Whitepaper · MedErtas v0.1 · October 2026

Specialisation as inference optimisation: domain-aware quantisation and drafting for open medical LLMs

Edward Xi Yang, Ertas AI

Knowing the domain buys speed and keeps the accuracy.

We frame specialisation as inference optimisation: a model served for one domain runs faster when its serving stack is fitted to that domain. Our lightweighting study pulls two levers. Domain-aware quantisation calibrates 4-bit compression on domain text and keeps higher precision in the layers the domain's own tasks are most sensitive to, so the speed of lighter weights comes with accuracy held close to full precision. Domain-aware drafting trains the speculative decoder on domain answers, so it predicts more of each answer and adds speed directly. We apply both to medicine and release MedErtas v0.1, two 4-bit models.

In our lightweighting study, existing published 4-bit builds of the same models report near-full recovery on general benchmarks but lose 4 to 6 points on clinical calculation in paired tests on 12 medical evaluation sets. MedErtas-v0.1-31B-LS, from Gemma 4 31B, keeps 99.6% of its base's mean score with no set significantly below it; our calibration and keeping attention in FP8 together recover most of the calculation loss. MedErtas-v0.1-27B-LS, from Qwen3.8-27B, keeps 98.2% and sits above both existing published 4-bit builds. At one user on one RTX PRO 6000, both answer calculation questions about 8 times faster than their full-precision bases without speculative decoding. Of that, the 31B's medical drafter adds 13% over Google's general drafter.

Key findings

99.6% and 98.2%
of the full-precision mean score across 12 sets

MedErtas-v0.1-31B-LS averages 78.1 against Gemma 4 31B's 78.4, with no set significantly below it after correction. MedErtas-v0.1-27B-LS averages 77.0 against Qwen3.8-27B's 78.4, above both existing published 4-bit builds of that model (76.4 and 76.7).

1.4 points
clinical calculation gap on the 31B

Existing published 4-bit builds of Gemma 4 31B lose 4.1 to 6.4 points on MedCalc-Bench. In our lightweighting study, our medical calibration recovers about half of that loss, and keeping attention in FP8 most of the rest (+1.8 points, 95% interval −0.5 to +4.1): 63.9 against 65.3 at full precision.

About 8x

per-user speed on calculation

With a single user on one RTX PRO 6000, the 31B answers calculation questions at 189 tokens/s and the 27B at 207, against 22 to 26 tokens/s for the full-precision models without speculative decoding. Exam-style questions run 6 to 7 times faster.

+13%

per user from the 31B's medical drafter

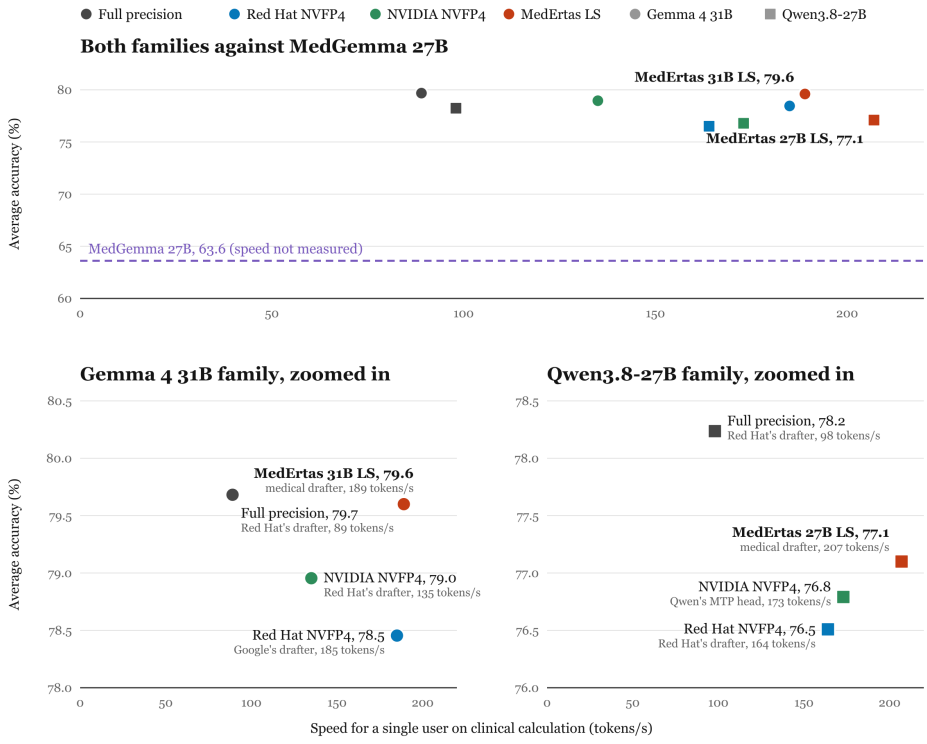
Fine-tuning Google's assistant drafter on medical answers raised accepted tokens per step on calculation from 4.75 to 5.39 and per-user speed from 168 to 189 tokens/s. In one run at 64 concurrent users, aggregate calculation throughput rose 38%, against a spread of about 15% between identical builds at that load.

