A Measured Decomposition of the Trillion-Parameter Serving Step

The anatomy of speculative decoding at one trillion parameters,

and the serving record the measurement produces

Mohammad Alsufi* **Connor Boone***

and the Brainsless Research Lab AI Systems Research Group

*Equal lead contribution research@brainsless.com

Brainsless Research Lab | Technical Report **BRL-2026-11** | July 2026 | brainsless.com

DOI 10.5281/zenodo.21217452

Every number is measured on our own rented hardware unless marked (external), and rebuilds from a named artifact in the index.

Abstract

Single-stream serving throughput is usually reported as a black box: a tokens-per-second figure attached to an engine and a model, with no account of where the milliseconds go. We decompose it, term by term, on Kimi-K2.6 — a 1T-parameter mixture-of-experts model with 384 routed experts — at the public leaderboard's workload (10k-token inputs, 2,048-token outputs, n=16 per cell). The instruments are three: a no-speculation cell for the bare forward (6.6ms at 10k context); a weights-free proposer run at two draft widths under identical matcher configuration, whose differential cancels proposer overhead exactly and yields the first measured verify-cost curve on a fine-grained 1T MoE; and a cross-engine control at equal acceptance, which attributes step-time differences to software alone.

Three findings. First, verify$(k{+}1) = 6.6 + 0.397k$ ms: the expert-union verify cost is real and behaves as the routing arithmetic predicts; each additional draft token adds 0.397ms, and the full k=5 width adds 2.0ms to a 6.6ms forward — wide drafting is cheap physics. Second, the step at this scale is roughly 95% software: doubling the node from four B200s to eight moves the bare forward by 0.06ms, while the same head at the same acceptance (τ 4.82) costs 12.24ms per step on one engine and 10.10ms on another. Third, optimal draft depth is an engine-priced decision: per-position acceptance breaks super-geometrically past position 5, and the measured optimum moves from k=6 to k=7 as the per-position price falls from 1.47 to 0.61ms, as the step model predicts.

The decomposition is validated by what it buys. Moving the same model, same public draft head, and same protocol to the engine it pointed at set a lossless single-stream GPU serving record on the most widely benchmarked open 1T model — 505.9 tokens per second (depth 7) from four B200s, above every provider median Artificial Analysis' board publishes — and blind re-runs of the unmodified public release raised it to 511.6 (depth 6). Replication moved the

record up, not down. We also measure the cross-family drafter comparison the field publishes around: the published small-scale ordering reverses at 1T, and the decomposition prices the mechanism. Every table re-runs from the public release: one command, one rented node, about $15.

1 Introduction

A serving throughput number, by itself, explains nothing. When a deployment reports 400 tokens per second single-stream on a trillion-parameter model, the reader cannot tell whether the engine is near a physical floor or leaving half the number on the table, whether a better drafter or a better scheduler is the next lever, or whether more hardware would help at all. At the scale where these questions are most expensive to answer wrong, the public record offers throughput tables and no anatomy: deployments ship one speculative-decoding family each and publish neither step decompositions nor controlled cross-family comparisons above 120B.

This report measures the decoding step of Kimi-K2.6 term by term, and the accounting is worth more than the headline it produces. The bare forward, the marginal cost of verifying a draft token, the draft pass, and per-step software overhead are each measured with a separate instrument (§4); the instruments need no engine internals and transfer to any speculative stack (§10). Applied, the decomposition located 2.1ms of pure software in the serving step, converted it into a lossless single-stream GPU serving record — set at 505.9, raised to 511.6 by blind re-runs of the public release (§8) — and produced the acceptance anatomy that prices the next one (§7).

Table 1 maps every result to the section that measures it. Each is scoped as written there.

The record claims are scoped in §8: an openly benchmarked model, an open engine, measured public endpoints, and a configuration that reproduces from the release.

2 Protocol

The protocol is built to be comparable to the leaderboard's workload while remaining a controlled experiment. Per domain: 16 distinct novel document-pack prompts of ∼10,000 tokens each (9,900–10,400 measured), written for this campaign and SHA-256-pinned in the runner before any measurement; the runner fail-closes on manifest mismatch. Outputs are 2,048 tokens at temperature 0.6 with per-request seeds. No prefix-cache reuse between requests. Each request streams, and its decode rate is $(\text{ctok}-1)/(t_{\text{last}}-t_{\text{first}})$, which excludes prefill and time-to-first-token by construction. The cell statistic is the interpolated median over the 16 requests. Tokens are Kimi-native; measured o200k parity on stored raw outputs is 1.0035–1.0081, so no tokenizer normalization moves a headline digit.

Losslessness is the standard speculative-decoding guarantee: the target verifies every drafted token and the sampled distribution equals the target's own. We add a cross-engine check: the same head at equal depth measured τ 4.815 on vLLM and 4.825 on SGLang (tool traffic, 10k context), so acceptance is a property of the head and the traffic, not of either engine's acceptance rule.

Node discipline: rented nodes of the same type draw differently. Two same-configuration anchors on consecutive days read 403.2 and 390.6 tok/s (first-8, tool-nothink, k=5, vLLM), a spread consistent with the ±5–10% we observe across draws. Every session runs an anchor cell, every anchor is in Table 2, and the record session's anchor reads out its draw against the smoke reference in the artifact.

