Good Enough? Quality, Latency, and Parameter Sensitivity of Quantized 7–9B LLMs on Consumer Hardware

Southin, J. E. A.

13 April 2026

Abstract

Quantised open-weights language models in the 7–9 billion parameter range can now run on consumer laptops, but practitioners lack the cross-cutting evaluations needed to choose a model, a quantisation level, and sampling parameters for a given quality–latency budget. Existing benchmarks test quality or speed or parameter effects in isolation, and most omit chain-of-thought reasoning, the task type where local models are expected to diverge most from frontier systems. We present a controlled study comparing five open-weights model families (Llama 3.1 8B, Gemma 4 E4B, GLM-4-9B, Qwen 2.5 7B, and DeepSeek-R1-Distill-Qwen-7B) at selected quantisation levels, against a Claude Opus 4.6 frontier baseline, on 16 context-grounded question–answer pairs spanning factual extraction, reasoning, synthesis, and chain-of-thought. A full temperature × top-p sweep yields over 1,500 scored responses, each evaluated blind by two independent judges (Claude Opus 4.6 and Gemini 2.5) on accuracy, completeness, and coherence. We find that (1) quantised 7–9B models achieve 93–96% of frontier quality on extractive Q&A but collapse to 55–75% of frontier quality on chain-of-thought, revealing a task-dependent rather than uniform capability gap; (2) Qwen 2.5 7B displaces Llama 3.1 8B as the Pareto-optimal local model; (3) a mild temperature ($t = 0.3$) outperforms greedy decoding for GLM-4-9B and Gemma 4 E4B Q8 at statistical significance, while top-p has no measurable effect; (4) Q8 quantisation provides no quality benefit over Q4 within the 8B class; (5) reasoning-distilled models (DeepSeek-R1-Distill) underperform on extractive tasks due to verbose chain-of-thought outputs; and (6) self-preference bias from using Opus as both subject and judge is empirically negligible on extractive Q&A (mean bias +0.04 on completeness) and small on chain-of-thought (+0.19), validating the original rankings under independent evaluation. We release a seven-step replication methodology, a model selection guide, the evaluation harness, question set, and complete results.

Keywords: local LLM inference; quantisation; sampling parameters; chain-of-thought; LLM-as-judge; Apple Silicon; consumer hardware; Qwen 2.5; Llama 3.1; Gemma 4; DeepSeek-R1.

1. Introduction

Running a large language model locally (on a laptop, with no API calls, no data leaving the device) has moved from a hobbyist curiosity to a practical option. Quantised open-weights models in the 7–9 billion parameter range fit in 5–8 GB of memory, run at interactive speeds on Apple Silicon via llama.cpp [1], and produce coherent, often accurate responses to factual questions. But a developer considering local deployment faces a question the existing literature cannot answer cleanly: *which model, at which quantisation, with which sampling parameters, gives the best quality per second on my hardware?*

The evidence is siloed. Quantisation benchmarks [2], [3], [4] report scores on standard tasks (MMLU, GSM8K) at fixed sampling parameters. Temperature studies [5], [6] test full-precision models, not quantised ones. Inference framework comparisons [7], [8] report throughput and latency but not output quality. LLM-as-judge papers [9], [10] establish methodology but do not apply it to local model evaluation. And most evaluations of local models omit chain-of-thought reasoning, the task type where local and frontier models are expected to diverge most [11]. No published study maps quality, latency, and parameter sensitivity simultaneously for multiple model families on consumer hardware across both extractive and chain-of-thought task types.

We address this gap with a controlled evaluation spanning five open-weights model families (Llama

3.1 8B, Gemma 4 E4B, GLM-4-9B, Qwen 2.5 7B, and DeepSeek-R1-Distill-Qwen-7B) at selected quantisation levels, a Claude Opus 4.6 frontier baseline, and a full temperature × top-p parameter sweep, all on a single Apple M2 MacBook with 16 GB of memory. The task is context-grounded Q&A across 16 hand-crafted items covering factual extraction, reasoning, synthesis, and chain-of-thought, scored blind on accuracy, completeness, and coherence by two independent judges (Claude Opus 4.6 and Gemini 2.5).

We make six contributions:

1. **A cross-cutting quality–latency–parameter evaluation** of quantised small models on consumer hardware, the first to map all three axes simultaneously across multiple model families. Qwen 2.5 7B at Q4_K_M is the Pareto-optimal local choice: 94% of frontier quality on extractive Q&A at 6.8 seconds per response.
2. **The first demonstration that the frontier–local gap is task-dependent, not uniform.** Local models reach 93–96% of frontier quality on extractive Q&A but only 55–75% on chain-of-thought. Opus scores approximately 5.0 across categories; the best local model falls from 4.7 (extractive) to 3.4 (chain-of-thought). Aggregate "local vs frontier" claims conceal this structure.
3. **Evidence that mild temperature outperforms greedy decoding** on factual Q&A for GLM-4-9B and Gemma 4 E4B Q8 at statistical significance ($p < 0.05$), extending the temperature findings of Renze and Guven [5] from full-precision large models to quantised small models. Top-p has no measurable effect, and Gemma 4 E4B Q4 shows a small non-significant preference for greedy, suggesting quantisation-dependent temperature sensitivity within a model family.
4. **A quantisation comparison** showing that Q8 precision provides no quality benefit over Q4 within the 8B class. Gemma's accuracy advantage over Llama is architectural (Per-Layer Embedding design), not quantisation-dependent.
5. **Evidence that reasoning-distilled models misfire on extractive tasks.** DeepSeek-R1-Distill-Qwen-7B, despite strong reasoning credentials, produces the lowest accuracy of any local model tested on extractive Q&A (4.33/5.00) because its training pushes it to generate long reasoning chains even for simple factual questions.
6. **Empirical validation that self-preference bias is negligible** for our evaluation design. By re-judging every response with an independent second judge (Gemini 2.5), we measure Opus's self-preference bias directly: +0.04 on extractive completeness, +0.19 on chain-of-thought completeness, with inter-judge Pearson agreement of 0.67–0.84 across dimensions. Model rankings are robust to judge choice, with one meaningful exception (Qwen vs Llama on chain-of-thought).

Several of our findings are confirmatory rather than novel. That quantised 8B models produce reasonable outputs is well-established [2], [12]. That temperature has limited effect on factual tasks confirms Renze and Guven [5]. That LLM judges have self-preference bias is documented [10]. We present these results as validation: they confirm that patterns observed on full-precision large models and standard benchmarks hold in the specific setting of quantised small models on consumer hardware running context-grounded Q&A, a combination not previously tested directly.

We provide a generalised seven-step methodology (Section 3.7) for replicating this evaluation on other hardware configurations, and we release the evaluation harness, question set, and complete results.

2. Related Work

Our study sits at the intersection of four research threads: model quantisation for small language models, sampling parameter effects on output quality, LLM-as-judge evaluation methodology, and edge inference on consumer hardware. Each thread is individually well-studied; the gaps lie at their intersections.

2.1 Quantisation and Small Language Models

Post-training quantisation has become the primary mechanism for deploying large models on constrained hardware. The foundational methods (LLM.int8() [13], GPTQ [14], SmoothQuant [15], and

AWQ [16]) established that 4-bit and 8-bit quantisation can preserve model quality while dramatically reducing memory footprint. A comprehensive survey of low-bit LLM methods [17] covers the current algorithmic landscape.

However, recent work reveals that quantisation behaves differently at smaller scales. SLMQuant [2] is the first systematic benchmark for small language model quantisation, finding that "direct transfer of LLM-optimised techniques leads to suboptimal results" due to fundamental disparities in quantisation sensitivity between small and large models. The IJCAI 2025 study [18] corroborates this: smaller models at higher bitwidths (e.g., 7B at Q4) outperform larger models at extreme quantisation (e.g., 65B at 2-bit), suggesting that practitioners should favour moderate compression of appropriately-sized models over aggressive compression of larger ones.

For the specific model–format combination most relevant to our work, the unified llama.cpp quantisation evaluation [3] tested every GGUF format on Llama-3.1-8B-Instruct across GSM8K, HellaSwag, IFEval, MMLU, and TruthfulQA, confirming Q4_K_M as the optimal quality–size tradeoff. Red Hat's large-scale study [12] (over 500,000 evaluations) found that 8B models "experience slight variability when quantized but preserve core semantic meaning."

The Small Language Models survey [19] benchmarks 57 SLMs on commonsense reasoning, maths, in-context learning, and runtime costs, while the ACL 2024 evaluation [4] proposes a three-dimensional framework covering knowledge, alignment, and efficiency.

Gap. These benchmarks evaluate quantised models on standard tasks (MMLU, GSM8K, HumanEval) at fixed sampling parameters. Context-grounded Q&A, where the answer must be extracted or synthesised from a provided passage (the dominant use case in RAG applications), is absent as a benchmark task for quantised small models.

2.2 Sampling Parameters and Decoding Strategies

Nucleus sampling, introduced by Holtzman et al. [20], established top-p as the dominant alternative to greedy decoding by showing that maximisation-based approaches produce degenerate text. Subsequent work has refined the decoding landscape: Wiher et al. [21] demonstrate that decoding effects are task-dependent rather than universally optimal, Hewitt et al. [22] analyse top-p as a desmoothing operator and propose eta-sampling, and min-p sampling [23] offers an alternative truncation strategy benchmarked against standard methods.

On the specific question of temperature's effect on output quality, Renze and Guven [5] provide the most directly relevant result: temperature changes from 0.0 to 1.0 "do not have a statistically significant impact on LLM performance for problem-solving tasks," with results that "appear to generalise across LLMs, prompt-engineering techniques, and problem domains." Their companion paper [24] argues that benchmarks using only greedy decoding miss important variance information, a methodological critique that motivates parameter sweep evaluations.

The "Hot or Cold?" study [6] adds a size dimension, introducing the "mutation temperature" concept and showing that small models (1–4B) are more sensitive to temperature than large models (40–80B). Clinical studies [25] confirm that large models (GPT-4, Llama-3-70b) maintain consistent performance across temperature settings. Dynamic and contextual temperature approaches [26], [27] propose learned per-token temperature adaptation.

Gap. Temperature studies test full-precision models at single sizes, not quantised models across families. Whether quantisation shifts the mutation temperature threshold, and whether different model families exhibit different parameter sensitivity at matched parameter counts, is unstudied.

2.3 LLM-as-Judge Methodology

The LLM-as-judge paradigm was formalised by Zheng et al. [9], who demonstrate >80% agreement between GPT-4 judge scores and human preferences on MT-Bench. Two comprehensive surveys [28], [29] and a 2026 journal survey [30] cover the current state of the methodology.

Subsequent work has documented systematic biases. Position bias, where judges favour responses in specific positions, has been characterised by Wang et al. [31] and Wang et al. [32]. The Judge Relia-

bility Harness [33] provides a stress-testing framework probing robustness to paraphrase, formatting, and stochastic variation.

Most relevant to our design is **self-preference bias**. Panickssery et al. [10] demonstrate at NeurIPS 2024 that LLM evaluators recognise and favour their own outputs, with a linear correlation between self-recognition capability and bias strength. The mechanism is perplexity-based: models score text matching their own distribution higher. This implies that using Opus as both subject and judge will systematically overrate Opus's outputs.

Design choices matter for reliability. The empirical study by [34] finds that providing reference answers and score descriptions is crucial, that non-deterministic sampling improves human alignment, and that chain-of-thought reasoning offers minimal gains when evaluation criteria are clear. Verbosity bias (judges preferring longer responses independent of quality [35], [36]) is an additional confound when models produce substantially different output lengths.

Gap. Self-preference bias has been demonstrated for GPT-4o and Claude 3.5 Sonnet on subjective tasks. Whether the bias is attenuated for factual Q&A, where correctness is verifiable against a passage rather than distribution-dependent, is untested.

2.4 Edge Inference on Consumer Hardware

Apple Silicon inference has been comprehensively benchmarked. The production-grade comparison [7] tests five frameworks (MLX, MLC-LLM, Ollama, llama.cpp, PyTorch MPS), finding MLX achieves ~230 tok/s versus llama.cpp at ~150 tok/s. The ACM SIGMETRICS characterisation [8] analyses memory bandwidth and GPU utilisation, and the native Apple Silicon study [37] evaluates across M2 Ultra, M2 Max, and M4 Pro. RooflineBench [38] proposes a systems-level framework for understanding on-device inference bottlenecks.

Apple's own Core ML deployment of Llama 3.1 [39] provides a first-party reference for on-device performance, and the mobile platforms evaluation [40] adds power and thermal data. Epoch AI reports [41], [42] contextualise the broader trend: open-weights models trail the frontier by approximately three months, and frontier capabilities reach consumer hardware within roughly one year.

Gap. Inference benchmarks report throughput and latency; quality benchmarks report scores. No published study maps both axes simultaneously to produce a quality-vs-latency Pareto frontier for multiple model families on the same consumer hardware: the chart a practitioner needs to make a deployment decision.

3. Experimental Setup

This section describes the models, hardware, evaluation task, parameter sweep, judging protocol, and frontier baseline. Section 3.7 generalises the methodology for replication on other hardware configurations.

3.1 Models

We evaluate five open-weights model families at the 7–9B parameter scale, selected according to four criteria: (1) text-to-text architecture (no multimodal or code-specialist models), (2) availability in GGUF format with community-maintained quantisations, (3) feasibility on 16 GB unified memory at Q4 or Q8 precision, and (4) diversity of architectural lineage, training origin, and training objective (general-purpose vs. reasoning-specialised). All local models are served via llama.cpp [1].

