POST-BOUNDARY BRIDGE: MUST LOCAL ATTENTION GO GLOBAL BETWEEN GLOBAL LAYERS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Hybrid Transformers interleave local attention with periodic Full layers to cut long-context cost. Recent work on sparse and local attention reveals a gap between theoretical cross-layer reach and effective information propagation. This raises a key question: when periodic Full layers already provide global communication, how should local computation be allocated? We argue that local layers need not maximize cross-layer relay range; they can instead emphasize dense local modeling and timely nearby exchange. Post-Boundary Bridge (PBB) realizes this view by preserving dense causal attention within blocks while adding selective cross-block connections under a single softmax, with no extra parameters or routing. In controlled 200M experiments, PBB+Full matches SWA+Full in perplexity and improves the main source-retrieval probe; score-matched controls show that allocation, not only budget size, matters. In larger-scale transfer, PBB+Full stays close to Full attention in perplexity on 350M dense and 2B MoE models, and improves exact entity–attribute retrieval by 14.02 percentage points on the 2B MoE model. At 512K context, it achieves 3.32× prefill and 2.11× cached-decoding throughput over Full attention. These results support allocating local computation to targeted nearby exchange rather than maximum relay range.

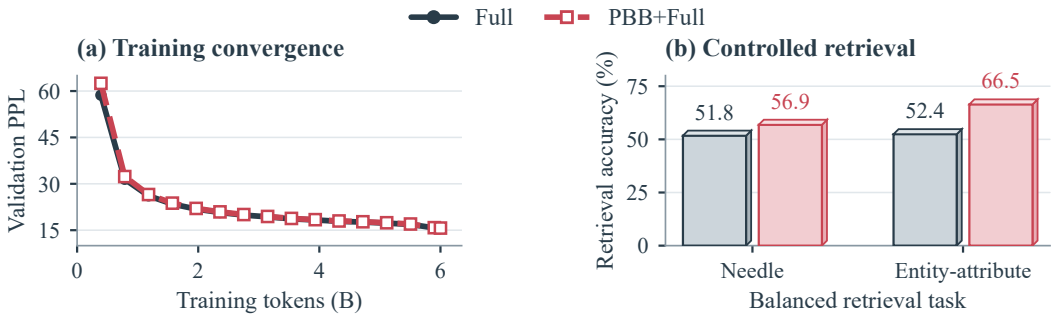

Figure 1: **Near-Full quality with stronger retrieval on 2B MoE.** (a) All 16 logged validation points. (b) Literal needle and exact entity–attribute retrieval, with 6,720 balanced examples per task and 25% chance accuracy. Both evaluations use 8K context.

1 INTRODUCTION

Full attention gives every token direct access to its entire causal prefix, but it incurs quadratic attention-score complexity during prefill and an ever-growing KV history during decoding. Sparse attention reduces these costs by restricting which past tokens each query can access. A widely used example is sliding-window attention (SWA), where each token attends only to a recent window of w positions, reducing attention-score complexity from $O(n^2)$ to $O(nw)$, or linear in sequence length for fixed w. The trade-off is that a local SWA layer no longer provides direct access to the full prefix: information beyond the window must propagate through cross-layer relay. A common compromise is

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

therefore to interleave SWA with periodic Full layers, as in Gemma 2 (Gemma Team et al., 2024), allowing most layers to remain local while periodically restoring global communication.

This hybrid setting raises a further question about the role of local attention. PowerAttention studies sparse attention through cross-layer receptive fields, making receptive-field expansion an explicit design objective and showing how sparse edge allocation governs structural reach across layers (Chen et al., 2025). More recently, DeepSeek-V4.1-Flash cites PowerAttention's observation that the effective receptive field of SWA can be smaller than its theoretical range, and uses it to motivate SWA Bounded Replay, which reconstructs only recent local states while retaining cached global information, with little reported quality loss (DeepSeek-AI, 2026). These observations raise a broader architectural question: once periodic Full layers already provide global communication, how much cross-layer relay range do the intervening local layers actually need? We argue that local layers need not maximize this relay range. Instead, local computation can emphasize dense local modeling and timely nearby exchange, while periodic Full layers carry long-range communication.

Post-Boundary Bridge (PBB) is a concrete realization of this view. Fixed block attention provides dense local modeling at low cost, but its boundaries sever even adjacent-token dependencies, and stacking identical blocks cannot repair this barrier. PBB preserves the dense causal prefix within each block while selectively restoring direct communication across block boundaries: targets in the first h positions of a block additionally read the final h positions of the preceding block, with all sources normalized under a single softmax. The block size b and bridge width h are independent design parameters; PBB does not require $b = 2h$. Our main experiments use the symmetric setting $b = 2h$ for simplicity. PBB requires no additional parameters, routing, or branch scales. As a result, it retains dense within-block modeling while introducing only limited, targeted cross-block exchange rather than broadly extending local relay range. Under our default $b = 128$, $h = 64$, 8K setting, PBB preserves SWA-128's one-layer exchange for the repaired boundary pairs while using 24.4% fewer local attention scores than SWA-128.

We first test this view in a controlled 200M setting. When all 20 layers are local, the broader local support of SWA-128 outperforms PBB in both perplexity and the main controlled source-retrieval probe. Once periodic Full layers are introduced, however, this advantage does not persist: the 3:1 PBB-to-Full hybrid (PBB+Full) and SWA+Full achieve comparable perplexity, while PBB+Full performs better on the main controlled source-retrieval probe. Score-matched controls, which hold the number of local attention scores fixed while changing where those scores are allocated, further show that the outcome depends on how the local attention budget is distributed rather than only on its size. We then examine transfer at larger scale. On a 350M dense model and a 2B MoE model, PBB+Full remains close to Full attention in perplexity while retaining stronger controlled source retrieval. On the 2B MoE model, exact entity–attribute retrieval improves by 14.02 percentage points over Full attention. These results show that PBB's effectiveness is not confined to the controlled 200M setting.

The same regular support also translates into practical systems benefits. Flash-PBB is an exact cached-decoding implementation of PBB, exploiting its structured support to fuse KV append with attention and execute the required local interactions efficiently. Relative to Full attention, PBB+Full reaches $3.32\times$ the prefill throughput and $2.11\times$ the cached-decoding throughput of Full attention at 512K context. PBB also provides additional acceleration over the established SWA baseline: at batch 64, the Flash-PBB local attention-and-cache-maintenance core is $8.91\times$ faster than the tested Flash-SWA backend, yielding 7.1% higher end-to-end decoding throughput for PBB+Full than SWA+Full.

Importantly, the fixed-ring property is not specific to $b = 2h$: for general $0 < h \leq b$, PBB's decode support is a contiguous interval of length at most $M = \max(b, 2h)$, admitting an exact relocation-free M-slot ring cache. Together, the modeling and systems results support the view that local computation between periodic global layers can be allocated toward targeted exchange rather than maximum relay range.

Our contributions are:

(i) An architectural view that separates global communication from timely local exchange, arguing that local layers between periodic Full layers need not maximize cross-layer relay range, together with systematic controlled and cross-scale empirical evidence supporting this view.

(ii) PBB, a general boundary-exchange formulation with independently parameterized block size and bridge width, providing a concrete realization of this view while preserving dense within-block modeling and selectively restoring targeted cross-boundary exchange.

(iii) Flash-PBB, an exact implementation that converts PBB's regular support into efficient cached decoding and delivers measured efficiency gains over both Full attention and the tested SWA backend.

2 Local Exchange Between Global Layers

2.1 From relay range to timely exchange

Static local attention isolates edge allocation through fixed graphs and predictable caches. SWA serves Mistral and Gemma 2 (Jiang et al., 2023; Gemma Team et al., 2024) with mature kernels (Dao et al., 2022; Dao, 2024); its relay expands structural reach (Chen et al., 2025). We ask what local edges should accomplish when Full layers already supply global communication.

We distinguish *input coverage*, supplied by a Full layer, from *timely exchange of newly computed features*, governed by subsequent local edges. A feature computed at $p-1$, the last position before a block boundary p, cannot reach p under Block+Full until the next Full layer restores the missing edge; one direct edge from $p-1$ to p delivers it at the next layer. Coverage alone therefore does not make a local graph timely. The graph statements below formalize both properties and motivate block prefixes with short boundary paths; learned quality tests this allocation.

2.2 Structural dependency closure

For a length-n sequence, positions are integers in $\{0,\dots,n-1\}$. Define the causal prefix $\mathcal{C}_t = \{s : 0 \le s \le t\}$ and

$$p(t) = b\lfloor t/b \rfloor, \qquad r(t) = t - p(t), \qquad B_j = \{jb,\dots,\min((j+1)b, n) - 1\}. \tag{1}$$

Here b is block size, $p(t)$ is the block start, and $r(t)$ is the target phase. Let $\mathcal{A}_\ell(t) \subseteq \mathcal{C}_t$ be the sources read at t in layer ℓ. With positionwise non-attention operations, structural dependency closure is

$$\mathcal{R}_0(t) = \{t\}, \qquad \mathcal{R}_{\ell+1}(t) = \mathcal{R}_\ell(t) \cup \bigcup_{s \in \mathcal{A}_{\ell+1}(t)} \mathcal{R}_\ell(s). \tag{2}$$

The first term includes the residual path. Agreement on $\mathcal{R}_\ell(t)$ suffices for agreement of the layer-ℓ representation at t. Learned parameters need not use every allowed path, but cannot create a path outside this closure.

Proposition 1 (Fixed-block barrier) *If every layer uses the same partition and $\mathcal{A}_{\mathrm{Block}}(t) = B_{\lfloor t/b \rfloor} \cap \mathcal{C}_t$, then $\mathcal{R}_L(t) \subseteq \{p(t),\dots,t\}$ for every depth L. At a block boundary p, position $p-1$ cannot influence position p.*

For a constructed K-way boundary-copy distribution, let a uniform label be revealed only at $p-1$, independently of the accessible target prefix. A fixed-block readout at p then has top-1 accuracy at most $1/K$ and expected cross-entropy at least $\log K$. Adding the direct edge removes the structural obstruction. Appendix A gives the proofs and independence conditions.

Proposition 2 (Sliding-window relay) *For $\mathcal{A}_{\mathrm{SWA}}(t) = \{\max(0, t-w+1),\dots,t\}$, where window size w includes the current position, L identical local layers give*

$$\mathcal{R}_L(t) = \{\max(0, t - L(w-1)),\dots,t\}. \tag{3}$$

Proposition 3 (Full-layer reachability persistence) *A Full layer makes the closure global, and later causal layers preserve it (Appendix A).*

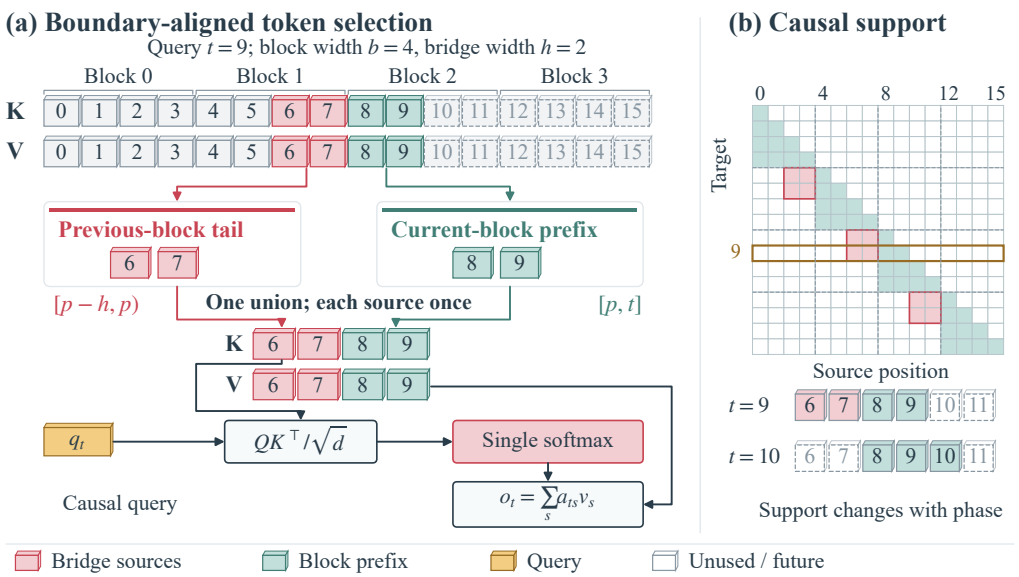

Figure 2: **Post-Boundary Bridge attention.** (a) The preceding block's tail and current causal prefix enter one softmax, with each source counted once. (b) The same sources in the causal mask. For $b = 4, h = 2$, query $t = 9$ reads 6–9; at $t = 10$, the bridge is inactive and the support becomes 8–10. Token grouping is conceptual, not a required KV staging operation.

Why a boundary bridge still matters after Full. Let $\tau_\ell(s, t)$ be the minimum number of additional layers needed for a perturbation introduced at position s after layer ℓ to reach t, with other layer-ℓ states held fixed. For PBB's repaired pairs, $s \in [p-h, p-1]$ and $t \in [p, p+h-1]$, any $0 < h \le b$, SWA window $w \ge 2h$, and next Full layer f,

$$\tau_\ell^{\text{Block+Full}}(s, t) = f - \ell, \qquad \tau_\ell^{\text{PBB+Full}}(s, t) = \tau_\ell^{\text{SWA+Full}}(s, t) = 1. \tag{4}$$

For the 3:1 schedule, PBB replaces a wait of up to four layers with next-layer exchange. An earlier Full layer has not broadcast this fresh feature (Appendix B). This separates two design objectives: achieving global input coverage and transmitting nearby features promptly. Periodic Full layers handle the first; a limited set of direct boundary edges can handle the second.

3 Post-Boundary Bridge Attention

Choose integer block size b and bridge width h independently, subject to $0 < h \le b$. For target $t = p + r$, PBB retains the dense causal block prefix and adds the preceding block's final h sources for the first h target phases:

$$\mathcal{A}_{\text{PBB}}(t) = \{p, \dots, t\} \cup \begin{cases} \{p-h, \dots, p-1\}, & p > 0,\ 0 \le r < h, \\ \varnothing, & \text{otherwise.} \end{cases} \tag{5}$$

Figure 2 traces the selected tokens through a single softmax. Here b controls within-block modeling and h the $h \times h$ boundary connection; $b = 2h$ is not required. Experiments fix $b = 128$, $h = 64$, and SWA window $w = 128$. These boundary edges are added to the causal block prefix without introducing new projections, gates, or branch scales. Every fourth layer is Full (layered graph in Appendix B; mask comparison in Figure 6).

For query q_t, key k_s, value v_s, and head dimension d_h, define $z_{ts} = q_t^\top k_s / \sqrt{d_h}$. The output before the shared output projection is

$$o_t = \frac{\sum_{s \in \mathcal{A}_{\text{PBB}}(t)} e^{z_{ts}} v_s}{\sum_{s \in \mathcal{A}_{\text{PBB}}(t)} e^{z_{ts}}}. \tag{6}$$

All main PBB models use this single softmax. A fused Triton kernel processes disjoint block and bridge tiles with log-sum-exp rescaling (Appendix A), native GQA, and backward propagation.

Flash-PBB specializes cached decode to the same support. Neither repeats KV heads or materializes a dense mask. Historical separately normalized branches are distinguished in Appendix G.

Flash-PBB: exact attention in a fixed ring. For any $0 < h \leq b$, PBB reads a contiguous interval $[a(t), t]$: $a(t) = p - h$ in noninitial repaired phases and $a(t) = p$ otherwise. This endpoint never moves backward, and the interval contains at most $M = \max(b, 2h)$ positions. Storing token s at slot $s \bmod M$ therefore preserves all needed KV entries without relocating retained entries. Appendix F.1 proves the general result and illustrates the updates. Our $b = 128, h = 64$ setting gives $M = 128$. Flash-PBB fuses KV append and attention in a 128-column tile with native GQA, reading wrapped sources directly without staging gathers or rollover copies.

Coverage and arithmetic. For noninitial blocks, direct coverage at source distance $d = t - s$ and target phase r is

$$C_{\text{PBB}}(d, r) = \mathbf{1}\{0 \leq d \leq r\} + \mathbf{1}\{r < h,\ r < d \leq r + h\}, \tag{7}$$

where $\mathbf{1}$ denotes an indicator. For a complete noninitial block, the average support size is $(b+1)/2 + h^2/b$. At $w = b = 2h$, PBB is a subset of SWA and each added local layer reaches one more preceding half-block (Appendix A). Thus PBB retains local relay while allocating fewer scores to it.

At 8K, PBB uses 24.4% fewer local attention scores than SWA-128 while retaining every within-block score. Exact counts appear in Appendix A. The 20-layer hybrid uses 26.76% of all-Full's attention scores; with fixed local widths this fraction approaches one quarter as context grows. Only five Full layers retain the unbounded KV history. We first examine modeling behavior, then whether the support yields execution gains.

4 Experimental Setup

We test local allocation at 200M and transfer against Full in wider dense/MoE models. Qwen2.5-like decoders (Qwen Team, 2025) train on filtered FineWeb-Edu (Penedo et al., 2024) with the Qwen2.5 tokenizer, rotary positions, root-mean-square normalization (RMSNorm), tied embeddings, 20 layers, two KV heads, and 8K sequences. The MoE has 64 routed experts, top-4 routing, and a shared expert. "+Full" denotes the 3:1 hybrid; other local names denote 20 local layers.

Pairs match parameterization, data order, exposure, optimizer, and schedule. Each update contains 393,216 tokens; 200M/350M/MoE budgets are 4.109B/7.078B/6.000B, approximately 20 tokens per dense or active parameter. AdamW uses $3 \times 10^{-4}/3 \times 10^{-5}$ peak/minimum rates, 5.56% warmup, a plateau to 88.89%, then cosine decay. We compare final checkpoints and retain 5,000-step and best-validation checkpoints (Appendix C).

Quality metrics. Quality evaluation uses trained checkpoints at 8K context; longer-context benchmarks measure computational efficiency using initialized weights and synthetic inputs. We measure held-out PPL, negative log-likelihood (NLL) by block phase, and zero-shot likelihood tasks using the LM Evaluation Harness (Gao et al., 2023). *LM-6* is the unweighted mean accuracy of LAMBADA, HellaSwag, PIQA, ARC-Easy, SciQ, and Winogrande. BLiMP (Warstadt et al., 2020) and ARC-Challenge are reported separately.

The probe retrieves a supplied token from four one-token candidates. Matched groups rotate all labels to separate source use from candidate priors. Fifteen distances, seven readout phases, and 16 groups per cell yield 1,680 groups/6,720 examples (25% chance accuracy). *Probe* is candidate accuracy; *source gain* is the correct label's mean log-probability increase under its supporting versus counterfactual sources, in nats. Neither metric is structural reachability. Prompts were frozen after Full-only calibration (Appendix D). Literal needle and exact entity–attribute retrieval each add 6,720 balanced examples on a separate distance grid (Appendix D.7).

