Beyond Parameter Count: Selecting Compact Language Models for Retrieval-Augmented Generation

Alexandros Papafragkakis
Department of Computer Science, University of Crete
Heraklion, Greece
Institute of Computer Science, Foundation for Research and Technology – Hellas
Heraklion, Greece

Yannis Tzitzikas
Department of Computer Science, University of Crete
Heraklion, Greece
Institute of Computer Science, Foundation for Research and Technology – Hellas
Heraklion, Greece

Abstract

Production deployment of Retrieval-Augmented Generation (RAG) systems requires balancing answer quality with computational constraints and response latency. While existing research has extensively investigated retrieval strategies and prompt engineering, the selection of the language model component remains relatively unexplored, particularly for resource-constrained scenarios where compact models (1B–8B parameters) offer practical advantages. This work systematically evaluates six compact open-source models within a controlled RAG pipeline across two complementary knowledge-intensive QA benchmarks. Our evaluation, using fixed retrieval configuration across over 16,400 test questions, reveals that, within the heterogeneous set of widely-deployed compact families we evaluate, parameter count alone is an unreliable predictor of RAG performance, even after controlling for retrieval quality. On MetaQA, Gemma 2 2B and Mistral 7B achieve statistically indistinguishable accuracy (81.66% vs 81.45%, McNemar $p = 0.588$) despite a 3.5× difference in parameters. On WC2014QA, Mistral 7B *significantly outperforms* Llama 3.1 8B (92.70% vs 89.28%, $p < 0.001$). Latency decomposition reveals that retrieval latency is relatively uniform across models, while generation latency varies substantially with architectural choices: Phi-3 Mini and Mistral 7B exhibit the fastest generation. The key takeaway is that strategic model selection can achieve higher accuracy than the largest evaluated model while reducing end-to-end latency by approximately 24% (Mistral 7B: 0.61s vs Llama 3.1 8B: 0.80s on WC2014QA). All accuracy comparisons are supported by McNemar significance tests.

CCS Concepts

• **Computing methodologies** → **Natural language generation**; • **Information systems** → **Question answering**; *Retrieval models and ranking*.

Keywords

retrieval-augmented generation, compact language models, efficiency-accuracy tradeoffs, question answering, benchmarking

1 Introduction

Large language models can generate fluent text but may produce unsupported statements when factual grounding is required. This occurs because model parameters are frozen post-training and knowledge is derived entirely from static training corpora. RAG addresses this limitation through a two-stage approach [1]: first retrieving relevant documents from local or external sources, then conditioning generation on these retrieved documents to ground responses in verifiable evidence. Recent surveys document the wide applicability and rapid evolution of RAG paradigms [24].

Most RAG research focuses on large-scale models (70B+ parameters) deployed on multi-GPU infrastructure. Such deployments are prohibitively expensive for resource-constrained organizations or mobile/edge applications. Several compact models have emerged recently, including Phi-3 Mini (3.8B), Gemma 2 (2B), and Llama 3.2 (1B and 3B variants). These can execute on single consumer GPUs with sub-second latencies and operational costs orders of magnitude lower than larger counterparts. Yet it remains unclear which compact models perform best in RAG contexts.

The literature provides limited guidance here. Existing studies typically evaluate single models in isolation, compare vastly different scales where conclusions do not transfer, or rely on proprietary APIs preventing reproducibility. We lack systematic understanding of how 2B vs 3B vs 7B models compare under the same retrieval setup, including *retriever*, *reranker*, and *prompt template*. Practitioners deploying RAG systems face this choice regularly but lack empirical data.

This paper addresses this gap through controlled evaluation of six compact models (Llama 3.2 1B, Gemma 2 2B, Llama 3.2 3B, Phi-3 Mini 3.8B, Mistral 7B, Llama 3.1 8B) in a controlled RAG pipeline with a fixed retrieval configuration and prompt template. We hold the retrieval and prompting components fixed (same embedder, hybrid search, reranker, and prompt template), so observed differences primarily reflect the generator model choice. Evaluation uses two widely used and complementary benchmarks: MetaQA [2] with 9,947 single-hop test questions requiring reasoning over movie data, and WC2014QA [3] with 6,482 single-hop test questions about the 2014 FIFA World Cup requiring precise entity extraction. We measure both accuracy (Hits@1, F1, Exact Match) and latency (retrieval, generation, end-to-end), and accompany all pairwise accuracy comparisons with McNemar significance tests [25].

Our contributions are fourfold: (1) a systematic comparison of six compact LMs in a controlled RAG pipeline with fixed retrieval configuration, isolating model effects from retrieval and prompting; (2) evaluation on two complementary benchmarks covering entity-centric single-hop QA over heterogeneous sources; (3) a joint accuracy–efficiency analysis with latency decomposition, supported by McNemar significance tests for all pairwise comparisons, identifying which accuracy differences are statistically reliable; and (4) deployment-oriented insights, including the finding that on our