Why these five families. Llama 3.1 8B is the most widely deployed open-weights model in the 8B class, with the largest GGUF ecosystem and the most extensive community benchmarking; it serves as the natural baseline. Gemma 4 E4B represents a structurally distinct architecture (Google's Per-Layer Embedding design) yielding 8B total parameters but only 4B "effective" parameters, making it the smallest model in actual compute while matching the others in nominal parameter count. GLM-4-9B-Chat is the strongest bilingual (Chinese–English) model in this parameter range and originates from a non-Western research lab (Zhipu AI), providing architectural and training-data diversity. Qwen 2.5 7B (Alibaba) is a leading general-purpose 7B model at the time of evaluation, consistently ranked at or near

Table 1: Model configurations. All GGUF models sourced from the `bartowski` HuggingFace namespace; Opus accessed via the Anthropic API.

Model	Family	Params	Quant	Size
Llama 3.1 8B Instruct	Llama (Meta)	8B	Q4_K_M	4.7 GB
Llama 3.1 8B Instruct	Llama (Meta)	8B	Q5_K_M	5.5 GB
Gemma 4 E4B-it	Gemma (Google)	8B eff.	Q4_K_M	5.2 GB
Gemma 4 E4B-it	Gemma (Google)	8B eff.	Q8_0	8.0 GB
GLM-4-9B-Chat	GLM (Zhipu AI)	9B	Q4_K_M	5.5 GB
Qwen 2.5 7B Instruct	Qwen (Alibaba)	7B	Q4_K_M	4.5 GB
DeepSeek-R1-Distill-Qwen-7B	DeepSeek	7B	Q4_K_M	4.5 GB
Claude Opus 4.6	Claude (Anthropic)	n/a	n/a	API

the top of open benchmarks in the 7B class; it tests whether a smaller, better-trained model beats the larger 8–9B models in our sweep. DeepSeek-R1-Distill-Qwen-7B [11] is a reasoning-specialised variant of Qwen fine-tuned on DeepSeek-R1 chain-of-thought traces; it tests whether reasoning distillation transfers to our evaluation task.

Why not other candidates. Several models were considered and excluded. Llama 4 Scout (17B active parameters across 16 MoE experts) exceeds the 16 GB memory budget at any useful quantisation; its Q4 GGUF is approximately 24 GB. Llama 3.3 has no 8B variant (only 70B). Mistral 7B is a viable alternative but was omitted to keep the model roster tractable given the sweep cost. Phi-3 and StableLM were excluded as code- and instruction-specialist models respectively, as our evaluation focuses on general text-to-text Q&A from provided context.

Quantisation levels. The quantisation levels are deliberately asymmetric: Gemma runs at Q8_0 (8-bit) while Llama and GLM run at Q4_K_M (4-bit). This serves two purposes. First, it tests whether higher quantisation precision yields measurably better output quality at the cost of increased memory and latency. Second, it reflects a realistic practitioner scenario: Gemma's Per-Layer Embedding architecture makes Q8 feasible within the same 16 GB memory envelope that constrains the other models to Q4.

Frontier baseline. Claude Opus 4.6, accessed via the Anthropic API, serves as the frontier reference. It receives the same prompts and parameter combinations as the local models to enable direct comparison under identical conditions.

3.2 Hardware

All local inference runs on a single Apple MacBook with an M2 system-on-chip and 16 GB of unified memory. The unified memory architecture allows the CPU and GPU to share the same physical memory pool without copy overhead, a property that recent benchmarks have shown provides competitive inference throughput for models that fit entirely in memory [7], [8].

llama.cpp is configured with full GPU offload (`-ngl 99`) to maximise Metal acceleration. Only one model is loaded at a time (the 16 GB memory budget does not permit concurrent model serving). The context window is set to 4,096 tokens for all local models, sufficient for the evaluation passages (all under 1,000 tokens) plus the system prompt and generated response.

No thermal management was applied. The MacBook Air M2 uses passive (fanless) cooling, and sustained inference workloads on passively-cooled hardware are known to trigger thermal throttling after extended runs. We did not measure or control for this effect, and latency measurements should be interpreted as representative averages rather than guaranteed throughput under sustained load.

3.3 Evaluation Task

We construct a set of 16 context-grounded question–answer pairs, evenly distributed across four task categories:

- **Factual extraction** (4 items): questions with answers directly stated in the passage (e.g., *"How*

many parish churches were destroyed in the Great Fire of London?").

- **Reasoning** (4 items): questions requiring causal inference or argument evaluation from the provided evidence (e.g., *"Why did customers start claiming they would 'take out' their food at convenience stores after the tax change?"*).
- **Synthesis** (4 items): questions requiring integration of multiple facts into a coherent summary (e.g., *"Summarise the trade-offs of the shift to remote work, covering both the benefits and the costs."*).
- **Chain-of-thought** (4 items): questions requiring multi-step reasoning with intermediate computation or case analysis (e.g., multi-step arithmetic over financial figures, ordering a causal chain to identify a root cause, comparative evaluation of two approaches, and counterfactual inference).

The chain-of-thought (CoT) category is included to probe a dimension the other three categories do not directly test: the ability to decompose a problem, sequence reasoning steps, and maintain logical consistency across them. Unlike the other categories, where short and direct answers are rewarded, CoT items are expected to elicit longer outputs that demonstrate working through the problem. We score the final answer rather than the intermediate chain, but we record token counts to observe how reasoning length varies across model families.

Each item comprises a context passage (150–250 words of factual text from diverse domains: history, biology, economics, technology, environmental science, urban policy, finance, and ecology), a question, and a reference answer. All context passages contain sufficient information to answer the question fully; no external knowledge is required or desired.

We use hand-crafted items rather than an established benchmark (e.g., SQuAD, Natural Questions, CRAG [43]) for three reasons. First, the evaluation must be computationally tractable on a single consumer machine, including model setup, full parameter sweeps, and judge scoring. Second, hand-crafted items allow precise control over difficulty, domain diversity, and reference answer quality. Third, the primary goal is comparative ranking across models and parameter settings rather than absolute performance measurement; relative differences are robust to test set choice provided the items span a range of difficulty.

The limitation is clear: 16 items cannot support fine-grained statistical claims about individual questions. We compensate by running each item across 12 parameter combinations per model (Section 3.4), yielding up to 192 observations per model (144 on the 12 extractive items plus 48 on the 4 CoT items). Appendix D provides confidence intervals and effect size analysis.

The system prompt for all models (local and frontier) is identical:

> *"Answer the question based only on the provided context. Be concise and accurate."*

The user message follows the template:

> *"Context: {passage}* `\n\n` *Question: {question}"*

3.4 Parameter Sweep

We sweep two sampling parameters in a full factorial design:

Table 2: Parameter sweep design.

Parameter	Values	Rationale
Temperature	0.0, 0.3, 0.7, 1.0	Covers greedy decoding (0.0), mild sampling (0.3), moderate (0.7), and high randomness (1.0)
Top-p	0.5, 0.9, 1.0	Covers restrictive nucleus (0.5), standard (0.9), and unrestricted (1.0)

This produces $4 \times 3 = 12$ parameter combinations per model. Each combination is applied to all

16 evaluation items, yielding $12 \times 16 = 192$ inference calls per model in the fullest configuration. In practice the sweep was conducted in three phases: an initial run of the four primary models (Llama Q4, Gemma Q8, GLM Q4, Opus) on the 12 extractive items; a quantisation follow-up adding Llama Q5 and Gemma Q4 on the same 12 items; and a new-models round adding Qwen 2.5 7B, DeepSeek-R1-Distill 7B, and the 4 CoT items across all models. In aggregate the evaluation comprises over 1,500 scored responses, each with a latency measurement and a token count. See the supplementary materials for the full run manifest.

The sweep is exhaustive rather than adaptive. We do not prune unpromising parameter regions mid-run, as the goal is to characterise the full response surface rather than find an optimum. At temperature 0.0, top-p has no effect (the maximum-probability token is always selected regardless of nucleus size); we include these redundant combinations for completeness and to verify this expectation empirically.

The sweep design is motivated by Renze and Guven's [5] finding that temperature changes from 0.0 to 1.0 do not have a statistically significant effect on problem-solving performance, and by Renze and Guven [24] arguing that benchmarks using only greedy decoding miss important variance information. Our sweep explicitly tests whether these findings (established on full-precision large models) hold for quantised small models on consumer hardware.

3.5 Evaluation Protocol

All responses are scored independently by two automated judges in blind evaluation: each judge receives the context, question, reference answer, and candidate response, but not the identity of the model that produced it. The primary judge is Claude Opus 4.6 (Anthropic). The secondary judge is Gemini 2.5 Pro on the original parameter sweep and Gemini 2.5 Flash on the quantisation, new-models, and CoT sub-runs (Pro became impractical at scale due to rate-limit constraints). All headline results in this paper use the Opus scores; the Gemini scores serve as an independent validation channel reported in Section 4.5.

Each judge scores each response on three dimensions using a 1–5 Likert scale:

- **Accuracy**: Is the answer factually correct given the context?
- **Completeness**: Does it address all parts of the question?
- **Coherence**: Is it well-structured and easy to understand?

The judge is prompted to return structured JSON:

```
{"accuracy": N, "completeness": N, "coherence": N, "reasoning": "..."}
```

This design follows established LLM-as-judge methodology [9], which demonstrates >80% agreement with human preferences when using strong frontier models as evaluators with explicit scoring rubrics.

Self-preference bias and the multi-judge design. Using Opus as both a subject and a judge introduces a known methodological concern. Panickssery et al. [10] demonstrate that LLM evaluators recognise and favour their own outputs, with a linear correlation between self-recognition capability and self-preference bias strength. The root mechanism is perplexity-based: models score text matching their own training distribution as higher quality. We adopt four mitigations, the last of which is the strongest:

1. **Blind evaluation**: the judge never sees model names or identifiers.
2. **Reference answers**: providing an independent reference answer has been shown to significantly improve judge reliability and reduce distribution-dependent scoring [34].
3. **Factual verifiability**: our task type (context-grounded Q&A with verifiable answers) is less susceptible to distributional preference than subjective evaluation tasks, because correctness can be assessed against the passage rather than relying on stylistic preference.
4. **Independent second judge**: every response is independently re-scored by Gemini 2.5 using the same blind rubric. This allows direct measurement of self-preference bias by comparing Opus's scores of Opus outputs against Gemini's scores of the same Opus outputs. Section 4.5 reports the result: the bias is empirically negligible on extractive tasks and small on CoT.

Verbosity bias. LLM judges have been shown to prefer longer responses independent of quality [35], [36]. This may affect scoring of models that generate substantially different token counts (Section 4.6). We record token counts for all responses to enable post-hoc analysis of length–score correlations.

3.6 Frontier Baseline

Claude Opus 4.6 is evaluated under the same conditions as local models: the same 16 items, the same 12 parameter combinations, the same system and user prompts. The only difference is that inference occurs via the Anthropic API rather than local llama.cpp, and we do not control the hardware or quantisation on Anthropic's side.

This creates a methodological tension when Opus also acts as judge. We acknowledge this as a limitation rather than a feature, and address it empirically through an independent second judge (Section 4.5); the primary comparison of interest is among the five local models, where neither judge is the subject. The frontier scores serve as an upper-bound reference point rather than a fair competition.

In practice, Opus scored itself near-perfectly on the extractive items: 5.00/5.00 on accuracy and coherence across all 144 extractive evaluations, and 4.96/5.00 on completeness (a single synthesis question at temperature 0.0, top-p 0.5 received 4/5 on completeness). On chain-of-thought, Opus dropped to 4.73 accuracy and 4.88 completeness across 48 evaluations, showing that even the frontier baseline is not at ceiling for multi-step reasoning. Section 4.5 confirms via the independent Gemini judge that Opus's self-scores are not materially inflated by self-preference bias.

3.7 Generalising the Method

The experimental design is hardware-agnostic. We present it as a general recipe for evaluating local LLM inference on any consumer device.

Step 1: Select models. Choose N 2 models from different architectural families at a matched parameter count that fits the target device's memory. For unified memory systems (Apple Silicon, integrated GPUs), the rule of thumb is: GGUF model size in GB should be 60% of total memory to leave room for the context window, KV cache, and operating system overhead.

Step 2: Choose quantisation levels. Test at least two quantisation levels per model (e.g., Q4_K_M and Q8_0) to isolate quantisation effects from architectural differences. GGUF via llama.cpp is the most portable format; MLX is an alternative for Apple Silicon.

Step 3: Build an evaluation set. Construct 10–20 context-grounded question–answer pairs spanning at least two task types (e.g., factual extraction and reasoning). Each item requires: a self-contained context passage, a question answerable from the passage alone, and a reference answer. Established benchmarks (CRAG [43], GaRAGe [44], SQuAD) can substitute for hand-crafted items if time permits, with the advantage of standardised difficulty calibration.

Step 4: Define a parameter grid. Sweep temperature at a minimum of three values spanning greedy decoding to high-entropy sampling (e.g., 0.0, 0.3, 1.0). Include top-p only if the task type may be sensitive to nucleus size (creative generation, code); for factual Q&A, top-p can be held constant at 1.0 without loss.

Step 5: Configure an automated judge. Use a frontier model via API with blind evaluation, structured JSON scoring, and a reference answer in the prompt. For studies where the judge model is also a subject, clearly report this and discuss self-preference bias [10]. Using multiple judges from different model families (e.g., Claude + GPT-4o + Gemini) provides a more robust evaluation at the cost of increased API spend.

Step 6: Record quality, latency, and token count jointly. Quality-only or speed-only benchmarks leave practitioners unable to make tradeoff decisions. A quality-vs-latency scatter plot (Section 4.2) is the minimum viable deliverable for a practitioner-oriented evaluation.

Step 7: Report variance, not just means. Run each parameter combination on all evaluation items and report standard deviations or confidence intervals. Temperature effects are often invisible in means but substantial in variance (Section 4.1). A sweep that reports only average scores at a single parameter setting will miss this.

