Imagination Augmented Generation: Learning to Imagine Richer Context for Question Answering over Large Language Models

Huanxuan Liao[1,2], **Shizhu He**[1,2], **Yao Xu**[1,2], **Yuanzhe Zhang**[1,2], **Kang Liu**[1,2], **Shengping Liu**[3], **Jun Zhao**[1,2]
[1] The Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3] Unisound, Beijing, China
liaohuanxuan2023@ia.ac.cn {yao.xu, shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Retrieval-Augmented-Generation and *Generation-Augmented-Generation* have been proposed to enhance the knowledge required for question answering over Large Language Models (LLMs). However, the former depends on external resources, and both require incorporating the explicit documents into the context, which results in longer contexts that lead to more resource consumption. Recent works indicate that LLMs have modeled rich knowledge, albeit not effectively triggered or activated. Inspired by this, we propose a novel knowledge-augmented framework, **Imagination-Augmented-Generation** (IAG), which simulates the human capacity to compensate for knowledge deficits while answering questions solely through imagination, without relying on external resources. Guided by IAG, we propose an **im**agine richer **c**ontext method for **q**uestion **a**nswering (IMcQA), which obtains richer context through the following two modules: *explicit imagination* by generating a short dummy document with long context compress and *implicit imagination* with HyperNetwork for generating adapter weights. Experimental results on three datasets demonstrate that IMcQA exhibits significant advantages in both open-domain and closed-book settings, as well as in both in-distribution performance and out-of-distribution generalizations [1].

1 Introduction

Knowledge-intensive tasks like question answering (QA) necessitate access to extensive world and domain knowledge (Berant et al., 2013; Joshi et al., 2017; Kwiatkowski et al., 2019). Recently, Large Language Models (LLMs) have displayed notable competencies in almost every task and industry within the "pre-train, prompt, and predict" paradigm (Liu et al., 2023b). However, LLMs lack the sufficient capability to independently handle knowledge-intensive tasks (Yu et al., 2023) and usually generate hallucinations (Zhao et al., 2023).

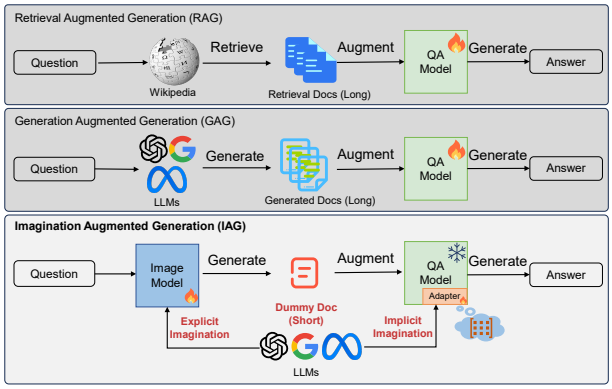

Figure 1: Compared with RAG (Top) and GAG (Middle), the proposed IAG (Bottom) eschews external resources, utilizing solely LLMs to imagine a shorter explicit document and more flexible implicit adapters.

In recent years, to alleviate the issue of hallucinations in LLMs and improve performance in knowledge-intensive tasks such as QA, researchers have proposed numerous knowledge-augmented methods for LLMs, which mainly include two categories: *Retrieval-Augmented-Generation* (RAG) (Guu et al., 2020) and *Generation-Augmented-Generation* (GAG) (Abdallah and Jatowt, 2023). RAG (Top part of Figure 1) retrieves related documents from external resources (e.g., auxiliary tools and domain documents) and then sends those retrieved documents and the question together into LLMs (Izacard and Grave, 2021). RAG has demonstrated formidable performance on several tasks (Lewis et al., 2020). However, RAG not only requires presupposed external resources but also necessitates more computational resources and longer processing times (Xu et al., 2023a). Take typical RAG method FiD (Izacard and Grave, 2021) as an example, the required computing resources and inference time continue to increase as the number of retrieved documents increases. While retrieving 100 documents requires processing over 12k

[1] Our code will be available at https://github.com/Xnhyacinth/IAG

tokens, it will result in an exceeding $100\times$ prompt length (decades times of GPU consumption) and an over $100^2\times$ computational time (Liu et al., 2023a).

To avoid dependence on external resources, GAG (Yu et al., 2023) has been proposed and can utilize LLMs like InstructGPT (Ouyang et al., 2022) to generate the relevant documents (Middle part of Figure 1). However, it requires additional financial costs (e.g., API calls) and still demands substantial computational resources and time. Additionally, both RAG and GAG utilize more explicit external resources (symbolized documents), and the quality of the obtained content significantly impacts downstream tasks (Li et al., 2023; Shaier et al., 2024). For example, Gao et al. (2024) indicates that noise in the documents negatively affects the performance. Therefore, there is an urgent need to explore new knowledge-augmented methods.

In fact, LLMs inherently contain rich knowledge and possess significant potential for resolving knowledge-intensive tasks (Bhagavatula et al., 2020). Enhancing the performance of specific tasks can be achieved by better activating relevant knowledge or expanding memory capacity without relying on external resources. For example, simply repeating the question twice (Xu et al., 2023b), just reviewing and consolidating knowledge by appending a straightforward prompt "As far as I know" (Yao et al., 2023), and using visual-language models to imagine images (Tang et al., 2023), all approaches enhance the performance of LLMs on downstream tasks. Inspired by this, we introduce a novel knowledge-augmented framework *Imagination-Augmented-Generation* (IAG) for LLMs, which simulates the human capacity to compensate for knowledge deficits solely through imagination in QA. As shown in the bottom part of Figure 1, for resolving knowledge-intensive tasks, IAG utilizes solely LLMs to imagine a shorter explicit document and more flexible implicit adapters.

Within the framework of IAG, we introduce an **im**agine richer **c**ontext method for **q**uestion **a**nswering (IMcQA). To sufficiently utilize the inherent knowledge of LLMs, we design two main modules to activate the various potential knowledge modeling in LLMs and obtain a richer context. Specifically, the **explicit imagination** module first uses symbol distillation to obtain the compressed context and then guides LLMs in generating a short and useful dummy document. Subsequently, the **implicit imagination** module utilizes the proposed HyperNetwork to generate LoRA weights to activate the task-processing ability of LLMs. Unlike the LoRA (Hu et al., 2021) stores task knowledge and ability in modules, the HyperNetwork learns to imagine hidden knowledge for each question.

