Inference Scaling Law for Retrieval Augmented Generation

Shu Zhou[1,2,3*], **Yuxuan Ao**[1,2,3*], **Yunyang Xuan**[1,2,3*], **Xin Wang**[5], **Tao Fan**[4], **Hao Wang**[1,2,3†]

[1]School of Information Management, Nanjing University, China
[2]Key Laboratory of Data Engineering and Knowledge Services in Jiangsu Provincial Universities (Nanjing University), China
[3]Jiangsu International Joint Informatics Laboratory, Nanjing University, China
[4]School of Public Administration, Nanjing University of Finance & Economics, China
[5]Baidu Inc, Beijing, China
shuzhou@smail.nju.edu.cn, yxao@smail.nju.edu.cn, yunyangxuan@smail.nju.edu.cn
xinwang2749@gmail.com, fantao0916@gmail.com, ywhaowang@nju.edu.cn

Abstract

Retrieval-augmented generation (RAG) has recently emerged as a powerful framework for knowledge-intensive natural language processing tasks, which leverages the strengths of both pre-trained language models and external knowledge. While significant progress has been made, the scaling behavior of these approaches during inference remains poorly understood. Towards this end, this paper presents a comprehensive study of *inference scaling law* for RAG models, which investigates how inference performance scales with respect to key factors including retriever model scale, generator model scale, number of retrieved documents, and context window size. Through extensive experiments on benchmark datasets, we establish empirical scaling laws that reveal power-law and sigmoid-type relationships between these factors and performance. We further build a joint *inference scaling law* with theoretical justification. With the proposed scaling laws, we can understand the performance tendency of RAG models under different computational resources. We believe our insights can pave the way for efficient and effective deployment of RAG models in more applications.

Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks (Brown et al. 2020). Building on this foundation, RAG models have emerged as a powerful paradigm, which enhances LLMs by seamlessly integrating external knowledge from a knowledge base (Lewis et al. 2020; Zhou et al. 2024). By conditioning the generation on retrieved documents from a knowledge database, RAG models have shown superior performance in various applications, including question answering (Zhou et al. 2025b), dialogue systems (Zhou et al. 2025d), and content creation (Shi et al. 2023).

In literature, there have been a range of RAG approaches proposed in recent years (Fan et al. 2024). Generally, RAG approaches follow two crucial steps, i.e. retrieval and generation. During retrieval, by comparing the similarity between them and the given query these approaches identify the relevant documents, which can be leveraged to expand the input for knowledge enhancement. During generation, they adopt LLMs to generate the desirable outputs with the expanded input in an autoregressive manner. There are also various advanced RAG approaches which enhance the retrieval process through indexing and postprocessing (Yang et al. 2024; Zamani and Bendersky 2024).

Despite the success of RAG, a principled understanding of how performance scales with various factors during inference remains underexplored. Although *scaling laws* have been explored thoroughly for LLMs (Xiong et al. 2024; Wang et al. 2024; Kaplan et al. 2020), they are not readily applicable in RAG models since they involve a complex interaction between the retriever and the generator, which brings in various new factors such as the number of retrieved documents and context window size. Given that the computational cost of RAG models is substantial, a comprehensive *inference scaling law* is highly anticipated to understand the trade-off between efficiency and effectiveness.

To bridge this gap, we introduce the first *inference scaling law* for RAG models. In particular, our aim is to develop a quantitative framework to understand how the inference performance of RAG models is influenced by several key factors including the retriever model scale, the generator model scale, the number of retrieved documents, and the context window size. We conduct extensive experiments by systematically varying their factors and measuring the inference performance on various benchmark datasets. Our empirical results reveal predictable scaling patterns including power-law and sigmoid-type relationships between these factors and RAG performance. On this basis, we develop a joint *inference scaling law*, which is justified by theoretical analysis. The proposed *scaling law* is also applied to provide insights into the performance trends of RAG models under varying computational resources.

Our main contributions can be summarized as follows: ❶ *New Perspective.* We are the first to study *inference scaling law* for retrieval-augmented generation models, which involve the retriever model scale, the generator model scale, the number of retrieved documents, and the context window

*These authors contributed equally.
†Corresponding author.

size. ❷ *Experiment Findings.* Extensive experiments on several benchmark datasets establish *inference scaling law* that characterizes the relationship between crucial factors and model performance. ❸ *Applications.* We apply our *inference scaling law* to different scenarios with varying computational resources and investigate the performance trends of RAG models.

Related Work

Scaling Laws in Large Language Models

Scaling laws (Urbizu et al. 2023; Zhuocheng et al. 2023; Wang et al. 2024; Tay et al. 2023) have significantly impacted the development of large language models (LLMs) by elucidating the relationship between model size, training data, and performance (Kaplan et al. 2020; Hoffmann et al. 2022). These insights have driven the creation of progressively larger models, like GPT-3 (Brown et al. 2020), PaLM (Chowdhery et al. 2022), and Gopher (Rae et al. 2021a), leading to substantial advancements in NLP. Research has also explored how factors such as model architecture (Rae et al. 2021b), data quality (Gao et al. 2020), and techniques like Mixture of Experts (Lepikhin et al. 2020) affect scaling. The exploration of *scaling laws* has expanded to multilingual learning (Conneau et al. 2020), code generation (Chen et al. 2021), and multimodal learning (Radford et al. 2021; Zhou et al. 2025a). As RAG models often use LLMs as their generator, these *scaling laws* are crucial for understanding RAG's inference performance, particularly regarding generator model size and its interaction with other scaling dimensions.