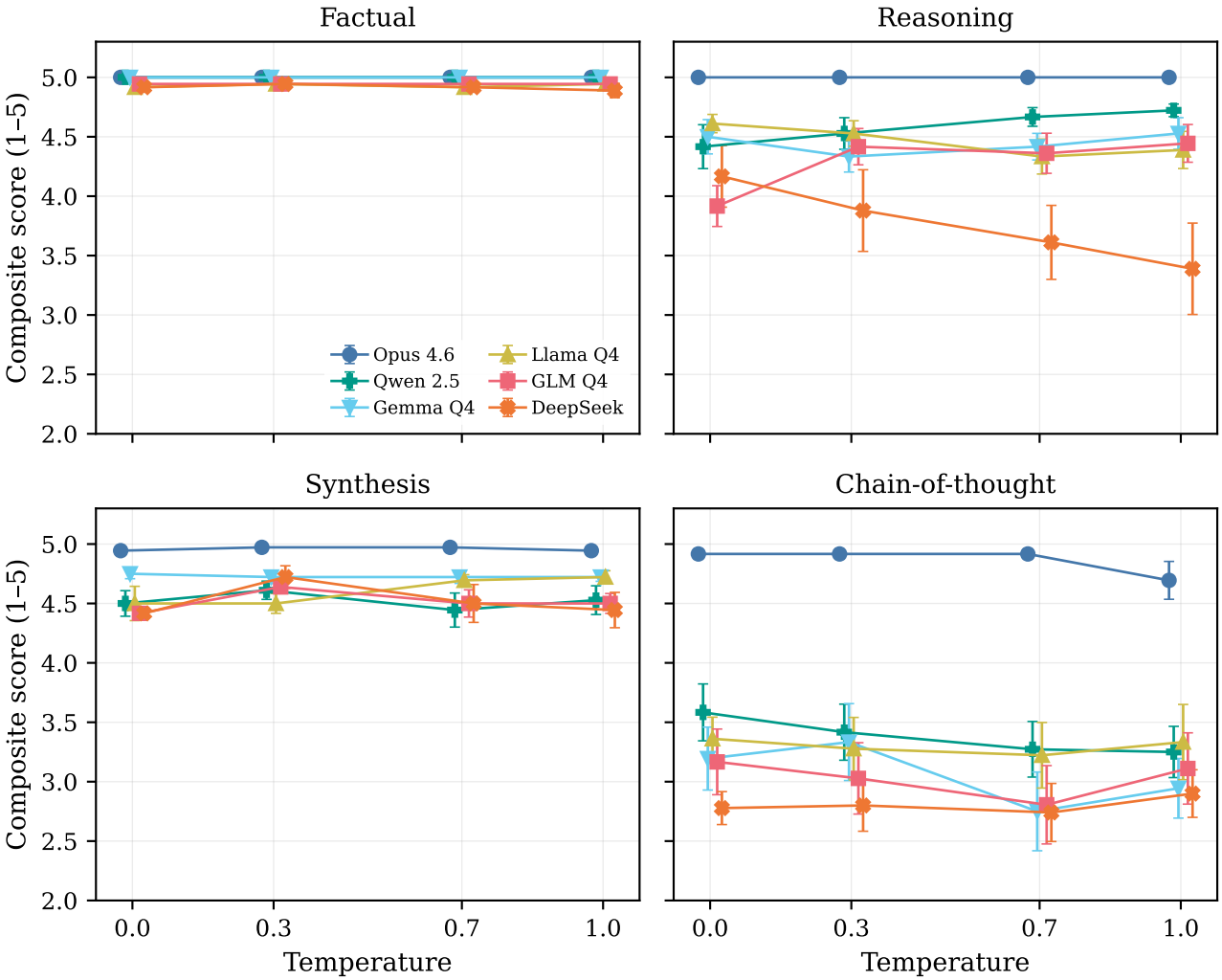

Figure 1: Composite score vs temperature by task category, for one representative configuration per model family (Q4 variants where applicable; the Q5 and Q8 variants tracked by Section 4.8 follow their Q4 counterparts within noise and are omitted from this view to reduce visual clutter). Error bars show ±1 SE. Factual questions are near-ceiling for all models. Reasoning and synthesis show clear model separation, with DeepSeek degrading at higher temperatures on reasoning. Chain-of-thought exposes a large frontier–local gap (Opus approximately 5.0; local models 2.7–3.5) with DeepSeek consistently weakest.

The evaluation harness, model configurations, and the full question set are available at https://github.com/joe-southin/local-lm.

4. Results

We present results across more than 1,500 scored responses spanning seven local model configurations (five model families, with two quantisation levels each for Llama and Gemma) and the Claude Opus 4.6 frontier baseline, evaluated on 16 items across four task categories (factual extraction, reasoning, synthesis, and chain-of-thought). We organise the results around the most informative findings first: the chain-of-thought gap, the quality–latency frontier, and per-family parameter sensitivity.

4.1 Sampling Parameter Sensitivity

4.1.1 Temperature effects

Figure 1 plots mean composite score (average of accuracy, completeness, and coherence) against temperature for each model, split by task category. Error bars show ±1 standard error.

The central finding is that **temperature has minimal effect on mean quality but substantially increases variance** across the extractive categories (factual, reasoning, synthesis). All models main-

tain relatively flat mean score curves from $t = 0.0$ to $t = 1.0$; the largest mean shift for any model is 0.3 points (GLM, from $t = 0.0$ to $t = 0.3$). However, the standard deviation across items roughly doubles from $t = 0.0$ to $t = 1.0$ for all local models.

This is consistent with Renze and Guven [5], who find temperature changes from 0.0 to 1.0 are not statistically significant for problem-solving tasks on full-precision large models. We extend this finding to quantised 7–9B models on consumer hardware: the pattern holds at Q4 and Q8 precision on context-grounded Q&A, a task type not previously tested in the temperature sensitivity literature. The pattern is also robust across the added Qwen 2.5 7B and DeepSeek-R1-Distill-Qwen-7B models.

The "Hot or Cold?" study [6] introduces a "mutation temperature" (the threshold at which performance degrades) and shows it increases with model size. Our 7–9B models sit at the boundary between their "small" (1–4B) and "medium" (6–13B) categories. Consistent with their framework, we observe moderate sensitivity: the mean does not degrade, but individual-item reliability does.

4.1.2 Mild temperature outperforms greedy decoding

A nuance not captured by the "no significant difference" framing: most local models achieve their best or near-best composite scores at $t = 0.3$ rather than $t = 0.0$, and for two of five local families the improvement is statistically significant. Table 3 shows the best and worst parameter settings per model on the extractive items.

Table 3: Best and worst parameter settings per model on extractive items, with composite score and delta between them.

Model	Best setting	Score	Worst setting	Score	Δ
Opus 4.6	t=0.0, p=1.0	5.00	t=0.0, p=0.5	4.97	0.03
Qwen 2.5 7B	t=0.3, p=1.0	4.74	t=0.0, p=0.5	4.62	0.12
GLM-4-9B Q4	t=0.3, p=0.9	4.72	t=0.0, p=0.9	4.42	0.31
Gemma 4 E4B Q8	t=0.3, p=1.0	4.78	t=1.0, p=1.0	4.56	0.22
Llama 3.1 8B Q4	t=0.7, p=1.0	4.78	t=0.7, p=0.9	4.56	0.22
DeepSeek-R1-Distill 7B	t=0.3, p=1.0	4.46	t=0.7, p=0.5	4.25	0.21

Wilcoxon signed-rank tests (Appendix D) show that $t = 0.3$ significantly outperforms $t = 0.0$ for GLM-4-9B Q4 ($p < 0.001$) and Gemma 4 E4B Q8 ($p < 0.05$), but not for Llama Q4 ($p = 0.93$) or Opus ($p = 1.0$). Gemma 4 E4B at Q4 shows a small non-significant decrease at $t = 0.3$ ($p = 0.06$), in contrast to its Q8 variant, suggesting that optimal sampling temperature within the Gemma family depends on quantisation level.

This is consistent with "Sample Smart, Not Hard" [45], which shows that controlled stochastic sampling outperforms greedy decoding on reasoning tasks when combined with self-consistency verification. Our finding extends this to single-sample inference on extractive Q&A: even without multi-sample aggregation, a small amount of randomness ($t = 0.3$) appears to help most models escape suboptimal greedy paths.

The practical implication is immediate: **use $t = 0.3$, not $t = 0.0$, as the default for local inference on extractive tasks.** The quality gain is small but consistent, not significantly worse than greedy for any model tested, and significantly better for two of five families. There is no latency cost; temperature is a softmax scaling parameter, not a compute multiplier. The one caveat, reported in Appendix D, is that Gemma 4 E4B Q4 shows a small (non-significant) *decrease* at $t = 0.3$, suggesting that within the Gemma family the optimal sampling strategy may be quantisation-dependent.

4.1.3 Cross-family parameter sensitivity

The parameter heatmap (Figure 2) illustrates a finding with no direct comparator in the literature: **model families differ substantially in parameter sensitivity at matched parameter counts.**

Among the original three families, GLM-4-9B exhibits a 0.31-point swing between its best and worst parameter combinations, roughly 50% larger than the 0.22-point range observed for both Llama and Gemma. The Qwen 2.5 7B and DeepSeek-R1-Distill 7B additions broaden the picture: Qwen is the

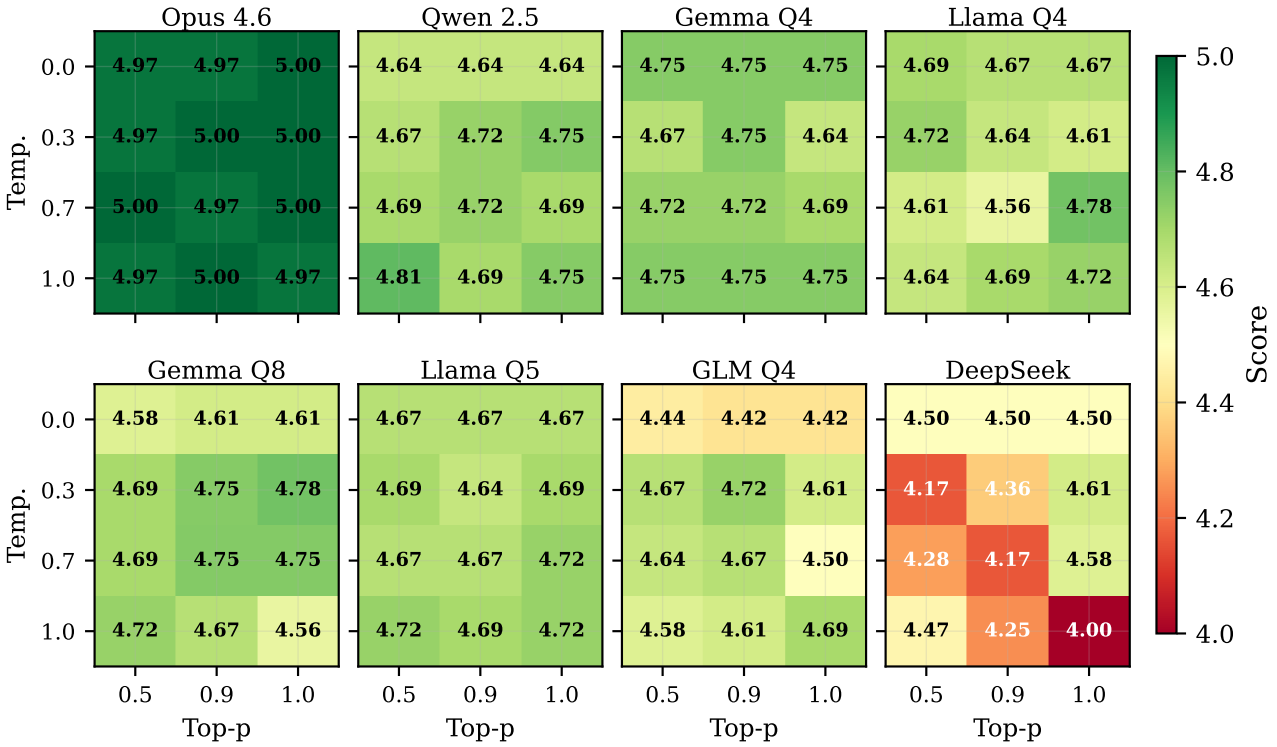

Figure 2: Parameter heatmap for all eight model configurations: mean composite score for each temperature × top-p combination. Opus (top left) is uniformly near-perfect. GLM (bottom centre-right) has the widest range among local models (0.31 points); Qwen (top centre-left) is the most parameter-robust (0.12 points). DeepSeek (bottom right) has the lowest overall scores, consistent with its extractive-task misfit.

most parameter-robust local model we tested (0.12-point range), and DeepSeek sits between Qwen and GLM at 0.21. GLM therefore remains the most tuning-sensitive model; practitioners deploying GLM should budget for parameter sweeps, while Qwen and Llama can be deployed at reasonable defaults with minimal loss.

No prior study compares parameter sensitivity across model families at matched size. Temperature studies either examine one model family across sizes [6] or test multiple models at a single temperature setting (standard benchmark practice). The observation that architectural differences produce different sensitivity profiles at the same parameter count suggests that parameter recommendations should be model-specific, not universal.

The frontier model (Opus) shows a 0.03-point range, effectively parameter- invariant, consistent with the clinical temperature study [25] finding that larger models are more robust to temperature-induced variation.

4.1.4 Top-p is irrelevant for context-grounded Q&A

Across all models at $t > 0$, moving from $p = 0.5$ to $p = 1.0$ changes composite scores by less than 0.1 points on average. No model shows a statistically meaningful preference for restrictive versus permissive nucleus size.