We evaluate the proposed IMcQA to various LLMs, including T5 (Roberts et al., 2020a) and Llama2 (Touvron et al., 2023). The experimental results across three QA datasets indicate that the proposed method yields performance gains while reducing computational expenses and time. Notably, it even outperforms baseline methods that retrieve and generate knowledge under the same document settings. In conclusion, the contributions of this paper are summarized as follows:

- We propose a new knowledge augmentation framework IAG to fully leverage the LLMs' intrinsic knowledge more efficiently without relying on external resources.
- We propose a novel QA method IMcQA that employs two main modules (explicit imagination and implicit imagination) to better utilize the knowledge stored in the LLMs and obtain richer context in QA.
- Experimental results indicate that the proposed method successfully activates the relevant internal knowledge of LLMs. IMcQA exhibits significant advantages in both open-domain and closed-book settings, as well as in both in-distribution performance and out-of-distribution generalizations.

2 Related Work

This paper mainly utilizes context compression, hypernetworks and knowledge distillation to achieve knowledge enhancement. The following will elucidate pertinent research across four facets.

Knowledge Enhancement has usually been adopted to alleviate the issue of insufficient knowledge in LLMs. There are two main methods: RAG (Sun et al., 2019) and GAG (Abdallah and Jatowt, 2023). The typical RAG method FiD (Izacard and Grave, 2021) retrieves the documents from an external knowledge base to answer questions. As LLMs can be considered a knowledge base, several studies (Liu et al., 2022) propose to extract knowledge from LLMs (e.g., GPT-3). For example, Yu et al. (2023) generates 10 relevant documents for world knowledge according to the question. However, RAG needs the related external resources, and

both RAG and GAG still need to obtain and utilize verbose explicit long contexts. Recently, there have been some methods to enhance the LLMs' ability through simulating human imagination of visual information. But they use existing visual-language models (Tang et al., 2023; Akter et al., 2024), while we prefer self-imagination to augment knowledge. Besides, they have not fully leveraged the parameterized knowledge within the models (Xu et al., 2023b; Kazemnejad et al., 2023). In this paper, the proposed method for augmenting knowledge not only obviates the need for external resources but also enhances the efficiency of extracting and activating internal knowledge within LLMs.

Context Compression has often been used to improve the efficiency of LLMs in processing long contexts. Recent studies (Mu et al., 2023) propose that long contexts be condensed into summary vectors (soft prompts) to ensure their effective utilization by LLMs. Simultaneously, some studies (Jiang et al., 2023) believe that information redundancy in lengthy texts and information entropy can be utilized to compress the contexts (Li et al., 2023). Unlike them, this paper is devoted to awakening the long-context modeling ability of LLMs. By learning an Imagine Model that can generate compressed contexts, the QA model that operates on short contexts can also possess a rich contextual understanding akin to the QA model designed for processing longer contexts.

Knowledge Distillation is a technique where a smaller model learns to mimic the predictions of a larger model, aiming to retain performance while reducing computational resources (Hinton et al., 2015). Recent studies (West et al., 2022) present symbolic knowledge distillation, a process that facilitates knowledge transfer from a teacher model via extracting training data to subsequently train a student model. In this paper, the process of obtaining compressed context during the explicit imagination resembles a form of symbolic distillation. Regarding training, our emphasis lies in distilling the long-context modeling abilities of LLMs.

Hypernetworks is designed to reduce the number of parameters (Ha et al., 2016), i.e., a small neural network generates parameters for another big neural network. Recent studies (Phang et al., 2022; Ivison et al., 2023) have explored the enhancement of model performance in zero- and few-shot settings through meta-learning involving hypernetworks. We utilize hypernetworks to acquire implicit imagine capabilities by dynamically generating LoRA for efficiency and generalization.

3 Method

In this section, we introduce the detailed method of IMcQA to activate LLMs' intrinsic knowledge and obtain a richer context for QA. The fundamental premise underlying this method is that QA with a richer context yields greater performance. Consequently, diverse methods are employed for questions lacking in richer contexts to activate knowledge within LLMs to replicate comparable effects to those achieved with richer contexts.

Specifically, IMcQA comprises two main modules. Explicit imagination with long context compression learns to imagine a short dummy document (§ 3.2). And implicit imagination with the HyperNetwork models hidden knowledge that learns a shared knowledge feature projection across questions (§ 3.3). The HyperNetwork is trained to generate lightweight LoRA modules, aiming to align the question and the internal knowledge. Besides, there is long context distillation in training, which learns the teacher's rich representations to compensate for missing knowledge in imagination (§ 3.4).

3.1 Formulation

The formulation of our task follows RAG for QA (Guu et al., 2020). Let $\mathcal{V}^*$ denote the infinite set, encompassing all potential strings over the tokens in vocabulary $\mathcal{V}$, and this includes the empty string. An instance within a QA dataset is defined as a triplet (q, a, c) comprising question q, answer a, and context c, where $q, a, c \in \mathcal{V}^*$. Conventionally, the context c is drawn from the knowledge corpus $\mathcal{Z}$, like Wikipedia, whereby $\mathcal{Z} \subset \mathcal{V}^*$.

The goal of QA is to learn a distribution function $p(a|q)$. In a closed-book setting, LLMs directly encode the given question q and generate the answer a (Roberts et al., 2020b). However, employing a direct approach of requesting models to output answers frequently results in poor performance, primarily attributable to the omission of a substantial amount of world knowledge. Therefore, a popular approach is the open domain setting, which marginalizes $p(a|q, c)$ over contexts c. Additional background details are available in A.1.