Retrieval Augmented Generation (RAG)

RAG models have emerged as a powerful approach for knowledge-intensive tasks, combining pre-trained language models with external knowledge retrieval (Zhou et al. 2025c; Lewis et al. 2020; Guu et al. 2020). RAG models enhance LLM capabilities by using retrieved documents to improve accuracy and reliability. (Lewis et al. 2020; Izacard et al. 2022) RAG models often use a retriever to select relevant documents and a generator to produce the output based on the query and the retrieved context (Lewis et al. 2020; Guu et al. 2020). Different retrieval (Karpukhin et al. 2020; Luan et al. 2021) and information fusion methods have been explored (Khattab and Zaharia 2020; Lin et al. 2023). RAG models have been applied in diverse applications, such as question answering, dialogue, and text generation (Lewis et al. 2020; Komeili, Lin, and Ren 2023; Lin et al. 2023; Izacard et al. 2022). However, the scaling behavior of RAG models during inference still remains largely unexplored.

Resource Allocation for Deep Learning

Efficient computational resource allocation is vital for deep learning model deployment, affecting both training and inference (Ji et al. 2024; Yao et al. 2024). Research has addressed resource allocation through techniques like gradient checkpointing and distributed training (Jain et al. 2019; NVIDIA 2022), model compression via pruning, quantization, and knowledge distillation (Verbraeken et al. 2020), and hardware selection for optimized performance (Zhang et al. 2022). Neural architecture search (NAS) also aims to automate the design of resource-efficient models (Elsken, Metzen, and Hutter 2019). However, these methods primarily target single-component models and do not address the unique challenges of RAG's dual-component architecture, where resources are divided between retriever and generator during inference. This allocation is crucial as it directly affects the trade-off between retrieval latency, generation latency, and overall model accuracy. This work aims to partially fill this gap by investigating resource allocation strategies tailored for RAG inference.

Methodology

This section outlines the methodology employed to investigate *inference scaling law* for RAG models. We detail the problem formulation, the model architecture used, the data and the evaluation protocol.

Problem Formulation

The core problem addressed in this work is to characterize the *inference scaling laws* for RAG models. We aim to understand and quantify how various factors influence RAG model performance during inference. These factors include the retriever model size (S_r), generator model size (S_g), number of retrieved documents (N_d) and context window size (C_w). We formulate the problem as:

$$\mathcal{P} = f(S_r, S_g, N_d, C_w), \tag{1}$$

where $\mathcal{P}$ represents a performance metric and f describes the relationship between these parameters and RAG model performance. Our goal is to discover and model the form of f, exploring the individual and combined impacts of these factors to derive an empirical model capturing the *scaling laws* of RAG inference performance.

Model Architecture

Retriever. The retriever encodes queries and documents into a shared embedding space using a dual-encoder architecture. We use Transformer-based encoders, specifically models from the BERT and GTR (Ni et al. 2022), with the [CLS] token embedding used for the representation. We employ FAISS (Douze et al. 2024) for fast retrieval of the top-K documents based on the inner product similarity.

Generator. The generator synthesizes the output text from the query and retrieved documents. We use T5 and OPT models for the generator. Inputs are concatenated, with the retrieved documents separated by a special [SEP] token. We use beam search for decoding.

Model Parameter Settings. We investigate the impact of model size by using models of different parameter sizes (BERT-base, BERT-large, GTR-XL, GTR-XXL, T5-small, T5-base, T5-large, T5-3b, T5-11b, OPT-1.3B, OPT-2.7B, OPT-6.7B, OPT-13B, OPT-30B). Embedding dimensions are standardized to 768.

RAG Model Inference Process. The inference process is formulated as:

$$v_q = f_r(q; \theta_r), \tag{2}$$

$$v_{d_i} = f_r(d_i; \theta_r), \forall d_i \in \mathcal{C}, \tag{3}$$

$$D = \text{Top-k}(v_q, \{v_{d_i}\}), \tag{4}$$

$$y = f_g(q, D; \theta_g), \tag{5}$$

where q is the query, f_r is the Retriever, θ_r are the parameters of the retriever, v_q is the query embedding, v_{d_i} is the embedding of document d_i in corpus $\mathcal{C}$, D is the set of top-k retrieved documents, f_g is the generator, θ_g are the parameters of generator, and y is the generated text.

Dataset

We use benchmark datasets for knowledge-intensive tasks for verification, which focus on question answering, including Natural Questions (NQ) (Kwiatkowski et al. 2019), TriviaQA (Joshi et al. 2017), WebQuestions (WQ) (Berant et al. 2013), and FEVER (Thorne et al. 2018). For retrieval, we use the entire Wikipedia corpus as the document collection. While the gold documents provided in the datasets are used for evaluation, the retrieval process is performed over the entire Wikipedia corpus.

Evaluation Protocol and Experimental Setup

We extensively evaluate RAG models in a zero-shot setting where neither the retriever nor the generator is fine-tuned on the downstream tasks. Here, we measure performance on the held-out test sets of the respective datasets.

Evaluation Metrics. We use the F1 score for question answering and also measure inference latency.