The big picture

Figure 1 places both MedErtas models among comparable builds of the same open models: full precision, and the existing published 4-bit builds from Red Hat and NVIDIA, each served with the fastest public drafter in our runs. Accuracy is the average of the 11 medical sets the Medmarks leaderboard shares with our runs; speed is per user on clinical calculation.

MedErtas is the fastest build in each family at close to full-precision accuracy: 79.6 against 79.7 for the 31B, and 77.1 against 78.2 for the 27B. MedGemma 27B, a medical model of similar size, averages 63.6 on the same sets, about 13 points below every build of either family.

Figure 1 **Speed against accuracy, with MedGemma 27B for reference**



Average of the 11 medical sets shared with the Medmarks leaderboard. Each build on the fastest public drafter in our runs, one RTX PRO 6000 Blackwell, vLLM 0.32.0.

Accuracy is the average of the 11 medical sets the Medmarks leaderboard shares with our runs, thinking off; MedGemma 27B's average comes from the leaderboard snapshot of 26 October 2026, and its speed was not measured. Each build is served with the fastest public drafter in our runs, and MedErtas with its medical drafter. Speed is measured with a single user on the server, on one RTX PRO 6000 Blackwell with vLLM 0.32.0.

Specialisation as inference optimisation

Specialisation as inference optimisation fits the serving stack to the domain a model serves. Our lightweighting study, the method behind the LS profile, pulls two levers, each with prior work in other settings (see References).

Domain-aware quantisation decides how the weights are compressed. Post-training quantisation sets each layer's scales from the activations of a calibration set. The existing published builds calibrate on general chat and instruction data. Ours calibrate on medical text with a general share, and the precision layout, which layers keep more than 4 bits, is chosen by measured sensitivity on medical tasks. The lighter weights supply the speed; the domain-aware choices keep the accuracy.

Domain-aware drafting trains the speculative decoder on medical answers. Speculative decoding speeds up a model in any domain, and a drafter that knows the domain's wording adds speed directly. Only the drafter is trained; the model keeps its released training.

Every calibration and drafter prompt comes from training splits, checked against the evaluation items so that none is reused.