3.2 Explicit Imagination with Compress

To get the context c, we utilize LLMs to imagine a short dummy document, which can mitigate *knowl-*

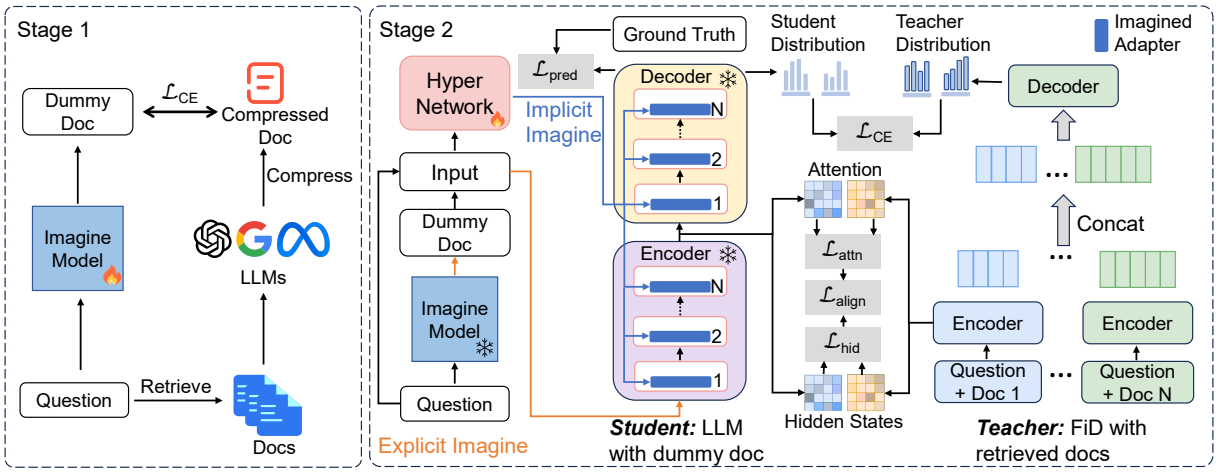

Figure 2: Overview of IMcQA method. In the inference phase, for each question, the Imagine Model imagines a short dummy document based on the question and the HyperNetwork generates specific LoRA weights. During training, there are two stages: the first stage is the **pre-training** of the **Imagine Model**, aiming at its ability to imagine a short dummy document based on the question, and the second stage is the **HyperNetwork fine-tuning** using long context distillation (§ 3.4) to learn a map from the question to the LoRA weights.

edge corpus error (Lee et al., 2023) by considering potentially useful contexts. In the view of compression, we greatly reduce input length, minimize noise, and elevate the salience of essential tokens.

As shown in the left part of Figure 2, to help LLMs fully utilize the knowledge and imagine compressed text, we first pretrain it on our collected question-compressed document pairs. By leveraging symbolic distillation, we employ the long-context compression method LongLLMLingua (Jiang et al., 2023) to condense a large corpus of retrieved documents. These compressed texts c' then serve as fine-tuning data alongside specific prompts p_q (A.2) and question-answer pairs for the **Imagine Model** G_θ (θ represents the model's parameter), which guides the model to think about its knowledge and imagine a short dummy document:

$$d = G_\theta(p_q(q; c')) \quad (1)$$

where d is the dummy document generated from the **Imagine Model**. This process enables LLMs to conceive compressed knowledge that robustly parallels the question's knowledge requirements.

3.3 Implicit Imagination with HyperNetwork

We advance upon LoRA (Hu et al., 2021) by suggesting the implementation of the HyperNetwork, which does not directly optimize the LoRA module but generates specific LoRA adapter weights using the inputs for QA (bottom part of Figure 2). This is akin to repeating the question in the prompt (Xu et al., 2023b) and incorporating certain topical cues

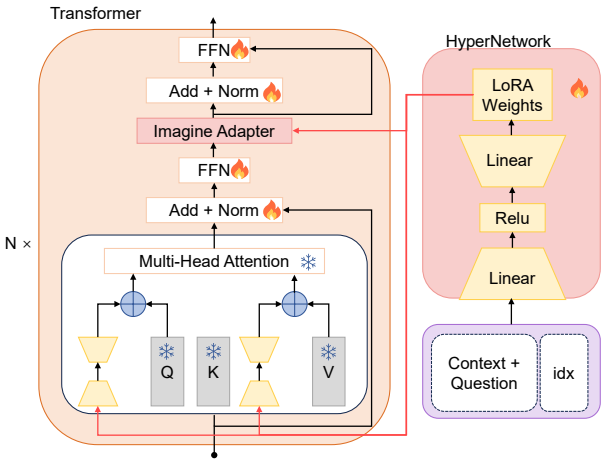

Figure 3: The Architecture of HyperNetwork. HyperNetwork generates LoRA adapter weights for each question. During training, only HyperNetwork, FFN, and Norm weights are updated.

to stimulate the model's recall of relevant questions (Wang et al., 2023). However, the distinction lies in the fact that they serve as wake-up features, whereas we are generating model parameters.

The HyperNetwork architecture for generating LoRA weights is exhaustively outlined in Figure 3. Specifically, D_k^q and U_k^q represent the low-rank projections of layer k correlated with the $\mathcal{Q}$, while D_k^v and U_k^v correspond to those associated with the $\mathcal{V}$. HyperNetwork represented as g_D and g_U, takes $concat(f, i_k^{\{q,v\}})$ as input, where f is the feature vectors that use the model's encoder to obtain and using whitening algorithm (Su, 2021) for dimen-

sionality reduction, $i_k^{\{q,v\}} \in \{0, \ldots, 2 \times \#\text{blocks}\}$ signifies the positional embedding differentiating between layers and between $\mathcal{QV}$. Each HyperNetwork is defined by weights W_d and W_u which represent the down and up projections respectively. Finally, the HyperNetwork equations for $D^{\{q,v\}}$ and $U^{\{q,v\}}$ can be expressed as:

$$f = \text{whitening}(\text{Encoder}(q; d)) \tag{2}$$

$$x = \text{concat}(f, i_k^{\{q,v\}} \mid i_k^q = 2k, i_k^v = 2k+1) \tag{3}$$

$$D^{\{q,v\}}, U^{\{q,v\}} = g_D(x), g_U(x) \tag{4}$$

where the **Encoder** represents the encoder of the model, **whitening** is a dimensionality reduction algorithm, and **concat** means to splice the content. g_D and g_U denote the descending and ascending dimensions of HyperNetwork. More formally,

$$g(x) = \text{MM}(\text{ReLU}(\text{MM}, W_d), W_u) \tag{5}$$

where MM stands for matrix multiplication, ReLU is a activation function.

3.4 Training with Long Context Distillation

Within the framework of knowledge distillation, components such as hidden representations (Jiao et al., 2020), attention dependencies (Wang et al., 2020), and relations among representations (Park et al., 2021) are regarded as valuable knowledge for transfer. In this paper, we consider long context distillation (LCD) as the contextualized knowledge that mainly guides the student.

