Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25-29 May 2026,Vancouver,BC

Smaller, Smarter, Greener: Reducing LLM Inference Emissions with RAG

Ethan Heavey[†,*], Paul Cook[†,+]
[†] University of New Brunswick

Abstract

The escalating computational demands of Large Language Models (LLMs) raise significant concerns regarding their environmental sustainability. While prior work has quantified training emissions, inference — which dominates a model's lifecycle carbon footprint — remains underexplored in holistic evaluations that jointly consider efficiency and effectiveness. This study investigates whether smaller models augmented with Retrieval-Augmented Generation (RAG) can achieve Pareto-optimal configurations that balance accuracy and carbon emissions better than larger, non-RAG models. We conduct experiments across three model families (DeepSeek-R1, Qwen3, Gemma 3) on two question answering datasets (HotpotQA, Natural Questions), measuring end-to-end emissions using CodeCarbon. Our results show that on Natural Questions, RAG enables models as small as 0.6B parameters to outperform 12B–32B models in terms of F_1 score with lower carbon emissions, in some cases achieving up to 90% emission reductions. However, on HotpotQA, the efficiency benefits are more nuanced, with RAG consistently improving F_1, but not always reducing emissions. Our work provides a systematic analysis of the efficiency–effectiveness trade-off of incorporating RAG, offering practical guidance for environmentally sustainable AI.

Keywords: Green AI, Retrieval-Augmented Generation, Pareto Optimality, Large Language Models

1. Introduction

The escalating computational demands of state-of-the-art Large Language Models (LLMs) raise concerns regarding their environmental sustainability and operational costs. In response, the field of Green AI [1] has emerged as a frontier in AI research, advocating for a greater emphasis on efficiency alongside raw performance.

While the carbon footprint of model training is well-documented [2, 3], a comprehensive understanding of inference-phase emissions is crucial. Existing work begins to quantify inference energy usage but often does so in isolation from traditional performance metrics [4, 5]. Noting that larger language models tend to use more energy, for both training and inference, than smaller language models [5], our work seeks to bridge that gap by conducting a holistic evaluation of whether smaller, retrieval-augmented models can be both more effective in terms of traditional evaluation metrics while also being more efficient in terms of energy usage and resulting carbon emissions than larger non-retrieval augmented models. We focus specifically on Retrieval-Augmented Generation (RAG) as it addresses a key limitation of static LMs: their inability to access current knowledge without retraining [6]. RAG compensates for reduced parametric knowledge in smaller models by providing relevant, up-to-date context from external knowledge bases, potentially enabling smaller models to match or exceed larger models' performance while consuming less energy.

Our investigation allows for the identification of Pareto-optimal approaches, which are configurations where no other model is simultaneously more accurate and more efficient. The study is guided by two research questions:

- **RQ1:** Can smaller LMs with RAG achieve higher effectiveness (measured by F_1 score) than larger LMs without RAG?
- **RQ2:** Can smaller LMs with RAG be more efficient, i.e., use less energy (resulting in lower carbon emissions), than larger LMs without RAG?

[*] ethan.heavey@unb.ca [+] paul.cook@unb.ca

Given the potential for external knowledge to compensate for reduced parametric knowledge in smaller models, we hypothesize that the answer to both research questions is "yes". We conduct experiments on two question answering datasets, HotpotQA and Natural Questions. On Natural Questions, we find that RAG can reduce emissions by up to 90% while outperforming models over 20 times larger in parameter count; however, on HotpotQA, a multi-hop question answering dataset, these benefits appear to be more nuanced, with RAG consistently improving F_1, but not always reducing emissions.

Although the focus of this study is on energy usage and carbon emissions, note that smaller models can also be better suited for deployment in resource-constrained environments or on edge devices, where, for example, the higher energy usage of larger models may be prohibitive [1]. Code to reproduce our experimental results is available at https://github.com/VeiledTee/LMPowerConsuption.

2. Related Work

"Green AI" — research prioritizing environmental efficiency — has been distinguished from "power-hungry AI" — research that pursues state-of-the-art performance regardless of computational cost [1]. This study notes that models are typically evaluated solely on traditional performance metrics (e.g., exact match and F_1 score) without reporting the energy cost.

The environmental impact of LLMs has been predominantly studied through the lens of training, a highly energy-intensive one-time event [2, 3]. However, focusing solely on training obscures the long-term carbon footprint of deployment. Inference costs can eventually dwarf training emissions over a model's lifetime, especially for widely adopted systems [4]. Despite growing awareness, evaluation paradigms lag behind: models are typically benchmarked on effectiveness (e.g., accuracy) alone, without reporting energy costs [1]. This lack of transparency is particularly problematic during inference, where energy measurement is complex but essential for holistic carbon accounting [4, 5]. Crucially, most prior work, with a notable exception being [4], does not jointly consider efficiency and effectiveness, making it impossible to identify Pareto-optimal configurations. Our work further contributes to addressing this gap.

A limitation of LMs is that, once trained, the out-of-the-box version of the model cannot provide "current" knowledge without further training. To bridge that gap, retrieval-augmented generation (RAG), a well-known method capable of reducing hallucinations in LMs and keeping their knowledge up to date [7, 8], can be used. RAG at its most simple is composed of two parts; a retrieval system and a generator [6]. This enables RAG systems to perform retrieval over up-to-date knowledge bases before relying on an LM to generate a natural language reply to the initial query.

Larger language models tend to use more energy, for both training and inference, than smaller language models [5]. In this paper, we consider whether a smaller LM with RAG can be more effective and more efficient than a larger language model without RAG.

It is important to note that RAG also introduces its own energy overhead. The energy required to perform the retrieval of relevant information to aid the generator model is essential to measure and account for. The longer prompt passed to the generator also increases the computational load during the generation step. This extra overhead must be weighed against the alternative; querying a model that relies solely on its internal parameters. Whilst the benefits of RAG are known [6], the impact it has on the efficiency–effectiveness trade-off, which we consider in this paper, is under-explored.