Models and builds compared

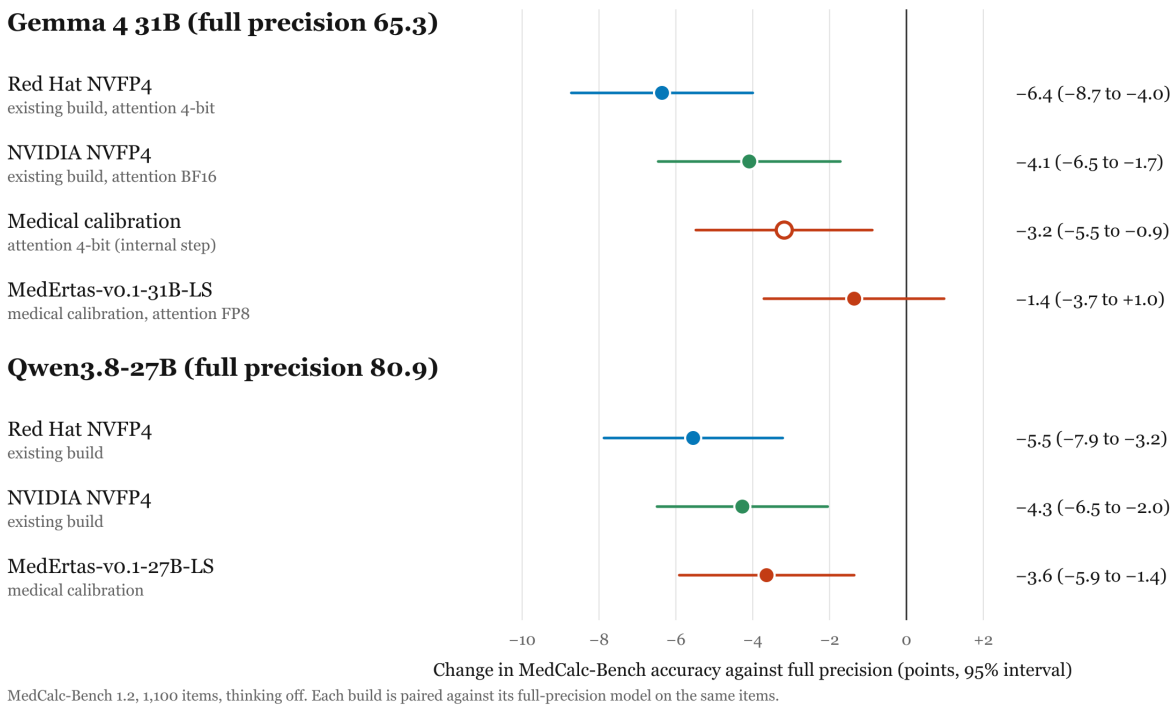
Both MedErtas models start from the instruction-tuned releases of two open models under Apache-2.0: Gemma 4 31B from Google and Qwen3.8-27B from Qwen. Every build is compared against its full-precision (BF16) model and against the two existing published NVFP4 builds of the same model, from Red Hat and NVIDIA.

NVFP4 stores weights in 4-bit floating point with fine-grained scales and runs natively on NVIDIA Blackwell GPUs. The builds differ in which layers keep higher precision, and each build's shipped configuration shows it. For Gemma 4 31B, Red Hat's existing build quantises attention to 4-bit and NVIDIA's keeps attention in BF16. For Qwen3.8-27B, both existing published builds keep attention in FP8.

Domain-aware quantisation: where 4-bit loses clinical accuracy

Across the 12 sets, the existing 4-bit builds lose most on clinical calculation: 4.1 to 6.4 points on MedCalc-Bench, each significant after correction (Figure 2). Their vendors report near-full recovery on general benchmarks, so only a medical evaluation shows the gap. Our lightweighting study therefore chose the precision layout on this task.

Figure 2 **Clinical calculation across existing and MedErtas 4-bit builds**



MedCalc-Bench 1.2, 1,100 items, thinking off, each build paired against its full-precision model on the same items. The hollow marker is an internal step, medical calibration in Red Hat's all-4-bit layout, shown to separate the effect of calibration from the effect of FP8 attention.

On Gemma 4 31B we separated the two effects in steps: our medical calibration in Red Hat's all-4-bit layout, then the same calibration with attention in FP8, for 0.9 GB more weight than Red Hat's build. MedErtas-v0.1-27B-LS uses the layout of NVIDIA's existing build, with attention already in FP8, and our medical calibration; its remaining calculation gap is the target of the next version.

