

MHA-RAG: IMPROVING EFFICIENCY, ACCURACY, AND CONSISTENCY BY ENCODING EXEMPLARS AS SOFT PROMPTS

Anonymous authors

Paper under double-blind review

ABSTRACT

Adapting Foundation Models to new domains with limited training data is challenging and computationally expensive. While prior work has demonstrated the effectiveness of using domain-specific exemplars as in-context demonstrations, we investigate whether representing exemplars purely as text is the most efficient, effective, and stable approach. We explore an alternative: representing exemplars as soft prompts with an exemplar order invariant model architecture. To this end, we introduce Multi-Head Attention Retrieval-Augmented Generation (MHA-RAG), a framework with the number of attention heads serving as a simple hyperparameter to control soft prompt-generation across different tasks. Across multiple question-answering benchmarks and model scales, MHA-RAG achieves a 20-point performance gain over standard RAG, while cutting inference costs by a factor of $10\times$ GFLOPs—delivering both higher accuracy and greater efficiency, invariant to exemplar order.

1 INTRODUCTION

With the rapid scaling of model parameters, tuning Foundation Models for domain adaptation has become increasingly challenging due to both computational and data constraints. As an alternative, In-Context Learning (ICL) has emerged as a training-free adaptation strategy (Xie et al., 2021; Min et al., 2022). Rather than updating model parameters through gradient descent, ICL conditions the model at inference time by providing a small set of task-specific exemplars directly in the input prompt. Pioneering work such as GPT-3 (Brown et al., 2020) demonstrated that LLMs exhibit strong in-context learning capabilities when presented with exemplars in natural-language form. Subsequent studies in Wei et al. (2022) show that adding chain-of-thought exemplars further enhances performance by encouraging LLMs to generate intermediate reasoning steps. Remarkably, in certain settings, ICL has even outperformed fine-tuning-based methods (Mosbach et al., 2023; Pingua et al., 2025; Mallen et al., 2022).

Despite its promise, ICL faces three fundamental limitations. First, representing exemplars in textual format leads to **increasing inference cost**, because attention complexity quadratically grows with context length (Vaswani et al., 2017). Second, RAG has been shown to give **unsatisfiable performance** in domains with out-of-distribution data (Yu et al., 2024; Gupta et al., 2024). Third, ICL is highly sensitive to exemplar order (**exemplar-order variance**): prior studies demonstrate that reordering exemplars can cause substantial performance fluctuations (Pezeshkpour & Hruschka, 2023; Zheng et al., 2023), and we observe similar phenomenon in our own experiments (see Table 2).

To mitigate these issues, soft-prompt-based tuning is a viable choice. Instead of representing exemplars as long sequences of text, soft prompts encode exemplar information as a fixed set of trainable continuous vectors, which are prepended to the input. Because these vectors are much shorter than raw text, inference cost scales with the number of soft tokens rather than the full text length, greatly reducing quadratic overhead. While methods have been proposed for compressing text exemplars into soft prompts (Mu et al., 2023; Chevalier et al., 2023; Cheng et al., 2024; Rau et al., 2024; Li et al., 2024), these approaches often incur a noticeable performance degradation compared to text-based ICL. An alternative line of research investigates whether soft prompting can stand as an effective parameter-efficient fine-tuning method for domain adaptation (Lester et al., 2021; Liu et al.,

2024), (Liu et al., 2021). Approaches such as ATTEMPT (Asai et al., 2022), Instance-Dependent Prompt Generation (Wu et al., 2022) and LoPA (Jain et al., 2024a) demonstrate that adapting language models with soft prompts is a cost-saving mechanism that can enhance their performance without modifying model weights.

Therefore, in this paper, we ask the question: Can we learn a soft-prompt representation over in-context exemplars that simultaneously leverages 1) the cost efficiency of soft prompts, 2) the adaptability of in-context learning with retrieved exemplars, and 3) the architecture of neural networks that provide an exemplar-order invariant framework?

To address this question, we propose the Multi-Head Attention Retrieval-Augmented Generation framework (MHA-RAG). MHA-RAG learns to represent in-context exemplars as compact soft prompts and employs a multi-head scaled dot-product-based attention (Vaswani et al., 2017) to capture rich interactions between the query and each exemplar. Moreover, MHA-RAG introduces the number of heads as a tunable hyperparameter, enabling flexible control over the length of the generated soft prompts while simultaneously enforcing order-invariant aggregation across exemplar representations. Our experiments with a wide range of language models on domains like Chemistry and Medical QA demonstrate that this design provides (i) substantial gains in inference efficiency by reducing quadratic token overhead, (ii) performance improvements over text-based retrieval and prompt-tuning baselines, and (iii) stable and consistent results with the property of order-invariance embodied in the design.

In this work, we make the following contributions:

- We address the three fundamental limitations in RAG, which include high inference cost, unsatisfactory performance, and exemplar order variance within domain adaptations.
- We propose the MHA-RAG framework, which learns compact soft-prompt representations over in-context examples to reduce inference cost, while achieving high performance.
- We propose an exemplar order invariant multi-head-attention architecture to control soft-prompt generation in MHA-RAG, where the number of heads is a tunable hyperparameter.
- Our experiments show on average a 20-point performance gain over standard RAG across benchmarks and models, while cutting inference cost by $10\times$ GFLOPs.

2 RELATED WORK

2.1 PROMPT-COMPRESSSION METHODS

Prompt-compression methods can be mainly divided into two categories: soft-prompt methods and hard-prompt methods. Soft-prompt methods compress long contexts into dense representations to reduce token count. GIST (Mu et al., 2023) learns “gist tokens” via modified attention masks, letting generation condition only on a compact set of vectors. AutoCompressor (Chevalier et al., 2023) and ICAE (Ge et al., 2023) similarly encode long inputs into summary vectors or memory slots that can be reused without re-encoding. More recent works like PISCO (Louis et al., 2025) and Embedding-based Memory Compressors (Dai et al., 2025) train memory tokens through distillation or pretraining, while CMT (Li et al., 2025) encodes documents into dense vectors via cross-attention. These methods achieve efficiency by replacing raw text with compact learned embeddings.

Hard-prompt methods remove tokens or text that contains less information. LLMlingua (Jiang et al., 2023) iteratively prunes tokens with low perplexity, while RECOMP (Xu et al., 2023) selects or summarizes sentences before prepending them to the input. Hierarchical methods, such as From Reading to Compressing (R2C) (Choi et al., 2024), compress at the token, chunk, and sentence levels, and training-free frameworks, like Perception Compressor (Tang et al., 2024), dynamically allocate compression ratios across different parts of the input. These techniques are often model-agnostic, but must balance compression rate with semantic fidelity.

