JustFit v2: Long-Context Local LLM Serving with Just-in-Time State Management

Yuhua Chen
yuhuachen.com

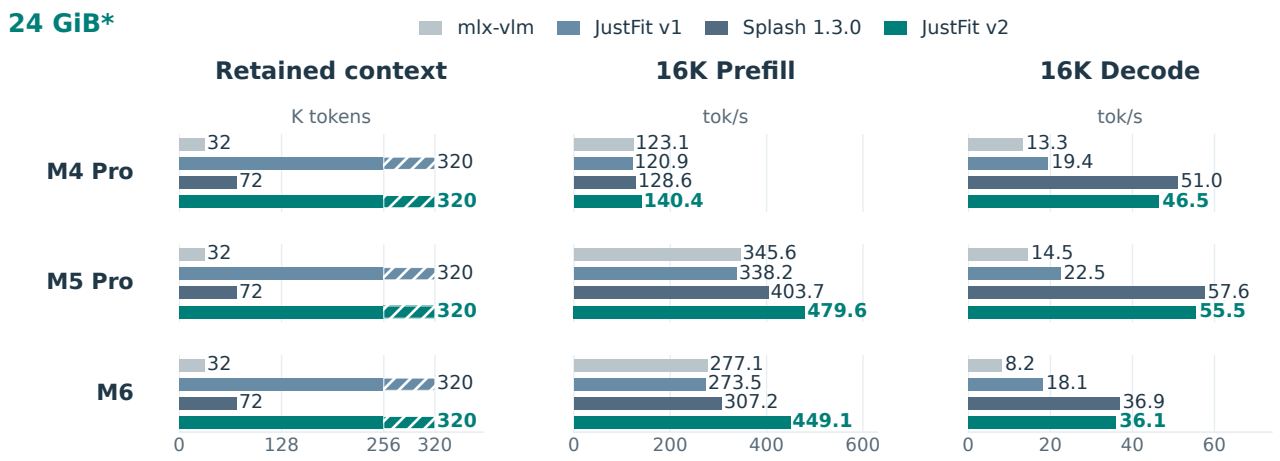


Figure 1. **Long histories with a live speculative draft.** Both JustFit generations reach a tested 320K B2 page capacity; v2 retains its draft during concurrent decoding. Solid bars show tested B1 bounds; hatching extends them to the separately tested B2 total. Speed uses eight fixed SPEED-Bench 16K tasks and 512 output tokens. V2 prefill is 9–46% faster than Splash 1.3.0; singleton decode is 2–9% slower. Capacity and speed are separate tests. *M6 has physical 32 GiB; 8 GiB is locked for its v1/v2 B2 tests only. Capacity, speed and historical-baseline protocols are specified in Section 8.1.

Abstract

Long-context local serving must fit more than a model: histories, a speculative draft, image encoding and concurrent requests compete for the same unified memory. JustFit v2 makes this live set affordable through four cooperating mechanisms: KVExec executes compressed history with bounded reconstruction and rotated-domain attention; PhaseSwap controls transient residency while keeping the draft live; StateTrans preserves committed composite state across scheduling and restart; and CrossFire shares work across CPU, GPU and Apple Neural Engine. Warm-prefix tests retain 320K aggregate page capacity across two live requests with a resident draft, preserving v1's capacity on two physical 24 GiB Macs and a 32 GiB system with 8 GiB locked. Across eight fixed-output 16K tasks, median decode is 2.0–2.5× v1. Against Splash 1.3.0, prefill improves by 9–46%, with a 2–9% singleton-decode cost. At 32K, four-request aggregate decode reaches 111.28 tokens/s on M5 Pro and 81.30 on M6, up from 56.84 and 34.58 at one request; M4 Pro largely plateaus. ANE vision offload reduces the measured maximum text delivery gap from 1.70 to 0.43 seconds at increased image latency, and cold-restart checks restore committed KV and recurrent state on all three devices. Together, these results show how coordinating state representation, ownership and execution extends local serving from fitting weights to sustaining long histories and concurrent work.

Contact: contact@yuhuachen.com. Code: https://github.com/YuhuaBillChen/justfit-llm.

1. Introduction

A local agent spends much of its life between requests: retaining a document, waiting for a tool, answering a follow-up, or accepting an image while another answer is already streaming. A quantized model that loads successfully need not support this service. Target weights, the speculative draft, compressed KV, recurrent state, visual features, and temporary matrices can still overlap beyond the physical memory available to the application.

Splash 1.3.0 provides model-specialized Metal execution and the DFlash2 learned draft (Inco AI, 2026b;a). Its evaluated four-bit Qwen3.8-27B package documented a 36 GB minimum and a 48 GB recommendation (Inco AI, 2026e;d). JustFit v2 targets a 24 GiB Mac with this target/draft package. Its central design question is how to retain long histories and serve concurrent work while keeping fast speculative decoding available.

JustFit v1 established three principles in MLX (Chen, 2026; Hannun et al., 2023): compact history should be expanded only for its next consumer (KVExec); components should remain attached only while an owner needs them (PhaseSwap); and execution-mode changes should preserve committed progress (StateTrans). Its concurrent policy used autoregressive decoding after batched speculation proved unattractive in the studied configurations. V2 changes the constraint: the speculative draft stays live beside concurrent histories. Memory must be recovered from execution intermediates, component overlap, and idle state without discarding the context needed by the decoder.

CrossFire extends these lifetime principles to heterogeneous execution: CPU/GPU column partitioning accelerates eligible prefill operations, while ANE vision encoding can overlap GPU text work. Language-model ANE execution joins the same phase and resource policy. The resulting design treats compute placement and memory ownership together, allowing each accelerator to contribute when its work fits the current live set.

The contributions are:

- **KVExec: representation-aware native execution.** Page-native TQ4, bounded head reconstruction, rotated-domain verification, and layout-aware staging retain compressed-history savings at the computation boundary.
- **PhaseSwap: residency with a live draft.** Component leases, mapped host embedding, reused arenas, and a shared verification-row budget control peak overlap while preserving concurrent speculative execution.
- **StateTrans: committed continuity.** Layered scheduling and composite-state publication, persistence, and restoration preserve valid progress. Ancestor retention makes earlier committed histories available after continuation and restart.
- **CrossFire: budgeted heterogeneous service.** CPU/GPU partitioning and full-tower ANE vision make otherwise idle compute available when shapes, phase, and resource admission permit it, with GPU fallback.

Figure 1 connects the design to system-level outcomes: JustFit v2 preserves v1's tested 320K aggregate retained capacity with a live speculative draft and delivers 2.0–2.5× its median decode rate on the fixed-output workload. The evaluation then explains the execution and residency choices, concurrent scaling, image interference, restoration and task outcomes. Table 8 separates whole-system comparisons from mechanism studies and functional checks.

2. One Budget, Different Lifetimes

The live set can be written as

$$M(t) = W_T(t) + W_D(t) + K(t) + S(t) + X(t) + V(t) + H(t). \quad (1)$$

where target and draft weights, retained KV, recurrent/speculative state, temporary workspaces, visual work, and host/allocator storage share physical unified memory. CPU mappings share this physical budget; SSD storage extends retained history between uses, while active attention consumes resident state.

The target interleaves 16 full-attention layers with 48 Gated DeltaNet layers (Yang et al., 2024). Each attention layer has four KV heads of dimension 256. TQ4 stores 128 bytes of indices and a two-byte norm per K or V vector, giving

$$\beta = 16 \times 2 \times 4 \times (128 + 2) = 16{,}640 \text{ B/token}. \quad (2)$$

FP16 KV would occupy 16 GiB at 256K retained positions, compared with 4,160 MiB of TQ4 payload. These figures exclude recurrent state, the draft, allocator effects, and execution workspaces.

The evaluated native cache uses 32-token pages (532,480 bytes per page), unlike v1's 256-token pages. A single 128K chain already needs 2.03125 GiB of native KV payload, before its composite state and metadata. This explains why a 2 GiB persistent-cache quota could not qualify full 128K restoration. A larger SSD budget removes that capacity obstruction, but cannot repair a missing state snapshot. Section 8.7 separates those causes experimentally.

3. KVExec: Execute in the Right Representation

3.1. From v1 to Native Pages

V1 used TurboQuant's normalized rotated MSE representation (Zandieh et al., 2026), page-addressed reconstruction, and stock floating attention consumers. Fusing page lookup, unpacking, centroid reconstruction, inverse rotation, and norm restoration avoided a sequence of temporary packed and floating arrays. Explicit evaluation prevented lazy operands from surviving across many layers. Later v1 head grouping reconstructed two of four KV heads at a time.

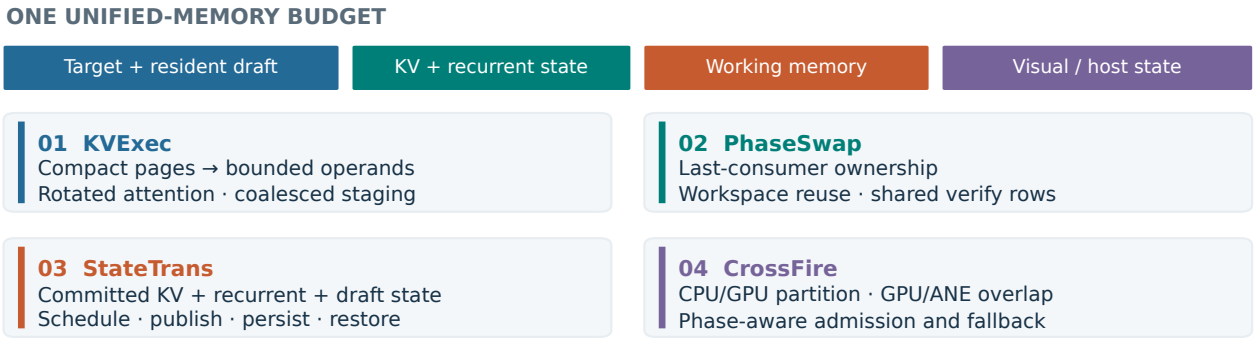


Figure 2. Four components, one memory budget. KVExec, PhaseSwap, and StateTrans extend v1's principles into a native speculative runtime. CrossFire adds heterogeneous execution. The four mechanisms coordinate the same live set through representation, residency, committed state and execution placement.

V2 integrates compact pages with the native allocator and Metal command graph. Decode and short verification consume packed history; long prefill reconstructs a K/V head into a BF16 staging plane and reuses it across query tiles. At 256K history, one head's K/V pair is 256 MiB; staging all four together would require 1,024 MiB. Explicit dependencies complete the old reader before that plane is reused by another head or layer. This calculation bounds the attention operand; process-level measurements include the rest of the live set.

3.2. Rotated-Domain Attention

Let the stored reconstructed key be $\widehat{k}_t = R_K^{\mathsf{T}} k'_t$, where R_K is orthogonal and k'_t includes centroid and norm reconstruction in the stored domain. With a separate value transform R_V,

$$\alpha_t = \mathrm{softmax}_t\Big((R_K q)^{\mathsf{T}} k'_t/\sqrt{d} + m_t\Big), \tag{3}$$

$$o = R_V^{\mathsf{T}} \sum_t \alpha_t v'_t. \tag{4}$$

The query rotates once and the weighted output returns once. The implementation avoids undoing the transform for every historical vector. Staged-head prefill and rotated verification are complementary choices: the former amortizes reconstructed operands over many query tiles, while the latter changes the domain of a short query's traversal.

The numerical validation follows the transformation from operator to model: identical-input references check low-precision arithmetic, fixed-prefix full-vocabulary comparisons check logits, and paired tasks check model outcomes. Historical M5 Pro real-model ROT0/1 repeats at 64K reduce verification-cycle time from about 145.0 to 120.9 ms. Section 9 distinguishes these mechanism measurements from final-profile attribution.

3.3. Layout-Aware Kernels

The coalesced V path changes the mapping from compressed token-major codewords to dimension-major staging. Adjacent lanes write adjacent token positions instead of widely separated addresses. Its selected comparison preserves the arithmetic and codebook. A historical M5 Pro 32K+6K experiment reports native TG 85.01 versus 95.88 tokens/s, with a 95.91 repeat; these measurements identify the benefit in that historical configuration.

The layout issue can be expressed without changing the quantizer. Let the staging plane be $V[d, t]$ with token stride one. Threads traversing d at a fixed t write addresses separated by the history length, whereas threads traversing t at fixed d write adjacent locations. Reassigning the reconstruction work to match the latter pattern changes transaction organization while retaining the same centroids, norms and output elements. The consumer's head/dimension mapping must remain consistent. Byte comparisons and unchanged acceptance distinguish this layout intervention from a precision change.

A separate compact softmax adapts upstream Page32 thread organization to staged TQ4 attention. Tile shape, query reuse, and barrier cost are evaluated rather than inferred from instruction count. Appendix A summarizes the mechanisms and measured alternatives.

3.4. Codec, Page Addressing, and Consumer Contracts

For a nonzero d-dimensional vector x, let $\rho = \|x\|_2$, H_d be an unnormalized Hadamard matrix and S a diagonal random-sign matrix. The normalized rotation and recon-

Table 1. Representation parameters for the evaluated hybrid model. Page payload includes all attention layers, both K/V and all KV heads.

Quantity	Value
Full-attention / recurrent layers	16 / 48
KV heads / head dimension	4 / 256
Q4 indices + norm per vector	128 + 2 bytes
KV payload per retained token	16,640 bytes
MLX / native page length	256 / 32 tokens
MLX page payload	4,259,840 bytes
Native page payload	532,480 bytes

struction are

$$y = Rx/\rho, \qquad R = H_d S/\sqrt{d}, \tag{5}$$

$$z_j = \arg\min_{k\in\{0,\dots,15\}} |y_j - c_k|, \tag{6}$$

$$\widehat{x} = \widehat{\rho} R^{\mathsf{T}} c[z]. \tag{7}$$

The nonuniform scalar centroids and midpoint thresholds are shared. Norm computation uses FP32, its stored value uses FP16, and eight four-bit indices occupy one 32-bit word. K and V use separate random signs. The two-byte norm explains why the effective reduction from FP16 is 3.94×, rather than exactly four. Quantization error comes from the centroid approximation and stored norm; execution error additionally depends on reconstruction and accumulation arithmetic.

For logical token t in a page of length P, the page-table lookup selects the physical page at $\lfloor t/P \rfloor$, and $t \bmod P$ selects the row. Sharing a prefix shares physical pages through references; allocating a continuation need not duplicate its prefix. The attention consumer must respect logical length and causal masking even when the final physical page is only partially occupied.

Storage and expansion have different scopes. At 256K positions, retained TQ4 across all attention layers occupies 4,160 MiB. Reconstructing all four KV heads for just one layer requires another 1,024 MiB in BF16. Limiting the expanded consumer set to one head bounds this operand to 256 MiB (Figure 3). This bound holds only if preceding consumers complete before their storage is reused; deferred execution can otherwise keep several generations of operands reachable.