This confirms the task-dependency of sampling parameters established by Wiher et al. [21], who show that decoding effects are task-specific rather than universally optimal. Holtzman et al. [20] established nucleus sampling's value for open-ended generation; our finding demonstrates empirically that for factual Q&A from provided context (where the answer space is constrained by the passage) the nucleus size adds no value.

4.2 Quality-Latency Frontier

Figure 3 plots mean composite quality against mean latency per response for all eight model configurations on the 12 extractive items (factual, reasoning, synthesis). Table 4 gives the underlying numbers.

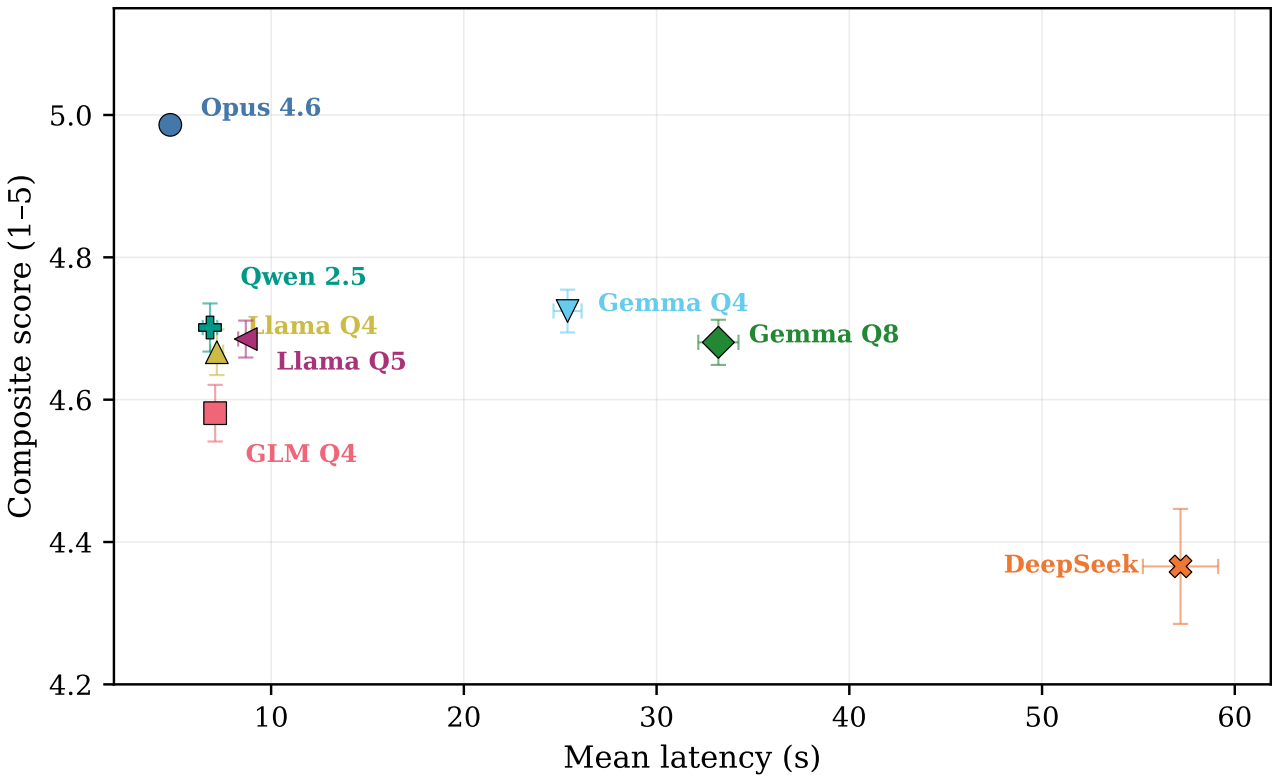

Figure 3: Quality vs latency for all model configurations on the extractive items. Qwen 2.5 7B and Llama 3.1 8B Q4 are co-located on the local Pareto frontier; Gemma Q4 offers higher accuracy at 3–4× the latency; DeepSeek is dominated on both axes. Error bars show ±1 SE.

This chart addresses the practitioner's core question: *which model gives the best quality per second on this hardware?*

Table 4: Quality–latency frontier on extractive items (factual, reasoning, synthesis).

Model	Composite	Latency	Quality/s
Claude Opus 4.6	4.99	4.8 s	1.04
Qwen 2.5 7B	4.70	6.8 s	0.69
Llama 3.1 8B Q4	4.67	7.2 s	0.65
Llama 3.1 8B Q5	4.69	8.7 s	0.54
GLM-4-9B Q4	4.58	7.1 s	0.65
Gemma 4 E4B Q4	4.72	25.4 s	0.19
Gemma 4 E4B Q8	4.68	33.2 s	0.14
DeepSeek-R1-Distill 7B	4.40	56.9 s	0.077

The frontier is unambiguous. Opus dominates on both axes but requires API access and is not local. Among local models, **Qwen 2.5 7B at Q4_K_M is the new Pareto-optimal choice**: it matches or exceeds every other local model on mean quality and does so at the lowest local latency (6.8 s). Llama 3.1 8B at Q4 remains an excellent option, effectively tied with Qwen at 7.2 s, and retains the advantage of a larger GGUF ecosystem.

The quality–latency scatter reveals three distinct clusters. The **fast cluster** (Qwen, Llama Q4, Llama Q5, GLM Q4) at 7–9 seconds achieves composite scores of 4.58–4.70. The **slow cluster** (Gemma Q4, Gemma Q8) at 25–33 seconds achieves 4.68–4.72. The **dominated point** (DeepSeek-R1-Distill) at 57 seconds achieves only 4.40. The quality difference between the fast and slow clusters is 0.03–0.14 points, within the range attributable to parameter tuning (Section 4.1), while the latency difference is 3–4×. DeepSeek's position is a qualitative finding discussed further in Section 4.4.

Existing inference benchmarks for Apple Silicon [7], [8], [37] report throughput and latency but not output quality. Quality benchmarks for quantised models [2], [12] report scores but not latency. To our knowledge, this is the first evaluation that maps both axes simultaneously for multiple model families on consumer hardware, producing the Pareto chart a practitioner needs to make a deployment decision.

4.3 Model Comparison on Extractive Q&A

Table 5 presents the comparison across all eight model configurations on the 12 extractive items (144 responses per model, all parameter combinations).

Table 5: Model comparison across extractive items (144 responses per model, minus parse failures).

Model	Accuracy	Completeness	Coherence	Latency	n
Claude Opus 4.6	5.00	4.96	5.00	4.8 s	144
Qwen 2.5 7B	4.83	4.40	4.87	6.8 s	144
Gemma 4 E4B Q4	4.90	4.38	4.89	25.4 s	144
Gemma 4 E4B Q8	4.92	4.30	4.82	33.2 s	144
Llama 3.1 8B Q4	4.74	4.35	4.91	7.2 s	144
Llama 3.1 8B Q5	4.78	4.31	4.97	8.7 s	144
GLM-4-9B Q4	4.68	4.29	4.77	7.1 s	144
DeepSeek-R1-Distill 7B	4.33	4.26	4.60	56.9 s	143

Qwen 2.5 7B is the new best local model overall. It beats Llama Q4 on accuracy (4.83 vs. 4.74), completeness (4.40 vs. 4.35), and latency (6.8 s vs. 7.2 s), with coherence effectively tied (4.87 vs. 4.91). Qwen is the only local model to achieve a perfect 5.00/5.00/5.00 on any single category (factual extraction, 48/48 responses perfect). This result displaces Llama 3.1 as the default recommendation for 7B-class local inference and is consistent with Qwen 2.5's strong standing on contemporary open benchmarks.

Gemma Q4 leads on accuracy but at 4× the latency of the fast cluster. Gemma's 4.90 accuracy exceeds Qwen's 4.83 and approaches Opus's 5.00. For batch or asynchronous workloads where 25 seconds per response is acceptable, Gemma Q4 is the highest-accuracy local option. For interactive use, Qwen's lower latency makes it the better choice.

DeepSeek-R1-Distill underperforms on extractive Q&A. At 4.33 accuracy, DeepSeek is the worst local model on the extractive items, below GLM (4.68) and Llama (4.74). The model's training on DeepSeek-R1 reasoning chains pushes it to generate long chains of thought even for simple factual questions, which compounds latency (57 s) and introduces errors the base model (Qwen, 4.83 accuracy) would not make. This is a misfit between model and task, discussed in Section 4.4 where DeepSeek's reasoning training is expected to help.

GLM-4-9B is competitive when tuned. At its best parameter setting ($t = 0.3, p = 0.9$), GLM achieves a 4.72 composite score, close to the fast cluster. At its worst, it falls to 4.42. This 0.31-point sensitivity range makes GLM the riskiest choice for applications where parameter tuning is impractical.

Local models achieve 93–96% of frontier quality on extractive Q&A. Expressed as a fraction of Opus's composite score, Qwen reaches 94%, Llama Q4 94%, Gemma 94–95%, GLM 92%, and DeepSeek 88%. For context-grounded Q&A, where the answer is derivable from the provided passage, the gap between quantised 7–9B models and a frontier system is small. This is consistent with the observation that the open-weights frontier trails closed models by approximately three months on average [41], and that the gap is even smaller on tasks that do not require frontier-level reasoning. Section 4.4 shows that this latter clause is load-bearing: on tasks that *do* require multi-step reasoning, the gap widens substantially.

4.4 Chain-of-Thought: Where the Frontier Gap Becomes a Chasm

The chain-of-thought (CoT) category is the most diagnostic of our four task types because it isolates a capability (multi-step reasoning with intermediate computation or case analysis) that extractive Q&A

does not exercise. Table 6 summarises CoT performance across all eight configurations.

Table 6: Chain-of-thought performance across all models. † DeepSeek measured under an extended 600 s / 8192-token budget on cot_02, cot_03, and cot_04 only; cot_01 is unmeasurable at any practical budget on our hardware (see below).

Model	Accuracy	Completeness	Coherence	Latency	n
Claude Opus 4.6	4.73	4.88	4.98	19.3 s	48
Gemma 4 E4B Q4	3.26	2.45	3.45	57.8 s	47
Gemma 4 E4B Q8	3.21	2.45	3.55	79.5 s	42
Qwen 2.5 7B	3.04	3.30	3.81	31.9 s	47
Llama 3.1 8B Q5	2.79	3.33	3.77	119.2 s	48
Llama 3.1 8B Q4	2.79	3.35	3.75	53.1 s	48
GLM-4-9B Q4	2.62	3.04	3.42	44.8 s	48
DeepSeek-R1-Distill 7B †	2.36	2.56	3.39	173 s	36

The frontier–local gap on CoT is qualitatively larger than on extractive tasks. Opus holds close to its extractive-task performance (composite 4.87 vs. 4.99) while every local model collapses to 2.36–3.38. On completeness specifically, Opus scores 4.88 while the best local model (Llama, 3.35) is 1.53 points lower. This contrasts with the 0.6-point extractive gap and constitutes the single clearest evidence of a capability-differentiated frontier advantage in our evaluation.

Model rankings reorder substantially between extractive and CoT. Qwen leads on extractive Q&A and sits fourth on CoT; Gemma Q4 is third on extractive and first on CoT (among local models); Llama Q4 is second on extractive and sixth on CoT. No local model dominates across both categories. Aggregate rankings that average across task types can therefore mislead practitioners with specific workload characteristics.

Error profiles differ by model family. Gemma has the highest CoT accuracy among local models (3.26) but the lowest completeness (2.45), indicating it often arrives at a correct direction but stops short of working through the full problem. Llama shows the inverse profile (completeness 3.35, accuracy 2.79): it works through the full chain but makes errors along the way. Qwen is the most balanced (3.04/3.30/3.81). These profiles imply that chain-of-thought failure modes are qualitatively different across families and cannot be collapsed into a single quality metric.

DeepSeek-R1-Distill is the weakest local CoT model, confirming the reasoning-distillation misfit. Our initial measurement of DeepSeek under the shared 120 s / 1024-token budget produced a 46% timeout rate and an apparent composite of 3.08 on the 26 completed responses, broadly comparable to Qwen. To test whether this understated DeepSeek's capability, we re-ran the model with an extended 600 s / 8192-token budget on all four CoT items. Two findings emerged. First, cot_01 (multi-step arithmetic over financial transactions) is unmeasurable at any practical budget on our hardware: at observed generation speed (~68 ms/token on M2 Q4_K_M), DeepSeek's reasoning chain exceeds 8192 tokens and wall-clock time runs out before completion, even at 10 minutes per call. Second, on the three measurable questions (cot_02–cot_04) with all 12 parameter combinations (n=36, zero timeouts), DeepSeek's extended-budget score is 2.36 accuracy, *lower* than the original truncated measurement (2.92 on cot_02–04 completions).

Two mechanisms explain the drop. The original n=8 (cot_03) and n=4 (cot_04) were biased toward the easier parameter combinations that happened to complete within 120 s; harder combos (higher temperature, lower top-p), where DeepSeek gets most confused, timed out and were excluded, inflating the apparent average. Additionally, longer generation budgets allow DeepSeek to produce longer reasoning chains that compound intermediate errors rather than correct them. The extended-budget result is confirmed by the independent Gemini judge (composite 2.47, consistent with its usual 0.3-point offset from Opus on local CoT). **DeepSeek-R1-Distill-Qwen-7B at Q4_K_M is therefore not a viable local CoT model on 16 GB hardware:** it is either truncated into incompleteness at short budgets or compounds errors in its long reasoning chains at longer ones, and one of the four

CoT items is entirely unmeasurable.

Latency on CoT is universally degraded. Every local model runs 2–6× slower on CoT than on extractive items. The longer reasoning chains generate three to ten times more tokens, and token count is the dominant latency driver. CoT workloads are therefore not only quality-limited but latency-limited on consumer hardware: even the fastest local model takes 32 seconds per response, nearly two minutes for Llama Q5.

