Universal Library of Engineering Technology Research Article

ISSN: 3064-996X | Volume 2, Issue 2

Open Access | PP: 49-60

DOI: https://doi.org/10.70315/uloap.ulete.2025.0202009

Joint Adaptation of Retrieval and Generation in RAG: Domain-Tuned Embeddings and LoRA-Adapted 4B Language Models

Dmitriy Ulybin

Senior ML Engineer, MTS Web Services ORCID: https://orcid.org/0009-0007-1073-3266

Abstract

Retrieval-augmented generation (RAG) is a common way to ground language models in domain documents, yet systems that must answer within a time budget depend on large generators and a reranking stage. The study aims to substantiate a design in which a domain-tuned embedding model and a compact generator of about four billion parameters, adapted with low-rank adaptation (LoRA), are tuned as one system. The method combines a structured review of 24 sources, comparative analysis, and secondary computation on benchmark tables. The results show that eight distracting passages left four generators with 43% to 97% of their gold-only accuracy, that a 2.7-billion-parameter model had a derived context-dependence indicator of 89%, and that published joint tuning raised exact match by 5 to 14 points. The article proposes a co-adaptation loop, a criterion for retiring the reranker, and a twelve-cell evaluation matrix. The conclusions state when a compact generator can replace a larger one. The material will interest engineers and researchers who build question answering systems with response-time limits.

Keywords: *Retrieval-Augmented Generation; Domain Adaptation; Text Embeddings; Dense Retrieval; Low-Rank Adaptation; Compact Language Models; Reranking; Context Noise; Time To First Token; Inference Cost.*

INTRODUCTION

Organizations have moved from trials of generative artificial intelligence to routine use, and the cost of each answer has become a design variable. The AI Index 2025 report records that 78% of organizations used artificial intelligence in 2024, compared with 55% in 2023, and that the use of generative models in business functions rose from 33% to 71% over the same year [1]. A McKinsey report published in March 2025, based on a survey conducted in July 2024, found that 78% of respondents reported AI use in at least one business function, up from 72% in early 2024 [2]. The report also found that more than 80% of respondents reported no tangible enterprise-level earnings impact from their organizations' use of generative AI [2]. Wider adoption therefore makes the cost and latency of individual answers an increasingly practical engineering concern.

Two trends shape the response to this pressure. Inference prices have fallen: the cost of querying a model at the GPT-3.5 level dropped from $20.00 to $0.07 per million tokens between November 2022 and October 2024, a reduction of more than 280 times [1]. Model size has fallen as well. In 2022 a score above 60% on the MMLU benchmark required a model with 540 billion parameters, and in 2024 the Phi-3-mini model reached that level with 3.8 billion parameters, a 142-fold reduction [1]. Demand for computation keeps growing regardless. Data centers in the United States consumed about 176 TWh in 2023, or 4.4% of national electricity use, and the Lawrence Berkeley National Laboratory projects 325 to 580 TWh by 2028, which corresponds to 6.7% to 12.0% of forecast consumption [3]. Systems that deliver the same answer quality with a smaller generator and fewer pipeline stages respond to the cost constraint and to the energy projection at once.

Retrieval-augmented generation (RAG), introduced by Lewis et al. [4], pairs a parametric generator with a retriever over an external corpus. Its quality in a specialized domain depends on two components that are usually tuned apart. Dense retrievers can lose accuracy under domain shift: in the BEIR benchmark of Thakur et al. [5], several fall behind the lexical BM25 baseline, while cross-encoder reranking improves BM25 retrieval on most tested datasets. In the benchmark latency comparison, the combined BM25-plus-reranker pipeline took approximately 450 ms per query on a graphics processor; this is not an isolated measurement of reranker overhead. Domain tuning of the embedding model with synthetic queries narrows this gap. Wang et al. [6]

Citation: Dmitriy Ulybin, "Joint Adaptation of Retrieval and Generation in RAG: Domain-Tuned Embeddings and LoRA-Adapted 4B Language Models", Universal Library of Engineering Technology, 2025; 2(2): 49-60. DOI: https://doi.org/10.70315/uloap.ulete.2025.0202009.

report gains of up to 9.3 nDCG@10 points, and Dai et al. [7] obtain an average advantage of more than 1.2 points from eight labeled examples per task. On the generation side, low-rank adaptation (LoRA), proposed by Hu et al. [8], adapts a frozen model through low-rank weight updates without added inference latency, and Biderman et al. [9] show that it maintains performance outside the target domain better than full fine-tuning does. Compact generators have become credible candidates for this role. Abdin et al. [10] describe a model with 3.8 billion parameters that reaches 69% on MMLU, and the Gemma Team [11] reports a model with 4 billion parameters that is competitive with a predecessor of 27 billion parameters.

The engineering motivation for this study comes from the author's work on MTS Navigator, a RAG-based assistant developed for a telecommunications operator. In an earlier configuration, a user waited approximately eight seconds for a usable response, and some users abandoned the interaction before it appeared. After domain-specific tuning of the embedding model improved first-stage retrieval, a separate reranker was no longer needed in the serving pipeline. A large language model was also replaced by a generator with about four billion parameters adapted using LoRA; the first words of an answer then appeared within approximately two to three seconds. This experience connects retrieval quality and generator size to a user-visible response-time constraint. It motivates the analysis below but is not treated as a controlled experiment, and no internal telemetry from the product enters the secondary calculations.

The gap lies between these two lines of work. Studies of context noise show that passages related to the query but lacking the answer reduce generator accuracy [12, 13]. Studies of joint tuning report gains for models of 7 to 65 billion parameters or for earlier sequence-to-sequence generators [14, 15, 16]. None of them examines the configuration that matters under a response-time limit: a domain-tuned embedder, a LoRA-adapted generator of about four billion parameters, and the decision to keep or remove the reranker, all assessed together on quality, latency, and cost. Whether a compact model benefits from cleaner retrieval more than a large one also remains open, because published comparisons vary model family and model size at the same time.

The aim of the study is to substantiate, on the basis of published evidence, a joint adaptation scheme for the retrieval and generation components of a RAG system with a response-time limit and to define the experimental conditions that would confirm or refute its expected benefit. Three research questions follow from the aim. The first asks how the composition of the training set affects retrieval. The second asks when reranking stops improving the result. The third asks how LoRA rank and data volume affect context-grounded answers and the retention of the general abilities of the model.

The scientific novelty consists in the systematization of published results on embedder tuning, generator adaptation, and context noise into a single conceptual model of joint adaptation, supplemented by a context dependence indicator, a criterion for retiring the reranker, and a factorial evaluation matrix for compact generators.

The author's hypothesis is that a compact generator depends on retrieved context to a greater degree than a large one and is therefore more exposed to noise in that context, so that an improvement in retrieval yields a proportionally larger gain for it. A corollary is that joint tuning of the embedder and the generator may allow the reranker to be removed and a smaller model to be used with little loss of answer quality. The article treats both statements as hypotheses to be tested and reports no new experimental confirmation of either.

The study has limitations that follow from its design. It relies on secondary data, so the figures come from experiments run on different hardware, corpora, and model families, and they are compared as orders of magnitude and directions of effect. Published measurements for models of four billion parameters inside RAG pipelines are still scarce, and the analysis draws on neighboring sizes from 2.7 to 9 billion parameters. The proposed evaluation design covers open and synthetic English-language corpora, including descriptions of films, books, and events with synthetic queries, and uses no proprietary corporate data. The MTS Navigator example is an operational motivation rather than a separately measured case study. These constraints narrow the scope of the claims and leave the logic of the proposed scheme intact.