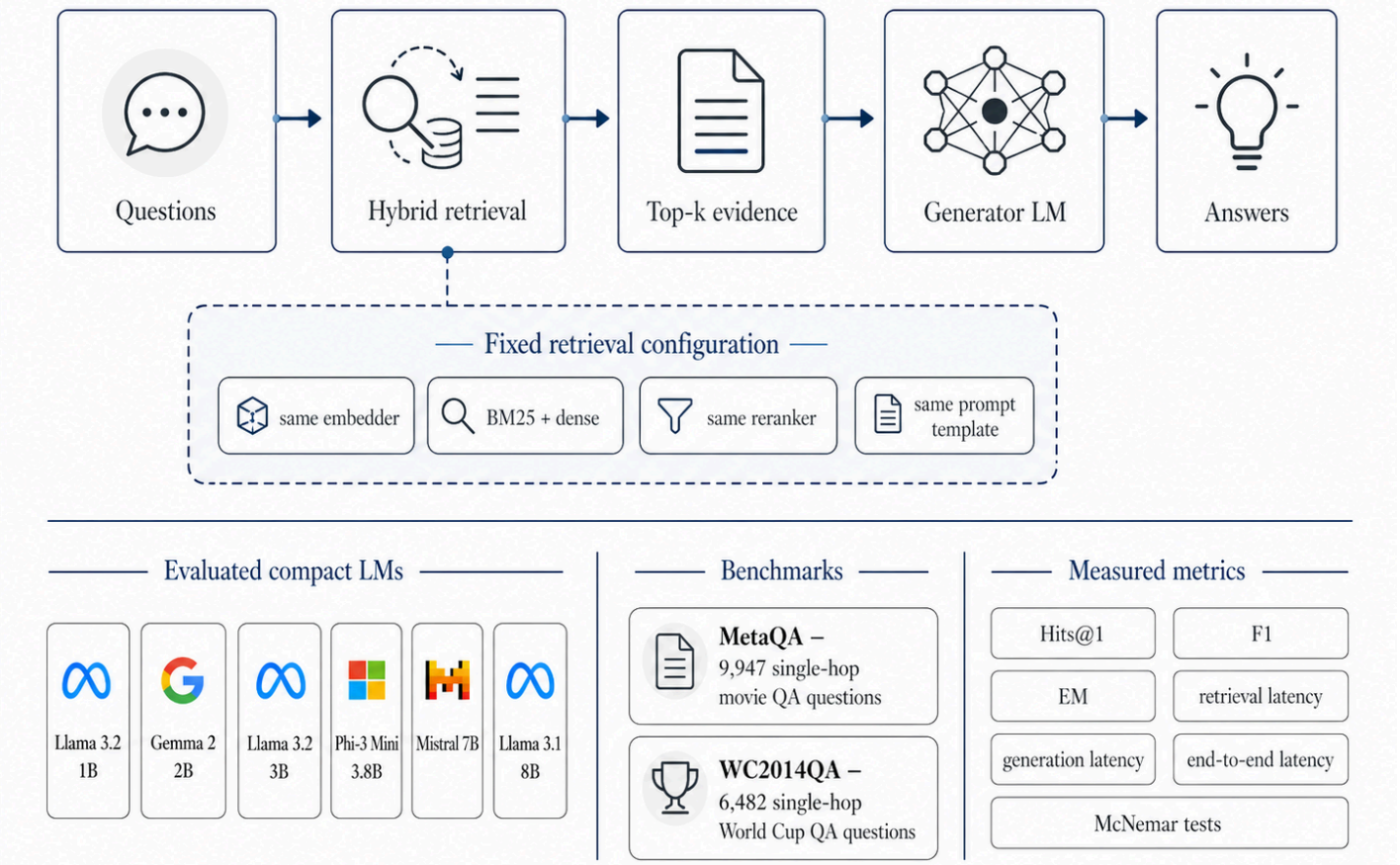

Figure 1: Overview of the study: controlled RAG pipeline with a fixed retrieval configuration, six evaluated compact LMs, two single-hop QA benchmarks, and the measured accuracy and latency quantities.

benchmarks Mistral 7B lies on the accuracy–latency efficiency frontier, matching or exceeding Llama 3.1 8B accuracy at substantially lower latency, and that retrieval latency is largely independent of generator size under a fixed retrieval configuration. Beyond aggregate accuracy and latency rankings, our per-question paired McNemar analysis reveals that the largest evaluated model is rarely Pareto-dominant on the accuracy–latency frontier, motivating model selection that accounts for architecture—not just size—as a first-class deployment decision rather than an afterthought. An illustration of the investigated task and its context is given in Figure 1.

The rest of this paper is organized as follows. Section 2 reviews background and related work on retrieval-augmented generation and compact language models. Section 3 details the experimental methodology, including the controlled RAG pipeline, the evaluated models, benchmarks, metrics, and statistical procedure. Section 4 reports accuracy and latency results across both benchmarks, accompanied by pairwise McNemar significance tests and an analysis of the accuracy–latency frontier. Section 5 discusses deployment-relevant observations, including error patterns and possible mechanistic explanations for the observed non-monotonic scaling. Section 6 consolidates practical deployment guidance, Section 7 discusses limitations, and Section 8 concludes.

2 Background and Related Work

Our work builds on two research areas: retrieval-augmented generation and compact language model development.

RAG Systems. Large language models generate text based solely on training data. When asked about recent publications or domain-specific facts, they either acknowledge limitations or fabricate details. RAG systems split the task: retrieve relevant documents from local or external sources, then condition generation on retrieved documents [1]; recent surveys provide a broader overview of the rapidly evolving RAG landscape [24]. Rather than relying on memorized patterns, the model extracts answers from provided reference text. Standard RAG pipelines have three components. During indexing, documents are chunked, embedded via sentence transformers [19], and stored in vector databases alongside lexical indexes [20]. At query time, retrieval searches for passages semantically similar to the question, often combining dense and sparse methods [4]. Generation formats passages as context, prepending them to questions before feeding to the LLM.

Most RAG research optimizes retrieval: improved embeddings, hybrid ranking, and multi-hop strategies for complex questions [5]. The generator receives less attention, typically treated as fixed. Researchers select a large model, optimize retrieval parameters, and

evaluate the complete system. This leaves one question relatively unexplored: does generator choice matter when high-quality context is provided, and how does it interact with latency constraints?

Compact Language Models. The evaluated models represent different design philosophies. Mistral 7B [6] employs grouped-query attention and sliding-window attention for efficient 8K-token context handling, achieving performance rivaling models twice its size. Gemma 2 2B [7] uses local attention patterns and GeGLU activations to run on minimal hardware. Phi-3 Mini [8] (3.8B) emphasizes high-quality training data over scale, focusing on structured reasoning relevant to RAG tasks. Llama 3.2 [9] offers 1B for edge deployment and 3B as middle ground, inheriting Llama 3 architectural improvements. These span an order of magnitude (1B to 8B) but share constraints: consumer hardware, interactive latency, and minimal memory. Whether architectural differences translate to meaningful RAG performance gaps remains unclear.

Related Work and Gap. Lewis et al. [1] introduced RAG combining DPR with BART, demonstrating gains on the benchmarks Natural Questions [10] and TriviaQA [11]. Subsequent work expanded along multiple dimensions: contrastive learning for retrievers [4] and iterative retrieval for multi-hop questions [5]. Parallel research on compact models includes distillation approaches (DistilBERT [12], TinyBERT [13]) and data curation methods (Phi-2 [14], StableLM [15]). However, these focus on general benchmarks (MMLU [16], reasoning tasks) rather than RAG-specific evaluation. No prior work systematically compares compact models under a controlled RAG setup with statistical significance testing. Studies either compare single models against baselines [17], test vastly different scales where conclusions do not generalize [18], or use proprietary APIs preventing reproduction.