In retrieval settings, compression enables handling many documents efficiently. COMPACT (Yoon et al., 2024) sequentially compresses document segments, while EXIT (Hwang et al., 2024) filters retrieved sentences to keep only relevant ones. xRAG (Cheng et al., 2024) projects a whole document into a single token embedding, and COCOM (Rau et al., 2024) pre-computes compact embeddings for

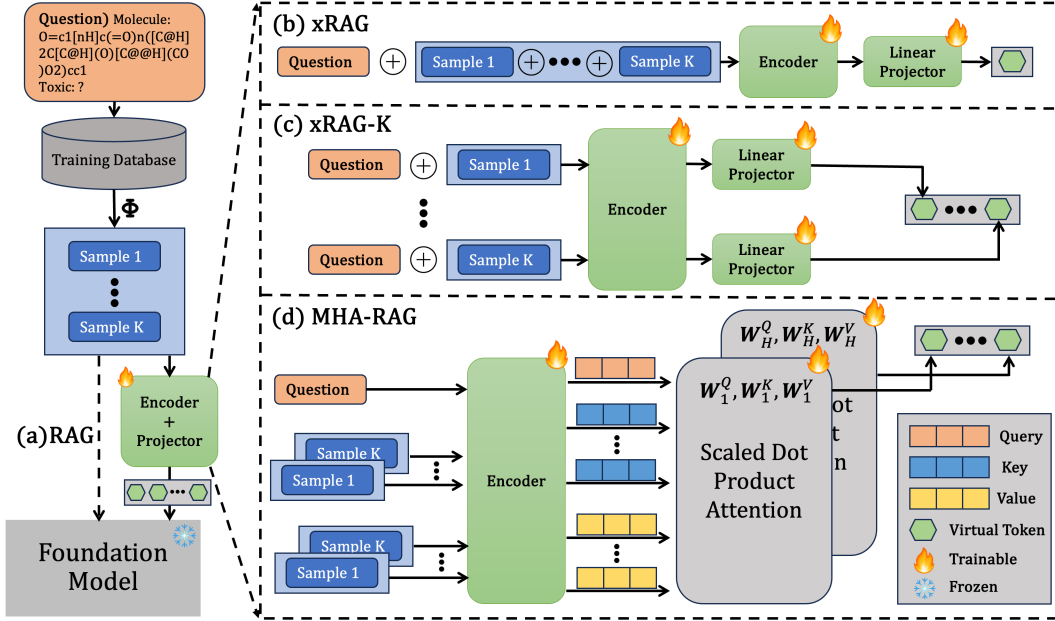


Figure 1: Comparison of domain-adaptation methods: (a) RAG uses retrieved exemplars directly as context, while (b) xRAG, (c) xRAG-K, and (d) MHA-RAG derive soft-prompt representations from exemplars. The figure illustrates how each method forms its representation (single vector in xRAG, K vectors in xRAG-K, multi-head representations in MHA-RAG) and highlights their trainable components. Among the four methods, only MHA-RAG—due to its use of scaled dot-product attention—is invariant to the order of exemplars.

offline reuse. Beyond token inputs, DyPRAG (Tan et al., 2025) injects retrieved knowledge as LoRA weight updates, and AttentionRAG (Fang et al., 2025) uses model attention to prune retrievals in a training-free manner. Together, these methods reduce redundancy in RAG pipelines while controlling cost. Our MHA-RAG approach employs a multi-head attention-based soft prompt over retrieved in-context exemplars, enabling rich interactions between the query and exemplars, while ensuring that exemplar aggregation is order-invariant. Moreover, we can use the number of heads in the multi-head attention mechanism as a tunable hyperparameter to control soft-prompt generation across various tasks and models.

2.2 RETRIEVAL METHOD VERSUS FINE-TUNING

Retrieval-Augmented Generation (RAG) augments Foundation Models with external documents, providing grounded and up-to-date responses that surpass static fine-tuning methods in many knowledge-intensive tasks (Lewis et al., 2020). The proposed RAG model achieved state-of-the-art accuracy on open-domain QA benchmarks by retrieving different documents per query, outperforming purely fine-tuned parametric baselines. Similarly, REALM (Gua et al., 2020) and RETRO (Borgeaud et al., 2022) demonstrated that retrieval integration can enable smaller models to rival or surpass much larger fine-tuned ones.

Moreover, RAG leverages retrieved examples as in-context demonstrations, enabling Foundation Models to adapt to new domains without retraining. This strategy has been shown to improve performance in low-data and specialised settings, with unsupervised domain-adaptation approaches demonstrating significant gains in tasks like sentiment analysis and named entity recognition, etc. (Gua et al., 2020; Lewis et al., 2020; Borgeaud et al., 2022; Jain et al., 2024b). More recent benchmarks highlight that retrieval augmentation offers a lightweight, cost-efficient alternative to fine-tuning by dynamically expanding the model’s accessible knowledge (Behrouz et al., 2024).

Another fundamental difference between RAG and fine-tuned models lies in test-time adaptability. RAG performs a fresh retrieval for each user query, dynamically tailoring the context to the question at hand. Recent work on dynamic test-time compute for LLMs explores strategies that allocate inference

resources adaptively based on query complexity. Snell et al. (2024) formalizes compute-optimal strategies—either using verifier-based search or adaptive response distribution, and outperforms much larger models under a fixed FLOPs budget. MHA-RAG leverages the benefits of RAG for domain adaptation—achieving superior performance in domain-specific tasks with less inference cost compared to RAG.

3 METHODOLOGY

The objective of domain adaptation is to adapt a language model f_θ to a new task with data $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^N$ such that $y = f_\theta(x)$.

3.1 PRELIMINARIES

Formally, domain adaptation involves updating the parameters of the model θ to maximize the likelihood of the response y , i.e., $\max_\theta \mathbb{E}_{(x,y) \sim \mathbb{D}} [\log p_\theta(y|x)]$. In practice, Foundation Models are often adapted using parameter-efficient fine-tuning (PEFT) methods (He et al., 2021), where only a subset of parameters $\theta' \subset \theta$ is updated. This approach reduces training costs while preserving performance.

Domain adaptation with in-context exemplars. In domains where parameter updates are not feasible, either due to limited training data or training being computationally expensive, adaptation can be achieved by providing a set of a few domain-specific samples $\{(x_k, y_k)\}_{k=1}^K \subset \mathbb{D}$ as in-context exemplars. In this setting, the model prediction takes the form $y = f_\theta(\{(x_k, y_k)\}_{k=1}^K, x)$. This adaptation can be further enhanced by leveraging a relevance or similarity function $\Phi(x, x')$. Specifically, for each input x , the top- K most relevant samples in $\mathbb{D}$ are retrieved as in-context exemplars—i.e., $(x_k, y_k) = \arg \max_{(x', y') \in \mathbb{D} \setminus \{(x, y), (x_1, y_1), \dots, (x_{k-1}, y_{k-1})\}} \Phi(x, x')$, for $1 \leq k \leq K$.

Representing in-context exemplars. While approaches based on RAG typically represent in-context exemplars in text form, we instead consider soft prompts as an alternative representation. To this end, we define an encoding function $g(\mathbb{D}_K|x) = \mathbf{Z}$ that embeds $\mathbb{D}_K$, the top- K in-domain samples retrieved for a given x , into a soft prompt $\mathbf{Z} \in \mathbb{R}^{d \times m}$, where m denotes the virtual prompt length and d is the hidden dimension of the language model. Similar to prompt tuning, the learned soft prompt is prepended to the word embeddings of x , thereby enabling domain adaptation without updating θ .