The work has practical value for organizations in the United States, where private investment in artificial intelligence reached $109.1 billion in 2024 and 40 notable models originated from domestic institutions [1]. Activity on this scale raises the question of how answer quality can be delivered at lower computational cost, and the projected growth of data center electricity demand turns that question into a matter of infrastructure planning [3]. The author's field, the tuning of embedding models and the LoRA adaptation of compact generators for systems with a response-time limit, addresses the question directly. A pipeline with a smaller generator and without a reranking stage needs fewer accelerator hours per answer and can run on hardware within reach of mid-sized firms, hospitals, and public agencies. Competence in building such pipelines on open data therefore supports a broader and more economical use of language technology across the national economy. The next section describes the sources and procedures used to examine these propositions.

MATERIALS AND METHODS

The gap stated above concerns the consistency of evidence across experiments that no single laboratory has run together, so the study uses a document-based design. Four methods were applied. A structured literature review identified peer-reviewed and institutional publications on RAG,

dense retrieval, parameter-efficient tuning, and inference efficiency. Comparative analysis placed the reported metrics of separate and joint adaptation side by side. Content analysis of technical reports extracted model sizes, benchmark scores, and latency measurements. Secondary computation derived two indicators, accuracy retention under noise and context dependence, from accuracy tables published in the reviewed sources.

Sources were searched in the ACL Anthology, the ACM Digital Library, the proceedings of NeurIPS, ICLR, and AAAI, Transactions on Machine Learning Research, and arXiv. The search combined the terms retrieval-augmented generation, dense retrieval, domain adaptation, synthetic queries, reranking, low-rank adaptation, small language model, irrelevant context, and time to first token. Inclusion required availability between January 2019 and April 2025, inclusive, a documented experimental protocol or survey methodology, and numerical results that could be checked against the source text. Blogs, press releases, and news items were excluded. Each figure used in the article was traced to a table, an abstract, or an executive summary of the cited source before it was included.

The source base comprises 24 publications. Sixteen are peer-reviewed papers from conferences and journals in natural language processing, information retrieval, and machine learning. Five are technical reports and preprints issued by university and corporate research groups. Two are institutional research reports, one from the Stanford Institute for Human-Centered AI and one from the Lawrence Berkeley National Laboratory. One is a consulting survey, accounting for approximately 4% of the source base and used only for adoption statistics.

Each part of the analysis rests on a distinct group of sources. The theoretical part draws on the RAG formulation [4], on LoRA [8], and on studies of retriever training [5, 6, 7, 17, 18]. The market part uses the adoption, cost, and energy statistics [1, 2, 3]. The practical part uses published case measurements: reranking studies [5, 19], noise experiments [12, 13, 20, 21], studies of joint and generator-side tuning [14, 15, 16, 22], model reports [10, 11], a characterization of latency by Shen et al. [23], and the evaluation framework of Es et al. [24].

Two indicators were computed for the secondary analysis. Accuracy retention under noise is the ratio of accuracy with a given number of distracting passages to accuracy with the gold passage alone. Context dependence is the share of gold-passage accuracy that is absent in the closed-book condition, calculated as one minus the ratio of closed-book accuracy to gold-only accuracy. Both indicators are dimensionless, which allows models with different absolute accuracy to be compared. They are derived values and are marked as such wherever they appear.

The evaluation design proposed in the next section reflects the experimental basis planned by the author. It compares a base and a domain-tuned embedder, a large and a compact generator, configurations with and without reranking, and the compact generator before and after LoRA adaptation. Outcome measures cover retrieval quality, answer quality, retention of general abilities, total response time, time to first token, and inference cost. The design is presented as a protocol, and no measurements from it are reported here. The following section applies these methods to the three research questions in turn.

RESULTS AND DISCUSSION

The proposed scheme is reasonable only if its levers act on the same order of magnitude as the costs they remove, so the review first establishes how large the reported effects are. Table 1 gathers the measurements in one place. They come from seven studies with different metrics, and a tabular form lets the reader compare the baseline and the adapted value row by row.

Table 1. Published effects and costs of retrieval-side, generation-side, and joint adaptation (compiled by the author based on [5, 6, 7, 14, 15, 16, 23])

Lever	Study and setting	Metric	Baseline	After adaptation
Embedder tuning with synthetic queries and cross-encoder pseudo labels	Six domain datasets	nDCG@10 relative to a zero-shot dense retriever	reference	up to +9.3 points
Embedder tuning from few-shot synthetic queries	11 retrieval sets, eight examples per task	average nDCG@10 relative to ColBERT v2 and SPLADE v2	reference	more than +1.2 points
BM25 retrieval with cross-encoder reranking of 100 candidates	18 datasets	datasets where reranked BM25 exceeds BM25	not applicable	16 of 18
BM25 retrieval with cross-encoder reranking of 100 candidates	One million documents, graphics processor	retrieval latency per query	14 to 20 ms (dense retrievers)	450 ms (BM25 with reranker)
Generator tuning with distractor passages	LLaMA2-7B, HotpotQA	accuracy, %	0.03 (base model with retrieval)	35.28
Generator tuning with distractor passages	LLaMA2-7B, PubMed QA	accuracy, %	58.8 (base model with retrieval)	73.30

Dual tuning of generator and retriever	LLaMA 65B, zero-shot	average score over knowledge-intensive tasks, %	45.1 (retrieval without tuning)	49.1
Dual tuning of generator and retriever	LLaMA 65B, five-shot	average score over knowledge-intensive tasks, %	51.1 (retrieval without tuning)	51.8
End-to-end joint training with a reconstruction signal	COVID-19 domain	exact match	2.95 (original RAG, fine-tuned)	8.32
End-to-end joint training with a reconstruction signal	News domain	exact match	7.26 (original RAG, fine-tuned)	14.08
End-to-end joint training with a reconstruction signal	Conversation domain	exact match	12.09 (original RAG, fine-tuned)	25.95
Retrieval placed on the response path	Gemma 2 with 9 billion parameters	time to first token	495 ms (no retrieval)	965 ms (with retrieval)

Three observations follow from the table. Retrieval-side tuning with synthetic data moves ranking quality by single-digit nDCG points, which is the order of benefit usually sought from a reranker, while BEIR reports approximately 450 ms for the BM25-plus-cross-encoder pipeline compared with 14–20 ms for selected dense retrievers, a difference of more than twentyfold between entire retrieval configurations [5]. The latency values date from 2021 hardware and are used here as a ratio. Generator-side tuning changes behavior sharply when the base model handles retrieved context poorly: the untuned LLaMA2-7B with retrieval scored 0.03% on HotpotQA and 35.28% after training with distractor passages [16]. Joint tuning can improve on individual-stage baselines in the tested settings, although the size of the gain depends on the model and evaluation setting. Lin et al. [14] record 4.0 points in the zero-shot setting and 0.7 points in the five-shot setting for a model of 65 billion parameters, and they note that retrieval-augmented instruction tuning gave a larger boost to the versions with 7 and 13 billion parameters.

