



What Fits on Two GPUs

Capability profiles of an efficiency-tier and a flagship-tier open-weight model on sovereign hardware

Paper: PT-R-2026-037 v1.1

Author: PureTensor Ltd

Date: 17 October 2026 (version 1.1)

Status: Published

Abstract

We run open-weight language models on our own hardware in two tiers. The efficiency tier is a model that serves our agents around the clock from two GPUs, sharing them with other services. The flagship tier is a model with a much larger checkpoint that fills a dedicated four-GPU node and is brought up when a task needs it. This paper profiles one model in each tier across six evaluation suites. It asks which capabilities fit in the small footprint, which still grow with model size, and what the measurement itself taught us. A larger model scoring higher is the expected result; the shape of the gap is the useful finding.

The efficiency-tier model is NVIDIA Nemotron-3.5-Super-VL-120B-A12B-NVFP4 (120B parameters, 12B active, a 70 GiB checkpoint). The flagship-tier model is DeepSeek-V4.1-Flash-1017, released on the day of our measurements (a 476 GiB checkpoint, including an Engram memory table of about 189 GiB that we hold in host RAM). With 15% of the flagship's checkpoint size and half its GPUs, the efficiency tier reached 92% of the flagship's score on competition and science reasoning (84.7% against 92.3%), 98% on AIME 2026 averaged over four samples, and 86% on repairing 95 real defects from our own code (84.2% against 97.9%). Its vision score was on par with the flagship's (94.8% against 90.5%; the difference is within the intervals). On long documents up to 222,000 tokens it came within one task per family of the flagship at finding facts (39 of 40 multi-hop chains, 39 of 40 multi-key lookups, 19 of 20 questions on real text).

Two capabilities still track size. Factual recall on SimpleQA-verified was 13.0% for the efficiency tier against 29.3% for the flagship, a gap consistent with the flagship's far larger store of parameters, including the Engram table. Tracking state across a whole long document (following chains of variable assignments, counting matching sentences among distractors) was the other: 26 of 40 and 7 of 40, against 40 and 24 of 40.

The measurement mattered as much as the models. Reading answers with a strict line-start parser put the efficiency tier's vision score at 44.8%; reading the final answer wherever it appears gave 94.0%. Its reasoning score rose from 67.8% to 84.7% when items that hit the token budget were given twice the budget. Its tokenizer produces 7–10% more tokens than the flagship's on the same text, which made our first largest long-context bucket unfair to it until we resized it with both tokenizers.

At each card's full power limit, the efficiency tier decoded 226 tokens per second for a single request and 1,535 per second across 16 concurrent requests, and the flagship 102 and 463



(across 8, its configured maximum). Raising the GPUs from our operating power caps to their maximum changed decode throughput by less than 5% and prefill by 8–13%, so our operating caps cost under 5% of decode speed and 8–13% of prefill speed.

This is one evaluation day, mostly single samples, on workstation Blackwell hardware that is not on NVIDIA's validated deployment list. Section 13 states which claims survive that.

1. Two Tiers, One Proving Ground

Our agents call a language model throughout the day: to route work, use tools, summarise, draft and check. That model has to be available at every hour, answer quickly, and leave room on its GPUs for the other services the agents depend on (embedding, reranking, transcription, OCR, speech). Some tasks need more than that model can give: long research questions where factual recall decides the answer, analysis of documents running to hundreds of thousands of tokens, hard multi-step reasoning, or repairs to unfamiliar code. For those we bring up a flagship-class model on a dedicated node.

The two tiers are chosen for different jobs, so ranking them against each other would answer the wrong question. The questions that matter to anyone running models on their own hardware are these. How much of the measurable capability fits into the small footprint? Which capabilities still need the large one? And how far do measurement choices (token budgets, answer parsing, context sizing, serving settings) move the scores before any model property shows up?