A fused reconstruction kernel performs page lookup, nibble unpacking, centroid lookup, inverse rotation and norm restoration into the final consumer operand. This writes the floating consumer operand directly, eliminating intermediate arrays. The MLX predecessor also forces evaluation before progressing to the next head group or layer. In the native path, command dependencies perform the equivalent lifetime enforcement.

For a query block with many rows, staging a head once avoids repeating reconstruction for each query tile. For a short speculative query, that amortization is weaker: rotated-domain attention keeps history in its stored domain and rotates the query instead. Consumer geometry selects the appropriate path. Masks, head mapping, and the state boundary remain identical requirements on either path.

4. PhaseSwap: Residency with a Live Draft

4.1. Last-Consumer Ownership

In v1, an untied LM head (644.14 MiB in the archived layout) could detach during intermediate prefill but had to return before logits. A generation owner retained the head through mixed prefill and decode. Vision used a separate media lease: encode, evaluate embeddings, and release the tower before ordinary text prefill. The singleton MTP predictor was released when the system switched to concurrent AR.

V2 retains target and draft during the studied concurrent mode. A mapped host-embedding gather reduces Metal residency without changing the unified-memory budget. Visual weights and optional prefill storage are admitted when needed. Nonoverlapping GDN packed/hidden workspaces share backing with later staging users; a separately measured arena change reduced allocation by 89 MiB. Packed and hidden storage remain separate when the gate simultaneously reads one and writes the other. Decode-owned RoPE constants survive the release of prefill-only arenas.

Elastic workspace releases prefill storage at its last consumer. Its initial isolated screen left capacity and throughput unchanged, illustrating the role of the limiting phase: a local allocation reduction helps the whole workload only when it reduces the peak live set.

4.2. Lifetime and Ownership Invariants

Let $O_c(t)$ be the owners requiring component c, and let $a_c(t)$ denote whether its evaluated storage is attached. Residency obeys

$$|O_c(t)| > 0 \implies a_c(t) = 1. \tag{8}$$

An owner outlives submission: an asynchronous reader can still require a component after its encoding function returns. A release therefore waits for its last consumer, not merely the end of the host function. Generation-bound owner identifiers prevent an old completion callback from releasing a new cohort's lease.

Host embedding gather illustrates the distinction between placement and lifetime. The native table is mapped read-only on the host and selected rows are gathered into shared input storage. This removes 715,177,984 bytes from the studied Metal fixed allocation, while the mapping still be-

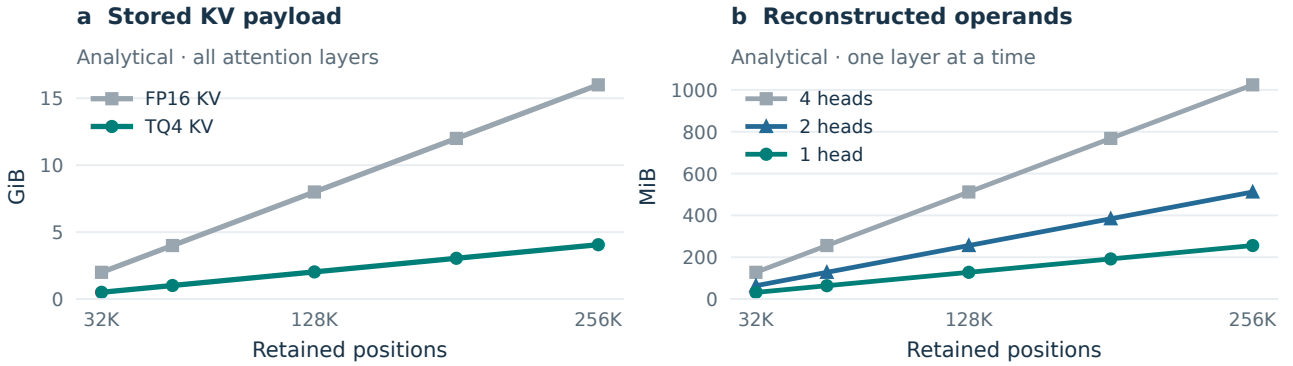


Figure 3. Analytical payload bounds. Left: attention KV across all 16 layers. Right: temporary BF16 K/V operands for one attention layer with four KV heads, two-head grouping, or one-head staging. Weights, recurrent state, allocator overhead and other workspaces are excluded.



Figure 4. Three consumers of the same packed history. The prefill path amortizes one head reconstruction across query tiles; short verification instead moves the query/output transforms around attention. Boxes show data and operations.

longs to the same physical memory budget. Gather completion must precede the target consumer. The unconstrained speculative path may require a draft/verify command split because proposal IDs are not known before drafting completes.

The host gather reproduces quantized row reconstruction and BF16 round-to-nearest-even behavior; its archived test compares 777 rows at each of two hidden sizes against the corresponding Metal embedding kernels. A reduced Metal allocation is thus paired with an arithmetic check, rather than inferred from an address-space change.

Workspace aliasing has an equally strict overlap test. A GDN packed input and its output cannot share storage while a gate reads one and writes the other. Later staging buffers can reuse backing only when their earlier consumers finish. Decode-owned RoPE constants must remain available even after prefill-only arenas are released. These ownership constraints determine the safe aliasing graph.

The predecessor's untied output head is a different object from the native token embedding: its 644.14 MiB can be detached during eligible intermediate prefill, but must return for logits. An incumbent decoder keeps its head lease while another request prefills. Conflating these components would incorrectly count the same phase-local saving as permanent capacity.

An elastic arena helps capacity when its lifetime overlaps the limiting peak. In its initial cold two-lane experiment, the second request still needed prefill storage at the limiting instant, and the binding constraint moved to host memory. Its mixed-arrival run reduced Metal peak by about 136 MB but did not improve that cold capacity boundary. This illustrates why the live-set maximum must be evaluated over the entire workload.

4.3. Bounded Verification Rows

Concurrency and speculation share a row budget. The shared-row design enforces

$$br \le 32, \qquad r \in \{8, 16\}. \tag{9}$$

A full B4 cohort uses eight rows per lane; B1/B2 can use sixteen. Scratch is sized by packed rows and lifetimes, rather than allocating sixteen rows independently for every lane.

Table 2. Resource effects from separately scoped measurements or analytical bounds. Rows are not additive reductions of a common process peak.

Mechanism	Quantity and scope	Before	After	Interpretation
Head grouping	BF16 operand, 192K (MiB)	768	384	Two of four heads at once
Head grouping	Synthetic MLX incremental peak, 192K (MiB)	3123.8	1589.3	Measured; latency +3.6%
Borrowed checkpoint	Synthetic MLX incremental peak, 224K (MiB)	3640	227.5	Cache-only save workload
Native staged head	BF16 operand, 256K (MiB)	1024	256	Analytical one-layer bound
Host embedding	Metal table residency (MiB)	682.047	0	Host mapping remains
Verification rows	Shared A8 workspace (MiB)	425.055	220.098	64-row versus 32-row allocation

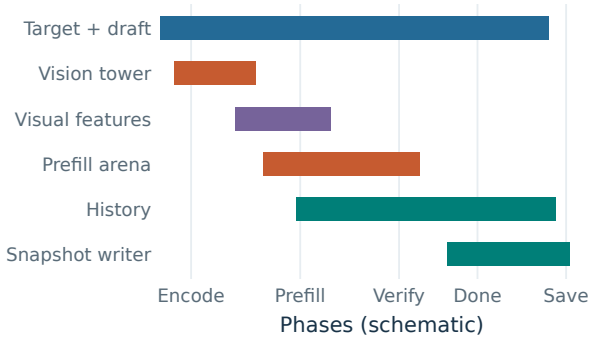


Figure 5. Conceptual resource lifetimes for an image-bearing request. Target/draft residency spans generation; vision and prefill storage end at their last consumers. A composite snapshot can outlive request completion through its own writer ownership. The native draft stays resident; the predecessor's concurrent-AR policy released its singleton predictor. Intervals are schematic.

With FFN A8 enabled in both arms, the archived shared-workspace comparison is 425.055 to 220.098 MiB. The separate non-A8 comparison is 421.234 to 219.500 MiB; each comparison holds its A8 setting fixed.

Wider 48/64-row alternatives were measured. A cooled historical comparison found only $+0.80\%$ at B3 and -0.32% at B4 with 204.957 MiB more shared workspace, motivating the context-first 32-row choice in that study. A cohort that shrinks from B4 to B2 can later widen; actual row width is recorded during common-active intervals.

4.4. Accepted Work, Row Packing, and Lookup Extension

A speculation cycle contains proposal construction, target verification, acceptance, and state reconciliation. Let a_r be committed tokens and τ_r the complete cycle time at verification width r. The useful rate is a_r/τ_r; a longer proposal is beneficial only if its accepted work offsets its draft, verify and replay cost. The row constraint is $\sum_i r_i \leq 32$; the evaluated homogeneous implementation specializes this to $br \leq 32$.

Table 3. Adaptive verification decision at a completed-step boundary. This describes the studied controller; it does not change target acceptance or sampling semantics.

Condition	Action
Row budget cannot fit width 16	Keep width 8
Two favorable prior observations	Select 16; singleton sticky count 3
Singleton sticky count remains	Select 16; decrement count
Otherwise	Select 8
Lookup head disagrees	Keep learned head; invalidate tail
Target rejects a suffix	Commit accepted boundary; reconcile state

The lookup path retains seven learned proposal tokens. A context-derived continuation supplies an extended proposal only if its first seven tokens agree with that learned head. On disagreement, the learned head remains and the lookup tail is invalidated. This prevents a repeated field name from replacing a correct learned value with a stale continuation. Target verification, rather than context lookup, decides which tokens become committed output.

The archived controller observes signals from the preceding completed step: whether the lookup tail can be filled and whether at least eight tokens were emitted. Two consecutive favorable observations allow wider verification when the row budget permits it. Singleton execution retains that preference for up to three further steps; multi-lane execution does not use that sticky interval. The controller uses these recent observations to adapt to available proposal work.

Acceptance and rollback align target KV, recurrent state, draft context, masks and logical offsets. Actual width is recorded during common-active intervals: B4 uses eight rows, while a subsequent B2 tail can widen to sixteen.

In the cooled M5 Pro study, the wider alternative has shared workspace 425.055 MiB rather than 220.098 MiB. At 4K input its TG improves by 3.815% for B3 and 2.387% for

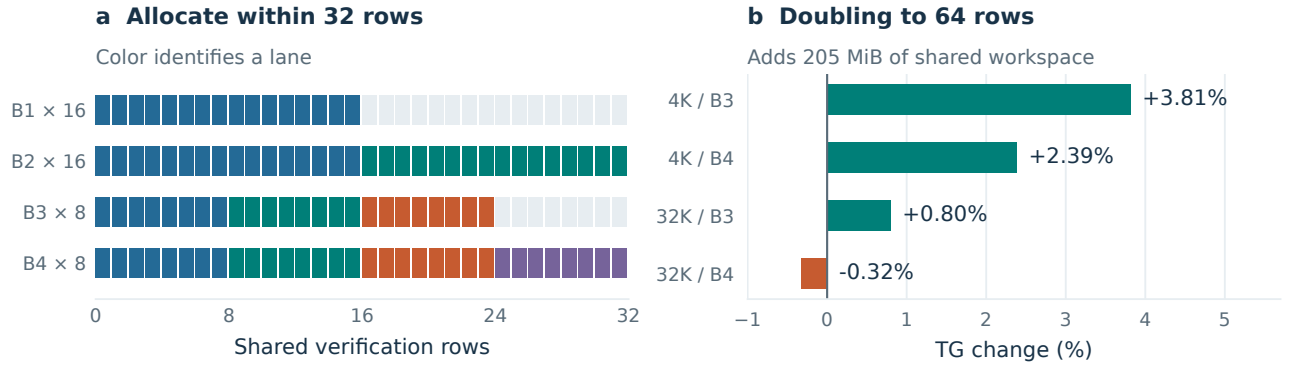


Figure 6. Left: conceptual packing of a shared 32-row verification workspace. B3 uses eight rows per lane and leaves eight unused; B4 uses all 32. Right: historical cooled M5 Pro comparisons of 64 versus 32 shared rows with prefill and target FFN A8 enabled. Wider rows add 204.957 MiB of shared storage; the 32K throughput differences are small.

B4; at 32K the changes are +0.798% and -0.321%. All eight paired outputs match. Process-peak differences are recorded separately from this shared-allocation delta.

5. StateTrans: Preserve Committed Progress

5.1. Admission, Scheduling, and Mode Changes

V1 distinguished queue arrival from admission. It reserved prompt pages plus a bounded output allowance, tracked shared-prefix references, and converted MTP to AR only at a completed round. Completion and cancellation first evaluated pending work, filtered departing rows, and then released page references. Generation-bound owner identities prevented an old finalizer from releasing a newer cohort's lease.

Native v2 applies the same continuity principle with a different resident set. Layered prefill retains a large arithmetic chunk while executing it in layer ranges. Intermediate state survives between ranges, and incumbent decode receives scheduling opportunities between slices. This differs from merely shrinking the token chunk. Evaluation reports incumbent decode, new-prompt prefill, and chat TTFT separately; the historical study improved the first two while increasing short-chat TTFT.

5.2. Lookup and Adaptive Useful Work

A long coding workflow repeats text that can cheaply supply proposal tokens. The studied design retains DFlash2's seven learned proposals and appends a context-lookup tail only when its head agrees; target verification still decides which output is committed. The context-lookup tail follows HyperQwen (syv-ai and HyperQwen contributors, 2026); the contribution here is its combination with learned proposals and bounded adaptive verification, rather than a new learned drafter.

Widening is useful only if accepted work grows faster than cycle cost:

$$a_{16}/a_8 > \tau_{16}/\tau_8. \tag{10}$$

The implemented heuristic uses completed-step emission and matching-tail opportunity, consecutive favorable observations, and limited singleton hysteresis, subject to the row budget. Width changes must preserve masks, recurrent replay, draft context, and rollback. The evaluation separates copy/edit workloads from sampling and constrained-decoding controls.

5.3. Composite Publication and Persistence

KV alone cannot resume this hybrid speculative model. A reusable point contains target KV, GDN state, and the draft context required at the committed boundary. Decode-end snapshots publish generated history after the tentative window is committed. Paged persistence shares chain segments; immutable state and page references survive until their transfers finish. Resource pressure can prevent publication even when the request itself completes.

V1's synchronous disk-only snapshot borrowed page-run views instead of materializing the whole packed cache. Its single-pass writer forwarded bounded payloads to an atomic destination. Native v2 extends this continuity to composite-state disk storage and sliced restoration. Root publication follows required segment data. A later state does not generally reconstruct the recurrent state of an earlier branch point, a distinction exposed by the restart study below.

