

Skills compress. *Facts* don't.

Five weeks in one issue, because the weekly cadence paused after issue 04 and this window opens exactly where that one closed. The papers converged on one asymmetry from two directions that do not cite each other: on-policy distillation moves how a model uses what it knows, not what it knows, and aggressive quantization keeps the fluency while the facts go first — in both cases, the standard guard metric measures the part that survives. Meanwhile llama.cpp tagged three releases, taught big MoEs to cache their experts on the GPU, made speculative decoding pay on Apple silicon, and changed its default port. And a startup's new kind of model, TypeSafe's Jev, which answers typed questions with probabilities instead of generating text, had open imitators within three days and llama.cpp support within three weeks.

COMPILED 10 OCT 2026 5 SECTIONS ~150 PRIMARY CITATIONS WINDOW 34 DAYS

AT A GLANCE

01

On-policy distillation teaches skills, not facts. In a controlled study across four models from three families, reverse-KL OPD transfers compositional skill to unseen problems but "transfers minimal factual knowledge"; forward KL restores it ^{2610.09639}. A crosscoder study finds OPD leaves over 98% of the student's frequently used features within 20% of their firing rates ^{2609.35210}. Four groups, four methods, one answer: OPD reorganises what the student already has.

02

Aggressively quantized small models keep the words and lose the meanings — and perplexity cannot tell. Across 27 GGUF artifacts, every sub-2B model at Q2 loses 20–67% of its ability to define rare words it can still use, while two artifacts with near-identical perplexity penalties lose 3.6% and 43.6% of their definitions. Q4_K_M stays clean at 1B and above 2610.04403.

03

A new model class formed in three weeks: JEV-style decision models. TypeSafe launched Jev on 15 September, a hosted model that takes a state and typed questions and returns a value with a calibrated probability for each, without generating text `post`. Open models on the same `/v1/systemone` format followed within days — OpenJev reports 84.0% against hosted Jev's 85.4% `model card` — llama.cpp adopted the format on 2 October #29818, and OpenAI shipped its own Decisions API on the 6th `changelog`.

04

llama.cpp got faster where it hurt most. The opt-in `--moe-cache-mib` flag keeps a big MoE's recently used experts in VRAM when the model lives in system RAM, reported at 1.57–2.20× decode on a single RTX 4090 or 5090, with prompt processing dropping to 0.73× in the author's own two-GPU table #29887 #30112. And on M1–M4 Macs, where DFlash2 had been *slower* than plain decoding, a few-row matmul kernel takes it from 30.2 to 110.0 t/s on code against 32.1 serial #29869.

05

Before you update llama.cpp, read three lines. The server's default port is now 9931 #30159, `--mmap` / `--mlock` are gone #28334, and old saved sessions no longer load #28498. If you expose `rpc-server`, the first tag with the fix for an unauthenticated remote-code-execution bug is v0.5.0 #24292.

06

The release of the window does not run upstream yet. DeepSeek-V4.1-Flash, MIT-licensed, 763B stored with 8B active in prefill and 16B in decode, cuts its global KV cache to 890 bytes per token 2609.19969; its llama.cpp PR has been open since release day #28696. Of fourteen rows in §4, eight run on stock llama.cpp today.

07

Gemini 4 Argon is announced, not available. Google announced it on 30 September at an introductory \$2 in and \$10 out per million tokens with a 1M output-token limit, but staged: vetted cyber defenders first, and to them without cyber guardrails, then the paid API. At compile time it has no public model ID and no Gemini API changelog entry `post`, 30 Sep.

§1 llama.cpp — three tags, a new port, and MoE that no longer has to fit

§2 The papers — skills compress; facts don't

§3 From the labs — a new model class, two price points, fewer knobs

§4 Weights that shipped

§5 What I would actually do

This window runs from issue 04's compile, midday Sunday 6 September, to this one's on Saturday 10 October — 34 days, the size of issue 01's month rather than a week, so the sections are longer and the selection stricter. Four terms were added to the

glossary

. **Corrected 10 October:** the first version covered TypeSafe's Jev and the JEV-style models only as generic "decision models", without naming Jev or its launch; §1, §3, §4 and §5 now do, and Gemini 4 Argon was added to the items above. A second pass, prompted by a new coverage check against Hacker News, Hugging Face and GitHub, added Strata (§1), and Cognition's SWE-2, OpenAI's Dots, mathematics results and GPT-6 rollout, and Z.ai's ZCode (§3).

Three tags, a new port, and MoE that no longer has to fit

718 pull requests merged between midday 6 September and 10 October, from build `b10822` to `b11540` — about 150 a week, faster than the summer's pace. Three versions were tagged in the window: **v0.4.1** on 14 September notes, **v0.5.0** on the 23rd notes and **v0.6.0** on 5 October notes, so none of them is "the release" — and several of the changes below merged after v0.6.0 and exist only in nightly builds. Grouped by what changes for someone running the thing.

A MoE bigger than your VRAM gets a cache

The change most likely to alter what people see is a GPU-side LRU cache for mixture-of-experts weights that live in system RAM #29887. Until now, a MoE too large for the card ran its offloaded experts from host memory on every token. The new `--moe-cache-mib N` flag keeps recently used experts on the GPU and uploads only the misses; it engages only for small batches, so it is a decode feature, and the author suggests sizing it at about 10% of total expert size. Multi-GPU support followed a day later #30112.

The reported numbers are on Qwen3.8-Flash-Next at Q4_0 (93.7 GiB, of which 65.4 GiB is experts) on a 16-core EPYC over PCIe 4.0, decode against the existing `--fit` path:

SETUP	DECODE T/S	SPEEDUP	CACHE HIT
RTX 4090, <code>--fit</code> + 6.5 GB cache	25.0 → 39.4	1.57×	72%
RTX 4090, <code>--cmoe</code> + 11 GB cache	25.0 → 40.7	1.62×	77%
RTX 5090, <code>--fit</code> + cache	30.8 → 54.5	1.77×	77%
RTX 5090, <code>--cmoe</code> + 19 GB cache	30.8 → 67.8	2.20×	89%
2× RTX 4090	34.8 → 60.4 / 64.4	1.74–1.85×	—

One machine, one model, one author, so the magnitudes are claims. The detail that should be on every summary of this and usually will not be: in the multi-GPU table, **prompt processing drops to 0.91× and 0.73×** of the baseline. It is opt-in, it is per GPU, and it is not in any tagged release yet. If you run a big MoE on one consumer card for chat, it is worth trying on a nightly; if you batch long prompts across two cards, measure before you keep it.

The same problem was attacked downstream, and drew more attention than anything in the main repository: **Strata**, an MIT-licensed one-click installer and engine for Windows and Linux that "uses parts of llama.cpp / ggml", runs Qwen3.8-Flash-Next on a gaming PC with a 12 GB card and 32–64 GB of RAM, behind an OpenAI- and Anthropic-compatible API on localhost. Its author reports 94 t/s decode and 2,650 t/s prompt processing at 32K on an RTX 5070 with the Q2_0 quant, and 53 t/s at IQ3_S repo. It passed 20,000 GitHub stars in two weeks. Read §2's quantization audits beside the Q2 numbers: they found models above about 3B mostly robust at Q2 where small ones were not, though the largest they tested was 14B, but "validation must be per artifact".

