Gemma 4 vs Ternary Bonsai 2 on five GPUs: *Bonsai decodes 1.6–2.8× faster; with drafters the gap shrinks to 0.8–1.4×.*

We ran Google's Gemma 4 E2B, E4B and 12B against PrismML's Ternary Bonsai 2 2B, 4B and 9B on five GPUs, paired by size, with the same engine builds, tests and prompts — on the bare engine, and served the way each family is meant to be served, with and without its multi-token-prediction drafter. This is a speed comparison only: the quality half was put on hold before it ran.

Bottom line

- **On the engine, Bonsai 2 decodes 1.62–2.84× faster than the same-size Gemma 4 on every card:** 1.62–2.26× for the 2B against E2B, 1.72–2.03× for the 4B against E4B, 2.08–2.84× for the 9B against the 12B, the widest on the DGX Spark each time. Its weights are 2.9–6.6× smaller.
- **Prompt processing is closer:** Bonsai prefills 1.13–1.37× faster in the 2B pair and 1.31–1.46× in the 9B pair, and the 4B pair is level (0.94–1.10×; Gemma E4B reads slightly faster on the RTX 3090 and DGX Spark).
- **Served, the gap narrows before any drafter:** without drafters the Bonsai 4B runs at 1.33–1.88× the speed of E4B and the 9B at 1.86–2.70× that of the 12B. Bonsai kept 70–94% of its engine speed in the server, Gemma 92–100%.
- **Gemma's drafter gains far more than Bonsai's:** 1.41–2.65× against 1.07–1.30×. With up to 2 drafted tokens, Gemma E2B and E4B accept about the same share of drafts as Bonsai (58–62% against 60–63%) and the 12B more (72–74%).
- **With each family's drafter,** Gemma E4B beats the Bonsai 4B on the RTX 5080, RTX PRO 4500 Blackwell and RTX 3090 (by 3–25%); the Bonsai 4B stays ahead on the DGX Spark and RTX 5070, and the Bonsai 9B ahead of the Gemma 12B on every card (1.06–1.40×).
- **Speed only.** The quality half of this comparison was put on hold before it ran. Nothing on this page says which family answers better.

1 · What we ran

Two families of small models with public weights, paired by the size in their names: **Ternary Bonsai 2** from PrismML (2B, 4B and 9B, in PrismML's 2-bit PQ2_0 format) and **Gemma 4** from Google (E2B, E4B and 12B, in Google's quantization-aware-trained Q4_0 files). The same engine builds, tests and prompts for both, on five GPUs.

Pair	Model and file	Revision	Parameters	Weights	Bits per parameter	Drafter
2B class	Ternary Bonsai 2 2B · PQ2_0	90fa37ad	1.88 B	0.51 GB	2.15	none published
	Gemma 4 E2B · QAT Q4_0	675cff42	4.63 B	3.33 GB	5.76	Google's MTP drafter, 59 MB
4B class	Ternary Bonsai 2 4B · PQ2_0	c4169eab	4.21 B	1.14 GB	2.16	MTP head (Q8_0) in PrismML's 13 Oct file @ e9fe290d
	Gemma 4 E4B · QAT Q4_0	4b4a2c1d	7.46 B	5.14 GB	5.51	Google's MTP drafter, 60 MB
9B class	Ternary Bonsai 2 9B · PQ2_0	4df66037	8.95 B	2.41 GB	2.15	MTP head (Q8_0) in PrismML's 13 Oct file @ c507c837
	Gemma 4 12B · QAT Q4_0	29d09777	11.91 B	6.96 GB	4.68	Google's MTP drafter, 254 MB

Parameters and weights as llama-bench counts them (weights = the bytes loaded, GB of 10^9 bytes); bits per parameter = weights × 8 ÷ parameters, over the whole file. Google's E names count effective parameters: the E2B and E4B files hold 4.63 B and 7.46 B. Repositories: `prism-ml/Ternary-Bonsai-2-{2B,4B,9B}-gguf` and `google/gemma-4-{E2B,E4B,12B}-it-qat-q4_0-gguf`; Gemma's drafters from `ggml-org/gemma-4-{E2B,E4B,12B}-it-GGUF`, sizes from our download receipts. Bonsai's engine and no-drafter rows used PrismML's 6 October files; its drafter rows the 13 October release, which carries retrained weights of the same shapes and types plus the MTP head.