5.4. Transitions, Publication, and Recovery Ordering

The reusable boundary is a tuple

$$C_t = (K_t, S_t^{\text{recurrent}}, D_t, \ell_t, I_t), \tag{11}$$

where K_t denotes target KV, D_t the draft context, ℓ_t logical offsets and I_t relevant input identities. Every element must

Table 4. Serving-transition ordering. These invariants apply across both execution modes; v2 retains its draft where v1 could switch to concurrent AR.

Step	Required action
1	Finish the current speculative round and its outstanding consumers.
2	Commit accepted progress; reconcile rejected tentative state.
3	Filter completed/cancelled requests; decrement page references.
4	Check page, output-reserve, phase and workspace admission.
5	Admit the new request; bind owners to the new cohort generation.
6	Interleave prefill slices with incumbent decode.
7	Publish only a consistent committed composite state.

describe the same committed prefix. Tokens proposed but not accepted do not advance that prefix. For multimodal inputs, matching text IDs alone is insufficient: overlapping segments must also carry the image identity.

Queue arrival alone does not consume an execution lane or change an incumbent's state. The MLX predecessor reserves an unshared request's prompt plus up to 8,192 output tokens in 256-token-page equivalents; shared prefixes require reference-aware accounting. That particular reserve policy is distinct from unlimited output completion. Native admission additionally accounts for its resident draft and selected verification workspace.

Checkpoint ownership is independent of request completion. The synchronous borrowed writer may retain page-run views because the producer cannot mutate them before serialization returns. An asynchronous writer must hold stable page/state references. Segment payload and state data become durable before a root is published; otherwise a visible root could reference incomplete data. On transfer failure, an optional save may be abandoned while generation and its source state remain valid.

Demand-allocated persistence staging. A later native revision removes a permanently reserved 196,083,712-byte (187 MiB) staging buffer. An asynchronous transfer instead retains an immutable owning source until IO completion. When snapshot-memory admission fails, a synchronous direct-to-disk path borrows the live lane state and waits for serialization before that state may change. Thus memory denial need not silently discard the recovery point, and cancellation cannot free a still-consumed source. Small and full-187 MiB native IO checks exercise this ownership and fallback; model-level cold/restart checks on each device reproduce 64 output tokens exactly with one composite-state hit and 8,704 disk KV tokens. Transfers retain their own memory and IO costs while eliminating the permanent reservation. The warm B2 capacity experiment in Section 8.1 uses this revision; the earlier cold-admission and persistence studies retain their original identities.

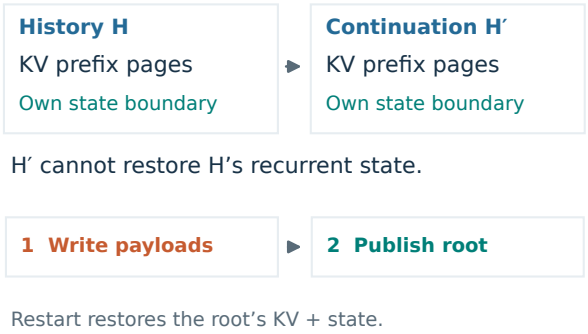


Figure 7. Composite continuation and ancestor retention. H and its continuation H′ share KV prefix pages but require their own recurrent-state boundary. A durable H′ does not reconstruct H's recurrent state. Publication exposes the root only after its required payloads. Arrows show publication dependencies.

Background saves can themselves prevent recovery by pinning pages needed for admission. A historical pressure test first restored an agent state, then immediately evicted it because pending saves pinned the pressure request's pages. The revised policy releases references as segments finish and allows pending work to yield under pressure. In the controlled follow-up, the next turn and a fork each restore 16,640 tokens and recompute 633, with disk-restore times 298 and 256 ms. Injected segment-write failures instead cause recomputation; all four requests complete, with no leftover root or temporary file. These tests concern ownership and failure ordering, distinct from the ancestor-retention defect in Section 8.7.

5.5. Layer Slices Preserve Matrix Efficiency

Shrinking a token chunk improves scheduling opportunities but can underutilize matrix hardware. Layered prefill retains a large chunk and executes consecutive layer ranges as separately schedulable parts. Hidden state, destination reservations and recurrent parity remain live between parts. The public chunk offset advances only when all required layers reach a consistent boundary. A final short prompt should not be attached to a long multi-part chunk if doing so would postpone its first logits until the long chunk completes.

The chunk target and slice target control different tradeoffs. The former changes arithmetic geometry and new-arrival wait; the latter changes opportunities to serve incumbent decode. Neither is a hard latency bound. The measured 32K setting uses a 2,000 ms chunk target and 500 ms slice target. At 128K, matching the incumbent's decode share is neces-

sary before attributing prefill improvements: a setting that gives decode more service can make the incoming prompt slower even while total service improves.

6. CrossFire: Heterogeneous Execution

6.1. CPU/GPU Partition and Arithmetic Selection

CrossFire is the v2 framework for using additional compute under the same physical budget. Intra-prefill CPU/GPU execution partitions projection columns. On M4 Pro, the selected CPU-SME path uses GPU-matched arithmetic and a bounded CPU fraction; the evaluated profile specifies 0.16 with eight threads. Earlier 0.20 or 0.33 studies are distinct from that evaluated recipe.

On M5 Pro, selective W4A8 prefill and target-verification FFN use hardware INT8 opportunities; the implementation follows the register-decoding approach credited to oMLX (PowerSpy and oMLX contributors, 2026). Prefill and verify switches are separate. Historical operator measurements found the studied M4 Pro INT8 path did not provide the M5 Pro advantage, so profile selection is device-specific. A8 means activation quantization and is distinct from ANE execution.

The inherited Splash language-FFN ANE path is a third execution form. Our integration concerns phase buffers, coexistence, retention/release, and numerical characterization. M6 enables this path in its selected profile. Its numerical behavior and history dependence are evaluated separately from the vision tower.

6.2. Complete ANE Vision Tower

CrossFire also moves supported vision encoding off a busy GPU. The full 27-layer vision tower is executed as nine groups with actual full-ANE/no-GPU placement checks. Authoring uses fixed capacity tiers and runtime position/mask inputs. M4 Pro and M5 Pro historical studies compiled 4K, 8K, and 14.4K variants; an oversized tier can waste work on padding even when the logical image is small.

The numerical work is part of the contribution. Equivalent LayerNorm range conditioning addresses measured variance overflow; explicit original GELU expressions and attention probability–value scaling address additional low-precision errors. These preserve model weights and real-arithmetic definitions. Device-local placement, memory metering, worker ownership, cancellation, and recovery supplement embedding comparisons (Apple, 2022).

The aligned M5 Pro historical recipe measured encoder cores of 0.903, 5.606, and 14.747 seconds on the three fixtures, with FP32-relative embedding L2 of 0.683%, 3.190%, and 2.987%.

6.3. Routing and Service Objectives

Text decoding for an image answer still uses the GPU; language prefill can additionally use ANE on M6. Routing must therefore consider the whole request and its peers, not just encoder speed. For image arrival a and completion times c_I, c_T, image latency is $c_I - a$, remaining text latency is $c_T - a$, and remaining makespan is $\max(c_I, c_T) - a$. The sum of waiting times is a different metric.

Historical routing experiments used measured tier costs and forced diagnostic routes to study overlap. The completed service instead qualifies ready 4K/8K tiers using execution phase, queue state, worker/payload leases, and current memory admission. These tier sizes count raw patches, not display pixels or language tokens. M4 Pro/M5 Pro permit vision ANE during GPU text prefill and decode. M6 additionally uses ANE for language prefill, so its vision policy permits decode-only overlap with no active prefill; image encoding stays on GPU during language prefill. Ownership and safe-point coordination serialize access to the shared accelerator. A slowdown by itself is not evidence of simultaneous conflicting ANE execution.

Idle or unknown phase, unsupported geometry or multi-image cases, queue contention, and denied memory can select GPU. Larger compiled tiers remain historical experiments rather than default serving policy. The final policy is not a calibrated latency predictor based on earlier placeholder or borrowed cost thresholds. Section 8.12 distinguishes functional qualification of this policy from the historical forced-route interference measurements.

6.4. Two Execution Forms and Their Synchronization

For $Y = XW$, the projection can partition output columns as $Y = [XW_{\mathrm{GPU}}, XW_{\mathrm{CPU}}]$. Both producers must complete before a consumer uses all of Y. The CPU fraction and thread count affect load balance; assigning more columns to CPU need not improve wall time. Synchronization must also avoid submitting a GPU wait that can time out before its CPU producer becomes runnable. Producer completion, host-job readiness and suffix submission are separate events that require traceable ordering.

The CPU numerical path must match the intended reconstruction arithmetic. Rounding dequantized weights to BF16 before an otherwise FP32 product can introduce a systematic difference. The studied CPU-SME work uses a GPU-matched reconstruction path and operator comparisons on the same inputs. W4A8 is a separate choice: selective INT8 activation arithmetic on supported GPUs may increase throughput while changing logits and acceptance. Prefill A8 and verify A8 therefore have independent controls.

The ANE path partitions work across requests rather than projection columns. An image must fit a supported capac-

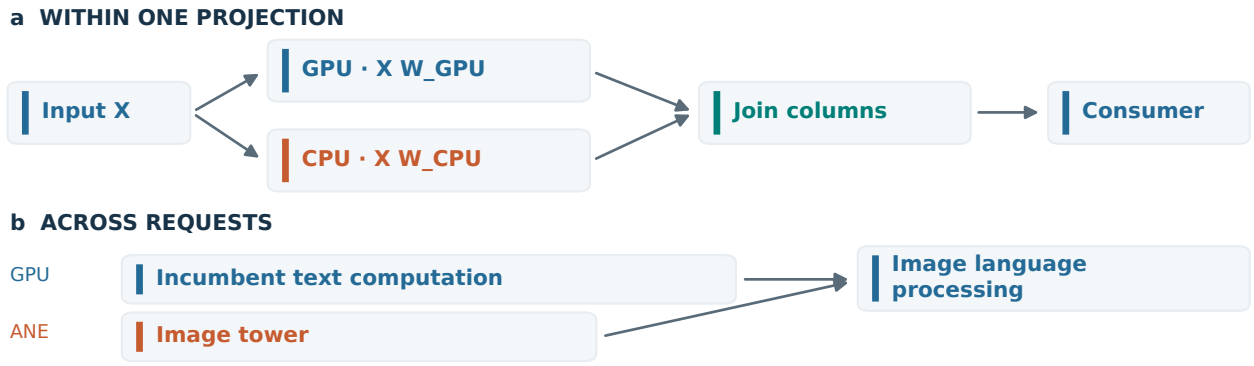


Figure 8. CrossFire's distinct forms. Top: CPU and GPU compute disjoint output-column partitions of a projection and join before its consumer. Bottom: ANE image encoding overlaps GPU text work; image-language processing still returns to GPU. Intervals are schematic.

ity tier, its compiled graph must have verified placement, and worker plus payload memory must be admitted. GPU text work can overlap the tower, but the resulting visual embedding and image-language prefill still consume GPU resources. Phase and contention determine the scheduling benefit.

6.5. Vision Graph Geometry and Numerical Conditioning

The full tower comprises 27 layers executed in nine three-layer groups. Fixed-capacity assets accept runtime positions and masks; four raw image patches merge into one language-side visual token. Capacity therefore must not be confused with language context length. Padding an image into a larger tier still executes dense work in the studied graph.

For 16 attention heads and query tiles of 1,024 rows, a three-layer group contains

$$N_{\text{tiles}} = 3 \times 16 \times \lceil C/1024 \rceil. \tag{12}$$

Capacities 4,096, 8,192 and 14,400 yield 192, 384 and 720 tiles. Dense attention pair work grows quadratically in capacity, while projections grow linearly. This explains why tier selection matters even when masked padding is semantically inert. Compiler time is measured separately from this operation count.

Low-precision errors require operator-level references. LayerNorm range conditioning addresses variance overflow while preserving the intended normalization; original GELU operand ordering avoids an unintended approximation change. Rescaling probability–value multiplication,

$$PV = ((sP)V)/s, \tag{13}$$

can move intermediates into a better representable range without changing real-arithmetic semantics. In a sampled historical correction with $s = 64$, relative error falls from 21.283% to 0.213%; splitting reduction alone leaves 21.280%. This local result is separate from the aligned full-tower recipe and from answer accuracy.

Table 5. Historical M5 Pro full-ANE tower observations. Each tier uses a different recorded image; these rows are not a same-image scaling experiment.

Tier capacity	Encoder core (s)	FP32-relative L2 (%)
4,096	0.903	0.683
8,192	5.606	3.190
14,400	14.747	2.987

Worker lifecycle is part of the interface: input ownership lasts through encoding, reservations are released on completion or cancellation, and unsupported shapes or denied admission take an explicit GPU route. A measured worker lease is not total accelerator memory; external service storage and shared pages can complicate physical attribution. The HTTP and concurrency studies therefore record actual encoding and reservation outcomes in addition to core timing.

7. Experimental Method

Platforms and configurations. The M4 Pro and M5 Pro systems each have 24 GiB of physical memory; M6 has 32 GiB. The earlier native batch matrices and 16K speed cohort use a 21,631 MiB native allocation setting, a separate 128 MiB host-headroom threshold, and a 22,015 MiB owned-process-group physical-memory guard. The later warm B2 capacity cohort uses a 22,015 MiB native budget and demand-allocated persistence staging; its host and guard protocol is recorded with that cohort. V1 uses a 21,000 MiB group guard. These limits account for different resources: native allocations, host headroom and sampled process-group footprint. Table 8 maps the experiments to

their protocols. Throughout the paper, device names identify complete evaluated configurations (Table 6). M4 Pro uses CPU-SME fraction 0.16 with eight threads and disables W4A8, lookup and language-model ANE. M5 Pro enables prefill/verify W4A8 and lookup. M6 additionally enables language-model ANE. Prefix reuse is enabled in the batch matrices; the stock-speed controls use fresh processes and disable candidate prefix reuse. Persistence comparisons change restore-point retention in separately identified builds.

Whole-stack baseline. Stock is upstream Splash 1.3.0 with INT8 KV, language ANE disabled and a 21,631 MiB native memory limit. The paired candidate uses TQ4 KV and its device-specific settings; model and prompt identities are frozen in the manifests. Stock retains a fixed 2,048 MiB host reserve and reports a 20,992 MiB hard budget; the candidate uses a 128 MiB host reserve. The comparison therefore covers representation, allocation policy, effective admission budgets and execution settings together. Paired PP trials have identical input-token counts and frozen prompts; generated trajectories and stop lengths vary. At M5 Pro 1K, stock/candidate arms reach the 1,024-token cap in 18/21 of 24 trials, respectively.

