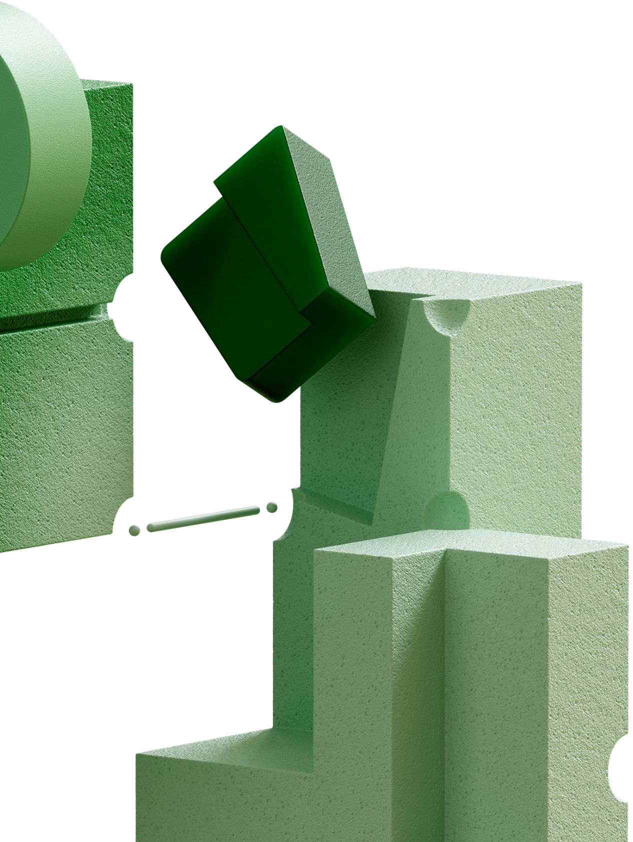


Inference Scaling for Long-Context Retrieval Augmented Generation

Zhenrui Yue*[1,2], Honglei Zhuang*[1], Aijun Bai[1], Kai Hui[1], Rolf Jagerman[1], Hansi Zeng[1,3], Zhen Qin[1], Dong Wang[2], Xuanhui Wang[1], Michael Bendersky[1]

[1]Google DeepMind
[2]University of Illinois Urbana-Champaign
[3]University of Massachusetts Amherst
*Equal contribution

https://arxiv.org/abs/2410.04343

Introduction

1

Introduction

Inference Scaling

A series of recent studies show that increasing the amount of inference-time computation can be similar (Agarwal et al., 2024), if not more effective (Snell et al., 2024) than allocating those computation to training in some scenarios.

Examples include:

- Scaling the number of examples in ICL
- Scaling best-of-N samples along with sequential revisions
- Scaling reasoning iterations (e.g., OpenAI's o1 model)
- ...

C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. arXiv preprint arXiv:2408.03314, 2024.
R. Agarwal, A. Singh, L. M. Zhang, B. Bohnet, L. Rosias, S. C. Chan, B. Zhang, A. Faust, and H. Larochelle. Many-shot in-context learning. In ICML 2024 Workshop on In-Context Learning.
A. Bertsch, M. Ivgi, U. Alon, J. Berant, M. R. Gormley, and G. Neubig. In-context learning with long-context models: An in-depth exploration. arXiv preprint arXiv:2405.00200, 2024.
OpenAI. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/

Introduction

Inference Scaling for RAG

With the advances in long-context LLMs, recent studies also attempt to better leverage the full context-length in retrieval-augmented generation (RAG) tasks.

Studies on inference scaling for RAG mostly focus on scaling the **number of documents** (Ram et al., 2023; Shao et al., 2024; Lee et al., 2024; Xu et al., 2024;) or the **length of documents** (Jiang et al., 2024).

O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. Incontext retrieval-augmented language models. Transactions of the Association for Computational Linguistics, 11:1316–1331, 2023.
R. Shao, J. He, A. Asai, W. Shi, T. Dettmers, S. Min, L. Zettlemoyer, and P. W. Koh. Scaling retrieval based language models with a trillion-token datastore. arXiv preprint arXiv:2407.12854, 2024.
P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, and B. Catanzaro. Retrieval meets long context large language models. In ICLR, 2024.
J. Lee, A. Chen, Z. Dai, D. Dua, D. S. Sachan, M. Boratko, Y. Luan, S. M. Arnold, V. Perot, S. Dalmia, et al. Can long-context language models subsume retrieval, RAG, SQL, and more? arXiv preprint arXiv:2406.13121, 2024a.
Z. Jiang, X. Ma, and W. Chen. LongRAG: Enhancing retrieval-augmented generation with long-context LLMs. arXiv preprint arXiv:2406.15319, 2024.

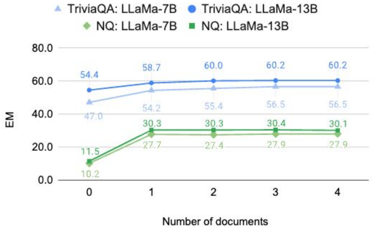
Introduction

Inference Scaling for RAG

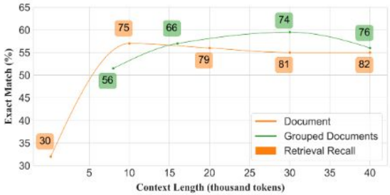
With the advances in long-context LLMs, recent studies also attempt to better leverage the full context-length in retrieval-augmented generation (RAG) tasks.

Studies on inference scaling for RAG mostly focus on scaling the **number of documents** (Ram et al., 2023; Shao et al., 2024; Lee et al., 2024; Xu et al., 2024;) or the **length of documents** (Jiang et al., 2024).

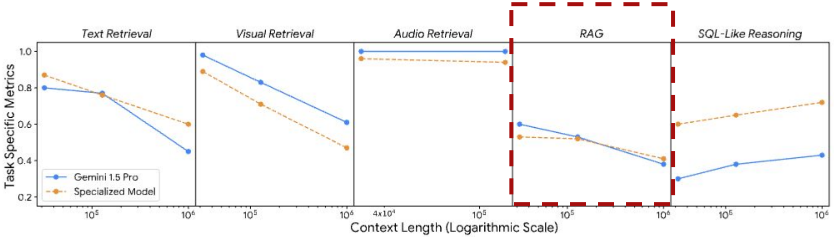
… And RAG performance does not always increase as the retrieved context increases!



(Ram et al. 2023)



(Jiang et al. 2024)



(Lee et al. 2024)

O. Ram, Y. Levine, I. Dalmedigos, D. Muhlgay, A. Shashua, K. Leyton-Brown, and Y. Shoham. Incontext retrieval-augmented language models. Transactions of the Association for Computational Linguistics, 11:1316–1331, 2023.
R. Shao, J. He, A. Asai, W. Shi, T. Dettmers, S. Min, L. Zettlemoyer, and P. W. Koh. Scaling retrieval based language models with a trillion-token datastore. arXiv preprint arXiv:2407.12854, 2024.
P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, and B. Catanzaro. Retrieval meets long context large language models. In ICLR, 2024.
J. Lee, A. Chen, Z. Dai, D. Dua, D. S. Sachan, M. Boratko, Y. Luan, S. M. Arnold, V. Perot, S. Dalmia, et al. Can long-context language models subsume retrieval, RAG, SQL, and more? arXiv preprint arXiv:2406.13121, 2024a.
Z. Jiang, X. Ma, and W. Chen. LongRAG: Enhancing retrieval-augmented generation with long-context LLMs. arXiv preprint arXiv:2406.15319, 2024.