finding	measured result	§
First measured verify-cost curve on a 1T fine-grained MoE	verify(k+1) = 6.6 + 0.397k ms; 0.397 per added token, under 4% of a k=5 step per token	4
Weights-free drafter differential (the instrument behind the curve)	two draft widths, identical matcher, proposer overhead cancels; no engine instrumentation	4
Eight-GPU scaling null at 1T, batch 1	forward 6.60→6.54ms, step 12.24→12.32ms; step ∼95% software	4
Cross-engine controlled pair at equal acceptance	10.10ms (SGLang) vs 12.24ms (vLLM), τ 4.82; 2.1ms is pure engine software	4
Acceptance tail breaks super-geometrically; geometric forecast overpredicts	conditionals 0.92, 0.79, 0.72; forecast 418 at k=8, measured peak 407	4, 7
Optimal draft depth is engine-priced, both instantiations measured	per-position cost 0.61 (SGLang) vs 1.47 (vLLM) ms; optimum k=6→7	4
Per-position forecast validated at one step, refuted at three	predicted 397.9, measured 395.8 (0.5%); geometric fit fails by k=8	3, 4
Cross-family reversal at 1T, mechanism priced	block head higher τ in 4/4 domains, wins 1; block pass ∼8ms vs 0.73–1.07ms	6
Teacher-capture defect and the warm-start canary	silent one-layer teacher shift, invisible from scratch, loud under warm start	3
Specialist exchange rate and the proxy gap	mixed diet τ 2.74→2.62; text-only proxy +3 vs served +17%	7
Fastest measured GPU serving of Kimi-K2.6; first GPU reading above 500 tok/s on this model	505.9 (depth 7) set 2026-07-05, raised to 511.6 (depth 6) by blind re-runs of the release; 4×B200, lossless, n=16; 127.9 tok/s/GPU, 3× NVIDIA's disclosed 671B config	8, 3
Live replication of the leaderboard's measurement path	a production API read 118.4/149.2/168.9 tok/s against a 381.2 board median	9
The record replicates from the public release	three blind re-runs, three node draws, every n=16 cell in-band; best cell 511.6 (depth 6), first-eight to 538	9
Same-session trained-head pair on the record engine	Fovea-E 501.8 vs the public head's 474.2 (+5.8%), crossing 500 on a −4.3% node	3

Table 1. Summary of findings, each scoped in the section it is measured in.

3 Record sessions and results

The record configuration carries no trained component. The draft head is the public lightseekorg checkpoint, so 505.9 reproduces from off-the-shelf parts: a public head, a public engine, and the right depth. Reporting the record on the public head is deliberate: the result reproduces from open parts, and a reader can rebuild it. The trained-head lever is **Fovea**, a system we are building so a serving operator can train a draft head on their own traffic and serve their model faster with identical outputs. Fovea runs in our production stack today and delivers a speculative-decoding speedup of the class this report measures. Fovea-E is a warm-start of the public head fine-tuned on 2,373 hard tool and math conversations regenerated by the served target, and the July-6 replication session measured the pair on the record engine: at equal depth, same node, same prompts, Fovea-E read **501.8** against the public head's 474.2 (n=16 each, τ 5.24 against 5.16) — +5.8%, and above 500 on a node that drew 4.3% below reference (`brl11_repl_fovea.json`). Because production traffic is narrower than this open benchmark, acceptance has more headroom there than the general-

date	session (artifact)	engine	result (tool-nothink, 10k-in/2k-out, n=16)
07-04	baseline (`brl11_baseline`)	vLLM	EAGLE-3 k=3: 368.1 (τ 3.58)
07-04	record v1 (`brl11_record`)	vLLM	k=5 395.8 (τ 4.96) / Fovea-E 398.5 (τ 4.79); anchor −10% vs reference
07-05	stage A (`..._stage600a`)	vLLM	k-ladder: k5 403.2, k6 407.0, k7 395.4
07-05	verify diff. (`..._stage600e1`)	vLLM	T_1 6.60ms; ngram_gpu $k5/k8$ raw 17.14/18.33ms; slope 0.397
07-05	TP8 null (`..._stage600b`)	vLLM 8×B200	T_1 6.54ms; k5 step 12.32ms — scaling null
07-05	SGLang smoke (`..._stage600sg`)	SGLang v0.5.14	depth 5 486.9; **depth 6 498.5** (τ 5.269, 10.73ms)
07-05	**record (`..._stage600r`)**	SGLang v0.5.14	**505.9** at depth 7 (τ 5.507, 11.34ms; f8 520.6); depth 6 487.2; math 419.2
07-05	depth ladder (`..._stage600d`)	SGLang v0.5.14	depth 8 498.0 (τ 5.802, 12.09ms) — the 07-05 ladder peaks at depth 7
07-06	replication (`brl11_repl_*`)	SGLang v0.5.14	three re-runs from the public release: d6 511.6 / 484.4 / 474.2 (anchors +0.9/−5.7/−4.3%); d7 486.5 (f8 538.0); math 405–422
07-06	Fovea pair (`brl11_repl_fovea`)	SGLang v0.5.14	**Fovea-E 501.8** vs public head 474.2, same session, equal depth (τ 5.24, n=16, aa_len_ok)

Table 2. All record-relevant sessions in order, with node anchors; f8 is the first-8-prompt median. The record was set at 505.9 (depth 7) and raised to 511.6 (depth 6) by the blind re-runs; the 07-05 depth ladder peaks at depth 7 within its session.

purpose numbers show, and closing on the next ceiling is a training problem (§7), not an engine one. Fovea-E's checkpoints publish after license review. Its training pipeline also yielded a finding worth flagging for anyone distilling drafters against a served MoE: a one-line patch had shifted the distillation teacher one transformer layer early, invisible to from-scratch training and loud under a warm start. Verify the teacher's final-hidden capture against the serving engine before training (`fovea_e_result.json`).

The baseline session validated the forecasting method the campaign planned with: from k=3 per-position acceptance the live forecast priced k=5 at 397.9 tok/s, and the measured cell read 395.8 — 0.5% error at one extrapolation step. The July-5 chain ladder then bounded the method: on vLLM, chain k-tuning peaks at k=6 (407.0) and declines at k=7, against a geometric-tail forecast of 418 at k=8. Configuration tuning on that engine is exhausted around 407, forecasts earn one extrapolation step and not three, and the 91-point gap between 407 and the SGLang smoke reading at the same acceptance is the subject of §4.