Batch throughput. We use two nominal input buckets (1K and 32K), eight fixed fixtures, three repeats and widths B1/B2/B4: 144 cohorts per device, or 24 per cell. Actual prompt lengths vary by fixture. Generation uses natural EOS and a 1,024-token cap. A barrier holds decode until all cohort prompts are ready. We report the arithmetic mean of per-cohort native aggregate decode rates, including the shrinking tail. Observed min–max whiskers show workload dispersion. All three matrices are complete. The M4 Pro matrix uses the pre-repair common source, identified separately from the later persistence build; its 144 launch records configure CPU prefill at fraction 0.16 with eight threads, with W4A8 and lookup disabled. CPU partitioning applies to eligible prefill operations. Persistent disk-cache quotas are 128 GiB on M4 Pro and 2 GiB on M5 Pro/M6. These configurable storage-retention limits and device-specific execution profiles accompany each result.

Image interference. The M5 Pro diagnostic starts one or three 32K text lanes, releases their decode barrier, and then submits an image. No-image, GPU-image, and forced-ANE-image arms use three rotated-order repeats. The image makes total concurrency B2 or B4. We measure the largest text delivery-event gap in the 30-second arrival window, image latency, and full text-cohort wall time. Delivery events may contain multiple tokens. The 8K and 14.4K visual tiers share the same no-image control. Forced routing and actual execution device are recorded separately.

Task quality. Task subsets retain their paired prompts, sampling settings, and output limits. M4 Pro uses 15 selected AIME questions and 30 questions each from MMLU-Pro and LiveCodeBench (LCB). M5 Pro uses a different 24-question LCB subset with a 32K output cap, and 30 AIME questions across three seeds. Seed repeats are not additional independent questions. Vision uses 30 MMMU-Pro questions containing 34 images; only questions actually encoded on ANE support ANE task-quality comparisons. These selected subsets serve as paired regression probes.

Restart protocol. A writer submits history H, repeats H, and submits a continuation H'. A fresh engine then requests H and H'; an additional control reverses that fresh-process order. We record cached/recomputed tokens, disk-KV reads, recurrent-state hits, time to first token (TTFT), and output equality. Disk reads are attributed per request: a second request can reuse pages already restored by the first.

8. Evaluation

8.1. System Generations: Capacity and Fixed-Output Speed

Figure 1 compares mlx-vlm, JustFit v1, upstream Splash 1.3.0 and JustFit v2. We freeze eight real code tasks in source order from the low-entropy block, excluding placeholder-only rows, in the official SPEED-Bench `throughput_16k/test` split (NVIDIA, 2026) before inspecting outputs. Each request generates exactly 512 tokens with greedy sampling and medium thinking, EOS and custom-stop termination disabled, and zero prompt-cache hits. Actual inputs span 17,053–18,253 tokens. PP divides input tokens by client TTFT; TG uses the 511 intervals after the first generated token. Medians summarize the eight tasks, not repeated trials of one prompt. Different numerical paths can produce different token trajectories; fixed lengths equalize output work, not semantic content or quality.

The mlx-vlm baseline is a preserved historical snapshot. Historical mlx-vlm/v1 use MXFP4 weights and MTP, whereas Splash/v2 use packed Q4 and DFlash. M4 Pro Splash uses a 1 GiB host reserve after original-default startup failures; M5 Pro/M6 retain the 2 GiB upstream reserve. Failed starts are retained outside completed trials. The original four-arm cohort contains 96 requests. A subsequent alternating old/new v2 runtime check contributes 16 requests per device; all 24 paired outputs match and median PP/TG changes remain within 1%. Figure 1 uses that refreshed v2 cohort. Speed runs use 21,631 MiB internally; a one-prompt-per-arm sanity check at 22,015 MiB yields equal outputs on all devices.

The retained-context panel reports successful tested bounds, not a searched maximum. B1 includes input and generated

Table 6. Earlier batch-matrix configuration dimensions (the batch-throughput study). The native allocation limit is 21,631 MiB, host reserve 128 MiB, and owned-group guard 22,015 MiB. Historical mechanism experiments use the explicitly described local settings.

Device	Physical GiB	CPU fraction / threads	Prefill / verify A8	Context lookup	Language ANE
M4 Pro	24	0.16 / 8	Off / off	Off	Off
M5 Pro	24	0	On / on	On	Off
M6	32	0	On / on	On	On

Table 7. Metric definitions used throughout the evaluation. A result is compared only within the same denominator and timing scope.

Metric	Definition and interpretation
Client-visible PP	Original input tokens divided by TTFT; recomputed rows do not inflate the numerator.
Native aggregate TG	Runtime-reported aggregate generated-token rate per cohort; includes the cohort tail under the stated protocol.
Common-active TG	Generated work during the interval when all intended decode lanes are active.
Client-visible TG	Generated tokens divided by the client generation interval; distinct from native kernel timing.
Maximum / p99 delivery gap	Gaps between observed stream delivery events in the specified window; events may contain multiple tokens.
Image latency	Image completion minus its own arrival; excludes preceding time spent by already-active requests.
Physical peak	Sampled footprint for the owned process group; can miss short peaks or double-count shared pages.
Metal allocation	Runtime allocation accounting; excludes some host and external accelerator-service storage.
Restore evidence	Cached/recomputed tokens, disk-KV reads and composite-state hits attributed to each cold or warm request.

positions; B2 is an aggregate over two live requests. JustFit v1 completes $(160\text{K}+16\text{K})+(128\text{K}+16\text{K})$ with ordered warm prefixes under a 21,000 MiB process-group guard. V2 completes $(128\text{K}+32\text{K})\times 2$ with a 22,015 MiB native budget and demand-allocated persistence staging; its draft remains resident. Both therefore reach a 320K page capacity, with distinct initialization and output shapes. V1's nearest final simultaneous logical sample is 327,677 positions (three below 327,680); its page high-water mark reaches 327,680. This is warm aggregate capacity across two live requests.

M4 Pro and M5 Pro have physical 24 GiB. M6 has physical 32 GiB; its final v1/v2 B2 tests reserve and lock 8 GiB, with compression and swap permitted in the remaining memory. M6 speed runs use the full physical configuration. At 16K, v2 improves median PP relative to Splash by 9.2%, 18.8% and 46.2% on M4 Pro, M5 Pro and M6, respectively. Median TG changes by -8.8%, -3.6% and -2.2%. The context-oriented configuration thus pairs faster prefill with a modest singleton-decode cost in this fixed-output workload.

8.2. Historical Native Development Progression

The development chain shows how execution changes recover the cost of compressed storage. Introducing TQ4 first reduces allocation and slows both rates; later representation-aware kernels and workspace changes recover throughput while keeping the smaller live set. Recorded outputs are identical across stages, with zero prefix-cache hits and no resource suspension or replay. Earlier snapshots retain checkpoint-publication failure counters. Scheduling, CPU/ANE execution and lookup are evaluated in their own studies below.

8.3. Batch Throughput and Matched Controls

Concurrency raises aggregate decode on M5 Pro and M6 while revealing a different scaling regime on M4 Pro (Figure 11). At 32K, M5 Pro grows from 56.84 to 83.84 to 111.28 tokens/s for B1/B2/B4; M6 grows from 34.58 to 56.60 to 81.30. Their B4 rates are about twice the singleton rate or more. M4 Pro changes from 40.92 to 43.32 to 43.12 tokens/s at 32K, and from 58.03 to 61.10 to 60.35 at 1K, yielding B4/B1 ratios of 1.054 and 1.040. Counter decomposition below relates this plateau to cycle-cost growth that nearly offsets additional useful output.

All three matrices contain 144 cohorts with accepted synchronization barriers and no recorded physical-memory guard trip. B1 native-prefill means at 1K/32K are 146.20/129.95 tokens/s on M4 Pro, 553.29/471.07 on M5 Pro and 444.44/403.75 on M6, computed from executed input tokens and native prefill wall time with zero matched prompt tokens. Prefill and fixed-width decode are both lower at 32K than at 1K on each device. These native rates exclude client and first-token overhead; the stock controls below use client-visible timing.

B1/B2 record no matched prompt tokens. B4 records 3,840/115,072 matched tokens at nominal 1K/32K on M5 Pro and 3,840/171,072 on both M4 Pro and M6. The 56,000-token difference follows a 28,000-token fixture pre-

Table 8. How to read the evidence. Each row answers a different question; later installation checks do not refresh historical performance results. K denotes 1,024 positions.

Evidence	Protocol	Supported interpretation
Figure 1: speed	Eight fixed 16K tasks per device/system; 512 forced tokens; later v2 cohort checked against the earlier cohort	End-to-end system rates; different numerical paths and decoders
Figure 1: capacity	Separate warm-prefix B1/B2 checks; v1/v2 B2 use different input/output shapes and budgets	Successful retained-position bounds; not a searched maximum or cold admission
Figure 9	Historical M5 Pro 32K+6K snapshots; one or two runs per stage	Development tradeoffs; not final-build single-factor attribution
Figure 11	Eight fixtures, three repeats; natural EOS, 1K output cap; decode barrier	Within-configuration batch scaling; not a fixed-length cross-device comparison
Figure 13	Forced GPU/ANE routes; image arrival during incumbent decode	Text continuity versus image latency under the recorded contention
Section 8.12	Phase-aware routing, cancellation and cold restart	Integrated functionality; not a throughput or quality distribution

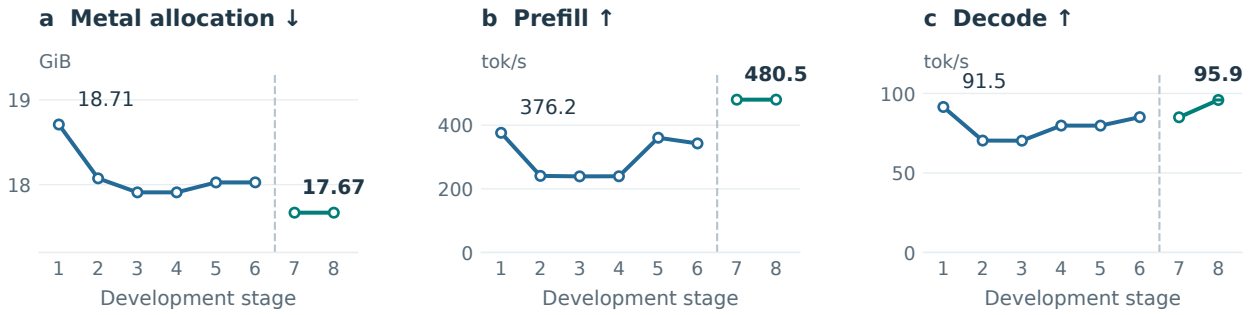


Figure 9. M5 Pro native-GPU development snapshots at fixed 32,768 input and 6,144 output tokens. Stages: (1) INT8 baseline; (2) TQ4; (3) scratch aliasing; (4) rotated verification; (5) staged prefill; (6) verify A8; (7) prefill A8 plus memory updates; (8) coalesced V staging. The dashed boundary separates build cohorts. Earlier stages have one run each; the last shows the median and range of two runs. Metal allocation excludes host and external accelerator memory; its axis starts above zero to resolve the 18.71-to-17.67 GiB change (-5.6%). Table 8 distinguishes this historical workload from the fixed-output code tasks.

fix: it matches in one of three anchor-zero repeats on M5 Pro and all three on M4 Pro/M6. The fixture set is the same. These observations document within-cohort prefix sharing alongside the disk quotas in the protocol. M4 Pro's maximum measured process-group footprint is 19.71 GiB across this matrix.

The stock controls show a consistent prefill benefit and workload-sensitive singleton decode. M5 Pro's mean decode rises by 8.20% at 1K and 3.42% at 32K (Table 9), whereas the fixed-output 16K cohort in Figure 1 falls by 3.6%. The sign change is compatible with the protocols: different prompts and context lengths, natural EOS versus fixed 512-token output, and means versus medians. Both measure client-visible speed; the native batch bars use a separate timing definition. These observations therefore describe different operating points of the same system.

Table 10 reports the paired dispersion over eight fixtures, each with three technical repeats. Both M5 Pro fixture-bootstrap intervals cross zero, so the positive mean TG differences remain tentative across workloads. M4 Pro's 32K prefill improves by 7.61%, while decode falls by 9.45%, exceeding the experiment's descriptive 7.5% slowdown criterion. The candidate GPU-only timing arm needed to isolate its CPU-partition contribution is absent. A separate M6 language-ANE toggle raises 32K visible PP from 360.72 to 399.74 and reduces TG from 36.34 to 34.66.

Useful work per cycle. Table 11 instantiates the useful-work/cycle-cost framework using counter differences within each cohort. For cohort i, let D_i be drafted tokens, A_i accepted draft tokens, O_i decode output tokens, N_i decode batches, and T_i accumulated decode-cycle milliseconds. We compute A_i/D_i, O_i/N_i, and T_i/N_i, then average each ratio across the 24 cohorts. Output tokens are measured directly, not reconstructed from accepted drafts. The peak column is the maximum observed owned-group footprint.

For M5 Pro 32K, mean output per batch grows from 4.95 to 16.13 (3.26×), while mean cycle time grows from 87.0 to 146.7 ms (1.69×). For M4 Pro 32K, output per batch grows from 4.53 to 15.78 (3.49×), but cycle time grows from 110.8

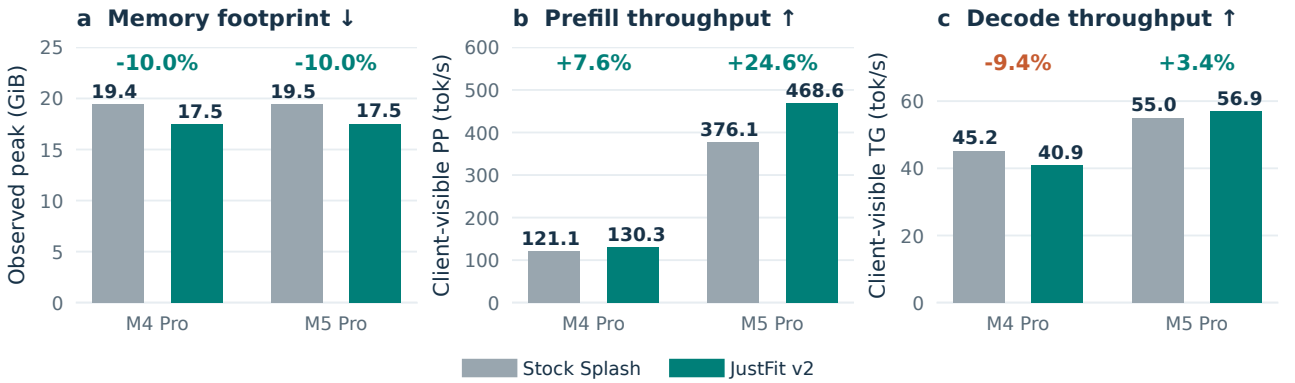


Figure 10. Memory and speed at 32K input on two 24 GiB systems, using the same 27B model package. Stock Splash and JustFit v2 are compared over 24 trials per arm and device (eight fixtures, three repeats). Memory is the maximum observed request-phase physical peak; PP and TG are arithmetic means of client-visible rates. Percentages compare those displayed summaries, not the mean of paired percentage changes. JustFit reduces the observed memory peak on both devices and improves PP; M4 Pro TG regresses while M5 Pro TG improves modestly. The whole-stack controls include KV format, execution settings and effective admission budgets.