Beyond model size and retrieval, the recent advent of "reasoning" models introduces another critical variable in the efficiency–effectiveness discussion. Reasoning models often use a Chain-of-Thought (CoT) methodology in which they explicitly generate intermediate reasoning steps before producing a final answer [9]. Whilst this can improve performance on complex logical tasks, it increases token counts per response, directly correlating to higher emissions [5]. CoT models therefore introduce a trade-off; invest more energy to achieve better performance on the task [10]. In this study we explore this by considering both reasoning and non-reasoning models.

3. Methodology

This section describes our experimental methodology including the datasets and language models used, the experimental setup, and the approach to estimating carbon emissions.

3.1. Datasets

To evaluate the efficiency–effectiveness trade-off in RAG systems, we require datasets that provide both questions with answers and the supporting context that can be used to answer them. Furthermore, to accurately measure the full carbon footprint of a RAG pipeline, we must account for the energy cost of the retrieval step. This necessitates a dataset where context can be retrieved algorithmically from a standard corpus, allowing us to instrument and measure retrieval emissions.

The first dataset we consider, HotpotQA [11], is a multi-hop question-answering dataset designed to test the ability to reason across multiple supporting documents. Each question requires the synthesis of information from two or more Wikipedia paragraphs to arrive at the correct answer. The dataset provides the gold-standard answers, and supporting paragraphs, along with the full Wikipedia dump (2017-10-01) from which they were drawn. We consider two LM configurations for the HotpotQA experiments: query only (QO), in which only the query from the dataset is passed to the LM, and gold standard (GS), where we pass the "gold standard" supporting paragraphs associated with each query as additional context. As in prior work, we limit the test set to $1,000$ randomly sampled instances [4].

We use HotpotQA for three key reasons: (1) its multi-hop nature presents a non-trivial reasoning challenge where access to external knowledge (via retrieval) can be leveraged, making it an interesting testbed for RAG; (2) the provided Wikipedia dump allows us to implement a TF-IDF retrieval system algorithmically identical to that originally used to find relevant context for each question [11]; (3) by reproducing this retrieval process, we can measure the energy consumption of retrieval. For our experiments, we use the *fullwiki* setting from [11], where the system must retrieve relevant paragraphs from the full corpus containing both relevant and irrelevant documents.

The second dataset we consider, the Natural Questions (NQ) corpus [12], is a large-scale dataset built from real, anonymized Google search queries. Each instance contains a user question, a Wikipedia page that potentially contains the answer, a "short answer" (one or more named entities), and a "long answer" (a paragraph from the Wikipedia page that contains the short answer). The short answer is the gold-standard answer for QA evaluation.

Kwiatkowski et al. [12] consider a "first paragraph" baseline, where the system retrieves only the first paragraph from the Wikipedia page linked to the query. We add this approach to our configurations, providing us with three for NQ: QO, GS (where we provide the long answer paragraph as context), and first paragraph (FP) — where we provide the first paragraph of each query's associated Wikipedia page as context. We filter the NQ test set to include only instances that provide both a short and long answer, and then again randomly select $1,000$ instances.

We consider NQ because it is widely used for question answering evaluation, and allows an extra RAG configuration (i.e., FP) for another point of comparison in our experiments. Note, however, that we cannot directly measure the retrieval emissions of NQ as the long answer paragraphs were chosen by human annotators. We therefore estimate retrieval emissions for NQ based on measurements for HotpotQA (discussed further in Section 3.3).

3.2. Models

In our investigation, due to hardware limitations, we focus on language models that can fit on a single GPU. Due to the nature of our research questions, we also sought out model families that have at least four different model sizes at 32B parameters or lower, allowing for multiple points of

comparison. We choose models available on Ollama,[1] as we use this tool to run our experiments. Under these constraints we select the DeepSeek-R1 [13] (1.5B, 7B, 14B, 32B), Qwen3 ([**qwen3**], 0.6B, 1.7B, 4B, 8B, 14B, 32B), and Gemma 3 [14] (1B, 4B, 12B, 27B) models.

Each model is evaluated on both datasets with the QO configuration. Each model except the largest in each family (i.e., DeepSeek-R1 32B, Qwen3 32B, and Gemma 3 27B) is evaluated in each RAG configuration, i.e., GS on both datasets and FP in the case of the NQ dataset. We do not consider the largest model in each family using RAG because we are interested in comparisons of smaller models with RAG to larger models without RAG. Furthermore, this choice somewhat reduces the computation required to carry out the experiments.

The DeepSeek models are distinguished from Qwen3 and Gemma 3 by their integration of Chain-of-Thought (CoT) reasoning, allowing them to generate intermediate reasoning tokens before producing a final response [13].[2] While this paradigm enhances performance on complex reasoning tasks, it introduces an inference overhead; CoT models can generate more internal tokens than their standard counterparts, directly increasing the total energy consumption per request [5]. In the context of our investigation, these models in the QO configuration represent a high-resource benchmark; we seek to analyze our research questions with CoT and standard models in mind to determine if the effectiveness gains (in terms of F_1) afforded by this "reasoning" justify the resulting increase in emissions when compared to smaller, RAG configurations. Note that, following prior work [5], we do not constrain output token length for any model during inference. Token counts therefore reflect each model's native verbosity — including any CoT thinking tokens in the case of DeepSeek-R1 — and will vary across models and configurations.

3.3. Experimental Setup and carbon estimation

Our experimental pipeline for evaluating a model M on a dataset D proceeds as follows:

(1) **Retrieval:** For each question $q_i \in D$, we use a TF-IDF retriever to measure the energy consumption of fetching the most relevant passages from the Wikipedia corpus. The retrieved paragraphs are not used later in the pipeline — we perform this step only to measure the emissions of retrieval.

(2) **Prompt Construction:** The configuration determines if we provide additional context with the query or not. For QO configurations we only prompt the LM with the question. For GS and FP configurations, we concatenate the context provided by the dataset for question q_i with the original question q_i into a structured prompt.

(3) **Generation:** The prompt is fed to the target language model M, which generates a free-text completion. Again, no maximum token limit is imposed in generation.

(4) **Evaluation:** Text generated by M is evaluated against the ground-truth answer using F_1, which is the standard evaluation metric used for each dataset. For NQ, the ground-truth answer is the short answer.