3 Experimental Methodology

We design a controlled comparison isolating generator effects from retrieval quality.

Experimental Infrastructure. All experiments use SemanticRAG [26], an existing open-source RAG pipeline, as experimental infrastructure, with no changes to retrieval or indexing components. We access the pipeline through its HTTP API endpoint, keeping the retrieval and indexing components fixed and only swapping the generator LLM parameter. This controlled setup ensures that observed performance differences stem solely from model selection rather than pipeline variations. Latency measurements include network overhead from HTTP communication.[1]

Models. Six compact models are evaluated: Llama 3.2 1B, Gemma 2 2B, Llama 3.2 3B, Phi-3 Mini 3.8B, Mistral 7B, and Llama 3.1 8B. All models are accessed with identical retrieval configuration (embeddings, hybrid search, reranking).

Retrieval Pipeline. The retrieval component uses dense embeddings (nomic-embed-text [21], 768-dim) combined with BM25 [20] for hybrid search. Documents are chunked into 500-character segments with 50-character overlap. Per question, the system retrieves candidate passages via hybrid search and applies cross-encoder reranking (gte-reranker-modernbert-base). The reranked passages are provided as context for generation (top-10 target). Generation models receive context plus question, producing answers with temperature=0 for deterministic output.

Benchmarks. MetaQA [2] contains movie questions at three difficulty levels (1-hop, 2-hop, and 3-hop reasoning over the underlying knowledge graph); we evaluate only on the 9,947 single-hop test questions. Multi-hop questions would require iterative retrieval, confounding our goal of isolating generator performance under fixed retrieval context. Therefore, restricting evaluation to 1-hop questions is well-suited to this research goal. The knowledge base consists of 43,234 movie entities derived from MetaQA triples, verbalized into natural language passages. WC2014QA tests factual extraction about the 2014 FIFA World Cup using 6,482 single-hop test questions with 1,236 supporting passages from news reports and match summaries.

Metrics. We report three accuracy metrics on normalized text; because the two benchmarks have different answer structures, the exact procedure differs slightly between them. In WC2014QA each question has a set of valid answers and predictions are single entities: we normalize a prediction by lowercasing, replacing spaces with underscores, and truncating at the first comma. Hits@1 counts the question correct if the normalized prediction is exactly equal to *any* valid answer (an exact-match membership test, not a substring test); EM compares it against the single first-listed valid answer only; and F1 is the token-level overlap between the prediction and the best-matching valid answer, averaged over questions. In MetaQA answers are genuine entity sets: we lowercase, strip punctuation, and split predictions and gold answers on "|" into entity sets. Hits@1 counts the question correct if the top predicted entity matches any gold entity; EM requires the predicted entity set to equal the gold set exactly; and F1 is the entity-level (set-overlap) F1 between the two sets, averaged over questions. In both benchmarks EM is the strictest metric and Hits@1 the most lenient, so EM $\leq$ Hits@1 by construction; the gap is largest on set-valued questions, where returning one valid member is credited by Hits@1 and partially by F1 but rejected by EM. Latency decomposes into retrieval time, generation time, and end-to-end time per query. We additionally apply McNemar's test [25] with continuity correction for all pairwise model comparisons on per-question binary correctness, reporting p-values with significance codes (*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, n.s. not significant). We report uncorrected p-values for transparency; given 15 pairwise comparisons per dataset, results with $p < 0.0033$ (Bonferroni-corrected at α=0.05) should be considered statistically robust to concerns about multiple comparisons. As all our headline findings (the M-7B vs L3.1-8B gap on WC2014QA, the L3.2-1B vs G2-2B gap on WC2014QA, and the G2-2B vs M-7B null on MetaQA) are reported at $p < 0.001$ or $p > 0.5$, they survive Bonferroni correction.

Procedure. For each model, we process all test questions under identical retrieval configuration. Retrieval is executed independently per run; retrieval latency and aggregate retrieval behavior are largely stable across runs and models (Figure 3), indicating that observed performance differences are primarily attributable to the generator.

Implementation Details. The remote API uses 4-bit quantization (bitsandbytes NF4 [22]) for all models. Individual queries

[1]Our evaluation code is publicly available at https://github.com/APapafragkakis/Compact-LLMs-RAG-Eval.

timeout at 60s; failed queries retry up to 3 times. Decoding is deterministic (temperature=0, max_new_tokens=128) with sequential single-query inference (batch size 1).

Latency Measurement Setup. Latency is measured on the client side via Python's `perf_counter()`, capturing the HTTP request-response round-trip to a remote inference server hosted on research infrastructure with a single 16 GB consumer-class GPU. Reported latencies therefore reflect end-to-end remote-API behavior (network + retrieval + generation), not local on-device inference. We report mean latency over the test sets after 5 warm-up queries. Latency is decomposed into retrieval service latency (embedding, hybrid search, reranking) and generation service latency (forward pass, decoding). Each model was evaluated in a separate single run; we observe non-negligible run-to-run variance in retrieval latency (up to roughly 17% across models on WC2014QA, despite identical retrieval inputs across runs), which we attribute to network and server-load fluctuations. Latency comparisons should therefore be read as point estimates rather than tightly-bounded measurements; in particular, the 24% end-to-end latency advantage of Mistral 7B over Llama 3.1 8B on WC2014QA is dominated by a much larger gap in generation latency (0.326 s vs 0.505 s) and is therefore robust to retrieval-side variance. Smaller latency differences between similar architectures should not be over-interpreted, as they fall within the run-to-run noise we observe.

Prompt Format and Normalization. Retrieved passages are formatted as a bulleted list under "Documents:" with task instructions to extract answers and respond with the exact entity name. For evaluation, we apply dataset-specific lightweight normalization: right-strip whitespace, replace spaces with underscores, lowercase, truncate at the first comma. Gold answers are provided in underscored format and lowercased for comparison. The precise definitions of EM, F1, and Hits@1 are consolidated in the Metrics paragraph above. For WC2014QA specifically, since some questions admit multiple valid answers (e.g., naming any player on a team), Hits@1 and F1 are computed against the best match across all valid answers, while EM compares only against the first-listed gold answer; this convention explains the systematic EM < Hits@1 gap on WC2014QA, which is most pronounced for set-valued questions.