Table 9. Matched client-visible prefill (PP) and decode (TG), tokens/s; 24 trials per arm and bucket. These whole-stack controls have a different timing definition from Figure 11.

Device	Input	Stock PP	JustFit PP	Change	Stock TG	JustFit TG	Change
M4 Pro	1K	132.18	142.72	+7.98%	56.43	57.22	+1.40%
M4 Pro	32K	121.05	130.27	+7.61%	45.20	40.93	-9.45%
M5 Pro	1K	422.04	524.55	+24.29%	65.54	70.92	+8.20%
M5 Pro	32K	376.13	468.62	+24.59%	55.02	56.90	+3.42%

to 367.9 ms (3.32×), while draft acceptance changes only from 0.504 to 0.497. M4 Pro cycle time is 1.27× M5 Pro at B1 and 2.51× at B4. Thus increasing cycle cost nearly offsets the increased useful output on M4 Pro; the aggregate counters narrow the observed plateau to width-dependent cycle cost, rather than a large collapse in acceptance. These counters cover complete cycles, including full occupancy and shrinking tails; Table 7 defines their aggregation.

Common-active TG counts callback-delivered tokens in the interval from the latest first-token event to the earliest final-token event. Its difference from native aggregate TG includes timing conventions and the cohort tail. B4/B1 ratios paired by fixture anchor and repeat are supplied in the data supplement, but B4 also adds other fixtures, so these are not equal-work matched requests.

At M5 Pro 1K, maximum request-phase footprint is 18.40 GiB for stock and 17.18 GiB for the candidate; at 32K it is 19.46 versus 17.52 GiB. These observations include different KV representations and allocation policies. They quantify the memory reduction of the complete configuration.

8.4. Capacity and Retained Progress

M4 Pro completes 128K singleton and total-300K B2 checks, a bounded two-hour/12-cycle run, and three-repeat B2/B3/B4 retention tests. These earlier experiments establish retained progress under their recorded configurations.

An M5 Pro two-lane workload retains 131,072 input plus 16,384 output tokens per lane, with common-active decode of 118.56 tokens/s and peak owned-group footprint of 20.97 GiB. One decode-state publication fails, separating successful generation from subsequent history reuse. A separate singleton retrieves the expected checkpoints from 261,632 prompt tokens and emits 36 tokens without recorded replay. The later warm-prefix checks in Section 8.1 extend B2 capacity to all three configurations with their own budgets and output shapes.

8.5. CrossFire: Continuity versus Image Latency

For the 8K tier, GPU image processing increases the median maximum text delivery gap from 0.107 to 1.702 seconds; ANE offload reduces it to 0.425 seconds. At 14.4K, GPU and ANE gaps are approximately 3.51 and 0.40 seconds. Image latency rises from 8.54 to 13.15 seconds at 8K and from approximately 15.7 to 23.1 seconds at 14.4K (Figure 13). CrossFire therefore trades image completion time for continuity of the incumbent text stream in these one-lane workloads. Recorded ANE preparation takes 6.21–6.33 s at 8K and 15.49–15.58 s at 14.4K, with approximately 6.9–7.7 s remaining until image completion. Median preparation is

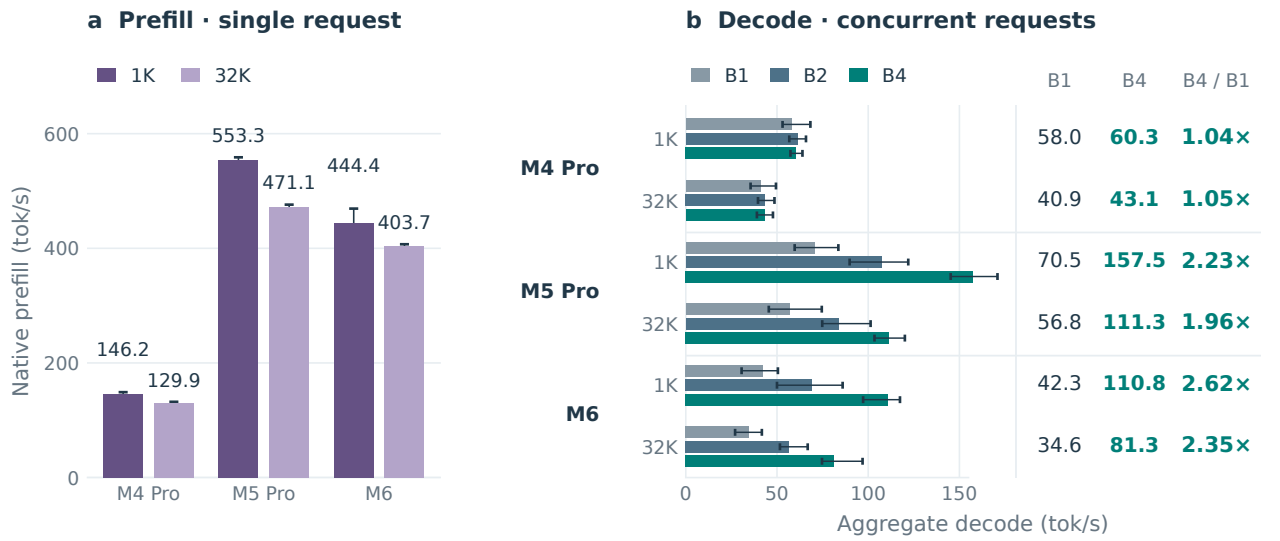


Figure 11. Context length and concurrent throughput in the same native experiment. Left: adjacent 1K/32K bars show B1 native prefill per device. Right: each context has B1/B2/B4 aggregate-decode bars; aligned columns give the B1 and B4 means (tokens/s) and their B4/B1 ratio. Exact B2 means are retained in the complete throughput table. Every bar is a mean of 24 cohorts; whiskers show observed min–max, not confidence intervals. Device profiles are listed in Table 6; native timing is defined in Table 7.

Table 10. Client TG dispersion and task+repeat pairing. Parentheses are sample SD; paired change is the mean of per-pair percentages, unlike the ratio of means in Table 9. Intervals are exploratory percentile bootstrap intervals resampling eight fixtures with all three repeats retained (20,000 draws). Positive counts are over eight fixture means. With only eight selected fixtures, percentile intervals can under-cover and do not establish population significance. For M4 Pro 32K, a fixture-level t sensitivity check gives $t = -2.27$ (7 df), $p \approx 0.058$; its 95% t interval includes zero.

Device	Input	Stock TG (SD)	Candidate TG (SD)	Paired % (SD)	Positive fixtures	Fixture 95% interval
M4 Pro	1K	56.43 (6.07)	57.22 (4.81)	+2.09 (10.26)	4/8	-3.77, +9.76
M4 Pro	32K	45.20 (5.70)	40.93 (4.04)	-8.61 (10.27)	1/8	-15.11, -1.30
M5 Pro	1K	65.54 (6.33)	70.92 (7.84)	+9.00 (13.65)	6/8	-1.31, +16.93
M5 Pro	32K	55.02 (7.35)	56.90 (9.70)	+3.53 (11.78)	5/8	-4.63, +11.37

about 73% and 99% of the corresponding GPU arm's entire image latency. These cross-arm ratios describe the harness intervals; Appendix C retains the full timing ranges.

Admission limits the result under higher concurrency. With three incumbent text lanes, some nominal ANE attempts fall back to GPU. One original GPU control overlaps background compilation and is unsuitable for a clean comparison. In a separate set of five uncontaminated B4 control records, both requested ANE routes encode zero images on ANE; the 8K record is retained inside a failed assertion payload, and the 14.4K attempt is denied its memory lease. These attempts document admission fallback rather than executed ANE throughput.

Twelve loopback HTTP requests exercise GPU/policy routing for 8K and 14.4K images. All content checks pass, reservations return to zero, and disconnect/recovery succeeds. For the long-prefill cases, text wall changes from 79.54 to 78.29 seconds at 8K and from 83.96 to 81.62 at 14.4K, while image latency increases. Each shape/arm has one observation, so this is a lifecycle and routing test rather than a sustained-service result.

A separate three-pair M4 Pro CPU/GPU/ANE coexistence study reduces mean total wall from 308.964 to 287.077 seconds (7.08%) with essentially unchanged decode rate. The measurement covers the combined CPU/GPU/ANE configuration.

8.6. Task Outcomes

M4 Pro compares GPU and CPU-assisted execution; M5 Pro LCB compares stock and JustFit, AIME compares A8 off/on, and MMMU-Pro compares GPU and actual ANE execution. Table 12 gives bounded regression evidence. M6's GPU AIME subset has 14 correct and one capped answer among 15; separate language-ANE on/off probes each pass four

Table 11. Counter-derived cycle measurements, 24 cohorts per cell. Native and common-active TG use different clock/window definitions. Rate, acceptance, output/batch and ms/batch columns are means of 24 cohort ratios; peak is the maximum footprint across those cohorts, not total machine or ANE service memory.

Device	Input	B	Native TG	Common-active TG	A/D	Tokens/batch	ms/batch	Peak GiB
M4 Pro	1K	1	58.03	57.85	0.610	5.27	91.1	17.67
M4 Pro	1K	2	61.10	60.97	0.604	9.36	153.2	17.67
M4 Pro	1K	4	60.35	59.02	0.579	16.82	279.8	18.52
M4 Pro	32K	1	40.92	40.83	0.504	4.53	110.8	17.24
M4 Pro	32K	2	43.32	43.08	0.501	8.59	199.0	18.28
M4 Pro	32K	4	43.12	42.75	0.497	15.78	367.9	19.71
M5 Pro	1K	1	70.52	70.41	0.554	5.32	75.5	17.16
M5 Pro	1K	2	107.31	117.46	0.538	9.01	84.1	17.16
M5 Pro	1K	4	157.53	166.77	0.573	17.18	109.6	18.40
M5 Pro	32K	1	56.84	56.70	0.506	4.95	87.0	17.16
M5 Pro	32K	2	83.84	87.16	0.501	8.81	105.2	18.13
M5 Pro	32K	4	111.28	116.21	0.522	16.13	146.7	19.95
M6	1K	1	42.29	42.17	0.535	5.11	121.1	17.80
M6	1K	2	69.17	76.03	0.516	8.91	128.8	17.82
M6	1K	4	110.79	119.94	0.558	17.04	154.4	18.80
M6	32K	1	34.58	34.50	0.497	4.71	136.3	17.80
M6	32K	2	56.60	58.89	0.503	8.78	155.5	18.78
M6	32K	4	81.30	86.02	0.524	16.38	201.9	20.87

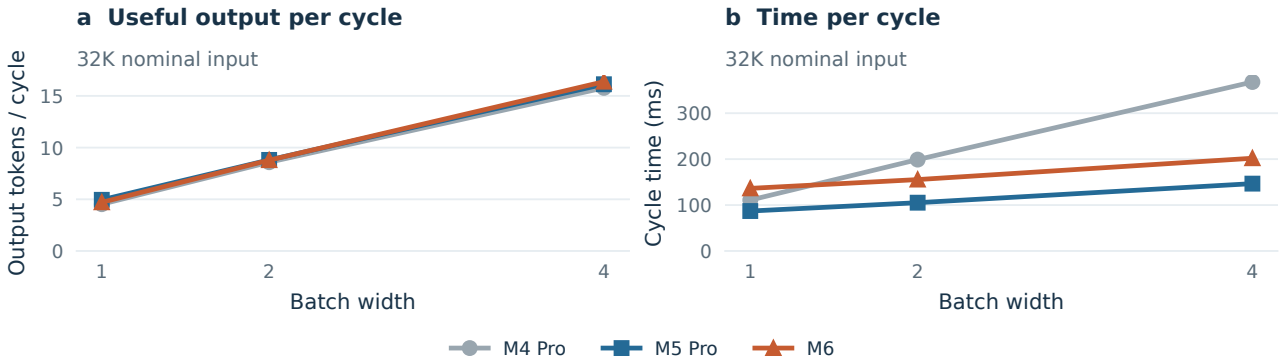


Figure 12. Measured useful work and cycle cost at nominal 32K input. Each point is the mean of 24 per-cohort counter ratios. Both useful output and cost increase with concurrency; these measurements include shrinking tails and are not isolated verification-kernel timings.

Table 12. Selected task probes: correct / capped or ungraded. Arm totals can agree despite different per-question outputs. M5 Pro AIME contains 30 questions at three seeds.

Device	Task (n)	Control	Treatment
M4 Pro	AIME (15)	14 / 1	14 / 1
M4 Pro	MMLU-Pro (30)	27 / 0	27 / 0
M4 Pro	LCB (30)	25 / 3	25 / 3
M5 Pro	LCB (24)	17 / 7	17 / 7
M5 Pro	AIME (90)	86 / 3	85 / 3
M5 Pro	MMMU-Pro (29)	26 / 0	24 / 0

coding and eight retrieval cases. The task sets differ across devices.

The M5 Pro three-tier vision profile encodes 27 questions through 4K and two through 8K with full-ANE/no-GPU placement. One remaining question exceeds 14.4K capacity and falls back to GPU. Thus this packet contains no 14.4K task-quality sample. The paired actual-ANE subset scores GPU 26/29 versus ANE 24/29 (Figure 14); the GPU control is reused after input/build/settings identity checks. Music_134 and Pharmacy_333 favor GPU at this seed. Additional M4 Pro outcomes follow below under their own protocol; Section 9 discusses statistical scope.

8.7. Restoring Composite History

The cache can omit an ancestor state when exactly one child subtree contains an ordinary state. Its KV chain can remain on disk even though the recurrent state needed to resume that ancestor is absent. We evaluate a retention rule that

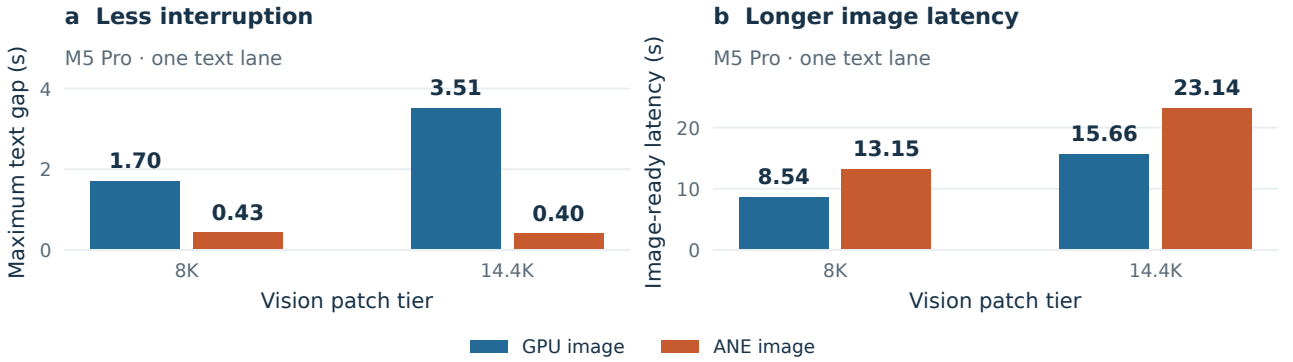


Figure 13. M5 Pro, one incumbent text lane plus one arriving image. Bars are medians of three repeats. All ANE-arm images in these cells execute on ANE. Offload reduces the maximum delivery gap but increases image latency. The no-image maximum gap is approximately 0.107 s.