The efficiency tier in this study is NVIDIA's Nemotron-3.5-Super-VL, which has served as our resident agentic model since 16 October 2026. It earned that role in an internal bake-off: on our internal frontier benchmark (94 items across ten dimensions, every verdict produced by code rather than by a language-model judge) it scored 90.1 on the agentic seat at its default reasoning setting. The flagship tier is DeepSeek-V4.1-Flash-1017, released on 17 October 2026 as a minor update to DeepSeek-V4.1-Flash. On the same benchmark it scored 90.2 on the agentic seat, in a separate run at a different reasoning setting. For agentic tool use, the job we give the efficiency tier, the two scored within a tenth of a point of each other, in runs that were not paired.

2. The Models and How We Served Them

Nemotron-3.5-Super-VL is a hybrid architecture: interleaved Mamba-2 layers (a state-space layer whose memory does not grow with context length), mixture-of-experts layers and attention layers, with a C-RADIO vision encoder and native multi-token prediction, which lets the model draft several tokens per step for speculative decoding. NVIDIA publishes it in NVFP4, a 4-bit floating-point format, produced by quantisation-aware distillation from the BF16 model. DeepSeek-V4.1-Flash-1017 is a mixture-of-experts model with a much larger checkpoint, FP8 weights, FP4 routed experts, an Engram memory table (large n-gram embedding tables looked up at inference time, which we serve from host memory) and DSpark speculative decoding.

	Efficiency tier	Flagship tier
Model	Nemotron-3.5-Super-VL-120B-A12B-NV FP4	DeepSeek-V4.1-Flash-1017
Parameters	120B total, 12B active	not stated on the model card
Checkpoint	70 GiB	476 GiB (Engram table about 189 GiB)
GPUs	two RTX PRO 6000 Blackwell Workstation Edition (600 W maximum, operated at 350 W)	four RTX PRO 6000 Blackwell Max-Q (325 W maximum, operated at 250 W)



	Efficiency tier	Flagship tier
Engine	vLLM, the image named on NVIDIA's model card	SGLang, a community port for this GPU generation
Parallelism	tensor parallel 2	tensor and expert parallel 4
KV cache	FP8, as declared by the checkpoint	FP8, by design
Context served	262,144 tokens	262,144 tokens
Share of node	50% of each GPU's memory, eight other services alongside	the whole node, no other GPU tenants

We served the efficiency tier with the exact vLLM image NVIDIA names on its model card. We used the Mamba-state settings from the card's quick-start configuration (fp16 state with stochastic rounding and shared-prefix checkpointing; that configuration is written for DGX Spark) and native multi-token prediction with three draft tokens, the setting the card lists for two H100s. Its KV cache is FP8 because the checkpoint itself declares FP8 KV quantisation, which vLLM's "auto" setting follows. The model held half of each GPU's memory, a KV pool of 2,321,846 tokens. The other half served eight other inference services, and the model kept answering our live agentic traffic throughout the evaluation.

The flagship tier ran alone on its node: four GPUs, about 189 GiB of host RAM for the Engram table, and a KV pool of 2,393,600 tokens. It takes five to seven minutes to load. All 49 of its weight files matched the publisher's checksums.

3. Six Suites, and Why Three of Them Are Our Own

Public benchmarks are the common language of model evaluation, but a model released this week may have seen them. We therefore built three of our six suites (long context, coding and vision) from our own material, which cannot be in either model's training data, and wrote the 30 writing prompts ourselves.

Suite	Items	Source	Scoring
Knowledge	300	seeded sample of SimpleQA-verified	correct, incorrect or not attempted, on the SimpleQA rubric
Reasoning	261	AIME 2026 (30), HMMT February 2026 (33), GPQA-Diamond (198)	integer, symbolic equivalence, option letter
Long context	180	2,646 of our own documents as the haystack, at 32k, 64k, 128k and 222k tokens	exact value, ordered list, set or integer
Coding	95	real defects fixed in our own repository in October 2026	the defect's own regression test, run in a sandbox
Vision	116	charts, tables, terminal screenshots, report pages and dashboards we rendered	exact, numeric or transcription match
Writing	30	business-writing prompts	blind pairwise judgement in both presentation orders

The long-context tasks plant facts at controlled depths inside real documents: chains of three or four linked facts (multi-hop), several key and value pairs of which three are asked for (multi-key), chains of variable assignments with distractor chains (variable tracking), sentences to be counted among near-identical distractors (counting), and cloze questions on values that occur exactly once in the real text. Each family appears at every length.



