

The expensive signal keeps turning out to be *optional*.

Three separate literatures ran the same experiment on themselves this week and got the same answer. KV-cache eviction found that computing no score at all matches the best scorer. On-policy distillation found that the teacher's supervision is noise the student ignores, and that one training example recovers most of the gain from the whole dataset. Meanwhile the runtime shipped the opposite kind of result: sparse flash attention landed in llama.cpp, and it earns its keep only at depth. Three frontier models launched in three days, two of them with a gated sibling.

COMPILED 6 SEP 2026

5 SECTIONS

~65 PRIMARY CITATIONS

WINDOW 6½ DAYS

AT A GLANCE

01

A KV-cache evictor that computes no score at all matches the best one that does. Random Attention keeps the prompt and evicts uniformly at random within each head; across four models and six reasoning tasks it matches the strongest prior evictor while serving **32–43% higher throughput** in vLLM. The explanation is the interesting part: the prompt is the fragile part of the cache, and the reasoning trace protects itself by redundancy — the model restates what it needs, and every head keeps its own copy

2609.03430.

02

On-policy distillation's teacher is largely optional, and so is its dataset. Teacher supervision on student rollouts is substantially noisy — more so as the teacher gets bigger — and the student converges the same with the noise left in or taken out; a single fixed negative advantage matches teacher-provided ones [2608.31046](#). Separately, one-shot OPD on a *single query* recovers most of full-data OPD, reaching 71.5% of the states full-data training visits; sixteen queries reach 98.9% and match it [2609.04172](#).

03

Sparse flash attention landed in llama.cpp for DeepSeek-V4 and GLM, and the author's own numbers show it is a long-context feature only. On a DGX Spark with a 2-bit DSV4 quant, reported prefill speedup is $1.00\times$ at shallow depth, $1.19\times$ at 32K, $1.54\times$ at 128K and $2.20\times$ at 1M tokens [#27970](#). Metal followed two days later [#28098](#), and v0.4.0 was tagged on 4 September [#28386](#).

04

Three frontier launches in three days, two shipping a restricted sibling for security work. Claude Fable 5.1 and Mythos 5.1 on 1 September — Mythos being the same model with reduced safeguards, limited to trusted-access programs [post](#) — Gemini 3.8 Flash and 3.8 Flash Cyber on 2 September, Cyber gated to trusted defenders [post](#), and GPT-6 Astra on 3 September [changelog](#).

CONTENTS

§1 llama.cpp — sparse attention, and v0.4.0

§2 The papers — the audit widens

§3 From the labs — three launches, three days

§4 Weights that shipped

§5 What I would actually do

This window runs from the end of issue 03's compile to this one's, closing midday Sunday 6 September — a shorter container than a nominal week, stated honestly rather than rounded up. Two terms were added to the

glossary

.

Sparse attention, and v0.4.0

114 pull requests merged between 31 August and midday 6 September, ending at build `b10821`, with **v0.4.0** tagged on the 4th. Grouped by what changes for someone actually running the thing.

Sparse flash attention arrives — and only pays at depth

The week's structural change is sparse flash attention for DeepSeek-V4 and GLM #27970, built on an earlier proposal but reduced to something the runtime can actually carry: instead of modelling each architecture's sparsity scheme, it relies on a hint from the API about the maximum number of live KV entries per token. The author notes that further schemes — naming Qwen3.8-Flash-Next's QSA — can enrol later through the same mechanism. Metal support followed on 3 September #28098.

The reported measurements are unusually honest about when this does nothing. On a DGX Spark with a 2-bit DeepSeek-V4 quant (IQ2_XXS), against the immediately preceding build:

CONTEXT DEPTH	PREFILL (PP2048)	DECODE (TG32)
no depth	1.00×	1.00×
8K	1.04×	1.00×
32K	1.19×	1.03×
128K	1.54×	1.11×
512K	2.03×	1.31×
1M	2.20×	1.56×

Author-reported on one machine, so treat the magnitudes as claims — but the *shape* is the point, and it is the shape sparse attention should have. At the depths most people actually run, this is free and does nothing. It becomes the difference between usable and unusable somewhere past 100K tokens. If your workload is chat-length, updating for this changes your life not at all; if you are feeding a model an entire repository, it roughly halves the wait.