Scaling Dimensions. We investigate the following scaling dimensions. Retriever model size (S_r): BERT-base (110M), BERT-large (340M), GTR-XL (1.24B), GTR-XXL (4.8B). Generator model size (S_g): T5-small (60M), T5-base (220M), T5-large (770M), T5-3B (3B), T5-11B (11B), OPT-1.3B (1.3B), OPT-2.7B (2.7B), OPT-6.7B (6.7B), OPT-13B (13B), OPT-30B (30B). Number of retrieved documents (N_d): 1 to 500. Context window size (C_w): 1 to 2048 tokens.

Experimental Procedure. We vary each dimension while keeping others fixed. For example, when varying N_d, we fix the retriever to BERT-base, the generator to T5-11B, and C_w to 2048.

Hardware and Software. All experiments are conducted on a server with 2 NVIDIA H100 GPUs. We use PyTorch and the Hugging Face Transformers library for implementation.

Statistical Significance. We use paired t-tests to test the statistical significance of observed performance differences in this work.

Inference Scaling Law for RAG Models

Retriever Model Size Scaling

Here, we investigate the impact of the retriever model size (S_r) on the inference performance of RAG models in a zero-shot setting. We conduct comprehensive experiments by varying the size of the pre-trained retriever model within the same model family while keeping the generator model and other parameters fixed.

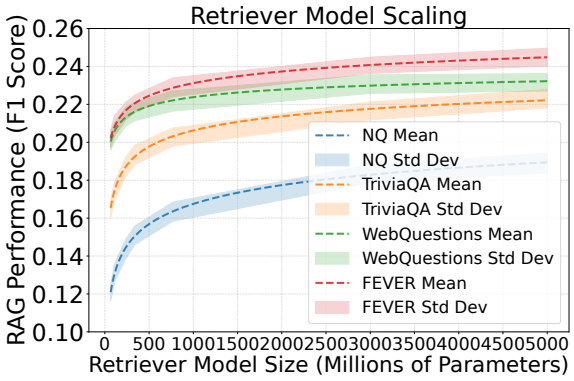

Figure 1: RAG performance with varying retriever model size. The dashed lines represent the fitted curves with different datasets.

Dataset	A	α	$\delta\eta$	R-squared
NQ	0.284	0.111	0.014	0.999
TriviaQA	0.223	0.187	0.043	0.999
WebQuestions	0.134	0.238	0.115	0.996
FEVER	0.181	0.106	0.137	0.997

Table 1: Fitted parameters of results for retriever model size scaling.

Experimental Setup. We employ a zero-shot RAG approach, where neither the retriever nor the generator is fine-tuned on the downstream tasks. We utilize pre-trained models of different sizes as retrievers: GTR-base (110M), GTR-Large (335M), GTR-XL (1.24B), GTR-XXL (4.8B). The generator is fixed to a pre-trained model T5-11B, the number of retrieved documents (N_d) is set to 100, and the context window size (C_w) is set to 512. For each model and dataset combination, we perform 50 independent inference runs and report the mean and standard deviation of the F1 score.

Scaling Law Fitting Results. Figure 1 illustrates the RAG model performance as a function of the retriever model size. Based on the experimental results, we fit the following power-law scaling function to the observed data:

$$\mathcal{P}(S_r) = A(1 - S_r^{-\alpha}) + \delta\eta, \tag{6}$$

where $\mathcal{P}(S_r)$ is the RAG performance, S_r is the retriever model size, and A, α, and $\delta\eta$ are the fitted parameters. We fit the *scaling law* using the least squares method. The fitting parameters are summarized in Table 1, and the fitness is indicated by the R-squared value. The fitting results for retriever model scaling offer several key insights: ❶ The experiments validate a power-law relationship between the retriever model size and RAG inference performance. ❷ The scaling curves also exhibit a saturation trend, indicating diminishing returns as the retriever model size increases beyond a certain point. ❸ The *scaling law* parameters vary across datasets, suggesting that the optimal retriever model size and the scaling behavior are influenced by dataset characteristics.

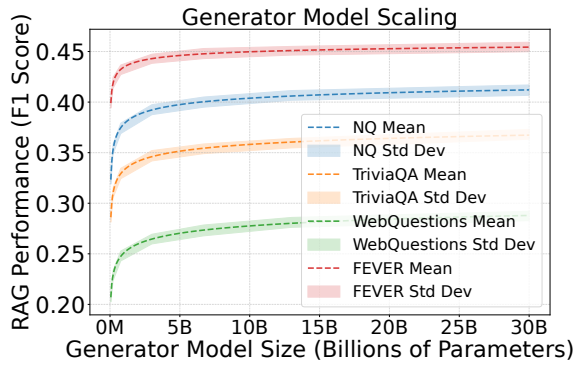

Figure 2: The RAG performance with varying generator model size.

Dataset	C	B	α	R-squared
NQ	0.298	0.294	0.223	0.999
TriviaQA	0.266	0.248	0.159	0.999
WebQuestions	0.362	0.252	0.118	0.999
FEVER	0.468	0.198	0.255	0.997

Table 2: Fitted parameters of results for generator model size scaling.

Generator Model Size Scaling

We investigate the impact of the generator model size (S_g) on the inference performance of RAG models. We conduct experiments by varying the size of the generator model while keeping the retriever model and other parameters fixed.

Experimental Setup. We also employ a zero-shot RAG approach. We utilize pre-trained models of different sizes as generators: OPT-1.3B(1300M), OPT-2.7B (2700M), OPT-6.7B (6700M), OPT-13B (13000M) and OPT-30B (30000M). We fix the retriever to BERT-large, and the number of retrieved documents (N_d) is set to 50, and the context window size (C_w) is set to 2048. For each model and dataset combination, we conduct 50 independent inference runs and report the mean and standard deviation of the performance metric.