- **Engine:** PrismML's fork of llama.cpp, which the Bonsai formats need, in two releases: llama-bench from `prism-b10743` (build adfffbe) and llama-server from `prism-b10777` (commit e3de715). The same binaries ran both families on each card: Bonsai on 16 October, Gemma on 17 October.
- **Engine speed:** llama-bench with every layer on the GPU and flash attention on — decode of 128 tokens and prefill of a 512-token prompt, on an empty context and with 8,192 tokens already in it, mean of 5 runs.
- **Real use:** llama-server, one request at a time in a 16,384-token context, 24 prompts (three each of templated text, JSON, code, summaries, prose, chat, math and creative writing), up to 1,024 tokens per answer, thinking on, the same per-prompt seeds for both families, each family's own recommended sampling (Bonsai: temperature 1.0, top-p 0.95, top-k 20, presence penalty 1.0; Gemma: temperature 1.0, top-p 0.95, top-k 64, no penalty). Decode is the server's own count: generated tokens over generation time, summed over the 24 answers.
- **Drafters:** the Bonsai 4B and 9B and all three Gemma models have an MTP (multi-token prediction) drafter, which proposes the next tokens for the main model to verify in one step. Bonsai's is a head inside a separate release file, run with PrismML's flags (`--spec-type draft-mtp --spec-draft-n-max 2 --spec-draft-n-min 0 --spec-draft-p-min 0`). Gemma's is a separate file loaded with `--model-draft`, run with the same flags and once more allowing up to 4 drafted tokens (`--spec-draft-n-max 4`).

Card	Arch	Memory, GiB	Power limit	Condition
RTX 5080	sm_120	15.5	360 W	our second host, one card at a time
RTX PRO 4500 Blackwell	sm_120	31.4	165 W	same host; power-capped
RTX 3090	sm_86	23.6	370 W	same host
DGX Spark (GB10)	sm_121	121.7 (unified)	—	unified memory shared with the CPU; its resident server stopped for both runs
RTX 5070	sm_120	11.9	250 W	desktop display card: screen on, apps open, Windows

Architecture and memory as the engine reports them. All five at stock clocks; one card at a time, with no other job on it (the RTX 5070 also drives the screen). Builds: CUDA 12.9 for sm_86 and sm_120 in containers (RTX 5080, RTX PRO 4500, RTX 3090); a native CUDA 13.0 build on the DGX Spark; MSVC with CUDA 12.9 on Windows for the RTX 5070.

2 · Engine speed

Decode, empty context

Card	Bonsai 2B	Gemma E2B	×	Bonsai 4B	Gemma E4B	×	Bonsai 9B	Gemma 12B	×
RTX 5080	636.0	338.3	**1.88×**	365.3	204.7	**1.78×**	254.1	104.7	**2.43×**
RTX PRO 4500 Blackwell	529.9	282.1	**1.88×**	301.8	172.2	**1.75×**	209.0	88.1	**2.37×**
RTX 3090	482.8	264.9	**1.82×**	285.9	166.5	**1.72×**	190.5	91.6	**2.08×**
DGX Spark	287.5	127.3	**2.26×**	142.1	70.0	**2.03×**	88.8	31.3	**2.84×**
RTX 5070	399.4	246.5	**1.62×**	225.3	114.8	**1.96×**	154.4	59.4	**2.60×**

Tokens per second, one stream. llama-bench from PrismML's llama.cpp fork `prism-b10743` (build adfffbe), every layer on the GPU, flash attention on, mean of 5 runs. Bonsai 2 = PrismML's 2-bit PQ2_0 files; Gemma 4 = Google's QAT Q4_0 files. × = Bonsai ÷ Gemma on the same card. Stock clocks; the RTX PRO 4500 runs at a 165 W limit; the RTX 5070 is a desktop display card with the screen on; the DGX Spark is unified memory.