Soudani et al. [22] reach a compatible conclusion from another direction. In their comparison of twelve language models, a small fine-tuned model with retrieval (StableLM2, 1.6 billion parameters) performed on par with or better than a larger one (Llama3, 8 billion parameters), and parameter-efficient tuning outperformed full fine-tuning when the tuned model was combined with retrieval, because it preserved the reasoning ability needed to use the passages. The end-to-end results of Siriwardhana et al. [15] show the same direction for an earlier generator: exact match rose by 5.37, 6.82, and 13.86 points across three domains once the retriever and the generator were trained together. The evidence is therefore consistent on direction and silent on the specific combination studied here, which is why the remaining subsections construct that combination step by step.

The first research question concerns the composition of the embedder training set. The reviewed studies vary four properties of that set. The first is the origin of queries. Dai et al. [7] generate up to eight synthetic queries for each of up to one million sampled documents from a prompt with no more than eight examples, and Wang et al. [17] train an embedding model on synthetic data alone in fewer than 1,000 steps and obtain further gains from a mixture of synthetic and labeled data. The second property is filtering. In the round-trip consistency filter of Dai et al. [7] a generated pair is kept only if the source passage is returned in the first position by an initial retriever. The third is the treatment of negatives. Qu et al. [18] combine cross-batch negatives with hard negatives that a cross-encoder has cleared of unlabeled positives. The fourth is the labeling signal: Wang et al. [6] replace binary labels with scores from a cross-encoder and report that the method needs less unlabeled target data than earlier approaches.

These studies judge composition by ranking metrics. For a RAG system the author proposes to judge it also by the noise profile that the tuned retriever hands to the generator. Cuconasu et al. [12] distinguish answer-bearing gold passages, semantically related but non-answer-bearing distracting passages, and unrelated random passages. Their central counterintuitive finding is that, although top-ranked distracting passages can reduce answer accuracy, supplementing a prompt with random passages can increase it by up to 35% relative to the comparison condition. In one reported setting with four retrieved documents, the absolute increase was 0.07; the 35% figure is a relative gain, not 35 percentage points [12]. This does not establish that random passages are universally beneficial: the effect depends on the experimental context and passage mix. A noise profile can therefore be defined for a context of k passages as three shares that sum to one: answer-bearing, distracting, and random. Two embedders with equal Recall@k can differ in the distracting share and in the mixture of other passages, leading to different answer outcomes.

The observation of Jin et al. [21] gives this proposal practical weight. With a stronger retriever, whose Recall@40 on Natural Questions was 0.90 against 0.73 for BM25, answer accuracy first rose and then fell as more passages were supplied, and the decline was more pronounced than with the weaker retriever. Domain tuning thus changes the kind of noise that reaches the generator. Random passages may become rarer while semantically related, non-answer-bearing passages remain prominent among retrieval errors; their effects on

generation should be measured separately. The training set of the embedder should therefore be assessed on two counts: the gain in recall and the distracting share that remains in the context. This second count connects the retrieval question to the behavior of the generator, examined next.

The central hypothesis concerns how generators of different sizes react to noise. The experiment of Cuconasu et al. [12] offers a direct view, because it holds the gold passage fixed and adds distracting passages in steps. Figure 1 plots the published accuracy values as curves, since the shape of the decline is easier to read from lines than from numbers in a table. The reader should compare the slope of each curve over the first two distractors and the level at which the curve flattens.

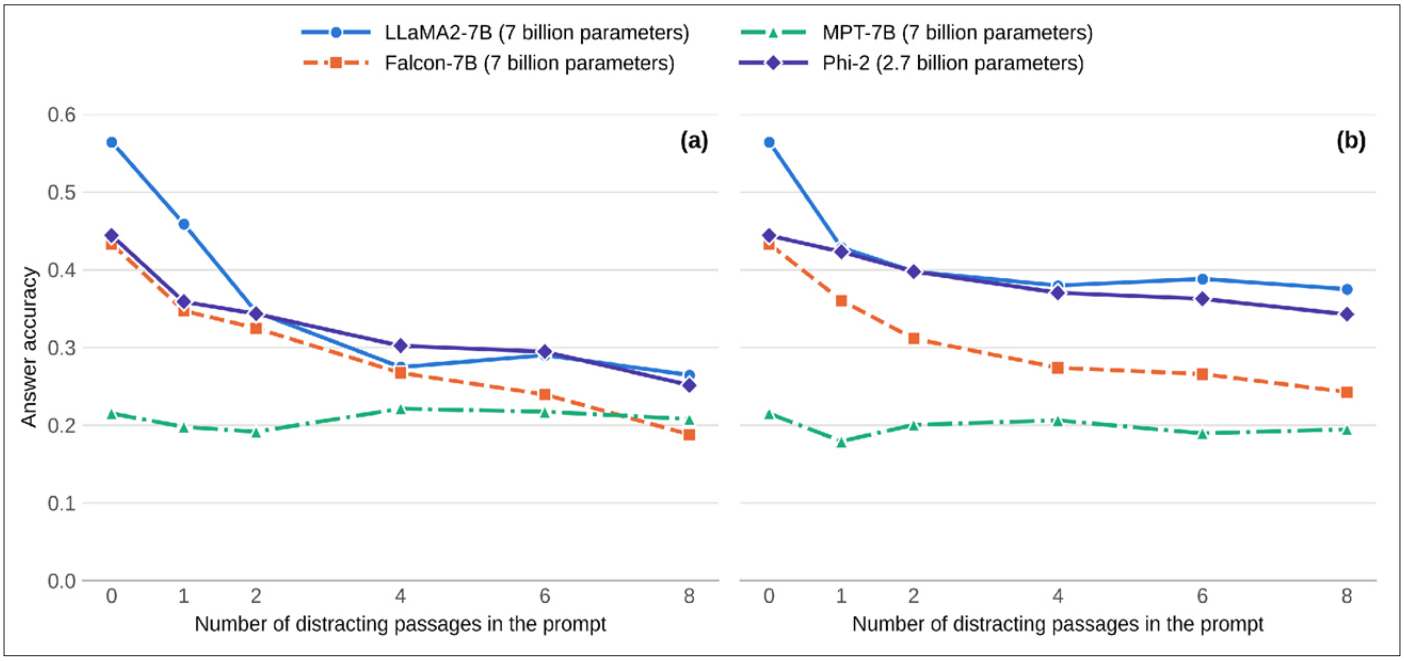


Figure 1. Accuracy of four open generators on NQ-open as distracting passages are added to the gold passage, with the gold passage placed far from the query (a) and next to the query (b) (compiled by the author based on [12])

Every model that uses context well loses accuracy as soon as one distractor enters the prompt, and the size of the loss depends on where the gold passage sits. With the gold passage far from the query, LLaMA2-7B falls from 0.5642 to 0.2643 at eight distractors, Falcon-7B from 0.4330 to 0.1878, and Phi-2 from 0.4438 to 0.2513. With the gold passage next to the query, the same models keep 0.3748, 0.2424, and 0.3423 [12]. MPT-7B behaves differently because it gains little from the gold passage to begin with: its accuracy is 0.2148 with the passage and 0.1205 without any documents.