A question is *set-valued* when several distinct entities are acceptable answers (e.g. "which soccer club is based in Mexico", or "what films does an actor appear in"). Set-valued questions are approximately 33% of single-hop MetaQA (mean 1.9, maximum 44 acceptable answers) and 36% of WC2014QA, the latter with a much heavier tail (mean 30.8, maximum 249 acceptable answers). For such questions Hits@1 and F1 are the appropriate metrics, as they credit any correct member (Hits@1) or partial set recovery (F1); EM, which on WC2014QA compares against a single first-listed answer, is not a meaningful criterion for set-valued questions and is reported only for completeness and comparability with prior work.

4 Results

We report accuracy and latency across both benchmarks for all six models, with pairwise statistical significance (McNemar's test) for all accuracy comparisons.

4.1 Accuracy

Tables 1 and 2 report baseline and RAG performance for all six models on MetaQA and WC2014QA respectively.

Table 1: MetaQA: Baseline vs RAG performance (N=9,947). All metrics in %. H@1 = Hits@1, EM = Exact Match. Best in bold.

	Baseline			RAG		
Model	H@1	F1	EM	H@1	F1	EM
L3.2-1B	5.49	5.77	0.80	72.56	59.42	29.96
G2-2B	18.53	15.73	6.39	81.66	79.91	67.96
L3.2-3B	23.52	19.11	7.61	85.34	78.52	60.53
Phi-3	7.96	6.07	0.69	79.42	71.89	50.34
M-7B	31.83	26.62	8.01	81.45	79.10	**68.32**
L3.1-8B	36.19	31.07	15.02	**86.15**	**81.35**	66.53

Table 2: WC2014QA: Baseline vs RAG performance (N=6,482). All metrics in %.

	Baseline			RAG		
Model	H@1	F1	EM	H@1	F1	EM
L3.2-1B	6.03	9.63	1.50	75.35	83.02	51.08
G2-2B	12.90	16.90	8.69	69.38	76.00	63.73
L3.2-3B	13.11	17.44	8.81	73.76	80.87	60.94
Phi-3	11.34	16.33	6.08	84.85	90.43	58.87
M-7B	8.21	15.65	7.16	**92.70**	**94.19**	64.12
L3.1-8B	20.66	25.18	11.60	89.28	92.20	**64.89**

MetaQA. Llama 3.1 8B achieves the highest Hits@1 at 86.15%, followed closely by Llama 3.2 3B at 85.34%. The 0.80pp gap, while small, is statistically significant ($p < 0.001$) given the 9,947-question test set (Table 3). Phi-3 Mini (79.42%) underperforms the smaller Llama 3.2 3B (85.34%) by 5.92pp, a statistically significant effect that is non-monotonic in parameter count ($p < 0.001$). Notably, *Gemma 2 2B (81.66%) and Mistral 7B (81.45%) are statistically indistinguishable* on MetaQA ($p = 0.588$), despite Mistral having 3.5× the parameters.

WC2014QA. Mistral 7B achieves the highest accuracy at 92.70%, significantly exceeding Llama 3.1 8B (89.28%, $p < 0.001$). Llama 3.2 1B reaches 75.35% versus 69.38% for Gemma 2 2B ($p < 0.001$), outperforming it despite having half the parameters; this illustrates that model family and training data can dominate size effects. Every one of the 15 pairwise differences on WC2014QA reaches statistical significance.

Baseline vs. RAG. All models show highly significant gains from RAG ($p < 0.001$ for all 12 baseline-vs-RAG comparisons). On MetaQA, absolute gains range from +49.62pp (Mistral 7B) to +71.46pp (Phi-3 Mini). On WC2014QA, gains range from +56.48pp (Gemma 2 2B) to +84.50pp (Mistral 7B). In relative terms, the smallest model (Llama 3.2 1B) shows the largest ratio gain on MetaQA (13.2×); on WC2014QA both Llama 3.2 1B (12.5×) and Mistral 7B (11.3×) show the highest relative gains.

4.2 Statistical Significance

We apply McNemar's test with continuity correction to each pair of models, using per-question Hits@1 correctness (Tables 3, 4).

Table 3: MetaQA pairwise McNemar tests (Hits@1, N=9,947). Δ in percentage points. Significance: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, n.s. not significant.

Comparison	Δ (pp)	p-value	Sig.
L3.2-1B vs G2-2B	+9.10	< 0.001	***
L3.2-1B vs L3.2-3B	+12.78	< 0.001	***
L3.2-1B vs Phi-3	+6.86	< 0.001	***
L3.2-1B vs M-7B	+8.89	< 0.001	***
L3.2-1B vs L3.1-8B	+13.58	< 0.001	***
G2-2B vs L3.2-3B	+3.68	< 0.001	***
G2-2B vs Phi-3	−2.24	1.3×10^{-8}	***
G2-2B vs M-7B	**−0.21**	**0.588**	**n.s.**
G2-2B vs L3.1-8B	+4.48	< 0.001	***
L3.2-3B vs Phi-3	−5.92	< 0.001	***
L3.2-3B vs M-7B	−3.89	< 0.001	***
L3.2-3B vs L3.1-8B	+0.80	1.5×10^{-4}	***
Phi-3 vs M-7B	+2.03	3.6×10^{-8}	***
Phi-3 vs L3.1-8B	+6.73	< 0.001	***
M-7B vs L3.1-8B	+4.69	< 0.001	***

Table 4: WC2014QA pairwise McNemar tests (Hits@1, N=6,482). Δ in percentage points.

Comparison	Δ (pp)	p-value	Sig.
L3.2-1B vs G2-2B	−5.97	3.4×10^{-15}	***
L3.2-1B vs L3.2-3B	−1.59	0.033	*
L3.2-1B vs Phi-3	+9.50	< 0.001	***
L3.2-1B vs M-7B	+17.36	< 0.001	***
L3.2-1B vs L3.1-8B	+13.93	< 0.001	***
G2-2B vs L3.2-3B	+4.38	< 0.001	***
G2-2B vs Phi-3	+15.47	< 0.001	***
G2-2B vs M-7B	+23.33	< 0.001	***
G2-2B vs L3.1-8B	+19.90	< 0.001	***
L3.2-3B vs Phi-3	+11.09	< 0.001	***
L3.2-3B vs M-7B	+18.94	< 0.001	***
L3.2-3B vs L3.1-8B	+15.52	< 0.001	***
Phi-3 vs M-7B	+7.85	< 0.001	***
Phi-3 vs L3.1-8B	+4.43	2.2×10^{-16}	***
M-7B vs L3.1-8B	**−3.42**	$\mathbf{1.9 \times 10^{-13}}$	***

Key takeaways from significance testing. On MetaQA, only one pair (G2-2B vs M-7B) is not statistically distinguishable; even the small 0.80pp gap between L3.2-3B and L3.1-8B is significant given the large test set. On WC2014QA, all 15 pairwise differences are significant; of particular deployment interest, *Mistral 7B significantly outperforms Llama 3.1 8B* ($\Delta = -3.42$pp, $p < 0.001$) and *Llama 3.2 1B significantly outperforms Gemma 2 2B* ($\Delta = +5.97$pp, $p < 0.001$).