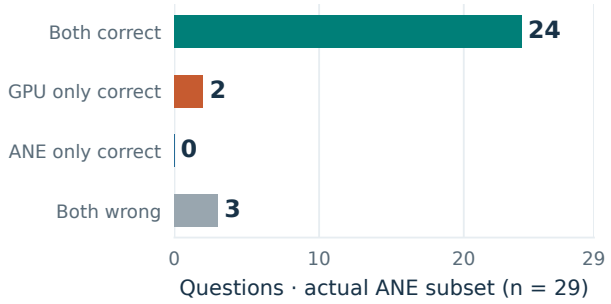


Figure 14. Paired outcomes on the 29 questions actually encoded by ANE. Two discordances favor GPU at the evaluated seed. The selected sample contains 29 paired questions.

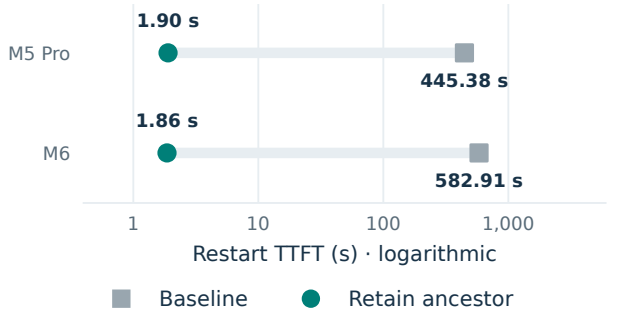


Figure 15. 128K original-history restart after a continuation. Compare the two retention policies within each device; fixture construction differs across devices. Logarithmic TTFT axis. Outputs match within each pair.

preserves such restore points when the stored chain is at least 32,768 tokens long.

For the M5 Pro 128K history, retaining the state restores 131,040 tokens, recomputes 32, and reduces TTFT from 445.38 to 1.90 seconds. The corresponding M6 pair changes from 582.91 to 1.86 seconds with the same restored token count (Figure 15). Output equality is checked separately from semantic answer correctness.

The threshold is defined on stored chain length, not nominal prompt length: on M5 Pro, a 32,768-token prompt stores 32,736 tokens and remains below threshold; a 32,800-token prompt stores 32,768 and benefits. Shorter ancestor histories are not universally preserved. In a fresh-process H′-first control, all ten combinations of five lengths and two retention policies show the expected disk-KV reuse, a recurrent-state hit, and matching outputs. This isolates continuation persistence from ancestor retention and from pages warmed by an earlier request.

A small-quota experiment bounds reported KV disk use at 3,221,143,552 bytes under a 3 GiB setting, but all six requested histories recompute in its oldest-first restart sequence. Outputs remain equal. The counter excludes some state/metadata storage. The next experiment follows a surviving chain through eviction and reconciles budget accounting with filesystem occupancy.

Integrated-build restart and request order. A separate M5 Pro experiment combines the ancestor-retention change with both image-submission repairs. It uses a 128 GiB persistent-cache budget and two independent writer/restart groups with H of 131,072 tokens and H′ of 131,104 tokens. The first group requests H first after restart; the second requests H′ first. Table 13 reports only the first request in each new process. Both restore the expected disk KV and recurrent state, recompute 32 tokens, and reproduce the writer's output token sequence. Each ordering has one deterministic functional observation.

The second request in each process obtains zero additional disk-KV hits because its pages are already resident. Its TTFT is 1.092 seconds (H′ after H) or 1.083 seconds (H after H′), with matching output. A disk-hit predicate applied

Table 13. M5 Pro integrated-build cold recovery. Only the first request after a fresh-process restart is counted. Both outputs exactly match their respective writer references.

First	Disk KV	State hits	Recompute	TTFT (s)
H	131,040	1	32	1.732
H′	131,072	1	32	1.731

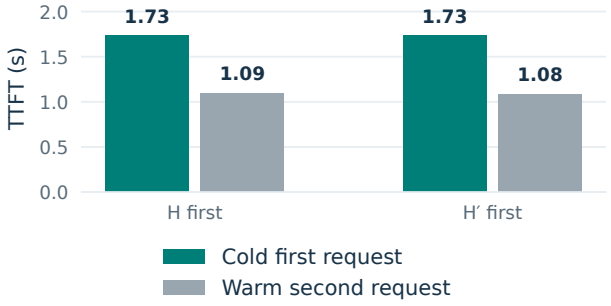


Figure 16. Request-order control on M5 Pro. One observation per bar. Cold restart and subsequent in-process reuse are different operations; the latter is not an additional cold-recovery sample.

indiscriminately to this second request would incorrectly report a restoration failure. Figure 16 therefore separates cold recovery from same-process reuse. These tests check output trajectories and restore counters on M5 Pro with language ANE disabled.

Real eviction and storage accounting. A separate 1 GiB control writes three independent 32,800-token histories. A restores 32,768 disk tokens before pressure. After B and C are written, A has been evicted and recomputes all 32,800 tokens, whereas C restores 32,768 disk tokens and recomputes 32. All three checked outputs match their cold references (Table 14). The first seeded history occupies 715 MiB of budget-accounted storage (528 MiB KV plus 187 MiB state), so two such resident disk copies cannot fit within 1 GiB. B is not requested after the final write, and no claim is made about its recoverability. Thus correct eviction can coexist with correct restoration of a surviving chain; eviction is not itself a restore defect.

Budget-accounted bytes must be distinguished from filesystem occupancy. After pressure, the 1 GiB control reports 1,073,594,368 accounted bytes, but 1,269,707,008 logical file bytes and 1,080,827,904 allocated bytes. The KV slot file is sparse: its logical length is 1,073,233,920 bytes while only 878,051,328 bytes are allocated; released slots can be hole-punched without shortening the file. Logical length must not be interpreted as physical SSD occupancy. Total allocated storage exceeds 1 GiB by about 6.76 MiB. The 7,233,536-byte difference between allocated and budget-accounted storage decomposes into 6,291,456 bytes of state-slot allocation slack, 540,672 bytes of KV-slot slack, and 401,408 bytes for records files. This reconciles the observed snapshot. The configured quota is therefore not an exact cap on total filesystem allocation in this observation. In each 128 GiB restart group, the corresponding counts are 3,003,252,736 accounted bytes, 3,004,049,152 logical bytes, and 3,021,840,384 allocated bytes. These measurements demonstrate survivor recovery and quantify the observed accounting overhead.

Table 14. M5 Pro small-budget eviction control. Each row is one observed request; state hits are one for the restored rows and zero for evicted A.

Request	Disk KV	Recompute	Exact
A before pressure	32,768	32	yes
A after pressure	0	32,800	yes
C after pressure	32,768	32	yes

8.8. Mechanism Measurements and Tradeoffs

The following earlier native and MLX experiments explain the execution and scheduling choices behind the design.

Rotated-domain verification reduces the M5 Pro cycle from 100.2/100.3 to 93.4/93.5 ms at 16K, and from 145.0/145.2 to 120.9/120.9 ms at 64K. The corresponding split-attention kernel changes from 18.0 to 10.9 ms and from 62.6 to 38.3 ms. End-to-end generation also changes its trajectory: at 16K, an explanation task improves from 28.7 to 30.2 tokens/s, while a copy task falls from 35.7 to 35.0. At 32K the two tasks change from 28.2 to 30.8 and 34.7 to 42.5; at 64K from 21.8 to 25.7 and 42.0 to 47.3. This variation motivates measuring both cycle cost and accepted work.

Coalesced V staging changes the address mapping rather than the codebook. In its historical M5 Pro 32K+6K workload, TG rises from 85.01 to 95.88 tokens/s, with a 95.91 repeat. The recorded outputs and acceptance are unchanged in that comparison. This layout experiment holds the codebook and recorded outputs fixed.

At 32K, layering improves incoming prefill by 13.1% and incumbent decode by 68.7%, with byte-identical outputs in all repeated runs. At 128K, the 2000/500 setting reallocates service toward decode and slightly slows the incoming prefill. The 4000/1500 setting approximately matches the control's incumbent TG and improves prefill. The policy selects a service tradeoff; a throughput-only summary would hide it.

8.9. Numerical Validation: Three Reference Levels

An independent codec reference checks unpacking, centroids, rotations and norms separately from target attention. A higher-precision attention reference on fixed inputs

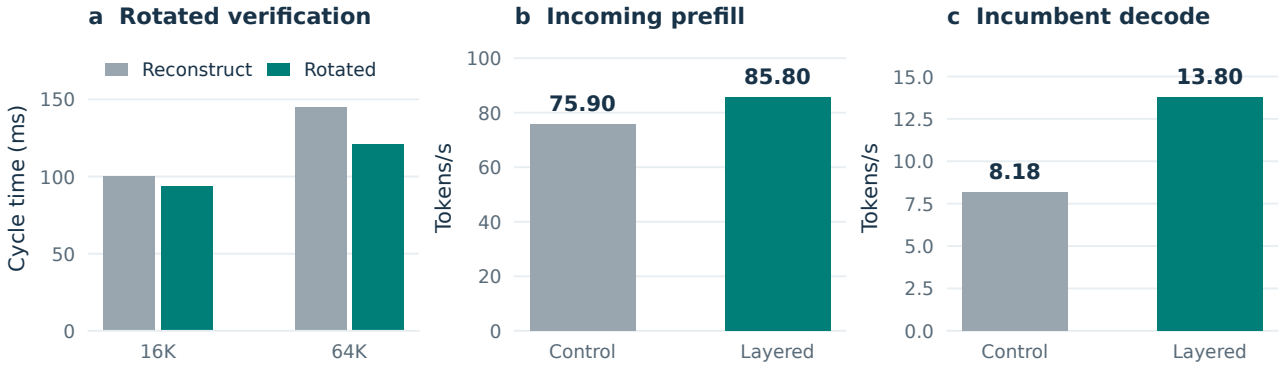


Figure 17. Historical mechanism measurements. Left: M5 Pro real-model verification cycles, two observations per arm at each length. Middle/right: M4 Pro 32K mixed-arrival medians, four control and three layered-prefill runs. Layering improves prefill and incumbent decode, but chat TTFT rises from 1.58 to 1.91 s. These are separate configurations and workloads.

Table 15. Layered-prefill tradeoffs on M4 Pro. The 32K rows are repeated medians; the 128K settings have one observation each. For the 128K layered settings the incumbent output budget is extended to span the full prefill. This prevents an artificially uncontended tail.

Incoming input	Policy	PP (tok/s)	Incumbent TG	Gap p99 (s)	Prefill wall (s)	Chat TTFT (s)
32K	Token-chunk control	75.9	8.18	1.984	431.8	1.58
32K	Layered 2000/500 ms	85.8	13.80	1.107	381.8	1.91
128K	Token-chunk control	42.9	5.2	3.99	3057.0	1.85–4.00
128K	Layered 2000/500 ms	41.5	14.1	1.11	3156.5	2.00–3.05
128K	Layered 4000/1500 ms	63.7	5.3	2.82	2056.5	3.10–4.37

then separates compression distortion from additional kernel arithmetic error. Finally, full-model comparisons track accepted tokens, output length and graded tasks. Agreement at one level does not imply agreement at another.

In the archived first-chunk TQ4 study, sampled vector reconstruction distortion is approximately 9.425–9.785%. A separate population-quadrature Lloyd–Max calculation with 16 scalar levels gives relative RMS of 9.747% for a standard Gaussian and 9.699% for a coordinate of a uniform 256-dimensional sphere, scaled to unit variance. Three initializations converge to the same values. Their proximity to the sampled reconstruction error provides a distribution-level consistency check under those scalar models. The quadrature script and converged centroids accompany the source. Across six captured sets at final query row 2,047, attention-output intrinsic relative RMS is 4.34–8.53%, while additional native arithmetic error is 0.165–0.175%. The study samples two prompts, three layers and all four KV heads. A fixed-history replay avoids confusing accumulated trajectory differences with a local operator defect.

Ambiguous task graders are another source of false conclusions. A fixture allowing two legal assignments and an underspecified day-boundary schedule were corrected individually; their original outputs remain evidence. Capped reasoning is recorded separately from completed wrong answers. This is why the public task tables retain both incorrect and capped counts rather than reporting only a single pass percentage.

Table 16. M4 Pro quality counts separated by diagnostic protocol. Actual ANE denotes at least one encoded image; aggregate requested-arm accuracy includes fallback. The coverage supplement changes routing coverage and is not pooled with the original diagnostic.

Arm	Requests	Correct	Actual ANE
gpu	30	25	0
ane-diagnostic	30	25	2
8K coverage	30	24	29

8.10. Additional M4 Pro vision and interference evidence

A separate M4 Pro packet completes 36 image-pipeline requests, 90 quality requests, and 18 image-arrival cohorts without execution failure. Task accuracy is reported separately. The quality packet comprises 30 GPU controls, 30 original diagnostic requests, and 30 additional 8K-coverage requests; the latter reuse the same questions and are not 30 new independent samples. Table 16 separates these protocols. The coverage supplement increases actual ANE execution from two questions to 29; the protocols remain separate in the table. Figure 18 and Table 17 show that, for one text lane, the maximum-gap median falls from 3.315 s (GPU image) to 0.474 s (ANE image), while image-ready

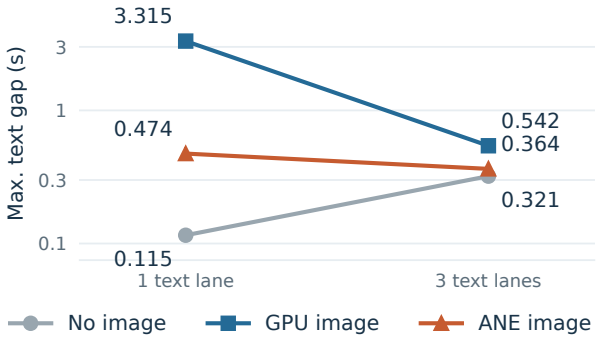


Figure 18. M4 Pro median maximum text-output gap in the image-arrival window (three observations per cell). All six requested-ANE cohorts actually encode on ANE. The vertical axis is logarithmic. Lower gaps describe continuity, not lower image latency.