3.2 MULTI-HEADED ATTENTION-BASED RAG

In this paper, we propose a multi-headed attention-based encoding function for RAG (MHA-RAG), where each head projects the top- K in-context exemplars into the embedding space of the language model as a single soft-prompt vector $\mathbf{z} \in \mathbb{R}^d$. In particular, performing attention with H heads results in the soft prompt $\mathbf{Z}_{MHA} = [\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(i)}, \dots, \mathbf{z}^{(H)}]$ of length $m = H$, where the output from the i^{th} head is

$$\begin{aligned} \mathbf{z}^{(i)} &= \text{ATTENTION}(\mathbf{q}^i, \mathbf{K}^i, \mathbf{V}^i) \\ &= \text{softmax}\left(\frac{\mathbf{q}^i \mathbf{K}^i}{\sqrt{d}}\right) \mathbf{V}^i, \end{aligned}$$

$$\text{with } \mathbf{K}^i = [\mathbf{k}_1^i, \dots, \mathbf{k}_k^i, \dots, \mathbf{k}_K^i] \text{ and } \mathbf{V}^i = [\mathbf{v}_1^i, \dots, \mathbf{v}_k^i, \dots, \mathbf{v}_K^i],$$

where $\mathbf{q}^i = \mathbf{W}_i^Q E_x$ is the attention query corresponding to a given x , and $\mathbf{k}_k^i = \mathbf{W}_i^K E_{x_k \oplus y_k}$ and $\mathbf{v}_k^i = \mathbf{W}_i^V E_{x_k \oplus y_k}$ are the keys and values corresponding to $(x_k, y_k) \in \mathbb{D}_K$, respectively.

$E_x \in \mathbb{R}^{d'}$ represents the dense representation of x obtained by a sentence-embedding model with hidden dimension d' ; $\oplus$ denotes the concatenation operator; and $\mathbf{W}_i^Q \in \mathbb{R}^{d \times d'}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times d'}$, and $\mathbf{W}_i^V \in \mathbb{R}^{d \times d'}$ represent the attention query, key, and value weights associated with head i , respectively. Overall, we optimize the following objective:

$$\max_{\varphi} \mathbb{E}_{(x,y) \sim \mathbb{D}, \mathbb{D}_K = \Phi(x, \cdot)} [\log p(y|g_{\varphi}(\mathbb{D}_K|x), x)],$$

such that $g_{\varphi}(\mathbb{D}_K|x) = \mathbf{Z}_{MHA}$, where φ represents the parameters of the encoding function g .

Benchmarks	RAG	xRAG	xRAG-K	MHA-RAG
<i>Qwen3-0.6B</i>				
BACE	<u>76.01</u>	60.88 $\downarrow$ -15.13	66.48 $\downarrow$ -9.53	75.08 $\downarrow$ -0.93
BBBP	66.62	68.16 $\uparrow$ +1.54	76.57 $\uparrow$ +9.95	<u>87.82</u> $\uparrow$ +21.20
ClinTox	53.30	42.54 $\downarrow$ -10.76	66.05 $\uparrow$ +12.75	<u>97.12</u> $\uparrow$ +43.82
PubMedQA	58.09	<u>72.49</u> $\uparrow$ +14.4	69.73 $\uparrow$ +11.64	66.52 $\uparrow$ +8.43
<i>Qwen3-4B</i>				
BACE	75.87	59.07 $\downarrow$ -16.8	55.46 $\downarrow$ -20.41	<u>76.27</u> $\uparrow$ +0.4
BBBP	69.33	64.44 $\downarrow$ -4.89	81.40 $\uparrow$ +12.07	<u>82.49</u> $\uparrow$ +13.16
ClinTox	44.24	40.19 $\downarrow$ -4.05	57.07 $\uparrow$ +12.83	<u>96.32</u> $\uparrow$ +52.08
PubMedQA	<u>79.26</u>	71.16 $\downarrow$ -8.10	70.36 $\downarrow$ -8.9	76.59 $\downarrow$ -2.67
<i>Llama3.2-3B-Instruct</i>				
BACE	49.89	54.85 $\downarrow$ -4.96	64.04 $\uparrow$ +14.15	<u>67.62</u> $\uparrow$ +17.73
BBBP	46.15	72.87 $\uparrow$ +26.72	84.09 $\uparrow$ +37.94	<u>87.61</u> $\uparrow$ +41.46
ClinTox	37.90	62.88 $\uparrow$ +24.98	86.89 $\uparrow$ +48.99	<u>94.44</u> $\uparrow$ +56.54
PubMedQA	71.38	75.08 $\uparrow$ +3.7	75.71 $\uparrow$ +4.33	<u>76.34</u> $\uparrow$ +4.96
$\Delta_{\text{RAG}}^{\text{avg}}$		-0.36	8.99	<u>19.66</u>

Table 1: Baseline comparison with RAG across benchmarks ($K = 5$). Performance is reported as effective accuracy—i.e., the geometric mean of the True-Positive and True-Negative rates ($\uparrow$: improvement relative to RAG, $\downarrow$: drop relative to RAG). Random guessing yields an effective accuracy of 50. Hyperparameters are tuned via sweeps: MHA-RAG ($H \in \{1, 2, 4, 8\}$). Results are averaged over 3 random seeds. Underlined indicates the best effective accuracy.

Context compression and lower inference cost. Prior work (Cheng et al., 2024) interprets the encoding function $g(\cdot)$ as a context-compression module achieving a compression ratio of $\frac{|\mathbb{D}_K|}{m}$. It reduces the tokenized length of the context $\mathbb{D}_K$ from $|\mathbb{D}_K|$ to m , thereby lowering overall inference cost during deployment. RAG without any compression yields a compression ratio of 1. In contrast, MHA-RAG achieves a higher and configurable compression ratio of $\frac{|\mathbb{D}_K|}{H}$ ($m = H$). Configurability comes from tailoring the soft-prompt size to the domain by varying the number of heads to adjust its representational capacity. In comparison, xRAG (Cheng et al., 2024), while achieving the highest possible compression (with $m = 1$), has a fixed ratio of $|\mathbb{D}_K|$ as it collapses the context into a single vector (see Figure 1 for reference).

Order Invariance. We argue that the soft prompts generated by MHA-RAG are invariant to the order of retrieved exemplars. This property follows from the use of scaled dot-product attention, which depends only on the set of input tokens rather than their order (Vaswani et al., 2017). Specifically, the inputs are exemplar embeddings $[E_{x_1 \oplus y_1}, \dots, E_{x_K \oplus y_K}]$, and permuting them does not alter the aggregated representation produced by each head. As a result, the generated soft prompt remains identical regardless of exemplar ordering. A proof of this property is provided in Appendix A.1.