4.3 Latency

Figure 2 visualizes the accuracy comparison between baseline and RAG configurations. Figure 3 shows per-query latency decomposed into retrieval and generation components. Under our fixed retrieval configuration, retrieval latency is relatively uniform across models (0.29–0.46s), reflecting identical embedding, indexing, and reranking. Because the retrieval pipeline is byte-for-byte identical across generators, this uniformity is expected by design rather than an empirical finding about the generators: we report it as a control confirming that our setup isolates the generator, and the residual run-to-run variation (Section 3) is attributable to network and server load, not to the choice of generator. Generation latency varies substantially and is primarily determined by model architecture rather than raw parameter count.

On WC2014QA, Phi-3 Mini (0.618s end-to-end) and Mistral 7B (0.614s) are the fastest, despite Mistral having nearly twice the parameters of Phi-3. Llama 3.1 8B is the slowest at 0.804s, yielding a **24% latency reduction** for Mistral 7B while achieving significantly higher accuracy. On MetaQA, Phi-3 Mini (0.689s) and Mistral 7B (0.721s) are again the fastest, primarily due to their markedly lower generation latencies (0.236s and 0.275s, respectively) compared to Llama-family models (0.457–0.485s).

4.4 The Accuracy–Latency Frontier

On WC2014QA, Mistral 7B strictly Pareto-dominates Llama 3.1 8B: higher accuracy (+3.42pp, $p < 0.001$) at lower latency (−0.19s). On MetaQA, the Pareto frontier spans four models: Phi-3 Mini anchors the low-latency end (0.689s, 79.42%); Mistral 7B (statistically tied with Gemma 2 2B) offers 94.6% of 8B accuracy at 21% lower latency; Llama 3.2 3B offers 98.6% of 8B accuracy at comparable latency; and Llama 3.1 8B achieves the highest accuracy but also the highest latency.

5 Analysis

Our results, backed by statistical testing, yield four deployment-relevant observations.

(1) Parameter count is a weak predictor of RAG performance within the compact regime. On MetaQA, Gemma 2 2B matches Mistral 7B (p=0.588). On WC2014QA, Llama 3.2 1B beats Gemma 2 2B ($p < 0.001$) and Mistral 7B beats Llama 3.1 8B ($p < 0.001$). These are statistically reliable effects, not point-estimate noise, and survive Bonferroni correction. We emphasize an important caveat: in our setup model family is confounded with size—only two of our six models (Llama 3.2 1B and 3B) share architecture and training pipeline with another evaluated model. A controlled size-only study would require a full within-family sweep, which we discuss as a limitation. What our findings *do* establish is more limited than a pure scaling claim: across the heterogeneous set of compact LM families in wide deployment that practitioners actually choose between, raw parameter count cannot be used as a reliable proxy for RAG accuracy.

(2) Retrieval latency is generator-independent; generation latency is architecture-dependent. Our fixed retrieval configuration produces nearly constant retrieval times (0.29–0.46s) across all models. Generation latency, by contrast, is not a monotonic function of size: Phi-3 Mini (3.8B) and Mistral 7B (7B) generate

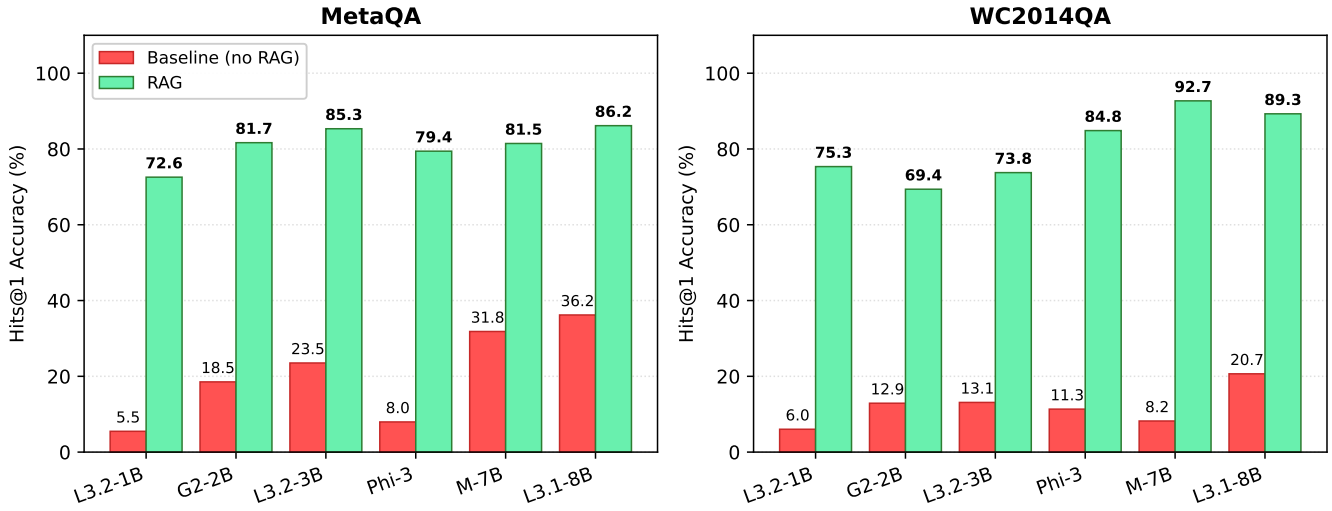

Figure 2: Hits@1 accuracy comparison: baseline (no RAG) vs. RAG-augmented configurations. All models show substantial and statistically significant gains from RAG on both benchmarks ($p < 0.001$). The smallest model (Llama 3.2 1B) improves by up to 13.2× on MetaQA, while Mistral 7B achieves the largest absolute improvement on WC2014QA (+84.50pp).