Decode with 8,192 tokens of context

Card	Bonsai 2B	Gemma E2B	×	Bonsai 4B	Gemma E4B	×	Bonsai 9B	Gemma 12B	×
RTX 5080	555.4	302.0	**1.84×**	310.9	185.7	**1.67×**	227.6	97.8	**2.33×**
RTX PRO 4500 Blackwell	467.3	254.2	**1.84×**	259.4	156.6	**1.66×**	189.7	82.5	**2.30×**
RTX 3090	454.5	245.9	**1.85×**	263.0	155.7	**1.69×**	180.5	85.5	**2.11×**
DGX Spark	255.6	116.4	**2.20×**	122.5	63.8	**1.92×**	80.5	29.3	**2.75×**
RTX 5070	359.3	229.0	**1.57×**	198.9	105.6	**1.88×**	141.6	56.2	**2.52×**

Prefill, 512-token prompt

Card	Bonsai 2B	Gemma E2B	×	Bonsai 4B	Gemma E4B	×	Bonsai 9B	Gemma 12B	×
RTX 5080	21,241	15,504	**1.37×**	11,470	10,426	**1.10×**	7,689	5,836	**1.32×**
RTX PRO 4500 Blackwell	16,647	12,401	**1.34×**	8,657	8,131	**1.06×**	5,718	4,129	**1.38×**
RTX 3090	12,817	10,565	**1.21×**	6,751	6,956	**0.97×**	4,463	3,419	**1.31×**
DGX Spark	9,439	8,353	**1.13×**	4,632	4,919	**0.94×**	3,191	2,192	**1.46×**
RTX 5070	14,045	12,306	**1.14×**	6,342	5,918	**1.07×**	4,154	3,028	**1.37×**

- **Bonsai decodes 1.62–2.84× faster on every card and every pair**, and the ratio is widest on the DGX Spark for all three pairs. The gap is smaller than the size gap: Bonsai's weights are 2.9–6.6× smaller.
- **Context costs the two families a similar share:** 8,192 tokens already in context cut Bonsai's decode by 5–15% and Gemma's by 5–11%; the decode ratio at that depth is 1.57–2.75×.
- **Prefill is much closer** — 1.13–1.37× for the 2B pair, 0.94–1.10× for the 4B pair, 1.31–1.46× for the 9B pair. The 4B pair is level; Gemma E4B reads slightly faster on the RTX 3090 and DGX Spark.
- Over the 5 runs, decode's standard deviation stayed under 5% of the mean. Prefill's reached 16%, mostly because the first run is slow.

3 · Real use, with and without each family's drafter

The engine figures leave out everything a server adds: sampling, the chat template, a growing context. In real use each answer runs through llama-server at the family's own sampling, with thinking on — once plain, and once with the family's drafter.

Bonsai 2 4B and Gemma 4 E4B

Card	Bonsai 4B	+ MTP head	Gemma E4B	+ drafter, 2	+ drafter, 4
RTX 5080	257.1	292.9	193.1	335.1	**367.0**
RTX PRO 4500 Blackwell	224.2	253.3	162.9	272.8	**279.9**
RTX 3090	217.0	242.2	158.2	249.9	**249.9**
DGX Spark	121.0	**153.1**	64.2	122.7	139.0
RTX 5070	196.8	**209.6**	113.4	192.4	200.1

Tokens per second, one stream. llama-server from PrismML's llama.cpp fork `prism-b10777` (commit e3de715), 24 prompts × up to 1,024 tokens, thinking on, each family's own sampling; decode from the server's timings. Bonsai 2 = PQ2_0 (the MTP column: PrismML's PQ2_0 file with its Q8_0 MTP head); Gemma 4 = QAT Q4_0 with Google's Q4_0 drafter allowed 2 or 4 drafted tokens. Same card conditions as section 2.

Bonsai 2 9B and Gemma 4 12B

Card	Bonsai 9B	+ MTP head	Gemma 12B	+ drafter, 2	+ drafter, 4
RTX 5080	196.6	**241.2**	98.0	194.3	224.6
RTX PRO 4500 Blackwell	168.6	**201.3**	82.9	163.4	185.5
RTX 3090	159.0	**192.4**	85.6	157.8	173.7
DGX Spark	80.0	**103.6**	29.6	61.6	74.0
RTX 5070	145.7	**167.3**	59.5	135.5	157.6

Gemma 4 E2B (no Bonsai 2B row)

Card	Gemma E2B	+ drafter, 2	+ drafter, 4	Drafter gain	Engine: Bonsai 2B	Engine: Gemma E2B
RTX 5080	310.5	479.2	485.7	1.56×	636.0	338.3
RTX PRO 4500 Blackwell	261.5	396.6	384.0	1.52×	529.9	282.1
RTX 3090	246.5	347.1	342.7	1.41×	482.8	264.9
DGX Spark	118.5	212.3	199.8	1.79×	287.5	127.3
RTX 5070	231.6	359.5	337.4	1.55×	399.4	246.5

Real use as above; the last two columns are llama-bench decode from section 2 for reference. The Bonsai 2B has no MTP head, and the 16 October run measured it on the engine only, so there is no served comparison for this pair.