(5) **Emission Accounting:** The total carbon emissions for the query are calculated as the sum of the retrieval emissions and the inference emissions.

We use the Code Carbon package [15] to measure carbon emissions. All of our experiments were run on a Windows 11 machine with a NVIDIA RTX 4090 24GB GPU, a Ryzen 7 7800X3D 8-core CPU, and 64GB RAM. Furthermore, all experiments were run in the same compute region (New Brunswick, Canada) which has a carbon intensity of $293.62 g CO_2 eq$/kWh.

To measure the retrieval emissions, for HotpotQA, we measure the emissions for retrieving context for all $1,000$ HotpotQA instances in a single, dedicated run. Note that we only do this once, for all HotpotQA experiments, because the retrieval setup (e.g., hardware, dataset, algorithm) is the same for all experiments using this dataset. As mentioned, we are unable to replicate the retrieval

[1] https://ollama.com/

[2] Qwen3 also has a CoT option; however, during preliminary experiments, we found that non-CoT Qwen3 models were more efficient and effective. We therefore only report results for Qwen3 without using CoT.

process used to find the long answer paragraphs from NQ because they were selected by human annotators. Therefore, since retrieval for both NQ and HotpotQA involves retrieving paragraphs from a Wikipedia-scale corpus, we use the measured retrieval emissions for HotpotQA as an estimate for the retrieval emissions for NQ.

To measure inference emissions, we instrument the generation function to begin tracking emissions right before providing the prompt to the LM, and stop tracking emissions immediately after the LM provides a response to the query.

4. Results

This section presents the efficiency–effectiveness trade-offs for parametric scaling versus retrieval augmentation. We first analyze trends within each model family and then synthesize cross-family insights to answer our research questions in their entirety. In analyzing results, we will often consider the Pareto frontier. In our analyses the Pareto frontier is the set of model configurations such that no other model configuration simultaneously achieves a higher F_1 score and lower emissions. In answering our research questions, we will focus on the QO configuration that achieves highest F_1 score as a point of comparison. We do this because, in some cases, the largest QO configuration achieves surprisingly low F_1 compared to smaller QO models.

4.1. Within-Family Analysis

Across all model families, we first validate the expected scaling trends: for Query-Only (QO) models, we expect that larger sizes should yield higher F_1 scores at the cost of increased emissions. For each family and dataset, we analyze these QO baselines before introducing the RAG configurations — GS and, in the case of the NQ dataset, FP. Holding model size constant, we then compare RAG configurations to their corresponding QO variants. We hypothesize that, when model size is fixed, RAG will trade higher effectiveness for lower efficiency, attributable to retrieval overhead and more input tokens processed at inference time leading to higher emissions [5]. Finally, to answer our research questions, we compare the RAG configurations against the QO model that achieves the highest F_1 score.

4.1.1. DeepSeek-R1

HotpotQA. Consistent with our expectations, larger QO models achieve higher F_1 scores at the cost of higher emissions (Figure 1, left; full results shown in Appendix A, Table 1). For example, F_1 rises from 0.07 to 0.18, while emissions increase by an order of magnitude, from QO 1.5B to QO 32B. When model size is fixed, each GS configuration follows the hypothesized trade-off, surpassing its QO counterpart in F_1 but with higher emissions.

To answer RQ1 and RQ2, we compare these GS configurations to the most-effective QO model, QO 32B. Several smaller GS models are more efficient and more effective than our point of comparison. Specifically, GS 7B and GS 14B both achieve a higher F_1 than QO 32B at a lower cost of emissions. Thus, for DeepSeek on HotpotQA, smaller LMs with RAG are more efficient and more effective than the best-performing, and larger, QO model. The answer to both of our research questions is therefore yes.

Natural Questions. The QO scaling trend also holds for NQ: larger models yield higher F_1 at the cost of emissions (Figure 1, right; full results shown in Appendix A, Table 1). Holding model size fixed and comparing RAG and QO configurations, we observe that each FP configuration achieves a higher F_1 while emitting *less* g CO_2eq than its QO counterpart. This result could be explained by the number of output tokens for each configuration (shown in Appendix A, Table 1); all FP configurations output fewer tokens than their QO counterpart. However, the scaling trend is less clear for the GS configurations. While GS 1.5B and 7B follow the expected trade-off (higher F_1, more emissions), GS 14B achieves a higher F_1 for *lower* emissions than QO 14B. Here we see the

same trend in number of output tokens as for the FP models; GS 14B outputs roughly 117k fewer tokens than QO 14B.

We now consider how smaller RAG configurations compare to the most effective QO model, QO 32B. We see that smaller FP and GS configurations outperform this model with higher F_1 and lower g CO_2eq. The most dramatic example of this is GS 1.5B, which achieves an F_1 of 0.24, surpassing the QO 32B model's F_1 of 0.14, while using only 14% of the emissions. For DeepSeek on NQ, the answer to both RQ1 and RQ2 is again a definitive yes, smaller RAG configurations can achieve higher F_1 with less emissions than a larger, non-RAG model.

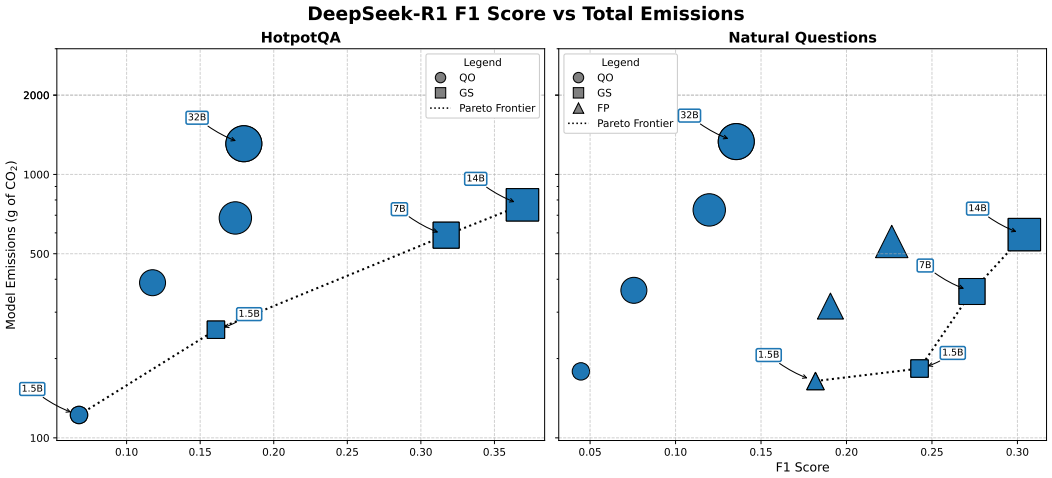

Figure 1. An efficiency–effectiveness comparison of the DeepSeek-R1 model family on the HotpotQA and NQ datasets. The y-axis is in log scale and shared between both plots. Model size is indicated by marker size.