Table 17. M4 Pro image-arrival medians. Text rate uses the fixed observation window; image time uses the packet image-ready measurement. These are descriptive cells, not confidence intervals.

Lanes	Arm	Gap (s)	Text tok/s	Image (s)
1	ane	0.474	22.80	36.91
1	gpu	3.315	15.50	32.97
1	none	0.115	30.50	–
3	ane	0.364	39.50	80.61
3	gpu	0.542	39.23	83.28
3	none	0.321	39.83	–

latency increases from 32.97 to 36.91 s. With three text lanes, corresponding gaps are 0.542 and 0.364 s and image-ready latencies are 83.28 and 80.61 s. The modest three-lane image-latency advantage is consistent with offloading tower work reducing GPU contention, but this is a possible explanation rather than an isolated mechanism measurement: three repeats per cell do not separate queueing, scheduling, and compute costs. The no-image controls retain smaller gaps. These M4 Pro observations complement the separately versioned integrated-build checks below.

8.11. Integrated-build regression scope

Four targeted M4 Pro checks on the integrated build pass: CPU and GPU image paths complete without GPU timeout or memory guard trips and reproduce reference outputs; 1K and 32K text controls each reproduce two recorded reference logit values bitwise and the 64 generated tokens exactly. The image fixture is answered incorrectly by both reference and candidate, so output preservation is not an accuracy pass. Section 8.12 extends these checks with three-device service cold recovery. The integrated M6 build also completes two large-image GPU-tower trials with language ANE enabled, a text control, and a forced 8K ANE-tower image. Its first cold-process 128K history request restores 131,040 disk KV tokens and one state snapshot, recomputes 32 tokens, and has a single observed TTFT of 1.85 s. The following branched request is in the same process: its 131,072 cached tokens include only 32 additional disk KV tokens, so it is not a second independent cold-restart observation. Both outputs match the writing process. These checks exercise the integrated execution and restoration paths.

8.12. Integrated Service: Phase Ownership and Committed History

The following experiments extend the preceding mechanism studies to the completed service path. They use the later integrated implementation, distributed as software version 2.3; JustFit v2 remains the system studied in this paper. Table 8 identifies these as integrated functional checks.

Phase-dependent heterogeneous execution. All three nodes execute both 4K and 8K vision tiers on ANE in their allowed phases. M4 Pro/M5 Pro admit vision ANE during GPU text prefill and decode; M6 admits it during decode without active prefill and retains GPU vision while language ANE prefill runs. Placement checks require full-ANE/no-GPU execution within the vision groups, in addition to native route counters. These checks connect the earlier forced-route diagnostics to the phase-aware service policy.

Cancellation must propagate from the request to worker ownership and payload reservations. Successful cancellation and subsequent recovery are observed on each device. The M6 test illustrates why admission and cancellation must be distinguished: its two-decode-lane cancellation attempt was denied ANE memory and ran on GPU. A subsequent one-lane interval exercised actual ANE cancellation and recovery. The two-lane routing cases and the one-lane cancellation case establish different properties; the failed proxy is retained.

Cold composite-state restoration. A common request is prepared, the service process is restarted, and its first recovery request is checked for disk reads, state reuse, and output equality. Each device restores 34,464 disk KV tokens, hits one composite-state snapshot, and reuses 34,464 tokens; outputs match the preparation run (Table 18). This supplies the previously absent M4 Pro service cold-restart evidence. The earlier 128K M5 Pro/M6 observations retain their own build and request-order identities. Section 9 identifies the remaining durability coverage.

History preservation through the frontend. State reuse also depends on the history delivered to the engine. A legacy image-request rewrite trimmed oldest messages using an approximate token count, changing the reusable prefix before native inference. The native-backend route now bypasses that legacy trimming and delegates context-limit validation to the engine. A synthetic image-bearing history sent

Table 18. Later integrated-service cold recovery, one qualified case per device. All outputs match the respective preparation request. Each case is the first recovery request after restart.

Device	Disk KV	State hits	Reused tokens
M4 Pro	34,464	1	34,464
M5 Pro	34,464	1	34,464
M6	34,464	1	34,464

through each M5 Pro/M6 broker contains 12,634 prompt tokens; a follow-up reuses 12,608 tokens (99.79%) and retrieves the early retained fact and image answer. The constructed history checks early-fact preservation together with prefix reuse.

Four frozen development retrieval tasks per device, at approximately 16K–34K actual input tokens, yield eight correct answers across M5 Pro/M6. M6 split counters confirm language ANE activity. One M5 Pro task resumes warm after a client interruption and is not cold-speed evidence. These development tasks extend functional retrieval coverage.

Bounded speculative output. Per-request thinking limits must survive the same frontend, protocol, and native boundaries as history. The restored control validates and propagates the budget into native request state; speculative retention respects the remaining budget, and the thinking segment closes at the cap rather than relying on a template suggestion. Bounded checks cover small caps, a 1,536-token alias, streaming, tools, and concurrent requests. This validates the serving control through the inherited speculative decoder. Installation-level reproduction is described in Appendix F.

9. Limitations

Comparison and attribution. Whole-system results combine weight formats, decoders, allocation policies, effective admission budgets and device-specific execution profiles. They establish the behavior of the evaluated configurations, not isolated chip rankings or individual-feature gains. Historical mechanism measurements explain design choices; a final-build factorial ablation of rotated attention, coalesced staging, lookup and adaptive width remains open. The M4 Pro stock comparison lacks a candidate GPU-only timing arm. Counter-derived cycle costs explain its batch plateau descriptively but do not separate verification, drafting, scheduling and hardware causes. No matched KIVI or KVQuant implementation comparison is included.

Workloads and statistics. The headline speed cohort contains eight fixed-output tasks; the natural-EOS matrices contain eight fixtures with three technical repeats and some B4 prefix sharing. Fixture variation, shrinking batch tails and mean-versus-median summaries affect comparisons. Min–max whiskers describe dispersion, and the exploratory eight-fixture bootstrap intervals can under-cover. M4 Pro's small B4-versus-B2 differences and M5 Pro's positive stock-control TG means are descriptive rather than established population improvements. Some early 16K trials overlapped dependency/asset provisioning. The 22,015 MiB speed sanity check covers one prompt per arm; the full speed distribution uses 21,631 MiB. Disk-quota effects on throughput are unmeasured. Two input buckets show an observed trend rather than a context-scaling law.

Capacity and memory scope. The 320K result is warm-prefix aggregate page capacity across two requests with a live draft. Cold admission, single-request 320K and matched long-context answer quality are separate questions. M6's locked 8 GiB reservation approximates available RAM on its physical 32 GiB system; OS, driver and compression behavior can differ from a physical 24 GiB device. M6 speed runs use its full memory configuration. Analytical operand bounds, Metal allocation, process-group footprint and external accelerator memory have different accounting scopes. Historical resource savings cannot be added into a single process-peak reduction.

Quality and heterogeneous service. Selected task subsets provide regression evidence, with quality equivalence unresolved. Embedding error, logit agreement, output equality and semantic correctness measure different properties. Forced ANE diagnostics establish behavior when execution is admitted; final-policy benefit also depends on phase, memory pressure and actual placement. Language-FFN and vision-ANE accuracy require separate controls. Harness preparation and residual image time do not isolate compute from queueing. The GPU-favoring vision outcomes, M4 Pro decode regression and denied ANE attempts remain explicit results.

Persistence and reproducibility. Cold checks restore 34,464 tokens on all three final service configurations, alongside earlier 128K M5 Pro/M6 studies. Repeated final-profile 128K restoration on every device, short-history retention policies, all eviction/branch orders, full-budget saturation and power-loss durability remain open. Request-order tests compare outputs and counters rather than every restored tensor. Filesystem overhead is reconciled for the observed snapshots, without a worst-case or indefinite-operation guarantee. Installation checks qualify artifact functionality at their stated workload sizes; they preserve the identities of the earlier performance evidence.

10. Related Work

Local execution and committed state. JustFit v1 supplies KVExec, PhaseSwap, and StateTrans (Chen, 2026) on MLX, an array framework for Apple silicon (Hannun et al., 2023). V2 retains these principles while changing the runtime and concurrent speculative resident set. Splash supplies native execution and DFlash2 the learned draft (Inco AI, 2026b;a). The baseline remains Splash 1.3.0; future JustFit releases will follow upstream updates such as 1.4 (Inco AI, 2026c). Gated Delta Networks (Yang et al., 2024) require the recurrent snapshot to accompany attention KV at a restore point.

Proposal generation. Target verification separates proposal generation from accepted output (Leviathan et al., 2023). Prompt Lookup Decoding reuses matching prompt substrings (Saxena, 2023); LLMA uses reference overlap (Yang et al., 2023); REST retrieves proposal tokens (He et al., 2024). Our context-tail implementation follows the HyperQwen lookup-augmented DFlash2 patch (syv-ai and HyperQwen contributors, 2026). JustFit integrates the learned prefix and lookup tail with adaptive width and a shared row budget.

KV compression and execution. KIVI studies asymmetric low-bit key/value quantization (Liu et al., 2024); KVQuant combines distribution-aware quantization with outlier handling (Hooper et al., 2024). TurboQuant studies compact rotated vector representations (Zandieh et al., 2026); KVExec additionally controls when and how stored operands are reconstructed for each consumer. PagedAttention manages serving pages (Kwon et al., 2023), while FlashAttention reduces attention IO (Dao et al., 2022). These address representation, allocation and IO; our contribution concerns their interaction with bounded consumers and a live draft on a shared physical budget.

Prefix reuse and service scheduling. RadixAttention in SGLang reuses shared prefix KV across calls (Zheng et al., 2024). CachedAttention retains conversation KV in a storage hierarchy, overlapping layer-wise preloading and asynchronous saves with computation (Gao et al., 2024). StateTrans addresses the composite boundary of a hybrid speculative model: target KV, recurrent state and draft context must resume the same committed prefix, including an ancestor retained after continuation. Sarathi-Serve studies prefill/decode scheduling (Agrawal et al., 2024); layered prefill here additionally yields within a large token chunk. FlexGen exploits hierarchical storage (Sheng et al., 2023); on unified-memory devices, CPU mappings consume the same physical capacity. CrossFire combines accelerator execution with phase ownership and memory admission, building on ANE deployment techniques (Apple, 2022) and the credited oMLX arithmetic implementation (PowerSpy and oMLX contributors, 2026).

11. Conclusion

JustFit v2 turns state lifetime into a serving resource. KVExec bounds the expanded history seen by each consumer; PhaseSwap keeps transient storage within the live-set budget; StateTrans carries committed progress across execution and restart; CrossFire uses heterogeneous compute under the same ownership rules. Together they preserve 320K aggregate retained capacity with a live draft, deliver 2.0–2.5× v1's median decode rate on the fixed-output workload, and improve prefill over Splash 1.3.0 at a modest singleton-decode cost. Concurrent scaling, reduced image-induced text gaps and composite-state recovery bring long histories and useful overlap to 24 GiB-class local serving.

References

Agrawal, A., Kedia, N., Panwar, A., Mohan, J., Kwatra, N., Gulavani, B. S., Tumanov, A., and Ramjee, R. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, 2024. URL https://www.usenix.org/conference/osdi24/presentation/agrawal.

Apple. Deploying transformers on the apple neural engine, 2022. URL https://machinelearning.apple.com/research/neural-engine-transformers.

Chen, Y. JustFit: Just-in-time state management for local LLM serving, 2026. URL https://arxiv.org/abs/2609.17475v2.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. FlashAttention: Fast and memory-efficient exact attention with IO-awareness, 2022. URL https://arxiv.org/abs/2205.14135.

Gao, B., He, Z., Sharma, P., Kang, Q., Jevdjic, D., Deng, J., Yang, X., Yu, Z., and Zuo, P. Cost-efficient large language model serving for multi-turn conversations with CachedAttention. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*, pp. 111–126, 2024. URL https://www.usenix.org/conference/atc24/presentation/gao-bin-cost.

Hannun, A., Digani, J., Katharopoulos, A., and Collobert, R. MLX: Efficient and flexible machine learning on apple silicon, 2023. URL https://github.com/ml-explore/mlx.

He, Z., Zhong, Z., Cai, T., Lee, J., and He, D. REST: Retrieval-based speculative decoding. In *Proceedings of*

NAACL-HLT, pp. 1582–1595, 2024. doi: 10.18653/v1/2024.naacl-long.88. URL https://aclanthology.org/2024.naacl-long.88/.

Hooper, C., Kim, S., Mohammadzadeh, H., Mahoney, M. W., Shao, Y. S., Keutzer, K., and Gholami, A. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/028fcbcf85435d39a40c4d61b42c99a4-Abstract-Conference.html.

Inco AI. DFlash 2: Keep drafting parallel, 2026a. URL https://inco.ai/blog/dflash2/.

Inco AI. Splash 1.3.0: A local inference engine for apple silicon, 2026b. URL https://github.com/incoai/splash/tree/1.3.0. Evaluated upstream baseline, version 1.3.0.

Inco AI. Splash 1.4.0 release page, 2026c. URL https://github.com/incoai/splash/releases/tag/1.4.0. Released October 20, 2026; subsequent upstream release.

Inco AI. Splash: A local engine built around the model, 2026d. URL https://inco.ai/blog/splash/. September 17, 2026 launch report; performance evaluated on a 48 GB M5 Pro.

Inco AI. Qwen3.8-27B-Splash: Model package and hardware requirements, 2026e. URL https://huggingface.co/incoai/Qwen3.8-27B-Splash. Accessed September 30, 2026; documents 36 GB unified memory, 48 GB or more recommended.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://arxiv.org/abs/2309.06180.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

Liu, Z., Yuan, J., Jin, H., Zhong, S., Xu, Z., Braverman, V., Chen, B., and Hu, X. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32332–32344, 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

NVIDIA. SPEED-Bench: throughput benchmark dataset. https://huggingface.co/datasets/nvidia/SPEED-Bench, 2026. Pinned throughput_16k/test revision and task identities accompany the experiment record.

PowerSpy and oMLX contributors. INT8-activation prefill kernels for Qwen3.5/3.6/3.8, 2026. URL https://github.com/jundot/omlx/pull/3548. Pull request 3548.

Saxena, A. Prompt lookup decoding, 2023. URL https://github.com/apoorvumang/prompt-lookup-decoding/.

Sheng, Y., Zheng, L., Yuan, B., Li, Z., Ryabinin, M., Fu, D. Y., Xie, Z., Chen, B., Barrett, C., Gonzalez, J. E., Liang, P., Ré, C., Stoica, I., and Zhang, C. FlexGen: High-throughput generative inference of large language models with a single GPU, 2023. URL https://arxiv.org/abs/2303.06865.

syv-ai and HyperQwen contributors. HyperQwen: Lookup-augmented DFlash2 drafting, 2026. URL https://github.com/syv-ai/HyperQwen/blob/main/patches/dflash2-lookup-drafting.patch. Source patch; accessed October 18, 2026.