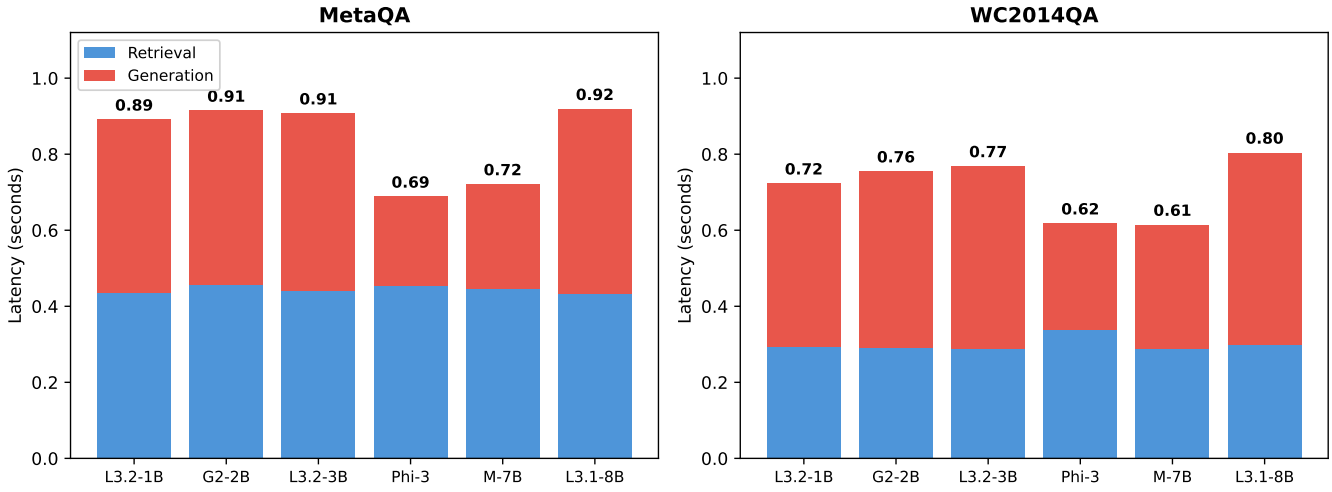

Figure 3: Latency breakdown per query. Retrieval latency is largely uniform across models under the same service configuration (0.29–0.46s); generation latency varies substantially with model architecture. Phi-3 Mini and Mistral 7B exhibit the fastest generation on both datasets, yielding the lowest end-to-end latencies.

faster than Llama-family models of comparable or smaller size on both datasets. This reflects architectural choices (grouped-query attention, sliding-window attention, efficient tokenizers) rather than raw capacity. The practical implication: *choosing the "smallest" model is not the latency-optimal strategy*; on WC2014QA Mistral 7B is Pareto-dominant over Llama 3.1 8B.

(3) RAG gains are large but differently distributed across datasets. All six models gain substantially and significantly from retrieval augmentation. In relative terms, the smallest models gain the most on MetaQA (13.2× for 1B, 4.4× for 2B, 2.4× for 8B). On WC2014QA, the pattern is less monotonic: Llama 3.2 1B (12.5×) and Mistral 7B (11.3×) show the highest ratios, reflecting both a weak parametric baseline (Mistral 7B: 8.21% without RAG on WC2014QA) and strong retrieval-grounded extraction. In absolute percentage points, the largest gains on WC2014QA belong to mid-to-large models (Mistral 7B: +84.5pp). The "smaller models benefit more" heuristic common in RAG literature thus holds in *relative* terms but is sensitive to the pairing of a model's pre-training coverage with the task domain.

(4) EM–Hits@1 gaps signal formatting, not capability, differences. On MetaQA, Llama 3.2 1B shows a 42.6pp gap between Hits@1 (72.56%) and EM (29.96%), while Llama 3.1 8B shows a

19.6pp gap (86.15% vs 66.53%). Smaller models often identify the correct entity but fail normalization (trailing tokens, synonyms, partial entity extraction). Lightweight post-processing could close much of this gap for compact models—a practical deployment lever we note for future work.

5.1 Error Analysis

Inspecting failure cases across models reveals three recurrent failure modes, each with distinct deployment implications.

(a) Format drift in compact models. Small models frequently identify the correct entity but append extraneous tokens. For the MetaQA question "what does [Grégoire Colin] appear in" (gold: `Before the Rain`), Llama 3.2 1B may return both the answer and the query entity. When the correct entity is ranked first (e.g. `Before the Rain | Grégoire Colin`), Hits@1 still credits the answer—only the top entity is tested against the reference set—whereas EM fails because the predicted set no longer equals the gold set, and F1 remains non-zero through partial overlap. When instead a spurious entity is ranked first (e.g. `Grégoire Colin | Before the Rain`), even Hits@1 fails, since the top entity is not a gold answer. This asymmetry—Hits@1 tolerates trailing extra entities but not a displaced top answer—accounts for much of the large EM–Hits@1 gap seen for the smallest models, and a lightweight post-processing step (stripping the query entity, or an LLM-based answer extractor) could recover many of these cases without retraining.

(b) Over-generation on set-valued questions. On WC2014QA questions asking for *any* member of a large set (e.g., "which soccer club is based in England", with 30 valid answers), models exhibit divergent behaviors: Mistral 7B occasionally produces generic responses, while other models enumerate subsets, sometimes hallucinating. Under our first-listed-answer EM scheme (§3), enumerating any non-canonical valid answer counts as correct under Hits@1 but as incorrect under EM, partly explaining the large EM–Hits@1 gap for models with high Hits@1 (e.g., Mistral 7B: H@1=92.7%, EM=64.1%). Prompt engineering for explicit set vs. single-answer mode could mitigate this.

(c) Retrieval misses propagate uniformly. When the top-retrieved passages fail to contain the answer, all six models degrade similarly, confirming that observed differences cannot be attributed to retrieval variance. Taken together, the three failure modes suggest that compact models can be substantially improved through lightweight post-processing and prompt refinement, without any changes to model weights or retrieval configuration.