Specifically, the teacher model FiD (Izacard and Grave, 2021), which utilizes longer contextual inputs and theoretically contains more information (richer context). It will activate more specific internal knowledge and serve as a supervisory model. The teacher model assists the student model T5 (Roberts et al., 2020a), which has the same size as the teacher and leverages short contextual inputs. This aids in activating richer feature representations and knowledge. The optimization objective for the student model at each mini-batch $z_r = (x_r, y_r)$ is:

$$\begin{aligned}\mathcal{L}_\text{s}(\theta_s, \theta_t, z_r) &= \alpha \mathcal{L}_\text{ce}(y_r, S(x_r; \theta_s)) \\ &+ (1-\alpha)\mathcal{L}_\text{ce}(T(x_r; \theta_t), S(x_r; \theta_s))\end{aligned} \tag{6}$$

where we have a teacher model denoted as $T(\cdot; \theta_t)$ and a student model denoted as $S(\cdot; \theta_s)$. The corresponding model parameters are θ_t and θ_s.

As shown in the right of Figure 2, we perform an additional representation alignment for better knowledge transfer. In our distillation, both teacher model and student model have the L layers, we feed the text into them and can obtain the corresponding output hidden states $\{H_l^t\}_{l=0}^L$, $\{H_l^s\}_{l=0}^L$, and attention matrices $\{A_l^t\}_{l=1}^L$, $\{A_l^s\}_{l=1}^L$. We suppose the student's l-th layer is aligned with the teacher's l-th layer, then the outputs of the student (i.e., H_l^s and A_l^s) should be close to the teacher's (i.e., H_l^t and A_l^t). For aligning hidden states, following (Park et al., 2021), we use cosine distance COS to calculate the proximity between the hidden states of the teacher and the student:

$$\mathcal{L}_\text{hid} = -\text{COS}(H_l^s, H_l^t) \tag{7}$$

While for aligning attention dependencies, we follow (Jiao et al., 2020) to optimize the mean square error (MSE) between the attention matrices of the teacher and the student:

$$\mathcal{L}_\text{attn} = -\text{MSE}(A_l^s, A_l^t) \tag{8}$$

The overall objective for knowledge transfer is:

$$\mathcal{L}_\text{align}(H_l^s, H_l^t, A_l^s, A_l^t) = \mathcal{L}_\text{attn} + \mathcal{L}_\text{hid} \tag{9}$$

The overall objective for training IMcQA is the weighted sum of the two objectives:

$$\mathcal{L} = \mathcal{L}_\text{s} + \lambda \mathcal{L}_\text{align} \tag{10}$$

4 Experiment

In this section, we conduct experiments to demonstrate the effectiveness and efficiency of IAG and IMcQA on QA. The experiment mainly answers four research questions (RQs):
RQ1: Can IAG achieve knowledge augmentation for answering questions over LLMs? (§ 4.4)
RQ2: Does our method have good knowledge activation and generalization abilities? (§ 4.5)
RQ3: Does IAG have advantages in effectiveness and efficiency compared to RAG and GAG? (§ 4.7)
RQ4: What is the role of explicit and implicit imagination modules in IAG and IMcQA? (§ 4.8)

4.1 Datasets

We evaluate the proposed approach on three public question answering datasets: NaturalQuestions (**NQ**) (Kwiatkowski et al., 2019), WebQuestions (**WQ**) (Berant et al., 2013) and TriviaQA (**TQA**) (Joshi et al., 2017). To evaluate the model performance, we use the exact match (EM) score for evaluating predicted answers (Rajpurkar et al., 2016). We provide dataset details in the Appendix A.4.

Models	Reader Params	# Documents	NQ	TriviaQA	WebQ
# closed-book setting					
T5-l (Roberts et al., 2020a)	770M	0	28.5*	28.7*	30.6*
T5-xl	3b	0	28.30	33.92	34.43
IMcQA-l (Ours)	770M	0	29.32	30.11	32.68
IMcQA-xl (Ours)	3b	0	**29.59**	**35.71**	**37.40**
# Retrieval Augmented Generation					
DPR (Karpukhin et al., 2020)	110M	100	41.5*	56.8*	41.1*
RAG (Lewis et al., 2020)	400M	10	44.5*	56.1*	45.2*
FiD-l (Izacard and Grave, 2021)	770M	10	46.7*	61.9*	48.1*
FiD-xl	3b	10	50.1*	66.3*	50.8*
# Generation Augmented Generation					
GENREAD-l (Yu et al., 2023)	770M	10†	40.3*	67.8*	51.5*
GENREAD-xl	3b	10†	42.6*	69.6*	52.6*
# Imagination Augmented Generation (Ours)					
IMcQA-l	770M	1†	42.32	65.48	45.28
IMcQA-xl	3b	1†	46.51	68.38	50.45
IMcQA-l	770M	10	49.92	69.67	51.52
IMcQA-xl	3b	5‡	**50.87**	**70.34**	**52.78**

Table 1: QA performances (%) of different methods on three datasets. The first part (closed-book setting) indicates that only utilize questions; The latter three parts utilize explicit documents. The best results are in bold. * means that those results are from existing papers, † denotes that the documents were generated (‡ indicates that the number of documents is reduced due to insufficient memory for distillation). More results can be seen in Appendix A.5.1.

4.2 Baselines

Both the moderately sized language model (< 1B) and the large language model (LLM) (≥ 3B) are under consideration. T5 (Roberts et al., 2020a) is selected as the backbone for our moderately sized language models. We evaluate our proposed IMcQA against several knowledge-enhanced approaches, which include RAG models such as DPR (Karpukhin et al., 2020), RAG (Lewis et al., 2020), and FiD (Izacard and Grave, 2021), as well as the GAG model GENREAD (Yu et al., 2023), and parameters efficient fine-tuning method LoRA (Hu et al., 2021). The Appendix A.3.2 provides further information about those baselines.

For the zero-shot settings of LLMs (≥ 3B), we use Llama2-7B and Llama2-13B (Touvron et al., 2023) as the basic model. We evaluate with four diverse settings: without retrieval, with retrieval, with LoRA, and using the proposed IMcQA.

4.3 Implementations

In the pretraining stage, the **Imagine Model** initialized with T5-large utilizes the generated question-compressed pairs. During the second stage, the teacher model employs a FiD reader with different sizes (FiD-l and FiD-xl) that are fine-tuned on the training split of target datasets. The student model freezes the backbone and updates solely the HyperNetwork, the feedforward neural network (FFN), and the normalization layers (Chen et al., 2023). More implementation details and experimental findings are in the Appendix A.3.