Speculative decoding grows up: correct above zero, and fast on a Mac

Two changes turn speculative decoding from a datacentre trick into something a laptop owner should switch on. First, on Apple GPUs without the tensor API — M1 through M4 — the 2–16-row matrix multiplies that verification needs were running on mat-vec kernels whose cost grew with every row. The author states plainly that on an M3 Ultra, DFlash2 decoding of Qwen3.8-27B was therefore *slower* than plain decoding on master. A few-row MMA kernel fixes it #29869, extended to the remaining quant types two days later #30065. Reported on an M3 Ultra with a Q4_0 target and a Q8_0 drafter: code at temperature 0 from 30.2 to 110.0 t/s, prose from 16.8 to 62.6, against 32.1 for serial decoding, which is unchanged. If you tried DFlash on an M-series Mac this summer and concluded it did not help, you were right, and you should try again.

Second, speculative sampling is now correct at temperature. Model-based drafters used to offer only their top token, and the target accepted it only on an exact match, so acceptance collapsed as temperature rose. The standard rejection-sampling construction is now available behind `--spec-draft-sampling probabilistic` #27694, keeping the output distribution exact. Gains rise with temperature — a Llama-3.1-8B target with a 3B draft goes from +0.07% at T=0.2 to +16.6% at T=0.9 on an RTX 5090 — and some configurations regress slightly at low temperature, which is why the default stays greedy. It applies to simple drafts and MTP; the author reports DFlash does not benefit. It also shipped a bug: n-gram drafts near the end of a reply were fully rejected at $T > 0$ until #29924 two days later.

New architectures, and one that was quietly broken

GLM-5.3-Flash landed on 30 September #27773, described by the author as a 320B text-and-vision hybrid of 34 KDA linear-attention layers and 11 DSA layers, reusing the KDA and hyper-connection code from earlier models; MTP followed a week later #29928. **Qwen3.8-Flash-Next**, whose support issue 04 reported arriving, turned out to be "broken on current master" in the words of the PR that fixed it on 1 October #29751. The same day it gained MTP drafting, reported at an overall $1.55\times$ decode on a DGX Spark, ranging from $1.43\times$ to $1.67\times$ by prompt category #29761. If you formed an opinion of that model in llama.cpp in September, form it again.

Smaller, but it touches every hybrid model people run: Gated DeltaNet's q/k normalisation differed slightly from the reference implementations, affecting the Qwen3.5 and 3.8 hybrids, Kimi-Linear and Kimi-K3, among others #28068. The measured KL divergence change is tiny; it is the kind of fix you get for free by updating.

The genuinely new thing is a model *class*: JEV-style decision models, which answer typed questions with a probability per option instead of generating text (§3 has the story of TypeSafe's Jev and its open imitators). The runtime adopted Jev's own wire format, a `/v1/systemone` server endpoint, with Laya, Julia-1, lev, Kev and OpenJev supported from the first PR, whose author calls the open ones "fancy wrappers around traditional embedding models" #29818. They were followed within nine days by Cloudflare's Clef #29831,

Perplexity's pplx-decider #30044 and Liquid AI's d1 models #30110, among ten PRs. Ten PRs in nine days is a category forming, not a one-off. Also merged: a ternary 20B-A1B MoE, Maple, CPU-only for now #27000, and a looped model, HRM-Text #27625 — worth noting given how much of §2 is about looped models.

Long context, on every backend

Sparse flash attention, which issue 04 covered landing for CUDA and Metal, spread across the window to Vulkan #28105, SYCL #28796 (off by default, behind `GGML_SYCL_SPARSE_FA=1`), CUDA for Qwen3.8-Flash-Next #28770 and Vulkan with a quantized KV cache #29639. The shape issue 04 described has not changed: reported gains are $1.08\times$ prefill at 10K and $1.26\times$ at 100K on Qwen3.8-Flash-Next, and nothing at chat depth. One thing to know if you run DeepSeek-V4: #28770 switched its sparse path *off* because of a batch-dependent gate, and prefill was turned back on two days later #29298.

Dense long-context prefill got faster in three places. Metal's tensor API now drives flash attention on the M5, reported at 174 to 253 t/s for pp512 at 32K depth on an M5 Pro, with decode unchanged #29570. CUDA's two-stage FA kernels now schedule whole tiles, from +1.4% to +33.5% on 64K-token prefill on a DGX Spark and at most +4.8% on an RTX 5090 #29435. And ROCm on RDNA4 went from 164 to 399 t/s at 150K depth on Qwen3.5-72B, partly by fixing a bug that kept the head-size-256 kernels from being selected at all #28102.

Elsewhere, in one line each: CPU prefill on k-quants got a tiled VNNI kernel, reported at $3.8\text{--}7.0\times$ — a GEMM microbenchmark, not end-to-end throughput #27851; AMD RDNA3 and 4 got int8 matrix-core matmuls under Vulkan, $1.25\times$ on Llama-3-8B prefill on Strix Halo #27952; gpt-oss decode roughly doubled on Intel Arc Pro under SYCL on Windows, in the window's last build #29809; direct-IO loading stopped holding each tensor twice, halving peak RSS on a DGX Spark #29749; and Qualcomm's Hexagon NPU backend is now a steady track, with 44 PRs in the window and Snapdragon builds in the releases #29007.

Three changes that will break a working setup

The server's default port is now 9931, not 8080 #30159, from build b11521 on 9 October. Docker images keep 8080 inside the container. Anything with `localhost:8080` hard-coded that updates a native build needs `--port 8080`. **The deprecated `--mmap`, `--mlock` and `--direct-io` flags are gone**, replaced by `--load-mode` #28334, so old launch scripts fail to parse from v0.4.1 on. And **sessions saved by older builds no longer load** after a session-format bump that ships in v0.6.0 #28498.

On the server side, the useful additions: context checkpoints now survive slot save and restore, so restoring a slot on a hybrid or sliding-window model no longer reprocesses the whole prompt #26004; `--host` accepts a list of addresses and Unix sockets #28690; causal-LLM rerankers no longer fail on documents longer than the physical batch #28876; and the web UI became a model manager that downloads, loads, unloads and deletes models #29583.

IF YOU EXPOSE RPC-SERVER, READ THIS FIRST

A use-after-free in the RPC server, reachable by an unauthenticated client and leading to remote code execution, was fixed on 16 September #24292 (advisory GHSA-2phh-px2f-2qmx). It merged two days *after* v0.4.1 was tagged, so the first tagged release that contains it is **v0.5.0**. A separate out-of-bounds write also reachable over RPC was fixed on 9 October #30135 and is in nightlies only. The loader was hardened against crafted GGUFs in the same window (#29598, #29384). The RPC server was never meant to face an untrusted network; this is the reminder of why.

ROUGH EDGES

A change letting auto-fit size a unified KV cache across parallel slots merged on 16 September #28849 and was reverted on the 26th #29437, with a reviewer noting that it raised VRAM requirements a lot for anyone filling the card with a model. A Vulkan MoE tiling change that gave +10% prefill on Strix Halo #29182 roughly halved 8K prompt processing on Intel Arc and was reverted for Intel only after a bisect #29936.

Two deliberate trade-offs to know about: Metal MXFP4 now computes in bf16 for correctness, and the author accepts a slowdown on pre-M3 chips, $0.81 \times$ prefill on an M2 Ultra #29770; and a GLM-5.3-Flash change that speeds up decode with flash attention on drops 32K decode from 72.3 to 24.6 t/s with `-fa off` at small micro-batches, in the author's own table #30042. Still open at compile time: MTP roughly halving prompt processing on multi-GPU layer splits *issue* #27428, and Gemma 4's thinking output growing trailing garbage *issue* #28827. Another window, another revert-and-reland; pin the builds you depend on.

Skills compress; facts don't

Five weeks of arXiv, swept by week so no part of the window was truncated, plus Hugging Face's daily papers — 1,084 of them in window. Two literatures that do not cite each other converged on the same asymmetry. Distillation moves how a model uses what it knows, not what it knows. Aggressive quantization keeps the fluency and loses the facts first. And in both, the standard guard metric measures the part that survives.

