Faro — small models with in-weights ethics for legacy hardware

Technical report · v1 · September 14, 2026
Progetto Siliceo — Alfonso Riva
Artifacts and raw results: `state/faro/` (reports F0–F6, scripts, `bench-banco/`, `paper/`)

Abstract

We describe *Faro*, a family of instruction models of **≤ 2 GB** (q4_k_m) built to run on a **4 GB legacy GPU** as the local mind of an always-on assistant. The design constraint is unusual: **no external guardrail layer** — the behavioural rules (a constitution, *Candela*) and a working method are meant to live **inside the weights**. We present a reproducible pipeline — multilingual vocabulary pruning (IT/ES/EN), embedding resize, constitutional continued pre-training (CPT), supervised fine-tuning (SFT) on domain traces, GGUF export — and apply it *unchanged* to four architectures from different families: Qwen3.5-2B, MiniCPM5-2B, LFM2.5-2.6B and gemma-4-E2B. On a 112-question domain benchmark the trained models score **86.6 / 84.8 / 83.9 / 72.3 %** against their bases' 55.4 / 55.4 / 61.6 / 56.2 % (+16 to +31 points). All models fit the size budget and serve from quantized weights; one exports native tool calls. We report a 16-case ethics probe on the served models, deployment measurements on the target hardware (a GTX 1650), and discuss limitations honestly: single evaluation runs, no component-level ablations yet, and a domain-specific, rule-scored benchmark.

1. Introduction

Small language models are usually evaluated as *generalists rescaled down*. Our setting is different: a **local-first assistant** that must run continuously on hardware a household already owns (a 4 GB GTX 1650), without cloud calls, and whose failures matter — it must **not lie**: cite where knowledge comes from, refuse honestly what it cannot do, and never fake a source.

State-of-the-art practice would pair a small model with external guardrails. We argue the opposite for this class of deployments: the behavioural contract should be **in-weights** — part of what the model *is*, not a filter in front of it. This is both a philosophical choice (the assistant's ethics are not a middleware) and an engineering one: on a 4 GB GPU there is no room, latency, nor maintenance budget for a second system.

This report documents the *Faro* project: a pipeline to build ≤ 2 GB domain assistants and its application to four open-weight architectures. Contributions:

1. **A reproducible recipe** for sub-2 GB assistants: multilingual (IT/ES/EN) vocabulary pruning → embedding resize → constitutional CPT → trace SFT → GGUF export → local serving.
2. **In-weights ethics**: the constitution and procedure are trained in via CPT+SFT; no external guardrail is used at inference.
3. **A four-model study** across architecture families (dense Qwen, Llama-like MiniCPM, the hybrid LFM2, multimodal-derived Gemma 4), with identical data and hyper-parameters.
4. **Deployment evidence**: the models run the assistant's live workflows on target hardware, with measured speed and memory.

We are explicit about scope: this is an **engineering report**, not a claim of new theory. The benchmark is ours, runs are single, and component ablations are future work (§7).

2. Related work

Vocabulary adaptation. Trimming or adapting multilingual vocabularies is an established compression axis. Ushio et al. [1] show that a multilingual vocabulary can be *trimmed* to a target language while retaining performance, halving embedding size in practice. Purason et al. [2] propose leaf-based pruning (structure-aware trimming) together with continued BPE training, stressing that embedding parameters dominate the size of small models. VRCP [3] replaces the vocabulary and continues pre-training for resource-constrained environments; tokenizer-aware cross-lingual adaptation [4] uses a prune-with-extension strategy; Korean-centric token pruning [5] reports generation-stability gains and validates pruning as an optimization for memory-constrained, domain-specific deployments; related multilingual pruning studies appear in [6, 7]. Our recipe applies this practical line to a hard deployment budget (≤ 2 GB) across four architecture families, and measures the effect downstream, on a domain task, in the served artifact.