***Scaling Law* Fitting Results.** Figure 2 illustrates the RAG model performance as a function of the generator model size. As shown in the figure, the RAG performance increases as the generator model size increases, with a steep increase at the beginning and then gradually saturating. Based on the experimental results, we fit the following power-law scaling function to the observed data:

$$\mathcal{P}(S_g) = C - BS_g^{-\alpha}, \tag{7}$$

where $\mathcal{P}(S_g)$ is the RAG performance, S_g is the generator model size, C, B and α are the fitted parameters. The exponent α governs the rate of performance increase. We fit the *scaling law* using the least squares method. The fitting parameters are summarized in Table 2, and the goodness of fit is indicated by the R-squared value. The fitting results of generator model scaling reveal several key insights as follows: ❶ The experiments validate the existence of a power-law relationship between the generator model size and RAG inference performance. ❷ The performance of RAG models exhibits substantial and rapid improvement in F1 score with initial increases in generator model size, highlighting the high sensitivity of RAG performance to generator size. ❸ The scaling curves demonstrate a clear saturation point, indicating diminishing returns from further increases in generator model size beyond a certain size. This suggests that simply scaling up the generator model indefinitely may not be the most efficient strategy for maximizing RAG performance.

Dataset	A	α	B	$\delta\eta$	R-squared
NQ	0.780	0.120	20.000	0.200	0.977
TriviaQA	0.749	0.149	14.999	0.301	0.956
WebQuestions	0.699	0.101	10.001	0.251	0.939
FEVER	0.649	0.079	25.000	0.350	0.968

Table 3: Fitted parameters of results for retrieval document number scaling.

Number of Retrieved Documents Scaling

We investigate the impact of the number of retrieved documents (N_d) on the inference performance of RAG models in a zero-shot setting.

Experimental Setup. We employ a zero-shot RAG approach. In particular, we fix the retriever to BERT-large, the generator to OPT-30B and set a fixed context window size (C_w) as 512. We vary the number of retrieved documents (N_d) from 1 to 100. And other parameters fixed.

***Scaling Law* Fitting Results.** As shown in Figure 3, the RAG performance increases with the number of retrieved documents initially, but eventually saturates. Based on the experimental results, we fit the following Sigmoid scaling function to the observed data:

$$\mathcal{P}(N_d) = \frac{A}{1 + e^{-\alpha(N_d - B)}} + \delta\eta, \tag{8}$$

where $\mathcal{P}(N_d)$ is the RAG performance, N_d is the number of retrieved documents, and A, α, B, and $\delta\eta$ are the fitted parameters. We fit the *scaling law* using the least squares method. The fitting parameters are summarized in Table 3, and the goodness of fit is indicated by the R-squared value.

The fitting results for a number of retrieved document number scaling reveal the following key insights as: ❶ RAG inference performance exhibits a clear saturation effect as the number of retrieved documents increases. ❷ There is a trade-off between leveraging external knowledge through more documents and maintaining computational efficiency in these RAG models.

Context Window Size Scaling

We investigate the impact of the context window size (C_w) on the inference performance of RAG models in a zero-shot setting. We conduct experiments by varying C_w while keeping the retriever model, generator model, and number of retrieved documents fixed.

Experimental Setup. We employ a zero-shot RAG approach. The retriever is fixed to a pre-trained BERT-large, and the generator is fixed to T5-11B.

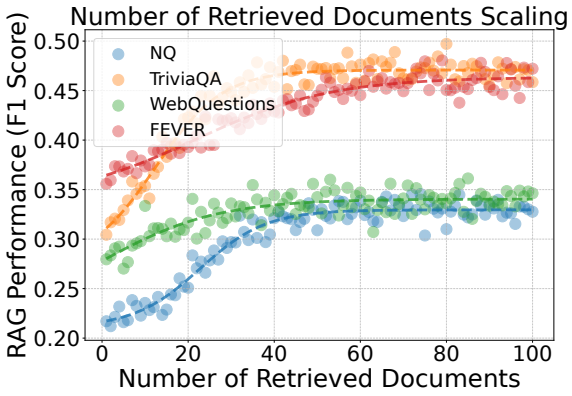

Figure 3: The RAG performance with a varying number of retrieved documents. The solid dots represent the data used for curve fitting, and the dashed lines represent the fitted curves, each corresponding to different datasets.

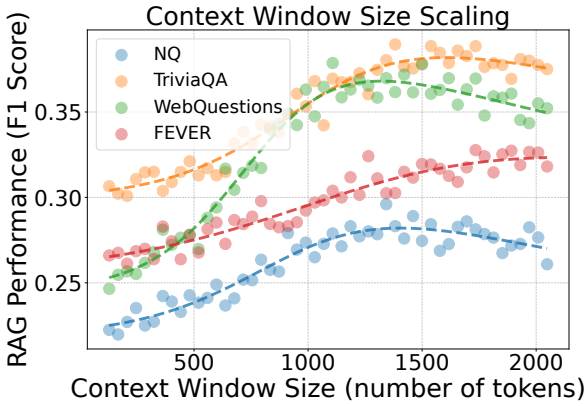

Figure 4: The RAG performance with varying context window size.