4.4 Main Results

Table 1 shows the performance results, full results including T5-Base are in the Appednidx A.5.1.

As shown in Table 1, when juxtaposed with **closed-book** models, **RAG**, and **GAG** methods, our proposed **IAG** framework IMcQA method exhibits state-of-the-art performance with the equivalent magnitude of document count and model size.

In the closed-book setting (in the upper part of the table), our method outperforms the baseline by an average of +2% EM score, indicating its excellence in utilizing internal knowledge with imagination. It's especially noteworthy that as the model size expands, the performance advantages of the imagination become ever more evident.

The following three parts of Table 1 show the experimental results under the open domain setting. Although our method only deals with one short

dummy document, it can still achieve results similar to or better than the RAG and GAG methods, which handle 10 documents. The findings reveal that IMcQA exploits imagined condensed text to strike a balance between efficiency and overhead. Moreover, when IMcQA utilizes 10 retrieved documents, it supersedes the performance of FiD and GENREAD with the EM average of +2.26% and +3.06%.

4.5 Out-Of-Distribution (OOD) Performance

To further demonstrate the generalizations of the IMcQA method and the importance of HyperNetwork, we also evaluate its performance in out-of-distribution (OOD) generalizations. Table 2 shows the IID and OOD performance of FiD, and IMcQA methods with different document settings when training on NQ (From NQ generalization to the other two datasets). Full OOD results of three datasets are shown in the Appendix A.5.2.

It is patently clear that an increment in document provision leads to better OOD performance, likely due to the presence of answer-oriented content within these documents. Remarkably, IMcQA can come within a relatively narrow 5% gap of FiD, even when utilizing a single imagined document as opposed to 10 retrieved ones.

Simultaneously, IMcQA generally showcases superior performance in OOD when provided with 10 retrieved documents. This superiority can be traced back to the pivotal role played by HyperNetwork in generating LoRA adapters' weights based on questions. This equips models with the capability to invoke and access internal knowledge based on context-specific discourse rather than confining to resolving distinct questions.

4.6 Zero-shot Results on LLMs

Figure 4 and Figure 5 illustrate the zero-shot results for LLMs implementing IMcQA. This research seeks to explore the possibility of enhancing LLMs via IAG. Due to the high computational demands of training, we only fine-tuned the HyperNetwork on a mixed dataset without LCD in this experiment and evaluated performance in a zero-shot setting. Detailed prompt information can be found in the Appendix A.2.

We discerned that Llama2's performance can be enhanced by imagining knowledge autonomously. While leveraging explicit imagined context could amplify the average EM +1%, this is not as significant as the improvement achieved by retrieving 10 documents, indicating the limitations of relying solely on prompt cues for triggering corresponding knowledge. IMcQA can enhance knowledge via two main imagination processes, escalating EM by +15.33% for NQ, +11.97% for TQA, and +16.38% for WQ. With IAG, Llama2-7B demonstrated an average improvement of +14% across the three datasets. This trend is also observed in Llama2-13B's results. This implies that even in zero-shot settings, our method can still offer substantial benefits to LLMs. More results can be seen in A.5.3.

Models	# Documents	NQ	TQA	WQ
T5	0	22.16	3.18	4.12
IMcQA	0	23.89	6.21	10.94
IMcQA	1†	40.14	46.61	18.92
FiD	10	46.81	53.93	24.02
IMcQA	10	**47.01**	**55.74**	**24.13**
T5-l	0	28.5*	3.18	4.12
IMcQA-l	0	29.32	10.17	14.06
IMcQA-l	1†	42.32	54.80	22.05
FiD-l	10	46.7*	57.93	25.12
IMcQA-l	10	**49.92**	**60.03**	**25.79**

Table 2: **OOD results**. The primary row in the table header delineates the dataset trained, while the underscored secondary row demonstrates the in-distribution performance (IID).

4.7 Training Cost and Inference Speed-up

We proceeded to measure the inference speed, documented in GPU time, and training time for 5000 steps on the NQ dataset, using T5-Base. The experiments were conducted on a single RTX 3090 GPU, maintaining a standard batch size of 8 during training and 1 during inference.

As evident from Table 3, the proposed method's advantage lies in its diminished requirement for parameter updates, which can be attributed to the shared HyperNetwork's utilization that generates LoRA adapters, thereby negating the necessity of

Models	Training Params	# Documents	# Avg Tokens	Inference Time	GPU Memory
T5	220M	**0**	**19.8**	**79.8s**	2828M
IMcQA	**139.3M**	**0**	**19.8**	82.3s	**2710M**
IMcQA	**139.3M**	1	522.1	214.6s	2882M
FiD	220M	10	1748.3	683.3s	4358M
GENREAD	220M	10	1912.5	704.8s	4412M
FiD	220M	100	16625.7	1293.2s	19048M

Table 3: Training and inference cost on the NQ. The backbone model is T5-Base.

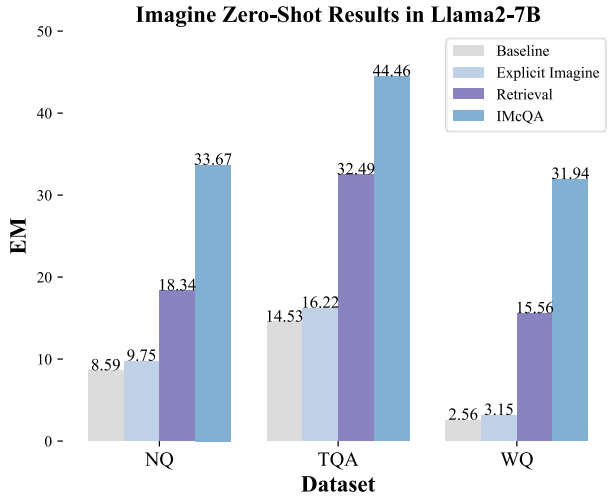

Figure 4: Zero-Shot results (EM, %) of Llama2-7B on three open-domain QA datasets.