4.1.2. **Qwen3**

HotpotQA. The expected scaling trends for QO models do not always hold; some larger models achieve higher F_1 scores at the cost of higher emissions, as expected, but there are exceptions (Figure 2, left; full results shown in Appendix A, Table 2). For example, QO 4B is the most power-hungry of all QO configurations, but does not achieve the highest F_1 score. Furthermore, QO 8B and QO 14B achieve higher F_1 than QO 0.6B and do so at the cost of *lower* emissions. When model size is fixed, most GS configurations follow the hypothesized trade-off, surpassing their QO counterparts in F_1 but with higher emissions. A notable exception is GS 1.7B, which achieves a higher F_1 than QO 1.7B (0.32 vs. 0.14) but emits far less CO_2 (32.52 vs. 53.76 g CO_2eq), due to generating only 5k output tokens compared to 81k.

To answer RQ1 and RQ2, we compare the GS configurations to the most effective QO model, QO 14B (0.26 F_1, 20.10 g CO_2eq). While smaller GS configurations achieve higher F_1 scores than QO 14B, none of them are more efficient. This appears to be due to differences in the number of output tokens. For example, GS 1.7B achieves 0.32 F_1 but outputs 81k tokens compared to QO 14B's 7k. Thus, for Qwen3 on HotpotQA, smaller RAG models are more effective, but not more efficient, than the best-performing QO model. The answer to RQ1 is yes but to RQ2 is no.

Natural Questions. The QO scaling trend is similar to what we see in the HotpotQA results; larger models generally yield higher F_1 at the cost of more emissions, although there are exceptions (Figure 2, right; full results shown in Appendix A, Table 2). In particular, QO 0.6B is less effective than QO 1.7B but uses more g CO_2eq, while QO 4B is the most effective of all QO configurations. Holding model size fixed and comparing RAG to QO configurations, there are only two (of the five)

FP configurations that match our expectations — FP 1.7B and FP 14B both achieve a higher F_1 than their QO counterpart at the cost of more emissions. However, the other three configurations deviated from our expectations; FP 0.6B, FP 4B, and FP 8B all achieve a higher F_1 score than their QO counterparts for *less* g CO_2eq. The GS configuration results also reflect a partial alignment with our initial hypotheses. For three of our five GS configurations,we observe increases in F_1 and emissions compared to their same-size QO counterparts. However, GS 1.7B and GS 4B both outperform their QO counterparts with higher F_1 for lower emissions.

We again focus on the most effective QO model, here QO 4B (0.24 F_1, 331.93 g CO_2eq), as our point of comparison. Multiple smaller FP and GS configurations outperform this model with higher F_1 and lower emissions. For example, GS 1.7B achieves an F_1 of 0.46 while using only 10% of the emissions (32.20 g vs. 331.93 g). Similarly, FP 8B (0.34 F_1, 41.52 g CO_2eq) and GS 0.6B (0.34 F_1, 28.80 g CO_2eq) both provide better effectiveness in terms of F_1 score at a fraction of the emissions.

For Qwen3 on NQ, while the patterns within same-size comparisons are mixed, the answer to both RQ1 and RQ2 is yes: smaller RAG models can be more effective (RQ1) and more efficient (RQ2) than larger non-RAG models.

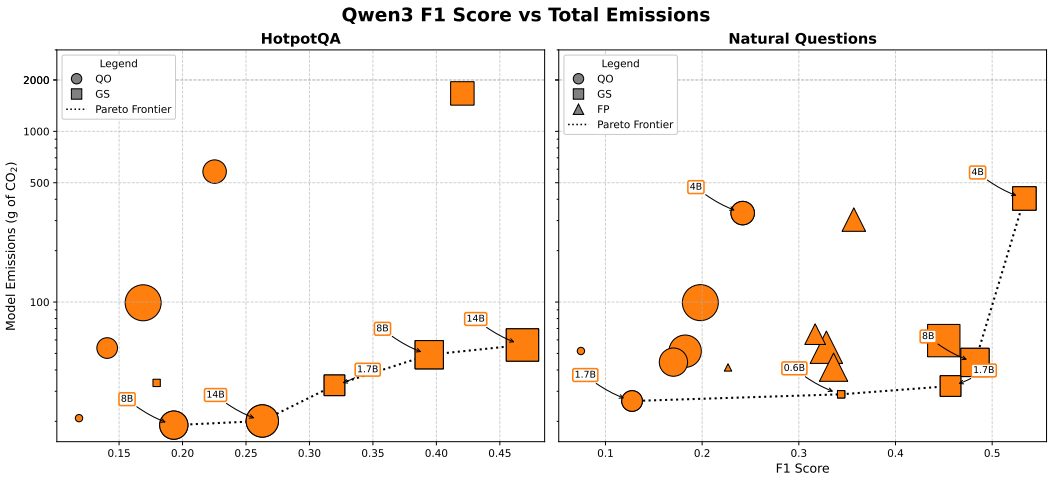

Figure 2. An efficiency–effectiveness comparison of the Qwen3 model family on the HotpotQA and NQ datasets. The y-axis is in log scale and shared between both plots. Model size is indicated by marker size.