5 Allocating Local Computation Between Global Layers

5.1 Preserving quality with fewer Full layers

Run	Architecture	PPL	Probe (%)	Source gain	LM-6 (%)
A	Full	21.52	69.84	1.645	45.78
A	PBB+Full	21.27	81.31	2.543	46.13
B	Full	21.38	68.26	1.811	46.55
B	PBB+Full	21.22	80.07	2.405	45.98

Table 1: **Full versus PBB+Full at matched training exposure.** Both architectures use GQA and 4.109B training tokens. PBB+Full has 15 local and five Full layers. Probe and LM-6 are accuracies in percent; source gain is in nats. Run identifiers are defined in Appendix C.

Replacing 15 of 20 Full layers with PBB improves retrieval and lowers PPL in both matched 200M comparisons (Table 1), using 26.76% of Full's attention scores (Table 4). Probe gains of 11.47/11.82 points have paired group-bootstrap 95% intervals $[10.46, 12.50]$/$[10.74, 12.95]$; source gain rises by 0.898/0.594 nats. The advantage persists at 5k and 10k updates (Appendix D.5) and on separate needle and exact entity–attribute tasks (Appendix D.7). BLiMP improves in both runs, ARC-Challenge improves, and LM-6 changes are mixed (Appendix D.3).

5.2 The preferred local graph depends on the schedule

Architecture	PPL ↓	Probe ↑	Source gain ↑	LM-6 ↑
Full	21.52	69.84	1.645	45.78
Block	30.42	35.83	0.315	44.85
SWA	23.31	54.27	0.911	**46.13**
PBB	24.05	46.80	0.647	44.95
Random-Edge	25.63	42.02	0.576	45.27
SWA+Full	21.29	76.50	2.305	45.82
Random-Edge+Full	21.61	75.68	1.934	46.02
WideBlock-192+Full	21.67	80.57	2.336	45.92
PBB+Full	**21.27**	**81.31**	**2.543**	**46.13**

Table 2: **Local-family and equal-arithmetic controls at 200M.** Pure PBB/Random-Edge and hybrid PBB/WideBlock-192 match their respective attention arithmetic and non-attention modules. Block's aggregate includes reachable and unreachable groups. Metric definitions match Table 1.

At the default geometry, SWA-128 contains every PBB edge, allowing a comparison with broader local support. In run A with all 20 layers local, SWA has lower PPL (23.31 versus 24.05) and higher probe accuracy (54.27% versus 46.80%) than PBB. With five Full layers, the probe ordering reverses despite comparable PPL: PBB+Full reaches 81.31% versus 76.50% for SWA+Full. Its paired advantage is 4.81 points, interval $[3.96, 5.62]$; source gain rises by 0.238 nats, interval $[0.203, 0.273]$. In run B, which lacks local-only counterparts, SWA+Full has slightly lower PPL (21.214 versus 21.222), but PBB+Full retains higher probe accuracy (80.07% versus 77.62%) and source gain (2.405 versus 2.053).

This contrast answers the allocation question: the stronger standalone local graph need not be the stronger complement to Full layers (Figure 3). PBB retains competitive PPL and improves the main probe with fewer local scores. On exact entity–attribute retrieval, accuracy favors SWA+Full in run A and PBB+Full in run B; source gain favors PBB+Full in both (Appendix D.7).

5.3 Edge placement matters at fixed arithmetic

Random-Edge keeps the block prefix and adds h^2 edges per adjacent block pair via fixed random source/target permutations. It matches pure PBB's added-degree histogram, normalization, and

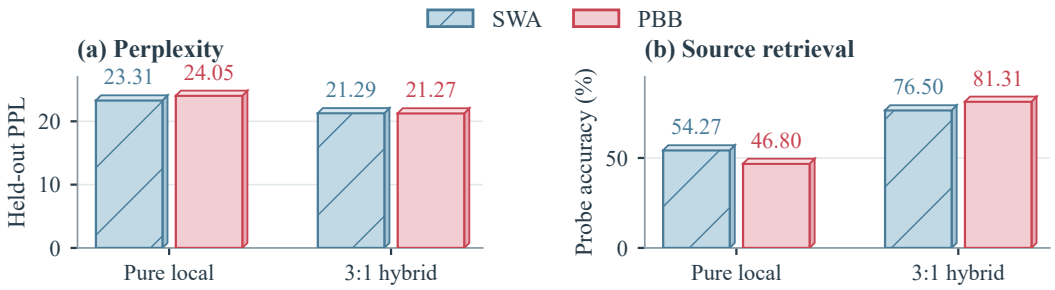

Figure 3: **The local graph's role depends on the schedule.** Grouped bars show 200M run A (seed 13). Pure SWA exceeds pure PBB; PBB+Full has comparable PPL and 4.81-point higher probe accuracy than SWA+Full.

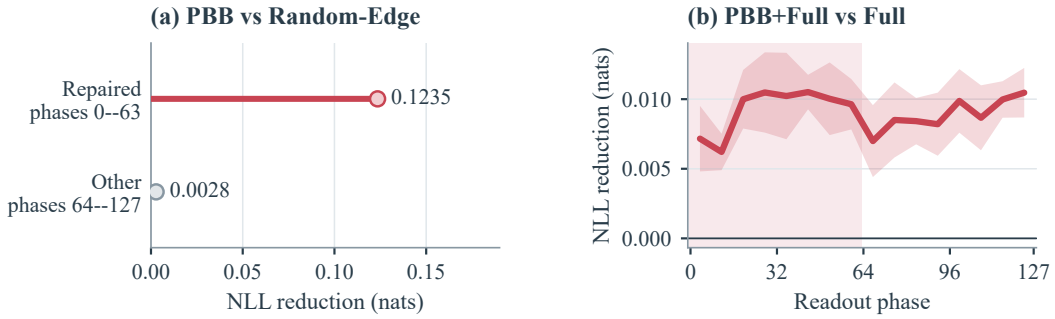

Figure 4: **Boundary-focused repair in natural text.** Positive denotes lower PBB NLL. (a) Pure PBB versus Random-Edge, means over each half-block. (b) Full minus PBB+Full: eight-phase means across two paired runs, with their observed range, not a confidence interval. Pale shading marks repaired phases.

parameters, but has higher PPL (25.63 versus 24.05) and lower probe accuracy (42.02% versus 46.80%). With Full layers, Random-Edge+Full reaches 21.61 PPL and 75.68% probe accuracy, versus PBB+Full's 21.27 and 81.31%. On the separate three-task battery it remains 1.73 points and 0.401 nats below PBB+Full in aggregate; exact entity–attribute accuracy alone reverses (+1.40 points for Random-Edge+Full), while PBB+Full retains higher source gain (Table 12). Pure Block confirms the barrier: exactly 25% accuracy and source gain below 10^{-8} on 1,280 unreachable groups.

WideBlock-192 exactly matches PBB's 8K attention-score arithmetic, including the final partial block, but uses wider block prefixes; both hybrids retain five Full layers. PPL is 21.67 versus PBB's 21.27. Probe accuracy is close (80.57% versus 81.31%; paired interval $[-0.13, 1.62]$ points), while source gain favors PBB by 0.207 nats, interval $[0.162, 0.252]$. Needle accuracy is also close (55.95% versus 55.80%). On exact entity–attribute retrieval, PBB improves accuracy from 54.05% to 58.07% and source gain from 1.332 to 1.975 (Appendix D.7). These controls distinguish edge allocation, not runtime cost.

5.4 Boundary exchange improves natural-text prediction

After excluding readouts with an end-of-document token in the preceding 128-position history, pure PBB reduces NLL relative to Random-Edge by 0.1235 nats over repaired phases, versus 0.0028 over the other half-block (Figure 4). The boundary reduction is 1.503 nats; excluding phase zero still leaves a 0.1016-nat reduction over phases 1–63. For PBB+Full versus Full, the mean across paired runs improves clean NLL at every block phase; run A improves 128/128 phases and run B 127/128 (Appendix C).

On 128 matched spans, truncating 8K context to 128 tokens raises suffix NLL by 0.2002/0.1999/0.2136 for PBB+Full/SWA+Full/Full. PBB+Full has lower absolute loss at all

four lengths and a similar truncation penalty to SWA+Full: difference 0.0003 nats, interval $[-0.0098, 0.0103]$. Identical suffixes lie outside repaired phases (Appendix D.9).

6 Transfer Across Model Families

Model	Architecture	PPL	Probe (%)	Source gain	LM-6 (%)
350M	Full	**17.13**	73.62	1.983	49.19
350M	PBB+Full	17.30	**74.79**	**2.339**	**49.55**
2B MoE	Full	**14.25**	75.62	2.326	52.02
2B MoE	PBB+Full	14.35	**77.77**	**2.714**	**52.94**

Table 3: **Stronger source retrieval with near-Full quality across model families.** Matched final pairs; MoE has 2.07B stored/300M active parameters. Probe uses 6,720 rows; dense/MoE PPL uses 8M/1M held-out targets.

We test transfer across model width and feed-forward organization with $b = 128$, $h = 64$, and the same 3:1 schedule. Both families retain near-Full PPL with stronger source use (Table 3). At 350M, probe accuracy, source gain, and LM-6 improve within 1.02% of Full's PPL. Needle and exact entity–attribute retrieval gain 0.80 and 2.68 points, with higher source gain on both (Appendix D.7).

Larger retrieval gains on the 2B MoE. After 6B training tokens, the MoE hybrid raises needle accuracy from 51.76% to 56.88% (+5.12 points) and exact entity–attribute retrieval from 52.44% to 66.46% (+14.02). Source gains rise from 1.347 to 1.683 and 2.061 to 2.917 nats (Appendix D.7), pairing higher accuracy with stronger source sensitivity. The original probe improves from 75.62% to 77.77%, including 45.54% versus 40.77% over distances 2,048–6,144 (Appendix D.6).

Preserving modeling quality and improving downstream scores. The PPL gap to Full is 0.68%. LM-6 gains 0.925 points, with task-stratified paired 95% interval $[0.194, 1.655]$ (Appendix D.3). All six task point estimates, BLiMP, and ARC-Challenge favor the hybrid; ARC-Easy and ARC-Challenge gain 1.68 and 1.45 points. Stronger retrieval thus coexists with near-Full LM quality and improved downstream performance.

7 Exact and Efficient Execution

PBB's regular support connects local allocation to execution: Flash-PBB reads a bounded interval and fuses KV append with attention. We evaluate its 128-slot specialization at $b = 128, h = 64$ against complete hybrids and local attention-and-cache-maintenance cores on RTX 4090, matching GQA (Ainslie et al., 2023), BF16, and expert backends. Exact-union checks cover empty bridges, multi-block gradients, and native GQA (Appendix F). Flash-SWA uses FlashAttention 2.8.3.post1's `flash_attn_with_kvcache`, a 128-token causal window, and a 256-position local cache.

At batch 64, the Flash-PBB local core is $8.91\times$ faster than Flash-SWA, yielding 7.1% higher PBB+Full throughput than SWA+Full (6,942 versus 6,484 tokens/s; Figure 5). Beyond batch 8, Flash-SWA time rises sharply while Flash-PBB stays below 5 μs per batched step. Flash-PBB launches two blocks per sequence versus twelve and allocates 128 local KV positions versus 256. SWA admits a 128-entry ring; a contiguous-cache rerun retains its timing knee (Appendix F.2).

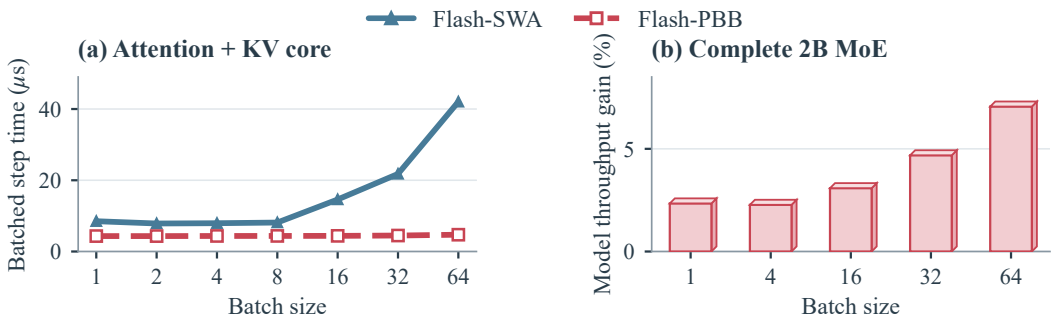

Figure 5: **Batched decode on RTX 4090 (BF16, 8K prefix).** (a) Attention and KV core; (b) complete 2B MoE. At batch 64, PBB gains $8.91\times$ core and 7.1% model throughput over SWA.

Resource profiling complements these timings. On an RTX 4060 Laptop GPU, the same 2B attention shape uses 212 versus 255 registers/thread and 40 versus 72 KiB dynamic shared memory/block. Global-load sector requests fall 86.0–92.7% across six phases (Appendix F.3); these count requests rather than DRAM bytes. The measurements link the implemented support and cache policy to lower resource use and faster execution.

Against all-Full attention, the 2B MoE hybrid reaches $3.32\times$ prefill and $2.11\times$ sequential cached-decode throughput at 512K. Persistent KV falls from 5.004 to 1.252 GiB because only five Full layers retain long history. Complete training-update throughput reaches $1.72\times$ Full at 64K. Context curves for all three architectures and capacity limits appear in Figure 11 and Appendix F; these lengths exceed the 8K quality evaluation.

8 Related Work

Structured sparsity spans Sparse Transformers, Longformer, BigBird, and ETC (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020; Ainslie et al., 2020). The 1D local attention used in image generation (Parmar et al., 2018) partitions queries into equal-width chunks; with equal query and memory widths $q = m = h$ and $b = 2h$, PBB's support coincides with it, while PBB's general family uses independent b, h and is not described by that uniform-chunk formula (Appendix A.3). We study this allocation between periodic Full layers, its learned behavior, and exact cached execution. Swin shifts partitions across layers (Liu et al., 2021); LongLoRA shifts during fine-tuning and uses Full attention at inference (Chen et al., 2024).

State-space hybrids, recurrence, and content-dependent routing offer other routes to long-range communication (Fu et al., 2023; Dai et al., 2019; Roy et al., 2021; Yuan et al., 2025; Lu et al., 2025); our axis is static edge allocation between Full layers. FlashAttention supplies IO-aware exact execution (Dao et al., 2022; Dao, 2024); KV-retention policies are another systems axis (Ge et al., 2024; Xiao et al., 2025). Long-context quality benchmarks test capability rather than execution capacity (Hsieh et al., 2024; Bai et al., 2024).

9 Conclusion

Local layers between periodic Full layers can prioritize dense block modeling and timely boundary exchange over maximum relay range. The stronger all-local pattern was not the stronger retrieval component under the same periodic-Full schedule. PBB realizes this targeted allocation with independent block and bridge widths. Controlled 200M comparisons and dense/MoE transfer retain near-Full quality with stronger controlled retrieval, while Flash-PBB turns the regular support into exact ring caching and faster decoding. The results support targeted local allocation as a joint modeling and execution design choice.

AI USE STATEMENT

This research was led by the author, who directed all AI-assisted work and performed the final correctness checks on the theoretical claims, the implementation, and the reported results. Generative AI tools assisted with: the theoretical development, including the formulation of the mathematical claims and their proofs; proposing and refining hypotheses; feedback on experimental design and methodology; implementing methods; screening, cleaning, and reformatting the training data; constructing the synthetic evaluation tasks and their generators, whose instances are produced deterministically by the released scripts; and interpreting results. They were not used to translate material; no qualitative or thematic analysis was performed, and no other task in the required-disclosure list applies to this work. Additionally, generative AI tools assisted with literature retrieval and discovery, drafting sections of the paper, and polishing the writing. AI tools served in an assistive role only: the author set the research direction, made the final research decisions, and takes responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendices A–F specify equations, proofs, architecture dimensions, training recipes, evaluation construction, and timing protocols. Tables and figures retain source-file and protocol provenance. The source package includes numerical plotting inputs and a standalone PBB kernel with multi-block forward/backward checks; compilation requires neither model weights nor evaluation datasets. This paper-rebuild and kernel-verification package is separate from the full training/evaluation artifact, maintained with the experiment records. Reproducing the exact filtered training stream additionally requires the retained-document selection; the screening protocol is described separately from the private lexicon.

ETHICS STATEMENT

These research language models are not instruction-tuned assistants. Web-derived training text is screened for explicit sexual promotion while retaining ordinary biological and health-related material; screening does not guarantee absence of harmful content. No human-subject study is conducted. Public benchmark text is evaluated locally, and server performance tests use synthetic inputs. Model outputs can contain factual errors and should not be treated as authoritative.

REFERENCES

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 268–284, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.19. URL https://aclanthology.org/2020.emnlp-main.19/.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. URL https://aclanthology.org/2024.acl-long.172/.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3639–3664, 2025. URL https://aclanthology.org/2025.acl-long.183/.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 995–1005, 2012.

Lida Chen, Dong Xu, Chenxin An, Xintao Wang, Yikai Zhang, Jiangjie Chen, Zujie Liang, Feng Wei, Jiaqing Liang, Yanghua Xiao, and Wei Wang. PowerAttention: Exponentially scaling of receptive fields for effective sparse attention. *arXiv preprint arXiv:2503.03588*, 2025. URL https://arxiv.org/abs/2503.03588.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. LongLoRA: Efficient fine-tuning of long-context large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.12307.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285/.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, 2024.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. Technical report, DeepSeek-AI, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash/blob/main/DeepSeek_V41_Tech_Report.pdf. Technical report; Sections 2.2, 3.2.2, and 6.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 1383–1392, 2018.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2212.14052.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2023. URL https://github.com/EleutherAI/lm-evaluation-harness. Software citation; evaluation configurations are specified in this paper.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.01801.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Leonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Aaron Gokaslan and Vanya Cohen. OpenWebText Corpus. https://skylion007.github.io/OpenWebTextCorpus/, 2019.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654. arXiv:2404.06654.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Gregory Kamradt. Needle in a haystack – pressure testing LLMs. *GitHub repository*, 2023. URL https://github.com/gkamradt/LLMTest_NeedleInAHaystack.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 388–395, 2004.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, Zhiqi Huang, Huan Yuan, Suting Xu, Xinran Xu, Guokun Lai, Yanru Chen, Huabin Zheng, Junjie Yan, Jianlin Su, Yuxin Wu, Neo Y. Zhang, Zhilin Yang, Xinyu Zhou, Mingxing Zhang, and Jiezhong Qiu. MoBA: Mixture of block attention for long-context LLMs. *arXiv preprint arXiv:2502.13189*, 2025. URL https://arxiv.org/abs/2502.13189.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel R. Bowman. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, 2022. doi: 10.18653/v1/2022.naacl-main.391. URL https://aclanthology.org/2022.naacl-main.391/.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4055–4064. PMLR, 2018. URL https://proceedings.mlr.press/v80/parmar18a.html.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115. arXiv:2412.15115v2.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI technical report, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. doi: 10.1162/tacl_a_00353. URL https://aclanthology.org/2021.tacl-1.4/.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020. doi: 10.1162/tacl_a_00321. URL https://aclanthology.org/2020.tacl-1.25/.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5c1ddd2e59df46fd2aa85c833b1b36ed-Abstract-Conference.html.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native Sparse Attention: Hardware-aligned and natively trainable sparse attention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23078–23097, Vienna, Austria, July 2025. doi: 10.18653/v1/2025.acl-long.1126. URL https://aclanthology.org/2025.acl-long.1126/.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17283–17297, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c8512d142a2d849725f31a9a7a361ab9-Abstract.html.