4 ADAPTING TO QUESTION-ANSWERING DOMAINS

4.1 EXPERIMENTAL SETUP

Benchmarks. In this work, we focus on domains where language models exhibit weak zero-shot performance, but can benefit substantially from in-context exemplars or documents. Such exemplars provide cues in the form of analogies to similar samples, domain-specific formats (e.g., SMILES strings for molecules), specialized vocabulary and facts, or templates for step-by-step reasoning.

We therefore benchmark across two groups of tasks. The first group involves limited-data molecular-property-prediction tasks: (a) *BACE* [train/test split: 1413/100]—binary classification of whether a molecule inhibits BACE1; (b) *BBBP* [train/test split: 1950/100]—prediction of blood–brain barrier penetration (Yes/No); (c) *ClinTox* [train/test split: 1384/100]—classification of molecules as clinically

trial toxic vs. non-toxic (Guo et al., 2023). The second group consists of a medium-scale task: (d) *PubMedQA* [train/test split: 30250/890]—biomedical question answering (Jin et al., 2019).

Performance Metrics. For yes/no classification tasks, we report the geometric mean of the True-Positive and True-Negative rates, because it equally values both classes and avoids the biases of accuracy or F1 under skewed label distributions. We call this quantity the “Effective Accuracy.” To quantify overall improvement relative to RAG, we compute $\Delta_{\text{RAG}}^{\text{avg}} = \sqrt[n]{\prod_{i=1}^n (100 + o_i^{\text{method}} - o_i^{\text{RAG}})} - 100$ where o_i is the score of a method on task i , and n is the total number of benchmark tasks across language models.

Foundation Models (θ). We evaluate two families of models: *Llama-3.2-3B-Instruct* and the *Qwen3* series, specifically *Qwen3-0.6B* and *4B* (Yang et al., 2025).

Sentence-Embedding Models (E_x). To encode in-context exemplars, we employ domain-specific embedding models or encoder-only language models. For molecular tasks (*BACE1*, *BBBP*, *ClinTox*), we use *ChemBERTa-2-10M-MTR* (Ahmad et al., 2022). For biomedical QA (*PubMedQA*), we use *Qwen3-Embedding-0.6B* (Zhang et al., 2025).

Retrieval Functions (Φ). We assume that a domain-specific retrieval function is provided. For molecular datasets, we retrieve the top- K relevant exemplars using Tanimoto Similarity (Tanimoto, 1958; Guo et al., 2023), which computes scaffold-level similarity between SMILES representations.

For *PubMedQA*, we retrieve the top- K documents most relevant to a question from the training corpus. Each document doc_k is embedded into a dense vector E_{doc_k} using *Qwen3-Embedding-8B* (Zhang et al., 2025), and retrieval is performed via cosine similarity.

Baselines. We compare against domain-adaptation baselines that do not fine-tune the foundation model. Instead, they leverage retrieved in-context exemplars from the training database. These include (a) *RAG*, which presents exemplars directly as text to the model, and (b) *xRAG* and (c) *xRAG-K*, which construct soft prompts from the exemplars. In *xRAG*, all exemplars are encoded into a single vector—i.e., $Z = \text{MLP}(E_{x \oplus x_1 \oplus y_1} \dots x_K \oplus y_K)$, where $Z \in \mathcal{R}^{d \times 1}$. *xRAG-K*, on the other hand, encodes each exemplar separately; i.e., $Z = \text{MLP}(E_{x \oplus x_1 \oplus y_1}) \oplus \dots \oplus \text{MLP}(E_{x \oplus x_K \oplus y_K})$ where $Z \in \mathcal{R}^{d \times K}$. Both methods employ a single-layer MLP to project the soft prompts from the embedding space of the sentence encoder into the base model’s input space. Refer Figure 1.

Training Details. Models are tuned for 10 epochs on the limited-data molecular benchmarks (*BACE1*, *BBBP*, *ClinTox*) and 1 epoch on the larger-scale benchmarks (*PubMedQA*). Learning rates are selected from $\{1e-5, 3e-5, 5e-4\}$. The execution platform used $8 \times$ V100 GPUs (32GB VRAM each). The embedding models are fine-tuned with LoRA (rank 64). The comparison of overall trainable parameters are provided in Appendix A.5.

We report results on additional benchmarks and experiments in Appendix A.2 and A.3.

4.2 BASELINE COMPARISON

Performance. As shown by the underlining in Table 1, MHA-RAG achieves the greatest improvement in effective accuracy over RAG for almost all configurations tested, with an average gain of 19.66, computed as the geometric mean of improvements across tasks. This improvement stems from its flexible representational capacity: by varying the number of attention heads—each with its own set of weights—the model can specialize in capturing different types of dependencies across exemplars. In contrast, *xRAG* performs extreme compression by collapsing all retrieved samples into a single vector, which can possibly lead to loss of information. Its extension, *xRAG-K* improves over RAG

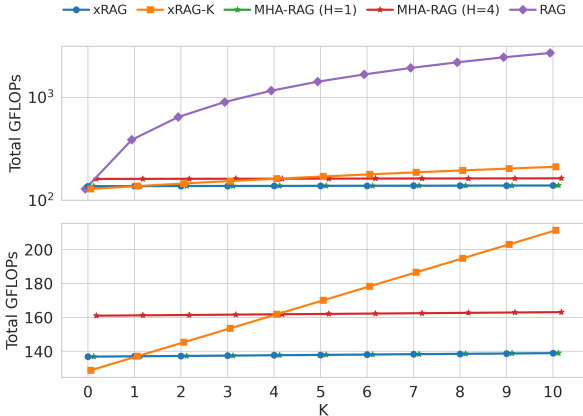


Figure 2: Comparison of inference compute in FLOPs with respect to K

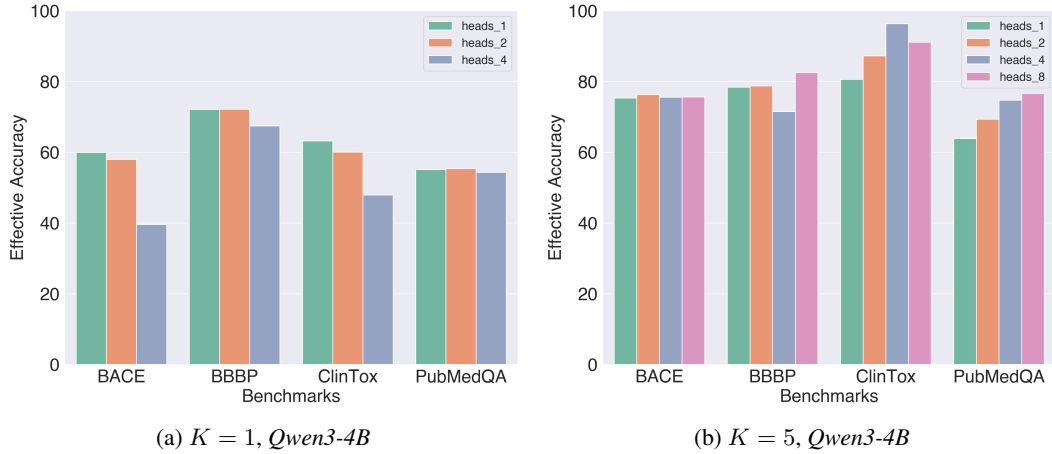


Figure 3: Varying number of heads in MHA-RAG. Performance averaged across 3 seeded runs.

by encoding each exemplar separately, but all exemplars are processed with identical weights. In contrast, MHA-RAG assigns distinct weights to each head and computes representations through attention over all exemplars, allowing head-specific representations.