4.1.3. **Gemma 3**

HotpotQA. Consistent with our expectations, larger QO models generally achieve higher F_1 scores at the cost of higher emissions (Figure 3, left; full results shown in Appendix A, Table 3). As an example, F_1 rises from 0.13 to 0.25, and emissions rise from 19.90 to 28.99 g CO_2eq when we increase QO configuration model size from 1B to 12B. The QO 27B model, however, is an exception, achieving by far the lowest F_1 (0.06) and highest emissions (261.75 g CO_2eq). When model size is fixed, each GS configuration follows the hypothesized trade-off, surpassing its QO counterpart in F_1 but with higher emissions.

To answer RQ1 and RQ2, we compare these GS configurations to the most-effective QO model, QO 12B (0.25 F_1, 28.99 g CO_2eq). While smaller GS configurations achieve higher F_1 scores, all are less efficient than QO 12B. For example, GS 4B achieves 0.38 F_1 but emits 31% more g CO_2eq than QO 12B. Thus, for Gemma 3 on HotpotQA, while RAG improves effectiveness (RQ1), smaller RAG models are not more efficient than the best-performing QO model. Although the answer to RQ1 here is yes, the answer to RQ2 is no.

Natural Questions. The QO scaling trend holds for NQ: larger models generally yield higher F_1 at the cost of higher emissions (Figure 3, right; full results shown in Appendix A, Table 3), although QO 27B again underperforms, with the lowest F_1 despite its size. Holding model size fixed and comparing RAG to QO configurations, the FP and GS configurations for both the 1B and 12B model sizes achieve higher F_1 than their QO counterparts at the cost of higher emissions. FP 4B and GS 4B, however, have a higher F_1 than QO 4B, but lower emissions. This could be due to differences in number of output tokens; FP 4B and GS 4B both only output 9k tokens, while QO 4B outputs 115k.

We again focus on the most effective QO model, QO 12B (0.30 F_1, 30.71 g CO_2eq), as our point of comparison. While most RAG configurations achieve higher F_1 scores at the cost of higher emissions, we note one exception: GS 1B achieves the same F_1 (0.30) with slightly lower emissions (29.70 g CO_2eq). This represents a small efficiency gain for the same level of effectiveness. This comparison tells us that here the answer to RQ2 is yes: a smaller RAG configuration, GS 1B, is more efficient than the most effective non-RAG configuration (QO 12B). Although the answer to RQ1 is not yes, it is remarkable that the much smaller GS 1B model is able to achieve the same F_1 score as QO 12B, the best non-RAG model.

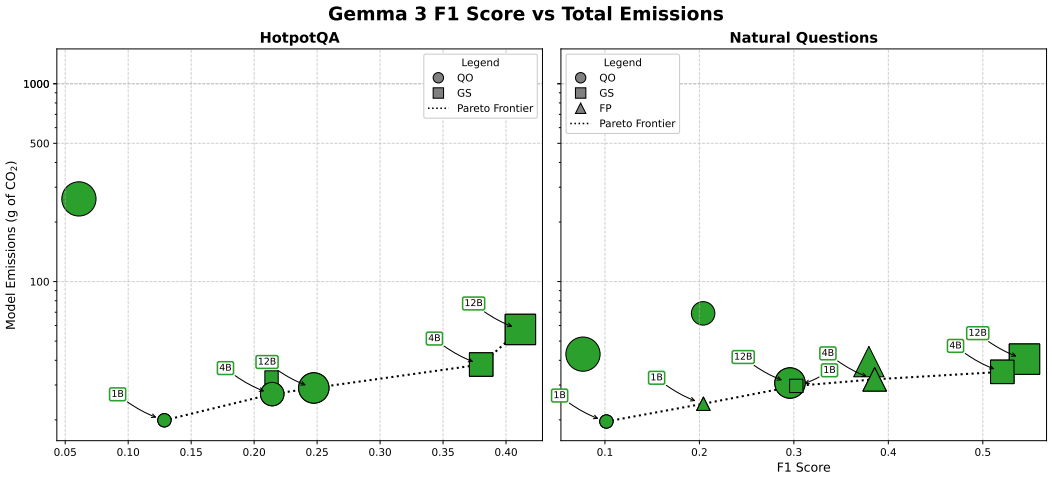

Figure 3. An efficiency–effectiveness comparison of the Gemma 3 model family on the HotpotQA and NQ datasets. The y-axis is in log scale and shared between both plots. Model size is indicated by marker size.

4.2. **Cross-Family Comparison**

This section synthesizes findings across the DeepSeek-R1, Qwen3, and Gemma 3 families to identify overall trends in the efficiency–effectiveness trade-off and determine the configurations on the global Pareto frontier.

HotpotQA. The results for HotpotQA, shown in Figure 4, indicate that RAG models are more effective than QO models. The models with highest F_1, i.e., the right side of Figure 4, are all GS configurations, which use RAG. It appears that RAG addresses the multi-hop reasoning requirements for HotpotQA more effectively than parametric scaling alone, at least for the model sizes considered. We observe that the Pareto frontier for HotpotQA consists of six points: two QO and four GS configurations. All GS configurations on the Pareto frontier achieve a higher F_1, but at the cost of higher emissions, than all QO configurations.

Notably, the Pareto frontier consists of Qwen3 and Gemma 3 models, but no DeepSeek models. The Deepseek models tend to have relatively high emissions compared to the others. Note that the DeepSeek models also tend to output orders of magnitude more tokens than the Qwen3 and Gemma

3 models, likely due to the thinking tokens these models produce. (Number of output tokens for all model configurations is shown in Appendix A, Tables 1, 2, and 3.)