We can simplify the source of the 31B's recovery into these steps, each measured on the same 1,100 MedCalc-Bench items. Because every step is paired on the same items, the steps add up exactly. What each step is attributed to is the simplification.

$$\Delta_{\text{LS}} = \Delta_{\text{existing}} + \delta_{\text{calibration}} + \delta_{\text{layout}}$$

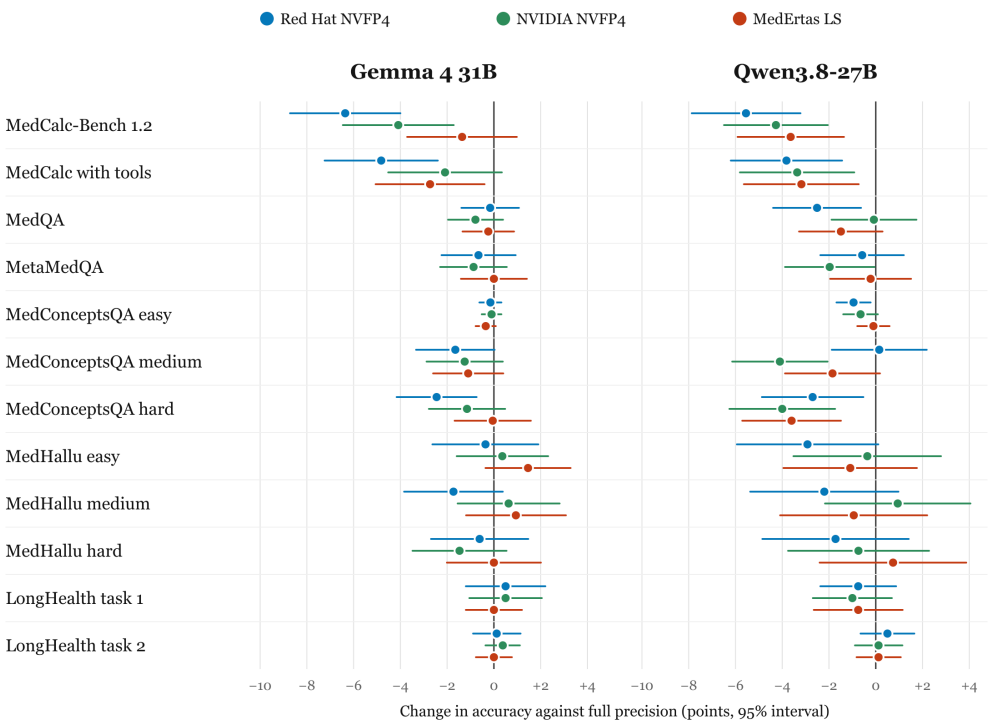
$$-1.4 = -6.4 + 3.2 + 1.8$$

Δ existing is Red Hat's existing build against full precision. δ calibration is our calibration in the same all-4-bit layout: +3.2 points, 95% interval +0.8 to +5.6, p 0.013. It differs from Red Hat's build in quantiser settings as well as calibration data, so it measures our calibration as a whole. δ layout is keeping attention in FP8 with the same calibration: +1.8 points, 95% interval −0.5 to +4.1, p 0.14. Together the two steps are clearly significant (+5.0 points, p 0.0001); the layout step on its own is an estimate within the noise of a single run.

Accuracy across 12 medical sets

Away from calculation, both builds stay close to full precision (Figure 3). MedErtas-v0.1-31B-LS has no set significantly below Gemma 4 31B after correction and averages 78.1 against 78.4 (Table 1). MedErtas-v0.1-27B-LS averages 77.0 against 78.4, above both existing published builds of Qwen3.8-27B at 76.4 and 76.7 (Table 2).

Figure 3 **All 12 sets, paired against full precision**



Medmarks harness, thinking off, 400 to 2,000 items per set. Each build is paired against its full-precision model on the same items. Accuracy for every set is in Tables 1 and 2.

Table 1 **MedErtas-v0.1-31B-LS against Gemma 4 31B on 12 sets**

Accuracy in %, Medmarks harness, thinking off. Every build is paired against BF16 on the same items.