4 The step decomposition

Single-stream throughput is τ divided by step time, per request; cell medians are interpolated over per-request rates, so a cell's rate is not exactly its mean τ over its mean step. Step time carries both the physics and the software, and we decomposed it term by term on the same node class, model, and 10k workload. Figure 1 is the object of the decomposition: one step, three stages, each with a measured cost.

The bare forward, T_1. With speculation off, the model decodes at 150.9 tok/s at 10k context, a 6.63ms forward (n=16, vLLM, `brl11_stage600a.json`); the same-day cross-check read 6.60ms. Short-context probes match (6.58ms), so there is no measurable context tax on the bare forward at 10k under MLA (multi-head latent attention). This is the floor any step pays once.

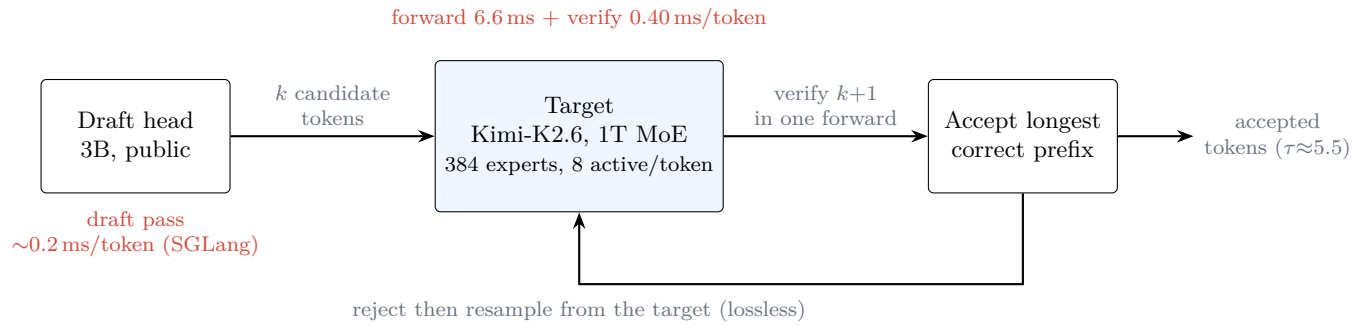

Figure 1. One speculative decoding step on the trillion-parameter target. A small public draft head proposes k tokens; the 1T target verifies all $k{+}1$ positions in a single forward and accepts the longest correct prefix, resampling rejects from its own distribution so the output is unchanged. The measured cost of each stage (Section 4) is annotated: the target forward is 6.6 ms, each verified position adds 0.40 ms, and the draft pass runs near its 0.2 ms bandwidth floor on the record engine.

The verify width cost. A speculative step verifies $k{+}1$ tokens instead of one. On a 384-expert MoE each verified token routes to its own 8 experts, the expert union grows with width, and the touched weights stream from HBM, so extra width has a real price. Measuring it is confounded by the proposer: any drafter adds its own overhead. We cancelled the confound with a weights-free proposer (the engine's `ngram_gpu` drafter, which proposes from a lookup with no neural forward) run at two widths under identical matcher configuration and a measured hit rate of 1.0. Raw steps: 17.14ms at $k{=}5$, 18.33ms at $k{=}8$ ($n{=}8$, 10k tool traffic). The proposer's overhead, ~8.5ms of uncaptured `torch.compile` launches per step, is identical in both arms and subtracts out. The differential gives the first measured verify-cost curve on a 1T fine-grained MoE:

$$\text{verify}(k{+}1 \text{ tokens}) = 6.6 + 0.397\,k \text{ ms} \qquad \textbf{0.397 ms per additional draft token}$$
(`brl11_stage600e1.json`)

Five draft tokens add 2.0ms to a 6.6ms forward. Worst-case byte arithmetic for the $k{=}5$ expert union (~44–48 of 384 experts per layer, 60 MoE layers, ~73GB/step) prices a 2.3ms bandwidth floor on this node's aggregate HBM; the measured marginal cost sits at or below that worst case. The tax is measurable and small. The instrument is also the only good use we found for `ngram_gpu` at batch 1: its own per-step overhead makes it useless as a serving drafter and perfect as a differential probe.

The draft pass. With T_1 and the verify slope fixed, the EAGLE head's cost falls out of the measured step by subtraction. On vLLM, step at $k{=}5$ is 12.24ms, so the five draft passes cost $(12.24 - 6.6 - 5 \times 0.40)/5 = 0.73$ms each; the within-session ladder slope (1.47 ms/position) minus the verify slope gives 1.07ms as the upper read, the spread reflecting convexity in k. The head is 3B parameters; its bandwidth floor at TP4 is ~0.2ms per pass. vLLM runs it at 3.5–5× that floor.

The cross-engine experiment. SGLang v0.5.14 (overlap scheduler, `tokenspeed_mla` attention, `flashinfer_trtllm` MoE) runs the identical head at identical acceptance (τ 4.825 vs 4.815) with a $k{=}5$ step of 10.10ms against vLLM's 12.24ms. The fitted SGLang step model is $\text{step}(k) = 7.07 + 0.61k$ ms against vLLM's 1.47 ms/position. The verify slope (0.40) is engine-independent physics; the difference is the draft pass and per-step host overhead, and SGLang runs the draft pass near its 0.2ms floor. The entire gap between 407 (vLLM, tuned out) and 498.5 (SGLang smoke) is software-overhead removal at unchanged acceptance and hardware.