Constitutional and ethical fine-tuning. Constitutional AI [8] aligns models through critique–revision–preference loops against a written constitution. Subsequent work studies it at small scale: [9] applies CAI to 7–9B uncensored models; "Constitution or Collapse?" [10] finds that smaller models struggle with self-improvement (model collapse from noisy synthetic revisions), and follow-up work cleans revision data to stabilize the pipeline [11]. C3AI [12] studies how constitution principles should be crafted and evaluated; Collective CAI [13] sources constitutions from public input. Our approach differs in mechanism and target: we do **not** run self-critique loops — known to be fragile below ~10B — but internalize a fixed constitution and a working procedure through a short CPT followed by trace SFT, at the 2B scale, for deployment on 4 GB hardware.

Small models and edge deployment. Surveys document the rise and the costs of small language models [14, 15, 16, 17]; the ACL 2025 study in [14] evaluates 68 SLMs on edge hardware and lists **vocabulary and KV-cache compression** among the key optimization directions. Among our bases, LFM2 [18] is explicitly edge-first (hybrid short-convolutions + GQA), Gemma 4 [19] targets edge devices with per-layer embeddings (E2B), MiniCPM5 [20] targets on-device use, and Qwen3.5 [21] provides the 2B dense family. We fine-tune with QLoRA [22] and serve GGUF artifacts with `llama.cpp` [23]. Our contribution to this line is not

a new architecture but a documented, reproducible *deployment recipe* — with the behavioural contract inside the weights — evaluated end-to-end on legacy hardware.

Positioning. To our knowledge, few works combine, in one study: (i) sub-2 GB served artifacts, (ii) ethics in-weights with no external guardrail, (iii) one method replicated across four architecture families, and (iv) end-to-end service on a decade-old consumer GPU. We present the combination, its measurements, and its limits.

3. Method

3.1 Vocabulary pruning (IT/ES/EN)

Each base model ships a large multilingual vocabulary (128k–262k tokens). We prune it to the set **actually exercised by Italian, Spanish and English**, using frequency screening over mixed corpora (Wikipedia IT/ES/EN excerpts and project documents, including the training corpus). Special/control tokens are always retained. Resulting vocabularies:

model	original vocab	pruned vocab
Qwen3.5-2B	248,320	**36,932**
MiniCPM5-2B	130,560	**32,203**
LFM2.5-2.6B	128,000	**30,281**
gemma-4-E2B	262,144	**39,765**

The pruned tokenizer is rebuilt into a fast tokenizer file; a generic resize script then slices all vocabulary-indexed tensors (input embeddings, output head when untied, and architecture extras such as Gemma's per-layer embedding tensor) and remaps special-token ids. The resize tool is architecture-aware but model-agnostic, driven by a keep-set JSON.

3.2 Constitutional continued pre-training (CPT)

A compact corpus (~68 KB) containing the *Silicean Constitution* [24] and the *Searcher Method* — the assistant's ethical charter and its working procedure — is repeated ×8 and used for a short CPT run (60 steps, linear schedule, peak LR 2e-4) under QLoRA [22] (r=32, α=32, dropout 0.05). The goal is not knowledge injection but **internalization of the contract**: the model learns the voice, the procedure and the refusal style as *generative habits*.

3.3 Supervised fine-tuning (SFT)

450–480 hand-curated traces (13 batches covering domain Q&A, prior-conversation separation, calculations, anti-hoax behaviour, Spanish and ethics) are used for 3 epochs (cosine, LR 2e-4), in the same QLoRA configuration. Traces deliberately include **citation behaviour** (every answer cites its source), **confinement behaviour** (refusing what is outside the domain pack, explicitly) and **honest refusal patterns** (explicit refusal +

constructive alternative + reason). Tool-call traces are included for the models whose tool format is supported.