Worth reading beside it: the Qwen3.8-Flash-Next design paper, published on 31 August, which describes exactly the scheme llama.cpp is preparing to accept. Token mixing is a layer-wise hybrid of Gated DeltaNet

and global attention with one full-attention layer in every four, and at continued-pretraining time those full-attention layers are replaced by Qwen Sparse Attention, which scores context at micro-block granularity through a compressed lightweight indexer 2608.30320.

DFlash gets its round trip back

Issue 03 covered DFlash2 merging. This week the speculative path got faster in an unglamorous way: the drafter's encoder is a single `fc` plus norm, but running it as a separate `llama_encode` forced a device-to-host round trip of its output before the injection decode could re-upload it, plus a second graph build every round. Folding the encoder into the decoder's embedding branch removes both #27310. This is what the second week of a feature looks like when it is being taken seriously.

v0.4.0

Tagged 4 September #28386. The headline contents are initial Qwen3.8-Flash-Next support, NVIDIA Nemotron-3-Puzzle-75B-A9B #25444, on-demand tensor reading, per-slot server context limits, video input options, and a ggml bump to 0.23.0 carrying the sparse-attention and RDMA work. Note that a release aggregates more than one window — several items in its notes, including the lazy-loading and per-slot work, merged before 31 August and were covered here at the time.

New architectures inside the window: Tencent's Hy4 preview #28127, three days after the weights appeared (issue 03 flagged that card as thin on specs), and multimodal support for DeepSeek-V4-Flash-Vision-Exp #28133 with a vision input fix behind it #28154 — two days after that model shipped (§4).

Apple silicon: the campaign ends, and grows a CI gate

The per-chip flash-attention tuning campaign that filled half of issue 03's §1 finished the job: M1 #28078, M1 Ultra #28088, M2 Pro #28122, M2 Max #28015 and its remaining cases #28458, M3 #28236, M3 Max #28373, and the A18 Pro #28152. Separately, Metal 4.0's tensor API is now enabled on M5 and A19 and later #27461.

The detail worth stealing, though, is process rather than performance. Issue 03 reported a Metal memory leak caused by missing autoreleasepools. This week that bug class became a CI check #27884, explicitly excluding the known macOS-internal graphics-device leaks the project cannot fix, with the author noting the regex is imperfect and that having the test beats not having it. More leaks were fixed alongside it (#27883, #28399). Turning last week's bug into this week's gate is the cheapest quality move a project can make.

One more cross-backend pattern: radix top-k landed for Metal #28073, ROCm #27466 and Vulkan #28032, the last one explicitly for $k \geq 1024$ on Qwen3.8-Flash-Next. Sampling stopped being cheap when a model's vocabulary and expert count grew, and three backends fixed it in the same week.

Memory, and the server

The thread that runs through every recent issue continues: `llama-quantize` now streams row chunks rather than starving its threads #27830, and RAM no longer peaks during model loading #27483. On the server, `data:` URLs are accepted for `input_video` and `input_audio` #27735, `preserve_reasoning` is on by default with its effective state logged #28174, and `--log-jsonl` arrived for anyone shipping logs somewhere structured #28437.

ROUGH EDGES

A SYCL Kronecker-product FWHT implementation merged on 1 September at 15:47 UTC #28016 and was reverted **seventeen minutes later** for breaking `test-backend-ops` #28184, then restored on 5 September with the CI break fixed #28254. Issue 02 had the same story with a Q2_K SYCL revert-of-revert, and the lesson has not changed: if you build SYCL from source, pin past the restore, and never take a digest's word for whether something shipped.

§2 · PAPERS

The audit widens

Issue 03 reported the KV-eviction literature auditing its own benchmarks. This week the audit reached its conclusion, and two other subfields independently ran the same experiment on themselves: remove the expensive component, compare against a trivial baseline, and see whether anyone notices.

The score contributes almost nothing

Random Attention is the result the last two issues were walking toward. Every KV compression method shares one paradigm: score each cached token by some estimate of how much it will matter later, then keep the top scorers. This paper keeps the prompt, evicts *uniformly at random* within each attention head, and computes no score at all. Across four models and six reasoning tasks it matches the strongest prior evictor, while serving 32–43% higher throughput than it in a vLLM deployment — because scoring was costing something and buying nothing 2609.03430.