Absolute accuracy values alone do not fully capture the relationships required to evaluate the hypothesis. Table 2 therefore presents the two derived indicators defined in the methods section context dependence and accuracy retention using the experimental results of Cuconasu et al. [12] (Panel A) and Chen et al. [13] (Panel B). The latter study examines how accuracy changes as the proportion of noisy passages among five supplied documents increases to 0.8. To extend the analysis beyond observed performance degradation, Panel C introduces complementary approaches to retrieval optimization and model adaptation: Promptagator [7] and LoRA [8]. These methods provide additional context for discussing potential improvements to RAG systems, although their effects on noise robustness were not directly evaluated in the experiments presented. The tabular format allows the derived performance indicators to be compared across models while distinguishing empirical findings from complementary optimization methods.

Table 2. Context dependence, noise robustness, and complementary optimization methods for retrieval-augmented generation (compiled by the author based on [7, 8, 12, 13]; derived accuracy metrics calculated from [12, 13])

Panel A. NQ-open, eight distracting passages added to the gold passage [12]

Model	Parameters, billion	Closed-book accuracy	Gold-only accuracy	Context dependence, %	Retention, gold far / gold near, %
LLaMA2-7B	7	0.1123	0.5642	80.1	46.8 / 66.4
Falcon-7B	7	0.1083	0.4330	75.0	43.4 / 56.0
MPT-7B	7	0.1205	0.2148	43.9	96.7 / 90.5
Phi-2	2.7	0.0488	0.4438	89.0	56.6 / 77.1

Panel B. RGB benchmark, English, five external documents [13]

Model	Parameters, billion	Accuracy at noise ratio 0, %	Accuracy at noise ratio 0.8, %	Retention at noise ratio 0.8, %
ChatGPT (gpt-3.5-turbo)	Not disclosed	96.33	76.00	78.9
ChatGLM-6B	6	93.67	70.67	75.4
ChatGLM2-6B	6	91.33	57.33	62.8
Vicuna-7B-v1.3	7	87.67	60.33	68.8
Qwen-7B-Chat	7	94.33	73.67	78.1
BELLE-7B-2M	7	83.33	64.67	77.6

Panel C. Complementary retrieval optimization and model adaptation methods [7, 8]

Method	Component	Experimental setting	Reported result	Relevance to RAG
Promptagator	Dense retriever	Task-specific retrieval trained using synthetic queries generated from no more than eight examples	More than 1.2 nDCG@10 points improvement over ColBERTv2 on average across 11 retrieval datasets	Improvement of task-specific retrieval quality
LoRA	Language model adaptation	Parameter-efficient fine-tuning of pretrained language models, including GPT-3 175B	Up to 10,000× fewer trainable parameters and approximately 3× lower GPU memory requirements than full fine-tuning	Efficient adaptation of language models to downstream tasks

Notes. Context dependence = (gold-only accuracy − closed-book accuracy) / gold-only accuracy × 100%. Retention in Panel A = accuracy with eight distracting passages / gold-only accuracy × 100%, separately for the far and near positions of the gold passage. Retention in Panel B = accuracy at noise ratio 0.8 / accuracy at noise ratio 0 × 100%. Panels A and B represent different benchmarks and experimental settings; their absolute accuracy values are not directly comparable. Panel C presents independently evaluated optimization methods rather than direct measurements of noise robustness.

The derived values separate two properties that the hypothesis had merged. Within these four tested models, the smallest generator has the highest derived context-dependence indicator; model size, architecture, and training are not controlled independently. For Phi-2, with 2.7 billion parameters, the derived context-dependence indicator is 89.0%, against 80.1% for LLaMA2-7B and 75.0% for Falcon-7B. Phi-2 scores 0.0488 in the closed-book condition on this task; its much higher gold-only accuracy indicates that retrieved evidence is important to its performance in this setting. Noise sensitivity, in contrast, does not follow size in these data. Phi-2 retains 56.6% and 77.1% of its gold-only accuracy at eight distractors, more than either model of 7 billion parameters that uses context well. In the RGB data, retention at a noise ratio of 0.8 ranges from 62.8% to 78.9%, and two models of one family and one size, ChatGLM-6B and ChatGLM2-6B, differ by 12.6 percentage points [13].

The published evidence therefore supports a refined version of the hypothesis. The comparison shows that the smallest tested generator has the highest context-dependence indicator, making the quality of retrieved evidence especially relevant in this setting; it does not establish a general size-dependent law. Sensitivity to distractors also appears to depend on factors other than parameter count, including training, which is a potential lever for LoRA adaptation. Yoran et al. [20] show that a mixture of relevant and irrelevant contexts during tuning makes a model resistant to irrelevant passages while it keeps its performance on relevant ones, and that 1,000 examples suffice. The claim that a generator of four billion parameters gains more from cleaner retrieval than a larger one remains untested, since no reviewed study varies size within one model family under controlled noise. The refined hypothesis shapes the conceptual model that follows.

The findings above argue for treating the embedder and the generator as one system with a shared training signal. The author's conceptual model has five stages. In the first stage, synthetic queries are generated for corpus passages and filtered for consistency, following the practice documented by Dai et al. [7]. The corpora planned by the author are open collections: descriptions of films, books, and events, together with an open document corpus. In the second stage, the embedder is tuned with a contrastive objective on these pairs, with hard negatives that a cross-encoder has screened [18] and with cross-encoder scores as soft labels [6]. The cross-encoder works here as a teacher and a filter. It is present during training and may be absent during serving, so part of what a reranker would contribute at query time is moved into the embedder in advance.

Figure 2 presents the whole model as a flow diagram. The scheme is a sequence of stages with one feedback path, and the order of the stages is the substance of the proposal, which a diagram conveys more directly than prose. The reader should follow the arrow that leads from the tuned embedder to the training contexts of the generator.