Inference Cost. Figure 2 further shows that across $K = 1 \dots 10$, RAG incurs substantially higher inference cost (in FLOPs) than MHA-RAG, due to its longer context length and the quadratic scaling of attention with respect to K . With a single head, MHA-RAG matches the cost of xRAG, and for head counts $< K$, it is more computationally efficient than xRAG-K.

Order (In)variance. From Table 2, we observe that all baselines—RAG, xRAG, and xRAG-K—exhibit non-zero variance when the order of in-context exemplars or documents is shuffled. For RAG, this sensitivity is expected because exemplars are concatenated as text, and order directly influences the model’s input. In xRAG, order dependence arises from the positional encodings of the sentence-embedding encoder model: when all exemplars are jointly encoded into a single vector, reordering alters the resulting embedding. In xRAG-K, although each exemplar is encoded independently, the position of each resulting vector within the soft prompt is determined by exemplar order, leading to downstream variance. In contrast, our proposed MHA-RAG is order-invariant by design, as formally established in Appendix A.1.

Findings. These results demonstrate that by spending an upfront cost for training, MHA-RAG achieves both higher effective accuracy than baselines and lower inference cost compared to RAG.

Benchmarks	RAG	xRAG	xRAG-K	Ours
BACE	3.18	2.80	1.12	0.0
BBBP	2.25	3.02	1.27	0.0
ClinTox	8.52	0.12	4.91	0.0
PubMedQA	4.30	0.61	2.00	0.0

Table 2: Order-(in)variance analysis for *Qwen3-4B* with $K = 5$: standard deviation in performance when exemplar order is randomized across 5 seeded shuffles. (Lower numbers are better; zero means order-invariant.) **Bold** shows the lowest standard deviation achieved.

4.3 CONTEXT SATURATION WITH SOFT PROMPTS

In this section, we examine how effective accuracy varies with the number of retrieved exemplars (K), focusing on context saturation (Vladika & Matthes, 2025), the point at which adding more context introduces noise and causes effective accuracy to plateau or decline.

As shown in Table 3, increasing K generally improves effective accuracy for both RAG ($K = 1 \rightarrow 5 \rightarrow 10$) and MHA-RAG ($K = 1 \rightarrow 5$), reflecting the benefit of richer context. A key finding is that MHA-RAG at $K=5$ generally outperforms RAG at both $K=5$ and $K=10$, indicating that it is more effective at extracting and representing information that is sufficient to answer the questions (Joren et al., 2024). Moreover, it achieves this performance with fewer FLOPs than RAG (See Figure 2).

Benchmarks	RAG			MHA-RAG		
	$K=1$	$K=5$	$K=10$	$K=1$	$K=5$	$K=10$
<i>Qwen3-4B</i>						
BACE	58.50	<u>75.87</u>	<u>75.87</u>	59.89	76.27	52.42
BBBP	67.50	<u>69.33</u>	68.01	72.12	82.49	80.43
ClinTox	30.94	44.24	<u>54.48</u>	63.19	96.32	89.44
PubMedQA	52.68	79.26	80.34	55.33	<u>76.59</u>	66.07
<i>Llama3.2-3B-Instruct</i>						
BACE	<u>51.71</u>	49.89	36.36	52.37	67.62	54.73
BBBP	35.81	46.15	<u>52.08</u>	71.78	87.61	86.80
ClinTox	38.59	37.90	<u>47.27</u>	64.95	94.44	89.33
PubMedQA	54.59	71.38	<u>73.53</u>	57.91	76.34	72.50

Table 3: Effective accuracy of MHA-RAG vs. RAG under varying K . Results are averaged over 3 random seeds. **Bold** indicates the best performance. Underlined indicates the value of K that achieves the best effective accuracy for a given method.

We further observe that MHA-RAG reaches context saturation earlier than RAG: effective accuracy peaks around $K=5$, whereas RAG continues to improve up to $K=10$. Beyond these points, additional exemplars degrade effective accuracy, likely due to the inclusion of less relevant or noisy samples, consistent with findings of Liu et al. (2023); Hsieh et al. (2024); Zhao (2023).

Findings. MHA-RAG enables more effective domain adaptation with fewer exemplars over RAG.

4.4 EFFECTIVE ACCURACY AS A FUNCTION OF ATTENTION HEADS

Having examined effective-accuracy variation with K , we now study the impact of the number of heads. From Figures 3a and 3b, we observe that when $K = 1$, increasing the number of heads does not improve effective accuracy, likely due to the limited context—only a single exemplar or document—offering little room for multiple heads to learn diverse representations. In contrast, at $K = 5$, increasing the number of heads generally leads to higher effective accuracy across benchmarks, suggesting that the benefit of multiple heads emerges only when sufficient context is available. Additional plots are provided in Appendix A.4.

Findings. MHA-RAG can effectively exploit additional context with an increasing number of heads.

4.5 COMPARISON WITH FINE-TUNING BASELINES

Because MHA-RAG requires some upfront training, we compare it against other fine-tuning methods. Specifically, we include LoRA (Hu et al., 2022) as a standard baseline for domain adaptation and evaluate against other soft-prompt fine-tuning methods—Prompt Tuning (PT) (Lester et al., 2021), Instance-Dependent Prompt Generation (IDPG) (Wu et al., 2022). As shown in Table 4, MHA-RAG achieves the highest average improvement of 47.45 in effective-accuracy over LoRA, with gains computed as the geometric mean across tasks. We next explain the performance gaps method by method, focusing on one type of task at a time.

Molecular-property-prediction tasks (BACE, BBBP, and ClinTox). As shown in Table 4, LoRA often overfits on limited-data BACE, BBBP, and ClinTox benchmarks, collapsing to predictions dominated by a single class. Further, on these tasks MHA-RAG outperforms both PT and IDPG, achieving $\Delta_{PT}^{\text{avg}} = 69.95$ and $\Delta_{IDPG}^{\text{avg}} = 7.01$. This gap can be explained by how soft prompts are constructed: PT learns a fixed, question-independent prompt shared across the entire task, while IDPG conditions its generated prompt on the input question, but ignores related training exemplars. MHA-RAG instead conditions the soft prompt on retrieved exemplars that are most relevant to the current query, allowing it to further leverage task-specific information. These results highlight the benefit of grounding adaptation in limited data settings with retrieved exemplars, rather than relying solely on parametric tuning.