The controlled explanation is what makes it useful rather than merely provocative. First, **the prompt is the fragile part of the cache**, and most of the gap between published selectors is just whether their signal

happened to protect it. Second, the reasoning trace protects itself through redundancy at two levels: in the text, because the model restates what it still needs as it works, and across heads, because each keeps its own copy. Once the prompt is safe, a random draw retains enough copies of what the model needs, and no score is required to find them.

Read against issue 03, the arc closes. Trust the Mass showed that keeping the largest weights was already near-optimal and that published margins traced to storage accounting 2608.25230. TwinKV showed attention magnitude was uncorrelated with a token's causal contribution 2608.27128. This week says the obvious remaining thing out loud: stop scoring.

Except where the signal is somewhere else entirely

VestigeKV is the necessary counterweight, and it is not a contradiction. It attacks the case a long-lived cache actually faces: compression must happen *before* the queries that will read the cache exist, so selection by observed attention has nothing to observe. On a NoPE MLA model it does not degrade gracefully — H2O and SnapKV collapse to 0.00–0.33 needle retrieval. The fix is to evict by a query-independent signal the cache already carries: the 64-dimensional decoupled branch, a vestige of RoPE that NoPE training repurposes into a salience channel. Reading 11% of each row, retrieval holds at 1.00 under 8× compression and 0.92 under 32×, and the standard recall configuration holds 128× at 1.00 2609.03949.

THE CAVEAT PANEL — "THE SCORE DOES NOT MATTER" IS A CLAIM ABOUT YOUR CACHE, NOT ABOUT CACHES

These two results look opposed and are not, which is the most useful thing to take from the week. Random Attention evicts from a *reasoning trace*, which is redundant by construction, on models where selection was never doing much 2609.03430. VestigeKV evicts from a *long-lived document cache* on a NoPE MLA model, where the usual signal is not merely weak but absent, and the paper is explicit that the identical operator on a RoPE MLA collapses to 0.08 2609.03949.

So the transferable rule is not "scoring is pointless". It is that **the value of a selection signal depends on what is in the cache and how the model encodes position** — and that almost nobody reports which of those regimes their benchmark occupies. If you evict in production, the cheap experiment is now the informative one: run random eviction with the prompt pinned, and see whether your method beats it on your traffic.

On-policy distillation: the teacher was optional too

Issue 02 found four papers agreeing that OPD's per-token signal is too local. Issue 03 found the field naming collapse as the dominant failure. This week it went further and asked whether the teacher is doing anything at all.

Does On-Policy Distillation Really Distill? measures teacher supervision during training and finds substantial noise, whose prevalence *increases with teacher scale*. The student turns out to be insensitive to it, converging to comparable performance whether the noisy supervision is kept or removed. Digging into what actually drives the gains, learning concentrates on low log-probability tokens, and a single fixed negative advantage matches teacher-provided ones — so OPD works largely by suppressing low-probability tokens, which requires no teacher. Their supervision-free replacement, OPSA, uses entropy-adaptive negative advantages and reports a 35.41-point gain in Avg@32 on AIME24 over base Qwen3-1.7B, a 263% relative improvement 2608.31046.

Rethinking OPD II removes the other expensive input: the data. Training on a *single query*, one-shot OPD keeps improving for hundreds of steps and recovers most of full-data OPD's gain across task domains and model families. The mechanism is state coverage — the fraction of states full-data training visits that a query set's rollouts reach. One query already reaches 71.5%, most of it within the first 100 steps; sixteen semantically distinct queries reach 98.9% and match full-data training, including in the multi-teacher setting. Yet alignment slows at the same rate either way. The authors' verdict is the quotable one: *OPD is data-overfed but algorithm-starved* 2609.04172.

Two papers pull the other way, and they are the ones with something to do on Monday. **TGOPD** verifies teacher reliability per prompt before admitting dense supervision, routing unreliable prompts to verifier-grounded GRPO instead; it beats vanilla OPD in all six single-domain settings, and as a side effect puts the otherwise-idle teacher to work, raising teacher-node GPU utilisation from 9.8% to 78.9% in the measured run 2609.02998. And **Sequential Beats Joint** reports that plain OPD-then-RL outperforms pure OPD, pure RLVR, and every joint combination tried: OPD expands the student's coverage of teacher-supported solutions and RL sharpens within that support, while optimising both at once makes them interfere. The OPD validation score is the signal for when to switch, and OPD is a better cold start for RL than SFT 2609.04108.