A Proofs and Exact Arithmetic Budgets

A.1 Structural dependency sufficiency

Fix a mask schedule, model parameters, and any auxiliary randomness shared between the compared forward passes. Input embeddings at position t depend on that token and fixed positional information, so agreement on $\mathcal{R}_0(t) = \{t\}$ implies identical initial states. Suppose the statement holds at layer ℓ. The residual input at t and every key/value source in $\mathcal{A}_{\ell+1}(t)$ depend only on the sets included in Equation (2). Agreement on their union makes these vectors identical. Query/key projections, normalization, attention, residual addition, and all positionwise operations then produce identical states at t. This proves sufficiency by induction. Parameter values may suppress an allowed dependency; they cannot create one outside the structural closure.

A.2 Fixed-block barrier and boundary-copy distribution

For the fixed-block mask, every source $s \in \mathcal{A}(t)$ satisfies $p(t) \le s \le t$. Inductively, $\mathcal{R}_\ell(s) \subseteq \{p(t), \dots, s\}$, and taking their union preserves the block-prefix bound. No depth introduces a source before $p(t)$.

To construct the task, choose a boundary $p > 0$ and draw Y uniformly from K distinct candidate labels. Encode Y only at position $p-1$. Draw the accessible block prefix and any fixed query independently of Y, and ask the readout at p to predict Y (or its corresponding next-token label). The model's readout is independent of Y by the barrier. Its top-1 success probability is therefore at most $1/K$. For any output distribution q on the candidate labels, with possible probability mass outside that set,

$$\mathbb{E}[-\log q(Y)] = -\frac{1}{K}\sum_{k=1}^{K}\log q(k) \ge -\log\left(\frac{1}{K}\sum_{k=1}^{K} q(k)\right) \ge \log K. \tag{8}$$

The first inequality is Jensen's inequality. The construction requires no redundant target-side cue or label-dependent accessible text. Natural retrieval tasks without this independence condition need not obey the same bound. Adding the direct causal edge $(p-1, p)$ removes this structural obstruction; it enables transmission, rather than guaranteeing that arbitrary trained parameters perform the task.

A.3 General PBB geometry and the chunked-local special case

Equation (5) defines PBB for integer $0 < h \le b$. The block width b fixes dense causal prefixes; h sets both the preceding tail length and the number of post-boundary targets receiving it. These choices are independent. For a noninitial block, the support's left endpoint is $p-h$ for $0 \le r < h$ and p for $h \le r < b$. Thus, for $h < b$, consecutive constant-endpoint query regions have alternating lengths h and $b-h$.

The 1D local construction in Parmar et al. (2018) partitions queries into equal-width chunks and associates each chunk with its own positions and preceding memory. Writing those widths as q, m, its causal support is $\{\max(0, q\lfloor t/q \rfloor - m), \dots, t\}$. At $b = 2h$, setting $q = m = h$ reproduces PBB's support. For $0 < h < b$ and $b \ne 2h$, PBB's alternating region lengths are unequal, so this uniform-chunk formula does not describe the general PBB family. For example, $b = 6, h = 2$ gives left endpoints 4, 6, and 10 at $t = 6, 8, 12$, with successive region lengths 2 and 4. The endpoint $h = b$ also has a chunk representation, with $q = m = b$. These correspondences describe particular supports; PBB's definition, boundary-latency argument, and general ring proof do not require them.

A.4 Sliding-window and PBB relay

For a sliding window, one layer expands an interval by at most $w-1$ positions to the left, with every intermediate source present. Induction yields Equation (3), including the finite-sequence truncation at position zero. A Full layer reads position s for every $s \le t$, and each $\mathcal{R}_\ell(s)$ contains s through its residual path. Its union is exactly $\mathcal{C}_t$. Later residual causal layers preserve this closure and cannot include future positions.



Figure 6: **Block modeling with direct boundary exchange.** Toy $b = w = 4$, $h = 2$ masks; dotted lines mark block boundaries. Small pointers mark query $t = 8$; the strips below show its sources: Block reads 8, SWA reads 5–8, and PBB reads 6–8. Hatched red cells add the bridge to the block prefix under one softmax.

For the default $b = 2h$, PBB can also be written using half-block chunks. Let $c(t) = \lfloor t/h \rfloor$. Then

$$\mathcal{A}_{\text{PBB}}(t) = \{\max(0, h(c(t)-1)), \ldots, t\}, \qquad \mathcal{R}_L^{\text{PBB}}(t) = \{\max(0, h(c(t)-L)), \ldots, t\} \quad (L \geq 1). \tag{9}$$

For an even chunk, the preceding chunk is the previous block's repaired tail; for an odd chunk it is already in the same block. This proves the first equality. Each additional layer admits one more preceding half-block chunk, proving the second. PBB therefore also relays information across layers. At the default widths, its relay expands more slowly than SWA's; periodic Full layers remove the need to obtain global structural coverage from local relay alone.

A.5 Direct coverage and single-softmax fusion

Write a source as $s = t - d$ with nonnegative backward distance d. The block support gives $d \leq r$. The bridge-exclusive support gives $r < d \leq r + h$, and exists only for $p > 0$ and $r < h$. These disjoint cases yield the coverage expression in Section 3 for any $0 < h \leq b$. In a complete noninitial block, the block prefixes contribute $b(b+1)/2$ scores and the bridge adds h^2, giving average support $(b+1)/2 + h^2/b$. Every PBB edge has $d \leq \max(b, 2h) - 1$; SWA with $w \geq \max(b, 2h)$ therefore contains it. In particular, this inclusion holds at the experimental $w = b = 2h$.

For two disjoint supports S_1, S_2, define $Z_i = \sum_{s \in S_i} e^{z_s}$ and $o_i = Z_i^{-1} \sum_{s \in S_i} e^{z_s} v_s$. Then $(Z_1 o_1 + Z_2 o_2)/(Z_1 + Z_2)$ is exactly attention on $S_1 \cup S_2$. Substituting $s_i = \log Z_i$ and dividing by $e^{\max(s_1, s_2)}$ gives the stable fusion

$$o_t = \frac{e^{s_1 - m} o_1 + e^{s_2 - m} o_2}{e^{s_1 - m} + e^{s_2 - m}}, \qquad m = \max(s_1, s_2). \tag{10}$$

An empty bridge uses $Z_2 = 0$ and returns o_1. Overlapping supports must be deduplicated; otherwise shared sources receive duplicated exponential mass and the function changes.

A.6 Exact arithmetic matching

Let E denote the number of unique causal scores per query head. For $n \geq w$ and n divisible by b,

$$E_{\text{Full}} = n(n+1)/2, \qquad E_{\text{Block}} = n(b+1)/2, \tag{11}$$

$$E_{\text{SWA}} = nw - w(w-1)/2, \qquad E_{\text{PBB}} = n(b+1)/2 + (n/b - 1)h^2. \tag{12}$$

For an arbitrary block size a, set $q = \lfloor n/a \rfloor$ and $u = n - qa$. Then

$$E_{\text{Block(a)}} = qa(a+1)/2 + u(u+1)/2. \tag{13}$$

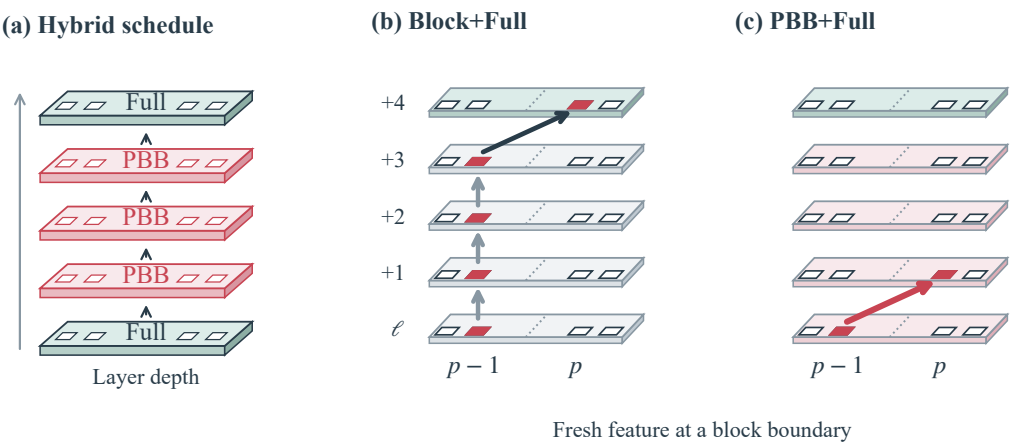

Figure 7: **Input coverage is not fresh-feature exchange.** Slabs denote successive layers. (a) Three local layers between Full layers. A fresh feature at $p-1$ after Full (red tile) reaches p at the next Full layer under Block attention (b), but in the next PBB layer (c). SWA also supplies this direct edge. Pale teal marks Full layers; only relevant paths are shown.

At $n = 8192$, $a = 192$ gives $q = 42$, $u = 128$, and $E = 786432$, exactly equal to PBB with $b = 128, h = 64$. Random-Edge selects 64 target phases and 64 previous-block source phases per boundary using seeded bijections; it therefore adds exactly 64^2 edges and matches PBB's added-degree histogram. Its fixed selection neither depends on model activations nor resamples during training.

Local pattern	Scores/head/local layer	Scores/head/20-layer 3:1 hybrid
Full	33,558,528	671,170,560
Block-128	528,384	175,718,400
SWA-128	1,040,448	183,399,360
PBB-128/64	786,432	179,589,120
Random-Edge	786,432	179,589,120
WideBlock-192	786,432	179,589,120

Table 4: Exact mathematical score counts at $n = 8192$. The Full row is all-Full. Random-Edge is trained both as a pure-local model and with the 3:1 Full schedule; the hybrid column gives its matching score budget.

Computing both $QK^\top$ and AV requires approximately $4d_hE$ multiply/add FLOPs per query head under the usual two-FLOP multiply-accumulate convention. Matching E, dimensions, projections, and other modules therefore matches the mathematical attention arithmetic for the designated controls. Kernel padding, tile layout, memory traffic, and launch overhead can still differ. Full, SWA, and PBB do *not* have identical model FLOPs simply because their token budgets match.

B The Value of Local Exchange with Global Communication

The design hypothesis is to retain useful local information exchange rather than maximize the range of local relay once a global mechanism is available. Three quantities must be distinguished: whether a path exists, how soon an intermediate feature can be transmitted, and how strongly that transmission affects a prediction. The first two admit graph statements. The third depends on learned transformations and the task. The following analysis connects these quantities without identifying them. The subsections below keep these levels apart: proved graph statements on paths, latency, and edge removal; external motivation from a production system; and interventions that remain untested. The measured retrieval gains are established by the training comparisons in Sections 5 and 6; this appendix does not identify their mechanism.

B.1 Input coverage versus intermediate-feature latency

Use a layered graph with nodes (ℓ, t), denoting the representation at position t after layer ℓ. An allowed attention or residual edge connects (ℓ, s) to $(\ell + 1, t)$. Define

$$\tau_\ell(s,t) = \min\{a \geq 1 : (\ell, s) \rightsquigarrow (\ell + a, t)\}, \qquad s < t, \tag{14}$$

with $\tau_\ell(s,t) = \infty$ if there is no such path. This is the minimum number of additional layers required for possible transmission. It is not the number of non-self attention hops on a fixed-depth path: residual edges can preserve a feature while it waits for a later cross-position edge.

Proposition 4 (Boundary exchange between Full layers) *Assume causal attention is the only cross-position operation, residual self paths are present, and each kth layer is Full. Let $f > \ell$ be the next Full layer, which is assumed to exist. For any integer $0 < h \leq b$ and fixed-block boundary $p > 0$, choose valid positions $s \in \{p - h, \ldots, p - 1\}$ and $t \in \{p, \ldots, p + h - 1\}$. Then*

$$\tau_\ell^{\text{Block+Full}}(s,t) = f - \ell \leq k, \qquad \tau_\ell^{\text{PBB+Full}}(s,t) = \tau_\ell^{\text{SWA+Full}}(s,t) = 1, \tag{15}$$

where SWA uses $w \geq 2h$; the experimental setting has $w = b = 2h$. For a source and target within one block, Block+Full also permits next-layer exchange.

Proof. Before layer f, fixed-block attention and residual edges cannot leave the source block. A residual path preserves access to position s until layer f, whose Full mask contains the edge to t. Hence the first possible cross-block arrival is $f - \ell$. For the specified source and target ranges, PBB contains (s, t) in its next local layer; the edge is also present if that layer is Full. Moreover $t - s \leq 2h - 1 \leq w - 1$, so the same conclusion holds for SWA. Within a block, Block attention contains every causal source–target edge. This proves the claims without requiring $b = 2h$.

The proposition concerns a fresh perturbation at (ℓ, s) with all other layer-ℓ states held fixed. Even if an earlier Full layer made every input token reachable, that layer has not broadcast this newly introduced intermediate perturbation. Thus input-level connectivity can be saturated while communication latency remains different. For a 3:1 schedule, a perturbation introduced just after a Full layer can cross the specified boundary in one additional PBB layer, instead of waiting four layers under Block+Full. Pure fixed-block attention has infinite cross-block latency; it is not merely a slower relay.

PBB matches SWA's one-layer exchange for these boundary pairs with 24.4% fewer total local scores at the default 8K geometry. The proposition therefore identifies a latency-preserving allocation: retain direct boundary exchange while leaving global communication to Full layers. Its prediction-level benefit depends on the learned transformations and task.

B.2 Weighted paths and effective influence

At a differentiable forward pass, write the block Jacobian as

$$J_\ell(t,s) = \frac{\partial H_t^{(\ell)}}{\partial H_s^{(\ell-1)}}, \qquad B_\ell(t,s) \geq \|J_\ell(t,s)\|_{\text{op}}. \tag{16}$$

Choose the nonnegative matrix B_ℓ to be zero outside the attention and residual support. The chain rule and triangle inequality give

$$\left\| \frac{\partial H_t^{(L)}}{\partial H_s^{(0)}} \right\|_{\text{op}} \leq (B_L B_{L-1} \cdots B_1)_{ts} = \sum_{\pi : s \rightsquigarrow t} \prod_{\ell=1}^{L} B_\ell(\pi_\ell, \pi_{\ell-1}). \tag{17}$$

Structural reachability determines where such paths are allowed, not their weights. These Jacobians include query/key dependence, normalization, residuals, and positionwise transformations; attention probabilities alone are not the full influence operator. For MoE, a local Jacobian description requires a neighborhood without a routing switch, or a finite-perturbation analysis instead.

Even a nonnegative normalized linear attention model does not imply exponential attenuation with hop count. Let S shift a scalar signal by one causal position and $T = (I + S)/2$. Away from the sequence boundary, a source m positions earlier contributes to T^{2m} with coefficient

$$(T^{2m})_{t,t-m} = \binom{2m}{m} 4^{-m} \sim \frac{1}{\sqrt{\pi m}}. \tag{18}$$

Each complete path has weight 4^{-m}, but there are $\binom{2m}{m}$ paths. Bounding one path does not bound their sum by the same factor. The deterministic shift $T = S$ preserves influence exactly along arbitrarily many hops; sharply peaked softmax weights can approximate this behavior at any fixed depth. General neural transformations can also amplify or reconstruct signals. An exponential influence law therefore needs additional contraction and path-multiplicity assumptions, which the present checkpoint evaluations do not establish.

Equation (17) nevertheless supplies a measurement target: compare influence by layer distance, source distance, and boundary phase. Effective receptive field should be defined using a declared influence or task metric and threshold, rather than substituted for the structural set $\mathcal{R}_L(t)$. Greater structural range permits more dependencies but need not increase any particular source's measured effect.

B.3 A SUFFICIENT CONDITION FOR INEXPENSIVE EDGE REMOVAL

For a single head, hold query, keys, and values fixed. Let A be a reference support, $S \subseteq A$ a nonempty retained support, and $D = A \setminus S$. Write a_s for the softmax probabilities on A, and $\mu = \sum_{s \in D} a_s$. Let o_A, o_S, o_D be the outputs normalized on their respective supports. For nonempty D,

$$o_A = (1-\mu)o_S + \mu o_D, \qquad \|o_A - o_S\| = \mu \|o_D - o_S\| \le 2V\mu, \tag{19}$$

where $V = \max_{s \in A} \|v_s\|$. If D is empty the difference is zero. The identity follows by splitting the numerator of attention into retained and removed terms. Small removed probability mass is sufficient for a small output change; it is not necessary, since retained and removed value averages may already be similar. Conversely, low edge count alone does not ensure a small change. Output projection multiplies the bound by its operator norm.

This identity applies to an SWA-to-PBB support restriction at the same hidden state, because default PBB is a subset of SWA. It makes the source-competition hypothesis concrete: removing sources renormalizes retained weights, while its immediate effect depends on both removed mass and value contrast. It does not imply that removed sources are noise or that renormalization improves prediction.

To extend the calculation across layers, let F_ℓ and $\widetilde{F}_\ell$ be the reference and restricted layer maps. Suppose the restricted map has Lipschitz constant K_ℓ on the relevant states and the same-state map discrepancy is bounded by ϵ_ℓ. From identical initial states, the state error obeys

$$\Delta_\ell \le K_\ell \Delta_{\ell-1} + \epsilon_\ell, \qquad \Delta_L \le \sum_{\ell=1}^{L} \epsilon_\ell \prod_{j=\ell+1}^{L} K_j. \tag{20}$$

This follows by adding and subtracting $\widetilde{F}_\ell$ evaluated at the reference state, then applying the Lipschitz bound and induction. Neither small ϵ_ℓ nor nonamplifying K_ℓ follows from the mask geometry. The separately trained PBB and SWA models also have different weights; Equation (20) is a same-checkpoint intervention model, not a derivation of their PPL ordering.