Introduction

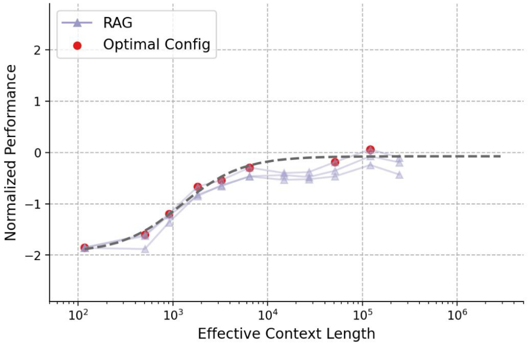
More Comprehensive Inference Scaling for RAG

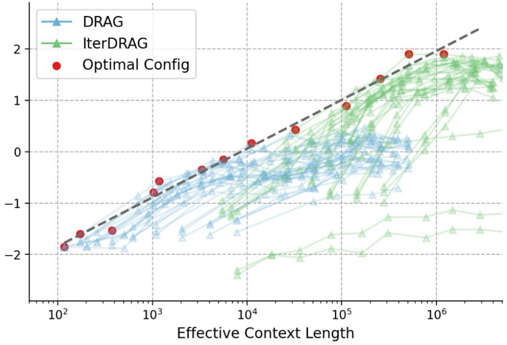
In this work, we conduct a more comprehensive study on inference scaling for RAG.

We study two more strategies to leverage inference computation in RAG tasks:

1. **Demonstration-Based RAG (DRAG):** Adding RAG demonstrations as in-context examples.
2. **Iterative Demonstration-Based RAG (IterDRAG):** Iteratively apply demonstration-based RAG to solve more challenging, multi-hop queries.

And we show that when optimally configured, these strategies enable RAG performance to increase (almost) linearly with the order of magnitude of the amount of inference computation.





O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In Findings of EMNLP, 2023.
H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In ACL, 2023.

Inference Scaling Strategies

2

Inference Scaling Strategies

Definition of “Inference Computation”

In our study, we measure the **inference computation** by “**effective context length**”:

- For *single-round* strategies, this is equivalent to the input context length to the LLM
- For *multi-round* strategies, this is the sum of the input context lengths for every rounds of LLM calls

This aligns well with the pricing model of many commercial LLMs, as the output token number for our tasks is often limited.

We consider a fixed-budget setting, where users are given a fixed budget of **effective context length** L_{max}

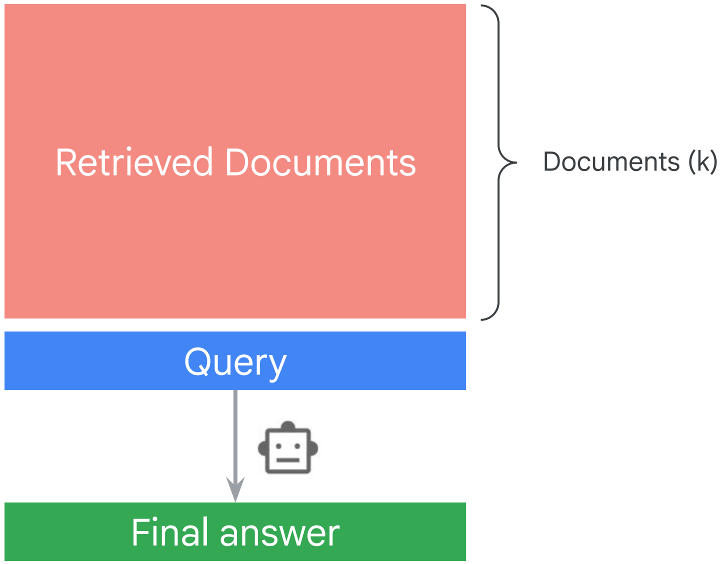
Inference Scaling Strategies

Vanilla RAG

In the vanilla RAG strategy, a retriever will retrieve *k* documents based on the query. The retrieved documents and the query are provided to the LLM as input, and the LLM outputs the answer.

To fully leverage the context window of LLMs, we can adjust the following parameter:

- Number of documents *k*



Inference Scaling Strategies

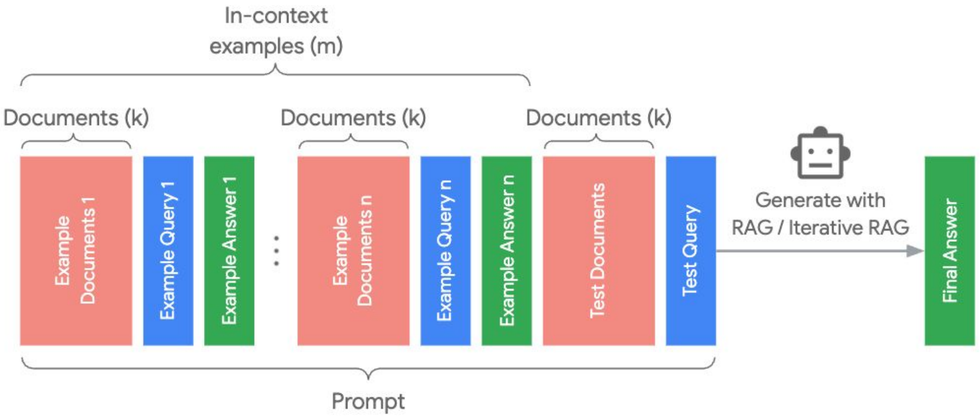
Demonstration-Based RAG (DRAG)

Adding demonstrations as in-context examples, where each demonstration include a complete RAG call: retrieved documents, query and answer.

Ideally, demonstrations allow models to learn how to locate the most relevant information and follow the formatting convention of answers.

This strategy’s effective context length can be controlled by 2 parameters

1. Number of documents *k*
2. Number of in-context examples *m*



O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In Findings of EMNLP, 2023.
H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In ACL, 2023.

Inference Scaling Strategies

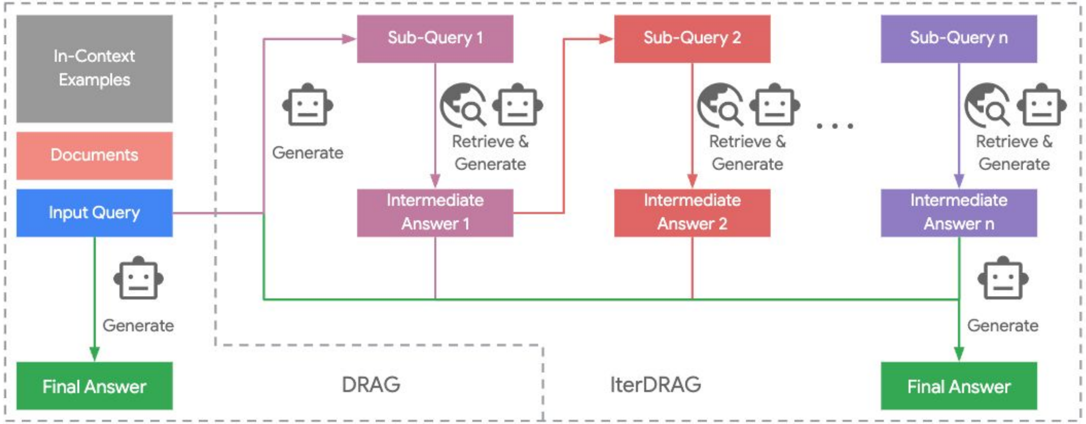
Iterative Demonstration-Based RAG (IterDRAG)

Based on the DRAG strategy, we can further allow the model to iteratively issue sub-queries based on tha answer from previous rounds.

Ideally, the iterative process allow models tackle queries requiring multi-hop reasoning.