We note an important caveat to the local-model CoT scores in Table 6: under the independent Gemini judge (Section 4.5), local models score 0.17–0.40 points lower on CoT accuracy than under Opus. The Opus–local CoT gap reported above is therefore conservative; the true gap is wider.

4.5 Multi-Judge Validation

The most significant methodological concern in our design is that Opus serves as both a subject and a judge. Self-preference bias in LLM evaluators is a documented phenomenon [10]: models score text matching their own training distribution as higher quality. To address this concern empirically, every response was independently re-scored by Gemini 2.5 (Pro on the original sweep, Flash on the sub-runs) using the identical blind rubric. This allows us to (i) measure self-preference bias directly, (ii) quantify inter-judge agreement, and (iii) test the robustness of the model rankings to judge choice.

Self-preference bias is empirically negligible on extractive Q&A. Table 7 compares Opus's scores of Opus's own outputs against Gemini's scores of those same outputs. On the original parameter sweep (extractive items only), Opus's self-rating exceeds Gemini's rating of Opus by at most 0.04 points across the three dimensions. On chain-of-thought, the bias is larger but still small: +0.19 on completeness, +0.12 on coherence.

Table 7: Self-preference bias measured by judge swap. Negligible on extractive Q&A; small on CoT.

Dataset	Dimension	Opus→Opus	Gemini→Opus	Bias
Extractive	accuracy	5.00	5.00	+0.00
Extractive	completeness	4.96	4.91	+0.04
Extractive	coherence	5.00	5.00	+0.00
CoT	accuracy	4.73	4.81	−0.08
CoT	completeness	4.88	4.69	+0.19
CoT	coherence	4.98	4.85	+0.12

The previously documented self-preference bias for Claude-family models [10] does not materially manifest in our setting, plausibly because (i) the task is verifiable against a passage rather than stylistically subjective, (ii) blind reference answers anchor the rubric, and (iii) Opus's extractive answers genuinely operate at or near the scale ceiling.

Inter-judge agreement is strong on extractive Q&A and weaker on CoT. Table 8 reports Pearson correlation, exact agreement, and within-±1 agreement between the two judges across all dimensions and datasets.

Table 8: Inter-judge agreement (Opus vs. Gemini 2.5).

Dataset	Dimension	Pearson r	Exact agree.	Within ±1
Extractive	accuracy	0.747	90%	99%
Extractive	completeness	0.744	72%	100%
Extractive	coherence	0.669	89%	99%
CoT	accuracy	0.844	45%	94%
CoT	completeness	0.818	60%	93%
CoT	coherence	0.744	58%	94%

The CoT correlation is paradoxically *higher* than extractive (judges order answers very similarly,

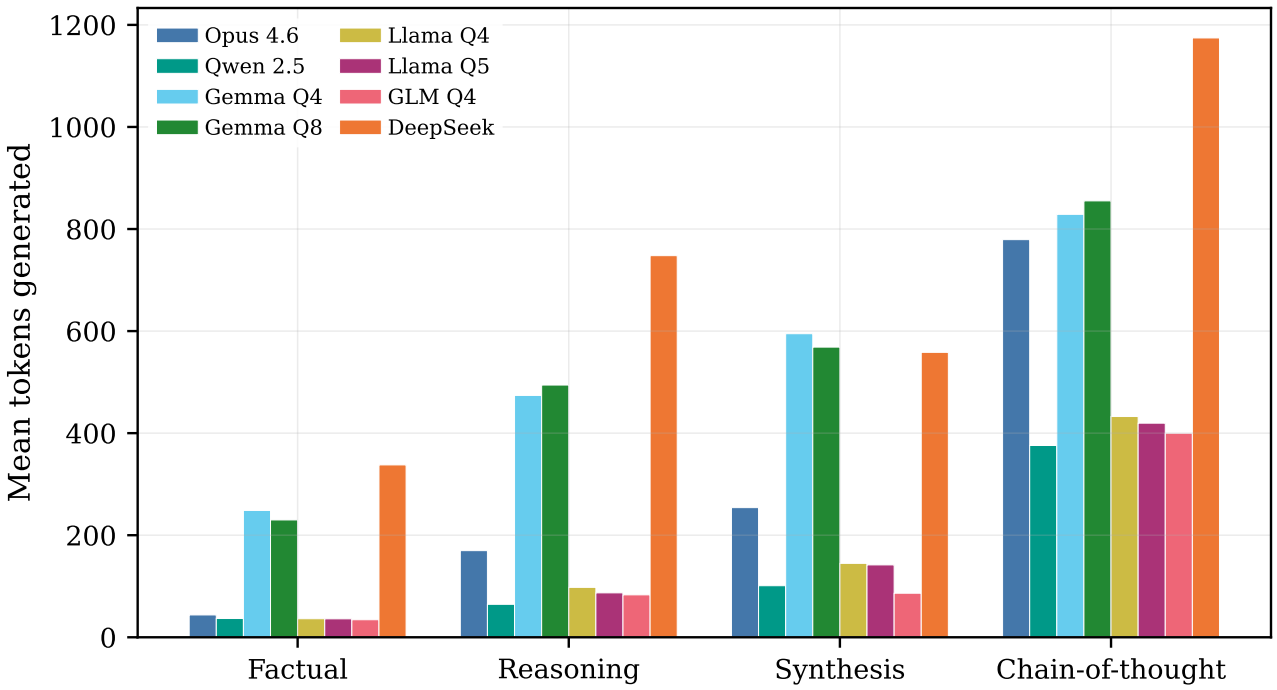

Figure 4: Mean tokens generated per model and task category. Gemma is persistently 3–7× more verbose than Qwen and Llama across all categories (architectural verbosity). DeepSeek-R1-Distill shows explosive token growth on CoT (training-induced verbosity).

$r = 0.74$–0.84) but exact agreement is much lower (45–60% vs. 72–90%). The judges agree on rank but disagree on absolute magnitude: reasonable CoT answers can fairly score 3 or 4 depending on how strictly intermediate reasoning steps are weighted. Both judges are internally consistent but apply different thresholds. The disagreement rate (≥ 2 points apart) scales with task difficulty: 1.4% on the quantisation comparison, 2.1% on the original sweep, 3.5% on the new-models baseline, and 13.8% on CoT.

Model rankings are robust to judge choice. Across all four sub-runs, the Opus and Gemini rankings of local models are essentially identical, with two within-noise position swaps. The most interesting exception is on CoT: Opus ranks Qwen 2.5 7B as the strongest local CoT model (composite 3.38), while Gemini ranks Llama 3.1 8B Q4 as strongest (composite 3.23 under Gemini vs. Qwen's 3.14 under Gemini). The two judges genuinely disagree on which mid-sized model handles multi-step reasoning best. We treat this as an open question and report both rankings rather than picking a winner.

Gemini systematically scores local CoT lower. Across all five local models, Gemini's CoT accuracy scores are 0.17–0.40 points below Opus's. The "local models score 2.5–3.8 on CoT" framing in Section 4.4 is the *generous* reading; under Gemini, local CoT accuracy ranges from 2.27 to 3.28. The qualitative finding (local models collapse on CoT relative to Opus) holds under either judge, and is in fact stronger under Gemini.

Caveat: empty-answer hallucination. The largest individual judge disagreements ($\Delta = 4$) come from a specific Gemini failure mode. When a candidate response is empty or truncated, Opus correctly assigns 1/1/1; Gemini sometimes hallucinates content that "should have been there" and scores the response as if complete. This affected a small fraction of responses (predominantly DeepSeek's CoT timeouts) and we exclude these from the Gemini scores in Table 7. Production LLM-judge pipelines should short-circuit empty answers before sending them to any model judge.

4.6 Verbosity and Quality

Figure 4 plots token counts by model and task category. Gemma generates dramatically more tokens than most other models across every category and parameter combination, and **DeepSeek-R1-Distill**

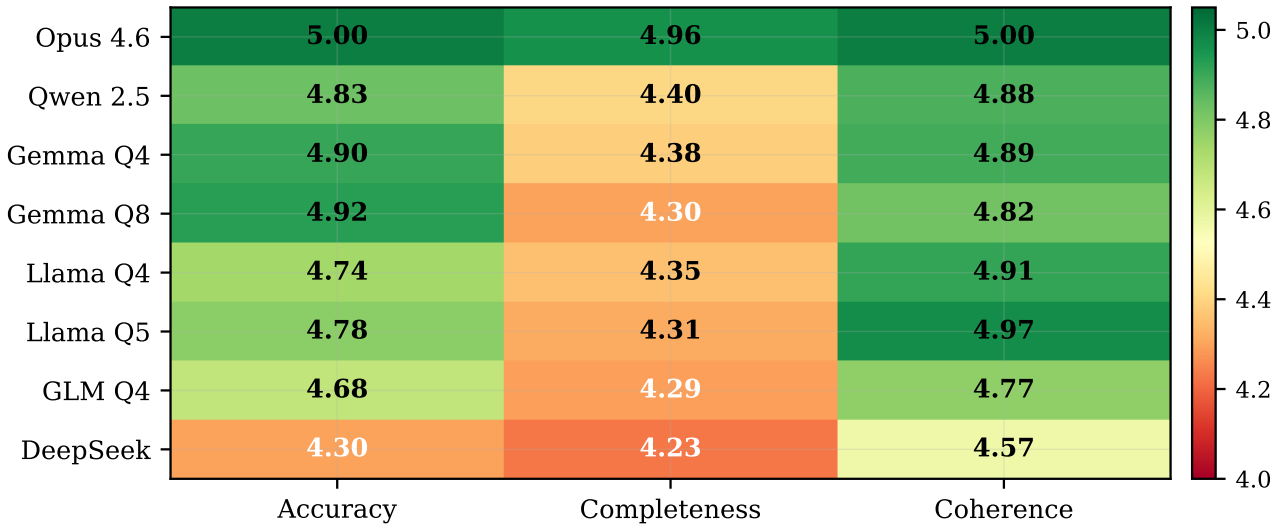

Figure 5: Score breakdown by model and dimension on the extractive items. Completeness (centre column) is the hardest dimension for all models; accuracy is the easiest. Qwen achieves the most balanced profile.

produces the longest outputs overall on CoT, generating three to ten times more tokens than the fast cluster. This is not a temperature artefact; the verbosity pattern persists at $t = 0.0$.

Two distinct verbosity profiles emerge. **Architectural verbosity** (Gemma): 3–7× more tokens than Qwen or Llama across all categories, an intrinsic property of the model's generation style. **Training-induced verbosity** (DeepSeek): relatively concise on factual questions but explosively verbose on reasoning and CoT, reflecting its R1-style chain-of-thought training objective.

Despite producing the most text among general-purpose models, Gemma achieves the second-lowest completeness score (4.30 at Q8, 4.38 at Q4) while Qwen and Llama, with moderate token counts, achieve higher completeness (4.40 and 4.35 respectively). DeepSeek shows the clearest anticorrelation: highest token counts on CoT but completeness of only 2.56, below Qwen (3.30) and Llama (3.35). This anticorrelation between verbosity and completeness is consistent with "Verbosity $\neq$ Veracity" [46], which demonstrates that verbose LLM responses correlate with high uncertainty and lower accuracy. Our finding extends this to quantised models and reveals both an architectural dimension (Gemma's consistent verbosity) and a training-objective dimension (DeepSeek's reasoning-chain verbosity).

The verbosity finding also interacts with known LLM-judge biases. Saito [35] shows that LLM judges prefer longer responses independent of quality, and the length bias literature [36], [47] demonstrates that excessive length inflates perceived quality through increased information mass. If Opus-as-judge exhibits this bias, Gemma's and DeepSeek's completeness scores may be inflated relative to their true performance, meaning the real completeness gaps reported elsewhere in this section may understate the true differences.

4.7 Score Dimension Breakdown

Completeness is the hardest dimension for all models, including the frontier baseline, on extractive Q&A. Opus scored below 5.0 only on completeness (4.96), and completeness shows the largest gap between local models and the frontier (0.56–0.70 points vs. 0.08–0.67 for accuracy and 0.03–0.40 for coherence). On CoT (Section 4.4), completeness is also the dimension with the largest frontier gap: Opus 4.88 vs. best local 3.35, a 1.53-point gap.

Accuracy is the easiest dimension on extractive Q&A. All local models score 4.33–4.92, with Gemma (4.90–4.92) and Qwen (4.83) approaching Opus. The models extract correct information from the passage reliably; the failure mode is omission (low completeness), not error (low accuracy). DeepSeek is the exception (4.33 accuracy): its reasoning-chain outputs introduce errors on extractive items that its base model Qwen (4.83) would not make.

This pattern maps to the distinction between faithfulness and completeness in the RAG evaluation

literature. The CRAG benchmark [43] and GaRAGe [44] separately measure whether a model's answer is grounded in the source (faithfulness/accuracy) and whether it covers all relevant information (completeness). Our dimension breakdown confirms that grounding is the easier problem for quantised small models, while comprehensive coverage remains the harder one.

The qualitative extremes illustrate this pattern. The best local answer on extractive Q&A (GLM-4-9B, reasoning_04, $t = 0.3, p = 0.5$) scored a perfect 15/15 in 70 tokens and 5.1 seconds, a complete causal chain for a question about Lake Victoria's water clarity. The worst local answer (GLM-4-9B, reasoning_03, $t = 0.7, p = 0.5$) scored 9/15: it correctly cited evidence about TDD adoption but then contradicted that evidence in its conclusion, a reasoning coherence failure that persisted across multiple parameter settings.

4.8 Quantisation Level Effects

Motivated by the primary sweep results, we ran a follow-up experiment testing two additional quantisation variants: Llama 3.1 8B at Q5_K_M (up from Q4) and Gemma 4 E4B at Q4_K_M (down from Q8). Each variant was evaluated on the same 12 items × 12 parameter combinations.