B.4 RELATION TO DEEPSEEK-V4.1-FLASH

DeepSeek-AI (2026) provide a production-scale example of prioritizing recent local exchange while retaining global history. Their Causal Encoder-Decoder has 20 encoder and 20 decoder layers. Section 2.2 gives an exact decoder SWA reconstruction span of an additional $n_{\text{win}}L/2$ prompt tokens, where $L/2$ denotes decoder depth. Citing effective receptive fields smaller than theoretical range (Chen et al., 2025), the report instead replays the most recent n_{win} tokens. Section 3.2.2 regenerates local SWA states while reusing cached global KV information; decoder replay uses recent encoder outputs and globally projected encoder KV. The report describes only a negligible impact on response quality from this approximate reconstruction and also simulates decoder replay during post-training.

This observation motivates our allocation hypothesis: when a global path retains long-range information, useful nearby transfer may justify more local computation than exact restoration of the full relay history. The interventions are distinct: bounded replay changes cache reconstruction with global sparse retrieval retained, whereas PBB changes local attention edges in a periodic-Full model. DeepSeek's CSA2 "Full Mode" differs from our all-prefix Full layer, and runtime local caches remain in its design. We use the reported quality observation as motivation, not as a PBB evaluation or

influence-decay theorem; the report's conclusion also identifies approximate reconstruction and sparse retrieval as possible error sources in untested cases.

The engineering connection is the separate treatment of local and global state. DeepSeek removes SWA state from long-lived persistent caching, keeps a short-lived local-state pool, and uses bounded replay on misses. PBB instead gives each local layer a phase-aligned support whose runtime KV can be maintained without relocation, while Full layers retain global history. Both designs avoid imposing the global state's storage or reconstruction policy on local state. Their mechanisms and memory scopes differ: bounded replay approximates cross-layer reconstruction, whereas Flash-PBB's ring layout exactly implements the prescribed local support. Section 7 supplies independent PBB measurements for this topology-to-execution connection.

B.5 Empirical implications and future interventions

Three observations support selective local exchange. Random-Edge controls show that edge placement matters at equal budget. Natural-text loss differences between pure PBB and Random-Edge concentrate at the repaired phases. The pure-local versus hybrid comparison shows that a smaller local support can yield higher probe accuracy when Full layers are present. In the context ablation, PBB+Full maintains lower absolute suffix NLL across all four lengths while the two hybrids have similar truncation penalties (Appendix D.9). These results connect targeted boundary exchange to measured behavior; the path-influence mechanism remains a hypothesis.

Future same-checkpoint interventions can separate these mechanisms. A source-dependent perturbation introduced at one layer-position node, with other states held fixed, would measure the arrival latency in Equation (15) and its final prediction effect. Removing bridge-exclusive edges versus matched-budget edges elsewhere would relate removed attention mass and value contrast to phase NLL and source gain. A global-KV-preserving replay intervention would separately test sensitivity to local reconstruction history; ordinary input truncation removes history from the global path too.

C Architecture, Data, and Optimization

Setting	200M dense	350M dense	2B MoE
Stored parameters	205,417,600	351,807,872	2,069,563,136
Active parameters	205,417,600	351,807,872	300,091,136
Layers	20	20	20
Hidden width	640	896	768
Dense FFN width	2,304	3,328	–
Query/KV heads	10/2	14/2	12/2
Head dimension	64	64	64
Routed experts / top-k	–	–	64/4
Routed / shared FFN width	–	–	640/800
Padded vocabulary	151,936	151,936	151,936
Sequence length	8,192	8,192	8,192
Tokens/update	393,216	393,216	393,216
Updates	10,449	18,000	15,259
Training tokens	4,108,713,984	7,077,888,000	6,000,082,944

Table 5: Architecture and exposure. Dense parameter counts include tied token embeddings. MoE active and stored counts are reported separately. Sparse/full comparisons keep these dimensions fixed.

All models use RoPE with base 10^6, RMSNorm with $\epsilon = 10^{-6}$, tied input/output embeddings, zero dropout, and initialization standard deviation 0.02. Attention uses BF16 computation with FP32 training master parameters. The common AdamW recipe (Loshchilov & Hutter, 2019) uses $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, weight decay 0.1, and gradient clipping at norm 1.0. The maximum/minimum LR is $3 \times 10^{-4}/3 \times 10^{-5}$. The schedule is warmup–stable–decay: a 5.56% linear warmup, a plateau to 88.89%, and cosine decay over the remaining updates. Integer step boundaries are recorded in each run configuration; for the 18,000-step run they are 1,000 and 16,000. For 200M they are 580 and 9,288. Physical micro-batch and gradient accumulation can vary with memory constraints while their

product remains 48 sequences. This is a common exposure recipe, not a separate per-architecture learning-rate search.

Training runs and result provenance. Full/PBB comparisons use two paired training seeds, 13 and 2025, at 200M, and one training seed, 13, for each architecture at 350M and 2B MoE. SWA+Full also has a final 200M evaluation at seed 2025; the remaining 200M controls use seed 13. In Table 1, run A denotes seed 13 and run B denotes seed 2025; slash-separated 200M effects in the main text follow this order. The local-family and scaling tables report seed 13. Figure 1 uses the seed-13 MoE trajectories. In Figure 4b, the red curve averages the two paired 200M runs after eight-phase binning, and the ribbon spans their observed range: PBB+Full improves clean NLL at 128/128 individual phases for seed 13 and 127/128 for seed 2025. Sample-bootstrap intervals describe evaluation-sample uncertainty, not variation across training seeds.

The main dense comparisons train on the same filtered token stream; seeds control initialization and sampling state. The MoE comparison likewise shares its data exposure and model dimensions. Scaling the dense family changes attention width and FFN width together, not only the FFN. The MoE experiment changes feed-forward organization and total capacity as well, so it tests transfer of the attention design across model families.

MoE top-4 probabilities are renormalized over the selected experts. The router load-balancing auxiliary-loss coefficient is 10^{-3}; reported validation/test NLL excludes this auxiliary term. All 20 feed-forward layers use routed and shared experts, with Qwen2-MoE implementation semantics, rather than substituting a different attention or recurrent module. The active-parameter count includes common parameters, router parameters, shared-expert parameters, and $4/64$ of the routed-expert parameters. The MoE LR boundaries are 848 warmup updates and decay starting at update 13,563. Activation checkpointing is enabled in training.

Data preparation. FineWeb-Edu documents undergo a conservative screening stage for explicit sexual promotion, followed by additional keyword screening and sampled manual review. Ordinary biological and health-related educational material is retained when not excluded by the conservative rules. Thirteen manually identified documents are excluded from the retained corpus; clean training text backfills their 24,594 tokens. Screening is imperfect and is not a guarantee of content safety. The model and evaluation implementation is separable from the private screening lexicon; reproducing the exact filtered corpus requires the retained-document selection, not just an informal statement that filtering occurred.

Document hashes determine disjoint train, validation, and test assignment before token packing. The retained training stream has 15,728,640,001 tokens. Validation and test each have 8,388,608 tokens; the documented test split contains 7,876 documents. The Qwen2.5 tokenizer and its end-of-text token 151643 are used throughout the main experiments. Training runs consume prefixes according to their budgets. Test data are never used for gradient updates.

PPL and phase conventions. Held-out PPL is $\exp(\text{mean token NLL})$, not the arithmetic mean of per-sequence perplexities. Dense models score 8,388,608 test targets; MoE pairs score the same 1,048,576-target prefix. Thus each architecture comparison is token matched, while cross-family absolute PPLs also differ in evaluation sample size. The readout phase is the position of the hidden state predicting the next token; the predicted-token phase is $(r+1) \bmod b$. The clean-NLL statistic excludes readouts with an end-of-text token in the 128-position input history ending at that readout. Phase comparisons use identical counts and the unsmoothed per-phase values; plotting averages do not change the reported statistics.

D Formal Structural and Likelihood Evaluation

D.1 Balanced source rotation

The formal v4 probe freezes 16 background/template anchors per distance–phase cell, with 15 distances and seven readout phases. Each cell has its own background spans, giving 1,680 distinct corpus offsets. The distance grid is

$$\{8, 16, 32, 64, 65, 96, 127, 128, 160, 256, 512, 1024, 2048, 4096, 6144\}.$$

Readout phases are $\{0, 16, 32, 63, 64, 96, 127\}$. The four candidate strings are `red`, `blue`, `green`, and `yellow`, each a single token, with IDs 2518, 6303, 6176, and 13753. Natural held-out text provides the background. Four templates state a secret word, recorded answer, key value, or code word. Source and readout positions are stored explicitly, with theoretical distance defined as their token-index difference.

Each group fixes background, positions, and template, then rotates all four source labels once. There are $16 \times 15 \times 7 = 1680$ groups and 6,720 rows. A candidate-prior-only predictor cannot improve over 25% within a complete group. Let $x^{(j)}$ be the prompt with source label j, and v_j its target token. Define

$$G = \frac{1}{4} \sum_{j=1}^{4} \left[\log p(v_j \mid x^{(j)}) - \frac{1}{3} \sum_{k \neq j} \log p(v_j \mid x^{(k)}) \right]. \tag{21}$$

The probabilities are normalized over the full model vocabulary. Positive G measures increased probability for a label when the source supports it, beyond the same label's probability under counterfactual sources. Accuracy selects the most likely candidate. Each group is evaluated as a unit for invariance and paired analyses; its four rows are not four independent background samples.

In a complete balanced group, the vocabulary log-partition terms cancel. Writing $a_j^{(k)}$ for the logit of label j under source rotation k gives the equivalent statistic

$$G = \frac{1}{4} \sum_{j=1}^{4} \left[a_j^{(j)} - \frac{1}{3} \sum_{k \neq j} a_j^{(k)} \right]. \tag{22}$$

Each prompt's log-partition appears once with coefficient $-1/4$ and three times with coefficient $+1/12$. Thus G is a counterfactual source-sensitive logit contrast, not a measure of full-vocabulary calibration. The recorded log-probability evaluation and this identity compute the same balanced-group quantity.

The manifest was frozen after Full-only calibration and before comparative architecture evaluation. WideBlock uses a 192-phase metadata view of the same frozen token tensors; its source positions and candidate labels do not change. Theoretical nonreachability requires identical logits across source rotations; the evaluator checks this with absolute tolerance 10^{-5}.

Numerical repeatability. The final 350M Full result uses all 6,720 rows at step 18,000: 73.6161% accuracy and source gain 1.983262. A separate complete evaluation gives 73.4970% and 1.983149. This small BF16 backend-sensitive variation preserves the Full/PBB ordering. A 32-row replay with FlashAttention 2.8.3.post1 before resumption showed maximum candidate-logit deviation 0.0625 and no candidate-ranking disagreement.

D.2 Paired uncertainty

For each pair, group IDs, source positions, and backgrounds are matched. We resample the 16 complete source-rotation groups within each of the 105 distance–phase cells, preserving equal cell weights, with 10,000 replicates and RNG seed 20260905 (Koehn, 2004; Berg-Kirkpatrick et al., 2012; Dror et al., 2018). Table 6 reports percentile 95% intervals with each complete four-label group as the resampling unit. Bootstrap intervals throughout this appendix describe sample uncertainty conditional on the evaluated checkpoints; training-seed replication is reported separately in Table 1. Differences, ratios, and aggregate metrics throughout the paper are computed from unrounded values, so recomputing them from displayed table entries can differ in the last digit. The following sensitivity analysis examines overlapping backgrounds.

Background-overlap sensitivity. A metadata audit finds 1,490 overlapping causal-prefix pairs among the 1,680 groups, forming 681 connected overlap components (largest: 15 groups). Using the full stored 8K spans gives 327 components (largest: 25). We additionally draw mean-one exponential weights per component, share them across its groups, normalize within each geometry cell, and average cells equally (10,000 draws; RNG seed 20260912). The resulting 2.5–97.5 percentile ranges retain positive accuracy differences for the five comparisons against Full or SWA and include zero for

WideBlock; all six source-gain differences remain positive. For 350M accuracy, the causal-prefix/full-span ranges are $[0.20, 2.14]$/$[0.22, 2.12]$ percentage points; for MoE they are $[1.20, 3.11]$/$[1.20, 3.13]$. The source package includes all ranges and overlap counts. These are geometric-overlap sensitivity ranges, rather than document-cluster confidence intervals, and are reported alongside Table 6.

Paired comparison	Accuracy difference (pp)	Source gain (nats)
200M, 13: PBB+Full − Full	11.47 [10.46, 12.50]	0.898 [0.858, 0.938]
200M, 2025: PBB+Full − Full	11.82 [10.74, 12.95]	0.594 [0.555, 0.633]
200M, 13: PBB+Full − SWA+Full	4.81 [3.96, 5.62]	0.238 [0.203, 0.273]
200M, 13: PBB+Full − WideBlock+Full	0.74 [-0.13, 1.62]	0.207 [0.162, 0.252]
350M, 13: PBB+Full − Full	1.18 [0.21, 2.14]	0.356 [0.315, 0.397]
MoE, 13: PBB+Full − Full	2.14 [1.13, 3.15]	0.388 [0.333, 0.442]

Table 6: Paired PBB+Full differences with cell-stratified group-bootstrap intervals. Positive values favor PBB+Full. The small accuracy difference against WideBlock includes zero, while its source-gain interval is positive.

D.3 Likelihood benchmarks

All likelihood tasks use zero-shot evaluation without a chat template. LAMBADA uses the `lambada_openai` task; HellaSwag, PIQA, ARC-Easy, SciQ, and ARC-Challenge report the harness's length-normalized accuracy. Winogrande and BLiMP report accuracy. LM-6 averages its six tasks equally. Table 7 gives every recorded final-checkpoint task value; Table 9 separates validation and held-out PPL, and Table 8 reports paired task differences and intervals.

Paired downstream analysis. We analyze retained per-item scores for Full and PBB+Full at both 200M seeds and the MoE endpoint. All 269,544 item pairs match in task, document ID, document content, prompt arguments, target, and filter; recomputed accuracies reproduce the reported values. Table 8 gives all eight task differences. We draw 20,000 paired bootstrap replicates with RNG seed 20260912, resampling the same items for both models. BLiMP resamples sentence pairs within each of its 67 fixed subtasks and averages the subtask means; LM-6 independently resamples items within each of its six fixed tasks and averages tasks equally. We report pointwise percentile intervals.

BLiMP's gains at 200M remain positive in both seeds. As a separate sensitivity to subtask composition, resampling the 67 paired subtask means gives ranges $[0.68, 3.04]$ and $[0.03, 3.02]$ points; this does not establish independence among linguistic templates. ARC-Challenge's positive point estimates have intervals crossing zero in all three pairs. For MoE, the equally weighted LM-6 improvement has interval $[0.194, 1.655]$, although most individual-task intervals cross zero. Retaining the same items jointly across the two fixed 200M seeds gives a mean LM-6 difference of -0.109 points, with interval $[-0.662, 0.435]$.

The supplementary numerical files include discordant item counts, two-sided exact McNemar tests, and Holm adjustments over eight tasks within each pair and all 24 task comparisons. These tests assume independent item pairs; the BLiMP subtask sensitivity separately probes template dependence. Pointwise intervals are not multiplicity-adjusted: MoE ARC-Easy's interval excludes zero, but its within-pair Holm-adjusted p-value is 0.309. This secondary analysis adds uncertainty estimates without changing checkpoints, metrics, or the original point estimates.

D.4 Local-family comparison

Figure 3 shows the seed-13 SWA/PBB comparison alongside the paired Full/PBB results in Table 1. Pure SWA has lower PPL and higher probe accuracy than pure PBB. Under the shared periodic-full schedule, the hybrids have comparable PPL, while PBB+Full has higher probe accuracy. The additional seed-2025 SWA+Full endpoint has 21.214 test PPL, 77.62% probe accuracy, and 2.053-nat source gain; PBB+Full has 21.222, 80.07%, and 2.405, respectively. Thus the local pattern's role in the full architecture matters: the best pure-local choice need not be the best hybrid component.

Model	Attention	Seed	LAMB.	Hella.	PIQA	ARC-E	SciQ	Wino.	BLiMP	ARC-C
200M	Full	13	23.13	30.52	60.83	44.49	63.40	52.33	78.06	25.09
200M	Full	2025	23.37	30.67	61.59	45.33	66.10	52.25	78.44	24.91
200M	Block	13	18.79	30.55	61.86	46.89	60.10	50.91	79.71	25.94
200M	PBB	13	22.47	30.45	61.10	45.92	60.10	49.64	80.37	25.60
200M	SWA	13	23.33	31.11	61.26	45.58	63.90	51.62	79.64	26.02
200M	PBB+Full	13	24.41	30.57	61.59	45.33	63.90	50.99	79.83	25.77
200M	PBB+Full	2025	23.37	30.94	61.43	45.12	64.70	50.36	79.90	25.26
200M	SWA+Full	13	24.90	30.69	60.83	45.20	63.00	50.28	79.89	26.02
200M	SWA+Full	2025	25.21	30.94	61.59	46.04	64.50	49.96	79.58	26.45
200M	WideBlock-192+Full	13	22.20	31.10	62.02	44.07	64.80	51.30	80.37	25.43
200M	Random-Edge	13	21.11	30.50	61.48	44.87	62.20	51.46	79.74	25.94
200M	Random-Edge+Full	13	22.43	30.68	61.97	44.49	64.30	52.25	80.43	25.34
350M	Full	13	28.16	35.79	64.36	49.45	67.10	50.28	80.75	25.34
350M	PBB+Full	13	28.45	35.67	64.09	49.03	69.60	50.43	80.40	25.60
MoE	Full	13	32.51	39.83	66.43	52.53	69.60	51.22	81.03	28.16
MoE	PBB+Full	13	32.84	39.99	67.30	54.21	71.00	52.33	81.09	29.61

Table 7: Final-checkpoint zero-shot likelihood results (percent). Hybrids use 3:1 local/full layers. ARC-E/ARC-C are Easy/Challenge. The eight task columns retain their specified metric definitions; LM-6 excludes BLiMP and ARC-C.