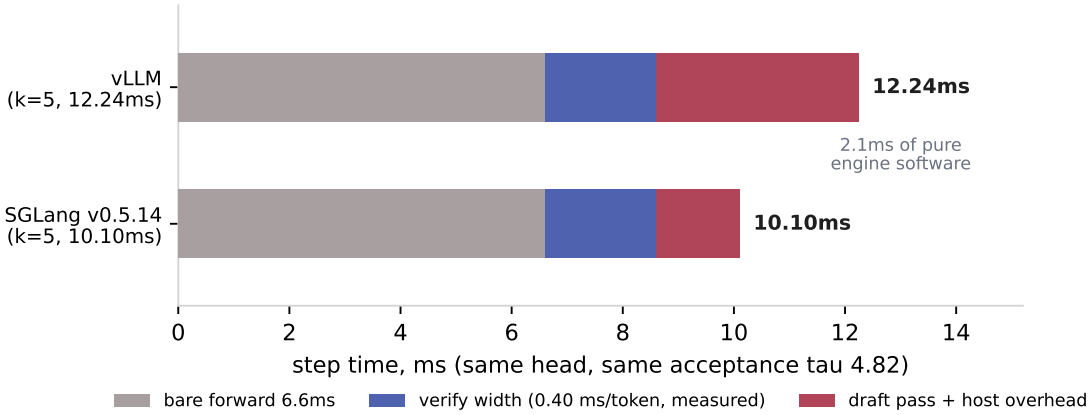

Figure 2. The k=5 step, decomposed, on two engines at identical acceptance (τ 4.82, same head). The bare forward (6.6ms) and the measured verify width (0.40 ms/token) are engine-independent; the remainder is draft pass plus host overhead, and the 2.1ms between the engines is pure software. Artifacts `brl11_stage600a/e1/sg.json`.

The TP8 null. If the step were bandwidth-bound, doubling GPUs would roughly halve it. Measured on 8×B200 TP8, vLLM, same protocol: T_1 = 6.54ms against 6.60ms on 4×; EAGLE k=5 step 12.32ms against 12.24ms (`brl11_stage600b.json`). Doubling the silicon bought 0.06ms. The per-GPU byte floor at TP8 is ∼0.3ms, so the measured step is ∼95% software overhead, and more hardware cannot remove software. This null is why the record runs on four GPUs.

The acceptance tail. Per-position conditional acceptance on tool traffic is flat near 0.92 through position 5, then decays super-geometrically: 0.92, 0.79 (position 6), 0.72 (position 7) (`brl11_stage600a.json`). This is why the vLLM ladder peaks at k=6, where each added position costs ∼1.5ms. On SGLang the added position costs ∼0.6–0.8ms, so the same tail supports one more level: the ladder reads 486.9 (depth 5), 487.2–498.5 (depth 6, two nodes), 505.9 (depth 7), 498.0 (depth 8), peaking at depth 7 (Fig. 3). Optimal depth is an engine-priced decision, and this campaign measured both instantiations.

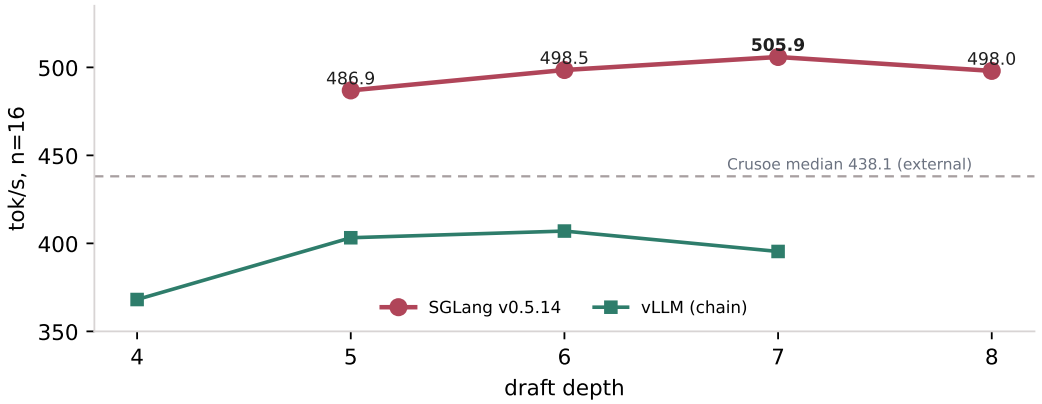

Figure 3. Draft-depth ladders, n=16. The cheaper per-position price on SGLang (0.61 ms/position vs vLLM's 1.47) moves the optimum from depth 6 to depth 7 and lifts the peak past 500. the leaderboard leader's same-day median marked for reference (Crusoe 438.1, external; §8). Artifacts `brl11_stage600a/sg/r/d.json`.

The decomposition composes into ceilings. On vLLM at measured acceptance a zero-cost drafter yields $\tau(k)/(6.6 + 0.4(k{+}1))$: 535 at k=5, 588 at k=7, a perfect-drafter ceiling near 590 on this engine class. On the SGLang step model, 600 tok/s needs $\tau \approx 7.2$, which the measured acceptance tail prices as a drafter-quality problem rather than a systems one. That work is active in the Fovea program, which trains the draft head on production traffic: on a distribution narrower than this open benchmark a specialized head clears more of the acceptance the ceiling needs, and it serves that traffic faster than a general head does here.

5 What the decomposition revises

Our earlier report in this series argued the expert-union verify cost was the governing term at 1T, from step times inferred as τ divided by throughput. The direct measurement revises the attribution. The verify tax exists and behaves as the routing arithmetic predicts, and it is 0.4 ms/token, under 4% of a k=5 step. The governing terms are the draft pass (0.73–1.07ms per position on the engine we first measured, ∼0.2ms on the engine that runs it well) and per-step host overhead. The earlier cross-family conclusion survives, and its mechanism sharpens: what punishes wide drafting on this model is the cost of producing and scheduling the draft, with the verify width a small additive term. The earlier number was inferred from throughput; this one is measured directly and supersedes it.