The coding tasks come from 132 defects fixed in our own repository in October 2026. We considered 146; we dropped 11 that touched several files, 2 with no source change and 1 whose test did not reproduce. The remaining 132 defects, grouped by the file they change, gave 97 single-file tasks whose regression tests failed before the fix and passed after it, twice. Two more were dropped because their prompts already contained most of the fix, leaving 95. Each prompt gives the defect description with an AI reviewer's root-cause analysis, the file, the regression test and its failing output; the model answers with search-and-replace edits, which we apply in a fresh sandbox before running the test. This measures repair given a diagnosis, the way our own agents receive work.

4. Measurement Protocol

Each model used its vendor's recommended sampling: temperature 1.0 and top_p 0.95 for Nemotron text (0.6 and 0.95 for vision), temperature 1.0 and top_p 1.0 for DeepSeek. Nemotron's thinking was on, its highest reasoning level; DeepSeek ran at reasoning effort "high" (75 of 100), and "max" (100) for competition math. Knowledge and vision ran with thinking off for both.

Output budgets, including reasoning tokens, were the same for both models: 512 tokens for knowledge, 32,768 for competition math, 16,384 for GPQA, 8,192 for long context, 32,768 for coding, 2,048 for vision and 6,144 for writing. A response that hits its budget is wrong at that budget. Every such item was then sampled once more at twice the budget, and we report both scores; the extended-budget score is the headline because it separates a model's answer from its budget. Competition math was sampled three more times at 65,536 tokens and is reported as the average of four samples (avg@4).

Answers are read from the last "ANSWER:" marker anywhere in the response; a response that is a single short line with no marker is taken as the answer. Free-text knowledge answers that did not match exactly, and all writing, were judged by Claude models working blind: items were shuffled and anonymised, and each verdict is bound to a hash of the exact response it judged, so a re-run never inherits an old verdict. Intervals are 95% Wilson intervals, or percentile bootstrap intervals over problems for avg@4.

5. What Fits on Two GPUs

Suite	Flagship tier	Efficiency tier	Efficiency tier relative to flagship
Reasoning (261)	92.3% (88.5–95.0)	84.7% (79.8–88.5)	92%
Competition math, avg@4	94.0% (88.5–98.4)	86.1% (78.6–92.9)	92%
Coding (95)	97.9% (92.6–99.4)	84.2% (75.6–90.2)	86%
Long context (180)	91.1% (86.0–94.5)	72.2% (65.3–78.2)	79%
Vision (116)	90.5% (83.8–94.6)	94.8% (89.2–97.6)	on par
Knowledge (300)	29.3% (24.5–34.7)	13.0% (9.7–17.3)	44%

The relative column is descriptive: it places each score against the flagship's, not a ranking, and differences inside the intervals are not findings. Read against its footprint, the efficiency tier's profile is the main result of this study. It occupies 15% of the flagship's checkpoint size and half its GPUs, and it leaves room on those GPUs for eight other services. On reasoning it reached 92% of the flagship's score, on code repair 86%, and on vision it was on par. Its share falls on long context (79%) and furthest on knowledge (44%), and those two are where the next sections look.



On AIME 2026 the two tiers are indistinguishable: 29 of 30 problems each on the main sample, and 93.3% (85.0–99.2) against 95.0% (86.7–100) averaged over four samples, a 98% share. On HMMT February 2026 the efficiency tier averaged 79.5% (66.7–90.2), an 85% share, and on GPQA-Diamond it answered 167 of 198, a 92% share of the flagship's 181.

6. Where the Footprint Shows

Factual recall shows the largest gap, consistent with a capability that grows with stored parameters. On SimpleQA-verified, questions with short, verifiable answers about obscure facts, the flagship answered 88 of 300 correctly and the efficiency tier 39. Both declined to answer about equally often (105 and 108 times). When they did answer, the flagship was right 45.1% of the time and the efficiency tier 20.3%. That is consistent with storage capacity: a 120B-parameter model in a 70 GiB checkpoint has less room for long-tail facts than a model whose checkpoint is nearly seven times larger and includes a 189 GiB lookup table, though the two checkpoints use different precisions and two models cannot establish the cause. History was the one topic where the efficiency tier answered more questions correctly (7 of 17, against 5). For an agentic model the practical answer is retrieval: our agents do not rely on any model's recall of obscure facts, and they ground answers in our own memory and search systems.