Dataset	A	α	B	β	$\delta\eta$	R-squared
NQ	0.102	0.003	950.967	0.00005	0.232	0.947
TriviaQA	0.186	0.004	1022.12	0.00006	0.296	0.960
WebQuestions	0.210	0.005	710.522	0.00007	0.232	0.935
FEVER	0.101	0.002	998.935	0.00004	0.257	0.951

Table 4: Fitted parameters of results for context window size scaling.

***Scaling Law* Fitting Results.** The number of retrieved documents (N_d) is set to 50. Figure 4 illustrates the RAG model performance as a function of the context window size. As depicted in the figure, the RAG performance generally increases with C_w initially, but the improvement diminishes and may eventually saturate or even decrease. Based on the experimental results, we fit the following scaling function to the observed data:

$$\mathcal{P}(C_w) = \frac{A}{1+e^{-\alpha(C_w-B)}}e^{-\beta C_w} + \delta\eta, \tag{9}$$

where $\mathcal{P}(C_w)$ is the RAG performance, C_w is the context window size, A, α, B, β and $\delta\eta$ are the fitted parameters. This function combines a sigmoid component, capturing the initial increase and saturation, with an exponential decay component ($e^{-\beta C_w}$), which accounts for the potential performance decrease at very large context window sizes due to noise or the generator's difficulty in handling excessively long contexts. B represents the inflection point of the sigmoid, and α controls the steepness of the sigmoid's rise. β controls the rate of the exponential decay. We fit the *scaling law* using the least squares method. The fitting parameters are summarized in Table 4, and the goodness of fit is indicated by the R-squared value. The fitting results for context window size scaling highlight the following key insights: ❶ The RAG inference performance exhibits a non-monotonic relationship with context window size, initially increasing but eventually saturating and potentially decreasing at very large sizes. ❷ There is a need to optimize the context window size, as excessively large windows do not guarantee better performance and can introduce noise or computational overhead. ❸ Increasing context window size involves a trade-off between leveraging more information and potentially incorporating irrelevant or noisy content.

Joint Scaling Laws

We investigate the joint influence of retriever model size (S_r), generator model size (S_g), number of retrieved documents (N_d), and context window size (C_w) on the RAG model performance.

Experimental Setup. We conduct experiments by varying the retriever and generator model sizes, number of retrieved documents, and context window size simultaneously. Due to the high dimensionality, we cannot visualize the full scaling behavior in a single figure. Therefore, we present the results through a fitted *scaling law*.

***Scaling Law* Fitting Results.** Based on the experimental results and our analysis, we propose the following joint *scaling law*. First, we define the function as follows:

$$\begin{aligned}&\mathcal{F}(S_r, S_g, N_d, C_w)\\ &= 1 + \exp\Bigg(-\Big[\alpha_r \log(S_r) + \alpha_g \log(S_g)\\ &+ \alpha_d \frac{1}{1+e^{-\beta_d(N_d-B_d)}}\\ &+ \alpha_w \frac{e^{-\beta C_w}}{1+e^{-\beta_w(C_w-B_w)}} - B\Big]\Bigg).\end{aligned} \tag{10}$$

Using this function, we define the joint *scaling law* as:

$$\mathcal{P}(S_r, S_g, N_d, C_w) = \frac{A}{\mathcal{F}(S_r, S_g, N_d, C_w)} + \delta\eta, \tag{11}$$

where $\mathcal{P}(S_r, S_g, N_d, C_w)$ is the RAG performance, S_r is the retriever model size, S_g is the generator model size, N_d is the number of retrieved documents, C_w is the context window size and the rest of the parameters are the fitted parameters. We fit the *scaling law* using the least squares method. The fitting parameters are summarized in Table 5, and the goodness of fit is indicated by the R-squared value. The fitted *scaling law* provides a way to approximate the RAG model inference performance considering the joint impact of retriever and

Dataset	A	α_r	α_g	α_d	β_d	B_d	α_w	β_w	B_w	B	$\delta\eta$	R-squared
NQ	0.750	0.110	0.101	0.187	0.101	20.002	0.198	0.001	999.999	9.999	0.201	0.975
TriviaQA	0.700	0.151	0.121	0.151	0.199	15.001	0.199	0.002	1199.998	15.001	0.301	0.961
WebQuestions	0.649	0.119	0.081	0.100	0.148	10.000	0.102	0.003	799.999	8.000	0.250	0.938
FEVER	0.599	0.081	0.100	0.101	0.100	24.999	0.098	0.0009	1500.001	11.999	0.349	0.952

Table 5: Fitted parameters for joint *scaling law*.

generator model sizes, the number of retrieved documents, and the context window size. However, the inherent complexity of RAG models and the various combinations of these factors may make the fitting results not always reliable.

Theoretical Analysis. In this part, we provide a theoretical analysis explaining how the joint *scaling law* preserves the marginal *scaling law* effects for each dimension, including the retriever model size (S_r), the generator model size (S_g), the number of retrieved documents (N_d), and the context window size (C_w). The following Theorem 1 demonstrates that the joint *scaling law* is a generalization of the marginal scaling laws and provides a way to predict the RAG model performance under various configurations.