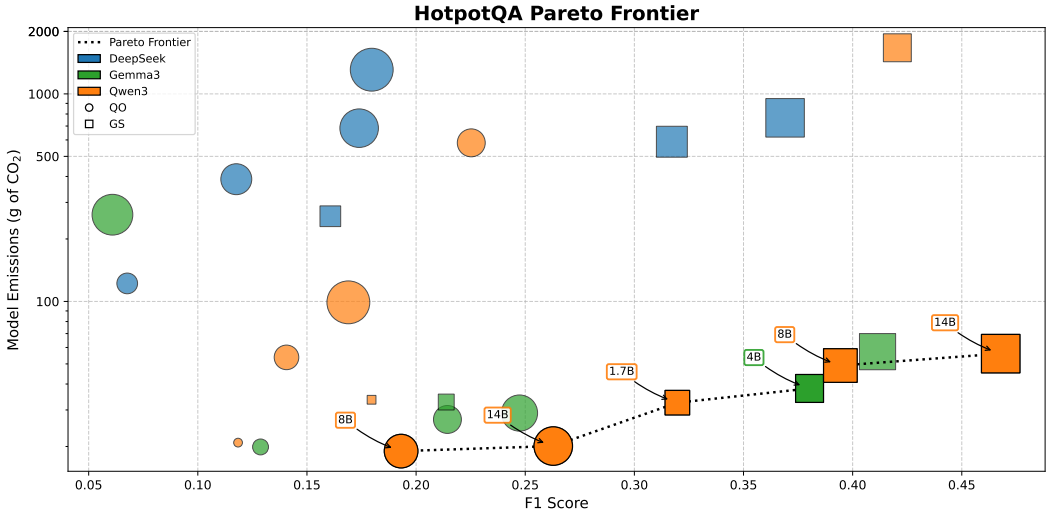

Figure 4. Efficiency and effectiveness for all model configurations on HopotQA. The y-axis is in log scale.

Natural Questions. The results for the NQ dataset are shown in Figure 5. We again see the advantages of RAG in terms of effectiveness; the models with highest F_1 are all RAG configurations. We again see that the DeepSeek models tend to have much higher emissions than the other models. The Pareto frontier includes configurations of each type: QO (once), FP (twice), and GS (four times).

An additional, and particularly interesting, point of comparison is the most effective QO model, Gemma 3 QO 12B, which is not on the Pareto frontier. Comparing Qwen3 GS 0.6B to Gemma 3 QO 12B, we see that Qwen3 GS 0.6B achieves a higher F_1 (0.34 vs. 0.30) for lower emissions (28.80 vs. 30.71 g CO_2eq). This further supports the finding that smaller RAG models can be both more effective and more efficient than non-RAG models.

Although on this dataset we see that smaller RAG models are able to be more effective and efficient than larger non-RAG models, the benefits of RAG were more nuanced on HotpotQA, where RAG consistently improved effectiveness, but did not always result in a more-efficient Pareto-optimal configuration compared to the best-performing non-RAG models. This finding may stem from differences in dataset complexity and its impact on output token length, a key driver of inference energy [5]. NQ is primarily a factoid retrieval task where answers are short entities; it appears that RAG helps models locate these concisely, often reducing output tokens (e.g., DeepSeek GS 14B generated 117k fewer tokens than its QO counterpart). In contrast, HotpotQA's multi-hop nature requires synthesizing information across paragraphs, a reasoning task that RAG appears to help, but at the cost of longer generations and increased emissions (e.g., Qwen3 GS 14B achieved the highest F_1 (0.47) but output 78k tokens versus QO 14B's 7k). Thus, RAG's carbon benefits appear to depend on whether retrieval reduces the generative burden (NQ) or merely shifts it to more complex reasoning (HotpotQA).

5. Conclusions

This study investigated the efficiency–effectiveness trade-offs of RAG through the lens of Green AI, specifically addressing whether smaller RAG models can be more effective than larger non-RAG counterparts in terms of F_1, while also being more efficient in terms of carbon emissions. Our findings provide a compelling, albeit nuanced, confirmation that this is the case.

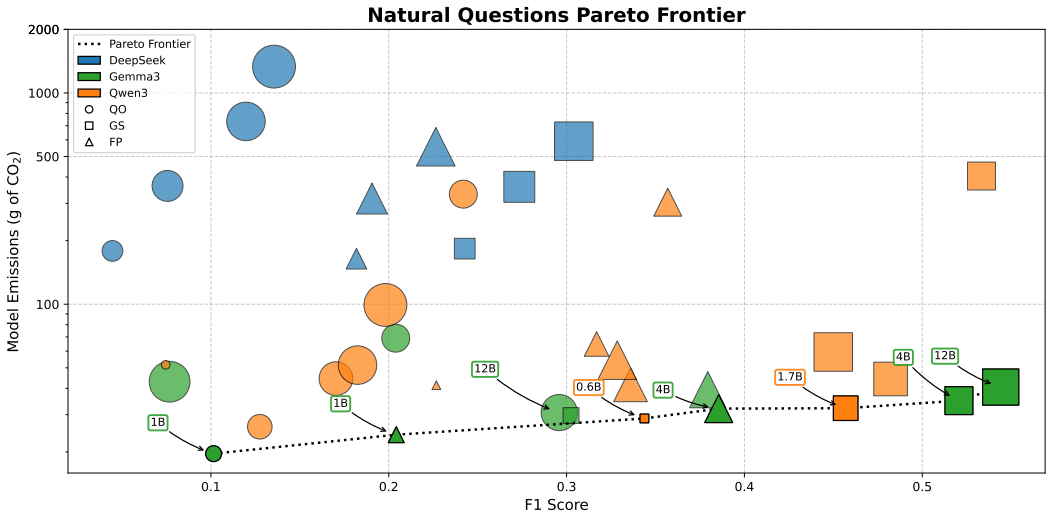

Figure 5. Efficiency and effectiveness for all model configurations on NQ. The y-axis is in log scale.

On the Natural Questions (NQ) dataset, we demonstrated that RAG enables substantial efficiency gains; models as small as 0.6B parameters were able to outperform 12B–32B models in terms of F_1 (RQ1) with lower carbon emissions (RQ2), in some cases achieving up to 90% emission reductions. For instance, the DeepSeek-R1 GS 1.5B configuration achieved an F_1 of 0.24 — surpassing the QO 32B model's 0.14 — while producing only 14% of the emissions. However, the benefits were more nuanced on the HotpotQA dataset. While RAG consistently improved effectiveness (RQ1), it did not always result in a more-efficient Pareto-optimal configuration compared to the best-performing non-RAG models (RQ2).

Ultimately, while RAG introduces its own energy overhead during retrieval and prompt processing, it remains a powerful tool for developing environmentally sustainable AI. By prioritizing augmenting smaller models, practitioners can potentially build more accurate QA systems that produce lower carbon emissions when used.

