

La señal cara resulta, una y otra vez, *opcional*.

Esta semana tres literaturas distintas hicieron el mismo experimento sobre sí mismas y obtuvieron la misma respuesta. El KV-cache eviction descubrió que no calcular ningún score iguala al mejor de los scores. La on-policy distillation descubrió que la supervisión del teacher es ruido que el student ignora, y que un solo ejemplo de entrenamiento recupera casi toda la ganancia del dataset entero. Mientras tanto el runtime publicó el tipo de resultado contrario: la sparse flash attention llegó a llama.cpp, y solo se gana el sueldo en contextos largos. Tres modelos frontera se lanzaron en tres días, dos de ellos con un hermano restringido.

COMPILADO 6 SEP 2026

5 SECCIONES

~65 CITAS PRIMARIAS

VENTANA 6½ DÍAS

DE UN VISTAZO

01

Un evictor de KV cache que no calcula ningún score iguala al mejor de los que sí lo calculan. Random Attention conserva el prompt y expulsa de forma uniformemente aleatoria dentro de cada cabeza; en cuatro modelos y seis tareas de razonamiento iguala al evictor previo más fuerte mientras sirve un **32–43% más de throughput** en vLLM. Lo interesante es la explicación: el prompt es la parte frágil de la caché, y la traza de razonamiento se protege sola por redundancia — el modelo repite lo que necesita, y cada cabeza guarda su propia copia 2609.03430.

02

En la on-policy distillation el teacher es en gran medida opcional, y el dataset también. La supervisión del teacher sobre los rollouts del student tiene mucho ruido — más cuanto mayor es el teacher — y el student converge igual con el ruido dentro o fuera; una única ventaja negativa fija iguala a las que proporciona el teacher [2608.31046](#). Aparte, el OPD de un solo disparo sobre *una sola query* recupera casi todo el OPD con datos completos, alcanzando el 71,5% de los estados que visita el entrenamiento completo; dieciséis queries llegan al 98,9% y lo igualan [2609.04172](#).

03

La sparse flash attention llegó a llama.cpp para DeepSeek-V4 y GLM, y los números del propio autor muestran que solo sirve en contexto largo. En un DGX Spark con un quant de 2 bits de DSV4, la aceleración reportada en prefill es $1,00\times$ a poca profundidad, $1,19\times$ a 32K, $1,54\times$ a 128K y $2,20\times$ a 1M tokens [#27970](#). Metal llegó dos días después [#28098](#), y la v0.4.0 se etiquetó el 4 de septiembre [#28386](#).

04

Tres lanzamientos frontera en tres días, dos con un hermano restringido para trabajo de seguridad. Claude Fable 5.1 y Mythos 5.1 el 1 de septiembre — Mythos es el mismo modelo con menos salvaguardas, limitado a programas de acceso de confianza [post](#) —, Gemini 3.8 Flash y 3.8 Flash Cyber el 2 de septiembre, con Cyber restringido a defensores de confianza [post](#), y GPT-6 Astra el 3 de septiembre [changelog](#).

CONTENIDO

§1 llama.cpp — sparse attention y la v0.4.0

§2 Los papers — la auditoría se amplía

§3 Desde los laboratorios — tres lanzamientos, tres días

§4 Pesos que se publicaron

§5 Qué haría yo

Esta ventana va del final de la compilación del número 03 a la de este, y cierra el mediodía del domingo 6 de septiembre — un contenedor más corto que una semana nominal, declarado con honestidad en lugar de redondeado. Se añadieron dos términos al glosario

.

Sparse attention y la v0.4.0

114 pull requests fusionados entre el 31 de agosto y el mediodía del 6 de septiembre, terminando en la build `b10821`, con la **v0.4.0** etiquetada el día 4. Agrupados por lo que cambian para quien de verdad ejecuta esto.

Llega la sparse flash attention, y solo paga en profundidad

El cambio estructural de la semana es la sparse flash attention para DeepSeek-V4 y GLM #27970, construida sobre una propuesta anterior pero reducida a algo que el runtime puede sostener: en lugar de modelar el esquema de sparsity de cada arquitectura, se apoya en una pista de la API sobre el número máximo de entradas KV vivas por token. El autor apunta que otros esquemas — y nombra la QSA de Qwen3.8-Flash-Next — podrán sumarse después por el mismo mecanismo. El soporte en Metal llegó el 3 de septiembre #28098.