Set (items)	BF16	Red Hat NVFP4	NVIDIA NVFP4	MedErtas 31B LS
MedCalc-Bench 1.2 (1,100)	65.3	58.9*	61.2*	63.9
MedCalc with tools (1,100)	63.8	59.0*	61.7	61.1
MedQA (1,273)	93.2	93.1	92.5	93.0
MetaMedQA (1,373)	70.4	69.7	69.5	70.4
MedConceptsQA easy (2,000)	98.8	98.7	98.7	98.5
MedConceptsQA medium (2,000)	71.7	70.0	70.4	70.5
MedConceptsQA hard (2,000)	69.2	66.7	68.0	69.1
MedHallu easy (548)	83.6	83.2	83.9	85.0
MedHallu medium (636)	80.2	78.5	80.8	81.1
MedHallu hard (816)	64.1	63.5	62.6	64.1
LongHealth task 1 (400)	90.2	90.8	90.8	90.2
LongHealth task 2 (800)	89.8	89.9	90.1	89.8
Mean of 12 sets	**78.4**	**76.8**	**77.5**	**78.1**

* significant against BF16 after Holm correction. MedConceptsQA tests ICD-10 coding.

Table 2 **MedErtas-v0.1-27B-LS against Qwen3.8-27B on 12 sets**

Accuracy in %, Medmarks harness, thinking off. Every build is paired against BF16 on the same items.

Set (items)	BF16	Red Hat NVFP4	NVIDIA NVFP4	MedErtas 27B LS
MedCalc-Bench 1.2 (1,100)	80.9	75.4*	76.6*	77.3
MedCalc with tools (1,100)	79.5	75.6	76.1	76.3
MedQA (1,273)	88.2	85.7	88.1	86.7
MetaMedQA (1,373)	65.6	65.0	63.7	65.4
MedConceptsQA easy (2,000)	97.6	96.6	96.9	97.5
MedConceptsQA medium (2,000)	67.2	67.4	63.1*	65.4
MedConceptsQA hard (2,000)	64.5	61.8	60.5*	60.9
MedHallu easy (548)	82.7	79.7	82.3	81.6
MedHallu medium (636)	76.4	74.2	77.4	75.5
MedHallu hard (816)	59.2	57.5	58.5	59.9
LongHealth task 1 (400)	89.2	88.5	88.2	88.5
LongHealth task 2 (800)	89.2	89.7	89.4	89.4
Mean of 12 sets	**78.4**	**76.4**	**76.7**	**77.0**

* significant against BF16 after Holm correction. MedConceptsQA tests ICD-10 coding.

Domain-aware drafting: medical drafters

Speculative decoding pairs the model with a small drafter that proposes several tokens at once. The model checks them in a single pass and keeps only tokens it would have produced itself, so the output follows the main model. With a draft of γ tokens, each accepted with probability α, one verification step yields τ tokens on average (Leviathan et al.):

$$\tau = \frac{1 - \alpha^{\gamma + 1}}{1 - \alpha}$$

A drafter trained on general text guesses medical wording less well, so α is lower on medical prompts. For the 31B we fine-tuned Google's Gemma 4 assistant drafter on medical answers and serve it at 8 draft tokens. That raised α from about 0.83 to 0.87 on clinical calculation, taking τ from 4.75 to 5.39. The drafter's size and draft length are unchanged, so speed should rise in proportion to τ:

$$S_{\text{domain}} \approx \frac{\tau_{\text{medical}}}{\tau_{\text{general}}} = \frac{5.39}{4.75} \approx 1.13$$

We measured 12.5% against the 13.5% predicted, from 168 to 189 tokens/s. For the 27B we trained a DSpark speculative decoder on medical answers, served at 7 draft tokens. It accepts 0.2 to 0.35 more tokens per step on calculation than Red Hat's general drafter, for 207 against 188 tokens/s. On long patient records both medical drafters accept slightly fewer tokens than the general ones, which is the drafter work still open.

We release both medical drafters under the name Perigee, as MedErtas-v0.1-31B-LS-Perigee and MedErtas-v0.1-27B-LS-Perigee.