This strategy's effective context length can be controlled by 3 parameters

1. Number of documents *k*
2. Number of in-context examples *m*
3. Number of iterations *n*



O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis. Measuring and narrowing the compositionality gap in language models. In Findings of EMNLP, 2023.
H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In ACL, 2023.

RAG Performance and Inference Computation Scale

3

RAG Performance and Inference Computation Scale

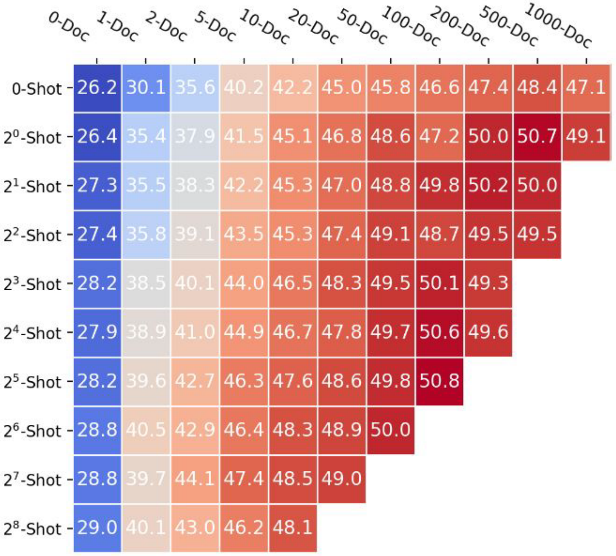
Fixed Budget Optimal Performance

Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

F1 Performance

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

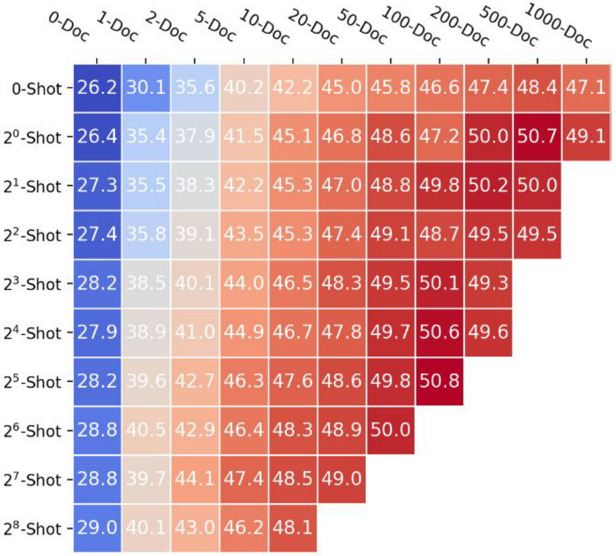
Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Assuming we can always find the optimal configuration, how will the performance scale with the budget?



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

F1 Performance

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

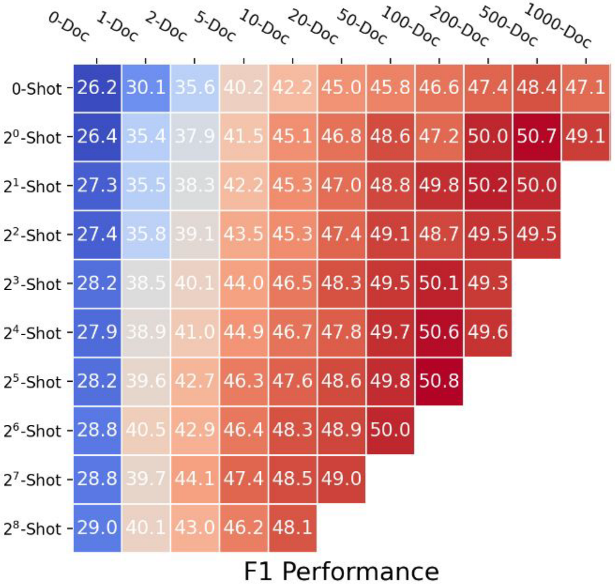
Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as $P^*(L_{max})$



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

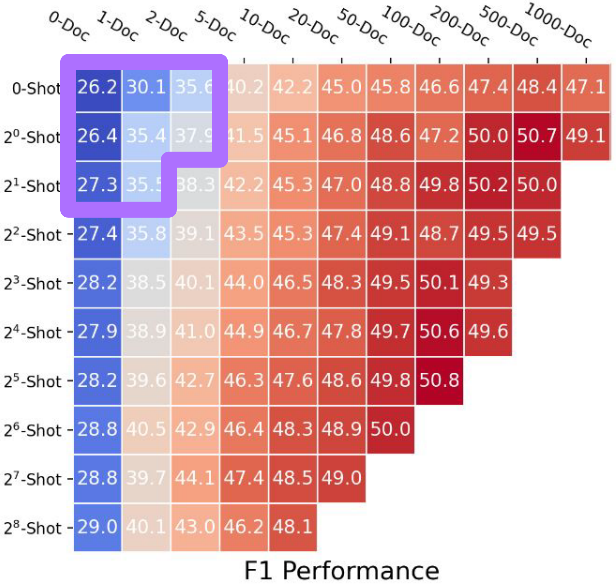
For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as $P^*(L_{max})$

Configurations within 8k token budget



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

Given a fixed budget *L*$_{max}$ and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

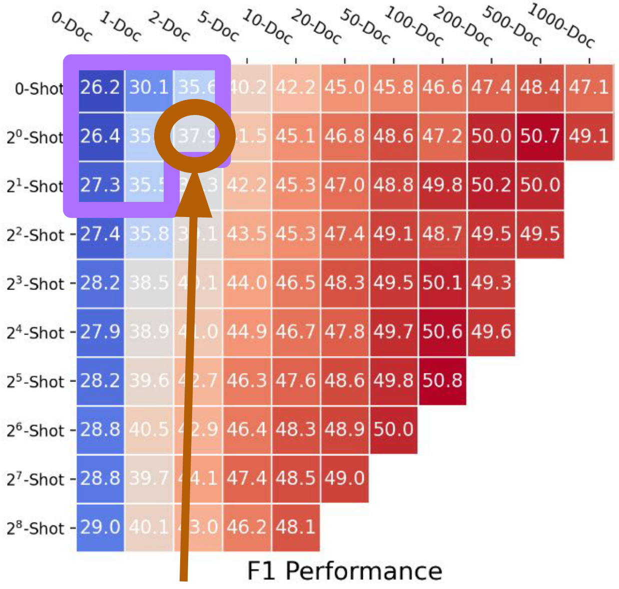
For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- …

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as *P*(L*$_{max}$ *)*

Configurations within 8k token budget



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.[illegible]	37.9	[illegible]1.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	3[illegible].3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	3[illegible].1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	4[illegible].1	44.0	46.5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

Optimal performance within 8k

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

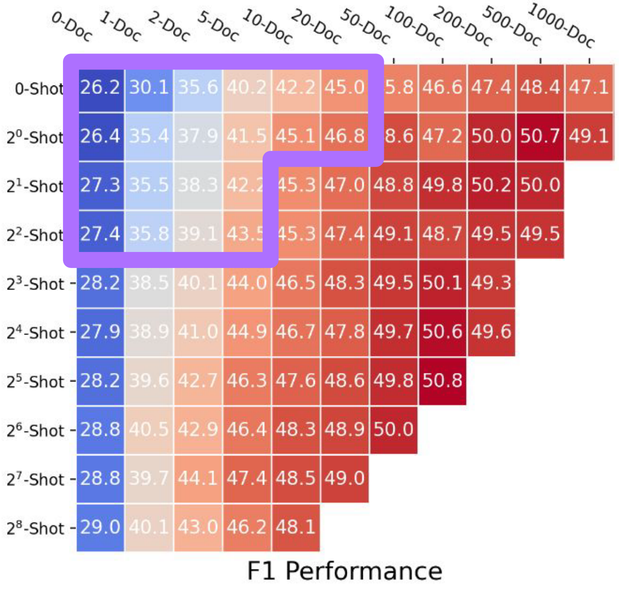
For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as $P^*(L_{max})$