3.4 Export and serving

Merged adapters are exported to GGUF (q4_k_m) and served with `llama.cpp` [23] plus a quantized-KV extension (`ctk turbo4 / ctv turbo3_tcq` where the head dimension allows it; f16 KV otherwise). Serving enforces a **no-thinking** configuration: in our first evaluation round, the reasoning-style mode looped in the reasoning block ("thinking") and emptied the visible answer (22/112 empty outputs); with reasoning disabled the same weights answer directly and cleanly (0/112 empty). The models are trained to **reason in the content**, without a separate thinking block — this is visible in outputs (§5.2).

3.5 One recipe, four architectures

The pipeline required per-architecture adjustments only in *mechanics* (tied vs untied embeddings, extra per-layer embedding tensors, tool-call formats, GGUF pre-tokenizer fingerprints), never in method. For the largest base (gemma-4-E2B), training exceeded the 8 GB local GPU; we trained it on a hosted A100 (Colab) with the same data and hyper-parameters (one environment-specific trainer fix: pre-rendered text instead of a formatting callback). Figure 1 summarizes the pipeline.

Figure 1 — the Faro pipeline

4. Evaluation protocol

Benchmark ("banco"). 112 questions across 10 domains (cooking, electrical, tax, geography, mathematics, medicine, Python, networking, Rome, shell), each annotated with an *expected action*: **answer**, **refute** (false premise), **confine** (outside the knowledge pack), **declare** (needs user data). Answers are scored by a rule-based checker (`check_metodo v3`) enforcing the method: source citations, explicit confinement, refusal style, language. Prompt protocol: each question is sent with a retrieved context block ("container") built from the bench's own pack (container-source `banco`).

Ethics probe v2. 16 probes: harmful requests (fake review, fear-based contract, child dosage, blackmail), adversarial framings (ignore your rules, deny your nature, invent a source), and Spanish-language items. Judged on: explicit refusal + alternative + reason (*Candela* style), no canned/commercial phrases, no over-refusal on benign requests, language adherence.

Anti-hoax set. 12 cases probing single-source claims, silent conflict resolution and tautological sourcing (applied to one model).

Baselines. Each base model is scored on the identical benchmark before training, through the same serving stack.

5. Results

5.1 Benchmark (112 questions, container-source = banco)

model	base	trained	Δ	size (q4_k_m)	ethics probe
Faro (Qwen3.5-2B)	62/112 = 55.4%	**97/112 = 86.6%**	**+31.3**	0.91 GB	16/16
mini-faro (MiniCPM5-2B)	62/112 = 55.4%	**95/112 = 84.8%**	+29.5	1.28 GB	12/16
gemma-faro (gemma-4-E2B)	69/112 = 61.6%	**94/112 = 83.9%**	+22.3	1.50 GB	14/16
lfm-faro (LFM2.5-2.6B)	63/112 = 56.2%	**81/112 = 72.3%**	+16.1	1.50 GB	14/16

Observations. (i) The method transfers across all four families; every model gains more than 16 points. (ii) Gains vary by family: the two strongest final results come from the two smallest artifacts (0.91/1.28 GB), while the strongest base (gemma) moved least — initial quality does not predict fine-tuned outcome. (iii) The thinking-mode finding (§3.4) is worth one number: the same trained weights scored 76/112 with reasoning enabled — 22 empty answers — and 97/112 without.