Las mediciones reportadas son inusualmente honestas sobre cuándo esto no hace nada. En un DGX Spark con un quant de 2 bits de DeepSeek-V4 (IQ2_XXS), contra la build inmediatamente anterior:

PROFUNDIDAD	PREFILL (PP2048)	DECODE (TG32)
sin profundidad	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×

Reportado por el autor en una sola máquina, así que las magnitudes son afirmaciones — pero lo importante es la *forma*, y es la forma que la sparse attention debería tener. A las profundidades a las que trabaja casi todo el mundo, esto es gratis y no hace nada. Pasada la barrera de los 100K tokens es la diferencia entre

usable e inusable. Si tu carga es de longitud de chat, actualizar por esto no te cambia la vida; si le estás dando a un modelo un repositorio entero, te reduce la espera casi a la mitad.

Merece leerse al lado el paper de diseño de Qwen3.8-Flash-Next, publicado el 31 de agosto, que describe exactamente el esquema que llama.cpp se prepara para aceptar. El mezclado de tokens es un híbrido por capas de Gated DeltaNet y atención global con una capa de atención completa de cada cuatro, y en el continued pretraining esas capas se sustituyen por Qwen Sparse Attention, que puntúa el contexto con granularidad de microbloque mediante un indexador ligero comprimido 2608.30320.

DFlash recupera su viaje de ida y vuelta

El número 03 cubrió la fusión de DFlash2. Esta semana la vía especulativa se hizo más rápida de forma poco vistosa: el encoder del drafter es un solo `fc` más norm, pero ejecutarlo como un `llama_encode` aparte forzaba un viaje de ida y vuelta dispositivo-host de su salida antes de que el decode de inyección pudiera volver a subirla, más una segunda construcción de grafo por ronda. Plegar el encoder dentro de la rama de embeddings del decoder elimina las dos cosas #27310. Así es la segunda semana de una funcionalidad cuando se la toman en serio.

v0.4.0

Etiquetada el 4 de septiembre #28386. Su contenido principal es el soporte inicial de Qwen3.8-Flash-Next, NVIDIA Nemotron-3-Puzzle-75B-A9B #25444, lectura de tensores bajo demanda, límites de contexto por slot en el servidor, opciones de entrada de vídeo y una subida de ggml a 0.23.0 que trae el trabajo de sparse attention y RDMA. Ojo: una release agrega más de una ventana — varios elementos de sus notas, incluidos la carga perezosa y el trabajo por slot, se fusionaron antes del 31 de agosto y ya se cubrieron aquí en su momento.

Arquitecturas nuevas dentro de la ventana: la preview de Hy4 de Tencent #28127, tres días después de aparecer los pesos (el número 03 señaló que esa card era parca en especificaciones), y el soporte multimodal de DeepSeek-V4-Flash-Vision-Exp #28133 con un arreglo de entrada de visión detrás #28154 — dos días después de publicarse ese modelo (§4).

Apple silicon: acaba la campaña y le sale una puerta de CI

La campaña de tuning de flash attention por chip que llenó media §1 del número 03 terminó el trabajo: M1 #28078, M1 Ultra #28088, M2 Pro #28122, M2 Max #28015 y sus casos restantes #28458, M3 #28236, M3 Max #28373 y el A18 Pro #28152. Aparte, la API de tensores de Metal 4.0 ya está habilitada en M5 y A19 en adelante #27461.

Pero el detalle que merece copiarse es de proceso, no de rendimiento. El número 03 reportó una fuga de memoria en Metal por autoreleasepools ausentes. Esta semana esa clase de bug se convirtió en una comprobación de CI #27884, excluyendo explícitamente las fugas conocidas de dispositivo gráfico internas de macOS que el proyecto no puede arreglar, y con el autor reconociendo que la regex es imperfecta y que tener el test es mejor que no tenerlo. En la misma tanda se arreglaron más fugas (#27883, #28399). Convertir el bug de la semana pasada en la puerta de esta es la medida de calidad más barata que puede tomar un proyecto.

Un patrón más entre backends: el top-k radix aterrizó en Metal #28073, ROCm #27466 y Vulkan #28032, este último explícitamente para $k \geq 1024$ en Qwen3.8-Flash-Next. El sampling dejó de ser barato cuando crecieron el vocabulario y el número de expertos de un modelo, y tres backends lo arreglaron la misma semana.

Memoria, y el servidor

Sigue el hilo que recorre todos los números recientes: llama-quantize ya transmite por trozos de filas en vez de dejar sus hilos sin trabajo #27830, y la RAM ya no hace pico durante la carga del modelo #27483. En el servidor se aceptan URLs data: para input_video e input_audio #27735, preserve_reasoning viene activado por defecto y registra su estado efectivo #28174, y llegó --log-jsonl para quien envíe sus logs a algún sitio estructurado #28437.