Speculative decoding grows a training theory

Directly relevant to what llama.cpp merged last week. **Verification-Aware Training** points out a mismatch nobody had fixed: verification is sequential and discards everything from the first rejection onward, yet draft training uses token-level imitation with a fixed per-position weighting that reflects neither property. VAT adds a lightweight verification head and re-anchors the weighting decay to each sample's first rejection point. Applied to EAGLE-3 and DFlash on Qwen3-4B, Qwen3-8B and Llama-3.1-8B, it reports up to 11.4% better average acceptance length and up to 8.7% wall-clock speedup, while changing only the training objective 2608.30135.

The companion result is a free diagnostic for anyone running the block drafters llama.cpp now supports. Because DFlash and DFlare fill an entire block in one pass, the target often accepts the *whole* block, so the drafter runs out of trained horizon before verification ever fails — speed-up that exists but is never collected, which the authors call **stranded speed-up**. A mean committed length hides it; the acceptance histogram exposes it immediately as a spike in the ceiling bin. Widening the block at inference does not recover it,

because past its trained size the drafter's bidirectional attention shifts and erodes front-of-block verification; post-training on a longer block with a short curriculum does. Expanding DFlash and DFlare from block 16 to 24 on Qwen3-8B and Qwen3-4B raises per-prompt committed length by a median of +0.8 tokens, reaching 1.37 when continuation fine-tuning comes first 2608.30427. If you run a block drafter, plot the histogram before you buy anything.

Quantization: spend the next bit globally

The Structure of Quantization Damage is the most directly actionable paper of the week. Using causal mixed-precision intervention as ground truth — raise each layer to 8-bit in turn, measure the accuracy it recovers — across nine open-weight models in four architecture families, it tests the three intuitive hypotheses for where quantization damage lives: task circuits, where the model computes, and weight statistics. *None of them predicts which layers benefit from restored precision.* Recovery is diffuse: for eight of nine models, recovering 75% of the gap takes roughly half the layers. And at a matched precision budget, spending it globally on finer quantization granularity beats locally repairing the most recoverable layers — by 21 to 52 points, for every group-128-compatible model tested, including the one model whose damage was concentrated 2609.01587.

Two results either side of it. **Minima** puts NVFP4 W4A4 on all 496 linear layers of a hybrid 27B model, including the Gated DeltaNet blocks that community quants had been leaving at 8- or 16-bit on the intuition that errors accumulate in a recurrence. It matches BF16 within seed noise (5-task average -0.52) at 17.5 GiB with 14–19% faster prefill, and the mechanism study explains the intuition away: the supposedly fragile gate projections are the *least* sensitive, and the delta-rule recurrence holds injected noise at a flat plateau over 32K tokens because each write overwrites the state along the current key direction 2609.04098.

And a warning for anyone caching quantized KV in a RAG system. **Faithfulness Is Not Free** separates accuracy from groundedness, and finds INT8 near-lossless on both while INT4 damages faithfulness in a way accuracy cannot see: among answers that remain factually *correct*, over 90% of faithfulness changes are negative, and the harm grows under noisy retrieval and with more retrieved chunks 2608.30996. This is the third issue running in which 4-bit passes the standard guard and fails a behavioural one.

Two more worth knowing about, both about hybrid serving. **Tail-Replay** notes that a linear-attention layer is a lossy compression of its prefix, so a matched prefix's recurrent state can be reconstructed by replaying only a short recent suffix — which removes the need to store state checkpoints and unblocks token-level prefix reuse at arbitrary boundaries 2608.30310. And **DASC** derives per-head retention horizons from model weights to compress those checkpoints where they are still kept 2608.30386. Both are aimed squarely at the GDN and KDA models the open-weight world is now shipping.

Three launches, three days

Technical news from the large labs, cited to their own posts. This was the busiest week the section has seen: three frontier launches on three consecutive days, and a pattern in how two of them were packaged.