Benchmarks	LoRA	Off-the-Shelf	PT	IDPG	MHA-RAG
<i>Qwen3-0.6B</i>					
BACE	0.0	36.67 ^{↑+36.67}	0.0	53.75 ^{↑+53.75}	<u>75.08</u> ^{↑+75.08}
BBBP	0.0	22.21 ^{↑+22.21}	0.0	86.52 ^{↑+86.52}	<u>87.82</u> ^{↑+87.82}
ClinTox	0.0	0.0	0.0	94.87 ^{↑+94.87}	<u>97.12</u> ^{↑+97.12}
PubMedQA	<u>73.94</u>	13.96 ^{↓-59.98}	38.99 ^{↓-34.95}	45.50 ^{↓-28.44}	66.52 ^{↓-7.42}
<i>Qwen3-4B</i>					
BACE	12.32	0.0 ^{↓-12.32}	0.0 ^{↓-12.32}	62.54 ^{↑50.22}	76.27 ^{↑-7.42}
BBBP	19.61	10.88 ^{↓-8.73}	11.10 ^{↓-8.51}	<u>87.14</u> ^{↑67.53}	82.49 ^{↑63.95}
ClinTox	0.0	0.0	0.0	94.36 ^{↑+94.36}	<u>96.32</u> ^{↑+96.32}
PubMedQA	78.31	53.65 ^{↓-24.66}	81.04 ^{↑+2.73}	<u>83.65</u> ^{↑+5.34}	76.59 ^{↓-1.72}
<i>Llama3.2-3B-Instruct</i>					
BACE	0.0	0.0	13.61 ^{↑+13.61}	35.28 ^{↑+35.28}	<u>67.62</u> ^{↑+67.62}
BBBP	54.77	0.0 ^{↓-54.77}	36.89 ^{↓-17.88}	87.14 ^{↑+32.37}	<u>87.61</u> ^{↑+32.84}
ClinTox	31.62	0.0 ^{↓-31.62}	61.88 ^{↑+30.26}	<u>94.87</u> ^{↑+63.25}	94.44 ^{↑+62.82}
PubMedQA	82.15	56.73 ^{↓-25.42}	64.68 ^{↓-17.47}	<u>89.29</u> ^{↑+7.14}	76.34 ^{↓-5.81}
$\Delta_{\text{LoRA}}^{\text{avg}}$		-17.96	-5.04	41.52	47.45

Table 4: Baseline comparison with other PEFT methods. Performance is reported as effective accuracy—i.e, the geometric mean of the True-Positive and True-Negative rates ([↑]: improvement relative to LoRA, [↓]: drop relative to LoRA). Random guessing yields an effective accuracy of 50. Hyperparameters are tuned via sweeps: Prompt Tuning (virtual tokens $\in \{1, 5, 10\}$), LoRA (rank $\in \{16, 32, 64\}$), and MHA-RAG ($H \in \{1, 2, 4, 8\}$). Underlined indicates the best effective accuracy.

Question Answering in PubMedQA. Unlike molecular-property-prediction tasks, answering questions in PubMedQA requires access to supporting documents. To ensure comparability, fine-tuning baselines are provided with golden documents as additional context sufficient for answering each question (Joren et al., 2024), presented directly to the foundation model during training and inference. In contrast, MHA-RAG encodes these documents into soft-prompt vectors via its multi-headed attention, rather than appending them directly to the model’s input. We also include a comparison with Off-the-Shelf model, where the foundation model is queried without access to these documents, serving as a measure of how much parametric knowledge alone supports question answering. MHA-RAG improves over Off-the-Shelf by an average effective-accuracy gain of 30.91, indicating that its soft prompts effectively capture and represent knowledge from golden documents. Lastly, relative to LoRA, MHA-RAG shows an average effective-accuracy drop of -5.01 , computed as the geometric mean of drops in PubMedQA. This is expected given that it compresses documents into at most eight soft-prompt tokens. However, inference with MHA-RAG in PubMedQA is computationally cheaper than LoRA, PT and IDPG, because the latter fine-tuning baselines incur inference costs at least as high as RAG (see Figure 2) when documents are pre-pended to the model input.

Findings. In domains with limited data, MHA-RAG is an effective method for adapting foundation models; it also efficiently compresses supporting documents into compact soft-prompt tokens.

5 CONCLUSION

In this paper, we propose MHA-RAG, an attention-based method inspired by RAG that utilize in-context exemplars to enhance performance. Our approach combines soft-prompt-based domain adaptation with an order-invariant architecture to reduce inference cost and stabilize performance. We also provide the number of heads in the architecture as a tunable parameter to control soft-prompt generation across different experimental setups. With a more efficient, effective, and consistent way of representing in-context exemplars, this work aims to position MHA-RAG as an alternative to RAG for adapting language models to domains. In future, we want to study how well MHA-RAG scales to tasks requiring inference with longer documents and how robust it is to ‘lost-in-the-middle’ problem (Liu et al., 2023), a common issue in RAG that appears with scaling..

REPRODUCIBILITY STATEMENT

We provide details of experimental settings, hyperparameters, and datasets in Section 4.1. Proofs of theoretical results are included in Appendix A.1. All source code and data processing scripts will be made publicly available with the camera-ready submission.

REFERENCES

- Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.
- Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. *arXiv preprint arXiv:2205.11961*, 2022.
- Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.
- Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one token. *Advances in Neural Information Processing Systems*, 37:109487–109516, 2024.
- Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. *arXiv preprint arXiv:2305.14788*, 2023.
- Eunseong Choi, Sunkyoung Lee, Minjin Choi, June Park, and Jongwuk Lee. From reading to compressing: Exploring the multi-document reader for prompt compression. *arXiv preprint arXiv:2410.04139*, 2024.
- Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. Pretraining context compressor for large language models with embedding-based memory. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28715–28732, 2025.
- Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. Attentionrag: Attention-guided context pruning in retrieval-augmented generation. *arXiv preprint arXiv:2503.10720*, 2025.
- Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*, 2023.
- Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.
- Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*, 2024.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.
- Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

- Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long T Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, et al. Found in the middle: Calibrating positional attention bias improves long context utilization. *arXiv preprint arXiv:2406.16008*, 2024.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C Park. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. *arXiv preprint arXiv:2412.12559*, 2024.
- Abhinav Jain, Swarat Chaudhuri, Thomas Reps, and Chris Jermaine. Prompt tuning strikes back: Customizing foundation models with low-rank prompt adaptation. *Advances in Neural Information Processing Systems*, 37:47297–47316, 2024a.
- Abhinav Jain, Chris Jermaine, and Vaibhav Unhelkar. Rag-modulo: Solving sequential tasks using experience, critics, and language models. *arXiv preprint arXiv:2409.12294*, 2024b.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmllingua: Compressing prompts for accelerated inference of large language models. *arXiv preprint arXiv:2310.05736*, 2023.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019.
- Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. *arXiv preprint arXiv:2411.06037*, 2024.
- Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.
- Dongfang Li, Zetian Sun, Xinshuo Hu, Baotian Hu, and Min Zhang. Cmt: A memory compression method for continual knowledge learning of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24413–24421, 2025.
- Zongqian Li, Yixuan Su, and Nigel Collier. 500xcompressor: Generalized prompt compression for large language models. *arXiv preprint arXiv:2408.03094*, 2024.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranajpe, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.
- Maxime Louis, Hervé Déjean, and Stéphane Clinchant. Pisco: Pretty simple compression for retrieval-augmented generation. *arXiv preprint arXiv:2501.16075*, 2025.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022.

- Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.
- Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.
- Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36:19327–19352, 2023.
- Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.
- Bhagyajit Pingua, Adyakanta Sahoo, Meenakshi Kandpal, Deepak Murmu, Jyotirmayee Rautaray, Rabindra Kumar Barik, and Manob Jyoti Saikia. Medical llms: Fine-tuning vs. retrieval-augmented generation. *Bioengineering*, 12(7):687, 2025.
- David Rau, Shuai Wang, Hervé Déjean, and Stéphane Clinchant. Context embeddings for efficient answer generation in rag. *arXiv preprint arXiv:2407.09252*, 2024.
- Jiayi Sheng, Luna Lyu, Jikai Jin, Tony Xia, Alex Gu, James Zou, and Pan Lu. Solving inequality proofs with large language models. *arXiv preprint arXiv:2506.07927*, 2025.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Yuqiao Tan, Shizhu He, Huanxuan Liao, Jun Zhao, and Kang Liu. Dynamic parametric retrieval augmented generation for test-time knowledge enhancement. *arXiv preprint arXiv:2503.23895*, 2025.
- Jiwei Tang, Jin Xu, Tingwei Lu, Zhicheng Zhang, Yiming Zhao, Lin Hai, and Hai-Tao Zheng. Perception compressor: A training-free prompt compression framework in long context scenarios. *arXiv preprint arXiv:2409.19272*, 2024.
- Taffee T Tanimoto. An elementary mathematical theory of classification and prediction. 1958.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Juraj Vladika and Florian Matthes. On the influence of context size and model choice in retrieval-augmented generation systems. *arXiv preprint arXiv:2502.14759*, 2025.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, VG Vydiswaran, and Hao Ma. Idpg: An instance-dependent prompt generation method. *arXiv preprint arXiv:2204.04497*, 2022.
- Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.
- Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. *arXiv preprint arXiv:2310.04408*, 2023.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. *arXiv preprint arXiv:2407.09014*, 2024.

- Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey. In *CCF Conference on Big Data*, pp. 102–120. Springer, 2024.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.
- Jiachen Zhao. In-context exemplars as clues to retrieving from large associative memory. *arXiv preprint arXiv:2311.03498*, 2023.
- Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. *arXiv preprint arXiv:2309.03882*, 2023.

A APPENDIX

A.1 MHA-RAG: IN-CONTEXT EXEMPLAR-ORDER INVARIANCE

In this section, we argue that the soft prompt generated by MHA-RAG is invariant to the order of in-context exemplars/documents.

Claim 1 (Exemplar-Order Invariance). *Let $\mathbf{q}$ be a query vector for a given x and for $k = 1 \dots K$, let $\mathbf{k}_k \in \mathcal{R}^d$ and $\mathbf{v}_k \in \mathcal{R}^d$ be keys and value vectors derived from each retrieved example. Define the attention weights as $\alpha_k := \frac{e^{s_k}}{\sum_{j=1}^K e^{s_j}}$ where score, $s_k := \frac{\mathbf{q} \cdot \mathbf{k}_k}{\sqrt{d}}$. The attention output for each head is given by $\mathbf{z} = \sum_{k=1}^K \alpha_k \mathbf{v}_k$. Then $\mathbf{z}$ is invariant under any permutation of the retrieved examples: permuting the indexing of the pairs $\{(\mathbf{k}_k, \mathbf{v}_k)\}_{k=1}^K$ does not change $\mathbf{z}$.*

Proof. Let π be a permutation of $1 \dots K$ and consider the permuted sequence $(\mathbf{k}_{\pi(1)}, \mathbf{v}_{\pi(1)}), \dots, (\mathbf{k}_{\pi(K)}, \mathbf{v}_{\pi(K)})$. For the permuted sequence, the scores and weights are

$$s'_k = \frac{\mathbf{q} \cdot \mathbf{k}_{\pi(k)}}{\sqrt{d}} = s_{\pi(k)} \quad \alpha'_k = \frac{e^{s'_k}}{\sum_{j=1}^K e^{s'_j}} = \frac{e^{s_{\pi(k)}}}{\sum_{j=1}^K e^{s_{\pi(j)}}}$$

Because $\{s_{\pi(j)} : j = 1, \dots, K\}$ is a reordering of $\{s_j : j = 1, \dots, K\}$, we have $\sum_{j=1}^K e^{s_{\pi(j)}} = \sum_{j=1}^K e^{s_j}$. Therefore, $\alpha'_k = \alpha_{\pi(k)}$.

Now, the head's resulting output under the permuted attention is $\mathbf{z}' = \sum_{k=1}^K \alpha'_k \mathbf{v}_{\pi(k)} = \sum_{k=1}^K \alpha_{\pi(k)} \mathbf{v}_{\pi(k)}$. Without loss of generality, reindex the sum by setting $j = \pi(k)$. Because π is a bijection, $k \mapsto j$ permutes the index set $\{1, \dots, K\}$, resulting in $\mathbf{z}' = \sum_{j=1}^K \alpha_j \mathbf{v}_j = \mathbf{z}$.

Thus $\mathbf{z}$ is unchanged by the permutation π . The result holds for any permutation, so the MHA's per-head output is order invariant. $\square$

A.2 BENCHMARKING ON MATH REASONING

In this section, we investigate whether MHA-RAG can improve performance on mathematical reasoning tasks. Specifically, we evaluate on *Inequality Math* [train/dev split: 1252/100], an Olympiad-level benchmark that requires proving bounds preserving inequalities and establishing relations between algebraic expressions (Sheng et al., 2025). For retrieval, we compute dense representations of question-exemplar pairs using *Qwen3-Embedding-8B* (Zhang et al., 2025) and rank candidates by cosine similarity.

We report results primarily on *Llama3.2-3B-Instruct*, since models in the Qwen3 family already achieve high off-the-shelf accuracy (e.g., *Qwen3-4B* at 65%), where adding retrieved exemplars led to performance drops. This observation is consistent with Sheng et al. (2025), who found that only certain model families benefit from in-domain exemplars. A plausible explanation is that high-performing models may have already been exposed to mathematical reasoning tasks during pre-training, reducing the marginal utility of retrieval.

IneqMath	Off-The-Shelf	RAG	xRAG	xRAG-K	MHA-RAG
<i>Llama3.2-3B-Instruct</i>					
Bound Accuracy	10.0	10.0	4.0	8.0	14.0
Relation Accuracy	22.00	24.0	24.0	24.0	22.0
Final Accuracy	16.00	17.0	14.0	16.0	18.0

Table 5: Baseline comparison on *IneqMath* requiring step-by-step reasoning (with $K = 2$). Accuracy is measured as an exact match of derived numerical values with ground-truth. Hyperparameters are tuned via sweeps: MHA-RAG ($H \in \{1, 2, 4\}$). **Bold** indicates the best performance. The model is trained for 1 epoch with a learning rate of $3e - 4$.