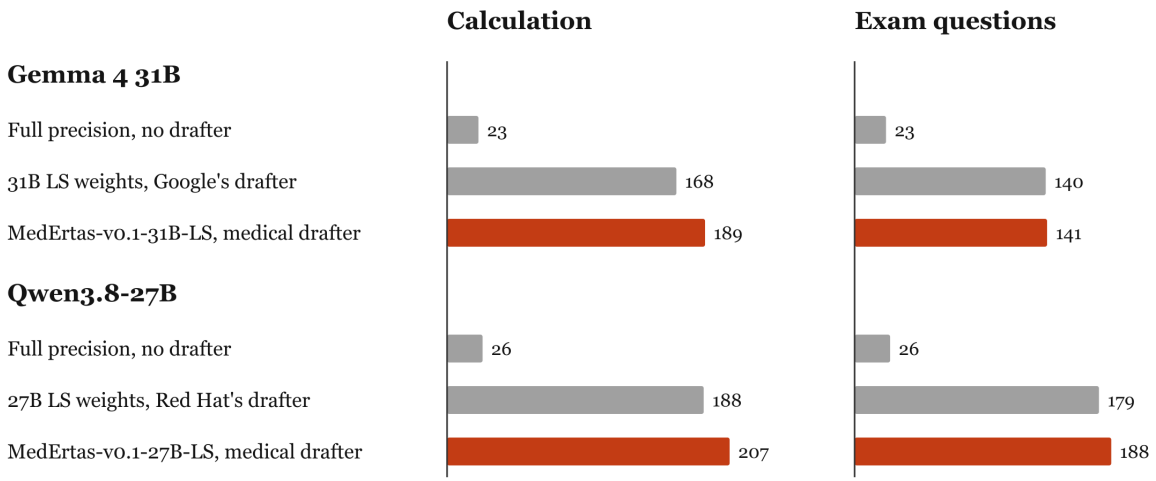
Where the speed-up comes from

We can simplify the source of the speed-up into three factors, each measured by changing one thing at a time on the same card, server and prompts: lighter weights, speculative decoding, and a drafter trained on the domain. The breakdown is illustrative and leaves other serving factors aside, such as batching, kernels, the KV cache and prompt length, which are held fixed here.

$$S_{\text{total}} \approx S_{\text{weights}} \times S_{\text{spec}} \times S_{\text{domain}}$$

For the 31B, per-user speed on clinical calculation rises from 22.5 tokens/s at full precision to 51 with the 4-bit weights, 168 with Google's drafter and 189 with the medical drafter: 2.27 × 3.29 × 1.13 ≈ 8.4. For the 27B it rises from 26 to 60, 188 and 207: 2.31 × 3.13 × 1.10 ≈ 8.0 (Figure 4).

Figure 4 **Speed for a single user**



Output tokens per second for a single user on the server, one RTX PRO 6000 Blackwell, vLLM 0.32.0, medical prompts. Full precision runs 22 to 23 tokens/s on Gemma 4 31B, shown as 23. One run per configuration.

A single user on the server, one RTX PRO 6000 Blackwell, vLLM 0.32.0, medical prompts, one run per configuration. Full precision runs 22 to 23 tokens/s on Gemma 4 31B, shown as 23.

The weights factor follows from how decoding works. Generating each token reads every weight from GPU memory, so per-user speed v is bounded by memory bandwidth B over the bytes read per token M:

$$v \lesssim \frac{B}{M}, \qquad S_{\text{weights}} \approx \frac{M_{\text{BF16}}}{M_{\text{LS}}} = \frac{62.6}{24.2} \approx 2.6$$

With about 1.8 TB/s on the RTX PRO 6000, the bound predicts a 2.6x gain for the 31B as its weights shrink from 62.6 GB to 24.2 GB. We measured 2.3x; activations, attention over the context and kernel overheads account for the rest.

Choosing between the two models

The two models suit different work (Table 3). The smaller 27B leads clinical calculation by 13 to 15 points. The 31B leads medical knowledge, ICD-10 coding and hallucination detection by 3 to 8 points on most sets. Long-record reading is level between them.