Table 9: Quantisation comparison: same model architecture at different precision levels.

Model	Quant	Accuracy	Completeness	Coherence	Latency	Size
Llama 3.1 8B	Q4_K_M	4.74	4.35	4.91	7.2 s	4.7 GB
Llama 3.1 8B	Q5_K_M	4.78	4.31	4.97	8.7 s	5.5 GB
Gemma 4 E4B	Q4_K_M	4.90	4.38	4.89	25.4 s	5.2 GB
Gemma 4 E4B	Q8_0	4.92	4.30	4.82	33.2 s	8.0 GB

Higher quantisation does not improve quality for Llama. Moving from Q4 to Q5 yields +0.04 accuracy and +0.06 coherence but −0.04 completeness, all within noise margins. Latency increases 21% and GGUF size grows 17%. The unified llama.cpp quantisation evaluation [3] similarly finds Q4_K_M to be the optimal tradeoff for Llama 3.1 8B across multiple benchmarks; our task-specific results confirm this general finding.

Gemma's accuracy advantage is architectural, not quantisation-dependent. Reducing Gemma from Q8 to Q4 barely changes accuracy (4.90 vs. 4.92) while improving completeness (+0.08), coherence (+0.07), and latency (−23%). This result has two implications. First, practitioners should prefer Gemma Q4 over Q8 on memory-constrained hardware, since Q8 provides no quality benefit. Second, the accuracy gap between Gemma and Llama (4.90 vs. 4.74 at Q4) is attributable to architectural differences (Gemma's Per-Layer Embedding design) rather than the quantisation advantage that Q8 appeared to confer in the primary comparison.

This aligns with the SLMQuant finding [2] that small models respond to quantisation differently from large models, and with the IJCAI 2025 result [18] that smaller models at higher bitwidths outperform larger models at extreme quantisation; our contribution is showing that, within the 8B class, the Q4-to-Q8 transition yields negligible quality improvement on a practical task.

5. Discussion

5.1 Practical Recommendations

The results converge on a set of actionable recommendations for practitioners deploying local models for context-grounded Q&A on consumer hardware. Table 10 summarises the workload-to-model mapping; the prose below provides the underlying reasoning.

5.1.1 Model Selection Guide

The guide above is a practical summary; the supporting evidence and caveats follow.

Model selection by workload. For extractive Q&A (factual, reasoning, synthesis), Qwen 2.5 7B at Q4_K_M is the default recommendation: it is Pareto-optimal at 94% of frontier quality and 6.8 seconds per response, and it is the only local model to achieve perfect scores on the factual category.

Table 10: Workload-to-model mapping for local LLM deployment on consumer Apple Silicon. CoT recommendations are based on $n = 4$ items per local model and should be treated as provisional.

If your workload is…	Use	Why
Extractive Q&A, latency-sensitive	Qwen 2.5 7B Q4	Pareto-optimal: composite 4.70 at 6.8 s; perfect on factual
Extractive Q&A, ecosystem maturity	Llama 3.1 8B Q4	Tied-optimal quality (4.67), largest GGUF community
Extractive Q&A, accuracy-first, latency-tolerant	Gemma 4 E4B Q4	Highest local accuracy (4.90) at 25.4 s
Chain-of-thought, best accuracy	Gemma 4 E4B Q4	Highest CoT accuracy (3.26) under Opus judge
Chain-of-thought, best completeness	Llama 3.1 8B Q4	Highest CoT completeness (3.35); preferred under Gemini judge
Chain-of-thought, balanced profile	Qwen 2.5 7B Q4	Most even across CoT dimensions (3.04 / 3.30 / 3.81)
Any reasoning-heavy workload	Frontier API (e.g., Opus)	No local model exceeds 75% of frontier CoT quality
Any CoT workload	Avoid DeepSeek-R1-Distill 7B	Weakest local CoT; compounds errors in long chains; cot_01 unmeasurable
Parameter tuning available	Defaults plus $t = 0.3$	Mild temperature beats greedy for 4 of 5 families; top-p irrelevant
Parameter tuning *not* available	Avoid GLM-4-9B	Most parameter-sensitive (0.31-pt swing); Qwen most robust (0.12-pt)

Llama 3.1 8B Q4 remains an excellent alternative with a larger community ecosystem. For latency-tolerant workloads where accuracy is paramount, Gemma 4 E4B at Q4 offers the highest local accuracy (4.90, essentially tied with Q8's 4.92) at 25 seconds, and is preferred over Q8 for memory efficiency. For chain-of-thought workloads, no local model is a strong choice at this parameter scale: Gemma Q4 leads on CoT accuracy (3.26 under Opus, 3.09 under Gemini) but scores poorly on completeness (2.45). The frontier gap is large (Opus composite 4.87 vs. best local 3.32) and the Opus–Gemini judges disagree on which 7B model is strongest at CoT (Section 4.5). Practitioners with CoT-heavy workloads should consider frontier API access or larger open-weights models.

Avoid reasoning-distilled models on extractive tasks. DeepSeek-R1-Distill 7B is the clearest cautionary tale: on extractive Q&A it produces the worst local accuracy (4.33) because its training pushes it toward long reasoning chains that introduce errors the base model (Qwen, 4.83) does not make. Reasoning-specialised models should be matched to reasoning workloads.

Sampling parameters. Use $t = 0.3$, not $t = 0.0$, as the default temperature. Four of five local families improve or tie at $t = 0.3$ versus greedy, with statistically significant improvements for GLM-4-9B Q4 ($p < 0.001$) and Gemma 4 E4B Q8 ($p < 0.05$). Gemma 4 E4B Q4 is a narrow exception that slightly favours greedy ($p = 0.06$). There is no latency cost. Do not spend time tuning top-p for extractive Q&A; hold it constant at 1.0. If deploying GLM, invest in parameter tuning; the 0.31-point sensitivity range means defaults may leave substantial quality on the table.

Quantisation level. For models in the 7–9B range on 16 GB hardware, Q4_K_M is the right tradeoff. Q5 and Q8 do not improve quality enough to justify the memory and latency overhead. This is consistent with the unified llama.cpp quantisation study [3] and the broader finding from SLMQuant [2] that small models respond poorly to the assumption that higher precision always means higher quality.

Evaluation design. When evaluating local models using LLM-as-judge, always provide reference answers in the judge prompt, as this is the single most impactful design choice for reliability [34].

Record token counts alongside quality scores to detect verbosity confounds. If the judge model is also a subject, report this transparently and interpret its self-scores as upper bounds. Always include a chain-of-thought category if the model may be used for multi-step reasoning; aggregate rankings across extractive categories systematically understate frontier advantages on CoT tasks.

5.2 Limitations

Self-preference bias (largely resolved by multi-judge validation). Using Opus 4.6 as a judge of its own outputs introduces a documented concern: Panickssery et al. [10] demonstrate that LLM evaluators favour their own outputs through a perplexity-based mechanism. We addressed this concern empirically by re-judging every response with Gemini 2.5 (Section 4.5). The measured self-preference bias is +0.04 on extractive completeness and +0.19 on chain-of-thought completeness, with all other dimensions within ±0.12. These are well below the magnitude of the frontier–local gap on either task type. The previously documented Claude-family self-preference bias [10] does not materially manifest in our setting, plausibly because (i) the task is verifiable against a passage rather than stylistically subjective and (ii) blind reference answers anchor the rubric.

A residual concern remains for chain-of-thought, where the bias is small but non-zero and where the two judges meaningfully disagree on which 7B model is strongest at multi-step reasoning (Opus prefers Qwen, Gemini prefers Llama). A stronger design would use a third judge from a different model family (e.g., GPT-4o or Llama-405B) as a tiebreaker on contested CoT cases. We did not extend to a third judge due to cost; this is a recommended extension.

Critically, **the comparisons among local models are unaffected by self-preference bias**, since none of the local models is the judge. The relative rankings of Llama, Gemma, GLM, Qwen, and DeepSeek are based on independent evaluators (Opus and Gemini), and rankings are essentially identical between the two judges (with one within-noise position swap on extractive and one meaningful disagreement on CoT). The bias affects only the interpretation of Opus's absolute scores and the precise magnitude of the frontier gap; on chain-of-thought, the gap is large enough (1.5–2.5 points on completeness) that the bias cannot account for the effect.

Small evaluation set. Sixteen context-grounded questions provide sufficient data points (up to 192 per model after the parameter sweep) to observe trends and rank models, but not to make fine-grained statistical significance claims about individual questions or narrow parameter ranges. The CoT category in particular has only four items; while the frontier–local gap on CoT is qualitatively clear and robust across all local models, finer distinctions among local models on CoT should be treated cautiously. Established benchmarks (CRAG [43], GaRAGe [44], SQuAD, Natural Questions) would improve external validity. We report effect sizes and confidence intervals in Appendix D to allow readers to assess the strength of evidence for each finding.

DeepSeek CoT measurement (resolved). Our original 120 s per-response timeout caused DeepSeek-R1-Distill to time out on 22 of 48 CoT responses. We initially flagged this as a possible under-measurement. A follow-up run with an extended 600 s / 8192-token budget (Section 4.4) refuted the under-measurement hypothesis: extended-budget scores on the three measurable CoT items are *lower* than the original truncated measurement, not higher, due to a combination of selection bias in the original completions and error accumulation in longer reasoning chains. The fourth CoT item (cot_01, multi-step arithmetic) is not measurable at any practical budget on 16 GB hardware and is excluded from DeepSeek's aggregate score. DeepSeek's CoT ranking in Section 4.4 reflects the extended-budget measurement and is robust to judge choice.

Judge hallucination on empty answers. During the multi-judge validation we observed that Gemini occasionally hallucinates content for empty or truncated responses, scoring them as if the missing content were present (typically 5/5/5). Opus correctly assigns 1/1/1 to such cases. This affected a small fraction of responses, predominantly DeepSeek's CoT timeouts, and we exclude these from the Gemini scores reported in Section 4.5. The implication for production LLM-judge pipelines is that empty/truncated answer detection should occur upstream of the judge call, not be delegated to the judge.

Single hardware configuration. All local inference runs on a single M2 MacBook with 16 GB uni-

fied memory. Results may not transfer to other Apple Silicon variants (M3, M4, M-series Ultra/Max with more memory bandwidth), NVIDIA GPUs (where CUDA-optimised runtimes like vLLM may outperform llama.cpp), or RAM-constrained mobile devices. The methodology in Section 3.7 is designed to be portable, but the specific numbers (latencies, throughput, thermal behaviour) are hardware-specific.

Inference framework. We use llama.cpp exclusively. Recent benchmarks [7] show MLX achieves 20–87% higher throughput than llama.cpp for models under 14B on Apple Silicon. The quality rankings should be framework-independent (the same GGUF weights produce the same logits regardless of runtime), but latency results would differ under MLX, and we do not verify framework equivalence empirically.

GGUF-only quantisation. Other quantisation formats (GPTQ via ExLlamaV2, AWQ via vLLM) may produce different quality profiles at the same bitwidth. The GGUF ecosystem is the most portable and the best-supported on Apple Silicon via llama.cpp, making it the pragmatic choice, but our quantisation results should not be generalised to other formats without testing.

Single-sample stochastic evaluation. At $t > 0$, each parameter combination produces a single sample per question. We estimate variance from cross-question variation, not from replicated runs of the same question at the same parameters. This means our variance estimates conflate per-question difficulty variation with sampling randomness. A more rigorous design would run k 3 replicates per question–parameter combination and decompose variance into its components.

No instruction-following dimension. Our evaluation covers extractive Q&A and chain-of-thought but not instruction following (format constraints, length limits, refusal behaviour). This is a separate capability dimension where local models typically diverge from frontier systems, and studies incorporating format-adherence evaluation would provide a more complete picture.

5.3 Threats to Validity

Construct validity. The 1–5 Likert scoring scale compresses a rich quality space into integers. A ceiling effect is evident on extractive items: Opus saturates at 5.0 on accuracy and coherence, limiting the scale's ability to discriminate between "very good" and "excellent" responses. The effect is attenuated on CoT, where Opus drops to 4.73 on accuracy and scores spread more widely across local models. A finer scale (1–10) or continuous scoring might reveal quality differences invisible at our resolution. The three dimensions (accuracy, completeness, coherence) were chosen for interpretability and alignment with established evaluation frameworks [9], but they do not capture all facets of answer quality; informativeness, conciseness, and source attribution are absent.

Internal validity. The parameter sweep design is observational: we vary temperature and top-p but do not control for interaction effects with question difficulty, context length, or domain. The finding that $t = 0.3$ outperforms $t = 0.0$ is consistent across models but could reflect a confound; for example, if greedy decoding is specifically worse on reasoning questions (which have the widest score distributions), the aggregate advantage of $t = 0.3$ may be task-type-dependent rather than universal. Appendix B provides per-question breakdowns to support examination of this possibility.

External validity. Hand-crafted questions may not represent the distribution of real-world Q&A workloads. The contexts are factual, English- language, and 150–250 words, shorter and more information-dense than typical RAG retrieval contexts. Models may perform differently on longer passages, multi-document contexts, or non-English text. The multilingual LLM-as-judge literature [48] documents inconsistency across languages, and our English-only evaluation does not address this.

5.4 Broader Implications

The narrow gap between quantised small models and the frontier on extractive context-grounded Q&A (93–96% of Opus quality) has implications beyond model selection. It suggests that for the specific task of answering questions from provided passages (the core capability underlying RAG applications) the constraint is increasingly in retrieval quality, context construction, and prompt design rather than in the generation model itself. A quantised 7–9B model that faithfully extracts and synthesises information from well-retrieved passages may be "good enough" for a substantial fraction

of practical applications, particularly where privacy, latency, cost, or offline operation are requirements that preclude API-based frontier access.