Our experiments were conducted on two QA datasets — HotpotQA and Natural Questions — which, while representative of factoid and multi-hop QA tasks, respectively, may not capture the full diversity of tasks to which RAG can be applied. Future work could consider additional QA datasets and additional tasks, such as fact verification and open-domain relation extraction. Evaluation in real-world deployment scenarios — such as domain-specific QA for healthcare or legal settings — could further validate the practical applicability of smaller retrieval-augmented models as a sustainable alternative to larger models which rely solely on parametric knowledge. Our evaluation considered three model families (DeepSeek-R1, Qwen3, and Gemma 3); results could differ for other architectures or training paradigms. Future work could therefore also consider additional model families such as Qwen3.5 [16], Gemma 4,[3] or NVIDIA's Nemotron-Cascade model [17].

Following prior work [5], we did not constrain output token length during inference. However, we observed that token output counts varied substantially across models and configurations. We further noted that some observed differences in efficiency between model configurations could be due to differences in number of output tokens. As such, future work could further explore efficiency and effectiveness under a constraint on output token length.

Emissions for NQ were estimated based on those measured for HotpotQA, as the original NQ retrieval process relied on human annotators and so could not be replicated algorithmically. Both datasets involve retrieving paragraphs from a Wikipedia-scale corpus, and as such we believe this is a

[3]Model card available at https://ai.google.dev/gemma/docs/core/model_card_4

reasonable approximation. Furthermore, retrieval emissions (4.22 g CO_2eq for all 1,000 HotpotQA instances, Appendix A, Table 2) accounted for a relatively small proportion of total emissions in large models (e.g., $< 1\%$ for Qwen3 GS 4B on HotpotQA) and remained a minority even in smaller models (e.g., $< 13\%$ for Qwen3 GS 1.7B on HotpotQA), suggesting that our overall findings are robust to this approximation. Nevertheless, future work could further consider retrieval emissions and explore alternative retrieval methods, such as more-contemporary dense retrieval approaches [e.g., 18]. Other directions for future work include exploring the energy overhead of chain-of-thought reasoning more systematically, as well as the impact of specific software stacks and decoding strategies [19].

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), RGPIN-2023-5871.

References

[1] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. "Green AI". In: *Commun. ACM* 63.12 (Nov. 2020), 54–63. ISSN: 0001-0782. DOI: 10.1145/3381831. URL: https://doi.org/10.1145/3381831.

[2] A. S. Luccioni, S. Viguier, and A.-L. Ligozat. "Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model". In: *Journal of Machine Learning Research* 24.253 (2023), pp. 1–15. URL: http://jmlr.org/papers/v24/23-0069.html.

[3] D. A. Patterson, J. Gonzalez, Q. V. Le, C. Liang, L.-M. Munguía, D. Rothchild, D. R. So, M. Texier, and J. Dean. *Carbon Emissions and Large Neural Network Training*. 2021. URL: https://api.semanticscholar.org/CorpusID:233324338.

[4] S. Luccioni, Y. Jernite, and E. Strubell. "Power Hungry Processing: Watts Driving the Cost of AI Deployment?" In: *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*. Rio de Janeiro, Brazil, 2024, 85–99. URL: https://doi.org/10.1145/3630106.3658542.

[5] S. Poddar, P. Koley, J. Misra, N. Ganguly, and S. Ghosh. "Towards Sustainable NLP: Insights from Benchmarking Inference Energy in Large Language Models". In: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the ACL: HLT (Volume 1: Long Papers)*. Ed. by L. Chiruzzo, A. Ritter, and L. Wang. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 12688–12704. URL: https://aclanthology.org/2025.naacl-long.632/.

[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. "Retrieval-augmented generation for knowledge-intensive NLP tasks". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546. URL: https://api.semanticscholar.org/CorpusID:218869575.

[7] 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*. 2024. arXiv: 2312.10997 [cs.CL]. URL: https://arxiv.org/abs/2312.10997.

[8] O. Ayala and P. Bechard. "Reducing hallucination in structured outputs via Retrieval-Augmented Generation". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*. Ed. by Y. Yang, A. Davani, A. Sil, and A. Kumar. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 228–238. URL: https://aclanthology.org/2024.naacl-industry.19/.

[9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[10] E. Strubell, A. Ganesh, and A. McCallum. "Energy and Policy Considerations for Deep Learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by A. Korhonen, D. Traum, and L. Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. URL: https://aclanthology.org/P19-1355/.

[11] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Ed. by E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 2369–2380. DOI: 10.18653/v1/D18-1259. URL: https://aclanthology.org/D18-1259/.

[12] T. Kwiatkowski et al. "Natural Questions: A Benchmark for Question Answering Research". In: *Transactions of the Association for Computational Linguistics* 7 (2019). Ed. by L. Lee, M. Johnson, B. Roark, and A. Nenkova, pp. 452–466. URL: https://aclanthology.org/Q19-1026/.

[13] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

[14] G. Team et al. *Gemma 3 Technical Report*. 2025. arXiv: 2503.19786 [cs.CL]. URL: https://arxiv.org/abs/2503.19786.

[15] B. Courty et al. *mlco2/codecarbon: v2.4.1*. Version v2.4.1. May 2024. URL: https://doi.org/10.5281/zenodo.11171501.

[16] Q. Team. *Qwen3.5-Omni Technical Report*. 2026. arXiv: 2604.15804 [cs.CL]. URL: https://arxiv.org/abs/2604.15804.

[17] B. Wang, C. Lee, N. Lee, S.-C. Lin, W. Dai, Y. Chen, Y. Chen, Z. Yang, Z. Liu, M. Shoeybi, B. Catanzaro, and W. Ping. *Nemotron-Cascade: Scaling Cascaded Reinforcement Learning for General-Purpose Reasoning Models*. 2026. arXiv: 2512.13607 [cs.CL]. URL: https://arxiv.org/abs/2512.13607.

[18] Q. Zhang, S. Chen, D. Xu, Q. Cao, X. Chen, T. Cohn, and M. Fang. "A Survey for Efficient Open Domain Question Answering". In: *Proceedings of the 61st Annual Meeting of the ACL (Vol 1)*. Ed. by A. Rogers, J. Boyd-Graber, and N. Okazaki. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 14447–14465. URL: https://aclanthology.org/2023.acl-long.808/.