Theorem 1. *When other factors are held constant, our proposed joint model $\mathcal{P}(S_r, S_g, N_d, C_w)$ can approximate the functional form of the corresponding marginal scaling laws. Specifically, it reproduces the power-law saturation observed for model sizes S_r, S_g and the sigmoid-type behavior observed for document count N_d and context window size C_w.*

Theoretical Justification for Theorem 1. We now provide a theoretical justification for our joint model by showing how it approximates the marginal scaling behaviors. Let's analyze the case for the retriever size, S_r, while keeping S_g, N_d, and C_w fixed. The function $\mathcal{F}(S_r, S_g, N_d, C_w)$ from Eq. 10 simplifies to:

$$\mathcal{F}(S_r) = 1 + \exp\left(-\left[\alpha_r \log(S_r) - \tilde{B}\right]\right) = 1 + e^{\tilde{B}} S_r^{-\alpha_r},$$

where $\tilde{B}$ is a constant that absorbs all terms related to the fixed factors. Let's define a new constant $C = e^{\tilde{B}}$. The joint model then becomes:

$$\mathcal{P}(S_r) = \frac{A}{1 + C S_r^{-\alpha_r}} + \delta\eta.$$

Here, we identify the key approximation step. For large values of S_r (which is the region of interest for scaling laws), the term $CS_r^{-\alpha_r}$ becomes small ($\ll 1$). We can therefore apply the first-order Taylor expansion $1/(1+x) \approx 1 - x$ for small x. This yields:

$$\mathcal{P}(S_r) \approx A(1 - CS_r^{-\alpha_r}) + \delta\eta = A - ACS_r^{-\alpha_r} + \delta\eta.$$

This approximated form now clearly exhibits a power-law relationship with S_r, consistent with the empirical marginal law for retriever scaling shown in Eq. 6. It is important to note that while the functional form is recovered, the parameters are not identical. For instance, the effective scaling coefficient in this approximation is a product $A \cdot C$, where C encapsulates the influence of all other factors. This demonstrates a key strength of our joint model: it explicitly accounts for how the scaling behavior of one component is modulated by the others. A similar analysis can be applied to show that the joint model approximates a power-law for S_g and sigmoid-type behaviors for N_d and C_w, thus justifying its formulation.

Scaling law Application in Computational Resource Allocation

In this section, we demonstrate how the derived inference *scaling laws* can guide the allocation of computational resources in RAG models during inference. Efficient resource allocation between the retriever and the generator is critical for maximizing performance under specific latency constraints in a zero-shot setting.

Experimental Setup

The computational resource allocation can be set as an optimization problem:

$$\max_{R_r, R_g} \mathcal{P}(S_r, S_g, N_d, C_w, R_r, R_g), \tag{12}$$

$$\text{subject to } R_r + R_g = R, \tag{13}$$

Here, R_r and R_g represent the computational resources allocated to the retriever and generator. We use inference latency as a proxy for computational resources. Let T_r and T_g be the inference latencies of the retriever and generator, respectively, and T be the total latency budget. We need to consider the following constraints as well:

$$T_r(S_r, N_d, R_r) + T_g(S_g, N_d, C_w, R_g) \leq T. \tag{14}$$

To investigate the impact of resource allocation, we conduct experiments where we vary the ratio $\frac{R_r}{R}$ while keeping the total budget T fixed. We leverage the *scaling law* relationships established in previous sections to estimate the performance of the retriever and generator under different resource allocations. We use the following latency models, based on our empirical observations and simplifying assumptions:

$$T_r(S_r, N_d, R_r) = \kappa_r S_r N_d / R_r, \tag{15}$$

$$T_g(S_g, N_d, C_w, R_g) = \kappa_g S_g N_d C_w / R_g, \tag{16}$$

where κ_r and κ_g are constants of proportionality for the retriever and generator, respectively. Here, these models assume a linear relationship between latency and model size, the number of retrieved documents, and context window size for the generator, in comparison with an inverse relationship with allocated resources.

We use the following configurations for our experiments, chosen to represent a range of scenarios. Retriever models are GTR-base (110M), GTR-Large (335M), GTR-XL

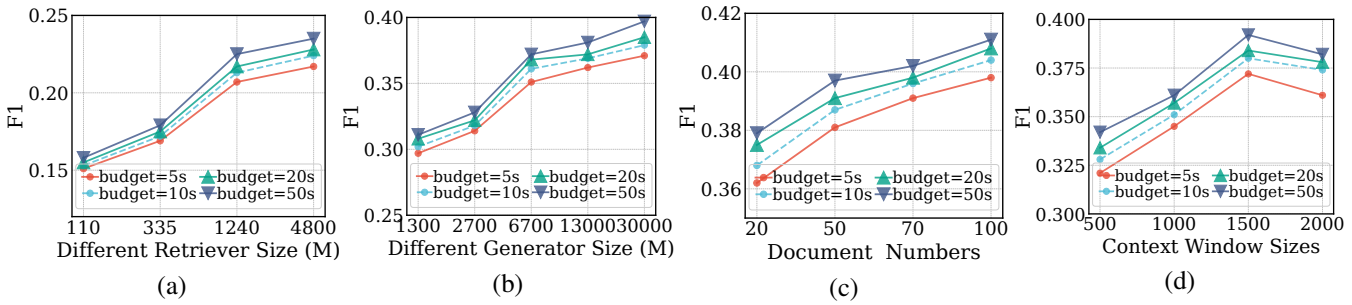

Figure 5: Different budgets in (a) retriever sizes, (b) generator sizes, (c) number of retrieved documents and (d) context window sizes on the TriviaQA dataset.