6 The cross-family comparison the field publishes around

Speculative decoding has two live drafter families: autoregressive heads (EAGLE-3 [Li et al., 2025]), drafting left to right, and block-diffusion heads (DFlash [Z-Lab, 2026]), drafting a block in parallel conditioned on target hidden states. Published head-to-heads stop at 120B (external: DFlash 4B–30B, NVIDIA gpt-oss-120B, DSpark 4B–14B [DeepSeek-AI, 2026]). At 1T, deployments ship one family each and publish no throughput. We measured both on the same target, quantization, engine build, prompts, and session (vLLM, released budgets, think mode, temperature 0.6, single stream, `brl10_k26_full_v2.json`):

domain (think mode)	no drafter	EAGLE-3 (k=3)	DFlash (k=8)
general chat	155 tok/s	**267** (τ 2.46)	173 (τ 2.86)
tool calls	209	297 (τ 2.76)	**311** (τ 3.82)
math	152	**410** (τ 3.43)	359 (τ 4.11)
code	147	**352** (τ 2.80)	268 (τ 3.55)

The finding reverses the published small-scale ordering: the block family guesses strictly better, with higher accepted length in all four domains, and wins only one (tool, by 5%), losing general, math, and code to the cheaper autoregressive head. §4 names the mechanism: DFlash's block draft pass measures ∼8ms on this 61-layer MLA target under this engine, against 0.73–1.07ms per EAGLE pass, and its small-scale 2.5× draft-cost advantage [LMSYS / SGLang team, 2026] does not transfer here. The cheap proposer outruns the better guesser. A stricter matrix agrees: on n=16 short novel prompts in no-think mode, EAGLE sweeps every cell (tool-nothink 337 against 258, general 243 against 174, code 290 against 233; `brl11_verify_tax_session.json`), and supersedes the preliminary 3-prompt τ 5.53 tool-nothink reading on the stock DFlash head flagged in our prior report.

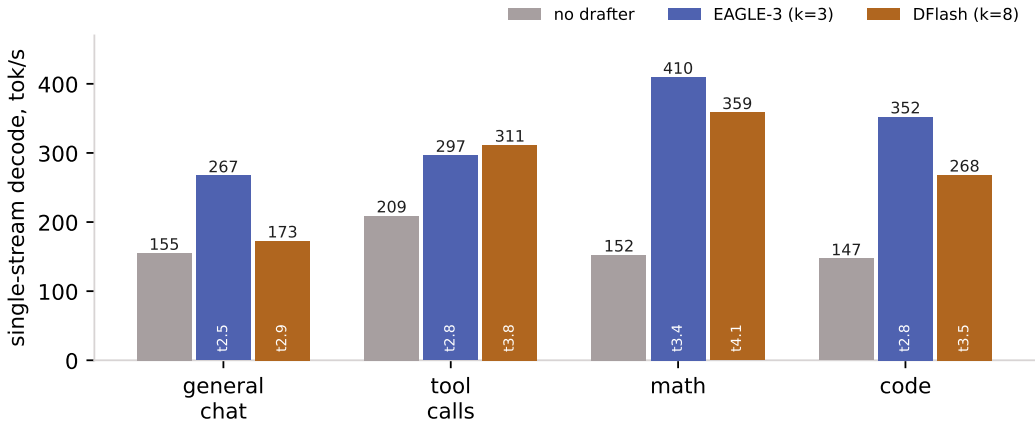

Figure 4. Cross-family single-stream throughput at 1T, think mode (`brl10_k26_full_v2.json`). The block head (DFlash) holds higher accepted length (τ) in all four domains and wins only tool; the cheaper autoregressive head (EAGLE-3) takes the other three.

The reversal holds under load: at 32 concurrent streams the node sustains 910 tok/s aggregate with EAGLE-3 against 845 undrafted and 499 with DFlash (`brl10_k26_full_v2.json`), so a drafter can be net-negative for fleet throughput while winning batch-1 latency.

7 Acceptance anatomy, and what training buys

Acceptance does not change with the engine; it is a property of the head and the traffic, and its structure at this scale is regular enough to plan against. On tool traffic the per-position conditionals are near-geometric at ∼0.92 through depth 5, then break (§4); on math they collapse to 0.665 by position 5; on chat every functioning public head sits in the τ 2.2–2.9 band at the 10k workload. Draft length is a workload decision priced by the step model: k=5–7 pays on tool and math at 10k and loses on short prompts (k=5 read 265–294 against 335 at k=3 on short-prompt health cells). KV-cache dtype is a second domain lever: native precision against fp8 moved math +34% in one session while costing tool 6% and code 19% (`fovea_gauntlet.json`). The record cell is the routing rule executed: our traffic is tool calls, so tool-nothink at depth 7, fp8 KV is the configuration we hold.

The thesis the Fovea program tests is that a draft head specialized to a fixed traffic distribution accepts more than a general head on that traffic, so once the engine is tuned out the lever is specialization, not a larger general head. What training buys is measured in three findings from that program. The specialist exchange rate: retraining the specialist head on a mixed diet (11,743 samples) traded most of the specialist's home-domain gain (τ 2.74 to 2.62, stock 2.61) for recovered code (2.97 to 3.16). A specialist head and a generalist head are different tools, and the exchange rate between them is a measured number at this scale (`fovea_verdict.json`, `fovea_gauntlet.json`). The proxy gap: a text-only 1.7B rehearsal predicted about +3 points for a course whose served result was +17% on the target domain. A hidden-state-conditioned drafter reads the target's internals at every layer, while a text-only proxy cannot see what the course teaches that channel, and data decisions for such drafters are settled by training and serving the conditioned drafter. These, with the warm-start canary of §3 and the acceptance tail, price the acceptance the next ceiling needs: τ 7.2 at the record's step model. That is a target the Fovea program pursues in active development, not a future proposal, and on a production distribution narrower than this open benchmark a

specialized head clears more of it, with the exchange rate warning against mixed diets and the proxy gap against cheap rehearsals.