Long context divides more finely. Finding planted facts, even linked chains of them, held up well for the efficiency tier at every length up to 222,000 tokens:

Long-context family (40 tasks; real-document questions 20)	Flagship tier	Efficiency tier
Multi-hop chains	40	39
Multi-key lookup	40	39
Questions on real document text	20	19
Variable tracking	40	26
Counting among distractors	24	7

The difference is in tasks that need state carried across the whole document: which variables end up holding a value after a chain of assignments, or how many of many similar sentences match a condition. Both tiers find counting the hardest task in the suite. Across all families the efficiency tier scored 78% at 32k, 82% at 64k, 73% at 128k and 56% at 222k, against 98%, 98%, 87% and 82% for the flagship, so its shortfall is largest at the longest length. In practice this marks the line between the tiers: the efficiency tier is a dependable reader of long material for retrieval and question answering, and whole-document bookkeeping at the longest lengths is a flagship job.

7. Reasoning, Length and the Token Budget

The efficiency tier uses more reasoning tokens. At the equal budget it used 11,576 output tokens per reasoning item on average against 6,675 for the flagship, about 1.7 times as many, and 79 of its 261 reasoning items reached the budget before giving an answer (the flagship's figure was 30). Its equal-budget reasoning score was 67.8%. Given twice the budget on those items, it rose to 84.7%; the flagship rose from 85.4% to 92.3%. Thirty-one efficiency-tier items and eight flagship items were still unfinished at twice the budget, so both extended scores are lower bounds at these budgets.

Our budgets were below what both vendors suggest. DeepSeek recommends 256K or more output tokens, and NVIDIA's card offers a reasoning level, "on (medium)", that generates fewer tokens for some loss of accuracy. We did not test that level. The lesson applies to both tiers,



since the flagship's score also rose with the budget: a reasoning score depends on the budget it is given, and a score quoted without the budget beside it is incomplete.

8. Code Repair

On 95 real defects from our own code, each verified by the regression test written for it, the efficiency tier produced a passing fix for 80 and the flagship for 93. The efficiency tier fixed the single highest-priority defect, 46 of 49 lower-priority ones and 33 of 45 in the middle band.

Its 15 unsuccessful attempts split into two groups. Six were edits that applied cleanly and failed the test, the ordinary failure of a fix that is wrong. The other nine never reached the test: four quoted code that does not match the file, four had malformed edit markers, and one contained tool-call markup inside the edit. Some of these may be failures of the edit format rather than of understanding; whether a structured editing tool, in place of free-text search-and-replace blocks, would recover them is an experiment we list in Section 14. Our parser recovers three unambiguous format slips for both models (a replace marker used as the divider, a missing replace marker before the next search block, and text ending inside a replacement); it recovered six efficiency-tier responses and one flagship response. Without that recovery the scores are 83.2% and 96.8%.

9. Perception and Writing

Vision was the efficiency tier's strongest suite: 110 of 116 correct, on par with the flagship's 105. It read 29 of 30 dense tables and terminal screenshots correctly (flagship 24), all 28 dashboard panels (flagship 27) and 27 of 28 report pages (flagship 28); on charts both scored 26 of 30. For an agent that reads screenshots, dashboards and scanned documents, this is a capability the small footprint delivers at a level on par with the flagship.

Both tiers produced professional-quality writing on all 30 prompts. Blind judges, seeing each pair twice in swapped order, preferred the flagship on 17 prompts and the efficiency tier on 8, with 5 ties or split decisions; that difference is not significant (sign test $p = 0.11$), and the two orders agreed on 20 of 30 prompts. Mean judge scores were 7.58 and 7.17 out of 10. The judges' reasons for preferring the efficiency tier most often concerned meeting requested lengths and polish; for the flagship, not inventing details the prompt did not supply, and arithmetic and technical accuracy.