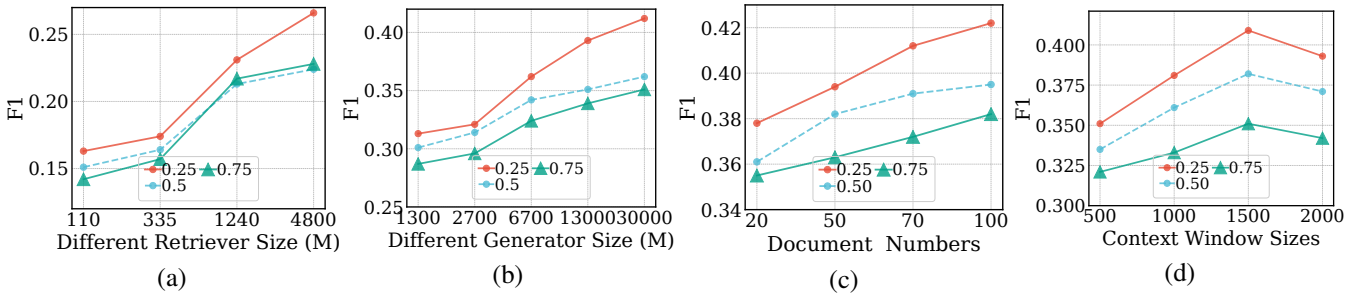

Figure 6: Different resource allocation radio $\frac{R_r}{R}$ in (a) retriever sizes, (b) generator sizes, (c) number of retrieved documents and (d) context window sizes on the TriviaQA dataset. *The larger the resource allocation ratio ↑, the higher the retriever resource allocation ↑ and the lower the generator resource allocation ↓.*

(1.24B), GTR-XXL (4.8B). Generator models are OPT-1.3B(1300M), OPT-2.7B (2700M), OPT-6.7B (6700M), OPT-13B (13000M) and OPT-30B (30000M). Number of retrieved documents are 20, 50, 70 and 100. Context window sizes (C_w) are 500, 1000, 1500 and 2000. Total latency budget (T) are 5s, 10s, 20s and 50s.

Results

Figure 5 and Figure 6 (a) (b) (c) (d) explore the impact of retriever model size, generator model size, retrieval document number and context window size on budget and resource allocation, respectively. We have the following observations:

- **Model Size:**
  - **Generator over Retriever:** With a limited budget, scaling the generator yields more significant and consistent performance improvements than scaling the retriever. Therefore, scarce resources should be prioritized for the generator.
  - **Mid-sized Retriever is Optimal:** The performance gain from increasing retriever size diminishes, especially under tight budget constraints. An excessively large retriever can become a bottleneck, whereas a mid-sized one offers a better balance between accuracy and computational efficiency.
- **Documents and Context:**
  - **Number of Retrieved Documents:** Increasing the number of retrieved documents shows diminishing returns. A powerful retriever can effectively utilize a smaller set of documents and maintain robust performance.
  - **Context Window Size:** Expanding the context window is initially beneficial, but an excessively large window can introduce noise and degrade performance. A strong generator is better equipped to handle longer contexts, mitigating these negative effects.

Conclusion

In this work, we conduct a comprehensive investigation of *inference scaling law* for RAG models and establish empirical *inference scaling law* to key factors including retriever and generator model scales, number of retrieved documents, and context window size. From the experimental results, we reveal power-law and sigmoid-type behaviors. This study lays the foundation for future research into more complex and dynamic scaling behaviors, ultimately paving the way for more powerful and efficient knowledge-intensive AI systems.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant No. 72574098, 72074108, 72504122) and Fundamental Research Funds for the Central Universities at Nanjing University (Grant No. 010814370338), Jiangsu Young Talents in Social Sciences and Tang Scholar of Nanjing University.

References

Berant, J.; Chou, A.; Frostig, R.; and Liang, P. 2013. Semantic parsing on freebase from question-answer pairs. 1533–1544.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. d. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2020. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Elsken, T.; Metzen, J. H.; and Hutter, F. 2019. Neural architecture search: A survey. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1997–2017.

Fan, W.; Ding, Y.; Ning, L.; Wang, S.; Li, H.; Yin, D.; Chua, T.-S.; and Li, Q. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6491–6501.

Gao, L.; Biderman, S.; Black, S.; Golding, L.; Hoppe, T.; Foster, C.; Phang, J.; He, H.; Thite, A.; et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, 3929–3938. PMLR.

Hoffmann, J.; Borgeaud, S.; Mensch, A.; Buchatskaya, E.; Cai, T.; Rutherford, E.; De Las Casas, D.; Hendricks, L. A.; Welbl, J.; Clark, A.; et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Izacard, G.; Caron, M.; Hosseini, L.; Riedel, S.; Bojanowski, P.; Joulin, A.; and Grave, E. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *arXiv preprint arXiv:2112.09118*.

Jain, P.; Phanishayee, A.; Mars, J.; Abbeel, P.; and Gonzalez, J. E. 2019. Checkmate: Breaking the memory wall with optimal tensor rematerialization. *arXiv preprint arXiv:1910.02653*.

Ji, J.; Li, Y.; Liu, H.; Du, Z.; Wei, Z.; Qi, Q.; Shen, W.; and Lin, Y. 2024. SRAP-Agent: Simulating and Optimizing Scarce Resource Allocation Policy with LLM-based Agent. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 267–293. Miami, Florida, USA: Association for Computational Linguistics.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611.

Kaplan, J.; McCandlish, S.; Henighan, T.; B. Brown, T.; Chess, B.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769–6781.

Khattab, O.; and Zaharia, M. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 39–48.