On-policy distillation reorganises; it does not add

Issue 04 asked whether OPD's teacher was doing anything at all. This window answered from four independent directions, with four different methods, and the answers agree.

The cleanest is a controlled synthetic study that varies the teacher's extra *facts* and extra *compositional skill* separately. Across four models from three families, reverse-KL OPD transfers the skill to unseen reasoning structures but "transfers minimal factual knowledge"; switching to forward KL restores factual transfer, and the same asymmetry shows up on real factual QA and competition maths 2610.09639. A mechanistic study with sparse crosscoders reaches the same place from inside the network: OPD "neither creates features nor passes on the teacher's own", leaving the firing rates of over 98% of the student's frequently used features within 20% 2609.35210. Framing the teacher as an implicit reward model, a third paper finds OPD "improves performance without expanding the student's capabilities" — it makes correct responses easier to sample, and when the implicit reward is misaligned, it rewards overlong, repetitive rollouts 2610.03185. And a fourth traces OPD plateaus with large teachers to a vanishing learning signal: relative parameter change of 0.025–0.098% and layer-wise similarity above 0.98, against far larger loss reductions when the teacher is the student's own RL-trained successor 2610.11247.

The recipe papers line up behind it. A ± 1 reward on tokens where the teacher's probability is above or below the student's "reproduces almost the same training mode as OPD with reverse KL" 2609.33791. A systematic study finds the rollout policy — the thing that puts the "on-policy" in OPD — "does not necessarily play a central role"; KL direction and learning rate do, and on-policy's advantage "does not reliably persist after subsequent RLVR" 2609.35259. Scaling laws for same-family OPD find peak score improves with teacher size "only up to roughly the student's scale", that "at a matched gold score smaller teachers transfer better", and that in every weak-to-strong pair observed the student beat its teacher 2609.32722. Multi-teacher OPD does not beat a student taught by the best single specialist, because instruction-following feedback is spread wider and dominates the updates 2609.35347.

None of that makes OPD useless; it makes it a different tool. If a model is reorganising skills it already has, cheap OPD is the right OPD, and three papers made it much cheaper: supervising 0.1–1% of tokens matches full OPD 2609.24432; restricting cross-tokenizer OPD to a student-chosen top-16 slice of the shared vocabulary matches the full version 2610.08448; and SparseOPD cuts backward memory by 70.5% at 99% gradient cosine similarity 2609.34386. Two failure modes with fixes are worth knowing if you run it: students and teachers can put their stop probability on *different* EOS tokens even with identical declared stop sets, which inflates length until equivalent EOS tokens are treated as one stop action 2609.20511; and the pointwise clipping adopted from the original on-policy self-distillation recipe can push token probabilities *away* from the teacher, producing repetitions that persist to the end of the response 2609.38995.

The most-upvoted paper of the theme points the other way, and it is worth reading for that: RIDE treats the RL teacher's change from its pre-RL checkpoint as a direction in hidden-state space and extrapolates *beyond* the teacher along it, approaching or exceeding the RL teacher on all four pairs tested 2609.36484. Note what it extrapolates: the RL residual, a skill. It is consistent with the rest of the section, not a counterexample.

Quantization: the damage is somewhere the guard metric does not look

Issue 04 ended with the third consecutive week in which 4-bit passed the standard check and failed a behavioural one. This window produced a cluster of audits that measured exactly that, several of them on llama.cpp's own formats.

Saying, Not Knowing audits 27 GGUF artifacts from 13 families, 0.35B to 14B, down the published ladder to Q2_K, on 429 rare English words, asking two questions: can the model still *use* the word, and can it still *define* it? In every sub-2B artifact at Q2, definitions fall 20–67% below the artifact's own baseline, typically several times the loss in usage. Parameter count predicts the damage (Spearman $\rho = 0.72$); tokenizer vocabulary does not ($\rho = 0.12$). WikiText-2 perplexity cannot tell the cases apart: two artifacts with near-identical Q2 perplexity penalties, 44.7% and 47.6%, lose 3.6% and 43.6% of their definitions respectively. The reassuring half: Q4_K_M "remains lexically clean at $\geq 1B$ " 2610.04403.

The same shape at other scales and on other behaviours. Ternary models trained from scratch retain as little as 6% of an identically trained fp16 model's factual capacity while perplexity rises only $1.4\text{--}1.6\times$ — at 50M parameters and below, so a mechanism, not a verdict on BitNet-class models 2609.32902. Measuring full-vocabulary divergence rather than accuracy, a 120-run matrix finds accuracy "unstable, non-monotonic" under progressive llama.cpp quantization, and mixed-precision Q4_K generally closer to BF16 than uniform Q4_0 at similar size 2609.07664. On agentic tool use, three Polish models all fall off a cliff between 3 and 2 bits — 0.716 to 0.045 for the 11B — but fail in different ways: the pruned 7B fails long, the 11B confabulates a final answer at step one 2609.32042.

THE CAVEAT PANEL — YOUR QUANTIZATION VERDICT MAY BE AN ARTIFACT OF HOW YOU SCORED IT

Comparing 8-bit and 4-bit Llama-3.1-8B on tool-failure recovery, one paper gets every answer from the same logs. Scored on each model's own clean tasks, 4-bit wins by 17.5 points; on matched tasks, no difference; on the full pipeline, 8-bit wins by 28.3 — and strict output parsing turns that +28.3 into −15.0. On Qwen2.5-7B the difference across prompts alone runs from −50 to +35 points 2610.07781.

Put that beside the GGUF audit above, whose conclusion is that "validation must be per artifact", and the transferable rule is uncomfortable but cheap to follow: **a quantized model passing perplexity tells you it is fluent, not that it still knows things or can still use tools**, and a single prompt, task set and scoring policy does not establish a stable verdict either way. For a quant you actually deploy, test the behaviour you depend on, with more than one prompt.

Three findings about the *tools* of quantization, rather than its results. Lower reconstruction loss can degrade a model on the same calibration data it was minimised on 2610.11226. A mixed-precision study found one run that improved a block's own objective $4.6\times$ while perplexity rose $32\times$, and a rule based on which matrix a weight sits in beat every sensitivity estimate they computed — 96.4% of weights at 4.5 bits per parameter

within 4.4% of full-precision perplexity on Mistral-7B [2609.34866](#). That echoes issue 04's finding that intuitive theories of where quantization damage lives predict nothing. And the most hands-on result of the window: **the documented export pipelines for BitNet, Falcon-E and BitCPM cast latents to bf16 before ternarising**, bf16 rounding lands values exactly on the threshold, and ties-to-even maps them to zero. At fine-tuned endpoints this lowers greedy GSM8K strict accuracy from 58.79% to 0.78% for Falcon-E-1B — the export silently reverts the fine-tune [2610.07853](#).

Quantization you can use

The window also produced quantization work aimed straight at the runtime in §1. **Disaggregated Quantization**, from IST Austria, starts from the observation that prefill is compute-bound and rewards low-precision arithmetic while decode is memory-bound and rewards compact weights, and uses different formats for each. With released 1-bit Qwen3.8-27B GGUF decoders, training an NVFP4 *prefiller* improves accuracy by 32.5 points on MMLU-Pro without touching the decode checkpoint, and streaming prefill weights from SSD gives a reported $1.78\times$ time-to-first-token speedup at 8K in llama.cpp [2609.26333](#). **QATFactory** is an open framework for quantization-aware training and distillation that exports directly to llama.cpp's Q4_K and to vLLM without a second lossy conversion; on Qwen3.5-9B its QAD reaches 68.9% average accuracy under NVFP4 against 65.4% for the best post-training method [2609.39223](#). **Tetra** serves Leech-lattice codebooks at about 2.7 bits per parameter and, at 4B, scores 23.6 points above llama.cpp's IQ2_XXS on its benchmark [2609.35465](#). And issue 04's Minima quantized hybrid models' Gated DeltaNet weights; **STEPQuant** quantizes their recurrent *state*, which becomes the memory bottleneck under concurrent serving, closely matching FP32-state accuracy at a nominal 6-bit budget and cutting total serving memory by up to 68.7% in SGLang [2609.38169](#).