Configurations within 16k token budget



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	5.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	8.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

F1 Performance

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

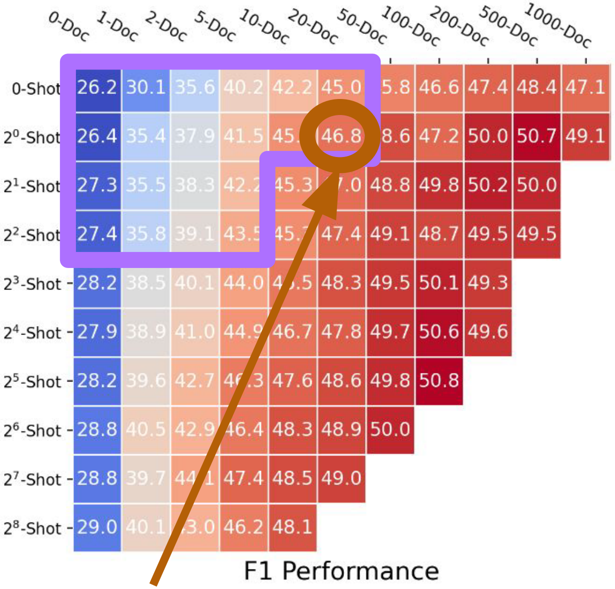
For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as $P^*(L_{max})$

Configurations within 16k token budget



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	5.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.[illegible]	46.8	8.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	[illegible]7.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.7	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	4[illegible].5	48.3	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.8	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

F1 Performance

Optimal performance within 16k

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

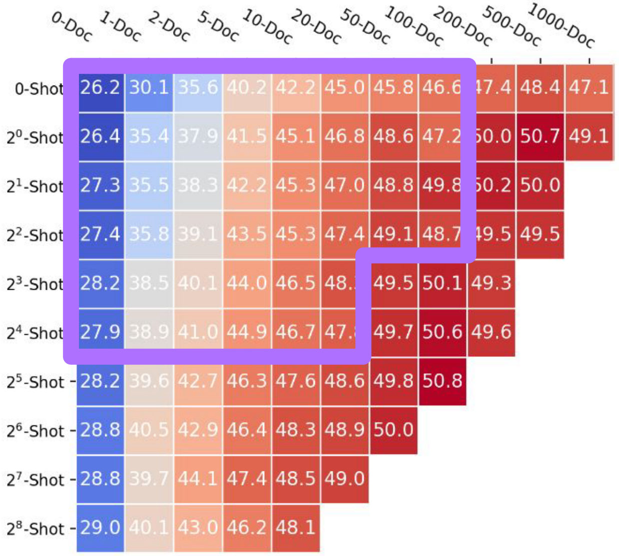
For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as $P^*(L_{max})$

Configurations within 32k token budget



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	50.2	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	48.7	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	48.	49.5	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	47.	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	46.4	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

F1 Performance

RAG Performance and Inference Computation Scale

Fixed Budget Optimal Performance

Given a fixed budget L_{max} and a strategy (DRAG, IterDRAG etc.), there can be different configurations satisfying the budget

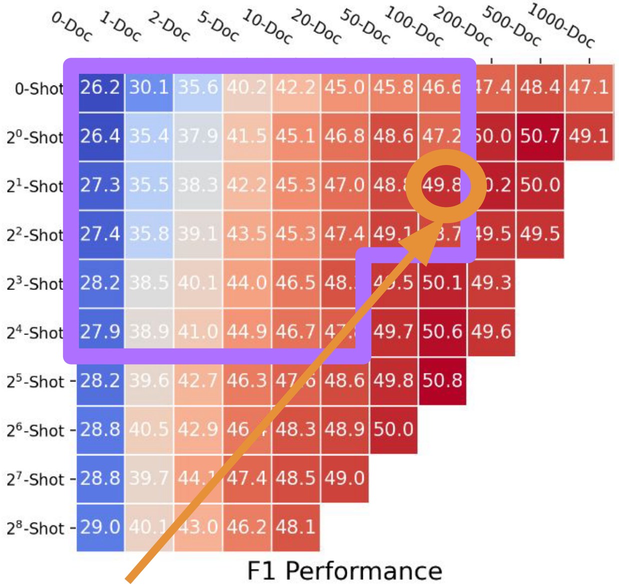
For example, if the budget is 8k tokens, and the strategy is DRAG, the following configurations can all satisfy the budget:

- Number of documents k=20, Number of demos m=1
- Number of documents k=10, Number of demos m=2
- Number of documents k=5, Number of demos m=4
- ...

We enumerate a set of configuration combination for each strategy.

Among all these configurations, we can find the optimal RAG performance, denoted as $P*(L_{max})$

Configurations within 32k token budget



	0-Doc	1-Doc	2-Doc	5-Doc	10-Doc	20-Doc	50-Doc	100-Doc	200-Doc	500-Doc	1000-Doc
0-Shot	26.2	30.1	35.6	40.2	42.2	45.0	45.8	46.6	47.4	48.4	47.1
2^0-Shot	26.4	35.4	37.9	41.5	45.1	46.8	48.6	47.2	50.0	50.7	49.1
2^1-Shot	27.3	35.5	38.3	42.2	45.3	47.0	48.8	49.8	[illegible]	50.0	
2^2-Shot	27.4	35.8	39.1	43.5	45.3	47.4	49.1	[illegible]	49.5	49.5	
2^3-Shot	28.2	38.5	40.1	44.0	46.5	[illegible]	[illegible]	50.1	49.3		
2^4-Shot	27.9	38.9	41.0	44.9	46.7	[illegible]	49.7	50.6	49.6		
2^5-Shot	28.2	39.6	42.7	46.3	47.6	48.6	49.8	50.8			
2^6-Shot	28.8	40.5	42.9	[illegible]	48.3	48.9	50.0				
2^7-Shot	28.8	39.7	44.1	47.4	48.5	49.0					
2^8-Shot	29.0	40.1	43.0	46.2	48.1						