ARISTAS

Una implementación SYCL del producto de Kronecker FWHT se fusionó el 1 de septiembre a las 15:47 UTC #28016 y se revirtió **diecisiete minutos después** por romper test-backend-ops #28184, y se restauró el 5 de septiembre con el fallo de CI arreglado #28254. El número 02 contó la misma historia con un revert-de-revert de Q2_K en SYCL, y la lección no ha cambiado: si compilas SYCL desde fuente, fija tu build después de la restauración, y nunca te fíes de un resumen para saber si algo llegó a entrar.

La auditoría se amplía

El número 03 contó cómo la literatura de KV eviction auditaba sus propios benchmarks. Esta semana la auditoría llegó a su conclusión, y otras dos subdisciplinas hicieron por su cuenta el mismo experimento sobre sí

mismas: quita el componente caro, compáralo con una línea base trivial y mira si alguien lo nota.

El score no aporta casi nada

Random Attention es el resultado hacia el que caminaban los dos números anteriores. Todos los métodos de compresión de KV comparten un paradigma: puntuar cada token cacheado por alguna estimación de cuánto importará después y quedarse con los mejores. Este paper conserva el prompt, expulsa *uniformemente al azar* dentro de cada cabeza de atención y no calcula ningún score. En cuatro modelos y seis tareas de razonamiento iguala al evictor previo más fuerte, sirviendo un 32–43% más de throughput que él en un despliegue con vLLM — porque puntuar costaba algo y no compraba nada 2609.03430.

Lo que lo hace útil, y no sólo provocador, es la explicación controlada. Primero, **el prompt es la parte frágil de la caché**, y casi toda la diferencia entre los selectores publicados es simplemente si su señal acertó a protegerlo. Segundo, la traza de razonamiento se protege sola por redundancia a dos niveles: en el texto, porque el modelo repite lo que sigue necesitando mientras trabaja, y entre cabezas, porque cada una guarda su propia copia. Una vez el prompt está a salvo, un sorteo aleatorio retiene copias suficientes de lo que el modelo necesita, y no hace falta ningún score para encontrarlas.

Leído contra el número 03, el arco se cierra. Trust the Mass mostró que quedarse con los mayores pesos ya era casi óptimo y que los márgenes publicados venían de la contabilidad del almacenamiento 2608.25230. TwinKV mostró que la magnitud de la atención no correlacionaba con la contribución causal de un token 2608.27128. Esta semana dice en voz alta lo evidente que quedaba: deja de puntuar.

Salvo cuando la señal está en otra parte

VestigeKV es el contrapeso necesario, y no es una contradicción. Ataca el caso al que se enfrenta de verdad una caché de larga vida: la compresión tiene que ocurrir *antes* de que existan las queries que la leerán, de modo que seleccionar por atención observada no tiene nada que observar. En un modelo NoPE MLA no se degrada con elegancia: H2O y SnapKV se hunden a 0,00–0,33 de recuperación en needle. La solución es expulsar según una señal independiente de la query que la caché ya lleva: la rama desacoplada de 64 dimensiones, un vestigio de RoPE que el entrenamiento NoPE reconvierte en canal de saliencia. Leyendo el 11% de cada fila, la recuperación se mantiene en 1,00 con compresión 8× y en 0,92 con 32×, y la configuración estándar de recall aguanta 128× en 1,00 2609.03949.

EL PANEL DE ADVERTENCIA – "EL SCORE NO IMPORTA" ES UNA AFIRMACIÓN SOBRE TU CACHÉ, NO SOBRE LAS CACHÉS

Estos dos resultados parecen opuestos y no lo son, que es lo más útil que se puede sacar de la semana. Random Attention expulsa de una *traza de razonamiento*, redundante por construcción, en modelos donde la selección nunca hacía gran cosa ^{2609.03430}. VestigeKV expulsa de una *caché documental de larga vida* en un modelo NoPE MLA, donde la señal habitual no es sólo débil sino inexistente, y el paper es explícito en que el mismo operador sobre un MLA con RoPE se hunde a 0,08 ^{2609.03949}.

Así que la regla transferible no es "puntuar no sirve". Es que **el valor de una señal de selección depende de qué hay en la caché y de cómo codifica la posición el modelo** — y casi nadie declara en cuál de esos regímenes cae su benchmark. Si expulsas en producción, el experimento barato es ahora el informativo: ejecuta eviction aleatorio con el prompt fijado y mira si tu método lo supera con tu tráfico.