5.2 Why Non-Monotonic Scaling?

The non-monotonic accuracy behavior we observe (e.g., L3.2-3B > Phi-3 on MetaQA; M-7B > L3.1-8B on WC2014QA) is statistically robust but warrants mechanistic interpretation. Three factors plausibly contribute. First, *architectural efficiency*: Mistral 7B combines grouped-query attention [23], in which several query heads share a single key/value projection, with sliding-window attention, which restricts each token to a fixed-size local context window. Both mechanisms reduce the memory and computation needed to attend over long inputs, so the model can process the full concatenation of retrieved passages efficiently without spending additional parameters on the attention mechanism. In our setting, where the answer must be located within a long retrieved context, this efficient long-context handling is a plausible contributor to its strong accuracy relative to similarly sized or larger models. Second, *training data composition*: Phi-3 Mini is optimized for structured reasoning via curated "textbook-quality" data, which may transfer less directly to entity-extraction tasks dominated by surface patterns. Third, *tokenizer coverage*: MetaQA contains rare movie entities and Phi-3's smaller vocabulary may fragment these into more subword units, reducing extraction fidelity. We cannot disentangle these factors with our current setup—a controlled ablation across matched model pairs would be required—but the aggregate finding remains: practitioners should not rely on parameter count as a performance proxy.

6 Deployment Guidance

Table 5 consolidates practical recommendations derived from our significance-tested results.

Table 5: Deployment guidance in compact RAG settings, grounded in our benchmarks.

Scenario	Model	Rationale
Factual/entity extraction (WC14)	Mistral 7B	Highest accuracy (92.70%); significantly beats 8B ($p < 0.001$); lowest latency.
Multi-entity QA (MetaQA)	Llama 3.2 3B	85.34% H@1, only 0.80pp below 8B; comparable latency; half the parameters.
Strict latency budget (< 0.65s)	Phi-3 Mini	Fastest end-to-end (0.618s on WC14, 0.689s on MetaQA); competitive accuracy.
Cost-/size-constrained ($\leq$ 3B)	Llama 3.2 3B	Best accuracy in the $\leq$ 3B tier on MetaQA; strong WC14 performance.

7 Limitations

Several limitations temper the generalizability of our findings.

Model family confound. Our six models span four families (Llama, Gemma, Phi, Mistral). Only Llama 3.2 1B and 3B share architecture and training pipeline, which means differences attributed to "size" conflate size with architectural choices (e.g., grouped-query attention, sliding-window attention), training data composition, and fine-tuning protocols. A strict scaling analysis would require a full within-family sweep; the present work should be read as an empirical comparison of deployable compact models under a unified RAG setup, not as a pure scaling study.

Benchmark scope. Both MetaQA and WC2014QA are single-hop, entity-centric QA benchmarks with short gold answers. We do not evaluate multi-hop reasoning, long-form generation, abstractive summarization, or tasks requiring synthesis across conflicting sources. Conclusions about accuracy ranking and optimal model selection may not transfer to such regimes; in particular, generator capacity likely matters more when retrieval provides noisier or more diverse context.

Hardware and quantization. All experiments use 4-bit NF4 quantization on a single consumer-class GPU. Absolute latency numbers are environment-dependent; we report them for reproducibility under our setup, not as universal benchmarks. Relative rankings may shift under FP16 inference, different GPU architectures, batch > 1 serving, or speculative decoding. Aggressive 4-bit quantization may also disproportionately degrade the larger models in our pool, which may attenuate the apparent performance of Llama 3.1 8B compared with a full-precision baseline.

Retrieval held fixed. We evaluate a single retrieval configuration (chunk size, top-k, reranker), so our findings establish the model ranking *under this configuration*; a sensitivity analysis over chunk size and top-k is important future work. By design this isolates the generator, but it also means we cannot characterize interactions between retrieval quality and generator capacity: a stronger retriever may benefit smaller models more by providing more self-sufficient context, while a noisier retriever could differentiate generators further, so the ranking may not be robust to alternative pipeline configurations.

Missing models and error analysis. Several relevant compact models available at submission time (Qwen 2.5 series, Gemma 2 9B) are not included, and we make no attempt to include API-only frontier models as a ceiling estimate. Our error analysis identifies failure modes (formatting mismatches, retrieval misses, overgeneration) but does not quantify per-category distributions across models.

8 Conclusion

We evaluated six compact language models (1B–8B parameters) within a controlled Retrieval-Augmented Generation pipeline across two question-answering benchmarks, reporting Hits@1, F1, and Exact Match alongside latency decomposition, with McNemar significance tests for all pairwise comparisons.

Figure 4 consolidates the main findings and visualizes the accuracy–latency trade-offs on both benchmarks: large RAG gains across all models, non-monotonic scaling in the 1B–8B range, Mistral 7B as the Pareto-dominant point on WC2014QA, and a four-model Pareto frontier on MetaQA (Phi-3 Mini, Mistral 7B, Llama 3.2 3B, Llama 3.1 8B).

Our main findings are: (i) within the heterogeneous set of compact families we evaluate, parameter count is a weak predictor of RAG accuracy in the 1B–8B range, with statistically significant non-monotonic effects (surviving Bonferroni correction) on both benchmarks; (ii) Mistral 7B lies on the accuracy–latency efficiency frontier, Pareto-dominating Llama 3.1 8B on WC2014QA with +3.42pp accuracy ($p < 0.001$) and 24% lower latency; (iii) retrieval latency is roughly generator-independent under a fixed retrieval configuration, while generation latency is determined primarily by architecture—models with efficient attention mechanisms (Phi-3 Mini, Mistral 7B) substantially reduce end-to-end latency at moderate or no accuracy cost; (iv) RAG provides large, highly significant gains across all compact models (+49.6pp to +84.5pp absolute improvements on Hits@1).

Deployment guidance should prioritize architecture and task fit over raw parameter count. For factual extraction workloads similar to WC2014QA, Mistral 7B is our top recommendation. For entity-centric QA similar to MetaQA, Llama 3.2 3B comes within 0.8 points of the 8B model on Hits@1 at substantially lower cost. When latency is strictly bounded, Phi-3 Mini provides the fastest end-to-end responses while retaining competitive accuracy. Our results suggest that with strong retrieval and careful model selection, compact models (2B–7B) achieve high absolute accuracy on entity-centric single-hop QA while remaining feasible on consumer-grade hardware.

Future work. Several directions follow directly from this study. A sensitivity analysis over the retrieval configuration (chunk size, number of retrieved passages, and reranker) would establish how robust the observed ranking is to pipeline settings, while lightweight training-free post-processing and prompt templates that distinguish single- from set-valued questions could close much of the EM–Hits@1 gap. Broader coverage—a within-family scaling sweep, additional compact models such as the Qwen 2.5 series and Gemma 2 9B, and tasks beyond single-hop entity-centric QA—would further disentangle size from architecture and clarify how far the present deployment guidance generalizes.