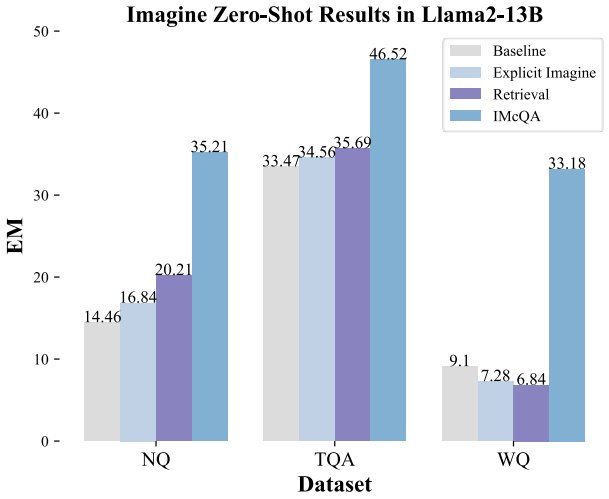

Figure 5: Zero-Shot results (EM, %) of Llama2-13B on three open-domain QA datasets.

individual LoRA adapters' setup. Despite the lack of a training advantage owing to distillation constraints, our approach achieves efficient reasoning through an extremely lightweight design. Compared to the other two methods, the token processing count is significantly decreased, while either outperforming them or showing negligible differences in performance. This represents an optimal trade-off between efficiency and computational demand. Moreover, unlike GAG, our approach incurs no financial costs associated with API calls, and the reduced model size facilitates faster generation.

4.8 Ablation Experiment

In this study, we introduced two key imagination processes to stimulate LLMs' internal knowledge: Explicit Imagination (EI) and Implicit Imagination (II). We particularly examined the influence of different imagination types on performance.

Figure 6 demonstrates that both EI and II are important for IMcQA. Omitting either one results in a considerable reduction in performance, with a drop exceeding 10% observed when EI is neglected. This is harmonious with the initial observation that performance improvement becomes more noticeable when relevant documents are available, thus underscoring EI's superiority.

The outcomes of Long Context Distillation (LCD) and the application of EI in the HyperNetwork also make marginal contributions to the overall results. This validates the previous assertion that a more extensive context tends to optimize performance, although with limited gains.

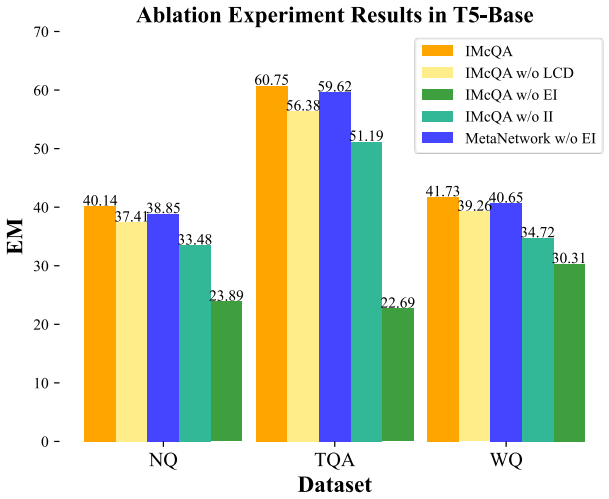

Figure 6: Ablation experiment results (%) of T5-Base on three open-domain QA datasets.

5 Conclusion and Future Work

This study proposes a novel knowledge-augmented strategy for LLMs, namely Imagine Augmented Generation (IAG), and a specific method IMcQA for open domain question answering. The proposed method effectively activates and utilizes intrinsic knowledge within LLMs through two imaginations: explicit imagination, and implicit imagination. Experimental results demonstrate a significant improvement in QA performance while remaining relatively lightweight. Although the main focus of this method is on one specific task, we believe these findings can offer a novel perspective on how to better harness the potential of LLMs. In the future, we plan to apply IAG to more NLP tasks and explore multimodal knowledge-augmented generation.

Limitations

While this study has demonstrated significant achievements in QA tasks, there are notable limitations:

Tasks. The proposed methods in the study are specialized specifically for QA. It remains unknown how effective they would be in other types of knowledge-intensive tasks, such as fact-checking or dialogue systems. Further validation is needed to assess the generalizations and applicability of this approach.

Multimodal. We have only considered imagined text and hidden representations. In future work, it is imperative to explore multimodal information including the impact of imagining images on performance.

Method. Our method relies on the knowledge learned by LLMs in the pre-training phase, which may limit the model's ability to quickly adapt to new information. The dependency on internal knowledge activation in IAG may lead to a less transparent decision-making process in the model, making it challenging to explain the logic behind the generated answers. In the future, there is a need to continue exploring adaptive knowledge enhancement methods to optimize results further.

Ethical Considerations

In this paper, we proposed a novel knowledge enhancement method aimed at leveraging the knowledge of LLMs. However, LLMs may generate inappropriate or discriminatory knowledge. Our approach does not introduce ethical concerns. The datasets we used are public, and there are no privacy issues.

References

Abdelrahman Abdallah and Adam Jatowt. 2023. Generator-retriever-generator: A novel approach to open-domain question answering. *arXiv preprint arXiv:2307.11278*.

Syeda Nahida Akter, Aman Madaan, Sangwu Lee, Yiming Yang, and Eric Nyberg. 2024. Self-imagine: Effective unimodal reasoning with multimodal models using self-imagination.

J. Berant, A. Chou, R. Frostig, and P. Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

David Ha, Andrew Dai, and Quoc V. Le. 2016. Hypernetworks.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew Peters. 2023. Hint: Hypernetwork instruction tuning for efficient zero-shot generalisation. *ACL*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020.

TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Amirhossein Kazemnejad, Mehdi Rezagholizadeh, Prasanna Parthasarathi, and Sarath Chandar. 2023. Measuring the knowledge acquisition-utilization gap in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4305–4319, Singapore. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Yejoon Lee, Philhoon Oh, and James Thorne. 2023. Knowledge corpus error in question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9183–9197, Singapore. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Yucheng Li, Bo Dong, Chenghua Lin, and Frank Guerin. 2023. Compressing context to enhance inference efficiency of large language models.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 364–378.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. 2022. Hypertuning: Toward adapting large language models without back-propagation.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020a. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020b. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Devendra Singh Sachan, Siva Reddy, William L. Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. In *Advances in Neural Information Processing Systems*.

Sagi Shaier, Lawrence E Hunter, and Katharina von der Wense. 2024. Desiderata for the context use of question answering systems.

Jianlin Su. 2021. You probably don't need bert-flow: A linear transformation comparable to bert-flow.

Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 2019. PullNet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2380–2390, Hong Kong, China. Association for Computational Linguistics.