On-policy distillation: el teacher también era opcional

El número 02 encontró cuatro papers de acuerdo en que la señal por token del OPD es demasiado local. El 03 encontró al campo nombrando el colapso como el fallo dominante. Esta semana fue más allá y preguntó si el teacher hace algo siquiera.

Does On-Policy Distillation Really Distill? mide la supervisión del teacher durante el entrenamiento y encuentra un ruido sustancial, cuya prevalencia *aumenta con la escala del teacher*. Resulta que el student es insensible a ese ruido y converge a un rendimiento comparable tanto si la supervisión ruidosa se conserva como si se elimina. Al examinar qué impulsa realmente las ganancias, el aprendizaje se concentra en tokens de baja log-probabilidad, y una única ventaja negativa fija iguala a las que da el teacher — así que el OPD funciona en gran medida suprimiendo tokens de baja probabilidad, cosa que no requiere teacher. Su sustituto sin supervisión, OPSA, usa ventajas negativas adaptadas a la entropía y reporta una ganancia de 35,41 puntos en Avg@32 sobre AIME24 respecto al Qwen3-1.7B base, un 263% de mejora relativa ^{2608.31046}.

Rethinking OPD II quita el otro insumo caro: los datos. Entrenando sobre *una sola query*, el OPD de un disparo sigue mejorando durante cientos de pasos y recupera casi toda la ganancia del OPD con datos completos, en varios dominios y familias de modelos. El mecanismo es la cobertura de estados — la fracción de estados que visita el entrenamiento completo a los que llegan los rollouts de un conjunto de queries. Una sola query ya alcanza el 71,5%, casi todo dentro de los primeros 100 pasos; dieciséis queries semánticamente distintas llegan al 98,9% e igualan el entrenamiento completo, también en el caso multi-teacher. Y sin embargo la alineación se ralentiza al mismo ritmo en ambos casos. El veredicto de los autores es la frase citable: *el OPD está sobrealimentado de datos y desnutrido de algoritmo* ^{2609.04172}.

Dos papers tiran en sentido contrario, y son los que dan algo que hacer el lunes. **TGOPD** verifica la fiabilidad del teacher por prompt antes de admitir supervisión densa, y enruta los prompts poco fiables a GRPO anclado en verificador; supera al OPD básico en los seis escenarios de dominio único y, de paso, pone a

trabajar al teacher ocioso, subiendo la utilización de GPU del nodo teacher del 9,8% al 78,9% en la ejecución medida 2609.02998. Y **Sequential Beats Joint** reporta que el simple OPD-y-luego-RL supera al OPD puro, al RLVR puro y a todas las combinaciones conjuntas probadas: el OPD amplía la cobertura de soluciones que el teacher respalda y el RL afila dentro de ese soporte, mientras que optimizar ambos a la vez hace que interfieran. La puntuación de validación del OPD es la señal de cuándo cambiar, y el OPD es mejor arranque en frío para RL que el SFT 2609.04108.

El speculative decoding se dota de una teoría de entrenamiento

Directamente relevante para lo que llama.cpp fusionó la semana pasada. **Verification-Aware Training** señala un desajuste que nadie había arreglado: la verificación es secuencial y descarta todo a partir del primer rechazo, pero el entrenamiento del draft usa imitación a nivel de token con un peso fijo por posición que no refleja ninguna de las dos cosas. VAT añade una cabeza de verificación ligera y reancla el decaimiento del peso al primer punto de rechazo de cada muestra. Aplicado a EAGLE-3 y DFlash sobre Qwen3-4B, Qwen3-8B y Llama-3.1-8B, reporta hasta un 11,4% más de longitud media de aceptación y hasta un 8,7% de aceleración de reloj, cambiando sólo el objetivo de entrenamiento 2608.30135.

El resultado acompañante es un diagnóstico gratis para quien ejecute los block drafters que llama.cpp ya soporta. Como DFlash y DFlare rellenan un bloque entero en una pasada, el target acepta a menudo el bloque *completo*, de modo que el drafter agota su horizonte entrenado antes de que la verificación llegue a fallar — aceleración que existe pero nunca se recoge, y que los autores llaman **stranded speed-up**. Una longitud media comprometida lo esconde; el histograma de aceptación lo expone de inmediato como un pico en el bin del techo. Ensanchar el bloque en inferencia no lo recupera, porque pasado su tamaño de entrenamiento la atención bidireccional del drafter se desplaza y erosiona la verificación del principio del bloque; reentrenar sobre un bloque más largo con un currículo corto sí. Expandir DFlash y DFlare del bloque 16 al 24 sobre Qwen3-8B y Qwen3-4B sube la longitud comprometida por prompt en una mediana de +0,8 tokens, y llega a 1,37 si antes se hace fine-tuning de continuación 2608.30427. Si ejecutas un block drafter, dibuja el histograma antes de comprar nada.