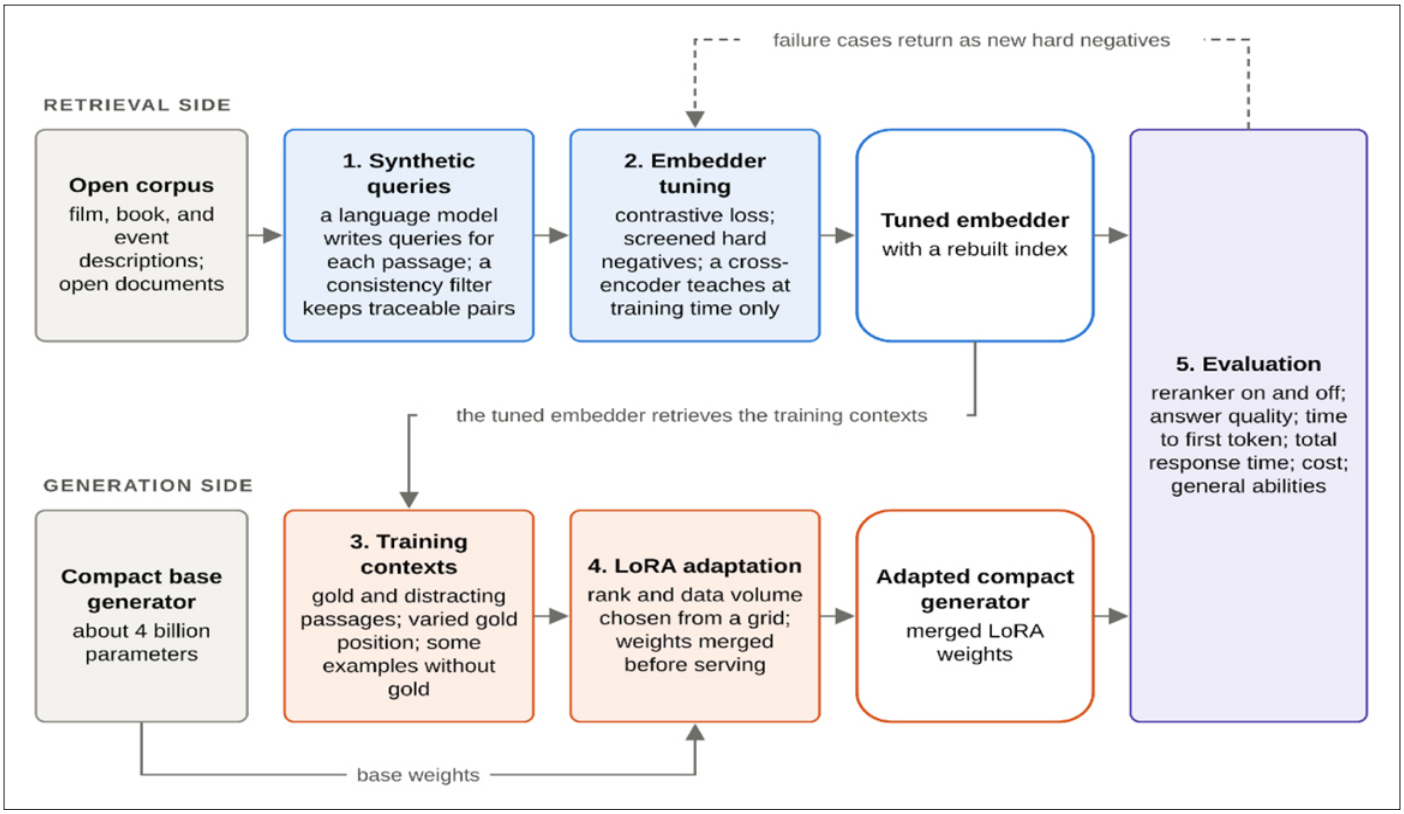


Figure 2. Conceptual model of joint adaptation of a domain embedder and a compact generator (compiled by the author based on [6, 7, 8, 16, 18, 20])

The third stage is the step that makes the adaptation joint. The tuned embedder, and not the base one, retrieves the context for every training question of the generator. Each retrieved passage is marked as gold or distracting, the gold passage is placed at varied positions, and a share of examples is built without any gold passage. Zhang et al. [16] trained with one gold and four distractor passages and found that the preferable share of examples containing the gold passage varied by dataset from 40% to 100%. The generator thus learns against the noise profile that it will meet in production. In the fourth stage, the compact generator is adapted with LoRA on these contexts, and the low-rank matrices are merged into the base weights before serving, which by construction adds no inference latency [8].

The fifth stage evaluates the system and closes the loop. Questions on which the adapted generator still fails are inspected, and the distracting passages that caused the failure are returned to the second stage as new hard negatives. The scheme differs from the dual instruction tuning of Lin et al. [14] and from the end-to-end training of Siriwardhana et al. [15] in where the coupling sits. Those methods pass a training signal from the generator to the retriever inside one optimization process, which in the end-to-end case requires the corpus to be encoded and indexed again during training. The proposed loop alternates two ordinary training jobs and rebuilds the index between rounds. It asks for no special infrastructure and suits a team that tunes a compact model on a single accelerator. Whether the alternating form reaches the quality of the integrated forms is an empirical question for the evaluation matrix presented below.

The second research question asks when reranking stops improving the result. Two published findings frame the answer. In the BEIR benchmark the cross-encoder improved on BM25 in 16 of 18 datasets, which shows its value over a weak first stage [5]. Jacob et al. [19] tested rerankers after strong dense retrieval and found that the gains diminish as more documents are scored and that quality can degrade beyond a certain number of candidates. The value of reranking is therefore conditional on the first stage, and joint adaptation changes the first stage. The author proposes to decide the question with three measurable conditions.

The first condition is rerank headroom, defined by the author as the difference between Recall@K for the candidate pool and Recall@k for the context that reaches the generator. A reranker can only promote relevant passages that lie between positions k and K, so headroom is an upper bound on what it can add. When the cross-encoder has already served as a teacher for the embedder [6], headroom is expected to shrink, and it can be measured on held-out queries before any generator is run.

The conditions are checked in sequence, and each of them can end the procedure, so Figure 3 presents the criterion as a decision flow. The diagram shows the three checks from left to right and the two outcomes below them.