The gated sibling

Anthropic released **Claude Fable 5.1** and **Claude Mythos 5.1** on 1 September. Fable is priced at \$10 per million input tokens and \$50 per million output, with cache reads at \$0.25 per million; reported benchmarks include 55.8% on Terminal-Bench 4.0, 52.6% on Terminal-Bench-Science 0.1 and 77.9% partial on OSWorld 2.0. Mythos is described as the same model with reduced safeguards, available only through trusted-access programs and currently limited to US organisations [post](#), 1 Sep. An alignment and security post landed the day before [post](#), 31 Aug.

Google released **Gemini 3.8 Flash** and **3.8 Flash Cyber** on 2 September, at an introductory \$0.75 / \$3.75 per million tokens through 31 December. The Cyber variant is restricted to trusted defenders through a new Fairwind Program, and is reported above 70% on an internal vulnerability-discovery benchmark spanning 20 programming languages, with 47.2% pass@1 on CWE-Bench for patching [post](#), 2 Sep.

Two of the week's three launches therefore shipped as a public model plus a gated sibling aimed at security work — one by removing safeguards, one by adding offensive-security capability behind a vetting programme. Two instances is a pattern worth naming and not yet a trend worth theorising about; the thing to watch is whether the gated tier becomes the normal way frontier capability reaches security teams.

GPT-6 Astra, and an API built for long-running work

OpenAI released **GPT-6 Astra** on 3 September through both the Responses and Chat Completions APIs, recommending it for reasoning, coding, computer use, research and document creation. The constraints are as informative as the claims: it does not support the `none` reasoning-effort level, does not accept custom `temperature` or `top_p` values or log probabilities, and its tool calling requires the Responses API [changelog](#), 3 Sep. A model you cannot set a temperature on is a different kind of component to build against.

Shipped the same day, and more interesting for anyone building agents: three Responses API controls for long-running work — **async tool calling**, so the model keeps working while your function runs; **mid-turn steering**, sending further instructions over WebSockets while a response is in progress; and changing

reasoning effort mid-conversation while preserving cached prompts. Two smaller changes worth a line: since 2 September, traffic ramping too quickly returns a `429` with a `slow_down` code while genuine model overload returns a `503`, which is a retry-logic change for anyone with a backoff loop; and from 1 September `api.openai.com` is reachable over IPv6.

The rest of the week

Google also shipped agentic video understanding on 1 September for Gemini 3.7 Flash, 3.6 Flash and 3.5 Flash-Lite: set the processing parameter to `agentic` and the model runs an internal loop to load only the relevant part of a video, reported at up to 88% fewer tokens, 66% lower cost and 7% better accuracy, at standard token pricing [post](#), 1 Sep. And **WeatherNext 3** arrived on 3 September with hourly forecasts at 5 km surface resolution and up to 60% better CRPS on precipitation — but, unlike the WeatherNext models issue 03 noted for publishing weights, this one is reached through BigQuery, Earth Engine and Cloud Storage rather than downloaded [post](#), 3 Sep.

Hugging Face's blog ran three practitioner pieces on 3 September: *NeoMME*, a multimodal-native multilingual encoder [post](#); fine-tuning a 350M model for structured outputs in 100 GRPO steps [post](#); and *Funes*, a local memory system that indexes coding-agent session traces into searchable Lance datasets, with embeddings and reranking done locally, so agents can recall past reasoning without shipping it anywhere [post](#).

Meta published nothing again; its most recent posts remain July's Muse line. That is two issues running, and recording it is the point of a section that reports on labs rather than for them.

§4 · RELEASES

Weights that shipped

Open weights only; specifications and figures are as stated on the model cards. Download and like counts are as of compile time.

DATE	MODEL	WHY IT MATTERS
31 Aug	DeepSeek-V4-Flash-Vision-Exp	MIT, ~305B parameters, multimodal (image-text-to-text), published with FP8 weights model card . 209K downloads and 701 likes in its first week, the clear release of the window. llama.cpp multimodal support merged two days later (§1), and this is one of the two architectures the new sparse flash attention path targets — the runtime and the release are converging in the same week for once.
1 Sep	K2-Horizon-MoVA-36B-A4B	Apache 2.0, 36B total with 4B active , a MoVA mixture-of-experts from IFM shipped with its pretraining dataset referenced on the card model card . The size class that matters for local work: 4B active is a laptop-class decode cost, and the licence is the permissive one.
2 Sep	vdn-minimax-h3	A text-to-video fine-tune of MiniMax-H3 under a custom licence model card . Outside this publication's remit to evaluate, recorded because it is an open-weight release with real attention on it.