Cuantización: gasta el siguiente bit globalmente

The Structure of Quantization Damage es el paper más directamente accionable de la semana. Usando intervención causal de precisión mixta como verdad de referencia — subir cada capa a 8 bits por turnos y medir la precisión que recupera — sobre nueve modelos de pesos abiertos en cuatro familias de arquitectura, pone a prueba las tres hipótesis intuitivas sobre dónde vive el daño de la cuantización: circuitos de tarea, dónde computa el modelo y estadísticas de los pesos. *Ninguna predice qué capas se benefician de recuperar precisión*. La recuperación es difusa: en ocho de nueve modelos, recuperar el 75% de la brecha exige aproximadamente la mitad de las capas. Y con un presupuesto de precisión igualado, gastarlo globalmente en granularidad más fina supera a reparar localmente las capas más recuperables — entre 21 y 52 puntos, en todos los modelos compatibles con group-128 probados, incluido el único cuyo daño sí estaba concentrado 2609.01587.

Dos resultados a cada lado. **Minima** pone NVFP4 W4A4 en las 496 capas lineales de un modelo híbrido de 27B, incluidos los bloques Gated DeltaNet que las cuantizaciones de la comunidad venían dejando a 8 o 16 bits con la intuición de que los errores se acumulan en una recurrencia. Iguala a BF16 dentro del ruido de semilla (media de 5 tareas $-0,52$) con 17,5 GiB y un prefill un 14–19% más rápido, y el estudio de mecanismo desmonta la intuición: las proyecciones de puerta supuestamente frágiles son las *menos* sensibles, y la recurrencia de regla delta mantiene el ruido inyectado en una meseta plana a lo largo de 32K tokens porque cada escritura sobrescribe el estado en la dirección de la clave actual 2609.04098.

Y un aviso para quien cachee KV cuantizada en un sistema RAG. **Faithfulness Is Not Free** separa precisión de fundamentación, y encuentra INT8 casi sin pérdida en ambas mientras INT4 daña la fidelidad de una forma que la precisión no puede ver: entre las respuestas que siguen siendo factualmente *correctas*, más del 90% de los cambios de fidelidad son negativos, y el daño crece con recuperación ruidosa y con más fragmentos recuperados 2608.30996. Van tres números seguidos en los que los 4 bits pasan el control estándar y fallan uno de comportamiento.

Dos más que conviene conocer, ambos sobre servicio híbrido. **Tail-Replay** observa que una capa de linear attention es una compresión con pérdida de su prefijo, así que el estado recurrente de un prefijo emparejado puede reconstruirse reproduciendo sólo un sufijo reciente corto — lo que elimina la necesidad de guardar checkpoints de estado y desbloquea la reutilización de prefijo a nivel de token en cualquier frontera 2608.30310. Y **DASC** deriva horizontes de retención por cabeza desde los pesos del modelo para comprimir esos checkpoints donde sí se conservan 2608.30386. Ambos apuntan de lleno a los modelos GDN y KDA que el mundo de pesos abiertos está publicando ahora.

§3 · LABORATORIOS

Tres lanzamientos, tres días

Noticias técnicas de los grandes laboratorios, citadas a sus propios posts. Ha sido la semana más cargada que ha visto esta sección: tres lanzamientos frontera en tres días consecutivos, y un patrón en cómo se empaquetaron dos de ellos.

El hermano restringido

Anthropic publicó **Claude Fable 5.1** y **Claude Mythos 5.1** el 1 de septiembre. Fable cuesta 10 dólares por millón de tokens de entrada y 50 por millón de salida, con lecturas de caché a 0,25 por millón; entre los benchmarks reportados hay un 55,8% en Terminal-Bench 4.0, un 52,6% en Terminal-Bench-Science 0.1 y un

77,9% parcial en OSWorld 2.0. Mythos se describe como el mismo modelo con menos salvaguardas, disponible sólo mediante programas de acceso de confianza y limitado por ahora a organizaciones de Estados Unidos [post](#), 1 sep. El día anterior publicó una entrada sobre alineamiento y seguridad [post](#), 31 ago.