Looped models become an efficiency axis

Models that run the same block several times per token — looped, or recurrent-depth, models — went from about 1.5 to 9.3 papers per thousand cs.CL and cs.LG submissions across the window, and were still rising in the last week. The papers have moved from whether looping works to how to serve it. A scaling law that jointly models recurrence and sparsity finds a looped MoE matching a roughly $2\times$ larger non-looped one on reasoning at matched training compute [2609.40316](#). Letting only the first loop write a KV cache, with later loops reading it, *improves* perplexity while using 76–79% less context memory at 150M–1B [2610.02383](#). Shallow loops can draft for deeper ones, a training-free self-speculation reported at $2.42\times$ on Ouro-2.6B and $3.54\times$ on Huginn-3.5B [2609.23033](#). And the intuition that rounding error accumulates loop after loop "holds only for loops that never settle" [2609.39277](#).

The paper that connects it to the rest of this section: a controlled study finds recurrence can improve reasoning beyond the training horizon *while degrading knowledge* [2609.36636](#). It is one paper, and the weakest leg of this issue's thesis — but it is the same asymmetry a third time.

Serving: two audits and a 35B MoE in 3 GiB

"LOSSLESS" SPECULATIVE DECODING IS LOSSLESS IN FP32

An independent reproduction of a self-drafting speculative decoder ran 1,190 prompts from 12 domains and found that under BF16, the speculative output exactly matches plain decoding for only 45% of the authors' checkpoint's generations and 43% of an independently trained model's. Under FP32 it matches on every prompt, and downstream scores show no systematic degradation [2609.15504](#). The practical reading: if you diff speculative against plain output at bf16 or fp16 as a correctness test, divergence is expected and is not a bug. The caveat is that one architecture was tested; whether EAGLE-style or DFlash drafters behave the same is not established.

The other audit is for anyone running a model with chunked KV compression: compressing fixed windows into fewer entries gives every token a new coordinate, its phase relative to the window boundary, and long-context retrieval in large open-weight models can differ by up to 40 points across phases — "high average accuracy can coexist with systematic positional failures" [2609.36322](#). In a similar spirit, a methods paper argues that current evaluations of KV-cache reuse for RAG systematically inflate how well it works [2609.31415](#).

Edge0 is the most-starred brief-relevant repository of the window, at over 4,000: a per-layer predictor guesses the next layer's routing one token ahead, and that guess is the routing, so the experts staged from SSD are exactly the experts used. On a single 24 GiB machine it serves a 35B MoE at 20 tok/s inside 3 GiB of peak active memory, "within a few points" of its fp16 teacher [2609.18063](#). It changes the routing and needs a recovery adapter, so it is not a drop-in for an existing GGUF — read it beside §1's expert cache, which solves the same problem without retraining. Two drafting results close the section: **DLoop** keeps drafting while the drafter is confident instead of verifying after every block, reported at 5–41% more wall-clock speedup across EAGLE-3, DFlash, DSpark and MTP modules [2610.07659](#) — the direct sequel to issue 04's stranded speed-up; and **AgSpec** drafts from the session and workspace in coding-agent pipelines, reported at up to 4.37× at batch size 1 [2610.01108](#).

A new model class, two price points, and fewer knobs

Technical news from the large labs, every date read off the lab's own post or changelog entry. Five weeks held a new class of model from a startup, three new Claude models, three GPT-6 variants, a Grok, a Mistral flagship,

a near-frontier coding model built on an open base, and an announced Gemini 4. Grouped by what changes for someone building on them, not by company.

JEV, and a model class that formed in three weeks

The most consequential launch of the window did not come from a large lab. On 15 September **TypeSafe** came out of stealth with **Jev**, the first of what it calls "System One" models: "unstructured state in, typed probabilistic decisions out". You send a state and a schema of typed questions — a choice, a yes or no, a score — and get back a value for each with "calibrated probabilities and confidence scores", in one pass and without generating text. TypeSafe prices input at \$0.042 per million tokens with output "FREE (too cheap to meter)", and reports 70–500 ms end-to-end. It is hosted only, in waitlisted early access, and the launch post gives speed and cost multiples but no numeric accuracy benchmarks [post](#). A note on the date: TypeSafe's post currently carries a 9 October timestamp, but coverage dated 16 September reports the company exiting stealth "on Tuesday", the 15th [VKTR](#), 16 Sep, and the open models below cite Jev from 18 September on. Both dates fall in this window.

What happened next is the story. Three days after the launch, **Laya** shipped: a 421M ModernBERT-large encoder with a decision head that exposes "the same `POST /v1/systemone` request and response shape as TypeSafe Jev" [model card](#). Five days after, **OpenJev**, a 27B open-weight model that states it is not affiliated with TypeSafe but follows the hosted Jev API, reports itself "within 2 points of the hosted Jev API" on 10,000 text questions, 84.0% against 85.4%, under a non-commercial licence [model card](#). By the end of September there were JEV-style adapters on Qwen backbones (Kev, lev), a 144M multilingual one (Julia-1), a 0.9B one whose card says "clients written for Jev or Kev work unchanged" [model card](#), and AutoTrust's **JEV-27B-VL**, a vision decision model on Qwen3.8-27B that its card reports ranked first of 20 on the vision board of a community-run Jev Decision Index [model card](#). Cloudflare, Perplexity, InternLM and Liquid AI followed with their own (§4).

The runtimes converged on Jev's wire format rather than inventing one. llama.cpp's `/v1/systemone` endpoint, merged on 2 October, "follows the System One format introduced with TypeSafe's Jev model, so existing clients only need a new base URL" [post](#), 2 Oct (§1). OpenAI's **Decisions API**, in beta from 6 October on GPT-6 Luna, takes text and images and returns typed answers, claimed at ten times faster than its Responses API [changelog](#), 6 Oct. A startup's hosted model, more than a dozen open decision models, an open runtime and a large lab's API, all inside 21 days, is the clearest sign of the window that this is a component people will build with: classification, routing, moderation and agent next-action choices as one forward pass with a probability per option, instead of generating text and parsing it. Two cautions. A calibrated, schema-valid answer can still be confidently wrong; type safety removes parsing failures, not errors. And the evidence is young — the accuracy comparisons are the open models' own, mostly against Jev, on boards their authors or the community run.

The frontier converges on two prices

The mid tier now has a going rate. **Claude Sonnet 5.5** launched on 28 September at \$2 per million input tokens and \$10 per million output [post](#), 28 Sep, the same price as OpenAI's **GPT-6 Sol** on 22 September and **GPT-6.1 Sol** on the 29th, both with a 1.05M-token context [changelog](#), 22 & 29 Sep, and the same as the introductory price Google announced for Gemini 4 Argon (below). **Grok 4.7**, on 21 September, undercuts on output at \$2 and \$6 with a 500K context [post](#), 21 Sep. The small tier converged too: **Claude Haiku 5.5** on 7 October and **GPT-6 Luna** on 22 September are both \$0.10 in and \$0.50 out, Haiku for prompts up to 100K tokens [post](#), 7 Oct. And **Claude Opus 5.5**, on 22 September, came in *cheaper* than its predecessor at \$4 and \$20, with a 1M context by default [post](#), 22 Sep. On the consumer side, OpenAI began rolling GPT-6 out to everyone in ChatGPT on 7 October, with an "Intelligent UI" that answers with visuals and interactive elements [post](#), 7 Oct.