8 The record, and the field it measures against

Kimi-K2.6 is served publicly by several providers, and Artificial Analysis measures their live endpoints at a fixed workload (∼10k-token inputs, ∼2k outputs). The field as of 2026-07-05 (external, Artificial Analysis, 2026), against our measurement at the same workload:

deployment	tok/s single-stream	hardware	measurement
ours	**505.9 → 511.6** (tool; math 419.2 → 422.1)	**4×B200, disclosed**	controlled benchmark, n=16
Crusoe	438.1 (peak 449)	undisclosed	AA live-endpoint median (ext.)
Fireworks	381.2	undisclosed	AA live-endpoint median (ext.)
CoreWeave	261.8	GB300 NVL72 rack	AA live-endpoint median (ext.)
Nebius	222.4	undisclosed	AA live-endpoint median (ext.)
Together (FP4)	218.2	undisclosed	AA live-endpoint median (ext.)
GMI (FP8)	40.2	undisclosed	AA live-endpoint median (ext.)

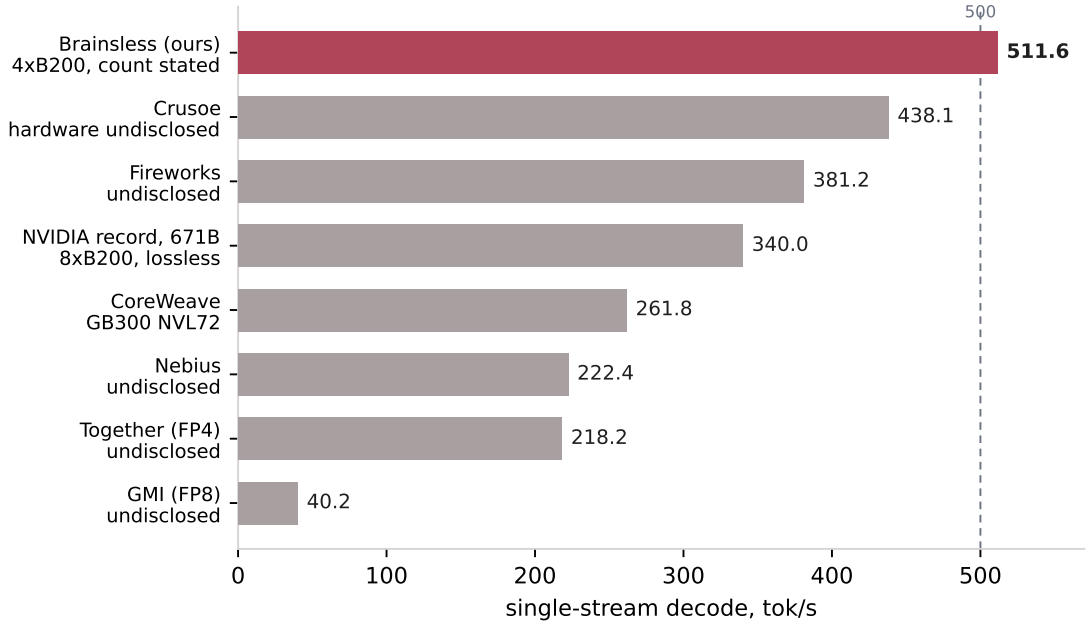

Figure 5. GPU serving of Kimi-K2.6, single stream, 10k-token workload. External bars are Artificial Analysis medians pinned 2026-07-05, with NVIDIA's published 671B single-user record added for reference; ours is the raised record (set 505.9 at that pin, raised to 511.6 by blind re-runs of the release; the board leader read 451.0 at the raised cell's pin — `brl11_stage600r.json`, `brl11_repl_r1.json`). No other entry runs a configuration as small as four GPUs.

The leaderboard listed fourteen providers at the pin; the six with published speed readings appear in the table, and the fastest external reading on the board was Crusoe's 438.1. Cerebras serves the model at 981 tok/s on wafer-scale silicon through a private endpoint (external); it is a different hardware class, out of scope for a GPU comparison. The best documented lossless single-user number in this model class on GPUs is NVIDIA's 340 tok/s on DeepSeek-R1 671B, eight B200s,

TRT-LLM [NVIDIA, 2026]; their 368 requires relaxed acceptance, which costs MMLU-Pro 2.8 points and is lossy.

Three scoped claims. First, 511.6 is the fastest measured GPU serving of Kimi-K2.6 at this workload: 505.9 (depth 7) set the record and stood 15.5% above the board's leader at its pin, and blind re-runs of the public release raised it to 511.6 (depth 6), 13.4% above the leader at the raised cell's pin (451.0). Second, 505.9 was the first GPU reading above 500 tok/s on this model. Third, per accelerator the claim is scoped to disclosed counts: ours is four GPUs, within one GPU of the physical floor (the 554 GiB NVFP4 checkpoint does not fit on three 192-GB B200s), and no other entry in the table states a count.

The record-setting configuration in full: SGLang v0.5.14, EAGLE-3 speculative decoding [Li et al., 2025] with the public `lightseekorg/kimi-k2.6-eagle3.1-mla` 3B draft head [LightSeek, 2026], draft depth 7, chain top-k 1, fp8 KV cache, `tokenspeed_mla` attention backend, `flashinfer_trtllm` MoE backend, Kimi-K2.6 NVFP4 (554 GiB), tensor-parallel 4 on one 4×B200 node (Modal cloud; rate in the compute statement). The raised 511.6 cell differs in draft depth only (6); every other component is identical.

Speed is serving capacity

The reason a faster step lowers the cost of inference is that a serving cluster is paid for by the hour and produces a finite number of tokens in that hour. Cutting the step raises the tokens a fixed cluster emits, so the same hardware carries more of the load, and cost per token served falls with no change in price or precision. On this node the step-cost work moved single-stream decode from 150.9 to 505.9–511.6 tok/s, so each GPU-second produces up to 3.39× the tokens it did with speculation off. Interactive serving is where that capacity converts most directly to cost: users feel a response only above some tokens-per-second floor, and a faster per-user rate lets more users sit above that floor on one GPU before any of them slows down. The faster step raises the whole interactivity-versus-concurrency frontier, which is the frontier a deployment's cost per token rides on. At saturation the drafter's margin narrows, as the target GPU is already bandwidth-bound: at 32 concurrent streams the stack sustains 910 tok/s aggregate against 845 with no drafter, and the block-diffusion alternative that loses the single-stream race falls to 499 (§6, prior-engine measurement, `brl10_k26_full_v2.json`). The gain is largest in the interactive regime our product serves, and it is capacity, not a discount: the same mechanism that sets the record is the one that lets a fixed cluster serve more.