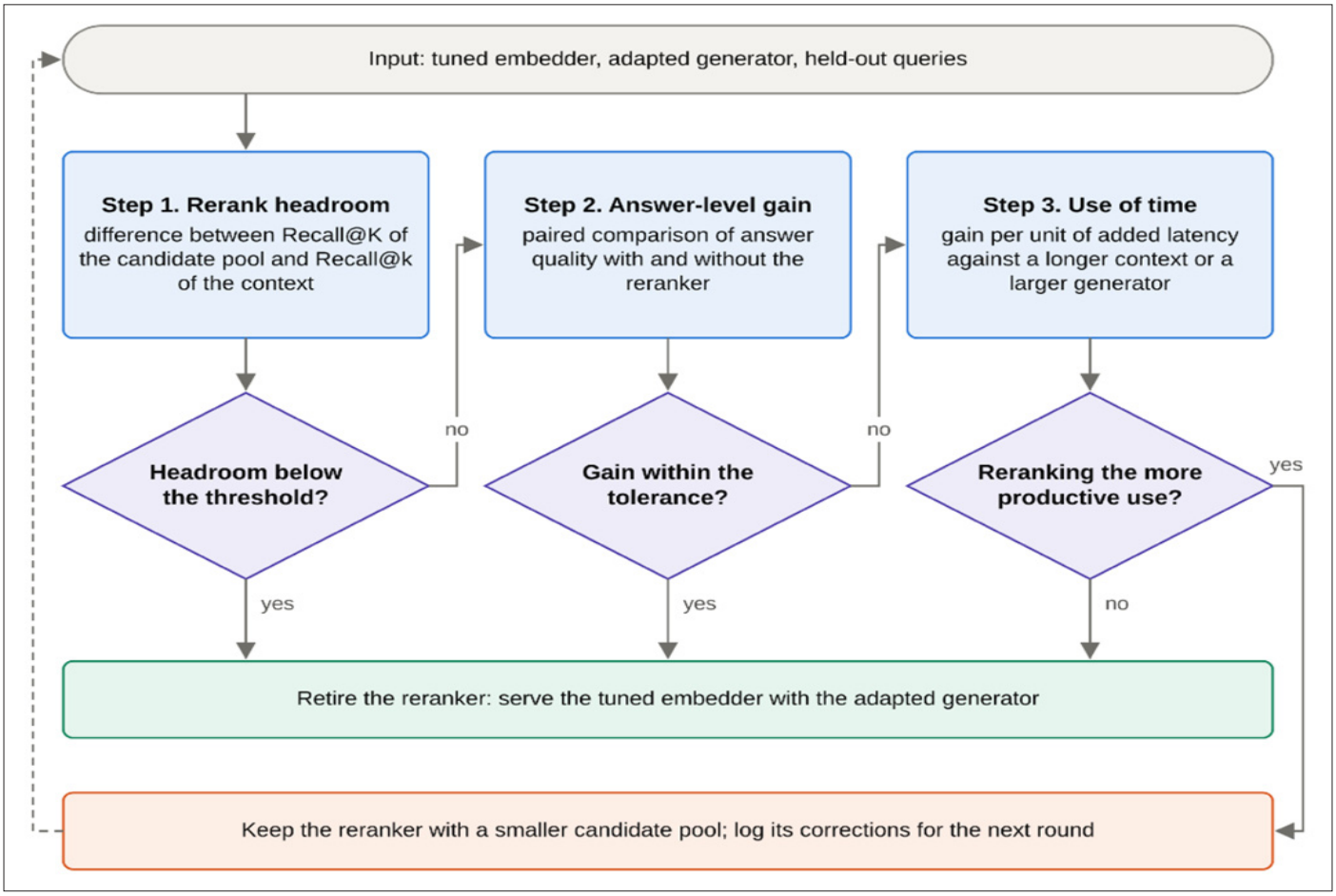


Figure 3. Decision flow for retiring the reranking stage after joint adaptation (compiled by the author based on [5, 6, 12, 19])

The second condition is the answer-level gain. Even with headroom available, reordering matters only if the generator is sensitive to order. For the unadapted LLaMA2-7B at eight distractors, placing the gold passage next to the query gives 0.3748 and placing it far away gives 0.2643 [12]. The difference of 0.1105 is the value of ordering for a generator that has not been trained for noisy context. A generator adapted on contexts with varied gold positions should narrow this difference, and the reranker then loses the second source of its benefit. The check is a paired comparison of answer quality with and without the reranker on the adapted generator, with a tolerance set in advance.

The third condition is the use of time. If a gain survives the first two checks, it is expressed as quality points per unit of added latency and compared with other uses of the same interval, such as a longer context or a larger generator. The reranker stays only if it is the more productive use. Reranking thus stops paying off when two things hold together: the embedder already places the gold passage inside the context, and the generator no longer depends on where inside the context it sits. Neither condition alone is sufficient, which is the argument for adapting both components before the decision is taken. When the reranker is kept, its corrections can be logged and added to the next round of embedder tuning, so each round moves the system toward the point where the stage can be dropped.

The third research question concerns the generator. The reviewed studies bound the choices from both sides. Hu et al. [8] varied the rank from 1 to 64 on the attention projections of GPT-3 and reported that a rank of one or two was sufficient for their tasks. Biderman et al. [9] trained LLaMA-2-7B with ranks of 16, 64, and 256 on all attention and feed-forward modules. They found that LoRA underperformed full fine-tuning in the target domains at standard low ranks, that it better maintained the base model's scores on HellaSwag, WinoGrande, and ARC-Challenge, and that full fine-tuning learned weight perturbations with a rank 10 to 100 times greater than typical LoRA configurations. Their learning and forgetting curves moved together: models trained on more data learned more and forgot more, and higher ranks shifted the curves in the same direction.

The volume of data needed for context-grounded answering appears modest in the published record. Yoran et al. [20] obtained resistance to irrelevant context from 1,000 examples, and Jin et al. [21] used 50,000 mixed samples for tuning aimed at long retrieved inputs. The author's interpretation is that answering from context is a behavioral skill: the model learns where to look, how to weigh competing passages, and when to decline. Domain facts arrive through retrieval and need not be stored in the weights. If this reading is correct, the task sits near the low-rank end of the range studied by Biderman et al. [9], and raising the rank or the volume

beyond the point of saturation would add forgetting without adding grounded accuracy.

The author therefore proposes to tune rank and volume as a two-axis grid with a guard metric. The grid covers ranks of 8, 16, 64, and 256 and training sets of 1,000, 10,000, and 50,000 examples, which spans the values documented in the reviewed studies. The target metric is answer quality on contexts with distractors. The guard metric is a retention ratio: the score of the adapted model on a fixed set of general benchmarks divided by the score of the base model, with the benchmark set of Biderman et al. [9] as a starting point. The selected cell is the one with the smaller rank and volume among those where the target metric has stopped improving beyond the tolerance and the retention ratio stays above the threshold set by the system owner. Retention matters in practice because a compact model in production often serves several tasks, and an adapter that harms the others cannot be merged into shared weights. The grid is a design for measurement, and its expected outcome is stated here as an assumption.

The time limit determines which of these gains can be realized, so the stages must be placed on a common time axis. Time to first token is the sum of query embedding, vector search, reranking, and the processing of the prompt by the generator. Total response time adds the decoding of the answer. Shen et al. [23] measured that retrieval made up 41% of end-to-end latency and 45% to 47% of time to first token in their setup, and that time to first token nearly doubled, from 495 ms to 965 ms, once retrieval was placed in front of a generator with 9 billion parameters. Stages that precede the first token are the costly ones for an interactive system.

Figure 4 shows the response path as a schematic timeline without numerical values, because stage durations depend on hardware. It compares a reference pipeline with the target pipeline and marks the stage that each adaptation affects.