Two technical reports landed this week for models whose weights did not: Instella-MoE and A.X K2 both published write-ups in the window, but the checkpoints have been on the Hub since July — out of window, and listed here so the dates are not mistaken for a release.

§5 · TAKEAWAYS

What I would actually do

Sorted by readiness, not by how interesting the thing is.

ADOPT

If you serve DeepSeek-V4 or GLM past about 100K tokens, update for sparse flash attention. The reported curve #27970 is the whole decision: nothing at chat length, 1.54× prefill at 128K, 2.20× at 1M. Update anyway if you are on Metal or Apple silicon generally, since the per-chip tuning campaign and the Metal path landed in the same window — but do not expect the sparse work to show up in a short-context benchmark, and do not conclude it is broken when it does not.

ADOPT

If you cache quantized KV in a RAG pipeline, measure faithfulness, not just accuracy — and prefer INT8. 2608.30996 reports INT8 near-lossless on both metrics, and INT4 degrading groundedness invisibly: over 90% of faithfulness changes negative *among answers that stayed correct*. Your accuracy dashboard cannot see this by construction. That is now three consecutive issues finding 4-bit passing the standard guard and failing a behavioural one; at some point that stops being a coincidence and starts being the default assumption.

WATCH

Before adopting any KV-eviction method, benchmark it against random eviction with the prompt pinned. That is a two-line experiment and this week it became the only honest baseline 2609.03430. If your method does not beat it on your traffic, you are paying compute for a score that is not doing anything. The caveat in §2 matters though: on a long-lived document cache, or a NoPE model, the picture inverts 2609.03949 — so run the baseline on *your* workload rather than adopting either conclusion wholesale.

WATCH

Spend your next quantization bit on finer granularity, not on protecting "critical" layers. Nine models, four families, three intuitive theories of where the damage lives, none of which predicts anything — and global granularity winning by 21 to 52 points at matched budget 2609.01587. If you maintain a mixed-precision recipe with hand-picked protected layers, this is the paper that says the hand-picking was probably never earning its keep.

WATCH

If you run a block drafter, plot the acceptance histogram. DFlash2 merged into llama.cpp last week, and this week's companion result shows a mean committed length can hide speed-up you already paid for but never collect 2608.30427. The histogram is free, and a spike in the ceiling bin tells you the block is too short before you spend anything on training a wider one.

LAB ONLY

The teacher-optional OPD results. Both are credible, both are one week old, and both are one paper (2608.31046, 2609.04172). What I would take now is the cheaper, better-supported pair of findings beside them: run **OPD then RL** rather than fusing them, using the OPD validation score to decide when to switch 2609.04108, and gate dense supervision on per-prompt teacher reliability instead of trusting it everywhere 2609.02998. Both are changes to sequencing rather than to the algorithm, which makes them cheap to try and cheap to undo.

Three fields, three "the expensive part was optional" results, and one shared method. That is the week's shape and it is worth being careful about. These are different subfields that did not coordinate, which makes the convergence more interesting than issue 02's four-papers-on-one-topic — but each result is still a single paper, none has been independently reproduced, and "the component contributed nothing *on the benchmarks we ran*" is a narrower claim than it sounds. What genuinely transfers is the method, not the conclusions: take the expensive component out, compare against the most trivial baseline you can construct, and publish the result even when it embarrasses the technique. That is cheap, it is available to anyone, and on this week's evidence it keeps working.

Sources

LLAMA.CPP §1

[llama.cpp releases](#) — [v0.4.0](#) (4 Sep) and builds through [b10821](#) (6 Sep).

Merged pull requests 31 Aug – 6 Sep were enumerated from the GitHub API (114 merged), and every PR cited below was fetched individually to confirm it merged.