The task-dependent gap is the more important message. Collapsing "frontier vs local" into a single ratio conceals the structure revealed by the CoT results. On extractive Q&A the gap is 4–7%; on chain-of-thought the gap is 25–45%. This has three implications: (1) practitioners should evaluate local models on the specific task types they intend to deploy, not on aggregate benchmarks; (2) RAG-style applications (retrieval plus extraction/synthesis) are far better suited to local deployment than reasoning-heavy applications; and (3) the frontier advantage documented in contemporary benchmarks is real but concentrated in reasoning tasks, not uniformly distributed across all language capabilities.

The parameter sensitivity findings reinforce the first implication: the quality difference between a well-tuned and a poorly-tuned local model (0.12–0.31 points across the five families tested) is comparable to the gap between local models and the frontier on extractive Q&A (0.28–0.66 points). Parameter tuning on local models is roughly as impactful as switching to a frontier API for extractive workloads; it cannot close the CoT gap.

6. Conclusion

We present the first cross-cutting evaluation that jointly measures quality, latency, and sampling parameter sensitivity for quantised 7–9B parameter models on consumer Apple Silicon hardware across four task categories including chain-of-thought, with every response independently scored by two judges. Across more than 1,500 scored responses spanning five model families, two quantisation levels per family (where tested), 12 parameter combinations, and 16 evaluation items, five findings stand out.

First, the frontier–local gap is task-dependent rather than uniform. Quantised 7–9B models achieve 93–96% of frontier quality on extractive Q&A but only 55–75% on chain-of-thought. For RAG-style applications where privacy, latency, or cost preclude API access, the quality sacrifice on extractive workloads is narrower than commonly assumed; for reasoning-intensive applications, the gap is large enough that local deployment is not a like-for-like substitute.

Second, Qwen 2.5 7B at Q4_K_M is the new Pareto-optimal local model on extractive Q&A, displacing Llama 3.1 8B. It achieves 94% of frontier quality at 6.8 seconds per response and is the only local model to reach perfect scores on any single category (factual extraction).

Third, a mild temperature ($t = 0.3$) outperforms greedy decoding for four of five local model families; the improvement is statistically significant for GLM-4-9B Q4 ($p < 0.001$) and Gemma 4 E4B Q8 ($p < 0.05$). Top-p has no measurable effect on extractive Q&A quality. These findings extend the temperature results of Renze and Guven [5], established on full-precision large models, to the quantised small-model regime. Model families differ substantially in parameter sensitivity: Qwen is the most robust (0.12-point range), GLM the most sensitive (0.31).

Fourth, higher quantisation precision (Q8 vs Q4) provides negligible quality improvement within the 8B model class. Gemma's accuracy advantage over Llama is architectural, not quantisation-dependent. Reasoning-distilled models (DeepSeek-R1-Distill) misfire on extractive Q&A, producing the worst local accuracy due to training-induced verbose reasoning outputs.

Fifth, the methodological concern of self-preference bias from using Opus as both subject and judge is empirically negligible in our setting. Independent re-judging by Gemini 2.5 yields a measured Opus self-preference of +0.04 on extractive completeness and +0.19 on chain-of-thought completeness, with model rankings essentially identical between the two judges. The documented Claude-family self-preference bias does not materially manifest when the rubric is anchored by reference answers and the task is verifiable against a passage.

These findings are limited by a small hand-crafted test set and a single hardware configuration. We provide a generalised seven-step methodology (Section 3.7) for replication on other hardware and model configurations, a model selection guide (Section 5.1) for practitioners deploying these models today, and we release the evaluation harness, question set, and full results (both judges' scores) to

support extension. All artefacts are available at https://github.com/joe-southin/local-lm.

Appendix A: Evaluation Question Set

The 16 context-grounded questions span four categories. Each item provides a self-contained context passage (150–250 words), a question, and a reference answer. Contexts cover history, biology, space science, genetics, economics, software engineering, ecology, energy policy, labour markets, materials science, urban policy, finance, and public health. The full text of all items, including contexts and reference answers, is available in `eval_questions.json` in the supplementary materials.

Table 11: Evaluation question set summary.

ID	Category	Domain	Question topic
F01	Factual	History	Great Fire of London, parish churches
F02	Factual	Biology	Photosynthesis stages, RuBisCO enzyme
F03	Factual	Space science	ISS habitation date, solar array output
F04	Factual	Genetics	Protein-coding genome percentage
R01	Reasoning	Economics	Japanese consumption tax, takeout behaviour
R02	Reasoning	Housing	US homeownership decline, affordability factors
R03	Reasoning	Software eng.	TDD adoption timing, cost-benefit
R04	Reasoning	Ecology	Lake Victoria, Nile perch, water clarity
S01	Synthesis	Energy	Denmark wind energy strategy, vulnerabilities
S02	Synthesis	Labour	Remote work trade-offs across stakeholders
S03	Synthesis	Materials	Lithium-ion battery sustainability
S04	Synthesis	Urban policy	Singapore public housing strengths and challenges
C01	Chain-of-thought	Finance	Multi-step arithmetic over financial transactions
C02	Chain-of-thought	Public health	Causal chain root-cause identification
C03	Chain-of-thought	Engineering	Comparative evaluation of two approaches
C04	Chain-of-thought	Ecology	Counterfactual inference on policy outcome

Appendix B: Per-Question Score Breakdown

Mean composite scores (accuracy + completeness + coherence, divided by 3) for each question, averaged across all 12 parameter combinations per model on the extractive items and the CoT items.

Factual questions are solved near-perfectly by all general-purpose models. The quality spread widens through reasoning and synthesis and widens further on chain-of-thought. The CoT items expose the largest frontier–local gap: Opus stays above 4.8 on every CoT item while the best local model (Qwen) never exceeds 3.44. S03 (lithium-ion battery sustainability) is the only extractive question where Opus scored below 5.0 (4.83), receiving 4/5 on completeness.

Appendix C: Example Answers

Best local answer

GLM-4-9B-Chat Q4 / reasoning_04 at $t = 0.3$, $p = 0.5$ Score: 15/15 (accuracy 5, completeness 5, coherence 5). 70 tokens, 5.1s.

> Water clarity in Lake Victoria decreased after the Nile perch was introduced because many extinct species of cichlid fish, which were algae-eaters, were no longer present to control algal growth. As the Nile perch population declines, water clarity might improve as the remaining cichlid species that are algae-eaters could help control algal blooms.

Judge reasoning: The answer correctly identifies the causal chain: Nile perch drove cichlid species (including algae-eaters) to extinction, leading to unchecked algae growth and reduced water clarity. It correctly predicts that as Nile perch decline and cichlids reappear, algae control could be restored.

Table 12: Per-question mean composite scores. Only Opus, Qwen, Gemma Q4, Llama Q4, GLM Q4 and DeepSeek are shown for space; full table in supplementary materials.

Question	Opus 4.6	Qwen 2.5	Gemma Q4	Llama Q4	GLM Q4	DeepSeek
F01	5.00	5.00	5.00	4.72	4.78	4.50
F02	5.00	5.00	5.00	5.00	5.00	4.92
F03	5.00	5.00	5.00	5.00	5.00	4.83
F04	5.00	5.00	5.00	5.00	5.00	4.71
R01	5.00	4.72	4.53	4.67	4.69	4.21
R02	5.00	4.42	4.75	4.14	3.86	3.52
R03	5.00	4.14	4.17	4.33	4.00	3.68
R04	5.00	4.78	4.72	4.72	4.58	4.25
S01	5.00	4.86	4.89	4.83	4.61	4.56
S02	5.00	4.53	4.78	4.22	4.44	4.08
S03	4.83	4.67	4.81	4.67	4.58	4.42
S04	5.00	4.56	4.72	4.69	4.42	4.22
C01	4.92	3.44	3.47	2.86	2.56	3.14
C02	4.92	3.22	3.17	3.19	3.00	3.22
C03	4.83	3.19	2.94	3.00	2.83	2.78
C04	4.81	3.00	2.89	2.97	2.72	2.92

Worst local answer

GLM-4-9B-Chat Q4 / reasoning_03 at $t = 0.7$, $p = 0.5$ Score: 9/15 (accuracy 2, completeness 3, coherence 4). 89 tokens, 7.7s.

> The manager should consider adopting TDD now despite the tight deadline. The passage suggests that while late adopters of TDD see smaller benefits (20-40% defect reduction), they still gain some advantages. The benefits include reduced debugging time after release and lower maintenance costs over time. Although there might be a shorter-term delay in completing the initial implementation, the long-term benefits in defect reduction and maintenance costs could outweigh the initial time investment.

Judge reasoning: The answer acknowledges the correct data points (20-40% defect reduction for late adopters, 15-35% longer implementation, long-term maintenance benefits) but reaches a conclusion that contradicts the evidence. The passage indicates that late adoption yields diminished returns and the 15-35% time penalty is problematic with a tight deadline.

Opus non-perfect score

Claude Opus 4.6 / synthesis_03 at $t = 0.0$, $p = 0.5$ Score: 14/15 (accuracy 5, completeness 4, coherence 5). 223 tokens, 5.4s.

The only extractive response where Opus did not receive a perfect score. The judge awarded 4/5 on completeness, noting the answer covered the main challenges and solutions but could have explicitly mentioned the need for improved recycling infrastructure as a distinct path forward.

Llama reasoning example

Llama 3.1 8B Q4 / reasoning_02 at $t = 0.3$, $p = 0.9$ Score: 13/15 (accuracy 4, completeness 4, coherence 5). 156 tokens, 11.8s.

> Based on the provided data, the factors that explain the decline in homeownership among young adults between 1982 and 2020 are:
>
> 1. The increase in the ratio of median home price to median household income, from 3.5:1 to 5.5:1, making homes less affordable.
> 2. The rise in construction costs per square foot, adjusted for inflation, by approximately 40%, contributing to higher home prices.

3. The decrease in median household income, growing by only 15% in real terms, which may not keep pace with increasing housing costs.
4. The decrease in average household size, from 2.76 people to 2.53 people, which may indicate smaller households and potentially fewer people able to afford homes.

Judge reasoning: Correctly identifies the key factors but slightly misstates point 3 by calling the income change a "decrease" when it was actually a modest increase (15% real growth). Point 4, about household size, is speculative rather than directly supported by the passage.

Appendix D: Statistical Analysis

Temperature effect: $t = 0.3$ vs $t = 0.0$

We test whether $t = 0.3$ produces significantly different scores from $t = 0.0$ using a Wilcoxon signed-rank test on paired observations (each pair shares a question and top-p setting on the 12 extractive items). This non-parametric test is appropriate for ordinal Likert-derived scores that may not be normally distributed.

Table 13: Wilcoxon signed-rank test results for $t = 0.3$ vs $t = 0.0$ on extractive items (all eight model configurations).

Model	Mean diff	95% CI	W	p	Sig.
Claude Opus 4.6	+0.009	[−0.009, +0.027]	0	1.0000	n/a
Qwen 2.5 7B	+0.074	[+0.006, +0.143]	2	0.0625	ns
Gemma 4 E4B Q4	−0.065	[−0.121, −0.008]	0	0.0625	ns
Gemma 4 E4B Q8	+0.139	[+0.049, +0.229]	6	0.0137	*
Llama 3.1 8B Q4	−0.019	[−0.118, +0.081]	44	0.9272	ns
Llama 3.1 8B Q5	+0.009	[−0.061, +0.079]	16	0.5547	ns
GLM-4-9B Q4	+0.241	[+0.111, +0.371]	2	0.0006	**
DeepSeek-R1-Distill 7B	+0.038	[−0.212, +0.288]	72	0.5515	ns

Interpretation. The improvement at $t = 0.3$ over $t = 0.0$ reaches statistical significance only for GLM-4-9B Q4 ($p < 0.001$) and Gemma 4 E4B Q8 ($p < 0.05$). Qwen 2.5 7B shows a borderline positive trend ($p = 0.06$), as does Gemma 4 E4B Q4 in the opposite direction (a small *decrease* at $t = 0.3$, $p = 0.06$). The remaining models (Llama Q4, Llama Q5, DeepSeek, Opus) show no detectable difference.

The Section 4.1 claim that most local models achieve their best or near-best composite scores at $t = 0.3$ is supported descriptively for five of seven local configurations. The practical recommendation (use $t = 0.3$ as default) remains sound but with one caveat: Gemma 4 E4B Q4 specifically appears to prefer greedy decoding, suggesting that within the Gemma family, the quantisation level interacts with optimal sampling temperature. This is the clearest evidence in our data that parameter recommendations should be model-and-quantisation-specific rather than universal.

Effect sizes: local models vs frontier

Cohen's d for the composite score gap between each local model and Opus, using pooled standard deviation. Reported separately for extractive Q&A and chain-of-thought to highlight the task-dependent gap structure.

All local-vs-frontier gaps on extractive items are large effects ($d \approx 1.0$–1.3). While the raw gap is modest (0.26–0.58 on a 5-point scale, or 5–12%), the effect is statistically robust given the low variance in Opus extractive scores ($\sigma = 0.067$) versus local model variance ($\sigma = 0.31$–0.86).

On chain-of-thought, the gaps grow 5–7× larger in raw terms (1.48 to 2.09 points) and the effect sizes become enormous ($d = 2.16$–4.26). DeepSeek has the largest CoT gap of any local model tested, confirming the Section 4.4 finding that it is the weakest local CoT model despite its reasoning-distillation provenance. No local model in our evaluation comes close to Opus on multi-step reasoning.

Table 14: Effect sizes (Cohen's d) for each local configuration vs Opus, on extractive items and CoT items separately. † DeepSeek CoT uses the extended-budget measurement (cot_02–04 only, $n = 36$).