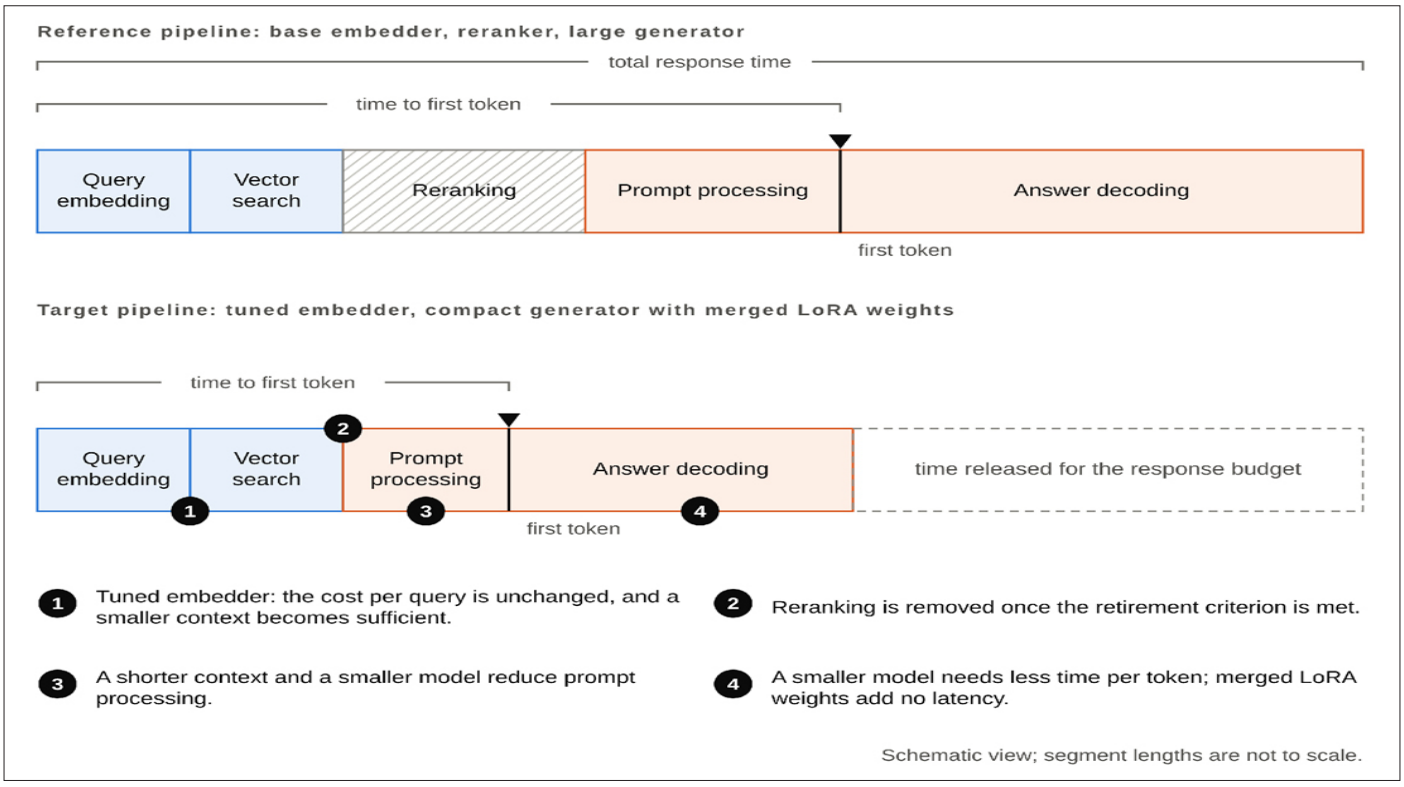


Figure 4. Stages of the response path and the points at which joint adaptation changes time to first token and total response time (compiled by the author based on [5, 8, 23])

The proposed changes affect both the interval before the first token and the subsequent answer-decoding stage. An unchanged embedder architecture adds no new serving stage, while improved retrieval could permit a shorter context; actual runtime still requires measurement. Removing the reranker eliminates one serving stage; the 450 ms measured in BEIR was the latency of BM25 retrieval with reranking, not the incremental cost of the reranker alone [5]. Merging LoRA updates into the base weights avoids a separate adapter computation at inference [8]. A compact generator can reduce prompt-processing and token-decoding costs, although realized latency depends on model architecture, hardware, and serving implementation. Quality risk, by contrast, is concentrated in two decisions, the smaller generator and the missing reranker, and these are the two decisions that joint adaptation is designed to offset.

Table 3 turns the scheme into a factorial evaluation matrix. A table is the suitable form because each row is a contrast between configurations that differ in one factor, which is what allows an effect to be attributed to a cause.

Table 3. Factorial evaluation matrix for joint adaptation under a response-time limit (compiled by the author based on [9, 12, 14, 16, 23, 24])

Contrast	Configurations compared	Question answered	Primary measures
1. Retrieval tuning	E0 against E1; generator fixed; no reranker	How the composition of the training set changes retrieval (first research question)	Recall@k, nDCG@10, distracting share in the context
2. Reranker before tuning	E0 with R1 against E0 with R0	Reference value of reranking over an untuned first stage	nDCG@10, answer accuracy, time to first token
3. Reranker after tuning	E1 with R1 against E1 with R0, for each generator	When reranking stops improving the result (second research question)	rerank headroom, answer-level gain, added latency
4. Generator size	GL against GC; base embedder E0	Quality difference attributable to size alone	answer accuracy, faithfulness, total response time, cost per 1,000 answers
5. Differential gain	Gain from E0 to E1 for GC against the same gain for GL	Whether the compact generator gains more from cleaner retrieval (central hypothesis)	difference in differences of answer accuracy, context dependence
6. LoRA adaptation	GC against GC with LoRA; tuned embedder E1	Effect of rank and data volume on context-grounded answers (third research question)	accuracy with distractors, retention under noise, faithfulness
7. General abilities	GC against GC with LoRA; no retrieval	Whether adaptation preserves abilities outside the domain	retention ratio on general benchmarks
8. Target against reference	E1, GC with LoRA, R0 against E0, GL, R1	Whether joint adaptation permits a smaller model without a reranker (corollary)	answer accuracy, faithfulness, time to first token, total response time, cost

Note. E0 is the base embedder and E1 the domain-tuned embedder; GL is the large reference generator and GC the compact generator of about four billion parameters; R1 and R0 denote configurations with and without the reranker.

The matrix contains twelve cells: two embedders, three generator variants, and two reranker settings. The fifth contrast is the direct test of the central hypothesis. It is a difference in differences, so it removes the constant advantage of the larger model and isolates the interaction between generator size and retrieval quality. A positive value would mean that the compact generator gains more. The eighth contrast tests the corollary by setting the target configuration against the conventional one. Answer quality is measured with reference answers where they exist and with the reference-free measures of Es et al. [24], namely faithfulness, answer relevance, and context relevance, where they do not. The latency measures follow the decomposition used by Shen et al. [23], and all configurations are to be timed on one hardware setup.

The author proposes a simple rule for selecting the operating point from the measured cells. Configurations that exceed the limit on time to first token or fall below the retention threshold are excluded first. Among the remaining ones, the set that is not dominated on answer quality and cost is identified. The selected configuration is the one with the lower cost among those whose quality lies within the agreed tolerance of the reference configuration. The rule makes the trade explicit: a stated amount of quality is exchanged for a measured reduction in latency and cost, and the tolerance is fixed before the measurements are read.

Several recommendations for teams that build RAG systems with a response-time limit follow from the analysis. The first is to measure the context dependence of the candidate generator on a domain sample before any tuning begins. A high value, such as the 89.0% computed for Phi-2, indicates that retrieval sets the ceiling and that embedder tuning deserves the first share of effort. The second is to build the embedder training set from synthetic queries with a consistency filter and screened hard negatives, and to keep a set of human-written queries apart for evaluation, so that the measured gain does not reflect the style of the query generator.

The third recommendation is to train the generator on contexts retrieved by the tuned embedder, with varied gold positions and a share of examples without a gold passage. The fourth is to keep the cross-encoder in the training pipeline as a teacher and a filter and to decide on its serving role with the retirement criterion instead of by convention. The fifth is to report time to first token, total response time, and cost per thousand answers next to every quality figure and to obtain them on one hardware setup. The sixth is to track the retention ratio at each increase of rank or data volume.

The scheme fits corpora whose content is stable and whose queries are of recurring types, such as catalogs of films, books, and events, where synthetic queries resemble real ones. Where the corpus changes quickly, embedder tuning can be repeated on a schedule while the LoRA adapter stays in place, because the adapter encodes a way of reading context and holds no corpus facts. The same separation lets one compact base model serve several domains with one embedder and one adapter per domain. These recommendations lead to the conclusions below.

CONCLUSION

The study set out to substantiate a joint adaptation scheme for the retrieval and generation components of a RAG system

with a response-time limit and to define the conditions for testing it. The aim has been met at the level that published evidence allows. The review showed that each lever has a documented effect: synthetic-query tuning of the embedder improved nDCG@10 by up to 9.3 points, generator tuning with distractor passages raised accuracy on HotpotQA from 0.03% to 35.28%, and joint training raised exact match by 5 to 14 points in three domains. In a separate BEIR benchmark comparison, the BM25-plus-cross-encoder retrieval pipeline took more than twenty times as long as selected dense retrievers; this ratio does not measure the incremental cost of reranking alone.