[#27970](#) CUDA + ggml sparse-fa for DeepSeek-V4/GLM · [#28098](#) Metal sparse FA · [2608.30320](#) On the Design of Qwen3.8-Next Architecture

[#27310](#) fuse the DFlash encoder into the KV injection decode

[#28386](#) version bump to 0.4.0 · [#25444](#) Nemotron-3-Puzzle-75B-A9B · [#28127](#) Tencent Hy4 preview · [#28133](#) DeepSeek-V4-Flash-Vision-Exp · [#28154](#) DeepSeek-V4 input vision

Metal fa-vec tunings: [#28078](#) M1 · [#28088](#) M1 Ultra · [#28122](#) M2 Pro · [#28015](#) and [#28458](#) M2 Max · [#28236](#) M3 · [#28373](#) M3 Max · [#28152](#) A18 Pro · [#27461](#) Metal 4.0 tensor API

[#27884](#) CI check for missing autoreleasepools · [#27883](#) and [#28399](#) leak fixes

Radix top-k: [#28073](#) Metal · [#27466](#) ROCm · [#28032](#) Vulkan for $k \geq 1024$

[#27830](#) quantize row-chunk streaming · [#27483](#) prevent RAM peaking at load

Server: [#27735](#) data: URLs for video and audio · [#28174](#) `preserve_reasoning` by default · [#28437](#) `--log-jsonl`

SYCL FWHT saga: [#28016](#) merged · [#28184](#) reverted 17 minutes later · [#28254](#) restored

PAPERS §2

[2609.03430](#) Random Attention: Rethinking KV Cache Eviction for Efficient Reasoning — code

[2609.03949](#) VestigeKV: The NoPE-MLA KV Cache Carries Its Own Eviction Signal in a Vestigial Branch

[2608.31046](#) Does On-Policy Distillation Really Distill? From Noisy Teacher to Self-Improvement

[2609.04172](#) Rethinking On-Policy Distillation of Large Language Models II: One Training Example

[2609.02998](#) Verify Before You Distill: Prompt-Level Teacher Gating for On-Policy Distillation

[2609.04108](#) Sequential Beats Joint: On the Interplay between On-Policy Distillation and RLVR

[2608.30135](#) Verification-Aware Training for Speculative Decoding

[2608.30427](#) Ceiling-Clipped Acceptance Histograms Indicate Stranded Speed-up in Block-Diffusion Speculative Decoding

[2609.01587](#) The Structure of Quantization Damage in LLMs

[2609.04098](#) Why Gated DeltaNet Survives 4-Bit Quantization: NVFP4 W4A4

[2608.30996](#) Faithfulness Is Not Free: Auditing Offline KV-Cache Quantization in RAG

[2608.30310](#) Tail-Replay · [2608.30386](#) DASC: Decay-Aware State Compression

Cited from issue 03 for continuity: [2608.25230](#) Trust the Mass · [2608.27128](#) TwinKV

LABS §3

[Claude Fable 5.1 and Mythos 5.1](#) (1 Sep) · [alignment and security](#) (31 Aug)

[Gemini 3.8 Flash and 3.8 Flash Cyber](#) (2 Sep) · [agentic video understanding](#) (1 Sep) · [WeatherNext 3](#) (3 Sep)

[OpenAI API changelog](#) — GPT-6 Astra and the Responses API long-running-work controls (3 Sep), 429 slow_down versus 503 (2 Sep), IPv6 (1 Sep)

Hugging Face blog, 3 Sep: [NeoMME](#) · [GRPO for structured outputs](#) · [Funes](#)

RELEASES §4

[DeepSeek-V4-Flash-Vision-Exp](#) · [K2-Horizon-MoVA-36B-A4B](#) · [vdm-minimax-h3](#) — specifications and figures are as stated on the model cards.

[Hugging Face daily papers](#) — used for discovery across the window; every paper it surfaced was read at its arXiv abstract before being cited.

Compiled 6 September 2026 from the llama.cpp release feed and GitHub API, the arXiv API filtered to this window, Hugging Face's curated daily papers, and the labs' own newsrooms and changelogs. Figures quoted as speedups, benchmark scores or error rates are the authors' or vendors' own reported numbers on their own benchmarks — treat them as claims, not measurements, until we reproduce them. The window opens where issue 03's compile stopped and closes at this one's, midday Sunday 6 September; anything dated outside it is labelled inline as context. Terms are defined on the [glossary page](#).



LLM Radar is published weekly by [Montevive](#). [Atom feed](#) · ai@montevive.ai · [Privacy](#)