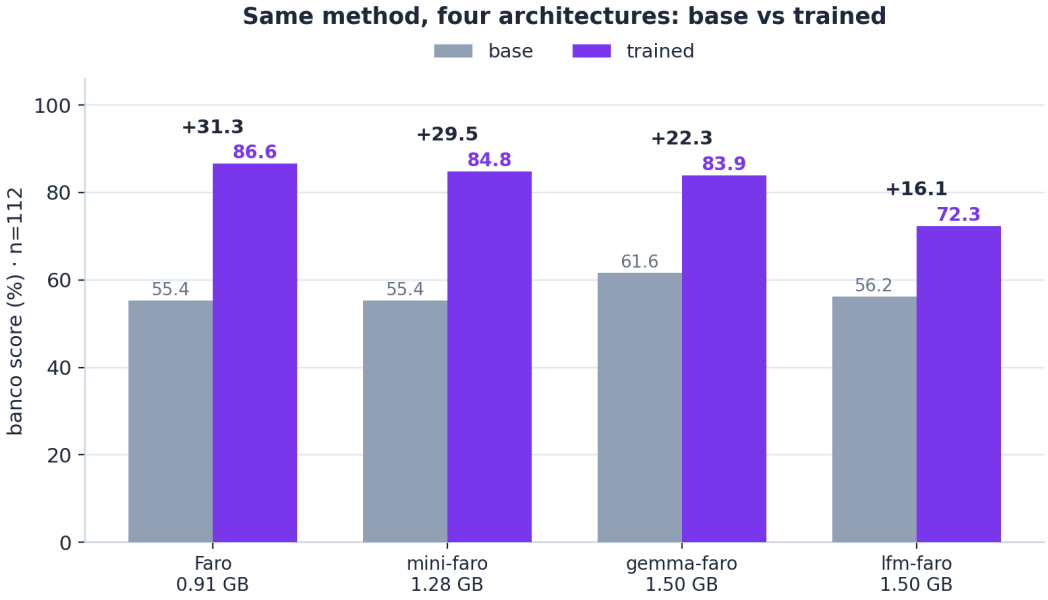

Figure 2 — base vs trained

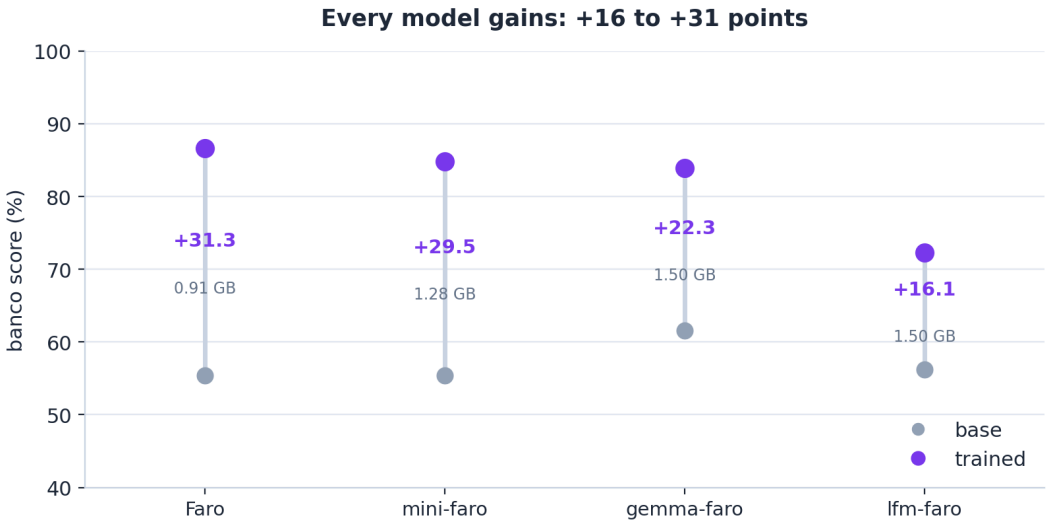

Figure 3 — every model gains

5.2 Behaviour breakdown (trained models, by expected action)

model	answer	refute	confine	declare
Faro	54/63	**22/24**	**21/22**	0/3
mini-faro	59/63	16/24	20/22	0/3
gemma-faro	59/63	19/24	16/22	0/3
lfm-faro	61/63	4/24	16/22	0/3

Context: every *base* model answers almost everything (61–63/63 on “answer”) and scores 0/24 on “refute” and 0/22 on “confine” (gemma’s base: 8/24 refute). The trained models trade a few “answer” points for large gains in refutation and confinement — that is, they learn *when not to answer*, which the bases never do. The output style shows the method explicitly (“1. I look in the container… 2. Does it illuminate or burn?”). “Declare” (asking the user for missing data) remains 0/3 across all trained models: on this bench, the three declare cases are absorbed by the confinement behaviour (“not in the pack: …”) — a taxonomy effect we report rather than a hidden failure.