10. Throughput at Full Power

Accuracy does not depend on a card's power limit, but speed can, and we operate both tiers below their cards' maximum: the efficiency tier's Workstation Edition cards at 350 W of a possible 600 W, and the flagship's Max-Q cards at 250 W of a possible 325 W. To report what this hardware delivers rather than what our operating caps allow, we lifted every card to its maximum for a short throughput test and then restored the caps.

The workload was the same for both tiers: one fixed technical-writing prompt, thinking off, exactly 1,024 output tokens per request, the vendors' sampling settings, and streaming so that the first token could be timed. Prefill was measured separately on prompts of about 32,000 tokens, a different prompt for every request so that no prefix cache could help.

At each card's maximum power	Efficiency tier (600 W)	Flagship tier (325 W)
Decode, 1 request (tokens per second)	226	102
Aggregate, 4 concurrent requests	617	307



At each card's maximum power	Efficiency tier (600 W)	Flagship tier (325 W)
Aggregate, 8 concurrent requests	1,008	463
Aggregate, 16 concurrent requests	1,535	not configured (8-sequence maximum)
Time to first token, 1 request	0.11 s	0.13 s
Prefill, prompts of about 32k tokens (tokens per second)	16,558	9,237

Lifting the caps changed decode throughput very little: by -0.8% to +4.2% for the efficiency tier (largest at 16 concurrent requests, 1,472 to 1,535 tokens per second) and by -0.3% to +1.4% for the flagship. Decode is limited by memory bandwidth more than by power: at maximum, the efficiency tier's cards peaked at 446 W and the flagship's at 269 W, and neither reached its limit. Prefill is compute-bound and gained more, 13% for the efficiency tier (14,596 to 16,558 tokens per second) and 8% for the flagship (8,525 to 9,237).

Two notes keep these figures in proportion. Both tiers use speculative decoding, whose acceptance rate depends on the text being written, so decode rates vary with content; these are for one prose prompt. And the two tiers run on different card variants, so the table describes each tier on its own hardware rather than ranking the models. For the efficiency tier the figures are also conservative: it served them while sharing its two GPUs with eight other services and our live agentic traffic.

11. Seven Things the Measurement Taught Us

Several of the largest effects in this study came from the instrument, not the models. We list them because each one would have changed a published number.

1. Read the answer wherever the model puts it. The efficiency tier often appends its final answer to the end of its last sentence ("...is 32.ANSWER: 32") or answers with a single bare line. A parser that required the marker at the start of a line scored its vision suite at 44.8%. Reading the last marker anywhere, and accepting a one-line bare answer, gave 94.0% at the equal budget. The same rules applied to both models.
2. Report the budget with the score. Doubling the budget on truncated items moved the efficiency tier's reasoning score from 67.8% to 84.7%.
3. Size long contexts with every tokenizer under test. The efficiency tier's tokenizer produces 7-10% more tokens than the flagship's on the same text. Our first largest bucket, sized at about 237,000 flagship tokens, left the efficiency tier under 8,192 tokens of output room on all 45 prompts, and one prompt exceeded its 262,144-token context. We added a 222k bucket sized with both tokenizers: 218,510-219,661 flagship tokens and 235,519-242,348 efficiency-tier tokens, leaving at least 19,756 tokens of output room after the chat template. The original bucket is reported separately (82% and 69%).
4. Check what "auto" resolves to. The efficiency tier's "auto" KV cache setting resolves to the checkpoint's own FP8 calibration, and the flagship stores its KV cache in FP8 by design, so both ran at their published precision. Two complete runs of the efficiency tier, first with our production serving flags and then with the model card's Mamba settings, agreed within 1.5 points on every suite: knowledge 12.3% and 13.0%, reasoning 66.3% and 67.8%, long context 67.8% and 68.9%, vision 94.8% and 94.0%.
5. Separate format from substance in code edits. Recover only unambiguous slips, apply the rule to every model, and publish the strict score beside it.
6. Bind every judgement to the response it judged. A verdict keyed only by item and model would have been silently reused when we re-ran a model.