Yang, N., Ge, T., Wang, L., Jiao, B., Jiang, D., Yang, L., Majumder, R., and Wei, F. Inference with reference: Lossless acceleration of large language models, 2023. URL https://arxiv.org/abs/2304.04487.

Yang, S., Kautz, J., and Hatamizadeh, A. Gated delta networks: Improving mamba2 with delta rule, 2024. URL https://arxiv.org/abs/2412.06464.

Zandieh, A., Daliri, M., Hadian, M., and Mirrokni, V. TurboQuant: Online vector quantization with near-optimal distortion rate. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5c802ef38ab6e366c2ea06eee554c088-Abstract-Conference.html.

Zheng, L., Yin, L., Xie, Z., Sun, C., Huang, J., Yu, C. H., Cao, S., Kozyrakis, C., Stoica, I., Gonzalez, J. E., Barrett, C., and Sheng, Y. SGLang: Efficient execution of structured language model programs, 2024. URL https://arxiv.org/abs/2312.07104.

A. Mechanisms and Design Alternatives

Table 19 summarizes the transition and supporting mechanisms. The implementation inventory and raw measurements accompany the source package. Mechanism inclusion does not imply that every feature has an isolated end-to-end ablation.

Table 19. Mechanism coverage and measured design alternatives.

Mechanism	MLX predecessor	Native extension
KV representation	Rotated MSE; 256-token pages; shared page references	Same compact representation; 32-token pages; native page/attention interfaces
Attention consumers	Fused inverse reconstruction; two-head grouping; explicit evaluation	One-head BF16 staging; rotated-domain verification; coalesced V layout; compact softmax
Components	Head and media leases; singleton MTP released for concurrent AR	Target/draft retained; host embedding gather; phase-scoped vision; generation-bound ownership
Workspaces	Bound lazy intermediates at evaluation boundaries	Alias disjoint intermediates; elastic prefill arena; decode-owned RoPE; shared verification rows
Transitions	Admission and output reservation; MTP/AR switch at complete rounds	Layered prefill; accepted-boundary reconciliation; adaptive width and shared-lane execution
Persistence	Borrowed page-run views; bounded atomic writer	Composite target/recurrent/draft snapshots; segment/root ordering; pressure-aware transfer and ancestor retention
Speculation	Singleton MTP; concurrent AR	Learned proposals plus matching lookup tail; adaptive 8/16-row verification; selective W4A8
CrossFire	Phase-coordinated GPU text and vision	CPU-SME/GPU column partition; complete ANE tower; image/text overlap and fallback; language-ANE integration

Unfilled comparisons. Warm-prefix B2 long-output capacity is supplied on all three configurations in Section 8.1, with the M6 memory-reservation limitation stated explicitly. Matched long-context task accuracy, three-device vision quality, final-configuration mechanism ablations, and full global-budget saturation/recovery remain open. M4 Pro batch throughput and bounded M5 Pro survivor recovery do not close those distinct comparisons. These omissions restrict the corresponding cross-device, attribution, and durability claims.

Reproducibility. The accompanying tables preserve counts, units, timing definitions, actual routes, and per-request restore counters. The separate provenance record maps measurements to exact input, build, profile, and source-result identities. Plot scripts consume these tables directly. Failed and partial observations remain in that record; they are not converted into successful comparison cells.

B. Complete Throughput and Interference Tables

Table 20. Batch matrix means with observed min–max ranges. A dash denotes an unavailable complete cell. Each completed cell has 24 cohorts.

Device	Input	B	n	Mean TG	Min TG	Max TG
M4 Pro	1K	1	24	58.03	53.06	68.31
M4 Pro	1K	2	24	61.10	56.73	65.89
M4 Pro	1K	4	24	60.35	57.38	63.96
M4 Pro	32K	1	24	40.92	35.53	49.45
M4 Pro	32K	2	24	43.32	39.59	48.62
M4 Pro	32K	4	24	43.12	38.99	47.79
M5 Pro	1K	1	24	70.52	59.69	83.65
M5 Pro	1K	2	24	107.31	89.82	121.96
M5 Pro	1K	4	24	157.53	145.22	170.87
M5 Pro	32K	1	24	56.84	45.51	74.55
M5 Pro	32K	2	24	83.84	74.81	101.31
M5 Pro	32K	4	24	111.28	103.44	120.10
M6	1K	1	24	42.29	30.60	50.50
M6	1K	2	24	69.17	50.00	85.98
M6	1K	4	24	110.79	97.21	117.42
M6	32K	1	24	34.58	27.04	41.70
M6	32K	2	24	56.60	51.62	66.83
M6	32K	4	24	81.30	74.68	96.91

Table 21. Original M5 Pro image-arrival diagnostic medians. L denotes incumbent text lanes; an image adds one request. Requested ANE includes fallback. Three observations per row; the three-lane 8K GPU cell includes the known background-compilation overlap.

Tier	L	Requested route	Actual ANE/n	Max gap (s)	Text wall (s)	Image latency (s)
8192	1	GPU	0/3	1.702	96.029	8.536
8192	1	ANE	3/3	0.425	95.288	13.155
8192	3	GPU	0/3	0.358	265.195	34.691
8192	3	ANE	2/3	0.211	264.495	33.665
14400	1	GPU	0/3	3.508	101.239	15.664
14400	1	ANE	3/3	0.398	100.818	23.137
14400	3	GPU	0/3	0.379	271.282	39.539
14400	3	ANE	3/3	0.208	269.646	37.465

Table 22. Separate uncontaminated three-text-lane controls. Both requested ANE routes have zero ANE encoding. The 8K ANE record comes from an assertion payload and is not a successful offload trial.

Tier	Requested route	Encoded	Gap (s)	Text wall (s)	Record status
8192	NONE	0	0.156	253.103	Recorded observation
8192	GPU	0	0.354	260.639	Recorded observation
8192	ANE	0	0.344	262.010	Assertion payload
14400	GPU	0	0.399	268.342	Recorded observation
14400	ANE	0	0.384	268.738	Recorded observation

C. Image-Arrival Timing Ranges

Requested and executed routes are separated. The 8K three-lane GPU control includes the documented background-compilation overlap and is not a clean control. A denied ANE request executes on GPU; its short attempted preparation interval is not ANE encoding time. Preparation remains a harness interval, and the residual to completion is not an isolated GPU stage.

Table 23. Observed minimum–maximum ranges in seconds; each row contains only one executed route. These are not confidence intervals.

Tier	L	Requested	Executed	n	Text max gap	Image completion	ANE preparation
8192	1	GPU	GPU	3	1.657–1.711	8.529–8.578	–
8192	1	ANE	ANE	3	0.425–0.426	13.147–13.267	6.210–6.332
8192	3	GPU	GPU	3	0.328–0.391	34.295–34.740	–
8192	3	ANE	GPU	1	0.334–0.334	34.587–34.587	–
8192	3	ANE	ANE	2	0.175–0.211	33.407–33.665	6.285–6.310
14400	1	GPU	GPU	3	3.503–3.531	15.660–15.714	–
14400	1	ANE	ANE	3	0.398–0.398	22.678–23.151	15.492–15.581
14400	3	GPU	GPU	3	0.372–0.427	39.131–39.902	–
14400	3	ANE	ANE	3	0.192–0.546	37.121–38.038	15.493–15.581

D. Restart Boundary Measurements

Table 24. Original-history H restart observations with a 128 GiB disk quota, separate from the M5 Pro/M6 2 GiB speed matrices. Prompt length, rather than a nominal case label, determines the boundary interpretation. Exact means byte-identical output within the paired comparison, not a semantic correctness score.

Device	Retention	Prompt	Cached	Recomputed	Disk KV	TTFT (s)	Exact
M5 Pro	Baseline	16384	0	16384	0	31.14	Yes
M5 Pro	Deep ancestor	16384	0	16384	0	31.08	Yes
M5 Pro	Baseline	32736	0	32736	0	68.79	Yes
M5 Pro	Deep ancestor	32736	0	32736	0	68.68	Yes
M5 Pro	Baseline	32768	0	32768	0	68.57	Yes
M5 Pro	Deep ancestor	32768	0	32768	0	68.59	Yes
M5 Pro	Baseline	32800	0	32800	0	68.66	Yes
M5 Pro	Deep ancestor	32800	32768	32	32768	0.62	Yes
M5 Pro	Baseline	131072	0	131072	0	445.38	Yes
M5 Pro	Deep ancestor	131072	131040	32	131040	1.90	Yes
M6	Deep ancestor	16025	16000	25	16000	0.54	Yes
M6	Baseline	16025	16000	25	16000	0.55	Yes
M6	Deep ancestor	32612	0	32612	0	81.06	Yes
M6	Baseline	32612	0	32612	0	80.61	Yes
M6	Deep ancestor	33011	32992	19	32992	0.75	Yes
M6	Baseline	33011	0	33011	0	82.29	Yes
M6	Deep ancestor	131072	131040	32	131040	1.86	Yes
M6	Baseline	131072	0	131072	0	582.91	Yes

The separate 2 GiB sequence is excluded from this cold-restart table: its second request can reuse pages warmed by the first. A single history's payload does not determine aggregate occupancy or prove eviction. The raw rows remain in the data supplement.

The fresh-process continuation-first control tests ten combinations of five lengths and two retention policies. The original controller's aggregate verdict is not substituted for the counters above: it also tests incremental disk reads on a second request whose shared chain can already be in memory. A valid restoration assessment records request order and residency before interpreting those reads.

E. Design Alternatives and Experimental Boundaries

Table 25. Measured alternatives and what their outcomes resolve.

Alternative	Observation	Design implication
Wider verification	204.957 MiB more shared workspace; 32K B3 +0.798%, B4 -0.321% TG.	Keep the smaller row budget for this context-first workload.
Elastic prefill arena alone	Initial cold B2 capacity does not improve; the limiting phase still needs the arena.	Planner headroom is not measured usable capacity.
Prefill-only A8	Historical PP 370.21 to 487.25 tokens/s, wall 96.62 to 76.49 s; natural TG -12.4%.	Measure generated work and verify separately; not a combined-A8 ablation.
Residual quantization	Sampled output RMS 0.05695 (4-bit), 0.13608 (3+1-bit), 0.06726 (4+1-bit).	These tested layouts do not improve the sampled reference error.
Oversized ANE graph	Padding still executes dense capacity-sized attention.	Select the smallest supported tier that fits original geometry.
Oldest-first restart under pressure	Six exact outputs but no restore in the recorded small-quota sequence.	Output correctness does not establish cache-survivor coverage.
128K partial-overlap scheduling	Incumbent output finishes before prefill, leaving an uncontended tail.	Exclude that row from matched full-overlap comparisons; extend incumbent output.

Residual-layout scope. The residual study uses six captured attention sets, eight construction seeds and final query row 2,047. The three layouts use 130, 132 and 164 bytes per 256-dimensional vector. This is an offline operator screen, not a public task score or a rejection of all residual quantizers. TurboQuant's QJL residual targets unbiased inner-product estimation (Zandieh et al., 2026); unbiasedness does not guarantee smaller squared error, particularly after nonlinear attention. The observed RMS ordering therefore does not itself diagnose a bug or establish the cause in this implementation.

Missing final comparisons. Final-build per-mechanism ablations, full three-device ANE task comparisons and full-budget filesystem-inclusive eviction/recovery tests remain missing. The M4 Pro batch matrix and bounded M5 Pro eviction-survivor checks are now measured under their explicitly stated configurations. Historical completed experiments above are retained under their own configurations; none fills a missing final cell automatically.

F. Artifact Identity and Bounded Reproduction

Three evidence layers. The performance tables identify their own runtime, profile and workload cohorts. Historical integrated-service checks use the 2.3.0rc14 software reference; later staging and capacity checks retain distinct native identities. The public software preview is 2.3.0rc24, available at `https://github.com/YuhuaBillChen/justfit-llm` and `https://test.pypi.org/project/justfit-llm/2.3.0rc24/`. A later installation check does not rerun or relabel a historical performance measurement.

The provenance records preserve these distinctions. `provenance/reference-artifacts.json` binds the historical source archive, wheel and sdist; `provenance/final-service-inputs.json` binds historical service inputs and receipt identities; `provenance/later-artifacts.json` identifies the subsequent native/staging revision. `provenance/public-preview.json` records the public preview and maps the index-built archive identities to the locally qualified package. Complete file contents agree even where archive metadata changes the outer SHA256. Some original audit receipts remain private; the published provenance supplies their identities rather than their private contents. Compiler output is not assumed to be universally deterministic.

Public preview installation. The preview pins 498 project files; its wheel, source archive and sdist contain the same 499-member project tree including the manifest. On each evaluated Mac, the qualified local distributions pass 113 source tests, 15 service tests and 30 installation/lifecycle checks. A shared prebuilt runtime is installed without receiver compilation, followed by text, two synthetic images, an eight-token thinking budget, streaming and cold-restart checks. Each restart case restores 8,704 disk KV tokens with one state hit and equal outputs. These checks have a smaller workload than the earlier 34,464-token service cases and do not replace them. A fresh M5 Pro environment additionally installs the public TestPyPI package, verifies its source contents, and passes the installation and runtime checks. Formal PyPI publication and Developer ID signing/notarization remain outside this completed preview.

Precompiled ANE assets. Two device-specific 4K/8K graph-and-cache archives per evaluated configuration are restored into an initially empty simulated user cache through the HTTPS installer, with certificate verification enabled. On each

device, both complete towers produce BF16 outputs byte-identical to their qualified references; every tower executes nine full-ANE/no-GPU groups. The 90 compiled files remain unchanged during loading and execution. Automatic phase-aware routing, memory admission and fallback are tested separately from forced-route timing studies. These receipts cover the exact evaluated chips and macOS 27.0.1 build 26A434, not portability across arbitrary chips, OS builds or compiler revisions. A simulated fresh cache is not a factory-clean machine. Public hosting of those large assets remains separate from the completed local HTTPS delivery experiment.

Installed graph assets and compiled file sets are checked against their SHA256 manifests before reuse and automatic-profile startup. Negative regression cases reject modified content even when size and modification time are preserved, missing or extra files, links and special files, without overwriting the user's cache. The default trust model still trusts local processes with the same user identity; these checks are not a sandbox against a concurrent malicious writer.

Scope. Synthetic dashboard and geometric fixtures avoid disclosing private screenshots; their expected answers and routing checks do not substitute for natural-image quality evaluation. Software tests, dependency checks and negative-input regressions provide bounded reproducibility evidence. They are not a complete native-code audit, fuzzing campaign or independent penetration test, and do not establish absence of vulnerabilities. The public software preview does not freeze this manuscript or turn installation evidence into a new performance dataset.