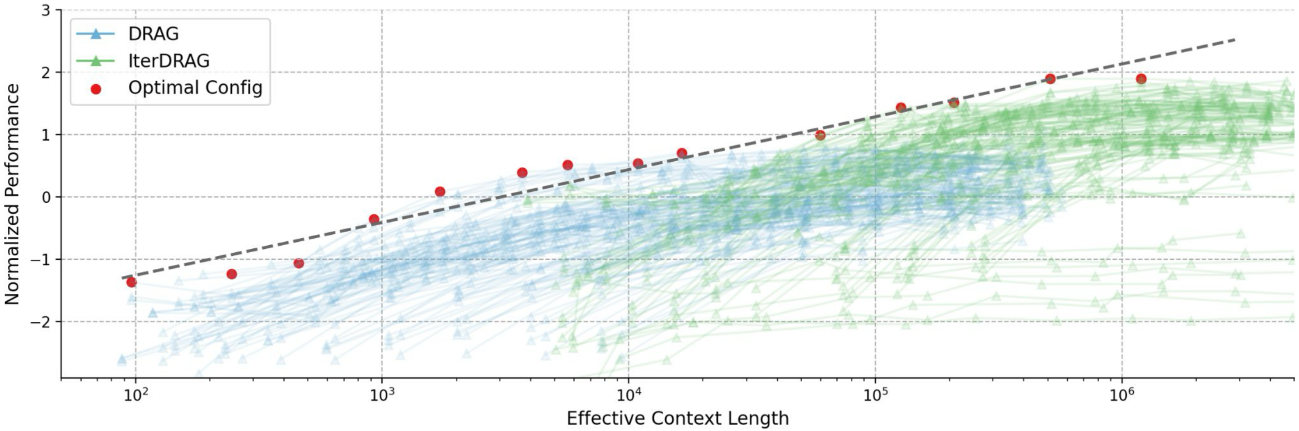
F1 Performance

Optimal performance within 32k

RAG Performance and Inference Computation Scale

RAG Performance vs. Inference Computation Scale

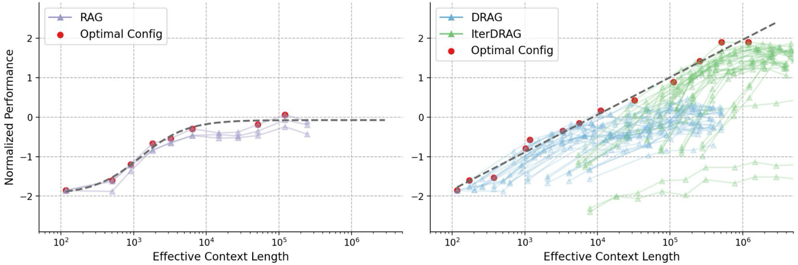
Plotting *P*(L$_{max}$)* with *L$_{max}$* for both strategies on all datasets with three metrics (EM, F1, Accuracy) and normalize them.



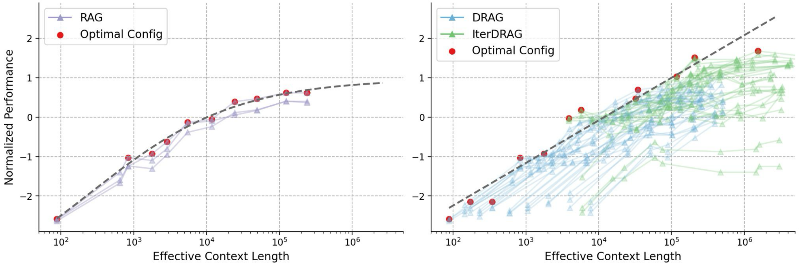
RAG Performance and Inference Computation Scale

RAG Performance vs. Inference Computation Scale

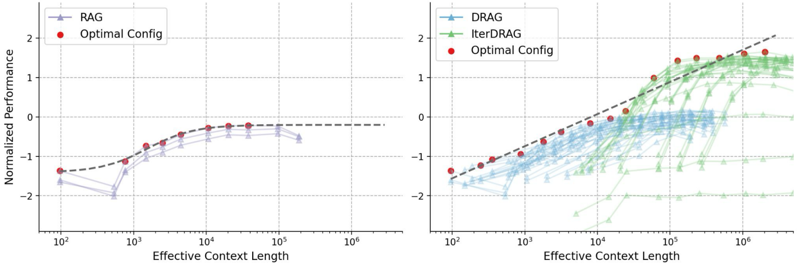
Comparing to only scaling the number of document in vanilla RAG on different datasets.



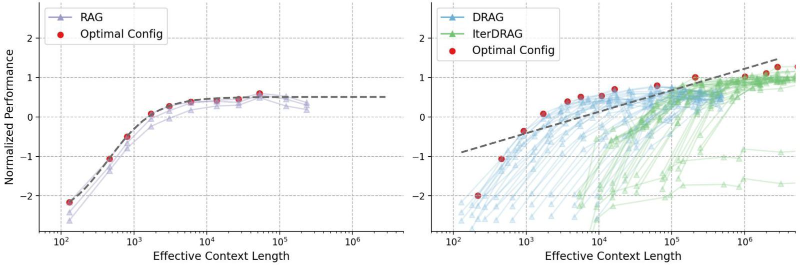
MuSiQue



Bamboogle



2WikiMultiHopQA



HotpotQA

RAG Performance and Inference Computation Scale

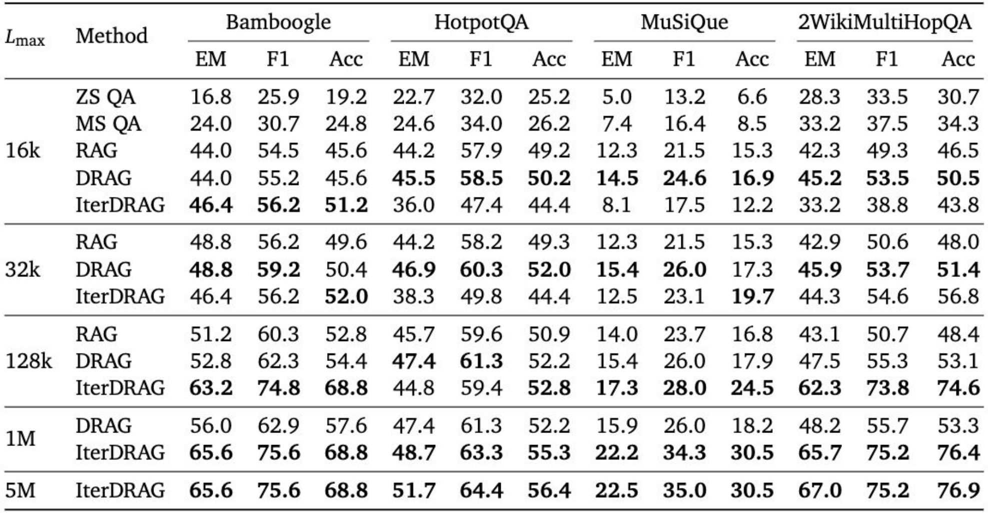
Comparing Strategies

Evaluated on 4 multi-hop open-book question answering datasets.

Baselines:

- **Zero-Shot QA (ZS QA)**: 0 retrieved document, 0 demonstration
- **Many-Shot QA (MS QA)**: 0 retrieved document, many demonstrations
- **RAG**: many retrieved documents, 0 demonstration

Optimal performance of different methods with varying maximum effective context lengths.



$L_{\max}$	Method	Bamboogle			HotpotQA			MuSiQue			2WikiMultiHopQA		
		EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
16k	ZS QA	16.8	25.9	19.2	22.7	32.0	25.2	5.0	13.2	6.6	28.3	33.5	30.7
	MS QA	24.0	30.7	24.8	24.6	34.0	26.2	7.4	16.4	8.5	33.2	37.5	34.3
	RAG	44.0	54.5	45.6	44.2	57.9	49.2	12.3	21.5	15.3	42.3	49.3	46.5
	DRAG	44.0	55.2	45.6	**45.5**	**58.5**	**50.2**	**14.5**	**24.6**	**16.9**	**45.2**	**53.5**	**50.5**
	IterDRAG	**46.4**	**56.2**	**51.2**	36.0	47.4	44.4	8.1	17.5	12.2	33.2	38.8	43.8
32k	RAG	48.8	56.2	49.6	44.2	58.2	49.3	12.3	21.5	15.3	42.9	50.6	48.0
	DRAG	**48.8**	**59.2**	50.4	**46.9**	**60.3**	**52.0**	**15.4**	**26.0**	17.3	**45.9**	**53.7**	**51.4**
	IterDRAG	46.4	56.2	**52.0**	38.3	49.8	44.4	12.5	23.1	**19.7**	44.3	54.6	56.8
128k	RAG	51.2	60.3	52.8	45.7	59.6	50.9	14.0	23.7	16.8	43.1	50.7	48.4
	DRAG	52.8	62.3	54.4	**47.4**	**61.3**	52.2	15.4	26.0	17.9	47.5	55.3	53.1
	IterDRAG	**63.2**	**74.8**	**68.8**	44.8	59.4	**52.8**	**17.3**	**28.0**	**24.5**	**62.3**	**73.8**	**74.6**
1M	DRAG	56.0	62.9	57.6	47.4	61.3	52.2	15.9	26.0	18.2	48.2	55.7	53.3
	IterDRAG	**65.6**	**75.6**	**68.8**	**48.7**	**63.3**	**55.3**	**22.2**	**34.3**	**30.5**	**65.7**	**75.2**	**76.4**
5M	IterDRAG	**65.6**	**75.6**	**68.8**	**51.7**	**64.4**	**56.4**	**22.5**	**35.0**	**30.5**	**67.0**	**75.2**	**76.9**