How the gap moves

Card	4B ÷ E4B, engine	served	with drafters	9B ÷ 12B, engine	served	with drafters
RTX 5080	1.78×	1.33×	**0.80×**	2.43×	2.01×	**1.07×**
RTX PRO 4500 Blackwell	1.75×	1.38×	**0.90×**	2.37×	2.03×	**1.09×**
RTX 3090	1.72×	1.37×	**0.97×**	2.08×	1.86×	**1.11×**
DGX Spark	2.03×	1.88×	**1.10×**	2.84×	2.70×	**1.40×**
RTX 5070	1.96×	1.74×	**1.05×**	2.60×	2.45×	**1.06×**

Bonsai ÷ Gemma. Engine = llama-bench decode (prism-b10743, PQ2_0 / QAT Q4_0); served = llama-server (prism-b10777) without drafters; with drafters = Bonsai with its MTP head against the faster of Gemma's two drafter settings. Below 1 = Gemma faster.

1. **The server takes more from Bonsai than from Gemma.** Without drafters Bonsai kept 70–94% of its engine decode, Gemma 92–100%, so the 4B's lead over E4B falls from 1.72–2.03× on the engine to 1.33–1.88× served. We have not isolated the cause: the two families differ in architecture and in their recommended sampling (Bonsai's adds a presence penalty and a top-k of 20).
2. **Gemma's drafter is worth far more than Bonsai's** (table below). At the same setting of up to 2 drafted tokens, Gemma E2B and E4B accept about the same share of drafts as Bonsai (58–62% against 60–63%) yet gain 1.41–1.91× against Bonsai's 1.07–1.30×; the 12B accepts more (72–74%) and gains 1.84–2.28×. Our reading,

not traced in the engine: a Bonsai step is so cheap that the head's own cost is a large share of it, while Gemma's slower step leaves more room for verifying several tokens at once.

3. **With drafters on both sides,** Gemma E4B is the faster on the RTX 5080, RTX PRO 4500 Blackwell and RTX 3090, the Bonsai 4B on the DGX Spark and RTX 5070; the Bonsai 9B stays ahead of the 12B everywhere (1.06–1.40×). Like for like at 2 drafted tokens the same cards fall the same way (4B ÷ E4B 0.87–1.25×).

Model	Drafter	Drafted tokens, max	Accepted	Gain over no drafter
Bonsai 2 4B	MTP head	2	60–62%	1.07–1.27×
Bonsai 2 9B	MTP head	2	62–63%	1.15–1.30×
Gemma 4 E2B	Google's MTP drafter	2	58–60%	1.41–1.79×
		4	42–43%	1.39–1.69×
Gemma 4 E4B	Google's MTP drafter	2	60–62%	1.58–1.91×
		4	44–46%	1.58–2.17×
Gemma 4 12B	Google's MTP drafter	2	72–74%	1.84–2.28×
		4	57–60%	2.03–2.65×

Range over the five cards. Accepted = accepted ÷ drafted tokens, summed over the 24 answers. llama-server `prism-b10777`; Bonsai PQ2_0 with its Q8_0 head, Gemma QAT Q4_0 with its Q4_0 drafter (59–254 MB).

- **By prompt kind, both drafters gain most on math and JSON and least on prose and creative writing.** On the RTX 5070, Bonsai's head made some kinds slower (4B creative writing 0.95×; 9B prose 0.92×); Gemma's drafter never fell below 1.10× on any kind.
- **Two drafted tokens or four:** allowing 4 changed Gemma's served decode by 0.94–1.01× on E2B, 1.00–1.13× on E4B and 1.10–1.20× on the 12B (4 ÷ 2), with acceptance falling to 42–60%. Bonsai's head was run only at PrismML's setting of 2.

4 · What to run

This is a speed reading only; choose the family on quality first, which we did not measure. PrismML has published newer Bonsai files since these runs (section 5).