Komeili, S.; Lin, B. Y.; and Ren, X. 2023. Faithful Reasoning with Large Language Models via Knowledge Retrieval. *arXiv preprint arXiv:2305.15395*.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; and Chen, Z. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Lin, K.; Zhao, W.; Ji, Z.; Yang, Z.; He, Y.; Zhou, L.; Meng, Y.; and Wei, J. 2023. Text generation with text-retrieval mixing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4276–4286.

Luan, Y.; Dai, Z.; Callan, J.; and Lin, J. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9: 346–361.

Ni, J.; Qu, C.; Lu, J.; Dai, Z.; Hernandez Abrego, G.; Ma, J.; Zhao, V.; Luan, Y.; Hall, K.; Chang, M.-W.; and Yang, Y. 2022. Large Dual Encoders Are Generalizable Retrievers. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9844–9855. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

NVIDIA. 2022. Efficient large-scale language model training on GPU clusters using Megatron-LM.

https://developer.nvidia.com/blog/efficient-large-scale-language-model-training-on-gpu-clusters-using-megatron-lm/.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. 2021a. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; Aslanides, J.; Henderson, S.; Ring, R.; Young, S.; et al. 2021b. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *arXiv preprint arXiv:2112.11446*.

Shi, G.; Zhou, S.; Wang, Y.; Shi, C.; and Liu, L. 2023. Generating patent text abstracts based on improved multi-head attention mechanism. *Data Anal. Knowl. Discovery*, 7(6): 61–72.

Tay, Y.; Wei, J.; Chung, H.; Tran, V.; So, D.; Shakeri, S.; Garcia, X.; Zheng, S.; Rao, J.; Chowdhery, A.; Zhou, D.; Metzler, D.; Petrov, S.; Houlsby, N.; Le, Q.; and Dehghani, M. 2023. Transcending Scaling Laws with 0.1% Extra Compute. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 1471–1486. Singapore: Association for Computational Linguistics.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Urbizu, G.; San Vicente, I.; Saralegi, X.; Agerri, R.; and Soroa, A. 2023. Scaling Laws for BERT in Low-Resource Settings. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 7771–7789. Toronto, Canada: Association for Computational Linguistics.

Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; and Rellermeyer, J. S. 2020. A survey on distributed machine learning. *ACM Computing Surveys (CSUR)*, 53(2): 1–33.

Wang, S.; Chen, Z.; Li, B.; He, K.; Zhang, M.; and Wang, J. 2024. Scaling Laws Across Model Architectures: A Comparative Analysis of Dense and MoE Models in Large Language Models. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 5583–5595. Miami, Florida, USA: Association for Computational Linguistics.

Xiong, Y.; Chen, X.; Ye, X.; Chen, H.; Lin, Z.; Lian, H.; Su, Z.; Niu, J.; and Ding, G. 2024. Temporal Scaling Law for Large Language Models. arXiv:2404.17785.

Yang, D.; Rao, J.; Chen, K.; Guo, X.; Zhang, Y.; Yang, J.; and Zhang, Y. 2024. Im-rag: Multi-round retrieval-augmented generation through learning inner monologues. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 730–740.

Yao, Y.; Jin, H.; Shah, A. D.; Han, S.; Hu, Z.; Ran, Y.; Stripelis, D.; Xu, Z.; Avestimehr, S.; and He, C. 2024. ScaleLLM: A Resource-Frugal LLM Serving Framework by Optimizing End-to-End Efficiency. *arXiv preprint arXiv:2408.00008*.

Zamani, H.; and Bendersky, M. 2024. Stochastic rag: End-to-end retrieval-augmented generation through expected utility maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2641–2646.

Zhang, L.; Zhou, H.; Zhao, S.; Hao, J.; Cao, Z.; and Yang, Y. 2022. MoComp: Multi-Agent Game Abstraction for General Multi-Agent Learning. *arXiv preprint arXiv:2203.14040*.

Zhou, S.; Wang, X.; Qiu, J.; Bu, W.; and Wang, H. 2025a. OracleNet: enhancing Oracle Bone Script recognition with Adaptive Deformation and Texture-Structure Decoupling. *npj Heritage Science*, 13(1): 273.

Zhou, S.; Wang, X.; Qiu, J.; Li, X.; Shi, B.; and Wang, H. 2025b. LOSDF: A logical optimization and semantic decoupling framework for question answering in multi-party conversations. *Information Processing & Management*, 62(5): 104200.

Zhou, S.; Wang, X.; Zhou, Z.; Yi, H.; Zheng, X.; and Wang, H. 2024. The master-slave encoder model for improving patent text summarization: A new approach to combining specifications and claims. In *International Conference on Neural Information Processing*, 254–269. Springer.

Zhou, S.; Xuan, Y.; Ao, Y.; Wang, X.; Fan, T.; and Wang, H. 2025c. MERIT: Multi-Agent Collaboration for Unsupervised Time Series Representation Learning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 24011–24028.

Zhou, S.; Zhao, R.; Zhou, Z.; Yi, H.; Zheng, X.; and Wang, H. 2025d. Enhancing Extractive Question Answering in Multiparty Dialogues with Logical Inference Memory Network. In *Proceedings of the 31st International Conference on Computational Linguistics*, 8725–8738.

Zhuocheng, Z.; Gu, S.; Zhang, M.; and Feng, Y. 2023. Scaling Law for Document Neural Machine Translation. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 8290–8303. Singapore: Association for Computational Linguistics.