RAG Performance and Inference Computation Scale

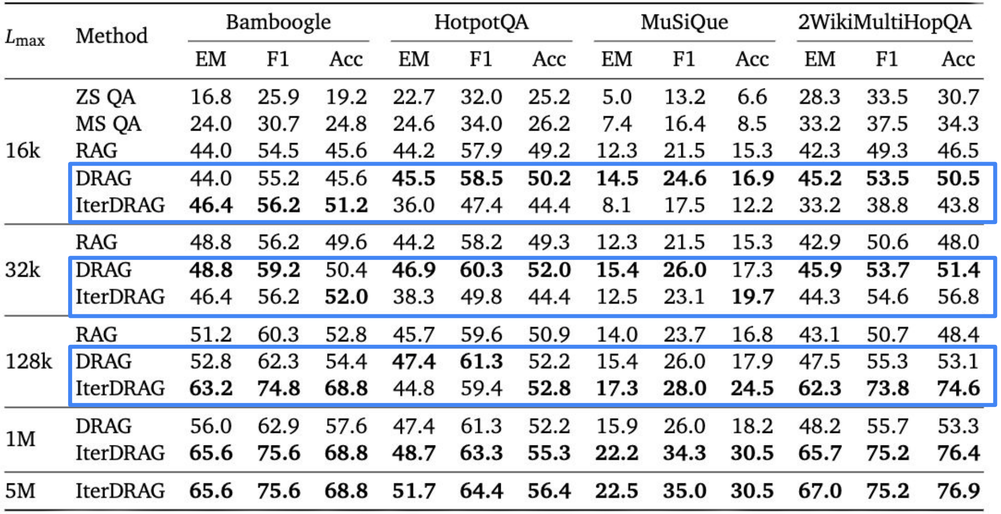
Comparing Strategies

Evaluated on 4 multi-hop open-book question answering datasets.

Baselines:

- **Zero-Shot QA (ZS QA):** 0 retrieved document, 0 demonstration
- **Many-Shot QA (MS QA):** 0 retrieved document, many demonstrations
- **RAG**: many retrieved documents, 0 demonstration

Optimal performance of different methods with varying maximum effective context lengths.



$L_{\max}$	Method	Bamboogle			HotpotQA			MuSiQue			2WikiMultiHopQA		
		EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
16k	ZS QA	16.8	25.9	19.2	22.7	32.0	25.2	5.0	13.2	6.6	28.3	33.5	30.7
	MS QA	24.0	30.7	24.8	24.6	34.0	26.2	7.4	16.4	8.5	33.2	37.5	34.3
	RAG	44.0	54.5	45.6	44.2	57.9	49.2	12.3	21.5	15.3	42.3	49.3	46.5
	DRAG	44.0	55.2	45.6	**45.5**	**58.5**	**50.2**	**14.5**	**24.6**	**16.9**	**45.2**	**53.5**	**50.5**
	IterDRAG	**46.4**	**56.2**	**51.2**	36.0	47.4	44.4	8.1	17.5	12.2	33.2	38.8	43.8
32k	RAG	48.8	56.2	49.6	44.2	58.2	49.3	12.3	21.5	15.3	42.9	50.6	48.0
	DRAG	**48.8**	**59.2**	50.4	**46.9**	**60.3**	**52.0**	**15.4**	**26.0**	17.3	**45.9**	**53.7**	**51.4**
	IterDRAG	46.4	56.2	**52.0**	38.3	49.8	44.4	12.5	23.1	**19.7**	44.3	54.6	56.8
128k	RAG	51.2	60.3	52.8	45.7	59.6	50.9	14.0	23.7	16.8	43.1	50.7	48.4
	DRAG	52.8	62.3	54.4	**47.4**	**61.3**	52.2	15.4	26.0	17.9	47.5	55.3	53.1
	IterDRAG	**63.2**	**74.8**	**68.8**	44.8	59.4	**52.8**	**17.3**	**28.0**	**24.5**	**62.3**	**73.8**	**74.6**
1M	DRAG	56.0	62.9	57.6	47.4	61.3	52.2	15.9	26.0	18.2	48.2	55.7	53.3
	IterDRAG	**65.6**	**75.6**	**68.8**	**48.7**	**63.3**	**55.3**	**22.2**	**34.3**	**30.5**	**65.7**	**75.2**	**76.4**
5M	IterDRAG	**65.6**	**75.6**	**68.8**	**51.7**	**64.4**	**56.4**	**22.5**	**35.0**	**30.5**	**67.0**	**75.2**	**76.9**

DRAG and IterDRAG consistently achieve higher performance than other strategies on different L_{max}

RAG Performance and Inference Computation Scale

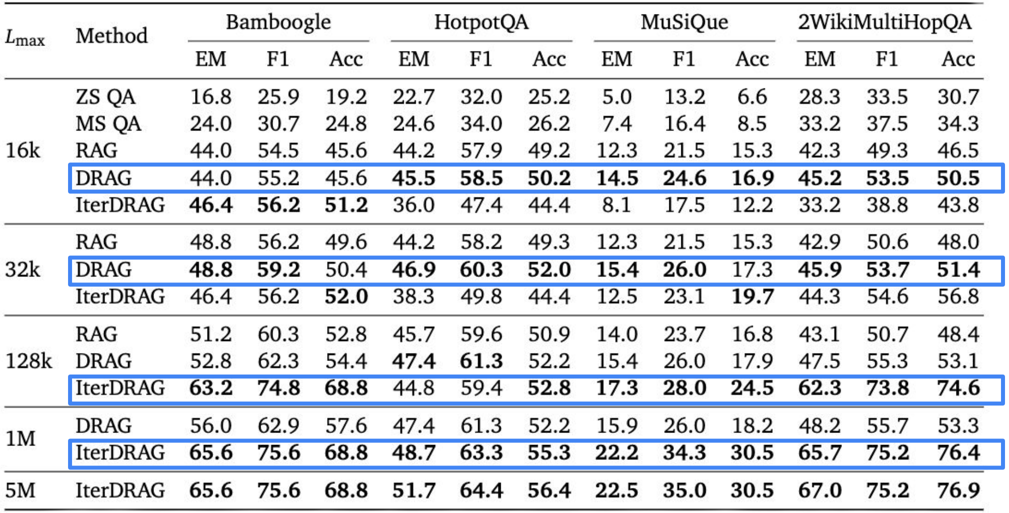
Comparing Strategies

Evaluated on 4 multi-hop open-book question answering datasets.

Baselines:

- **Zero-Shot QA (ZS QA):** 0 retrieved document, 0 demonstration
- **Many-Shot QA (MS QA):** 0 retrieved document, many demonstrations
- **RAG:** many retrieved documents, 0 demonstration

Optimal performance of different methods with varying maximum effective context lengths.