[19] J. Fernandez, C. Na, V. Tiwari, Y. Bisk, S. Luccioni, and E. Strubell. "Energy Considerations of Large Language Model Inference and Efficiency Optimizations". In: *Proceedings of the 63rd Annual Meeting of the ACL (Volume 1: Long Papers)*. Ed. by W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 32556–32569. URL: https://aclanthology.org/2025.acl-long.1563/.

Appendix A. Tables

Dataset	Model	Context	F_1	Output Tokens (k)	Total g CO_2eq	Inference g CO_2eq	Retrieval g CO_2eq
HotpotQA	1.5B	QO	0.07	328	122.13	122.13	0.00
		GS	0.16	607	257.31	253.09	4.22
	7B	QO	0.12	468	388.11	388.11	0.00
		GS	**0.32**	**643**	**588.13**	**583.90**	**4.22**
	14B	QO	0.17	449	683.63	683.63	0.00
		GS	**0.37**	**451**	**766.82**	**762.60**	**4.22**
	32B	*QO*	*0.18*	*429*	*1307.89*	*1307.89*	*0.00*
NQ	1.5B	QO	0.04	438	178.78	178.78	0.00
		FP	**0.18**	**379**	**164.31**	**160.09**	**4.22**
		GS	**0.24**	**416**	**183.29**	**179.06**	**4.22**
	7B	QO	0.08	420	363.12	363.12	0.00
		FP	**0.19**	**349**	**316.44**	**312.22**	**4.22**
		GS	**0.27**	**388**	**359.41**	**355.18**	**4.22**
	14B	QO	0.12	475	733.36	733.36	0.00
		FP	**0.23**	**348**	**556.50**	**552.28**	**4.22**
		GS	**0.30**	**358**	**591.06**	**586.84**	**4.22**
	32B	*QO*	*0.14*	*434*	*1332.48*	*1332.48*	*0.00*

Table 1. Results for all DeepSeek-R1 models and configurations across HotpotQA and NQ. The most effective (highest F_1) QO model on each dataset is shown in italics. RAG methods that achieve equal or greater effectiveness (F_1) with greater efficiency (Total g CO_2eq) than the italicized query only method are shown in boldface.

Dataset	Model	Context	F_1	Output Tokens (k)	Total g CO_2eq	Inference g CO_2eq	Retrieval g CO_2eq
HotpotQA	0.6B	QO	0.12	10	20.88	20.88	0.00
		GS	0.18	18	33.59	29.37	4.22
	1.7B	QO	0.14	81	53.76	53.76	0.00
		GS	0.32	5	32.52	28.30	4.22
	4B	QO	0.23	974	580.94	580.94	0.00
		GS	0.42	2648	1668.02	1663.80	4.22
	8B	QO	0.19	8	19.00	19.00	0.00
		GS	0.39	8	49.11	44.89	4.22
	14B	*QO*	*0.26*	*7*	*20.10*	*20.10*	*0.00*
		GS	0.47	8	56.03	51.81	4.22
	32B	QO	0.17	26	99.00	99.00	0.00
NQ	0.6B	QO	0.07	98	51.67	51.67	0.00
		FP	**0.23**	**50**	**41.39**	**37.16**	**4.22**
		GS	**0.34**	**13**	**28.80**	**24.58**	**4.22**
	1.7B	QO	0.13	15	26.31	26.31	0.00
		FP	0.32	78	64.85	60.62	4.22
		GS	**0.46**	**13**	**32.20**	**27.98**	**4.22**
	4B	QO	0.24	514	331.93	331.93	0.00
		FP	0.36	467	304.13	299.90	4.22
		GS	0.53	596	404.34	400.12	4.22
	8B	QO	0.17	28	44.45	44.45	0.00
		FP	**0.34**	**17**	**41.52**	**37.30**	**4.22**
		GS	**0.48**	**16**	**44.28**	**40.05**	**4.22**
	14B	*QO*	*0.18*	*22*	*51.51*	*51.51*	*0.00*
		FP	0.33	19	54.35	50.13	4.22
		GS	0.45	19	59.43	55.21	4.22
	32B	QO	0.20	26	99.20	99.20	0.00

Table 2. Results for all Qwen3 models and configurations across HotpotQA and NQ. The most effective (highest F_1) QO model on each dataset is shown in italics. RAG methods that achieve equal or greater effectiveness (F_1) with greater efficiency (Total g CO_2eq) than the italicized QO method are shown in boldface.

Dataset	Model	Context	F_1	Output Tokens (k)	Total g CO_2eq	Inference g CO_2eq	Retrieval g CO_2eq
HotpotQA	1B	QO	0.13	4	19.90	19.90	0.00
		GS	0.21	8	32.75	28.53	4.22
	4B	QO	0.21	5	27.00	27.00	0.00
		GS	0.38	5	38.08	33.86	4.22
	12B	*QO*	*0.25*	*5*	*28.99*	*28.99*	*0.00*
		GS	0.41	4	57.28	53.06	4.22
	27B	QO	0.06	90	261.75	261.75	0.00
NQ	1B	QO	0.10	6	19.62	19.62	0.00
		FP	0.20	6	24.12	19.90	4.22
		GS	**0.30**	**7**	**29.70**	**25.48**	**4.22**
	4B	QO	0.20	115	69.08	69.08	0.00
		FP	0.39	9	32.05	27.83	4.22
		GS	0.52	9	34.97	30.74	4.22
	12B	*QO*	*0.30*	*8*	*30.71*	*30.71*	*0.00*
		FP	0.38	8	39.40	35.18	4.22
		GS	0.54	10	40.55	36.32	4.22
	27B	QO	0.08	10	42.98	42.98	0.00

Table 3. Results for all Gemma 3 models and configurations across HotpotQA and NQ. The most effective (highest F_1) QO model on each dataset is shown in italics. RAG methods that achieve equal or greater effectiveness (F_1) with greater efficiency (Total g CO_2eq) than the italicized QO method are shown in boldface.