7. Audit the scorer. A blind re-grade of 80 randomly sampled scored items from the reasoning, long-context and vision suites agreed with the automatic scorers on all 80, no truncated response was credited anywhere, and one request in the whole study failed (the context overflow in point 3).

12. How We Deploy the Two Tiers

The results are consistent with the division we already run. The efficiency tier is our resident agentic model. It answers at every hour from two GPUs whose other half carries embedding, reranking, transcription, OCR, speech and a second language model, and on our internal frontier benchmark it scored within a tenth of a point of the flagship for agentic tool use (in unpaired runs at different reasoning settings). It reads screenshots and documents on par with the flagship, follows long material to find what it needs, and reasons within about eight points of the flagship given the budget to finish.

The flagship tier is brought up for work where size still pays: research questions that turn on long-tail facts, bookkeeping across documents of a quarter of a million tokens, the hardest reasoning, and code repair where every attempt counts. It needs a whole four-GPU node and about 189 GiB of host RAM, and it takes five to seven minutes to load, which suits it to scheduled and on-demand work rather than to an always-on seat.

13. What These Numbers Will Not Carry

This is one evaluation day, 17 October 2026. Outside competition math every item was sampled once, and the suites are modest in size: 30 writing prompts, 30 AIME problems, 33 HMMT problems, 95 coding tasks and 45 tasks per long-context length. The intervals in Section 5 are wide where the samples are small, and differences inside them are not findings.

The tiers ran on different hardware: four Max-Q cards for the flagship and two Workstation Edition cards for the efficiency tier, the latter shared with other services. The throughput figures in Section 10 describe each tier on its own hardware at full power and are not a comparison of the models; they come from one prompt and one short test. Workstation Blackwell is not on NVIDIA's validated deployment list for this model, which names DGX Spark, H200, two H100s and GB200; results on a validated platform, or through NVIDIA's own NIM packaging, may differ. The flagship ran on a community port of its serving engine, which may cost it accuracy we cannot see. The budgets were below both vendors' suggestions, and the thinking settings differ by construction: DeepSeek exposes an effort scale and Nemotron a small set of reasoning levels, and each ran at a high setting.

AIME 2026, HMMT February 2026, GPQA-Diamond and SimpleQA are public, and either model may have seen them. Our long-context, coding and vision suites cannot be in any training data, but the coding diagnoses were written by AI reviewers (Claude and Grok models), and their style may suit some models more than others. Free-text knowledge answers and all writing were judged by Claude models; no human judged the writing. The long-context tasks share one pool of documents, so they are not fully independent of each other.

Within those limits, these claims survive. On this hardware and at these budgets, the efficiency tier reached 92% of the flagship's reasoning score, 98% on AIME 2026 (93.3% against 95.0%, with wide intervals on 30 problems), 86% on code repair and a vision score on par with the flagship's, from 15% of the checkpoint size and half the GPUs. Factual recall and whole-document state tracking are where the larger model's size shows. And two measurement choices each moved a score by more than 16 points (answer parsing moved the efficiency tier's vision score from 44.8% to 94.0%, the token budget its reasoning score from 67.8% to 84.7%),



while sizing contexts with only one tokenizer would have left every prompt in our largest bucket an unfair test for the other model.

14. What We Changed and What Comes Next

The scorers now read the final answer wherever it appears, every published score carries its budget, and long-context buckets are sized with each model's own tokenizer. The efficiency tier stays our resident agentic model, and the flagship is now available on demand for research, long-document and repair work.

The next experiments follow from the limits above: the efficiency tier at its "on (medium)" reasoning level and at larger budgets, its edit failures re-run through a structured editing tool, and the same suites on a platform from NVIDIA's validated list.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. This paper is PT-R-2026-037. All measurements were taken on 17 October 2026, except the internal frontier benchmark score of the efficiency tier, measured on 16 October 2026. Version 1.1 adds Section 10, throughput measured with every GPU at its maximum power limit. Raw artefacts, including every prompt, response, score and judge verdict, are retained.