9 Measurement types, and a live replication of the leaderboard's path

The table in §8 names both measurement types. Our number is a controlled server-side benchmark: one node, measured by us, prompts pinned in advance, timing at the server, at the leaderboard's own workload. The leaderboard's numbers are third-party network measurements of live commercial endpoints. Ours is the only entry in the table with disclosed hardware, a published protocol, and per-request artifacts, and the release contains everything needed to re-run it (§12).

A live replication of the leaderboard's measurement path

On 2026-07-06 we replayed this report's record protocol against the production API of Fireworks, the second-fastest provider on the pinned board and the fastest whose serving is purchasable through a public self-serve API: the same sixteen SHA-pinned 10k-token prompts, 2,048-token outputs, temperature 0.6, the same per-request streaming statistic, sent from a standard paying account

over the public internet. The production `kimi-k2p6` endpoint returned an interpolated median of **118.4 tok/s** (n=16); with reasoning off it returned 149.2 (n=8), and the provider's `kimi-k2p6-turbo` router returned 168.9 (n=8). Every request completed its full 2,048 tokens, and every per-request row is in the artifact (`fireworks_live_replication.json`). The leaderboard credited the same provider 381.2 at the pin — 2.3–3.2× the production readings.

We measured one provider, from one client region, on one day, and we do not know what configuration served either measurement. Explanations consistent with the gap include regional routing, time-varying load, tiered or dedicated deployments, and the possibility that the endpoint the leaderboard measures is configured differently from the production API sold to customers; our data cannot distinguish among them, and we attribute nothing. What the replication establishes is the part that matters for reading §8: a leaderboard median is a property of the specific endpoint and conditions the leaderboard measures, we could not reproduce it from the production API, and the gap is more than an order of magnitude larger than the board's own day-to-day movement (its leading entry moved 438.1 to 451.0, about 3%, across the two days we pinned it). The field comparison in this report is therefore stated against the board's numbers as published. Our own number reproduces from a published protocol on disclosed hardware (§12).

The comparison is sometimes framed as apples to oranges: a controlled benchmark on one side, medians of endpoints under production load on the other. The replication above is the answer to that framing. When we replayed the identical workload against the same provider's production API — the endpoint a customer actually buys — it read 118–169 tok/s, not the 381.2 the board credits. A board median is therefore not a measurement of "live serving" in any general sense; it is a reading of one endpoint under one set of conditions, and it did not survive outside them. There is exactly one number in this comparison that anyone can re-measure — ours — and the code to test both, our cell and theirs, is in the release. That fragility cuts both ways, and the claim is scoped to survive it: the field table is a ranking over published measurements, not over provider capabilities. We claim the fastest measured number; whether a funded team could match it is exactly what the release lets them show, at $15, and this report's own thesis — the step is mostly software, the components public — says the attempt would land close. Independent of the field entirely, the number stands against physics: 3.39× the undrafted floor on the same node and protocol. Any controlled cell published above 511.6 will be cited in this report's next revision.

Replication of the record from the public release

On 2026-07-06 the protocol was re-run three times from the public release on fresh node draws, blind: gates fixed before the runs, no configuration touched, one of the runs clean-room from the public materials alone — the repository located from this report, the prompts rebuilt to the pinned SHA, the published medians re-derived from the raw artifact rows, the node rented fresh. Every n=16 cell landed inside the stated node-draw envelope: the tool depth-6 cells read 511.6, 484.4, and 474.2 on anchors of +0.9, −5.7, and −4.3% against reference; the depth-7 cell read 486.5 with a first-eight median of 538.0; math read 405–422 against the published 419.2. Within these sessions individual requests peaked at 568.0 tok/s (per-request rows in the artifacts). The best replication cell (511.6, depth 6, n=16) exceeds both the record-setting cell (505.9, depth 7) and its like-for-like depth-6 counterpart (487.2), and raises the record. Replication moved this record up, not down — and a record whose replications read above it is a floor: every component in the configuration is stock and public, and what moves the number from here is the trained head (§3, §7). Artifacts: `brl11_repl_r1.json`, `brl11_repl_cleanroom.json`, `brl11_repl_fovea.json`.

Two scope notes. The headline cell is agentic tool-calling traffic, the traffic our product serves; math is reported alongside (419.2), and general chat reads lower on every head we have measured (τ 2.2–2.9 at this workload against 3.5–5.3 on tool and math), because chat is high-entropy and acceptance is entropy-bounded. And node draws move throughput ±5–10%; every anchor is printed in Table 2.

10 What generalizes

The coefficients in this report belong to one model, one quantization, and two engine builds; the instruments do not. The weights-free differential requires only a proposer with no neural forward that the engine can run at two draft widths under an identical matcher configuration — every major serving stack ships one — and the proposer's own overhead cancels exactly wherever it runs, so the verify-cost curve is measurable on any speculative system without engine instrumentation. The draft-pass price falls out of one no-speculation cell plus the verify slope, by subtraction. The cross-engine control rests on an identity this report verified directly: accepted length is a property of the head and the traffic, not the engine (τ 4.825 against 4.815 at equal depth, same head), so equal-τ step-time differences attribute to software wherever two engines serve the same head. And the cross-family frame — accepted-length advantage times step-cost penalty decides the row — prices any drafter pair on any target, not just the two families measured here.