Tianyi Tang, Yushuo Chen, Yifan Du, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Learning to imagine: Visually-augmented natural language generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9468–9481, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023a. Recomp: Improving retrieval-augmented lms with compression and selective augmentation.

Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, and Jian guang Lou. 2023b. Re-reading improves reasoning in language models.

Yunzhi Yao, Shaohan Huang, Li Dong, Furu Wei, Huajun Chen, and Ningyu Zhang. 2022. Kformer: Knowledge injection in transformer feed-forward layers. *ArXiv*, abs/2201.05742.

Yunzhi Yao, Peng Wang, Shengyu Mao, Chuanqi Tan, Fei Huang, Huajun Chen, and Ningyu Zhang. 2023. Knowledge rumination for pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3387–3404, Singapore. Association for Computational Linguistics.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *International Conference for Learning Representation (ICLR)*.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Chong Meng, Shuaiqiang Wang, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2023. Knowing what llms do not know: A simple yet effective self-detection method.

A Appendix

A.1 Background

Our task formulation follows retrieval augmented models for QA (Guu et al., 2020; Sachan et al., 2021). Let $\mathcal{V}^*$ denote the infinite set, encompassing all potential strings over the tokens in vocabulary $\mathcal{V}$, and this includes the empty string. An instance within a QA dataset is defined as a triplet (q, a, c) comprising question q, answer a, and context c, where $q, a, c \in \mathcal{V}^*$. Conventionally, the context c is drawn from the knowledge corpus $\mathcal{Z}$, like Wikipedia, whereby $\mathcal{Z} \subset \mathcal{V}^*$.

The goal of QA is to learn a distribution function, represented as $p(a|q)$, wherein the models decode a string a that serves as an abstractive answer to a given query q. In a closed-book setting, LLMs directly encode the given question and predict the answer (Roberts et al., 2020b). Specifically, considering the context c as the empty string,

the reliance is solely on the model parameters, i.e., $\hat{a} = \arg\max_{a \in \mathcal{V}^*} p(a|q, \theta)$, where θ represents the LLMs' parameters. However, employing a direct approach of requesting models to output answers frequently results in subpar performance, primarily attributable to the omission of a substantial amount of world knowledge during the process. Therefore, a popular approach is open domain setting, which marginalizes $p(a|q, c)$ over contexts c in the knowledge corpus (Lewis et al., 2020; Sachan et al., 2021) or generated from models (Yu et al., 2023). Given the computational infeasibility of calculating probabilities for all contexts, $p(a|q, c)$ is approximated to the sum of probabilities for top k contexts, i.e., $p(a|q, c) = \sum_{c \in TopK(q)}^{c_i \in c} p(a|q, c_i)p(c_i|q)$, where $Topk(q)$ denotes the set of resulting top k passages after the retrieval or generated with a query q.

A.2 Prompts for Explicit Imagine with LLMs

The prompt for explicit imagination of the Imagine Model to imagine a short dummy useful document is:

Imagine contexts based on the question: \n input \n Contexts: \n

Table 7 shows the full prompts for zero-shot results on LLM that we use for open domain QA: NQ, TQA, WQ.

A.3 Experimantal Settings

In this section, we describe the implementation of our experiments in detail, including the baseline methods, backbone models, and hyperparameters. Our model is built based on the T5 (Roberts et al., 2020a). Differing from fine-tuning all model parameters θ of the updated Pre-trained Language Model (LLM), LoRA (Hu et al., 2021) freezes all pre-trained Transformer parameters and optimizes only the parameters of each LoRA adapter. We employ LoRA to train a parameter-efficient fine-tuning baseline. Drawing from this, our approach updates only the parameters of the HyperNetwork to generate the weights for each LoRA adapter. This method is adopted based on LongLoRA's (Chen et al., 2023) recommendations and experimental findings, demonstrating improved performance when the normalization and FFN layers components are updated. This is because: 1) dynamically generating LoRA weights enhances generalization and parameter sharing, and 2) LoRA performs comparably to fine-tuning but mitigates the risk of catastrophic forgetting.

For the baseline, most of the hyperparameters are the default parameters of FiD (Izacard and Grave, 2021). For LoRA (Hu et al., 2021), add the LoRA module only to the $\mathcal{QV}$ of the attention layers and also release the normalization and FFN layers.

We consider conducting experiments using three different sizes of T5, namely T5-base, T5-large, T5-3b, and Llama2-7B, Llama2-13B (Touvron et al., 2023). Due to memory constraints and online distillation limitations, A100 supports processing 20 documents for T5-3b, while Llama2 does not support distillation. All experiments with T5-3b are conducted on 2 A100 GPUs, T5-large on 2 A6000 GPUs, and T5-Base on 2 RTX 3090 GPUs. However, experiments with Llama2-7b and 13b, except for IMcQA on 2 A100 GPUs, are tested on 8 RTX 3090 GPUs.

A.3.1 Hyperparameters

The detailed hyperparameter setting is as shown in Table 4. For the LoRA modules, we set the α 32 and the *lora rank* 32.

Models	Documents	Steps	Lr	Batch Size
T5	0	40000	1e-4	8
LoRA-Base	0	40000	5e-4	8
IMcQA	0	50000	1e-3	8
LoRA-l	0	40000	1e-4	4
IMcQA-l	0	50000	5e-4	4
FiD-3b	0	40000	1e-4	2
LoRA-3b	0	40000	1e-4	4
IMcQA	0	50000	1e-4	1
LoRA-Base	0†	40000	5e-4	8
IMcQA	0†	50000	1e-3	8
LoRA-l	0†	40000	1e-4	4
IMcQA-l	0†	50000	5e-4	4
LoRA-3b	0†	40000	1e-4	2
IMcQA-3b	0†	50000	1e-4	1
IMcQA	10	50000	5e-4	1
IMcQA-l	10	50000	5e-4	1
FiD-3b	10	40000	1e-4	1
IMcQA-3b	10	50000	1e-4	1

Table 4: Hyperparameter Settings.

A.3.2 Baselines

DPR (Karpukhin et al., 2020) generates by searching for the most relevant documents through dense vector space representation.