	200M, seed 13		200M, seed 2025		2B MoE, seed 13	
Task	Δ	95% interval	Δ	95% interval	Δ	95% interval
LAMBADA	$+1.281$	$[+0.310, +2.251]$	$+0.000$	$[-0.912, +0.931]$	$+0.330$	$[-0.699, +1.358]$
HellaSwag	$+0.050$	$[-0.498, +0.597]$	$+0.269$	$[-0.279, +0.807]$	$+0.159$	$[-0.408, +0.747]$
PIQA	$+0.762$	$[-0.762, +2.285]$	-0.163	$[-1.687, +1.360]$	$+0.871$	$[-0.653, +2.394]$
ARC-Easy	$+0.842$	$[-0.715, +2.399]$	-0.210	$[-1.684, +1.263]$	$+1.684$	$[+0.126, +3.241]$
SciQ	$+0.500$	$[-1.700, +2.700]$	-1.400	$[-3.600, +0.800]$	$+1.400$	$[-0.500, +3.300]$
Winogrande	-1.342	$[-4.578, +1.896]$	-1.894	$[-5.209, +1.342]$	$+1.105$	$[-1.973, +4.183]$
ARC-Challenge	$+0.683$	$[-1.195, +2.560]$	$+0.341$	$[-1.536, +2.133]$	$+1.451$	$[-0.512, +3.413]$
BLiMP	$+1.775$	$[+1.488, +2.066]$	$+1.458$	$[+1.173, +1.743]$	$+0.058$	$[-0.207, +0.331]$
LM-6	$+0.349$	$[-0.431, +1.121]$	-0.567	$[-1.345, +0.200]$	$+0.925$	$[+0.194, +1.655]$

Table 8: Paired downstream differences (PBB+Full minus Full, percentage points) and pointwise 95% bootstrap intervals. All comparisons use final checkpoints. LM-6 keeps six equally weighted tasks; BLiMP keeps 67 fixed subtasks. Full task values appear in Table 7.

D.5 Source retrieval throughout training

Table 10 reports 5k, 10k, and final evaluations on the same frozen 6,720-row, four-candidate probe. PBB+Full exceeds Full at every listed checkpoint in both seeds and exceeds SWA+Full at all three seed-13 checkpoints. The endpoint advantage therefore also appears at intermediate stages within each training trajectory. Main comparisons use matched final training budgets.

D.6 Distance-resolved source use

Figure 8 retains all fifteen distances of the frozen probe, averaging the seven phases and sixteen groups per phase: 112 groups at each distance. Positive differences favor PBB+Full. The figure separates the two 200M seeds, 350M, and MoE, and includes the seed-13 SWA hybrid comparison.

At 200M, aggregating distances up to 128 tokens gives PBB+Full advantages over Full of 17.22 and 15.60 percentage points for seeds 13 and 2025. The corresponding seed-13 advantage over SWA+Full is 6.28 points. Gains are not confined to nearby sources: over distances 2,048, 4,096, and 6,144, the MoE hybrid achieves 45.54% accuracy versus Full's 40.77%, with a source-gain increase of 0.227 nats. These are exploratory decompositions of existing measurements, not separately tuned probes. The aggregate does not imply improvement at every distance: the 200M Full comparison reverses at

Model	Attention	Seed	Val PPL	Test PPL	Probe (%)	Source gain
200M	Full	13	22.26	21.52	69.84	1.645
200M	Full	2025	22.12	21.38	68.26	1.811
200M	Block	13	31.28	30.42	35.83	0.315
200M	PBB	13	24.82	24.05	46.80	0.647
200M	SWA	13	24.09	23.31	54.27	0.911
200M	PBB+Full	13	21.95	21.27	81.31	2.543
200M	PBB+Full	2025	21.94	21.22	80.07	2.405
200M	SWA+Full	13	22.00	21.29	76.50	2.305
200M	SWA+Full	2025	21.92	21.21	77.62	2.053
200M	WideBlock-192+Full	13	22.40	21.67	80.57	2.336
200M	Random-Edge	13	26.43	25.63	42.02	0.576
200M	Random-Edge+Full	13	22.35	21.61	75.68	1.934
350M	Full	13	17.80	17.13	73.62	1.983
350M	Block	13	25.94	25.04	35.19	0.387
350M	PBB+Full	13	17.97	17.30	74.79	2.339
MoE	Full	13	15.61	14.25	75.62	2.326
MoE	PBB+Full	13	15.69	14.35	77.77	2.714

Table 9: Final dense/MoE checkpoints, including both 200M hybrid controls. Validation and test PPL use separate splits. Probe results use 6,720 balanced rows; source gain is averaged over complete four-label rotations.

Seed	Step	Full	SWA+Full	PBB+Full
13	5,000	71.71	76.22	80.68
13	10,000	72.02	78.29	82.17
13	10,449	69.84	76.50	81.31
2025	5,000	76.01	–	79.64
2025	10,000	70.71	–	81.09
2025	10,449	68.26	77.62	80.07

Table 10: 200M probe accuracy (percent) across training. Every reported entry uses 1,680 complete source-rotation groups; seed-2025 SWA+Full has a final-only evaluation.

6,144, 350M loses at the shortest distances, and MoE has mixed short-distance results. The complete curves make these differences explicit.

D.7 Literal needle and exact entity–attribute retrieval

We report two additional controlled synthetic tasks, evaluated on the same final checkpoints as the main comparisons; the equal-arithmetic control below adds a third family at short source distances. *Literal needle retrieval* inserts a statement such as `The secret word is red.` into natural held-out text and repeats its prefix at the readout. Four templates use "secret word," "recorded answer," "key value," and "code word." *Exact entity–attribute retrieval* (the original task identifier `semantic_exact`) inserts an attribute statement such as `Alice's ribbon was red.` and queries `Alice's ribbon was`. The four entity–object pairs are Alice–ribbon, David–scarf, Maria–ribbon, and James–scarf. Source and query wording match within each task; the latter measures exact entity–attribute retrieval, not paraphrase understanding.

Both tasks use distances

$$\{128, 256, 512, 768, 1024, 1536, 2048, 2560, 3072, 3584, 4096, 4608, 5120, 6144, 7168\}$$

and readout phases $\{0, 16, 32, 63, 64, 96, 127\}$. At each distance–phase cell, 16 anchors cycle through the four templates or entity–object pairs. The readout is at $8064 + r$, the source label at $8064 + r - d$, and the stored input contains 8,192 tokens. Causal scoring uses only the prefix through the readout. A complete four-label rotation changes only the source label token; background, query, and positions remain fixed. Each task has 1,680 groups and 6,720 scored rows. The formal manifest was generated before these checkpoint evaluations, with generation seed 20260915, and shared unchanged across architectures. Accuracy and source gain G follow the balanced-group definitions above. These tasks have their own distance grid and are reported separately from the original phase–distance probe.

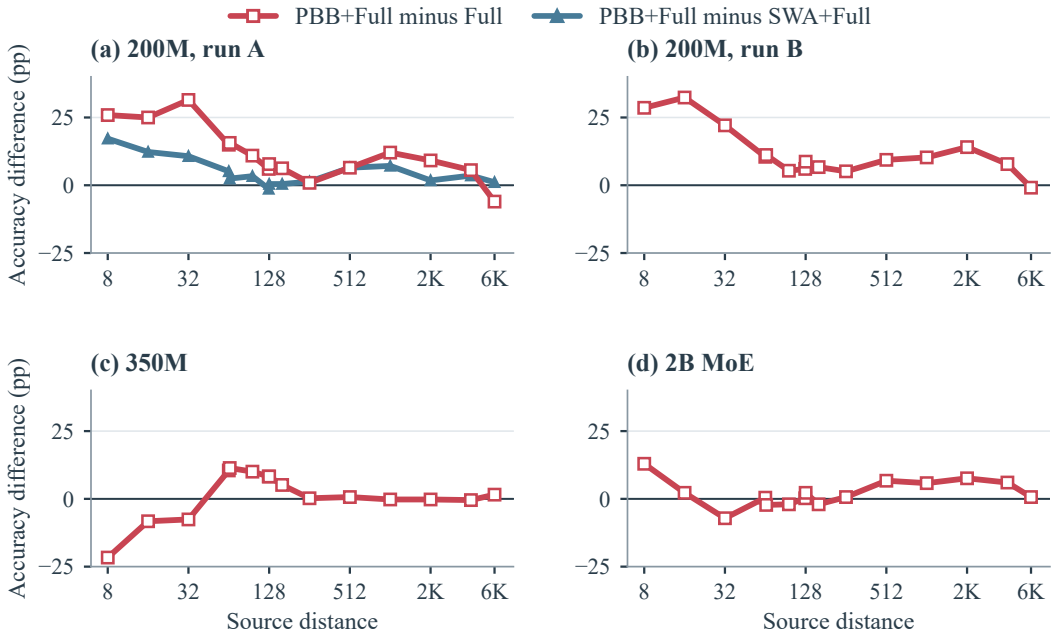

Figure 8: **Where the hybrid's source-retrieval gains occur.** Accuracy differences in percentage points on all fifteen measured distances; every point averages 112 matched source-rotation groups. Red compares PBB+Full with Full; blue additionally compares it with SWA+Full at 200M seed 13. Lines connect point estimates, without fitted interpolation or uncertainty bands. Positive and negative differences use the same vertical scale across panels.

			Literal needle		Semantic exact	
Scale	Seed	Attention	Acc.	G	Acc.	G
200M	13	Full	52.38	1.1450	53.32	1.5474
		Block	25.00	0.0000	25.00	0.0000
		SWA	25.19	0.0200	25.10	0.0044
		PBB	25.24	0.0117	25.03	0.0028
		Random-Edge	25.58	0.0276	25.46	0.0188
		SWA+Full	53.65	1.3513	61.18	1.9106
		Random-Edge+Full	52.46	1.1833	59.46	1.7166
		PBB+Full	55.80	1.4202	58.07	1.9749
		WideBlock-192+Full	55.95	1.4384	54.05	1.3320
200M	2025	Full	48.82	1.0522	52.77	1.5327
		SWA+Full	51.25	1.1719	52.16	1.4800
		PBB+Full	55.03	1.3685	56.47	1.7375
350M	13	Full	50.04	1.1877	53.35	1.7135
		Block	25.00	0.0000	25.00	0.0000
		PBB+Full	50.85	1.2871	56.03	2.1348
2B MoE	13	Full	51.76	1.3469	52.44	2.0605
		PBB+Full	56.88	1.6827	66.46	2.9172

Table 11: **Literal needle and exact entity–attribute retrieval across all 17 final checkpoints.** Each task uses 6,720 balanced rows, with 25% chance accuracy. Accuracies are percentages and G is source gain in nats. Names ending in +Full use the 3:1 hybrid schedule. Values are point estimates; the original phase–distance probe is a separate evaluation.

PBB+Full improves both accuracy and source gain over Full on both tasks in all four matched comparisons: the two 200M seeds, 350M, and 2B MoE (Table 11). At 2B MoE, needle accuracy increases by 5.12 percentage points with a 0.336-nat source-gain increase; exact entity–attribute accuracy increases by 14.02 points with a 0.857-nat gain increase. For the latter task, 10,000 paired bootstrap draws of complete four-label groups within 105 distance–phase cells give intervals of $[12.78, 15.31]$ points and $[0.778, 0.938]$ nats, conditional on the two fixed checkpoints. The point estimates are reproduced from the saved group outputs by `paired_entity_attribute_20260925.py`. At 200M seed 13, WideBlock-192+Full and PBB+Full have close needle accuracy (55.95% and 55.80%), while PBB+Full has higher exact entity–attribute accuracy (58.07% versus 54.05%) and source gain (1.975 versus 1.332). SWA+Full has the highest run-A exact entity–attribute accuracy among the 200M models, whereas PBB+Full has slightly higher source gain. Accuracy and source sensitivity thus give complementary views of the local graph.

Table 12 also reports a third family, *Bridge-window*, which reuses the literal-needle construction under the same stored 8,192-token causal input: the same four templates, the same source and query wording, the readout at $8064+r$, and a single-token source label rotated over the four candidate colors. The only change is the distance grid, $\{32, 40, 48, 56, 63, 64, 65, 80, 96, 112, 127, 128, 129, 160, 192\}$, which brackets the 128-token block boundary. Readout phases, anchors per cell, and cell count are unchanged, so the family again contributes 1,680 groups; label rotation and the accuracy and source-gain definitions are those of Appendix D. The Balanced row averages the three family-level entries with equal weight, covering 5,040 of the 8,400 generated groups.

	Accuracy (%)			Source gain G (nats)		
Task family	PBB+Full	R-E+Full	Δ	PBB+Full	R-E+Full	Δ
Literal needle	55.80	52.46	+3.35 [2.40, 4.32]	1.420	1.183	+0.237 [0.200, 0.275]
Bridge-window	92.57	89.35	+3.23 [2.31, 4.14]	3.118	2.410	+0.708 [0.665, 0.751]
Semantic exact	58.07	59.46	−1.40 [−2.50, −0.28]	1.975	1.717	+0.258 [0.203, 0.314]
Balanced	68.81	67.09	+1.73 [1.17, 2.31]	2.171	1.770	+0.401 [0.373, 0.429]

Table 12: **Equal-arithmetic control with Full layers.** Random-Edge+Full (R-E+Full) matches PBB+Full's added-degree histogram but randomizes edge placement. Differences are PBB+Full minus R-E+Full; positive favors PBB+Full. Intervals are group-bootstrap 95% intervals over 1,680 groups per task and 5,040 groups for the balanced average.

D.8 Natural long-context sanity checks

We additionally evaluated the final 2B MoE Full and PBB+Full checkpoints on LongBench-v2 (Bai et al., 2025) and QuALITY (Pang et al., 2022). These runs are retained as sanity checks rather than main evidence. For LongBench-v2, all 503 multiple-choice examples were scored after deterministic head–tail truncation to 8,192 tokens; every document required truncation. For QuALITY, we used all 2,039 development questions whose complete article, question, and four answer choices fit within 8,192 tokens, without truncation. In both cases, a base model selects the option with the lowest mean token NLL. Each context condition is paired with a no-context version containing the same question and choices. This protocol is not directly comparable to either benchmark's official leaderboard.

Benchmark	Model	Context	No context	Gain
LongBench-v2 (503)	Full	23.46	22.27	+1.19
	PBB+Full	24.65	25.25	−0.60
QuALITY (2,039)	Full	26.48	27.51	−1.03
	PBB+Full	26.83	26.78	+0.05

Table 13: Natural long-context sanity checks on the 2B MoE checkpoints. Entries are four-choice accuracy in percent; chance is 25%. Gain is context minus no-context accuracy. LongBench-v2 uses a fixed 8K head–tail view, whereas QuALITY includes only complete documents that fit within 8K.

Both benchmarks remain near four-choice chance accuracy, and context does not yield a consistent improvement over the paired no-context condition. We therefore do not use these numbers to rank

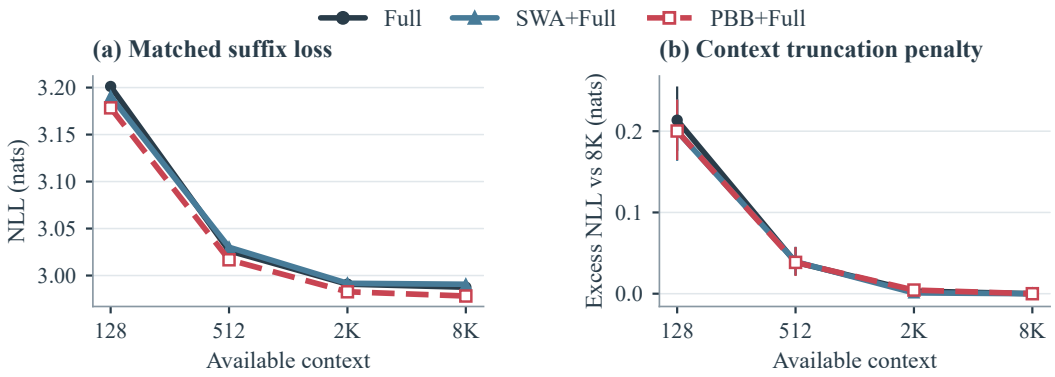

Figure 9: The completed 128-span context ablation. Left: absolute suffix NLL. Right: excess over each model's 8K-context NLL with sequence-bootstrap 95% intervals. All three architectures score identical targets.

the attention architectures in the main text. Our interpretation is that, for these pretrained base models, instruction-like question answering, option semantics, and task-format adaptation dominate the measurement before long-context access becomes discriminative. The context/no-context parity supports that interpretation but does not isolate its cause. The controlled retrieval tasks instead test source use directly while holding the readout format, candidates, positions, and backgrounds fixed.

D.9 Paired context-window ablation

The completed ablation uses 128 uniformly spaced, nonoverlapping 8K spans from the held-out test stream and the 200M seed-13 final checkpoints. All three architectures score the same 64 final target tokens of each span under contexts 128, 512, 2,048, and 8,192: 8,192 target tokens per architecture and context setting. These suffix readouts are the scoring hidden states at the final 64 positions of each window, indices 8,128–8,191 of the 8,192-token span, each predicting the following token; they therefore occupy phases 64–127, not the directly repaired first half-block. Truncated position origins reset, but the chosen context lengths preserve block phase, and RoPE relative offsets remain matched. The protocol records every corpus offset; it does not select spans based on architecture performance.

Table 14 reports mean suffix NLL and excess (Figure 9). Percentile intervals use 10,000 paired sequence-bootstrap replicates with seed 20260905. At 128 tokens, the PBB+Full minus SWA+Full penalty difference is 0.000289 nats, with interval $[-0.009844, 0.010333]$. The corresponding difference against Full is -0.013370 nats, with interval $[-0.025428, -0.001357]$. PBB+Full achieves lower absolute suffix NLL at all four settings, with a truncation penalty similar to SWA+Full and smaller than Full at 128 tokens.

Architecture	Context	Suffix NLL	Excess NLL	95% interval
Full	128	3.2012	0.2136	[0.1759, 0.2550]
Full	512	3.0263	0.0387	[0.0225, 0.0570]
Full	2,048	2.9908	0.0032	[-0.0014, 0.0089]
Full	8,192	2.9876	0.0000	[0.0000, 0.0000]
SWA+Full	128	3.1906	0.1999	[0.1634, 0.2392]
SWA+Full	512	3.0301	0.0394	[0.0228, 0.0578]
SWA+Full	2,048	2.9917	0.0010	[-0.0028, 0.0055]
SWA+Full	8,192	2.9907	0.0000	[0.0000, 0.0000]
PBB+Full	128	3.1784	0.2002	[0.1645, 0.2389]
PBB+Full	512	3.0168	0.0385	[0.0218, 0.0575]
PBB+Full	2,048	2.9827	0.0045	[0.0002, 0.0098]
PBB+Full	8,192	2.9782	0.0000	[0.0000, 0.0000]

Table 14: Matched context ablation, 128 test spans. Excess NLL is relative to each model's 8K suffix NLL; brackets give sequence-bootstrap 95% intervals.

E CHECKPOINT-LEVEL VALIDATION

Tables 15–17 (Figure 10) report synchronized validation trajectories under the shared warmup–stable–decay schedule. The curves show how each pair develops across training, while final held-out comparisons use equal-budget endpoints. Architecture pairs share the learning rate at each matched update; comparisons across updates also reflect the schedule.

