Zheng, J.; Zhuo, D.; Chen, T. DecodeX: Exploring and Benchmarking of LDPC Decoding across CPU, GPU, and ASIC Platforms. In *HotMobile '26: Proceedings of the 27th International Workshop on Mobile Computing Systems and Applications, Atlanta, GA, USA, 25–26 February 2026*; Association for Computing Machinery: New York, NY, USA, 2026; pp. 127–132. [CrossRef]
12. Kato, S.; Lakshmanan, K.; Rajkumar, R.; Ishikawa, Y. TimeGraph: GPU Scheduling for Real-Time Multi-Tasking Environments. In Proceedings of the 2011 USENIX Annual Technical Conference (USENIX ATC 11), Portland, OR, USA, 15–17 June 2011; pp. 17–30. Available online: https://www.usenix.org/conference/usenixatc11/timegraph-gpu-scheduling-real-time-multi-tasking-environments (accessed on 29 August 2026).
13. Elliott, G.A.; Ward, B.C.; Anderson, J.H. GPUSync: A Framework for Real-Time GPU Management. In *Proceedings of the 2013 IEEE 34th Real-Time Systems Symposium, IEEE Computer Society, Vancouver, BC, Canada, 3–6 December 2013*; IEEE: New York, NY, USA, 2013; pp. 33–44. [CrossRef]
14. Gheibi-Fetrat, A.; Safari, S.; Hessabi, S.; Sarbazi-Azad, H. *RTGPU: Real-Time Computing with Graphics Processing Units: Architectures, Challenges, and Solutions for Time-Critical GPU Systems*; Synthesis Lectures on Computer Architecture; Springer Nature: Cham, Switzerland, 2026. [CrossRef]
15. Richardson, T.J.; Shokrollahi, M.A.; Urbanke, R.L. Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 619–637. [CrossRef]
16. Chen, J.; Dholakia, A.; Eleftheriou, E.; Fossorier, M.P.C.; Hu, X.Y. Reduced-Complexity Decoding of LDPC Codes. *IEEE Trans. Commun.* **2005**, *53*, 1288–1299. [CrossRef]
17. Hocevar, D.E. A Reduced Complexity Decoder Architecture via Layered Decoding of LDPC Codes. In *Proceedings of the IEEE Workshop on Signal Processing Systems (SIPS), Austin, TX, USA*; IEEE: New York, NY, USA, 2004; pp. 107–112. [CrossRef]
18. Kelkar, A.; Dick, C. NVIDIA Aerial GPU Hosted AI-on-5G. In *Proceedings of the 2021 IEEE 4th 5G World Forum (5GWF)*; IEEE: New York, NY, USA, 2021; pp. 64–69. [CrossRef]
19. Notari, E.F.; Travassos, X.L. 5G New Radio Open Radio Access Network Implementation in Brazil: Review and Cost Assessment. *Telecom* **2025**, *6*, 24. [CrossRef]
20. Arabian, F.; Shoushtari, M. A Comparative Study of Waveforms Across Mobile Cellular Generations: From 0G to 6G and Beyond. *Telecom* **2025**, *6*, 67. [CrossRef]
21. Lin, X.; Kundu, L.; Cammerer, S.; Huang, Y.; Dick, C.; Santhosam, C.; Gadiyar, R.; Wiesmayr, R.; Studer, C. AI-Native 6G: Empowering Intelligent RAN with Accelerated Compute. *IEEE Wirel. Commun.* **2025**, *32*, 11–14. [CrossRef]
22. Coles, S. *An Introduction to Statistical Modeling of Extreme Values*; Springer Series in Statistics; Springer: London, UK, 2001. [CrossRef]
23. Künsch, H.R. The Jackknife and the Bootstrap for General Stationary Observations. *Ann. Stat.* **1989**, *17*, 1217–1241. [CrossRef]
24. Politis, D.N.; Romano, J.P. A Circular Block-Resampling Procedure for Stationary Data. In *Exploring the Limits of Bootstrap*; LePage, R., Billard, L., Eds.; Wiley Series in Probability and Mathematical Statistics; John Wiley & Sons: New York, NY, USA, 1992; pp. 263–270.
25. Lahiri, S.N. *Resampling Methods for Dependent Data*; Springer Series in Statistics; Springer: New York, NY, USA, 2003. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.