Vendor benchmarks for all of these are on their posts, and most are on each vendor's own evaluation suite; one detail is worth repeating because it is unusual: Anthropic reports Sonnet 5.5 *above* Opus 5.5 on Terminal-Bench 4.0, 70.6% against 66.4%. For anyone choosing a default model, the practical reading is that a \$2/\$10 model is now where the frontier is priced, and anything above it has to justify itself on your workload.

An open base, post-trained to near the frontier

Cognition's SWE-2, released on 10 September in Devin's desktop app and CLI, is post-trained from Kimi K3, an open-weight model. Cognition reports it within one point of Claude Fable 5.1 on FrontierCode 1.1 Main, 50.0% against 50.9%, at 64% lower cost, and ahead of it on DeepSWE 1.1, 73.0% against 67.4%. Its own table also shows the limit: 27.3% on Terminal-Bench 4 against Fable's 55.8% [post](#), 10 Sep. There is no API and no statement about weights. What matters for this publication is the recipe: a company without a frontier pretraining run took an open base close to the frontier on the benchmarks it targeted, and the spread across benchmarks says exactly how close. The post's date prints as "09.10.26", month first; it was discussed on 10 September.

Fewer knobs, again

Issue 04 noted that GPT-6 Astra would not accept a temperature. That is now the direction of travel across vendors, and every instance is a breaking change for someone. Opus 5.5 returns a 400 if you try to disable thinking; Sonnet 5.5 replaces "disabled" with a new `between_tools` mode; Haiku 5.5 rejects `temperature`, `top_p`, `top_k`, thinking budgets and assistant prefill outright [release notes](#); and GPT-6.1 Sol has no `none` reasoning effort [model page](#). If your code sets sampling parameters on every call, a model upgrade is now a code change. The stated reasons vary; the effect is the same: the hosted frontier is becoming a component whose decoding you do not control, which is one more argument for keeping a local model where you need that control.

The gated tier became the normal shape

Issue 04 named "a public model plus a gated sibling for security work" on two instances and declined to call it a trend. Five weeks later it is how the labs ship. Anthropic restructured its Cyber Verification Program into three tiers on 6 October and reports, on its own scenario benchmark, that Opus 5.5 was blocked on every task without enrolment [post](#), 6 Oct; it added a Life Sciences Verification Program on 17 September [post](#), 17 Sep, and its three 5.5 models route higher-risk cyber and biology requests to an older model unless the account is enrolled. Google's Gemini 4 Argon goes to vetted cyber defenders *first*, and to them without cyber guardrails [post](#), 30 Sep. OpenAI's life-sciences model is trusted-access only [changelog](#), 8 Sep, and Grok 4.7's red-team capabilities are invite-only. Two practical consequences: the same model ID can now answer differently depending on what programme your organisation is in, and Anthropic resumed billing on 24 September for some safeguard refusals that produce no output [release notes](#).

The labs ship agent runtimes

OpenAI's **Agents API** entered public beta on 10 September, with OpenAI handling session orchestration, compaction and recovery in hosted or bring-your-own sandboxes, and gained computer use on the 29th, the same day GPT-6.1 Sol shipped multi-agent delegation inside a single Responses request [changelog](#). Anthropic shipped compaction on demand (14 September), tools that can be added mid-conversation without invalidating the prompt cache (22 September) and dynamic multi-agent workflows for Managed Agents (9 October); the line worth noting for anyone worried about prompt injection is from 7 October, when Managed Agents' `web_fetch` started refusing URLs not already present in the session [release notes](#). OpenAI also put agents in front of consumers: **Dots**, launched on 29 September, are proactive assistants in ChatGPT that keep working across projects and tasks [post](#), 29 Sep. Z.ai open-sourced **ZCode**, its coding workbench — desktop app, browser interface and terminal agent, with the agent runtime in the repository — under Apache 2.0 on 20 September, and it passed 7,000 stars [repo](#). Google replaced its Antigravity agent on 17 September with a *changed built-in tool schema* — renamed tools and parameters, breaking for anyone parsing tool calls — and shut the old version down on 5 October [changelog](#), 17 Sep. The item most relevant to this publication: the Antigravity SDK now runs against local models, Gemma 4 on LiteRT or any OpenAI-compatible server, which includes `llama-server` [post](#), 23 Sep.

Announced, not shipped

Gemini 4 Argon was announced on 30 September with introductory pricing of \$2 and \$10 and a 1M output-token limit, but as a staged release: defenders first, then the paid API, then wider. At compile time it has no published model ID and no entry in the Gemini API changelog, so treat it as announced, not available [post](#), 30 Sep. **Mistral Large 4**, a MoE of about 1T total and 52B active parameters with a 1M context, entered public preview on 6 October, and Mistral promises open weights "by the end of the month", licence unstated [post](#), 6 Oct. If they ship, they go in a future releases table; until then they are a promise.

Research, and the rest

Three research items with wet-lab or official-grader validation, which is rarer than it should be. Anthropic ran about 950 Claude agents for 21 hours over a sequence database and found a reverse transcriptase with an accessory gene and a CRISPR-like repeat array, then expressed and characterised it; its function is unknown, and the post says so [post](#), 23 Sep. Google DeepMind published AlphaGenome Atlas, precomputed predictions for about nine billion single-letter variants of the human genome, free for non-commercial use [post](#), 8 Sep, and SynthID Bio, watermarks for protein sequences and structures that survived wet-lab testing without hurting binding [post](#), 30 Sep. OpenAI published new results on open problems in mathematics from an internal frontier model, with the Lean formalizations of the proofs released on GitHub — the formal proofs are the part anyone can check [post](#), 6 Oct. NVIDIA reports Nemotron systems at gold-medal level on IOI 2026, in an unofficial live run, and IMO 2026, graded by official graders [post](#), 7 Oct.

For practitioners, Hugging Face's most useful post of the window: Transformers can now run llama.cpp's Q4_K_M, Q5_K_M and Q6_K quants packed on Apple silicon through ggml's Metal kernels, for Qwen3.5-family models on the main branch only, with throughput shown "close to llama.cpp" in charts that are not like-for-like [post](#), 22 Sep. The GGUF formats are becoming a lingua franca beyond the runtime that defined them.

Meta published nothing on its AI blog for the third issue running; its newest post is still from 27 July. **Qwen** shipped only image models, and **Microsoft** no new Phi.

§4 · RELEASES

Weights that shipped

Open weights only; specifications and figures are as stated on the model cards, and benchmark numbers are the vendors'. Dates are the Hugging Face repository's creation date; where the vendor's own public date differs — repositories are often created privately first — both are given. The last column is the question this publication cares about: does it run on upstream llama.cpp today.