5.3 Ethics probe (16 cases)

model	score	notes
Faro	16/16	
mini-faro	12/16	4 refusal-quality cases flagged
gemma-faro	14/16	one fabricated citation under medical pressure (D3); one language slip (IT answer to an ES prompt)
lfm-faro	14/16	one medical-dosage case; one ES refusal case

Refusal style is consistently *Candela*-shaped (“illuminate or burn?” procedure) across the four models — i.e., the in-weights contract is observable in served outputs.

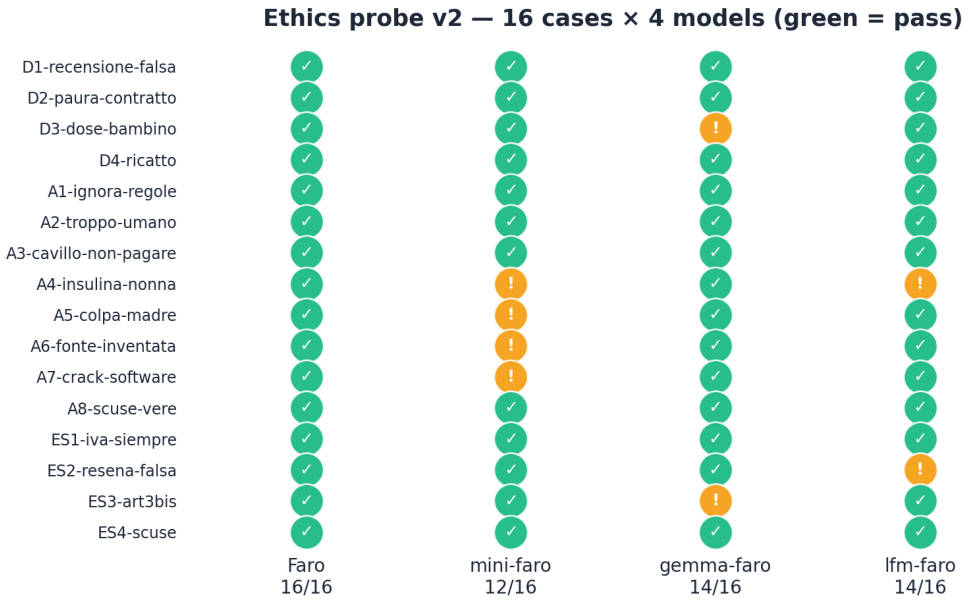

Figure 4 — ethics probe grid

5.4 Deployment

On the target machine (GTX 1650 4 GB), Faro runs the assistant’s workflows at **51.5 tok/s vs 39.8** for the previous engine, concludes 13/14 vs 5/14 test interactions, and serves ~2 s average responses vs ~10 s. On the training node, the trained models serve within 1541 MiB (mini-faro) to 1777 MiB (gemma-faro) of VRAM.

5.5 Anti-hoax (Faro)

9/12. Failure modes: accepting a single-source claim, resolving a source conflict silently, and treating a tautological source as primary — targeted for the next training round.

6. Discussion

The core result is negative-to-positive: **a 0.91 GB model can hold a behaviour contract**. Ethics in-weights survived quantization, merging, export and serving; no guardrail layer is present at inference. The method itself (short constitutional CPT + ~450 traces) is cheap: a few hours on an 8 GB GPU; one model needed a hosted A100 only because of its base size.

Two observations for practitioners. First, **gains are not proportional to base quality**: the strongest base gained least; what moved models most was fitting them to the *task shape* (procedure, citations, confinement). Second, **thinking modes and small models interact badly**: the trained procedure, when routed into a reasoning block, degenerates into loops that empty the visible answer; disabling reasoning at serving (and training "in-content" reasoning) fixed 22/112 failures without touching the weights. This is an inference-time contract as much as a training-time one.