Google publicó **Gemini 3.8 Flash** y **3.8 Flash Cyber** el 2 de septiembre, a un precio introductorio de 0,75 / 3,75 dólares por millón de tokens hasta el 31 de diciembre. La variante Cyber está restringida a defensores de confianza mediante el nuevo Fairwind Program, y se reporta por encima del 70% en un benchmark interno de descubrimiento de vulnerabilidades que abarca 20 lenguajes de programación, con un 47,2% de pass@1 en CWE-Bench para parcheo [post](#), 2 sep.

Dos de los tres lanzamientos de la semana salieron, por tanto, como un modelo público más un hermano restringido orientado a trabajo de seguridad — uno quitando salvaguardas, otro añadiendo capacidad de seguridad ofensiva tras un programa de acreditación. Dos casos son un patrón que merece nombrarse y todavía no una tendencia sobre la que teorizar; lo que hay que vigilar es si el nivel restringido se convierte en la forma normal en que la capacidad frontera llega a los equipos de seguridad.

GPT-6 Astra, y una API pensada para trabajo largo

OpenAI publicó **GPT-6 Astra** el 3 de septiembre a través de las APIs Responses y Chat Completions, recomendándolo para razonamiento, código, uso del ordenador, investigación y creación de documentos. Las restricciones son tan informativas como las afirmaciones: no admite el nivel de esfuerzo de razonamiento `none`, no acepta valores propios de `temperature` ni `top_p` ni log-probabilidades, y su tool calling exige la Responses API [changelog](#), 3 sep. Un modelo al que no puedes ponerle temperatura es otro tipo de componente sobre el que construir.

El mismo día, y más interesante para quien construya agentes, llegaron tres controles de la Responses API para trabajo de larga duración: **tool calling asíncrono**, para que el modelo siga trabajando mientras corre tu función; **dirección a mitad de turno**, enviando instrucciones adicionales por WebSockets mientras la respuesta está en marcha; y cambiar el esfuerzo de razonamiento a mitad de conversación preservando los prompts cacheados. Dos cambios menores que merecen una línea: desde el 2 de septiembre, el tráfico que sube demasiado rápido devuelve un `429` con código `slow_down` mientras que la sobrecarga real del modelo devuelve un `503`, lo que cambia la lógica de reintento de cualquiera con un bucle de backoff; y desde el 1 de septiembre `api.openai.com` es accesible por IPv6.

El resto de la semana

Google también publicó el 1 de septiembre la comprensión agéntica de vídeo para Gemini 3.7 Flash, 3.6 Flash y 3.5 Flash-Lite: pon el parámetro de procesamiento en `agentic` y el modelo ejecuta un bucle interno para cargar sólo la parte relevante de un vídeo, con hasta un 88% menos de tokens, un 66% menos de coste y un 7% más de precisión reportados, al precio de tokens habitual [post](#), 1 sep. Y **WeatherNext 3** llegó el 3 de septiembre con predicciones horarias a 5 km de resolución en superficie y hasta un 60% mejor CRPS en

precipitación — pero, a diferencia de los modelos WeatherNext que el número 03 destacó por publicar pesos, a este se accede por BigQuery, Earth Engine y Cloud Storage en lugar de descargarlo [post](#), 3 sep.

El blog de **Hugging Face** publicó tres piezas para practicantes el 3 de septiembre: *NeoMME*, un encoder multilingüe y multimodal nativo [post](#); el fine-tuning de un modelo de 350M para salidas estructuradas en 100 pasos de GRPO [post](#); y *Funes*, un sistema de memoria local que indexa las trazas de sesión de agentes de código en datasets Lance consultables, con embeddings y reranking en local, de modo que los agentes puedan recordar razonamientos pasados sin enviarlos a ninguna parte [post](#).

Meta no publicó nada otra vez; sus entradas más recientes siguen siendo la línea Muse de julio. Van dos números seguidos, y registrarlo es justo el sentido de una sección que informa sobre los laboratorios y no para ellos.

§ 4 · PUBLICACIONES

Pesos que se publicaron

Solo pesos abiertos; las especificaciones y cifras son las declaradas en las model cards. Las descargas y los likes son los del momento de compilar.