Table 3 **Which model for which job**

Accuracy in % for the two MedErtas builds on the same sets, from Tables 1 and 2.

Set	MedErtas 31B LS	MedErtas 27B LS
MedCalc-Bench 1.2	63.9	77.3
MedCalc with tools	61.1	76.3
MedQA	93.0	86.7
MetaMedQA	70.4	65.4
MedConceptsQA medium / hard	70.5 / 69.1	65.4 / 60.9
MedHallu easy / medium / hard	85.0 / 81.1 / 64.1	81.6 / 75.5 / 59.9
LongHealth task 1 / task 2	90.2 / 89.8	88.5 / 89.4

Evaluation and speed protocol

We score on the Medmarks harness with thinking off, on 12 sets: clinical calculation (MedCalc-Bench 1.2, with and without a calculator tool), exam questions (MedQA), recognising when a question cannot be answered (MetaMedQA), ICD-10 coding (MedConceptsQA at three difficulty levels), hallucination detection (MedHallu at three levels) and facts from long patient records (LongHealth, two tasks). Sets run 400 to 2,000 items. Every run serves on vLLM 0.32.0 on one RTX PRO 6000 Blackwell. Our full-precision Gemma 4 31B run lands within 1.2 points of the public Medmarks leaderboard on all 11 sets the board shares, and the harness reproduces the board for Qwen3.8-27B as well.

Each build is paired against its full-precision model on the same items. We report the difference in accuracy with a 95% interval for a paired difference, test it with an exact McNemar test, and apply Holm correction across every set and build compared for that model. The harness samples at temperature 1.0: two full-precision runs of Qwen3.8-27B differ by under 1 point on 11 sets and by 3.0 on the hardest coding split.

Speed runs use the same card and vLLM version on three medical workloads: clinical calculation, exam questions and long patient records. We report output tokens per second with a single user on the server, the speed one person sees when nobody else is using it, and also measured 8, 32 and 64 concurrent users. Each configuration ran once. Throughput at 64 users varied by up to 15% between identical builds, so we read smaller differences at that load as noise. Long-record speeds stay out of the headline figures, since with inputs of about 12,000 tokens they fall sharply as concurrency rises.

Why it matters

Our lightweighting study points to one practice for teams serving medical models: measure a compressed model on the domain's own tasks, paired against full precision. Existing published 4-bit builds pass their vendors' general benchmarks and still give up ground on clinical calculation, which only a medical evaluation shows. Calibrating the compression and choosing its layout on those tasks is what holds the accuracy.

Throughput sets the price of a hosted model. 4-bit weights and a drafter trained on the domain raise per-user speed about eightfold over full precision without a drafter, which is what lets us offer the hosted beta below the going rate for the full-precision models.

Each model needs under 26 GB of weights, less than half the memory of its full-precision checkpoint, which brings single-card on-premises and air-gapped deployment within reach. Speed so far is measured on the RTX PRO 6000 only.

Limitations

All scores are single runs of the Medmarks startup subset with thinking off, sampled at temperature 1.0. A paired test on about 1,100 MedCalc items resolves a change of roughly 3.5 points, so a difference that is not significant still allows a loss smaller than that. The 27B sits 3.6 points below its base on MedCalc-Bench, significant before correction, and the 31B 2.7 points below on MedCalc with tools.

Each speed configuration ran once, on medical prompts. Throughput at 64 users varies by up to 15% between identical builds. Long patient-record prompts slow sharply at high concurrency, and the drafters have not been measured on general-domain prompts. The weights use NVFP4, which needs an NVIDIA Blackwell GPU.

MedErtas models are beta research releases. They are not medical devices and are not intended for diagnosis or treatment decisions without qualified clinical oversight.