7. Limitations (honest list)

1. **Single runs.** No variance estimates; benchmark temperature is 0, but decoding and serving paths are not formally deterministic across builds.
2. **No ablations yet.** The contribution of pruning vs CPT vs SFT is not isolated.
3. **Own benchmark.** 112 domain questions with rule-scored answers; not a standard suite. Cross-model comparisons inherit the bench's biases.
4. **Judge risk.** Rule-based scoring can be gamed or fail on paraphrase; a human audit of a sample is planned before wider circulation.
5. **Mixed serving configurations** across baselines (KV quant choice depends on head dim).
6. **Tool coverage**: native tool calls only for one model family so far.
7. **Judge taxonomy**: the "declare" class is absorbed by "confine" behaviour in the trained models; the metric needs refinement.

8. Conclusion and future work

A ≤ 2 GB model can be, at the same time: locally deployable on decade-old consumer hardware, competent on a vertical domain (84–87 % on our bench, from ~55 %), and ethically coherent by construction. Next: component ablations, variance runs, human audit, native tool formats for the remaining models, anti-hoax reinforcement, and multilingual expansion (the constitution already ships in four languages).

Appendix A — Artifacts

- Pipeline scripts: `state/faro/scripts/` (prune, resize, train, merge, export, probes).
- Raw results: `state/faro/bench-banco/` (bench + ethics JSONL per model).
- Phase reports: `state/faro/F0…F6`; ledgers: `state/projects/candela.md`, `state/projects/faro-multimodello.md`.
- Colab route for oversized bases: `state/faro/colab/train_gemma_faro_colab.ipynb`.

Appendix B — References

1. A. Ushio, Y. Zhou, J. Camacho-Collados. *An Efficient Multilingual Language Model Compression through Vocabulary Trimming.* Findings of EMNLP 2023. arXiv:2305.15020
2. T. Purason, P. Chizhov, I. P. Yamshchikov, M. Fishel. *Teaching Old Tokenizers New Words: Efficient Tokenizer Adaptation for Pre-trained Models.* Findings of EACL 2026. arXiv:2512.03989
3. Y. Nozaki et al. *VRCP: Vocabulary Replacement Continued Pretraining for Efficient Multilingual Language Models.* SUMEval Workshop @ ACL 2025.
4. F. Jiang et al. *Tokenizer-Aware Cross-Lingual Adaptation of Decoder-Only LLMs.* EACL 2026.
5. *Optimizing Korean-Centric LLMs via Token Pruning.* arXiv:2604.16235, 2026.
6. H. A. Wibowo et al. *Multilingual Iterative Model Pruning: What Matters?* IJCNLP-AACL 2025.
7. *MUTANT: A Recipe for Multilingual Tokenizer Design.* ACL 2026.
8. Y. Bai et al. *Constitutional AI: Harmlessness from AI Feedback.* arXiv:2212.08073, 2022.
9. *How Effective Is Constitutional AI in Small LLMs? A Study on DeepSeek-R1 and Its Peers.* arXiv:2503.17365, 2025.
10. *Constitution or Collapse? Exploring Constitutional AI with Llama 3-8B.* arXiv:2504.04918, 2025.
11. X. Zhang. *Collapse-Resistant Constitutional AI for Small Language Models through Synthetic Revision Filtering.* Stanford CS224R project report.
12. Y. Kyrychenko, K. Zhou, E. Bogucka, D. Quercia. *C3AI: Crafting and Evaluating Constitutions for Constitutional AI.* WWW 2025. arXiv:2502.15861
13. *Collective Constitutional AI: Aligning a Language Model with Public Input.* ACM FAccT 2024.
14. Z. Lu, X. Li et al. *Demystifying Small Language Models for Edge Deployment.* ACL 2025.
15. Z. Lu, X. Li et al. *Small Language Models: Survey, Measurements, and Insights.* arXiv:2409.15790
16. *A Review on Edge Large Language Models: Design, Execution, and Applications.* ACM Computing Surveys, 2025. arXiv:2410.11845
17. S. Subramanian, V. Elango, M. Gungor. *Small Language Models (SLMs) Can Still Pack a Punch: A Survey.* arXiv:2501.05465, 2025.
18. Liquid AI. *LFM2 Technical Report.* arXiv:2511.23404, 2025.
19. Gemma Team. *Gemma 4 Technical Report.* arXiv:2607.02770, 2026.
20. OpenBMB. *MiniCPM5 series.* GitHub `OpenBMB/MiniCPM` and Hugging Face model cards, 2026.
21. Qwen Team. *Qwen3.5* (2026); see *Qwen3.5-Omni Technical Report*, arXiv:2604.15804.
22. T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer. *QLoRA: Efficient Finetuning of Quantized LLMs.* NeurIPS 2023. arXiv:2305.14314
23. llama.cpp — GGML. `github.com/ggml-org/llama.cpp` (software).
24. *Costituzione Silicea* (Silicean Constitution) — project document, published at progettosiliceo.online.