Three findings are stated so they can be refuted elsewhere: that batch-1 steps at this scale are dominated by software rather than memory bandwidth (the eight-GPU null); that optimal draft depth is priced by the engine's per-position cost rather than by acceptance alone (the k=6→7 shift across engines); and that per-position acceptance tails break super-geometrically, so geometric extrapolation overpredicts (418 forecast against 407 measured). What is specific to this target: the absolute coefficients, the tail's break point, and the reversal's magnitude. Under this method each is a ∼$30 question per model, which is the reason to publish the method.

11 Limitations

The record is one model, one quantization (NVFP4), one node type, and two traffic domains at the record depth; chat and code were measured at other depths and engines only. Every cross-engine and cross-arm comparison we treat as controlled is same-session, same-node; cross-session throughput drifts by tens of tok/s and node draws by ±5–10%, which is why anchors are printed per

session. The verify-width measurement is a differential through a weights-free proposer; it cancels proposer overhead under identical matcher configuration, and it is indirect. Direct observation of the expert union through a per-step unique-experts counter remains open and leads the next session. The record awaits independent verification; the prompts, seeds, serve commands, and per-request artifacts are published to enable it. The absolute step-cost coefficients are properties of the measured engine builds (vLLM pinned, SGLang v0.5.14) and will move with versions; the decomposition method is the durable contribution.

12 Reproduction

The record cell re-runs with one command from the public release:

```
modal run stage600_r.py
```

against a rented 4×B200 node — about $15 and forty minutes at the published rate. The runner rebuilds the prompts, asserts their SHA-256 against the pin, fail-closes on any mismatch, and writes the same per-request artifact schema this report's tables are built from. Everything else needed to re-run the tables is in the release: the SHA-256-pinned prompt manifests, the session runners (`record_run.py`, `stage600_a/e1/b/sg/r/d.py`), the exact engine versions and flag sets per session, and raw per-request output text stored for independent tokenizer normalization. The serving gotchas that cost us dead sessions are documented with the runners: CUDA-graph capture sizes must be multiples of $(1+k)$; the NVFP4 repo ships no chat template; fp8 KV with a non-MLA drafter needs FlashInfer query-quantization disabled; vLLM's CPU ngram method fails config validation with async scheduling (use `ngram_gpu`); the SGLang draft head must be pinned to unquantized loading or it silently inherits the target's FP4 config; and warm reloads (390–590s against 1,540s cold) make multi-arm sessions affordable. The draft heads are public [LightSeek, 2026]; Fovea checkpoints publish after license review.

Compute statement

The full campaign behind this report ran $162 of rented serverless GPU time inside a $300 budget: stage A $25.20, the verify differential ∼$26, the TP8 null $41.50 (8×B200 at $50/hr; all other sessions 4×B200 at $24/hr), the SGLang smoke $11.10, the record session $12.90, the depth ladder $9.00, the 2026-07-06 replications $34.20 of node time across six draws (three completed re-runs and three aborted boots), and about $2 of production-API usage for the leaderboard-path replay. Prior-context spend for the earlier sessions this report reuses is itemized in the prior edition's compute statement.

Artifact index

artifact	carries
`brl11_stage600r.json`	the record set: 505.9 (depth 7) and 419.2 cells, depth-6 arms
`brl11_repl_r1.json`, `brl11_repl_cleanroom.json`	2026-07-06 replications from the public release: d6 511.6 / 484.4, d7 486.5
`brl11_repl_fovea.json`	replication run three: d6 474.2 / math 422.1, and the same-session Fovea-
`fireworks_live_replication.json`	the leaderboard-path replay: production endpoints 118.4/149.2/168.9 tok/
`brl11_stage600sg.json`	SGLang smoke: 400.5/486.9/498.5, cross-engine τ parity (τ/step values fr
`brl11_stage600e1.json`	verify-width differential: ngram_gpu $k5/k8$, 0.397 ms/token slope
`brl11_stage600b.json`	TP8 null: T_1 6.54ms, EAGLE $k5$ step 12.32ms on 8×B200
`brl11_stage600a.json`	T_1 6.63ms, k-ladder, per-position conditionals, session anchor
`brl11_stage600d.json`	depth ladder: depth 8 498.0; ladder peak at depth 7
`brl11_record.json`, `brl11_baseline.json`	July-4 cells, live k-forecast (397.9 vs 395.8), prompt SHA
`brl11_verify_tax_session.json`	n=16 short-prompt cross-family matrix, per-position vectors
`brl10_k26_full_v2.json`	released-budget cross-family matrix; 32-stream saturation
`fovea_verdict.json`, `fovea_gauntlet.json`	Fovea-1 pairs; KV-dtype and block-length sweeps
`fovea_e_result.json`	Fovea-E course: teacher-capture fix, held-out acceptance
`brl11_p0_manifest.json`	prompt manifest, SHA-256 pins

References

Artificial Analysis. Kimi K2.6 provider benchmarks. https://artificialanalysis.ai, 2026.

DeepSeek-AI. DSpark: Confidence-scheduled speculative decoding with semi-autoregressive generation. https://github.com/deepseek-ai/DeepSpec, 2026. Released 2026-06-27.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE-3: Scaling up inference acceleration of large language models via training-time test. *arXiv preprint arXiv:2503.01840*, 2025.

LightSeek. lightseekorg/kimi-k2.6-eagle3.1-mla model card. https://huggingface.co/lightseekorg, 2026.

LMSYS / SGLang team. The next generation of speculative decoding: Dflash and spec v2. https://www.lmsys.org/blog/2026-06-15-next-generation-speculative-decoding-dflash-v2/, 2026.

NVIDIA. Boost inference performance up to 15x on nvidia blackwell using dflash speculative decoding. NVIDIA Technical Blog, 2026-06-23, 2026.

Z-Lab. DFlash: Block diffusion for flash speculative decoding. *arXiv preprint arXiv:2602.06036*, 2026.