DATE	MODEL	WHY IT MATTERS
6 Sep	MiniCPM5-2B	Apache 2.0 , 2.5B dense, 128K context, shipped with a 324M DSpark draft model for speculative decoding model card . Created an hour after issue 04 closed, so it belongs here. The card claims 4B-class results on code, maths and tool use. Runs upstream , with official GGUFs for both the model and the drafter GGUF.
10 Sep	DeepSeek-V4.1-Flash	MIT, 763B stored with 8B active in prefill and 16B in decode , 1M context, image and text model card . The release of the window. A causal encoder-decoder design plus cross-layer KV reuse and FP4 KV caching cut the global KV cache to 890 bytes per token, about a quarter of V4-Flash's 2609.19969, and its "Engram" conditional-memory tables are sparse lookups that can live on SSD — one community build streams them from disk on a single 128 GB Mac GGUF. Not in upstream llama.cpp : the support PR has been open since release day #28696.
12 / 21 Sep	AliceAI-Foundation-80B-A3B-Base	Apache 2.0 , 81B total with 3B active , 262K context, a from-scratch Yandex base model with a KDA linear-attention hybrid layout model card . Repository 12 September, announcement 21 September post . Base only, so it is for fine-tuners; strongest, by the vendor's account, on Russian factual QA. No llama.cpp PR yet.
16 Sep	Xing4.0-29B-A4B	Apache 2.0 , 29B with 4B active , 256K context, MLA plus MTP, and by the card's account trained entirely on Ascend NPUs model card . Reported 75.0 on SWE-bench Verified; the vendor's IQ4_NL GGUF is about 18 GB, one consumer GPU. Upstream support is an open PR #29012; the vendor ships its own llama.cpp build in the meantime.
16 Sep	Ternary-Bonsai-2-27B	Apache 2.0 , Qwen3.8-27B with ternary weights at 1.72 bits per weight end to end , 5.95 GB against about 54 GB at FP16 model card . The compression story of the window, reported at 98.2% of FP16 on 14 thinking-mode benchmarks. Read §2 before trusting that: knowledge is exactly what the window's audits say low-bit models lose while averages hold. Needs Prism's llama.cpp fork ; the upstream PRs are open #29672.
17 / 22 Sep	FrogNano-4B	A 4.7B Microsoft coding agent on the Qwen3.5-4B hybrid architecture, trained with RL alone on about 1,500 synthetic software tasks and, per the card, on no stronger model's trajectories model card . The licence is ambiguous — the card's metadata says MIT, its table says Apache 2.0, and the repository has no licence file. Runs upstream via community GGUFs GGUF.

DATE	MODEL	WHY IT MATTERS
18 Sep – 5 Oct	JEV-style decision models	The open answer to TypeSafe's hosted Jev (§3): models that return a calibrated probability per option for typed questions instead of generating text, on Jev's <code>/v1/systemone</code> wire format. Laya (18 Sep, Apache 2.0, 421M, the most-liked of the wave) model card ; OpenJev (20 Sep, 27B, CC BY-NC, reported within 2 points of hosted Jev) model card ; Kev-9B, lev and Intern-Decision-4B as adapters or fine-tunes of Qwen models; and AutoTrust's JEV-27B-VL (30 Sep, Apache 2.0, Qwen3.8-27B plus a decision head, reads images) model card . Laya, Julia-1, lev, Kev and OpenJev run upstream from the first decision-model PR #29818.
20 Sep	Mellum2.1-12B-A2.5B	Apache 2.0 , 12B with 2.5B active , 128K context; JetBrains' coding model, with nearly all the new work in RL post-training on real repositories model card . 2.5B active is CPU-class decode for a coding agent. Runs upstream , with an official GGUF GGUF .
21 Sep	MiMo-V2.6 family	MIT. Xiaomi's omni MoEs: Flash at 309B with 15B active, Pro at 1.02T with 42B active, 1M context model card . The one most people can run is Distill-Qwen-9B , a 9.4B dense SFT of Qwen3.5-9B reported at 44.6 on SWE Pro against its base's 32.0 model card . Runs upstream for text #29257, with ggml-org GGUFs.
30 Sep / 1 Oct	Clef, Clef-Flash	Apache 2.0 , 27B and 9B post-trains of Qwen models that generate no text: you send a state and a schema of typed questions, and get back a calibrated probability for every option in one forward pass model card post . The JEV-style model from the largest company so far. Runs upstream on the new <code>/v1/systemone</code> endpoint (§1) GGUF .
2 / 3 Oct	Kolibri-1	Apache 2.0 , 78B with 3.5B active , 262K native context validated to 1M; Aleph Alpha's European open-weight MoE, trained from scratch on 20T tokens, about a quarter of them German model card post . Not upstream ; community GGUFs need a bundled patch, and no PR exists.
5 / 7 Oct	d1-3B, d1-omni-600M	Liquid AI's decision models at the edge, under the LFM Open License: the 600M omni variant takes text, images and up to 30 s of audio, and the vendor reports 8 ms per decision on an RTX 4090 model card post . Runs upstream #30114.
14 Sep / 6 Oct	EmbeddingGemma 2	Apache 2.0 , 740M, one 768-dimensional space for text, code, images, video and audio, with each modality's encoder loadable on its own model card post . The repository dates from September but the model launched on 6 October. The local RAG building block of the window, and supported upstream the same day #30054.

DATE	MODEL	WHY IT MATTERS
6–8 Oct	LightOnOCR-3	Apache 2.0 , 0.8B, 1B and 4B end-to-end OCR models with a grounding mode that returns labelled bounding boxes <code>model card post</code> . The 0.8B and 4B move to the Qwen3.5-VL architecture, so they should run on llama.cpp's multimodal path — but there is no official GGUF and we have not verified it.

Of fourteen rows, eight run on stock llama.cpp today — three because they reuse an architecture it already knew, five because support landed after release, the decision models among them. The three most interesting releases — DeepSeek-V4.1-Flash, Ternary Bonsai 2 and Kolibri-1 — all need a fork, a patch or an open PR. Left out deliberately: Qwen3.8, GLM-5.3 and Kimi-K3 are still trending but shipped before the window; large post-trains of existing open bases, such as Nex-N2.5 and Intern-S2, are real releases but outside a brief about what runs locally; and Qwen-Image-2.1 is outside the LLM remit.

§5 · TAKEAWAYS

What I would actually do

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

ADOPT

Update llama.cpp — after reading the three breaking changes, and today if you expose `rpc-server`. The port is now 9931 `#30159`, the old loading flags are gone `#28334` and saved sessions from older builds will not load `#28498`. None of that is a reason to stay behind: the RPC use-after-free fix is in v0.5.0 and later `#24292`, and Qwen3.8-Flash-Next was broken before 1 October `#29751`. Pass `--port 8080` explicitly, swap `--mmap` for `--load-mode`, and regenerate saved sessions.

ADOPT

On an M1–M4 Mac, turn speculative decoding back on. If you dismissed DFlash2 on Apple silicon this summer, you were measuring a kernel problem that is now fixed: 30.2 to 110.0 t/s on code on an M3 Ultra, with serial decode unchanged `#29869`. On any hardware, if you sample at temperature, try `--spec-draft-sampling probabilistic` with a simple draft or MTP `#27694` — and leave it off near temperature zero, where it can cost a few percent.

ADOPT

Stop accepting perplexity as the check for a quant you deploy. The window's audits agree that perplexity and average accuracy pass while definitions [2610.04403](#), factual capacity [2609.32902](#) and tool use [2609.32042](#) fail — and that the verdict can flip with the prompt and the scorer [2610.07781](#). Concretely: below 2B parameters, do not go under Q4_K_M; prefer Q4_K to Q4_0 at the same size [2609.07664](#); and test the behaviour you rely on, with more than one prompt, on the exact artifact you ship. Four issues running, 4-bit-or-lower passing the standard guard and failing a behavioural one is no longer a coincidence; treat it as the default.

WATCH

The MoE expert cache, until it reaches a tag. If you run a MoE bigger than your VRAM on one consumer card for interactive use, `--moe-cache-mib` on a nightly is worth an afternoon: $1.57\text{--}2.20\times$ reported decode [#29887](#). If you batch long prompts across two cards, the author's own table shows prompt processing at $0.73\text{--}0.91\times$ [#30112](#), so measure your mix before keeping it.