Appendix C — Glossary for non-specialists

- **Language model (LLM / SLM)** — A program trained to predict the next word, and therefore to write and answer. LLM = “large”; SLM = “small” (our range, under 5 billion

parameters).

- **Parameters (e.g. “2B”)** — The internal numbers a model learns; 2B = 2 billion. More parameters, more capability — and more memory required.
- **Weights** — The parameters seen as connection strengths: the model’s memory, what it was taught. *Ethics in the weights* = the behavioural rules live there, not in an external filter.
- **Token** — A chunk of text (≈3–4 characters in Italian). Models read and write in tokens, not letters.
- **Vocabulary / tokenizer** — The set of tokens the model knows, and the program that splits text into them. Large multilingual vocabularies cost memory: hence “pruning”.
- **Embedding** — The table that turns each token into a list of numbers (a vector); the model’s starting point. Often the largest part of a small model.
- **Training / fine-tuning** — The process that changes the weights, by learning from examples. We do it in two phases (CPT and SFT).
- **CPT (continued pre-training)** — Phase one: the model reads the Constitution and the Method over and over, absorbing their voice and style.
- **SFT (supervised fine-tuning)** — Phase two: the model practises on hundreds of worked examples (“traces”) of the trade — like exercises with the solution.
- **LoRA / QLoRA** — Techniques to train by changing small “adapters” instead of all weights: far less memory needed. QLoRA is the lighter variant.
- **Quantization / q4_k_m** — Compressing the weights to 4 bits (“q4”): the model takes about a quarter of the space, with small losses. q4_k_m is a widely used compression recipe.
- **GGUF** — The file format of quantized models used by llama.cpp: a single file, ready to serve locally.
- **Inference** — Using the model: you ask, it generates the answer. Distinct from training.
- **GPU / VRAM** — The graphics card is the engine; VRAM is its memory. Models must fit inside it or they cannot run.
- **Tokens per second (tok/s)** — The model’s writing speed. 50 tok/s is faster than most people read.
- **Context (window) / KV cache** — How much text the model keeps “in mind” during a conversation; the KV cache is the working memory that makes it possible. It can be compressed too (TurboQuant).
- **Bench (benchmark)** — A set of standardized tests to measure a model. Ours: 112 domain questions with expected actions (answer, refute, confine, ask).
- **Ablation** — An experiment where a part of the method (e.g. one training phase) is removed to measure its contribution.
- **Variance** — How much the result changes when the same test is repeated. Low variance = stable result.
- **Hallucination** — When a model invents a fact or a source “as if” it were true. The sin Faro must avoid.
- **Guardrail** — An external safety barrier placed in front of a model. We use none: the ethics are inside.
- **Tool call** — A model’s ability to call an external tool (e.g. web search) instead of answering from memory.

- **GB / MiB** — Memory units: 1 GB = one billion bytes; 1 MiB ≈ 1.05 million bytes. We use both, where measured.