L_{max}	Method	Bamboogle			HotpotQA			MuSiQue			2WikiMultiHopQA		
		EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
16k	ZS QA	16.8	25.9	19.2	22.7	32.0	25.2	5.0	13.2	6.6	28.3	33.5	30.7
	MS QA	24.0	30.7	24.8	24.6	34.0	26.2	7.4	16.4	8.5	33.2	37.5	34.3
	RAG	44.0	54.5	45.6	44.2	57.9	49.2	12.3	21.5	15.3	42.3	49.3	46.5
	DRAG	44.0	55.2	45.6	**45.5**	**58.5**	**50.2**	**14.5**	**24.6**	**16.9**	**45.2**	**53.5**	**50.5**
	IterDRAG	**46.4**	**56.2**	**51.2**	36.0	47.4	44.4	8.1	17.5	12.2	33.2	38.8	43.8
32k	RAG	48.8	56.2	49.6	44.2	58.2	49.3	12.3	21.5	15.3	42.9	50.6	48.0
	DRAG	**48.8**	**59.2**	50.4	**46.9**	**60.3**	**52.0**	**15.4**	**26.0**	17.3	**45.9**	**53.7**	**51.4**
	IterDRAG	46.4	56.2	**52.0**	38.3	49.8	44.4	12.5	23.1	**19.7**	44.3	54.6	56.8
128k	RAG	51.2	60.3	52.8	45.7	59.6	50.9	14.0	23.7	16.8	43.1	50.7	48.4
	DRAG	52.8	62.3	54.4	**47.4**	**61.3**	52.2	15.4	26.0	17.9	47.5	55.3	53.1
	IterDRAG	**63.2**	**74.8**	**68.8**	44.8	59.4	**52.8**	**17.3**	**28.0**	**24.5**	**62.3**	**73.8**	**74.6**
1M	DRAG	56.0	62.9	57.6	47.4	61.3	52.2	15.9	26.0	18.2	48.2	55.7	53.3
	IterDRAG	**65.6**	**75.6**	**68.8**	**48.7**	**63.3**	**55.3**	**22.2**	**34.3**	**30.5**	**65.7**	**75.2**	**76.4**
5M	IterDRAG	**65.6**	**75.6**	**68.8**	**51.7**	**64.4**	**56.4**	**22.5**	**35.0**	**30.5**	**67.0**	**75.2**	**76.9**

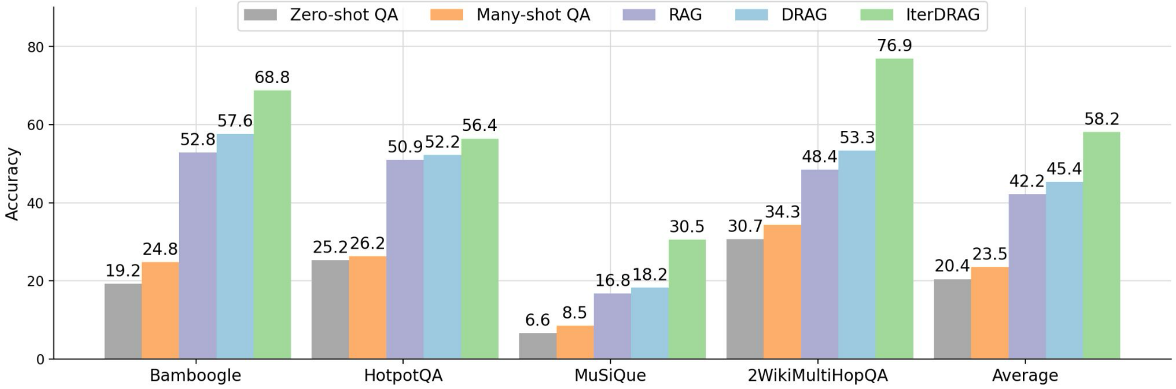
DRAG excels with shorter effective context lengths but IterDRAG scales more effectively for longer ones

RAG Performance and Inference Computation Scale

Comparing Strategies

Comparing the optimal performance with effective context length L_{max} up to 5M

DRAG and IterDRAG can achieve better performance than baselines



DRAG and IterDRAG with the optimal configuration leverage the context window better than RAG.

DRAG and IterDRAG with the optimal configuration leverage the context window better than RAG.

How to find the optimal configuration without brute-force?

Inference Computation Allocation for RAG

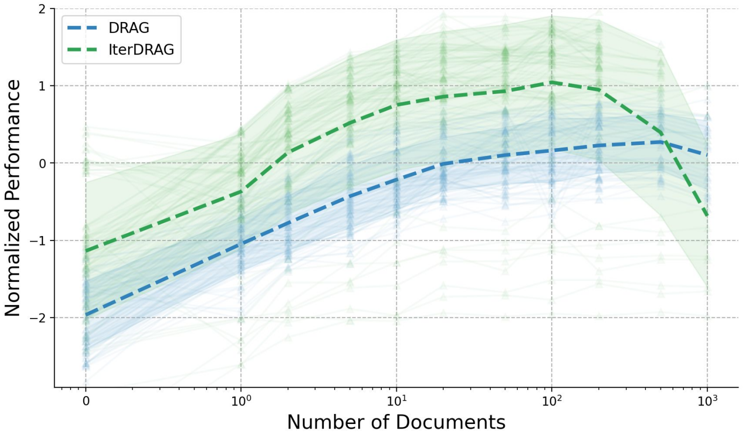
4

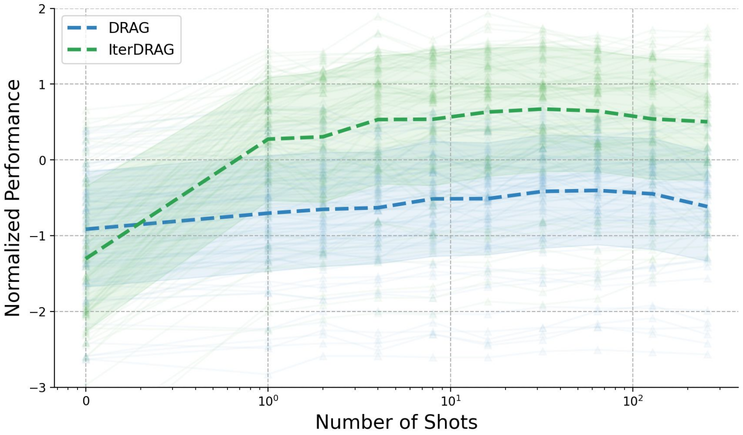
RAG Performance and Inference Computation Scale

RAG Performance vs. Different Parameters

Plotting DRAG and IterDRAG performance vs. individual parameter: number of documents and number of demonstrations.

1. *Number of documents* is more helpful than *number of demonstrations*, as the curve has steeper slope
2. IterDRAG benefits more from increasing number of demonstration





Inference Computation Allocation for RAG

Introducing a Quantitative Model

Denote the parameters of the strategies as $\theta = [k, m, n]^{\top}$ we can formulate the computation allocation model as

$$\sigma^{-1}(P(\theta)) \approx (a + b \odot i)^{\top} \log(\theta) + c$$

where

- *a*, *b*, *c* are parameters to learn;
- *i* is a vector of informativeness that can be easily estimated for each dataset individually;
- σ is a link function and ⊙ refers to element-wise product.