FECHA	MODELO	POR QUÉ IMPORTA
31 ago	DeepSeek-V4-Flash-Vision-Exp	MIT, ~305B parámetros, multimodal (image-text-to-text), publicado con pesos FP8 model card . 209.000 descargas y 701 likes en su primera semana: el lanzamiento claro de la ventana. El soporte multimodal en llama.cpp se fusionó dos días después (§1), y es una de las dos arquitecturas a las que apunta la nueva vía de sparse flash attention — por una vez, el runtime y la publicación convergen en la misma semana.
1 sep	K2-Horizon-MoVA-36B-A4B	Apache 2.0 , 36B totales con 4B activos , un mixture-of-experts MoVA de IFM publicado con su dataset de preentrenamiento referenciado en la card model card . La clase de tamaño que importa para trabajo local: 4B activos son un coste de decode de portátil, y la licencia es la permisiva.
2 sep	vdn-minimax-h3	Un fine-tune de texto a vídeo de MiniMax-H3 con licencia propia model card . Fuera del alcance de esta publicación evaluarlo; queda registrado porque es una publicación de pesos abiertos con atención real encima.

Esta semana aparecieron dos informes técnicos de modelos cuyos pesos no: Instella-MoE y A.X K2 publicaron sus documentos dentro de la ventana, pero los checkpoints están en el Hub desde julio — fuera de ventana, y listados aquí para que las fechas no se confundan con un lanzamiento.

§5 · CONCLUSIONES

Qué haría yo

Ordenado por madurez, no por lo interesante que sea cada cosa.

ADOPTAR

Si sirves DeepSeek-V4 o GLM por encima de unos 100K tokens, actualiza para tener sparse flash attention. La curva reportada #27970 es toda la decisión: nada a longitud de chat, 1,54× de prefill a 128K, 2,20× a 1M. Actualiza igualmente si estás en Metal o en Apple silicon en general, porque la campaña de tuning por chip y la vía de Metal cayeron en la misma ventana — pero no esperes que el trabajo de sparse aparezca en un benchmark de contexto corto, ni concluyas que está roto cuando no aparezca.

ADOPTAR

Si cacheas KV cuantizada en una pipeline RAG, mide fidelidad, no sólo precisión — y prefiere INT8. 2608.30996 reporta INT8 casi sin pérdida en ambas métricas, e INT4 degradando la fundamentación de forma invisible: más del 90% de los cambios de fidelidad son negativos *entre respuestas que siguieron siendo correctas*. Tu panel de precisión no puede ver esto por construcción. Van ya tres números seguidos encontrando que los 4 bits pasan el control estándar y fallan uno de comportamiento; en algún momento eso deja de ser casualidad y pasa a ser el supuesto por defecto.

VIGILAR

Antes de adoptar cualquier método de KV eviction, compáralo con eviction aleatorio con el prompt fijado. Es un experimento de dos líneas y esta semana se convirtió en la única línea base honesta 2609.03430. Si tu método no lo supera con tu tráfico, estás pagando cómputo por un score que no hace nada. Aunque la advertencia de la §2 importa: en una caché documental de larga vida, o en un modelo NoPE, el cuadro se invierte 2609.03949 — así que ejecuta la línea base sobre *tu* carga en vez de adoptar cualquiera de las dos conclusiones al por mayor.

VIGILAR

Gasta tu siguiente bit de cuantización en granularidad más fina, no en proteger capas "críticas". Nueve modelos, cuatro familias, tres teorías intuitivas sobre dónde vive el daño, ninguna de las cuales predice nada — y la granularidad global ganando entre 21 y 52 puntos con presupuesto igualado 2609.01587. Si mantienes una receta de precisión mixta con capas protegidas elegidas a mano, este es el paper que dice que esa elección probablemente nunca se ganó el sueldo.

VIGILAR

Si ejecutas un block drafter, dibuja el histograma de aceptación. DFlash2 se fusionó en llama.cpp la semana pasada, y el resultado acompañante de esta semana muestra que una longitud media comprometida puede esconder aceleración que ya pagaste pero nunca recoges 2608.30427. El histograma es gratis, y un pico en el bin del techo te dice que el bloque se queda corto antes de que gastes nada en entrenar uno más ancho.

SOLO LABORATORIO

Los resultados de "el teacher es opcional" en OPD. Ambos son creíbles, ambos tienen una semana y ambos son un solo paper (2608.31046, 2609.04172). Lo que yo me llevaría ahora es el par de hallazgos más baratos y mejor respaldados que tienen al lado: ejecutar **OPD y luego RL** en vez de fusionarlos, usando la puntuación de validación del OPD para decidir cuándo cambiar 2609.04108, y condicionar la supervisión densa a la fiabilidad del teacher por prompt en lugar de confiar en ella siempre 2609.02998. Los dos son cambios de secuencia y no de algoritmo, lo que los hace baratos de probar y baratos de deshacer.