Acknowledgments

The authors thank the Information Systems Laboratory (ISL) at FORTH-ICS for access to the SemanticRAG infrastructure and the computational resources used in this study. This work acknowledges the support of ECHOES project, funded by the European Union under the Horizon Europe programme, Grant Agreement No. 101157364.

References

[1] P. Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.
[2] Y. Zhang, H. Dai, Z. Kozareva, A. J. Smola, and L. Song. Variational reasoning for question answering with knowledge graph. In *AAAI*, volume 32, 2018.
[3] L. Zhang, J. Winn, and R. Tomioka. Gaussian attention model and its application to knowledge base embedding and question answering. *arXiv preprint arXiv:1611.02266*, 2016.
[4] V. Karpukhin et al. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781, 2020.
[5] W. Xiong et al. Answering complex open-domain questions with multi-hop dense retrieval. In *ICLR*, 2021.
[6] A. Q. Jiang et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.
[7] Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
[8] M. Abdin et al. Phi-3 technical report. *arXiv preprint arXiv:2404.14219*, 2024.
[9] A. Dubey et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
[10] T. Kwiatkowski et al. Natural questions: a benchmark for question answering research. *TACL*, 7:453–466, 2019.
[11] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pages 1601–1611, 2017.
[12] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*, 2019.
[13] X. Jiao et al. TinyBERT: Distilling BERT for natural language understanding. In *Findings of EMNLP*, pages 4163–4174, 2020.
[14] Y. Li et al. Textbooks are all you need II: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.
[15] M. Bellagente et al. Stable LM 2 1.6B technical report. *arXiv preprint arXiv:2402.17834*, 2024.
[16] D. Hendrycks et al. Measuring massive multitask language understanding. In *ICLR*, 2021.
[17] G. Izacard and E. Grave. Leveraging passage retrieval with generative models for open domain question answering. In *EACL*, pages 874–880, 2021.

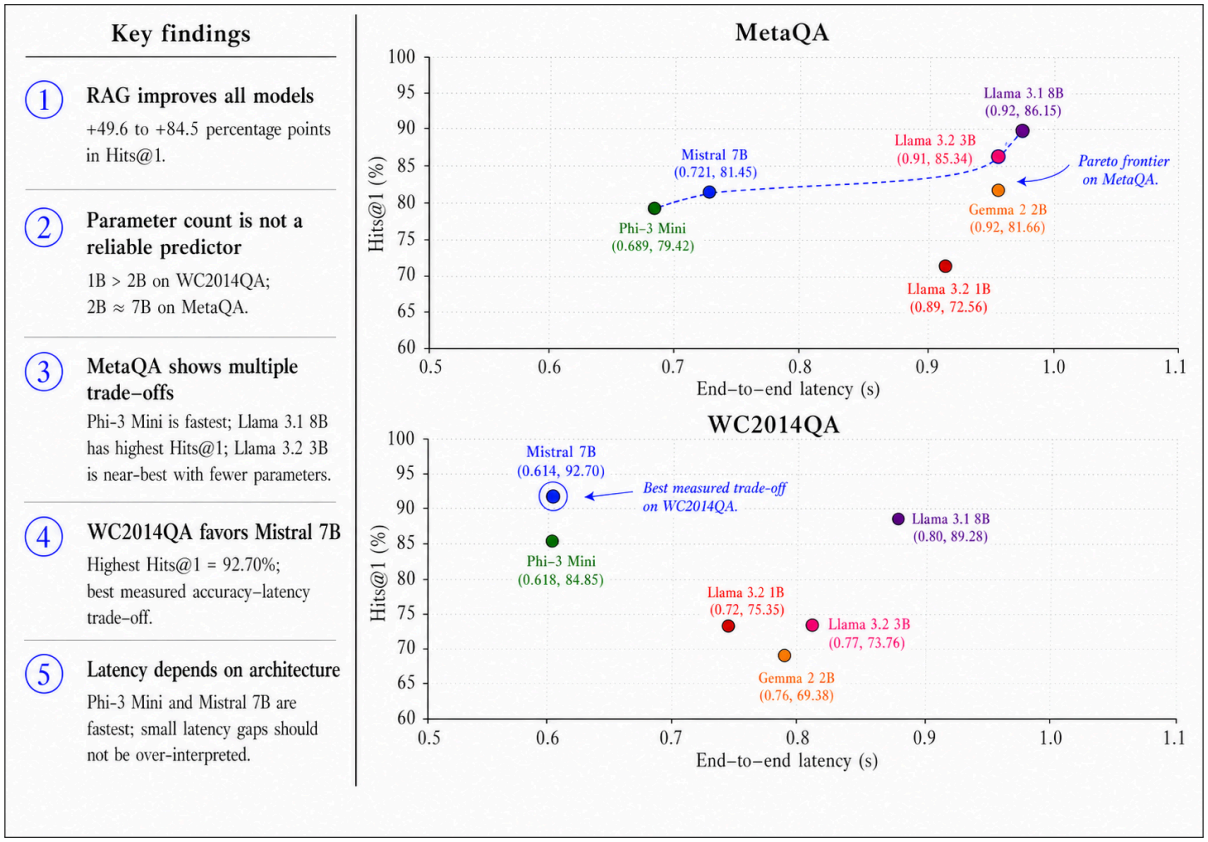

Figure 4: Summary of the main findings: large RAG gains across all models, non-monotonic scaling in the 1B–8B range, and accuracy–latency trade-offs on both benchmarks. On WC2014QA, Mistral 7B is the Pareto-dominant point; on MetaQA, the Pareto frontier spans Phi-3 Mini, Mistral 7B, Llama 3.2 3B, and Llama 3.1 8B.

[18] H. Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[19] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP*, pages 3982–3992, 2019.

[20] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[21] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.

[22] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *NeurIPS*, 2023.

[23] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *EMNLP*, pages 4895–4901, 2023.

[24] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[25] Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[26] I. Sapidis, V. Zervos, M. Mountantonakis, and Y. Tzitzikas. Interactive and provenance-aware search and QA over documents using LLMs, RAG and knowledge graph verbalization. In *New Trends in Theory and Practice of Digital Libraries (TPDL 2025)*, volume 2694 of *Communications in Computer and Information Science*, pages 248–257. Springer, 2025.