The three research questions received the following answers. On the composition of the training set, the literature supports synthetic queries with consistency filtering and screened hard negatives, and the article adds the noise profile of the retrieved context as a second criterion next to recall, reflecting Cuconasu et al.'s finding that random passages can improve accuracy by up to 35% relative while related distractors can reduce it [12]. On reranking, the stage stops paying off when rerank headroom is small and the adapted generator no longer depends on passage order; a three-step decision flow makes both conditions measurable. On LoRA, the published learning and forgetting curves and the small data volumes reported for noise resistance suggest that context-grounded answering sits at the low-rank end of the studied range, and a grid with a retention ratio as a guard metric is proposed to locate the saturation point.

The central hypothesis is supported in a refined form. Secondary computation yielded a context-dependence indicator of 89.0% for a 2.7-billion-parameter generator, compared with 75.0% and 80.1% for two 7-billion-parameter models. The comparison indicates greater measured context dependence for the smallest tested generator, but differences in model family and training prevent attributing that result to model size alone. Sensitivity to distracting passages did not follow size: retention at eight distractors ranged from 43% to 97% across four models. Whether a generator of four billion parameters gains more from cleaner retrieval than a larger one remains to be measured, and the fifth contrast of the evaluation matrix is designed for that measurement.

The practical significance of the work lies in three instruments that an engineering team can apply without new infrastructure: the context dependence indicator for choosing where to invest tuning effort, the criterion for retiring the reranker, and the twelve-cell matrix with its rule for selecting the operating point. Together they convert the choice of a smaller generator from a matter of intuition into a sequence of measurements on quality, time to first token, total response time, and cost. The next step planned by the author is to run the matrix on open corpora of film, book, and event descriptions with synthetic queries and to publish the measured cells.

REFERENCES

1. Maslej, N., Fattorini, L., Perrault, R., Gil, Y., Parli, V., Kariuki, N., Capstick, E., Reuel, A., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Niebles, J. C., Shoham, Y., Wald, R., Walsh, T., Hamrah, A., Santarlasci, L., . . . Oak, S. (2025). *The AI Index 2025 annual report*. AI Index Steering Committee, Institute for Human-Centered AI, Stanford University. https://doi.org/10.48550/arXiv.2504.07139
2. Singla, A., Sukharevsky, A., Yee, L., Chui, M., & Hall, B. (2025). The state of AI: How organizations are rewiring to capture value. McKinsey & Company. https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-how-organizations-are-rewiring-to-capture-value
3. Shehabi, A., Smith, S. J., Hubbard, A., Newkirk, A., Lei, N., Siddik, M. A. B., Holecek, B., Koomey, J., Masanet, E., & Sartor, D. (2024). *2024 United States data center energy usage report*(Report No. LBNL-2001637). Lawrence Berkeley National Laboratory. https://doi.org/10.71468/P1WC7Q
4. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459-9474. https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
5. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., & Gurevych, I. (2021). BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the 35th Conference on Neural Information Processing Systems, Track on Datasets and Benchmarks*. https://doi.org/10.48550/arXiv.2104.08663
6. Wang, K., Thakur, N., Reimers, N., & Gurevych, I. (2022). GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 2345-2360). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.168
7. Dai, Z., Zhao, V. Y., Ma, J., Luan, Y., Ni, J., Lu, J., Bakalov, A., Guu, K., Hall, K., & Chang, M.-W. (2023). Promptagator: Few-shot dense retrieval from 8 examples. In *International Conference on Learning Representations (ICLR 2023)*. https://doi.org/10.48550/arXiv.2209.11755
8. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR 2022)*. https://doi.org/10.48550/arXiv.2106.09685

9. Biderman, D., Portes, J., Gonzalez Ortiz, J. J., Paul, M., Greengard, P., Jennings, C., King, D., Havens, S., Chiley, V., Frankle, J., Blakeney, C., & Cunningham, J. P. (2024). LoRA learns less and forgets less. *Transactions on Machine Learning Research*. https://doi.org/10.48550/arXiv.2405.09673

10. Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A. A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Cai, Q., Chaudhary, V., Chen, D., Chen, D., . . . Zhou, X. (2024). *Phi-3 technical report: A highly capable language model locally on your phone*. arXiv. https://doi.org/10.48550/arXiv.2404.14219

11. Gemma Team. (2025). *Gemma 3 technical report*. arXiv. https://doi.org/10.48550/arXiv.2503.19786

12. Cuconasu, F., Trappolini, G., Siciliano, F., Filice, S., Campagnano, C., Maarek, Y., Tonellotto, N., & Silvestri, F. (2024). The power of noise: Redefining retrieval for RAG systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*(pp. 719-729). Association for Computing Machinery. https://doi.org/10.1145/3626772.3657834

13. Chen, J., Lin, H., Han, X., & Sun, L. (2024). Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence, 38*(16), 17754-17762. https://doi.org/10.1609/aaai.v38i16.29728

14. Lin, X. V., Chen, X., Chen, M., Shi, W., Lomeli, M., James, R., Rodriguez, P., Kahn, J., Szilvasy, G., Lewis, M., Zettlemoyer, L., & Yih, S. (2024). RA-DIT: Retrieval-augmented dual instruction tuning. In *International Conference on Learning Representations (ICLR 2024)*. https://doi.org/10.48550/arXiv.2310.01352

15. Siriwardhana, S., Weerasekera, R., Wen, E., Kaluarachchi, T., Rana, R., & Nanayakkara, S. (2023). Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. *Transactions of the Association for Computational Linguistics, 11*, 1-17. https://doi.org/10.1162/tacl_a_00530

16. Zhang, T., Patil, S. G., Jain, N., Shen, S., Zaharia, M., Stoica, I., & Gonzalez, J. E. (2024). *RAFT: Adapting language model to domain specific RAG*. arXiv. https://doi.org/10.48550/arXiv.2403.10131

17. Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., & Wei, F. (2024). Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 11897-11916). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.642

18. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W. X., Dong, D., Wu, H., & Wang, H. (2021). RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 5835-5847). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.466

19. Jacob, M., Lindgren, E., Zaharia, M., Carbin, M., Khattab, O., & Drozdov, A. (2024). *Drowning in documents: Consequences of scaling reranker inference*. arXiv. https://arxiv.org/abs/2411.11767v1

20. Yoran, O., Wolfson, T., Ram, O., & Berant, J. (2024). Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations (ICLR 2024)*. https://doi.org/10.48550/arXiv.2310.01558

21. Jin, B., Yoon, J., Han, J., & Arık, S. Ö. (2025). Long-context LLMs meet RAG: Overcoming challenges for long inputs in RAG. In *International Conference on Learning Representations (ICLR 2025)*. https://doi.org/10.48550/arXiv.2410.05983

22. Soudani, H., Kanoulas, E., & Hasibi, F. (2024). Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region* (pp. 12-22). Association for Computing Machinery. https://doi.org/10.1145/3673791.3698415

23. Shen, M., Umar, M., Maeng, K., Suh, G. E., & Gupta, U. (2024). *Towards understanding systems trade-offs in retrieval-augmented generation model inference*. arXiv. https://doi.org/10.48550/arXiv.2412.11854

24. Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations* (pp. 150-158). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.eacl-demo.16

Copyright: © 2025 The Author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.