- **Around 9B:** the Bonsai 2 9B with its MTP head is the faster of the pair on all five cards, 1.06–1.40× ahead of Gemma 4 12B with its drafter, from 2.41 GB of weights against 6.96 GB.
- **Around 4B:** on the RTX 5080, RTX PRO 4500 Blackwell and RTX 3090, Gemma 4 E4B with its drafter at up to 4 drafted tokens is the faster; on the DGX Spark and RTX 5070, the Bonsai 2 4B with its MTP head.
- **Around 2B:** on the engine the Bonsai 2B decodes 1.62–2.26× faster than E2B; we have no served comparison.
- **Always load Gemma's drafter.** It costs 59–254 MB and raised served decode 1.41–2.65×. Allow up to 4 drafted tokens for E4B and the 12B; for E2B, 2 was as fast or faster on 4 of 5 cards.
- **Bonsai's MTP file is worth using** (1.07–1.30×), most on the DGX Spark and on math and JSON prompts.
- **A Bonsai 9B on a DGX Spark:** PrismML's smaller PTQ1_0 file decoded 1.13× as fast as PQ2_0 there on the engine, but 0.93–0.97× as fast on the discrete cards (16 October run, same build). Elsewhere, PQ2_0.

- **On small cards the size gap counts:** Bonsai's weights are 0.51–2.41 GB against Gemma's 3.33–6.96 GB, which leaves more of a 12 or 16 GB card for context. We did not measure how much context fits.

5 · Caveats

- **Speed only.** A matched quality run of both families (same harness, same sampling) was planned and put on hold before any job started. Nothing here ranks the families on answers.
- **Tokens per second is not time to an answer.** Inside the 1,024-token window, the Bonsai 4B and 9B gave no answer text yet, only reasoning, on 19–23 of 24 prompts; Gemma E2B and E4B on 3–9; the Gemma 12B on 13–19. How long each family thinks before answering was not measured.
- **Cut-offs:** every Bonsai answer ran to the 1,024-token cap; Gemma reached it on 16–20 of 24 (E2B), 12–17 (E4B) and 22–23 (12B). Decode is counted over the tokens generated, so a shorter answer reads at a slightly shorter context — a small effect at this length.
- **Sampled decoding, one run per arm** over 24 prompts with seeds shared by both families. Each family ran its own recommended sampling, which barely moves decode speed but can move drafter acceptance: the served columns compare the two families as they are meant to be served, not two engines.
- **Uneven drafter settings:** Bonsai's head ran only at 2 drafted tokens; Gemma's at 2 and 4, and the "with drafters" ratios take the faster of Gemma's two.
- **Same engine build for both:** yes. Every llama-bench row of both families carries build adfffbe (`prism-b10743`), and both server runs started the same `prism-b10777` binaries, compiled once per machine type (section 1). It is PrismML's fork, not upstream llama.cpp; Gemma's speed on upstream llama.cpp was not measured here.
- **A day apart:** Bonsai on 16 October, Gemma on 17 October, on the same cards, binaries, containers and flags.
- **Bonsai's files have changed since:** the engine and no-drafter rows use PrismML's 6 October files, the drafter rows its 13 October release. PrismML has since published newer files for all three sizes (14 and 15 October; checked 27 October), which we have not measured. Gemma's published weights and drafters are byte for byte the files we ran (sha256, checked the same day).
- **Containers on our second host loaded the GPU backend twice.** A check on the RTX 5080 (Bonsai 4B) with one backend read decode 365.7 against 365.9 tok/s and prefill 11,442 against 11,444: no effect, and both families ran the same way.
- **Conditions:** the RTX 5070 is a desktop display card and ran with the screen on and apps open; the DGX Spark's unified memory is shared with its CPU; the RTX PRO 4500 ran at its 165 W limit on the Bonsai 9B and Gemma 12B. Peak GPU temperature 36–60 °C on the discrete cards and 69–77 °C on the DGX Spark.
- **Not measured:** Gemma on our RTX PRO 6000 and RTX 5090 (so neither card is in this comparison), many users at once, contexts beyond 8,192 tokens, other quantizations of either family, GPU-side sampling (`--backend-sampling`), and image input.

Every figure on this page was computed from the raw rows of runs made on 16 and 17 October 2026 on our own machines. Run directories: `bench/runs/20261017-gemma4-speed` (Gemma 4: the scripts, every llama-bench and server row, telemetry and download receipts) and `bench/runs/20261016-bonsai2-speed` (Ternary Bonsai 2 on the same instruments). The rows are kept; to see them, or to send a model you want measured, write to hello@netsphere.com.ai. Online: netsphere.com.ai/reports/gemma-4-vs-bonsai-2-speed/