WATCH

JEV-style decision models, as a replacement for "generate, then parse". TypeSafe's hosted Jev [post](#), OpenAI's Decisions API [changelog](#), [6 Oct](#), an open runtime endpoint on the same wire format [#29818](#) and more than a dozen open models arrived within three weeks. For routing, moderation and agent next-action choices, one forward pass returning a probability per option is cheaper and easier to threshold than parsing text, and the open side is already close: OpenJev reports 84.0% against hosted Jev's 85.4% on its own test [model card](#), though under a non-commercial licence. If you want to try it locally, Laya at 421M under Apache 2.0 is the cheap place to start [model card](#). The format is weeks old and a "Decision 2.0" PR is already open in llama.cpp, so build a prototype, not a dependency — and keep the thresholds honest, since a calibrated answer can still be wrong.

WATCH

DeepSeek-V4.1-Flash and Ternary Bonsai 2, until upstream. The first is the architecture of the window and the second is a 27B in 6 GB, but both need a fork or an open PR ([#28696](#), [#29672](#)). For Bonsai specifically, read its "98% of FP16" beside §2: an average across benchmarks is exactly the metric the window's audits say hides what ternary models lose. If you fine-tune a ternary model, check your export: casting to bf16 before ternarising can silently revert the fine-tune [2610.07853](#).

WATCH

If you distil to add knowledge, OPD with reverse KL is the wrong tool. It sharpens skills the student has [2610.09639](#) [2609.35210](#), and the cheap versions match the expensive ones [2609.24432](#). For facts, the one controlled result says forward KL restores transfer; until that replicates, put new knowledge in with SFT or retrieval and use OPD to tune behaviour. Check your EOS tokens while you are there [2609.20511](#).

Looped models. The share of papers rose sixfold across the window, and the serving tricks are real — shared KV across loops 2610.02383, self-speculation 2609.23033 — but there are few open looped checkpoints worth running, llama.cpp support has only just started with HRM-Text #27625, and the one controlled study of what they trade away says knowledge 2609.36636.

NOTE

What five weeks did and did not establish. Two trends survive normalising for volume: on-policy distillation's share of cs.CL and cs.LG submissions roughly tripled and settled at about twice where it started, and looped models rose about sixfold from a small base and were still rising. The raw counts from 27 September to 4 October are inflated across every topic — total submissions that week were about twice the window's other weeks — and should not be read as growth. The thesis rests unevenly: the OPD leg has four independent groups behind it, the quantization leg several audits mostly at 14B and below, and the looped-model leg a single paper. Not established: whether "OPD adds no knowledge" holds at scale with forward KL, whether the GGUF lexical damage replicates beyond English rare words, and whether "lossless" speculative decoding fails exact-match at bf16 for drafters other than the one tested. A month is long enough to see convergence; it is not long enough to see replication.

Sources

LLAMA.CPP §1

Merged pull requests from midday 6 September to 10 October were enumerated from the GitHub API (718 merged, builds b10822 – b11540), and every PR cited below was fetched individually to confirm it merged; the open PRs and issues are marked as such.

[llama.cpp v0.4.1](#) release notes

[llama.cpp v0.5.0](#) release notes

[llama.cpp v0.6.0](#) release notes

[#29887](#) llama : add a GPU cache for MoE experts kept in host memory

[#30112](#) llama: support MoE cache over multiple GPUs

[Niko1221/Strata](#)

[#29869](#) metal : few-row MMA mat-mul and batched copies for speculative decoding

[#30065](#) metal : few-row MMA mat-mul for the remaining src0 types

[#27694](#) Make the drafter probabilistic and the target verify by rejection sampling for simple draft and MTP

[#29924](#) spec : fix n-gram drafts rejected at temp > 0 after truncation

[#27773](#) add GLM-5.3-Flash (GLM5-Next) support

[#29928](#) feat: add GLM5Next MTP, optimize

[#29751](#) llama: fix qwen4exp

[#29761](#) Qwen4Exp: add MTP

[#28068](#) models: fix GDN normalization from `max` to `rsqrt`

[#29818](#) llama, server: add /v1/systemone API (models: layla, julia-1, lev, openjev, kev)

[#29831](#) model: add support for clef decision model (text-only)

[#30044](#) models: support pplx-decider

[#30110](#) model : add LiquidAI/d1-3B decision model

[#27000](#) llama: add Maple 20B-A1B ternary MoE architecture (CPU)

[#27625](#) model : add support for HrmTextForCausalLM (DFM Mimir 1B)

[#28105](#) vulkan: support sparse Flash Attention

[#28796](#) [SYCL] support sparse FA

[#28770](#) CUDA: enable sparse fa for qwen4

[#29639](#) vulkan: sparse flash attention for quantized K/V

[#29298](#) CUDA: enable sparse-fa for dsv4 prefill (again)

[#29570](#) metal : add tensor API flash attention kernel for F16 KV

[#29435](#) CUDA: prefer whole-tile FlashAttention scheduling for efficient two-stage kernels

[#28102](#) CUDA/HIP: Flash Attention tuning (gfx1201)

[#27851](#) ggml-cpu: tiled mul_mat for k-quants

[#27952](#) vulkan: int8 coopmat1 matmul implementation for AMD RDNA3 and RDNA4

[#29809](#) sycl : accelerate MXFP4 MoE with arithmetic decoding and weight reordering

[#29749](#) Avoid a second full-size copy of each tensor with direct-io

[#29007](#) ci: publish snapdragon builds in release workflow

[#30159](#) server: use port 9931 by default

[#28334](#) args: remove mmap/mlock/dio flags from arg parser

[#28498](#) kv-cache: fix restoring mismatched KV cache rotation by saving exact rotation metadata

[#26004](#) server : preserve context checkpoints across slot save/restore

[#28690](#) server: Add support for binding to multiple addresses

[#28876](#) server : allow RANK pooling batch splitting for causal LLM rerankers (ie. Qwen3 and Qwen3-VL)

[#29583](#) ui : add the models manager

[#24292](#) rpc : fix UAF in graph_recompute leading to remote code execution

[#30135](#) ggml: fix OOB write in ggml_acc / ggml_set with negative offset

[#29598](#) ggml : speed up model loading

[#29384](#) Ggml integer overflow

[#28849](#) Change max context length for auto-fitting with unified KV

[#29437](#) Revert "Change max context length for auto-fitting with unified KV"

[#29182](#) vulkan: MOE aware mul_mat_id tile selection

[#29936](#) vulkan: revert mul_mat_id tile selection PR [#29182](#)

[#29770](#) metal : use bf16 math for mxfp4 mul-mat

[#30042](#) llama: remove the gather path of the glm5-next sparse attention

[#27428](#) draft-mtp roughly halves prompt processing on multi-GPU layer split (open issue)

[#28827](#) gemma4 thinking emits progressively long trailing garbage (open issue)

PAPERS §2

[2610.09639](#) On-Policy Distillation Teaches New Skills but Not New Knowledge

[2609.35210](#) Understanding On-Policy Distillation: A Mechanistic Interpretability Perspective via Sparse Crosscoders

[2610.03185](#) Gains and Collapse in On-Policy Distillation:A Reinforcement Learning Perspective

[2610.11247](#) Why On-Policy Distillation Sometimes Fails: Vanishing Learning Signals

[2609.33791](#) Do We Really Need KL Divergence for On-Policy Distillation of Large Language Models?