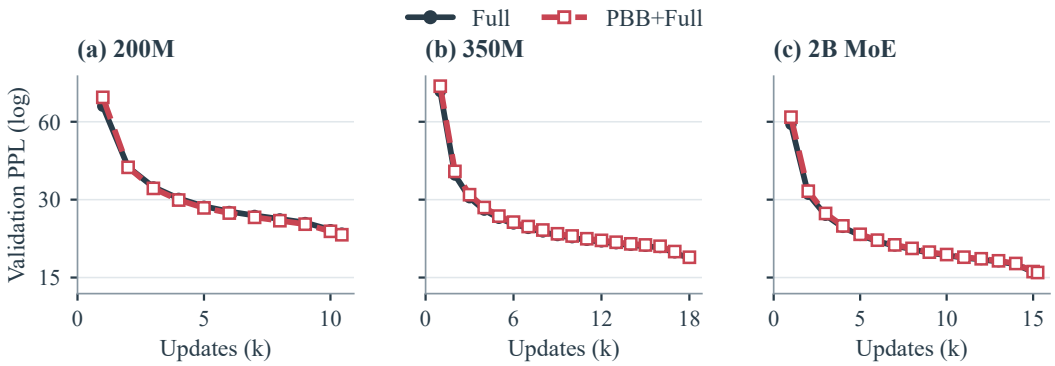

Figure 10: Synchronized seed-13 validation trajectories for Full and PBB+Full. Axes show model-specific update budgets ($k = 1000$) and logarithmic PPL. The exact values and relative gaps are given in the following tables.

Step	Full PPL	PBB+Full PPL	Gap (%)
1,000	68.94	74.56	8.14
2,000	40.11	40.00	-0.28
3,000	33.57	33.16	-1.21
4,000	30.33	29.89	-1.43
5,000	28.27	27.88	-1.39
6,000	26.99	26.62	-1.34
7,000	26.05	25.63	-1.62
8,000	25.27	24.88	-1.54
9,000	24.46	24.12	-1.38
10,000	22.95	22.63	-1.38
10,449	22.26	21.95	-1.39

Table 15: 200M, seed 13: synchronized validation PPL. Gap is $100(\mathrm{PPL}_{\mathrm{PBB+Full}}/\mathrm{PPL}_{\mathrm{Full}} - 1)$.

Step	Full PPL	PBB+Full PPL	Gap (%)
1,000	78.55	82.34	4.83
2,000	37.36	38.58	3.27
3,000	30.53	31.35	2.69
4,000	27.34	27.98	2.33
5,000	25.43	25.91	1.92
6,000	24.18	24.56	1.59
7,000	23.24	23.62	1.62
8,000	22.52	22.90	1.68
9,000	21.81	22.14	1.52
10,000	21.40	21.71	1.48
11,000	20.91	21.19	1.32
12,000	20.61	20.88	1.30
13,000	20.28	20.54	1.27
14,000	20.00	20.24	1.23
15,000	19.82	20.05	1.14
16,000	19.61	19.82	1.08
17,000	18.70	18.90	1.06
18,000	17.80	17.97	0.93

Table 16: 350M, seed 13: synchronized validation PPL under the same definition.

Step	Full PPL	PBB+Full PPL	Gap (%)
1,000	58.68	62.50	6.50
2,000	31.48	32.35	2.74
3,000	26.12	26.53	1.58
4,000	23.45	23.75	1.27
5,000	21.84	22.04	0.93
6,000	20.73	20.92	0.94
7,000	19.88	20.06	0.89
8,000	19.32	19.44	0.65
9,000	18.66	18.80	0.74
10,000	18.31	18.41	0.54
11,000	17.87	17.98	0.59
12,000	17.62	17.73	0.61
13,000	17.28	17.41	0.74
14,000	16.88	17.00	0.69
15,000	15.73	15.82	0.52
15,259	15.61	15.69	0.50

Table 17: 2B MoE, seed 13: synchronized validation PPL. Both models receive 6.000B tokens.

F KERNEL, TRAINING, AND CACHED-INFERENCE PROTOCOLS

Several execution series appear below under similar names but at different scopes, implementations, and devices, so their numbers are not interchangeable. Appendix F.11 at the end of this appendix indexes them by scope, batching, hardware, and repetition.

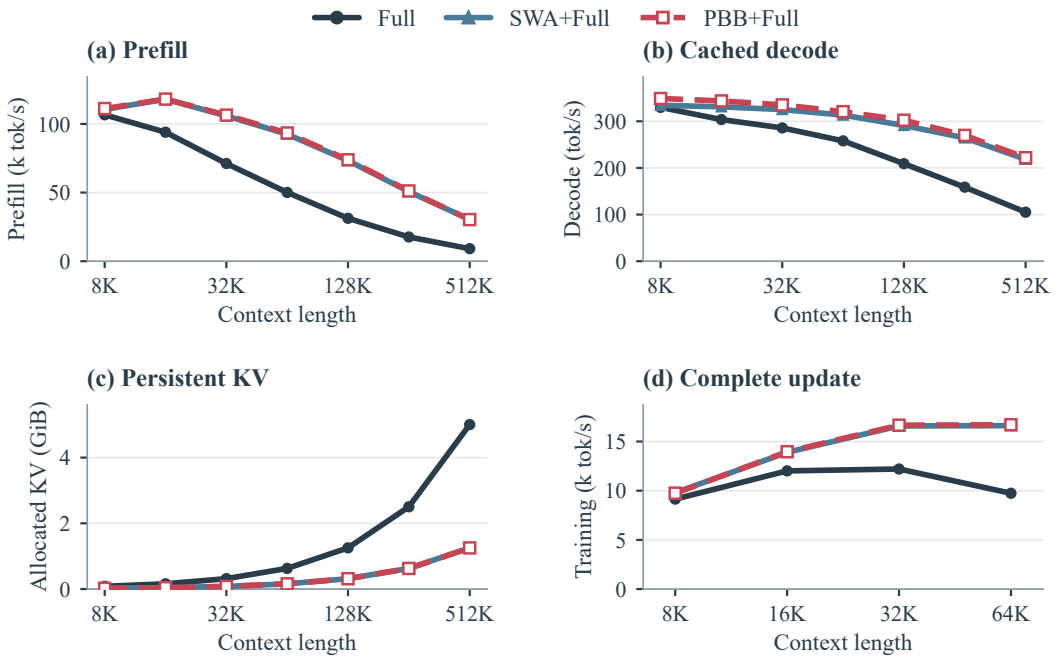

Figure 11: **Whole-model execution across context lengths.** Full, SWA+Full, and PBB+Full on RTX 4090: (a) cache-populating prefill; (b) sequential cached decode; (c) physical KV allocation; (d) complete training updates. Curves show within-process medians and ranges under the phase-specific protocols below. All successful lengths are shown; 1M inference and training at 128K or above exhaust memory. $K = 1024$; k tok/s uses $k = 1000$.

F.1 PHASE-ALIGNED SUPPORT AND EXACT RING STORAGE

For any integer $0 < h \le b$, write $t = p + r$ as in Section 3. The oldest accessible position is

$$a(t) = \begin{cases} p - h, & p > 0, \quad 0 \le r < h, \\ p, & \text{otherwise}, \end{cases} \qquad \mathcal{A}_{\mathrm{PBB}}(t) = \{a(t), \dots, t\}. \tag{23}$$

Within a noninitial block, $a(t)$ is constant through the repaired phases and increases to p when $r = h$, if $h < b$. At the next boundary it is $p + b - h \ge p$. When $h = b$, all phases are repaired and the endpoint increases by b at each boundary. Initial-block transitions are also nondecreasing. The support size is at most $2h$ in repaired phases and b otherwise, giving the bound $M = \max(b, 2h)$.

Proposition 5 (Relocation-free PBB cache for independent widths) *Under sequential, single-token causal decoding with any integer $0 < h \le b$, an M-slot ring, $M = \max(b, 2h)$, stores token s at slot $s \bmod M$ and preserves every KV entry required by current and future PBB readouts without relocation.*

Proof. The current support is an interval of at most M positions, so all required tokens have distinct residues modulo M. Appending token t overwrites $t - M$ when that position exists. Since $t - a(t) + 1 \le M$, we have $t - M < a(t)$: the overwritten token is outside the current support. Monotonicity of a means that no later query requires it. Before the ring fills, writes overwrite no existing token. Induction over sequential writes proves preservation. Absolute RoPE positions are

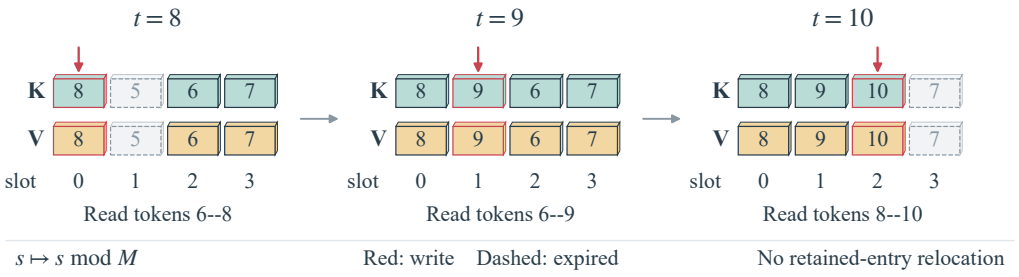

Figure 12: **Relocation-free Flash-PBB cache.** Consecutive toy states ($b = 4$, $h = 2$): each slot stores K and V for the numbered token; s overwrites slot $s \bmod 4$ without moving retained entries. Support changes from 6–9 at $t = 9$ to 8–10 at $t = 10$. Pale dashed entries are expired; red marks writes. In general, $M = \max(b, 2h)$ suffices; models use $M = 128$.

unchanged by physical slots. The maximum support M is attained in the steady state, so the ring needs no spare entries for the prescribed single-token readout.

The experimental $b = 2h$ setting specializes this result to $M = b$. Equation (9) then gives $a(t) = \max(0, h(\lfloor t/h \rfloor - 1))$. After the initial block, the number of sources is

$$|\mathcal{A}_{\text{PBB}}(t)| = h + 1 + (t \bmod h) \le 2h = b. \tag{24}$$

Its steady-state phase average is $(3h + 1)/2$, or 96.5 at $h = 64$, versus SWA's 128 sources. This count describes logical reads; physical allocation and executed tile work are separate quantities.

At the default width, bridge phases read the preceding tail in slots 64–127 followed by the current prefix in slots 0–63; the other phases read the current prefix contiguously. Flash-PBB implements these layouts directly in a 128-column, four-warp FlashAttention-derived kernel, fusing KV append and attention without staging gathers or buffer compaction. Native GQA groups query heads over shared KV heads. The implementation retains the compatible low-precision softmax/reduction order; it does not add another normalization or approximate the support.

For batch B, H_{KV} KV heads, head dimension d_h, and element size u, an M-slot ring uses $2BMH_{\text{KV}}d_h u$ bytes. With $B = 1$, $M = b = 128$, $H_{\text{KV}} = 2$, $d_h = 64$, and BF16, this is 65,536 bytes. The previous Flash-PBB and tested Flash-SWA paths allocate 256 positions, or 131,072 bytes, and relocate 65,536 bytes of retained KV payload every 128 steps. The ring removes this relocation, equivalent to 512 payload bytes per token per layer averaged over phases. Payload is an array-copy count, not a measured DRAM transaction count. SWA also admits a 128-entry ring; the measured allocation reduction is relative to the tested backend, not an intrinsic SWA storage lower bound.

F.2 Batched Flash-PBB decode

The 2B MoE batch sweep compares Flash-PBB with Flash-SWA at batch sizes 1, 2, 4, 8, 16, 32, and 64 on an RTX 4090. The local-core scope measures batched attention, KV append, and rollover maintenance over all 128 phases using CUDA Graphs; each value is time per batched step, not time per generated token. The complete-model scope measures 256-token continuations at batches 1, 4, 16, 32, and 64, including projections, experts, KV updates, and token selection. Both use BF16, two KV heads, head dimension 64, and an 8K prefix per sequence. The saved raw results are in `efficiency_data/batched_decode_20260920/`. An independent same-device local-core rerun reproduces the batch-8-to-16 transition; a second decode-core protocol also measures Flash-SWA/Flash-PBB time ratios of 1.85, 3.32, and 8.73 at batches 8, 16, and 64, respectively. Replacing SWA's strided cache view with a contiguous 128-slot view leaves the knee intact: 14.59 versus 14.54 μs at batch 16 and 41.92 versus 41.93 μs at batch 64. The rerun records are in `efficiency_data/batched_decode_20260924_rerun/`. Table 18 gives every measured batch point used in Figure 5. Each local-core entry is a within-process median over seven timed CUDA-graph samples (three warmup replays, 20 in-graph iterations per sample) of a 128-phase

single-layer graph, reported per step; each complete-model entry is the median of three 256-token continuation runs from matched seeded initialization.

	Local attention core			Complete hybrid		
Batch	SWA μs	PBB μs	Speedup	SWA tok/s	PBB tok/s	Gain
1	8.51	4.33	1.96×	341	349	2.3%
2	7.86	4.33	1.82×	–	–	–
4	7.94	4.36	1.82×	975	997	2.3%
8	8.13	4.37	1.86×	–	–	–
16	14.60	4.39	3.33×	2,436	2,511	3.1%
32	21.78	4.49	4.85×	4,027	4,216	4.7%
64	42.03	4.72	**8.91**×	6,484	6,942	**7.1%**

Table 18: Matched 2B MoE decode batch sweep. Core speedup is SWA/PBB time; whole-model gain is PBB+Full throughput relative to SWA+Full. Dashes mark batch sizes without a complete-model run.

F.3 Measured execution resources and memory requests

We profile the same compiled SM89 kernels on a local RTX 4060 Laptop GPU using Nsight Compute 2024.1.1, Torch 2.6.0+cu124, and FlashAttention 2.8.3.post1. The synthetic BF16 inputs use batch 1, 12 query heads, two KV heads, and head dimension 64, matching the MoE attention shape. At position origin 8K, chronological decode samples phases $\{0, 20, 63, 64, 80, 127\}$, covering bridge entry, bridge exit, and maximum support. New/old PBB outputs match bitwise over 384 steps.

Each backend is run in three independent processes per cache policy, with rotating backend order: 18 runs and 108 profiled attention calls. Cold-cache collection uses kernel replay with cache flushing; retained-cache collection uses application replay without flushing after identical warmup. The profiler selects only the attention kernel, including fused KV append but excluding separate rollover copies. Table 19 reports the measured resources and memory requests.

Metric	Flash-PBB	Previous PBB	Flash-SWA
Registers/thread	212	255	255
Dynamic shared memory/block (KiB)	40	72	72
Physical local KV (KiB)	64	128	128
Global-load sectors/call	1,112–2,120	1,112–2,120	15,168
Global-store sectors/call	67	67	156
Thread-local load sectors/call	0	32	2,880
Thread-local store sectors/call	0	32	5,952

Table 19: **Lower resource use and fewer memory requests.** RTX 4060, 2B attention shape. Counts repeat exactly across all three processes and both cache policies at each phase; ranges span the six sampled phases, not uncertainty intervals. Registers and shared memory are launch resources; sector counts are hardware-observed requests.

Flash-PBB uses 212 registers/thread and 40 KiB dynamic shared memory/block, versus 255 and 72 KiB for Flash-SWA: reductions of 16.9% and 44.4%. All kernels use 128 threads/block; PBB launches two query-group blocks and SWA twelve query-head blocks. These resource attributes also match the independent rental-device launch traces. The new binary has zero stack and static local-memory allocation.

Global-load requests fall from SWA's 15,168 sectors/call to PBB's 1,112–2,120, a reduction of 86.0–92.7% at the sampled phases. The counter is `l1tex__t_sectors_pipe_lsu_mem_global_op_ld.sum`; it counts sector requests, including repeated requests, rather than unique source bytes. New PBB also records zero thread-local load/store requests, compared with 2,880/5,952 for SWA. Thread-local memory here is CUDA thread-private storage, not the attention KV cache; the SWA disassembly contains corresponding `LDL/STL` stack accesses, which are absent from the new PBB kernel.

New and previous PBB have identical global-load requests within attention. The new implementation instead reduces register/shared-memory allocation, removes the previous kernel's 32/32 thread-local load/store requests, and eliminates rollover relocation. The smaller tile, grouped dispatch, and ring addressing jointly produce the tested implementation; the request reduction versus SWA is not explained by edge count alone. The packaged observations preserve every phase and repeat, and regenerate the figure without a GPU.

L2 read and write diagnostics. L2 counters also favor Flash-PBB: median read-sector counts are lower at all six phases under both cache policies. Median write-sector counts are 80–81 for Flash-PBB versus 6,121 for Flash-SWA. Table 20 retains each phase's median and full three-run range. These are `lts__t_sectors_op_read.sum` and `lts__t_sectors_op_write.sum`, distinct from the L1/TEX requests reported in Table 19. Because chip-global L2 counters also include concurrent desktop activity, this table is diagnostic rather than an isolated estimate of kernel traffic or DRAM-bandwidth savings.

Cache	Phase	PBB reads	SWA reads	PBB writes	SWA writes
Cold	0	8,063 [7,813, 16,485]	24,262 [22,759, 41,837]	80 [80, 80]	6,121 [6,121, 6,172]
Cold	20	8,288 [8,006, 8,338]	27,228 [23,372, 41,698]	80 [80, 80]	6,121 [6,121, 6,121]
Cold	63	8,881 [8,703, 9,410]	23,228 [23,078, 24,195]	80 [80, 80]	6,121 [6,121, 6,121]
Cold	64	8,063 [7,968, 8,080]	28,953 [26,571, 32,765]	80 [80, 80]	6,121 [6,121, 6,172]
Cold	80	8,316 [8,204, 16,669]	26,572 [22,737, 27,933]	80 [80, 80]	6,121 [6,121, 6,121]
Cold	127	9,012 [8,704, 11,249]	26,957 [22,858, 30,435]	80 [80, 80]	6,121 [6,121, 6,121]
Retained	0	7,977 [7,867, 9,344]	23,284 [22,773, 27,508]	80 [80, 80]	6,121 [6,121, 6,121]
Retained	20	7,992 [7,990, 8,247]	22,696 [22,692, 27,731]	81 [80, 144]	6,121 [6,121, 6,122]
Retained	63	8,822 [2,214, 10,307]	25,832 [22,732, 27,235]	80 [80, 80]	6,121 [6,121, 6,168]
Retained	64	7,976 [7,669, 15,513]	22,718 [22,691, 22,720]	80 [80, 80]	6,121 [6,121, 6,121]
Retained	80	7,918 [1,469, 9,431]	29,668 [22,712, 36,323]	80 [80, 80]	6,121 [6,121, 6,163]
Retained	127	8,762 [8,681, 9,986]	22,696 [22,690, 30,192]	80 [80, 80]	6,121 [6,121, 6,122]

Table 20: **L2 sector counts during profiled attention: Flash-PBB versus Flash-SWA.** RTX 4060, 2B attention shape. Each entry is median [minimum, maximum] over three runs; all phases and both cache policies are shown. Chip-global diagnostic scope, including background activity.