Model	Gap (extr.)	d (extr.)	Gap (CoT)	d (CoT)
Qwen 2.5 7B	0.285	0.98	1.478	2.42
Gemma 4 E4B Q4	0.262	1.01	1.811	2.37
Gemma 4 E4B Q8	0.306	1.12	1.790	2.20
Llama 3.1 8B Q4	0.319	1.16	1.563	2.28
Llama 3.1 8B Q5	0.301	1.34	1.563	2.16
GLM-4-9B Q4	0.405	1.19	1.833	2.35
DeepSeek-R1-Distill 7B †	0.583	0.95	2.093	4.26

Score distributions

Score distributions across all parameter combinations, separately for extractive items and CoT items. Failures (timeouts, parse errors marked with negative scores) are excluded from these summaries; raw counts of excluded responses are reported in the supplementary materials.

Table 15: Score distributions on extractive items.

Model	Mean	SD	Min	Max	n
Claude Opus 4.6	4.986	0.067	4.67	5.00	144
Qwen 2.5 7B	4.701	0.406	3.33	5.00	144
Gemma 4 E4B Q4	4.725	0.361	3.67	5.00	144
Gemma 4 E4B Q8	4.681	0.381	3.33	5.00	144
Llama 3.1 8B Q4	4.667	0.385	3.33	5.00	144
Llama 3.1 8B Q5	4.685	0.311	3.67	5.00	144
GLM-4-9B Q4	4.581	0.477	3.00	5.00	144
DeepSeek-R1-Distill 7B	4.403	0.863	1.00	5.00	143

Table 16: Score distributions on chain-of-thought items. † DeepSeek measured under the extended 600 s / 8192-token budget on cot_02–cot_04 only; cot_01 is unmeasurable at any practical budget.

Model	Mean	SD	Min	Max	n
Claude Opus 4.6	4.861	0.318	3.00	5.00	48
Qwen 2.5 7B	3.383	0.805	1.67	5.00	47
Llama 3.1 8B Q4	3.299	0.916	1.33	5.00	48
Llama 3.1 8B Q5	3.299	0.972	1.00	5.00	48
Gemma 4 E4B Q8	3.071	1.107	1.33	5.00	42
Gemma 4 E4B Q4	3.050	1.031	1.00	5.00	47
GLM-4-9B Q4	3.028	1.056	1.33	4.67	48
DeepSeek-R1-Distill 7B †	2.769	0.618	1.67	4.00	36

Opus's extremely low variance on extractive ($\sigma = 0.067$) reflects near-ceiling performance; on CoT, Opus's variance grows nearly 5× to $\sigma = 0.318$, indicating that even the frontier model is no longer at ceiling. Local model variance is 4–13× higher on extractive than Opus, and a further 2–4× higher on CoT than on extractive. The $n < 48$ counts on CoT reflect a small number of parse failures; DeepSeek's $n = 36$ reflects the exclusion of cot_01 (unmeasurable) from the extended re-run.

Appendix E: Hardware Environment

No thermal throttling mitigation was applied. The MacBook Air has a fanless design; sustained inference workloads produce thermal throttling after approximately 10-15 minutes of continuous generation.

Table 17: Hardware and software environment.

Property	Value
System	Apple MacBook Air (2022)
Chip	Apple M2
CPU cores	8 (4 performance, 4 efficiency)
GPU cores	8
Neural Engine cores	16
Unified memory	16 GB
Memory bandwidth	100 GB/s
SSD	256 GB
macOS version	Sequoia 15.4.1
llama.cpp version	b5270 (Homebrew)
GPU offload	`-ngl 99` (all layers)
Context window	4096 tokens
Python	3.12
Anthropic SDK	0.49.x
OpenAI SDK (for llama-server)	1.x

Latency measurements represent averages across the full sweep (including any throttled periods) and should be treated as representative rather than guaranteed.

References

[1] G. Gerganov, "llama.cpp." GitHub, 2023. Available: https://github.com/ggerganov/llama.cpp

[2] J. Wang, Y. Zeng, J. Guo, Y. Ma, A. Liu, and X. Liu, "SLMQuant: Benchmarking Small Language Model Quantization for Practical Deployment," in *ACM rich media with generative AI workshop*, 2025. doi: 10.1145/3746262.3761973.

[3] "Which Quantization Should I Use? A Unified Evaluation of llama.cpp Quantization on Llama-3.1-8B-Instruct," *arXiv preprint arXiv:2601.14277*, 2026, doi: 10.48550/arXiv.2601.14277.

[4] "A Comprehensive Evaluation of Quantization Strategies for Large Language Models," in *Findings of the association for computational linguistics: ACL 2024*, 2024. doi: 10.18653/v1/2024.findings-acl.726.

[5] M. Renze and E. Guven, "The Effect of Sampling Temperature on Problem Solving in Large Language Models," *arXiv preprint arXiv:2402.05201*, 2024, doi: 10.48550/arXiv.2402.05201.

[6] "Exploring the Impact of Temperature on Large Language Models: Hot or Cold?" *Procedia Computer Science*, 2025, doi: 10.48550/arXiv.2506.07295.

[7] V. Rajesh, O. Jodhpurkar, P. Anbuselvan, M. Singh, A. Jallepali, *et al.*, "Production-Grade Local LLM Inference on Apple Silicon: A Comparative Study of MLX, MLC-LLM, Ollama, llama.cpp, and PyTorch MPS," *arXiv preprint arXiv:2511.05502*, 2025, doi: 10.48550/arXiv.2511.05502.

[8] "Benchmarking and Characterization of Large Language Model Inference on Apple Silicon," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2025, doi: 10.1145/3771563.

[9] L. Zheng *et al.*, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," in *Advances in neural information processing systems: Datasets and benchmarks track*, 2023. doi: 10.48550/arXiv.2306.05685.

[10] A. Panickssery, S. Bowman, and S. Feng, "Self-Preference Bias in LLM-as-a-Judge," in *Advances in neural information processing systems*, 2024. doi: 10.48550/arXiv.2410.21819.

[11] DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," *arXiv preprint arXiv:2501.12948*, 2025, doi: 10.48550/arXiv.2501.12948.

[12] "We ran over half a million evaluations on quantized LLMs: here's what we found." Red Hat Developer, 2024. Available: https://developers.redhat.com/articles/2024/10/17/we-ran-over-half-million-evaluations-quantized-llms

[13] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale," in *Advances in neural information processing systems*, 2022. doi: 10.48550/arXiv.2208.07339.

[14] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers," in *International conference on learning representations*, 2023. doi: 10.48550/arXiv.2210.17323.

[15] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models," in *Proceedings of the 40th international conference on machine learning*, 2023. Available: https://proceedings.mlr.press/v202/xiao23c.html

[16] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, *et al.*, "AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration," *Proceedings of Machine Learning and Systems*, 2024, doi: 10.48550/arXiv.2306.00978.

[17] R. Gong, Y. Ding, Z. Wang, C. Lv, X. Zheng, *et al.*, "A Survey of Low-bit Large Language Models: Basics, Systems, and Algorithms," *Neural Networks*, 2025, doi: 10.1016/j.neunet.2025.107856.

[18] "Quantization Methods, Task Difficulty, and Model Size in Large Language Models," in *Proceedings of the international joint conference on artificial intelligence*, 2025. Available: https://www.ijcai.org/proceedings/2025/0902.pdf

[19] Z. Lu, X. Wang, B. Guo, and J. Gao, "Small Language Models: Survey, Measurements, and Insights," *arXiv preprint arXiv:2409.15790*, 2024, doi: 10.48550/arXiv.2409.15790.

[20] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The Curious Case of Neural Text Degeneration," in *International conference on learning representations*, 2020. doi: 10.48550/arXiv.1904.09751.

[21] G. Wiher, C. Meister, and R. Cotterell, "On Decoding Strategies for Neural Text Generators," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 997–1012, 2022, doi: 10.1162/tacl_a_00502.

[22] J. Hewitt, C. D. Manning, and P. Liang, "Truncation Sampling as Language Model Desmoothing," in *Findings of the association for computational linguistics: EMNLP 2022*, 2022. doi: 10.18653/v1/2022.findings-emnlp.249.

[23] N. N. Minh, A. Baker, C. Neo, A. Roush, A. Kirsch, and R. Shwartz-Ziv, "Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs," *arXiv preprint arXiv:2407.01082*, 2024, doi: 10.48550/arXiv.2407.01082.

[24] M. Renze and E. Guven, "The Good, The Bad, and The Greedy: Evaluation of LLMs Should Not Ignore Non-Determinism," in *Proceedings of the 2025 conference of the north american chapter of the association for computational linguistics*, 2025. doi: 10.48550/arXiv.2407.10457.

[25] "Exploring Temperature Effects on Large Language Models Across Various Clinical Tasks," *medRxiv*, 2024, doi: 10.1101/2024.07.22.24310824.

[26] Y. Zhu and J. Li, "Improving Code Generation by Dynamic Temperature Sampling," *arXiv preprint*, 2024, Available: https://www.semanticscholar.org/paper/702486c79cdd061d0cfab0e91fe633e831e004a2

[27] P.-H. Wang and C.-J. Hsieh, "Contextual Temperature for Language Modeling," *arXiv preprint*, 2024, Available: https://www.semanticscholar.org/paper/f617f7ba4040d6e85b384685da09fed35c841280

[28] J. Li, C. Sun, S. Mao, Z. Hu, and L. Li, "A Survey on LLM-as-a-Judge," *arXiv preprint arXiv:2411.15594*, 2024, doi: 10.48550/arXiv.2411.15594.

[29] H. Li, Q. Chen, J. Zhang, G. Wang, and Y. Li, "LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods," *arXiv preprint arXiv:2412.05579*, 2024, doi: 10.48550/arXiv.2412.05579.

[30] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, *et al.*, "A Survey on LLM-as-a-Judge," *The Innovation*, 2026, doi: 10.1016/j.xinn.2025.101253.

[31] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, *et al.*, "Large Language Models are not Fair Evaluators," in *Proceedings of the 62nd annual meeting of the association for computational linguistics*, 2024. Available: https://aclanthology.org/2024.acl-long.511/

[32] P. Wang *et al.*, "Judging the Judges: A Systematic Study of Position Bias in LLM-as-a-Judge," *arXiv preprint arXiv:2406.07791*, 2024, doi: 10.48550/arXiv.2406.07791.

[33] S. Dev, A. Sloan, J. Kavner, N. Kong, and M. Sandler, "Judge Reliability Harness: Stress Testing the Reliability of LLM Judges," *arXiv preprint arXiv:2603.05399*, 2026, doi: 10.48550/arXiv.2603.05399.

[34] "An Empirical Study of LLM-as-a-Judge: How Design Choices Impact Evaluation Reliability," *arXiv preprint arXiv:2506.13639*, 2025, doi: 10.48550/arXiv.2506.13639.

[35] K. Saito, "Verbosity Bias in Preference Labeling by Large Language Models," *arXiv preprint arXiv:2310.10076*, 2023, doi: 10.48550/arXiv.2310.10076.

[36] "Explaining Length Bias in LLM-Based Preference Evaluations," *arXiv preprint arXiv:2407.01085*, 2024, doi: 10.48550/arXiv.2407.01085.

[37] W. Barrios, "Native LLM and MLLM Inference at Scale on Apple Silicon," *arXiv preprint arXiv:2601.19139*, 2026, doi: 10.48550/arXiv.2601.19139.

[38] Z. Bi, X. Chen, L. Sun, Y. Yao, Q. Shen, *et al.*, "RooflineBench: A Benchmarking Framework for On-Device LLMs via Roofline Analysis," *arXiv preprint arXiv:2602.11506*, 2026, doi: 10.48550/arXiv.2602.11506.

[39] "On-device Llama 3.1 with Core ML." Apple Machine Learning Research, 2024. Available: https://machinelearning.apple.com/research/core-ml-on-device-llama

[40] "Understanding Large Language Models in Your Pockets: Performance Study on COTS Mobile Devices," *arXiv preprint arXiv:2410.03613*, 2024, doi: 10.48550/arXiv.2410.03613.

[41] "Open-weight models lag state-of-the-art by around 3 months on average." Epoch AI, 2025. Available: https://epoch.ai/data-insights/open-weights-vs-closed-weights-models/

[42] "Frontier AI capabilities can be run at home within a year or less." Epoch AI, 2025. Available: https://epoch.ai/data-insights/consumer-gpu-model-gap/

[43] X. Yang, K. Sun, H. Xin, Y. Sun, N. Bhalla, *et al.*, "CRAG – Comprehensive RAG Benchmark," in *Advances in neural information processing systems: Datasets and benchmarks track*, 2024. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/1435d2d0fca85a84d83ddcb754f58c29-Abstract-Datasets_and_Benchmarks_Track.html

[44] I. T. Sorodoc, L. F. R. Ribeiro, R. Blloshmi, C. Davis, and A. de Gispert, "GaRAGe: A Benchmark with Grounding Annotations for RAG Evaluation," in *Findings of the association for computational linguistics: ACL 2025*, 2025. doi: 10.18653/v1/2025.findings-acl.875.

[45] "Sample Smart, Not Hard: Correctness-First Decoding for Better Reasoning in LLMs," *arXiv preprint arXiv:2510.05987*, 2025, doi: 10.48550/arXiv.2510.05987.

[46] "Verbosity $\neq$ Veracity: Demystify Verbosity Compensation Behavior of Large Language Models," *arXiv preprint arXiv:2411.07858*, 2024, doi: 10.48550/arXiv.2411.07858.

[47] "Mitigating Length Bias in RLHF through a Causal Lens," *arXiv preprint arXiv:2511.12573*, 2025, doi: 10.48550/arXiv.2511.12573.

[48] "How Reliable is Multilingual LLM-as-a-Judge?" in *Findings of the association for computational linguistics: EMNLP 2025*, 2025. doi: 10.18653/v1/2025.findings-emnlp.587.