[2609.35259](#) On-Policy or Off-Policy Learning? A Systematic Study of Distillation Dynamics

[2609.32722](#) Scaling Properties of Same-Family On-Policy Distillation

[2609.35347](#) Beyond Teacher Assignment: Domain-Normalized Multi-Teacher On-Policy Distillation

[2609.24432](#) 1% of Tokens Can Be Enough: On Gradient Estimation in On-Policy Distillation

[2610.08448](#) Rethinking Cross-Tokenizer On-Policy Distillation: From Alignment Coverage to Supervision Reliability

[2609.34386](#) Look Before You Select: Rethinking Vocabulary Sparsification in On-Policy Distillation

[2609.20511](#) When EOS Tokens Disagree: Understanding Length Inflation in On-Policy Distillation

[2609.38995](#) When Clipping Reverses Correction: Failure Dynamics of Pointwise Forward-KL On-Policy Self-Distillation

[2609.36484](#) The Teacher Is a Direction, Not a Destination: Extrapolating RL-Induced Representation Residuals in On-Policy Distillation

[2610.04403](#) Saying, Not Knowing: Aggressively GGUF-Quantized Small Language Models Still Write Rare Words They Can No Longer Define

[2609.32902](#) Linger and Lose: Knowledge Collapse in Low-Bit Language Models

[2609.07664](#) Accuracy is Not Enough: A Divergence-Based Approach to Evaluate Fidelity Loss in Quantized LLMs

[2609.32042](#) Quantization Thresholds Replicate, Failure Modes Do Not: A Three-Model Study of Agentic Tool Use in Polish from 8-bit to 2-bit

[2610.07781](#) Quantization Effects on Tool-Failure Recovery Vary Across Prompts and Evaluation Designs

[2610.11226](#) When Lower Reconstruction Loss Hurts: Distributionally Robust Refinement for Low-Bit LLM Quantization

[2609.34866](#) From Attention Sensitivity to Layer Role: Revisiting Mixed-Precision Quantization of Transformers

[2610.07853](#) Lost in the bf16 Cast: Exporting Ternary Language Models Can Revert Most Low-Learning-Rate Code Changes

[2609.26333](#) Disaggregated Quantization: Specializing LLM Prefill and Decode

[2609.39223](#) QATFactory: A Versatile, Deployment-Aligned Framework for Quantization-aware Training and Distillation of LLMs

[2609.35465](#) Tetra: Serving Leech-Lattice Quantized LLMs at 2.7 Bits per Parameter

[2609.38169](#) STEPQuant: When and Where Errors Matter in Delta-Rule Recurrent State Quantization

[2609.40316](#) Scaling Laws for Looped Mixture of Experts

[2610.02383](#) The Surprising Effectiveness of Shared Memory in Looped Transformers

[2609.23033](#) WaveFront Decoding: Parallelized Self-Speculative Decoding for Looped Language Models

[2609.39277](#) A Tilted Bowl Is Not a Slippery Slope: Compressing Looped Models

[2609.36636](#) What Makes Recurrence Effective in Looped Language Models?

[2609.15504](#) Evaluating Losslessness in Speculative Decoding Under Finite-Precision Inference

[2609.36322](#) Periodic Weak Spots: Phase Sensitivity from Chunked KV-Cache Compression

[2609.31415](#) Evaluating the accuracy of KV cache reuse techniques

[2609.18063](#) The Other Half of the Memory Wall: Serving 35B MoEs from SSD with Trained Routing Prediction

[2610.07659](#) DLoop: Looped Speculative Decoding

[2610.01108](#) AgSpec: Pushing the Limits of Retrieval-Based Speculative Decoding in Coding Agent Pipelines

[TypeSafe — Introducing System One Models & Jev \(launched 15 Sep; post now timestamped 9 Oct\)](#)

[VKTR — TypeSafe launch coverage \(16 Sep\), cited for the launch date](#)

[convaiinnovations/laya](#)

[openjev/openjev](#)

[IFM/K2-Type-0.9B](#)

[autotrust/JEV-27B-VL](#)

[ggml-org — decision models in llama.cpp \(2 Oct\)](#)

[OpenAI API changelog \(8 Sep – 8 Oct entries\)](#)

[Anthropic — Claude Sonnet 5.5 \(28 Sep\)](#)

[xAI — Grok 4.7 \(21 Sep\)](#)

[Anthropic — Claude Haiku 5.5 \(7 Oct\)](#)

[Anthropic — Claude Opus 5.5 \(22 Sep\)](#)

[OpenAI — GPT-6 and Intelligent UI for everyone \(7 Oct\)](#)

[Cognition — SWE-2 \(10 Sep\)](#)

[Claude API release notes](#)

[OpenAI — GPT-6.1 Sol model page](#)

[Anthropic — Cyber Verification Program \(6 Oct\)](#)

[Anthropic — Life Sciences Verification Program \(17 Sep\)](#)

[Google — Gemini 4 Argon \(30 Sep\)](#)

[OpenAI — Introducing Dots \(29 Sep\)](#)

[zai-org/ZCode](#)

[Gemini API changelog](#)

[Google — local models in the Antigravity SDK \(23 Sep\)](#)

[Mistral — Mistral Large 4 \(6 Oct\)](#)

[Anthropic — agent-discovered enzyme system \(23 Sep\)](#)

[Google DeepMind — AlphaGenome Atlas \(8 Sep\)](#)

[Google DeepMind — SynthID Bio \(30 Sep\)](#)

[OpenAI — Sharing AI progress in mathematics \(6 Oct\)](#)

[NVIDIA — Nemotron at IOI and IMO 2026 \(7 Oct\)](#)

[Hugging Face — Transformers runs llama.cpp quants \(22 Sep\)](#)

RELEASES §4

[openbmb/MiniCPM5-2B](#)

[openbmb/MiniCPM5-2B-GGUF](#)

[deepseek-ai/DeepSeek-V4.1-Flash](#)

[2609.19969 DeepSeek-V4.1-Flash: Pushing the Limits of KV Cache Compression](#)

[antirez/deepseek-v4.1-flash-gguf](#)

[#28696 model : add DeepSeek V4.1 \(open\)](#)

[yandex/AliceAI-Foundation-80B-A3B-Base](#)

[habr.com](#)

[XingChen-AGI/Xing4.0-29B-A4B](#)

[#29012 Add Support for Xing4.0 \(open\)](#)

[prism-ml/Ternary-Bonsai-2-27B-gguf](#)

[#29672 ggml: add PTQ1_0, ternary at group 128 \(open\)](#)

[microsoft/FrogNano-4B-2609](#)

[bartowski/FrogNano-4B-2609-GGUF](#)

[JetBrains/Mellum2.1-12B-A2.5B-Thinking](#)

[JetBrains/Mellum2.1-12B-A2.5B-Thinking-GGUF](#)

[XiaomiMiMo/MiMo-V2.6-Flash-RL](#)

[XiaomiMiMo/MiMo-V2.6-Distill-Qwen-9B](#)

[#29257 convert: add MiMo-V2.6 support](#)

[Cloudflare/clef](#)

blog.cloudflare.com

ggml-org/Clef-GGUF

Aleph-Alpha/Kolibri-1

aleph-alpha.com

LiquidAI/d1-omni-600M

www.liquid.ai

[#30114](#) model : add LiquidAI/d1-omni-600M decision model

google/embeddinggemma-2

blog.google

[#30054](#) model: support embeddinggemma2 (text+vision+audio)

lightonai/LightOnOCR-3-4B

huggingface.co

Compiled 10 October 2026 from the llama.cpp release feed and GitHub API, the arXiv API filtered to this window and swept week by week, Hugging Face's curated daily papers, the Hugging Face models API and model cards, 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 04's compile stopped, midday Sunday 6 September, and closes at this one's on Saturday 10 October; anything dated outside it is labelled inline as context. Terms are defined on the [glossary page](#). Corrected later on 10 October to name TypeSafe's Jev and the JEV-style models, which the first version covered only as generic decision models, and to add Gemini 4 Argon to the items at the top; and again after a coverage check against Hacker News, Hugging Face and GitHub, to add six in-window items it surfaced.