From Table 5, we observe that MHA-RAG yields a slight improvement over both off-the-shelf and RAG, whereas training with xRAG and xRAG-K results in a performance drop. A possible explanation is that the *Inequality Math* dataset contains multiple problems that rely on the same theorems or follow similar reasoning steps. If retrieval fails to surface such structurally related exemplars, the model cannot fully benefit from exemplar conditioning. Future work could therefore focus on improving retrieval quality—e.g., by incorporating reasoning-aware similarity metrics—which may allow MHA-RAG to better exploit shared problem structure and yield stronger overall gains.

A.3 DOES ADDING EXEMPLARS BACK HELP?

Benchmarks	$K=5$			$K=10$		
	$c=0$	$c=1$	$c=5$	$c=0$	$c=1$	$c=5$
<i>Qwen3-4B</i>						
BACE	76.27	75.27	<u>79.03</u>	52.42	76.27	79.19
BBBP	82.49	68.56	88.14	80.43	70.01	<u>87.5</u>
ClinTox	<u>96.32</u>	76.63	94.55	89.44	70.72	99.04
PubMedQA	76.59	70.87	83.02	66.07	72.88	<u>82.92</u>
<i>Llama3.2-3B-Instruct</i>						
BACE	66.16	79.96	78.82	54.73	74.23	<u>77.75</u>
BBBP	87.61	88.99	89.74	86.80	85.44	<u>86.94</u>
ClinTox	94.44	88.97	98.40	89.33	<u>94.49</u>	88.97
PubMedQA	76.34	76.56	<u>83.17</u>	72.50	76.45	83.37

Table 6: Effect of re-inserting top- c exemplars in textual form (up to $c = 5$, given a fixed inference budget) into the in-context prompt while still using top- K for soft-prompt computation in MHA-RAG. The $K = 5$, $c = 0$ column corresponds to the MHA-RAG column of Table 1. **Bold** indicates the best effective accuracy. Underlined indicates the value of c that achieves the best effective accuracy for a given K .

To investigate whether performance can be further improved, we augment MHA-RAG’s soft prompts with a small number of exemplars directly included in the model’s context at inference. Specifically, we add the top- c retrieved exemplars alongside the soft prompts and study the effect across different K values.

As shown in Table 6, effective accuracy improves consistently with increasing c , with the best results achieved at $c = 5$ for both $K = 5$ and $K = 10$. The FLOPs analysis in Figure 4 reveals that increasing K from 5 to 10 incurs only a minor cost, while increasing c from 0 to 5 leads to a logarithmic rise in FLOPs. These results suggest that c can serve as a tunable knob for balancing accuracy gains against inference cost, allowing practitioners to adapt MHA-RAG to different computational budgets.

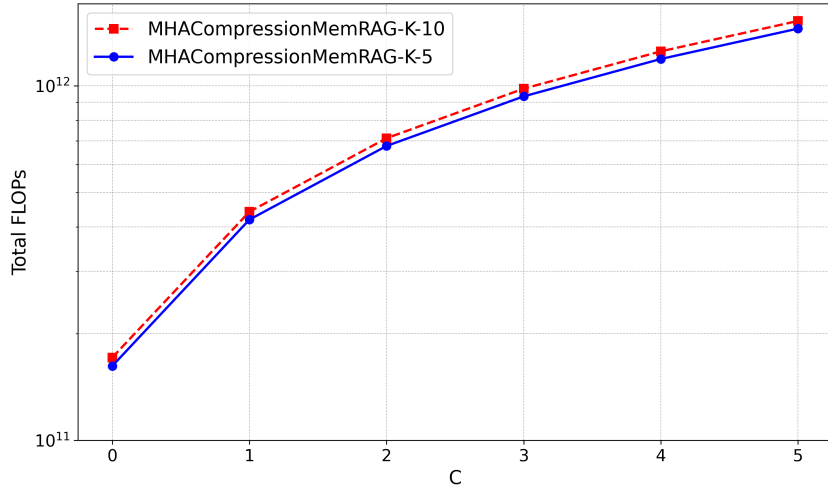


Figure 4: Total FLOPs for inference with encoder ChemBERTa-10M-MTR and foundation model Qwen3-4B using an increasing number of exemplars c in the context, given $K = 5$ and $K = 10$ as input to create a soft prompt.

A.4 MHA-RAG: ABLATION ON NUMBER OF HEADS

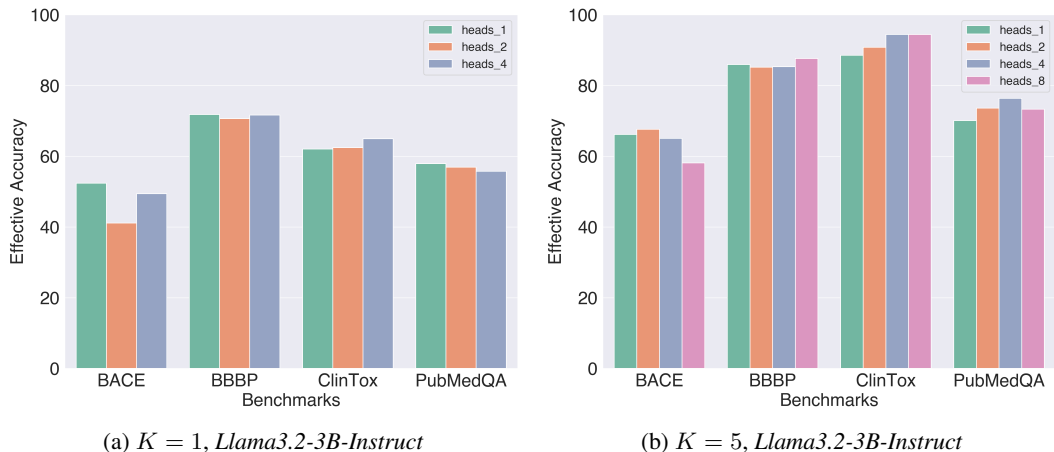


Figure 5: Varying number of heads in MHA-RAG. Performance averaged across 3 seeded runs.

A.5 TRAINABLE PARAMETERS

Training Methods	Qwen3-0.6B	Qwen3-4B	Llama3.2-3B-Instruct
<i>Retrieval-Based Baselines</i>			
xRAG	1.59M	2.19M	2.38M
xRAG- K	1.59M	2.19M	2.38M
MHA-RAG ($m = 1$)	1.38M	4.16M	4.75M
MHA-RAG ($m = 2$)	3.57M	7.11M	8.3M
MHA-RAG ($m = 4$)	5.93M	13.03M	15.39M
<i>PEFT Baselines</i>			
PT ($m = 10$)	10.24K	25.6K	30.72K
IDPG ($m = 1$)	1.59M	2.19M	2.38M
LoRA ($r = 64$)	40.37M	132.12M	97.26M

Table 7: Number of trainable parameters when a given foundation model is updated with a specific baseline approach. For retrieval-based baselines, we report overall trainable parameters with *ChemBERTa-2-10M-MTR* as the Encoder.