F.4 Exact union and numerical checks

The added PBB edges form 64×64 target-by-source rectangles: 64 post-boundary queries each read 64 sources from the previous block. The fused kernel visits causal block tiles and these rectangles without constructing a dense sequence-square mask. Projections and KV heads are shared with the original GQA layer. Each source enters the single-softmax normalization once.

The RTX 4090 BF16 forward check compares PBB against exact union references at lengths 128, 256, and 1,024. Maximum output error is at most 0.015625, with cosine similarity above 0.999997. Its length-128 backward check covers the initial block: gradient cosine similarities exceed 0.999994, with maximum errors up to 0.03125. The multi-block check below additionally tests bridge gradients. At 8K, a separate LSE-specialization forward check reports identical outputs and finite log-partition values.

A supplementary multi-block check exercises the grouped4 backward implementation on an RTX 4060 Laptop GPU with Torch 2.11.0+cu128 and Triton 3.6.0. It uses batch 1, 12 query heads, two KV heads, head dimension 64, and BF16 inputs, with an independently constructed FP32 dense-union reference. Two random tensor seeds cover lengths 128, 256, and 1,024. At both multi-block lengths, additional cases restrict the upstream gradient to repaired queries 128–191 and inspect gradients of previous-block keys and values at positions 64–127. All ten cases pass the prescribed relative-L_2 threshold of 0.02 and cosine threshold of 0.9998. Across outputs and dQ, dK, dV, the largest relative error is 0.002562 and the smallest cosine is 0.9999966; maximum absolute dV error is 0.03741. The previous-block gradients have nonzero reference norm and relative error below 0.002416. The source package includes the kernel snapshot, executable check, and per-case results for this correctness experiment.

These BF16-versus-reference checks cover initial blocks, empty bridges, causal masks, and native GQA at the stated numerical tolerances. The CPU algebra checks separately verify 1,980 graph/depth configurations, the disjoint LSE identity, and exact PBB/WideBlock score equality.

The new training sweep uses the grouped4 PBB backward path and Triton grouped experts. All three architectures use the same compatibility adjustment for Triton 3.2: a loop-invariant tile-count constant is declared outside the expert loop. Sparse/empty-expert forward and backward checks precede timing; no architecture-specific change is made to expert arithmetic.

F.5 Shared hardware and scope

The primary execution matrix uses RTX 4090 (48 GB) GPUs with matched software and phase-specific benchmark settings. Per-run hardware records are retained in the accompanying numerical artifact. The model has 2,069,563,136 stored and 300,091,136 active parameters, 20 layers, hidden width 768, 12 query heads, two KV heads, and head dimension 64. Full has 20 Full layers; hybrids have 15 local and five Full layers at zero-based indices 3, 7, 11, 15, and 19. SWA has window 128; PBB has block 128 and bridge-exclusive width 64. All paths use native GQA without repeating KV heads.

The execution benchmarks use seeded, randomly initialized weights and synthetic device-resident inputs to isolate architecture execution from data I/O. Trained MoE routing and serving requests can produce different expert loads. Quality is evaluated separately on trained checkpoints at 8K; the longer-context sweeps measure execution speed and capacity under the specified runtime.

The rental sweeps use Torch 2.6.0+cu124, Triton 3.2.0, Transformers 5.10.2, and FlashAttention 2.8.3.post1, with eight CPU threads for whole-model runs and four for the new decode-core comparison. Processes start separately for each case. All 33 successful whole-model cells, including seven updated PBB inference cells, pass the recorded no-external-SM checks. Full/SWA and training data are unchanged; only PBB inference is replaced. Table 21 defines the protocols. Earlier laboratory-card measurements appear separately in Appendix G.

Protocol	State / expert backend	Work per measurement	Repetition
Context training	FP32 masters, BF16 AMP; Triton grouped experts	Micro-batch 1; complete update including clearing	Two warmups; five updates
Cached prefill	BF16; grouped-matrix-multiply expert dispatch	Full input, KV population, final-position logits	Three warmed calls at every successful length
Sequential decode	BF16; Triton grouped experts; CUDA Graphs for all models	256 incremental tokens, advancing positions and KV state	Three alternating eager/graph pairs; graph times reported
Attention layer	BF16; native GQA; kernel and complete-layer scopes	Prefill or 256 sequential layer steps	Five prefill calls and five decode continuations per scope

Table 21: Primary efficiency protocols on RTX 4090 (48 GB). AMP denotes automatic mixed precision. Each phase uses matched settings across architectures. Reported medians and min–max ranges summarize within-process repetitions.

F.6 Complete-update context sweep

The sweep fixes 131,072 input tokens per optimizer update and micro-batch 1 through 128K. At contexts 8K, 16K, 32K, 64K, and 128K, gradient accumulation is 16, 8, 4, 2, and 1. The 256K, 512K, and 1M attempts use one sequence per update as a separate capacity group. This fixed-batching protocol matches architectures at each length while exposing how context length changes attention cost and accumulation passes.

Parameters, gradients, and fused-AdamW moments are FP32 and GPU-resident. Computation uses BF16 autocast, non-reentrant activation checkpointing, the same Triton grouped MoE implementation, and fused vocabulary loss. AdamW uses learning rate 3×10^{-4}, betas $(0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, and gradient clipping at norm 1.0. Seeded initialization uses seed 13. Each timed update includes gradient clearing, forward computation, language-model and router auxiliary losses, accumulated backward, clipping, and the optimizer update. Device synchronization brackets timing;

setup, first-use compilation, data allocation, and validation are excluded. Synthetic inputs are reused across accumulation passes.

Two complete warmup updates precede five measured updates. Table 22 reports their median and observed range, totaling 60 timed updates across twelve successful cases. Speedup is the ratio of architecture and Full median throughputs at the same length.

Context	Architecture	GA	k tok/s [min, max]	s/update	Peak GiB
8K	Full	16	9.15 [9.06, 9.15]	14.33	32.93
8K	SWA+Full 3:1	16	9.79 [9.79, 9.80]	13.38	32.93
8K	PBB+Full 3:1	16	9.74 [9.68, 9.75]	13.45	32.93
16K	Full	8	12.01 [11.98, 12.02]	10.91	33.93
16K	SWA+Full 3:1	8	13.90 [13.76, 13.93]	9.43	33.93
16K	PBB+Full 3:1	8	13.96 [13.82, 13.98]	9.39	33.93
32K	Full	4	12.21 [12.20, 12.21]	10.74	36.38
32K	SWA+Full 3:1	4	16.58 [16.55, 16.61]	7.90	36.38
32K	PBB+Full 3:1	4	16.66 [16.64, 16.67]	7.87	36.38
64K	Full	2	9.74 [9.74, 9.74]	13.46	41.28
64K	SWA+Full 3:1	2	16.61 [16.60, 16.62]	7.89	41.28
64K	PBB+Full 3:1	2	16.70 [16.69, 16.72]	7.85	41.28
128K	All three architectures	1	OOM	–	–
256K	All three architectures	1	OOM	–	–
512K	All three architectures	1	OOM	–	–
1M	All three architectures	1	OOM	–	–

Table 22: Complete training-update sweep, all three architectures. GA is gradient accumulation; k tok/s uses $k = 1000$. Successful cells each contain five measured updates. All 128K–1M attempts fail: 128K during the timed stage and the longer cases during warmup; no failed cell contributes a throughput estimate.

At 64K, PBB+Full, SWA+Full, and Full reach 16.703k, 16.614k, and 9.738k tokens/s, respectively. PBB/Full is $1.715\times$, while PBB/SWA is $1.005\times$. Peak allocated memory is 41.283 GiB for all three, dominated by parameter, optimizer, and activation storage. All 128K cases complete two warmup updates, then fail on a 768 MiB allocation during the timed stage; each longer attempt also exhausts memory. Under this configuration, 64K is the largest context completing all five timed updates. Training memory covers the full process; persistent KV storage is measured separately in inference.

F.7 Exact sequential graph decoding

The cached runner uses matched initialization with seed 1309, BF16 weights/computation, and batch 1. Prefill creates KV state and returns last-position logits using the same grouped-matrix-multiply expert dispatch for all architectures. Before decode, every architecture switches to the same Triton grouped expert implementation. Each incremental step updates the cache, executes all layers, projects the output vocabulary, and takes a GPU argmax. No full-prefix recomputation is performed. PBB local layers use 128-slot rings; the tested SWA backend retains 256-position buffers. Full layers retain the input plus 385 positions reserved for checks and continuations. Allocation does not change either local pattern's logical support.

Prefill uses FlashAttention for Full/SWA and fused Triton for PBB. Decode uses the FlashAttention KV-cache API for Full/SWA and Flash-PBB's exact ring layout for PBB. A bank of 128 phase-specific CUDA Graphs handles local geometry and updates. Absolute RoPE positions and Full-cache lengths advance on device, and each graph consumes the preceding step's selected token. Graph replay includes projections, all expert layers, cache updates, output logits, and argmax, implementing the complete sequential generation step.

Every successful whole-model case checks 384 sequential positions against eager execution with the same low-precision expert backend. All full-vocabulary logits match bitwise, and the final absolute position and Full-cache length are explicitly checked. The span covers every local phase and three block rollovers, validating graph execution and state updates within each architecture.

F.8 CACHED TIMING AND CAPACITY RESULTS

Table 23 reports all 24 requested cached cases. After a separately recorded cold prefill, each successful case measures three warmed cache-populating calls. The repetition count is identical from 8K through 512K. Graph capture, correctness checks, state restoration, and cold-start work are excluded from steady-state throughput.

Following the graph correctness check, each case measures three alternating eager/graph pairs; the table reports the three graph continuations. Each restores the prefix state and generates the same greedy trajectory of 256 successive tokens. CPU wall time is bracketed by device synchronization; CUDA-event timings are retained separately. The context labels exclude the continuation. All three architectures complete 8K, 16K, 32K, 64K, 128K, 256K, and 512K. The packaged numerical records retain cold times, all repeats, parity outcomes, and memory scopes.

Context	Architecture	Prefill k/s [min, max]	Decode/s [min, max]	Peak GiB	KV GiB
8K	Full	106.63 [105.25, 107.75]	329.81 [326.02, 332.30]	4.24	0.082
8K	SWA+Full 3:1	110.55 [108.21, 110.90]	334.32 [334.00, 340.66]	4.18	0.022
8K	PBB+Full 3:1	111.18 [110.80, 111.24]	348.86 [346.98, 352.56]	4.18	0.021
16K	Full	93.96 [93.10, 94.13]	303.67 [302.67, 306.55]	4.61	0.160
16K	SWA+Full 3:1	118.00 [114.32, 118.04]	330.99 [330.45, 333.66]	4.49	0.042
16K	PBB+Full 3:1	118.07 [116.31, 119.77]	343.64 [337.62, 345.86]	4.49	0.041
32K	Full	71.14 [71.07, 71.18]	285.78 [284.54, 289.13]	5.35	0.316
32K	SWA+Full 3:1	105.84 [105.76, 105.85]	325.08 [322.13, 326.70]	5.12	0.081
32K	PBB+Full 3:1	106.39 [105.49, 106.89]	335.16 [333.28, 337.64]	5.12	0.080
64K	Full	50.09 [49.91, 50.11]	258.02 [257.79, 260.48]	6.84	0.629
64K	SWA+Full 3:1	92.49 [92.43, 92.54]	313.25 [312.59, 314.77]	6.37	0.159
64K	PBB+Full 3:1	93.39 [93.38, 93.44]	320.58 [318.48, 323.06]	6.37	0.158
128K	Full	31.26 [31.26, 31.39]	209.05 [208.46, 209.46]	9.81	1.254
128K	SWA+Full 3:1	73.27 [73.20, 73.32]	290.99 [290.40, 294.31]	8.87	0.315
128K	PBB+Full 3:1	73.87 [73.85, 73.91]	302.32 [301.12, 303.83]	8.87	0.314
256K	Full	17.64 [17.59, 17.71]	158.71 [158.41, 158.92]	15.75	2.504
256K	SWA+Full 3:1	50.67 [50.61, 50.73]	264.37 [263.57, 265.73]	13.87	0.628
256K	PBB+Full 3:1	51.18 [51.08, 51.23]	269.81 [269.66, 273.71]	13.87	0.627
512K	Full	9.13 [9.13, 9.13]	105.11 [105.09, 105.12]	27.63	5.004
512K	SWA+Full 3:1	30.39 [30.34, 30.44]	218.34 [217.43, 218.83]	23.88	1.253
512K	PBB+Full 3:1	30.31 [30.21, 30.47]	221.87 [221.58, 223.74]	23.88	1.252
1M	Full	OOM	–	–	–
1M	SWA+Full 3:1	OOM	–	–	–
1M	PBB+Full 3:1	OOM	–	–	–

Table 23: Cache-populating prefill and sequential graph decode, complete context matrix. Successful rates are medians over three measurements, with observed ranges; decode continuations contain 256 tokens. KV is physical cache storage, excluding graph workspace. OOM denotes an attempted prefill that exhausted memory.

At 512K, PBB+Full records 30.307k prefill tokens/s versus Full's 9.130k ($3.319\times$). Median sequential decode is 221.873 versus 105.106 tokens/s ($2.111\times$). SWA+Full records 30.386k prefill and 218.340 decode tokens/s. PBB/SWA hybrids allocate 1.25183/1.25275 GiB of KV versus Full's 5.00367 GiB, because only five layers retain the long global cache (Figure 13). Halving PBB's 15 local buffers saves 983,040 additional bytes, small beside the global cache at this length.

At 1M ($1{,}048{,}576$) input tokens, Full fails during cold prefill on a 5.00 GiB allocation with 2.56 GiB free. Both hybrids fail on a 6.00 GiB allocation, with 4.11 GiB free for SWA and 4.13 GiB for PBB. Whole-model execution completes through 512K; the isolated-layer tests do not imply whole-model 1M capacity. Decode-process peaks include the graph pool and reference buffers in addition to model state.

F.9 EXECUTION TIME PER TOKEN

For measured throughput v tokens/s, the steady-state GPU time per million tokens is $10^6/(3600v)$ hours. Figure 14 converts the primary throughput measurements into execution-time costs. At 512K,



Figure 13: **Memory in the completed rental-card matrix.** Left: prefill peak allocated memory. Right: physical KV allocation, including continuation reserve. Hybrid curves nearly overlap. Graph-capture and decode-process peaks are separate from persistent KV storage.

PBB+Full reduces prefill time per token by 69.9% and decode time per token by 52.6% relative to Full. The 64K training-update ratio corresponds to 41.7% less time at fixed tokens/update. These are steady-state costs under the specified protocols; a request's combined cost depends on its input and output lengths.

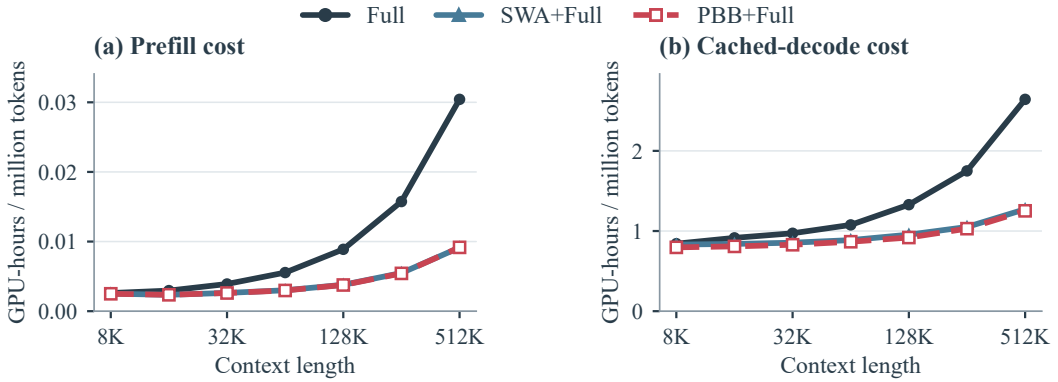

Figure 14: **Steady-state inference time per token.** Costs are inverse median throughput from the primary execution matrix. Prefill uses input tokens; decode uses output tokens and excludes prefill and capture. Lower is better. Hybrid curves nearly overlap.

F.10 Historical projected-layer comparison

The earlier layer matrix uses the previous 256-slot Flash-PBB implementation, not Ring128. It remains a reference separating attention from projection costs; the new core comparison is in Appendix F.2. It tests Full, Flash-SWA, and PBB from 8K to 1M with the MoE attention dimensions: batch 1, hidden width 768, 12 query heads, two KV heads, head dimension 64, and BF16. All 24 cases complete, yielding 96 measurements across prefill/decode and two scopes. The *kernel* scope includes attention dispatch, KV append, local compaction, and device-position updates. The *complete attention layer* additionally includes Q/K/V projections, RoPE, and output projection, but not RMSNorm, residual addition, experts, or vocabulary projection.

Prefill uses five warmed calls per scope. Decode uses five 256-step sequential layer continuations with phase-varying synthetic inputs and CUDA Graphs. Every one of the 128 phases is visited twice per continuation. Each scope first passes a 384-step bitwise check against its eager layer reference, including three cache rollovers. Figure 15 separates the scopes; Table 24 retains the ratios, and Tables 25–28 the wall-time medians.

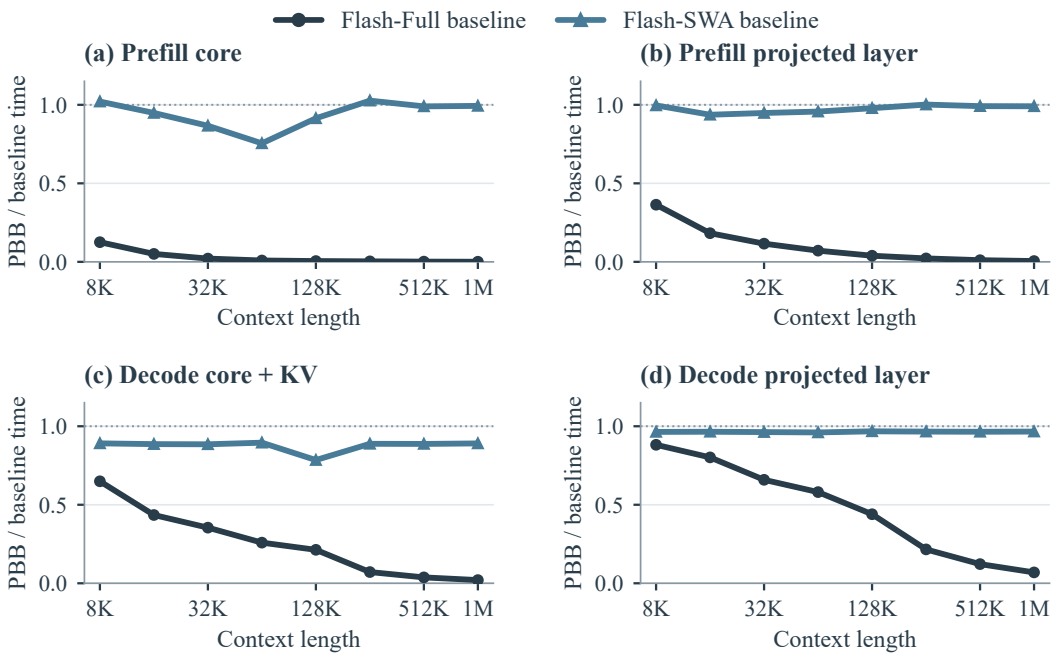

Figure 15: **Previous Flash-PBB implementation, kernel and projected-layer scopes.** Time relative to Full and Flash-SWA; lower is faster. Each point is a ratio of five-measurement medians using synthetic states. Ring128 is measured separately in Figure 5.