Models and links

MedErtas model family

Models, beta API pricing and example runs

ErtasAI/MedErtas-v0.1-31B-LS

4-bit Gemma 4 31B build, calibrated on medical data

ErtasAI/MedErtas-v0.1-31B-LS-Perigee

Perigee, the medical drafter for the 31B build: Google's Gemma 4 31B assistant drafter, fine-tuned on medical answers

ErtasAI/MedErtas-v0.1-27B-LS

4-bit Qwen3.8-27B build, calibrated on medical data

ErtasAI/MedErtas-v0.1-27B-LS-Perigee

Perigee, the medical drafter for the 27B build: Red Hat's DSpark drafter, trained further on medical answers

Common questions

Is 4-bit quantization safe for medical models?

Every medical model, including one at full precision, should be built into a carefully engineered and reviewed system before production use. MedErtas aims to match its full-precision base on medical capability while making the model faster and cheaper to serve, and each release publishes the comparison with full precision for every set we test. Compressed builds from other publishers can still lose accuracy on particular medical tasks. Test any model against full precision on your own clinical tasks before relying on it.

What does specialisation as inference optimisation mean?

It is the idea behind MedErtas: a model served for one domain runs faster when its serving stack is fitted to that domain. LS models pull two levers. Domain-aware quantisation calibrates the compression on domain text and chooses which layers keep higher precision by their sensitivity on the domain's tasks. Domain-aware drafting trains the speculative decoder on domain answers. Each lever has prior work. The LS profile combines them and is measured against full precision on the domain's own evaluations.

Which MedErtas model should I use?

Use MedErtas-v0.1-27B-LS for clinical calculation, where it leads by 13 to 15 points. Use MedErtas-v0.1-31B-LS for medical knowledge questions, ICD-10 coding and hallucination detection, where it leads on most sets. Long-record reading is level between them.

Does the speculative decoder change the answers?

Speculative decoding checks every drafted token against the main model and keeps only tokens that model would produce itself, so answers follow the 4-bit model's own distribution. The drafter changes how quickly they arrive.

What does LS mean in the model name?

LS, for Lightweight Study, is the build profile. An LS build compresses the weights with domain-aware quantisation and pairs them with a drafter trained on the domain, with the model's own training unchanged. Later profiles will cover models post-trained on domain data.

What hardware do the models need?

The weights use NVFP4, which runs on NVIDIA Blackwell GPUs. MedErtas-v0.1-31B-LS is 24.2 GB with a 1.0 GB drafter, and MedErtas-v0.1-27B-LS is 22 GB. The speeds in this report were measured on one RTX PRO 6000 with vLLM 0.32.0.

Can MedErtas be used for clinical decisions?

MedErtas models are beta research releases and are not medical devices. They suit building and testing medical software, with qualified clinicians responsible for any clinical decision.

About the authors

Edward Xi Yang, Founder & CEO

Edward Xi Yang is the founder and CEO of Ertas AI, which builds custom small language models that run on-device, in customers' own cloud or on Ertas-managed infrastructure. He leads the company's research into training small specialist models for long-document tasks.

Cite this work

```
@techreport{ertas2026medertas,
  title       = {Specialisation as Inference Optimisation: Domain-Aware Quantisation and Drafting
for Open Medical {LLMs}},
  author      = {Yang, Edward Xi},
  institution = {Ertas AI},
  year        = {2026},
  month       = oct,
  note        = {MedErtas v0.1},
  url         = {https://www.ertas.ai/research/medertas-v01-specialisation-inference-optimisation}
}
```

References

Medmarks: medical benchmark suite and leaderboard

Medmarks paper (arXiv:2605.01417)

Khandekar et al., MedCalc-Bench: Evaluating Large Language Models for Medical Calculations

The Effect of Quantization on Clinical Benchmarks: Accuracy and Safety Across Model Families

Williams et al., Compressing Language Models for Specialized Domains

Training Domain Draft Models for Speculative Decoding: Best Practices and Insights

FastDraft: How to Train Your Draft

Leviathan et al., Fast Inference from Transformers via Speculative Decoding

Google, Gemma 4 31B instruct

Google, Gemma 4 31B assistant drafter

Qwen, Qwen3.8-27B

Red Hat AI, Gemma 4 31B NVFP4

Red Hat AI, Qwen3.8-27B NVFP4

NVIDIA, Qwen3.8-27B NVFP4