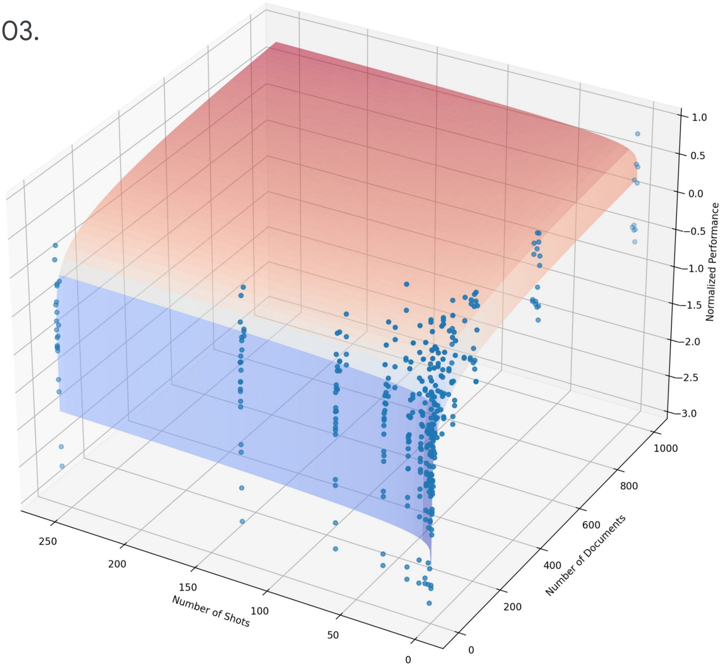
The informativeness *i* include informativeness for adding a document, and informativeness for adding a demonstration, estimated by

1. i_{doc} = performance difference between k=1 document and k=0 document
2. i_{shot} = performance difference between m=1 demo and m=0 demo respectively
3. i_{iter} = 0 as we do not find an accurate way to estimate informativeness of adding an iteration

Inference Computation Allocation for RAG

Estimated Model

The estimated model has an R^2 of 0.903.

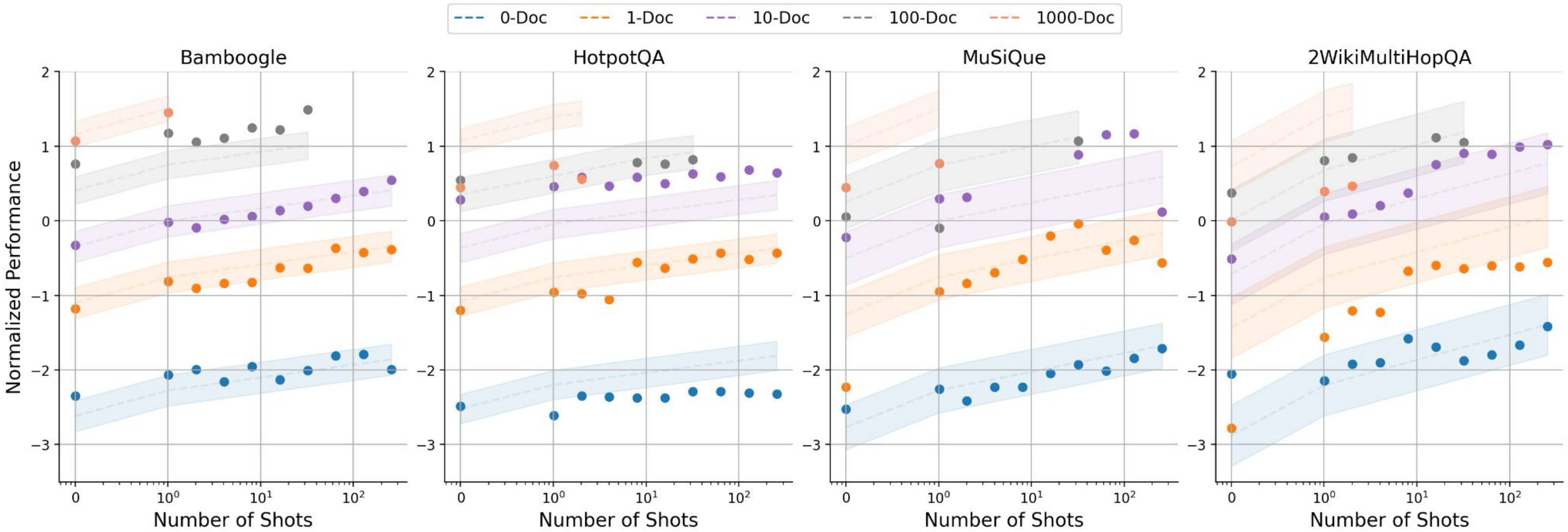


Inference Computation Allocation for RAG

Estimated Model

The estimated model has an R^2 of 0.903.

Plot the model estimation (shaded area) vs. the actual performance for a few slices of configurations:



Inference Computation Allocation for RAG

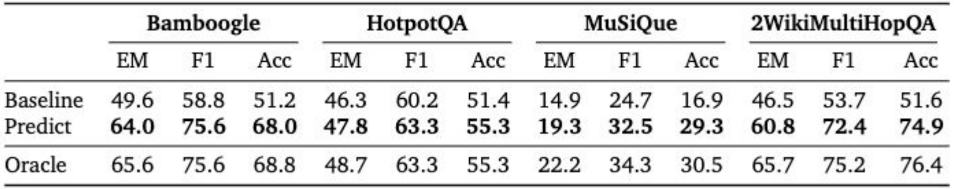
Predict the Optimal Configuration

Evaluate how well the model generalize in two different ways:

1. Use 3 datasets to fit the model and predict the optimal on the other one
2. Use data from shorter effective context lengths to fit the model and predict the optimal on longer effective context lengths

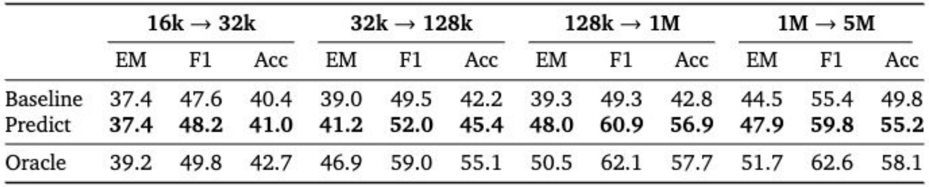
*Baseline = always 8-shot and fill the context length with as many documents as possible

Generalization to other datasets



	Bamboogle			HotpotQA			MuSiQue			2WikiMultiHopQA		
	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
Baseline	49.6	58.8	51.2	46.3	60.2	51.4	14.9	24.7	16.9	46.5	53.7	51.6
Predict	**64.0**	**75.6**	**68.0**	**47.8**	**63.3**	**55.3**	**19.3**	**32.5**	**29.3**	**60.8**	**72.4**	**74.9**
Oracle	65.6	75.6	68.8	48.7	63.3	55.3	22.2	34.3	30.5	65.7	75.2	76.4

Generalization to longer context lengths



	16k → 32k			32k → 128k			128k → 1M			1M → 5M		
	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc
Baseline	37.4	47.6	40.4	39.0	49.5	42.2	39.3	49.3	42.8	44.5	55.4	49.8
Predict	**37.4**	**48.2**	**41.0**	**41.2**	**52.0**	**45.4**	**48.0**	**60.9**	**56.9**	**47.9**	**59.8**	**55.2**
Oracle	39.2	49.8	42.7	46.9	59.0	55.1	50.5	62.1	57.7	51.7	62.6	58.1

Summary

5

Summary

01

We comprehensively investigate inference scaling for RAG in the regime of long-context LLMs.

We use two inferences scaling strategies: DRAG and IterDRAG.

02

With an enriched toolbox to scale inference computation, we observe that the optimal RAG performance can scale almost linearly with the order of magnitude of inference computation.

This is different from previous observations where the RAG performance tends to saturate or drop when only scaling the number of documents.

03

We develop a quantitative model to predict RAG performance for a specific inference parameter configuration.

The model can be used to *find the optimal configuration* for a given inference computation budget.

Experiments show reasonable predictive power.

Thank you.

Google DeepMind