Context	Prefill core	Prefill complete	Decode core	Decode complete
8K	0.980	1.004	1.122	1.037
16K	1.055	1.067	1.128	1.036
32K	1.154	1.055	1.130	1.039
64K	1.325	1.045	1.117	1.041
128K	1.094	1.022	1.274	1.033
256K	0.974	0.998	1.126	1.035
512K	1.010	1.009	1.127	1.036
1M	1.007	1.009	1.123	1.035

Table 24: Single-layer PBB/Flash-SWA throughput ratios across the full context grid; larger is faster. Columns distinguish prefill/decode and core/complete-layer scopes. All cells complete. Ratios are reciprocals of the corresponding time ratios in Figure 15.

PBB decode-core throughput exceeds Flash-SWA at every measured length: by 11.7–13.0% at seven lengths and 27.4% at 128K. Including the shared projection and RoPE work gives complete-layer gains of 3.3–4.1%. Both local supports remain bounded as the prefix grows. Prefill-core PBB/Flash-SWA ratios range from 0.974 to 1.325, with the largest advantage at 64K and small regressions at 8K and 256K.

Against Full at 1M, the complete PBB attention layer takes 57.30 ms versus 11,278.11 ms for prefill and 45.20 versus 653.49 μs per sequential decode step: $196.84\times$ and $14.46\times$ layer-level speedups. Whole-model measurements through 512K appear in Figure 11, including the five Full layers and non-attention modules in each hybrid.

Each local layer allocates 256 KV positions, or 131,072 bytes with two 64-dimensional BF16 KV heads. Within this common allocation, Flash-SWA retains 127 logical past sources at each readout and PBB uses 64–127 over the tested steady-state phases, before including the current token. Full's buffer contains input length plus 385 positions. Reported layer memory peaks additionally include reference and benchmark fixtures.

Context	Attention	Prefill ms [min, max]
8K	Flash-Full	0.98 [0.96, 1.12]
8K	Flash-SWA	0.12 [0.12, 0.24]
8K	PBB	0.12 [0.12, 0.33]
16K	Flash-Full	3.23 [3.13, 3.26]
16K	Flash-SWA	0.17 [0.17, 0.31]
16K	PBB	0.16 [0.15, 0.44]
32K	Flash-Full	11.82 [11.71, 12.08]
32K	Flash-SWA	0.28 [0.28, 0.42]
32K	PBB	0.25 [0.22, 0.66]
64K	Flash-Full	44.25 [43.82, 44.44]
64K	Flash-SWA	0.50 [0.50, 0.65]
64K	PBB	0.38 [0.38, 0.61]
128K	Flash-Full	161.12 [160.86, 161.55]
128K	Flash-SWA	0.89 [0.88, 1.02]
128K	PBB	0.81 [0.80, 1.04]
256K	Flash-Full	639.46 [638.97, 641.99]
256K	Flash-SWA	1.94 [1.94, 2.13]
256K	PBB	1.99 [1.98, 2.26]
512K	Flash-Full	2671.31 [2651.86, 2680.67]
512K	Flash-SWA	3.82 [3.81, 3.97]
512K	PBB	3.78 [3.77, 4.00]
1M	Flash-Full	11174.10 [11107.23, 11193.26]
1M	Flash-SWA	7.53 [7.53, 7.71]
1M	PBB	7.48 [7.47, 7.90]

Table 25: Single-layer prefill, attention-core scope. Wall-time medians and observed ranges over five calls, in milliseconds.

F.11 Execution-series index

Table 29 indexes the execution series of this appendix and of Appendix G by scope, batching, hardware, and repetition.

Context	Attention	Prefill ms [min, max]
8K	Flash-Full	1.44 [1.35, 1.51]
8K	Flash-SWA	0.53 [0.51, 0.65]
8K	PBB	0.52 [0.51, 0.65]
16K	Flash-Full	3.82 [3.76, 3.85]
16K	Flash-SWA	0.74 [0.74, 0.76]
16K	PBB	0.70 [0.70, 1.33]
32K	Flash-Full	13.12 [13.04, 13.20]
32K	Flash-SWA	1.60 [1.60, 1.62]
32K	PBB	1.52 [1.52, 1.54]
64K	Flash-Full	43.89 [43.88, 44.19]
64K	Flash-SWA	3.27 [3.26, 3.30]
64K	PBB	3.13 [3.12, 3.15]
128K	Flash-Full	166.98 [166.94, 167.09]
128K	Flash-SWA	6.66 [6.64, 6.71]
128K	PBB	6.52 [6.50, 6.54]
256K	Flash-Full	657.53 [656.18, 657.88]
256K	Flash-SWA	14.37 [14.32, 14.40]
256K	PBB	14.40 [14.34, 14.44]
512K	Flash-Full	2713.55 [2708.23, 2718.89]
512K	Flash-SWA	28.88 [28.80, 29.00]
512K	PBB	28.64 [28.56, 28.90]
1M	Flash-Full	11278.11 [11272.08, 11292.66]
1M	Flash-SWA	57.83 [57.52, 58.23]
1M	PBB	57.30 [57.21, 57.48]

Table 26: Single-layer prefill including projections and RoPE. Wall-time medians and observed ranges over five calls, in milliseconds.

Context	Attention	Decode μs/token [min, max]
8K	Flash-Full	19.24 [19.23, 19.37]
8K	Flash-SWA	14.03 [14.02, 14.17]
8K	PBB	12.50 [12.49, 12.63]
16K	Flash-Full	28.65 [28.64, 28.82]
16K	Flash-SWA	14.06 [14.05, 14.19]
16K	PBB	12.46 [12.45, 12.60]
32K	Flash-Full	35.15 [35.09, 35.30]
32K	Flash-SWA	14.07 [14.06, 14.23]
32K	PBB	12.45 [12.45, 12.59]
64K	Flash-Full	48.58 [47.58, 48.68]
64K	Flash-SWA	14.04 [14.04, 14.20]
64K	PBB	12.57 [12.46, 12.82]
128K	Flash-Full	67.27 [67.21, 67.44]
128K	Flash-SWA	18.27 [16.40, 21.01]
128K	PBB	14.33 [14.23, 14.57]
256K	Flash-Full	174.24 [174.00, 174.81]
256K	Flash-SWA	14.03 [14.01, 14.15]
256K	PBB	12.46 [12.45, 12.97]
512K	Flash-Full	331.46 [331.19, 331.72]
512K	Flash-SWA	14.03 [14.03, 14.15]
512K	PBB	12.45 [12.45, 12.57]
1M	Flash-Full	616.64 [616.27, 616.82]
1M	Flash-SWA	14.07 [14.07, 14.22]
1M	PBB	12.53 [12.53, 12.69]

Table 27: Single-layer sequential decode, attention and KV-update scope. Wall time per step over five 256-step continuations, in microseconds; median and observed range.

Context	Attention	Decode μs/token [min, max]
8K	Flash-Full	49.96 [49.96, 50.06]
8K	Flash-SWA	45.70 [45.68, 45.78]
8K	PBB	44.06 [44.04, 44.16]
16K	Flash-Full	54.95 [54.94, 59.59]
16K	Flash-SWA	45.65 [45.65, 45.76]
16K	PBB	44.06 [41.84, 44.22]
32K	Flash-Full	66.69 [66.68, 66.86]
32K	Flash-SWA	45.64 [45.58, 45.71]
32K	PBB	43.94 [43.93, 44.09]
64K	Flash-Full	75.52 [75.43, 78.81]
64K	Flash-SWA	45.65 [45.65, 45.78]
64K	PBB	43.86 [43.83, 44.08]
128K	Flash-Full	102.93 [102.88, 103.06]
128K	Flash-SWA	46.77 [46.74, 46.88]
128K	PBB	45.26 [45.16, 45.32]
256K	Flash-Full	209.40 [209.23, 212.59]
256K	Flash-SWA	46.77 [46.74, 46.87]
256K	PBB	45.18 [45.15, 45.23]
512K	Flash-Full	369.51 [369.20, 369.58]
512K	Flash-SWA	46.76 [46.72, 46.90]
512K	PBB	45.12 [45.12, 45.23]
1M	Flash-Full	653.49 [653.16, 654.44]
1M	Flash-SWA	46.79 [46.74, 46.83]
1M	PBB	45.20 [45.14, 45.25]

Table 28: Single-layer sequential decode including projections and RoPE. Wall time per step over five 256-step continuations, in microseconds; median and observed range.

Series	Scope	Batch	Hardware, kernel	Repetition	Reported
Whole-model matrix	Prefill with KV population; cached decode; complete update at 131,072 tokens	1	RTX 4090 (48 GB); Flash-PBB Ring128, FlashAttention 2.8.3.post1	3 warmed prefill calls; 3 eager/graph decode pairs; 2 warmups and 5 updates	Table 21; Figs. 11, 13, 14; Tables 22, 23
Batched decode sweep (F.2)	Decode attention core over all 128 phases, per step; complete 256-token continuation	1–64; model 1, 4, 16, 32, 64	RTX 4090; Flash-PBB Ring128, Flash-SWA	Core: median of 7 CUDA-graph samples; model: median of 3 runs	Fig. 5; Table 18
Resource profiling (F.3)	Attention kernel only, six sampled phases	1	RTX 4060 Laptop; Nsight Compute 2024.1.1, SM89	3 processes $\times$ 2 cache policies; 108 profiled attention calls	Tables 19, 20
Layer comparison (F.10)	Attention kernel and complete attention layer, 8K–1M	1	Rental card; previous 256-slot Flash-PBB	5 prefill calls; 5 decode continuations, per scope	Fig. 15; Table 24 and the four following
Prefill without KV cache (G.4)	Prefill that does not populate a persistent cache, 16K–128K	1	Laboratory card; three backend kernels	3 independent runs; 5–15 timed calls each	Fig. 16; Table 32
8K training recipe (G.3)	Complete training update, 393,216 tokens	2×24	Laboratory card; Triton grouped experts	1 warmup; 3 timed updates	G.3 text

Table 29: Execution-series index. Series differ in implementation version, scope, batching, hardware, and repetition, so timings are comparable within a row but not across rows. "Previous" and the laboratory/rental cards mark the historical implementations and devices of Appendices F.10 and G; parenthesized numbers are subsections of this appendix.

G EARLIER BRIDGE-FAMILY EXPERIMENTS

The development experiments below examine residual-branch Bridge models and frozen-checkpoint mask interventions. Their model families, fusion rules, and prompt protocols differ from the main experiments; their exploratory probe results precede the balanced source-rotation design in Appendix D and are reported separately. This section also retains earlier execution benchmarks under their original batching and timing protocols.

G.1 124M, 1024-TOKEN CONTROLLED TRAINING

The earlier GPT-2-like setting used learned absolute positions, a GPT-2 tokenizer, OpenWebText, 1,024-token context, and three training seeds (Radford et al., 2019; Gokaslan & Cohen, 2019). Bridge-family layers used separately normalized residual branches with shared projections. With block and auxiliary outputs o_1, o_2, the layer computes $W_O(o_1 + o_2)$, so a source in both supports participates in both normalizations. PBB limits auxiliary write-back to the post-boundary half; SE-Bridge extends the previous-block source range. Table 30 gives the reported aggregate measurements.

Attention	Validation PPL	Needle	Bridge-window	Clean semantic
Full	28.56	0.967	0.989	0.946
Block	36.79	0.429	0.415	0.392
Bridge	31.74	0.437	0.812	0.420
PBB	31.38	0.438	0.842	0.451
SE-Bridge	30.84	0.436	0.983	0.516
SWA	30.56	0.513	0.991	0.493

Table 30: Historical 124M/1K results under the residual-branch training and probe protocol; the PBB row is the separately normalized residual-branch version, not the single-softmax PBB of the main text. Bridge-family entries use shared-projection branches. Probe columns are accuracies in $[0, 1]$, not percentages; PPL is validation perplexity.

SWA led the local models on language modeling and standard needle retrieval (Kamradt, 2023), while wider-source SE-Bridge recovered the boundary-window deficit. Clean semantic retrieval was 0.516 for SE-Bridge and 0.493 for SWA at the evaluated checkpoints. This pattern motivated separating source coverage, write-back geometry, and fusion in subsequent experiments.

G.2 FROZEN QWEN2.5-7B, 8K HYBRID INTERVENTION

The earlier 7B experiment imposed masks on a pretrained Qwen2.5 base checkpoint, retaining its weights and GQA projections without fine-tuning. Every fourth layer remained Full, and sparse PBB/SE-Bridge layers used a single-softmax union implemented with a dense mask. Evaluation ranked candidates on 832 prompts; the overall score in Table 31 weights task families by their displayed example counts.

Probe	Examples	Block+Full	SWA+Full	PBB+Full	SE+Full
Needle	64	0.313	0.547	0.578	0.578
Bridge-window	312	0.458	0.583	0.599	0.590
Semantic boundary	312	0.401	0.471	0.455	0.446
Semantic exact	48	0.375	0.521	0.479	0.500
Paraphrase	48	0.313	0.500	0.458	0.458
Distractor	48	0.000	0.167	0.146	0.104
Weighted overall	832	0.386	0.506	0.502	0.494

Table 31: Historical 7B frozen-checkpoint hybrid intervention, 3:1 local/full; a mask intervention on an unmodified pretrained checkpoint, reported separately from the main experiments. Probe entries are accuracies in $[0, 1]$, not percentages. Displayed values are rounded from the original result table.

The frozen-checkpoint comparison separates literal/boundary retrieval from semantic behavior: PBB+Full leads SWA+Full on needle and bridge-window retrieval, while SWA+Full leads on the

semantic variants. Their task-weighted aggregate accuracies are close, with gains concentrated in different task families.

G.3 Earlier 8K recipe benchmark

The earlier optimized benchmark uses the training-run batch configuration: micro-batch 2, accumulation 24, and 393,216 tokens/update. Both architectures use Triton grouped experts, FP32 parameters/gradients/Adam moments, BF16 autocast, activation checkpointing, and no offload or state sharding. The runtime is Torch 2.11.0+cu128 with FlashAttention 2.8.3.post1. Gradient clearing occurs outside this timed region; forward, loss, accumulated backward, clipping, and optimizer update are included.

After one warmup, Full update times are 24.8275, 26.0002, and 31.6251 seconds; PBB+Full times are 22.3821, 22.4117, and 22.4044 seconds. Their medians yield 15,123.6 and 17,550.8 tokens/s (+16.05%). Both record 33.932 GiB peak allocated and 35.072 GiB reserved memory. This batching configuration is reported separately from the micro-batch-1 context sweep in Figure 11d.

G.4 Earlier prefill without persistent KV caching

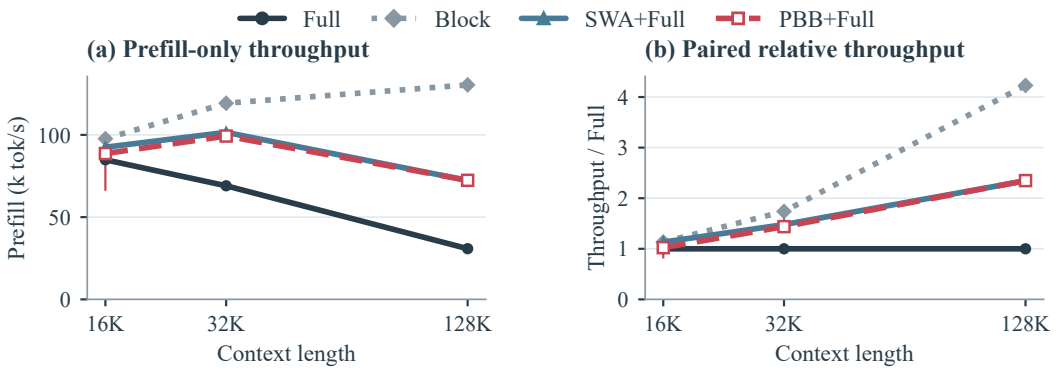

Figure 16: Prefill without KV-cache population: median and observed range across three independent runs.

The three independent laboratory-card prefill runs use BF16 parameters and computation, batch 1, and the same grouped-matrix-multiply MoE implementation for every architecture. Full uses causal FlashAttention, SWA uses the Flash window kernel, and PBB uses fused union attention. A timed call processes the full input and projects the final hidden position to vocabulary logits, without populating a persistent KV cache.

Each run uses three warmup and 15 timed calls at 16K, two warmup and ten timed calls at 32K, and one warmup and five timed calls at 128K. Three independently launched summaries provide the median and observed min–max ranges in Table 32. Speedup is paired within each run before summarization. Before each case the launcher requires ten idle observations with zero GPU utilization and low-clock P8 state; an unrelated inactive 4,252 MiB allocation remains on that device. Only records passing these occupancy checks enter the table.

Tokens	Architecture	Median tok/s	[Min, max]	Paired speedup
16,384	Full	84,858	[81,689, 87,423]	1.000×
16,384	Block	97,651	[96,244, 100,636]	1.134×
16,384	Flash SWA+Full 3:1	92,496	[87,186, 97,330]	1.132×
16,384	PBB+Full 3:1	88,655	[66,006, 89,400]	1.023×
32,768	Full	69,050	[68,573, 69,065]	1.000×
32,768	Block	119,239	[117,914, 120,480]	1.739×
32,768	Flash SWA+Full 3:1	101,527	[98,687, 102,100]	1.479×
32,768	PBB+Full 3:1	99,304	[98,174, 101,935]	1.438×
131,072	Full	30,818	[30,793, 30,825]	1.000×
131,072	Block	130,376	[129,940, 130,553]	4.230×
131,072	Flash SWA+Full 3:1	72,275	[72,111, 72,334]	2.345×
131,072	PBB+Full 3:1	72,379	[72,132, 72,383]	2.348×

Table 32: Prefill-only matrix across three independent runs. Throughput is tokens/s; speedup is paired against Full. All architectures use the same initialized 2B MoE dimensions. Block is pure local; SWA+Full and PBB+Full are 3:1 hybrids.