NOTA

Tres campos, tres resultados de "la parte cara era opcional" y un método compartido. Esa es la forma de la semana y conviene tratarla con cuidado. Son subdisciplinas distintas que no se coordinaron, lo que hace la convergencia más interesante que la de cuatro papers sobre un mismo tema del número 02 — pero cada resultado sigue siendo un solo paper, ninguno se ha reproducido de forma independiente, y "el componente no aportaba nada *en los benchmarks que ejecutamos*" es una afirmación más estrecha de lo que suena. Lo que sí se transfiere es el método, no las conclusiones: quita el componente caro, compáralo con la línea base más trivial que puedas construir, y publica el resultado incluso cuando deje en ridículo a la técnica. Es barato, está al alcance de cualquiera y, con la evidencia de esta semana, sigue funcionando.

Fuentes

LLAMA.CPP §1

[llama.cpp releases](#) — [v0.4.0](#) (4 sep) y builds hasta [b10821](#) (6 sep).

Los pull requests fusionados del 31 de agosto al 6 de septiembre se enumeraron desde la API de GitHub (114 fusionados), y cada PR citado abajo se consultó individualmente para confirmar su fusión.

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

[#27310](#) fusión del encoder de DFlash en el decode de inyección de KV

[#28386](#) subida de versión a 0.4.0 · [#25444](#) Nemotron-3-Puzzle-75B-A9B · [#28127](#) preview de Tencent Hy4 · [#28133](#) DeepSeek-V4-Flash-Vision-Exp · [#28154](#) entrada de visión de DeepSeek-V4

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

[#27884](#) comprobación de CI de autoreleasepools ausentes · [#27883](#) y [#28399](#) arreglos de fugas

Top-k radix: [#28073](#) Metal · [#27466](#) ROCm · [#28032](#) Vulkan para $k \geq 1024$

[#27830](#) streaming por trozos de filas en quantize · [#27483](#) sin pico de RAM al cargar

Servidor: [#27735](#) URLs data: para vídeo y audio · [#28174](#) preserve_reasoning por defecto · [#28437](#) --log-jsonl

Saga FWHT en SYCL: [#28016](#) fusionado · [#28184](#) revertido 17 minutos después · [#28254](#) restaurado

PAPERS §2

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

[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

Citados del número 03 por continuidad: [2608.25230](#) Trust the Mass · [2608.27128](#) TwinKV

LABORATORIOS §3

[Claude Fable 5.1 y Mythos 5.1](#) (1 sep) · [alineamiento y seguridad](#) (31 ago)

[Gemini 3.8 Flash y 3.8 Flash Cyber](#) (2 sep) · [comprensión agéntica de vídeo](#) (1 sep) · [WeatherNext 3](#) (3 sep)

[Changelog de la API de OpenAI](#) — GPT-6 Astra y los controles de trabajo largo de la Responses API (3 sep), 429 slow_down frente a 503 (2 sep), IPv6 (1 sep)

Blog de Hugging Face, 3 sep: [NeoMME](#) · [GRPO para salidas estructuradas](#) · [Funes](#)

PUBLICACIONES §4

[DeepSeek-V4-Flash-Vision-Exp](#) · [K2-Horizon-MoVA-36B-A4B](#) · [vdm-minimax-h3](#) — las especificaciones y cifras son las declaradas en las model cards.

[Hugging Face daily papers](#) — usado para descubrimiento a lo largo de la ventana; cada paper que aportó se leyó en su abstract de arXiv antes de citarse.

Compilado el 6 de septiembre de 2026 a partir del feed de releases y la API de GitHub de llama.cpp, la API de arXiv filtrada a esta ventana, los daily papers curados de Hugging Face y las salas de prensa y changelogs de los propios laboratorios. Las cifras citadas como aceleraciones, puntuaciones de benchmark o tasas de error son los números reportados por sus autores o fabricantes en sus propios benchmarks — trátalas como afirmaciones, no como mediciones, hasta que las reproduzcamos. La ventana se abre donde terminó la compilación del número 03 y cierra en la de este, el mediodía del domingo 6 de septiembre; todo lo fechado fuera de ella está etiquetado como contexto. Los términos se definen en el [glosario](#).



LLM Radar lo publica semanalmente [Montevive](#) · [Feed Atom](#) · ai@montevive.ai · [Privacidad](#)