Models	# Documents	NQ			TQA			WQ		
		NQ	TQA	WQ	NQ	TQA	WQ	NQ	TQA	WQ
T5	0	22.16	3.18	4.12	2.65	21.8	3.15	0.88	2.95	28.3
LoRA-Base	0	16.17	4.71	6.89	3.15	21.16	0.00	1.33	3.04	26.38
IMcQA	0	23.89	6.21	10.94	5.31	22.69	6.30	3.23	5.10	30.31
LoRA-Base	1†	37.17	45.20	15.62	19.57	55.37	12.50	14.15	30.89	28.88
IMcQA	1†	40.14	46.61	18.92	24.78	60.75	12.82	17.70	35.24	41.06
FiD	10	46.81	53.93	24.02	28.57	63.32	17.83	18.81	41.88	41.78
IMcQA	10	**47.01**	**55.74**	**24.13**	**31.77**	**64.95**	**19.52**	**24.43**	**48.10**	**46.36**
T5-l	0	28.5*	3.18	4.12	2.65	28.7*	3.15	0.88	2.95	30.6*
LoRA-l	0	17.70	7.49	8.66	3.54	23.87	4.72	0.00	5.65	29.13
IMcQA-l	0	29.32	10.17	14.06	7.02	30.11	7.81	2.65	7.06	32.68
LoRA-l	1†	37.61	48.50	20.71	20.54	62.71	14.81	15.36	33.83	39.37
IMcQA-l	1†	42.32	54.80	22.05	26.11	65.48	18.11	18.58	47.46	45.28
FiD-l	10	46.7*	57.93	25.12	34.29	61.9*	19.64	27.65	53.87	48.1*
IMcQA-l	10	**49.92**	**60.03**	**25.79**	**34.35**	**69.67**	**20.28**	**30.19**	**54.94**	**51.52**

Table 5: **OOD results**. The primary row in the table header delineates the dataset trained, while the underscored secondary row demonstrates the in-distribution performance. IMcQA attains optimal performance both in-distribution and OOD under diverse document configurations.

FiD (Izacard and Grave, 2021) retrieve relevant documents and send them separately to the Encoder, then fuse the information in the Decoder.

GENREAD (Yu et al., 2023) prompt LLMs like InstructGPT (Ouyang et al., 2022) to generate a large number of relevant documents and let the reader process them.

LoRA We use LoRA (Hu et al., 2021) to obtain an efficiently fine-tuned baseline and compare it with our method.

A.3.3 Evaluation

For QA datasets, we choose the exact match (EM) score (Rajpurkar et al., 2016) as the evaluation metric. An answer is deemed correct if it aligns with any of the responses in the list of acceptable answers after normalization. Normalization involves transforming the text into lowercase, omitting articles, punctuation, and eliminating redundant spaces.

A.4 Downstream Evaluation Datasets

We use the following three Open-Domain QA for the experiments (§ 4.1).

- NaturalQuestions ((Kwiatkowski et al., 2019)) contains questions corresponding to Google search queries. The open-domain version of this dataset is obtained by discarding answers with more than 5 tokens, each accompanied by a Wikipedia article containing the answer.
- TriviaQA ((Joshi et al., 2017)) contains questions gathered from trivia and quiz-league websites. The unfiltered version of TriviaQA is used for open-domain question answering, each question is accompanied by pages from web and Wikipedia searches that may contain the answer.
- WebQuestions ((Berant et al., 2013)) contains questions from web queries matched to corresponding entries in FreeBase (Bollacker et al., 2008).

A.5 Full Experimental Results

A.5.1 Supervised Performance

As shown in Table 8, our initial observations indicate that regardless of the method implemented, supplying a certain quantity of related documents can expedite improvement and enhance performance in QA. FiD (Izacard and Grave, 2021) model outclasses all baseline models in performance. Notably, utilizing FiD-xl with a mere 10 documents yields performance on par with that attained through the use of FiD-l with 100 documents. Larger models not only encapsulate more knowledge but also demonstrate a superior ability to activate and apply this knowledge efficiently.

Additionally, in comparison with LoRA (Hu et al., 2021) methods, IMcQA enhances EM scores by an average of +2.2%. In the closed-book setting, the LoRA method manifests a substantial de-

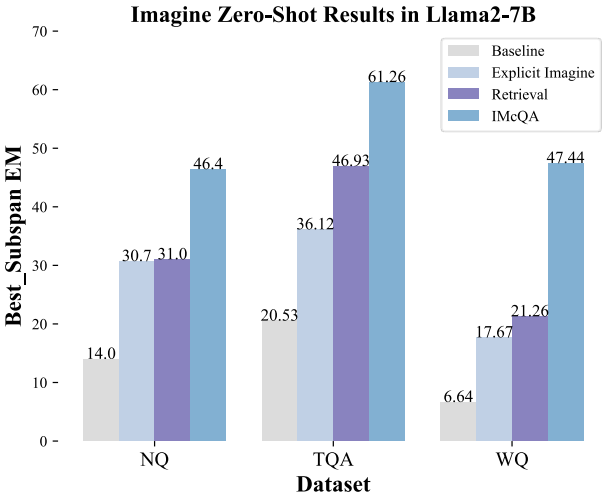

Figure 7: Zero-Shot results (Best_Subspan EM, %) of Llama2-7B on three open-domain QA datasets.

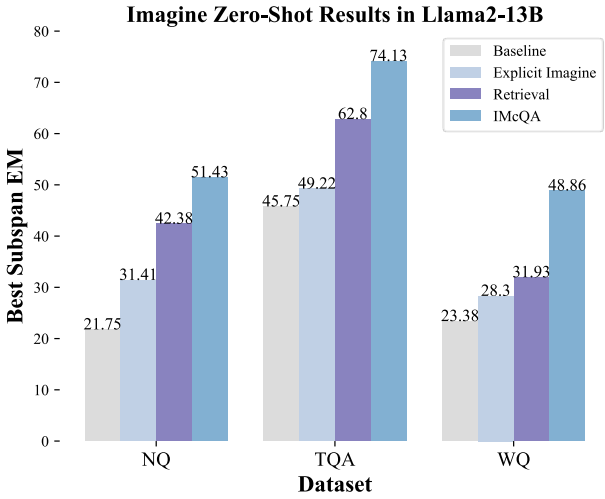

Figure 8: Zero-Shot results (Best_Subspan EM, %) of Llama2-13B on three open-domain QA datasets.

crease in performance, likely attributable to the inadequacy of learning sufficient knowledge via questions for storage in the LoRA module. On the other hand, IMcQA harnesses both explicit and implicit imaginations to exploit knowledge for improved outcomes. These results indicate that the knowledge stored in the LLMs' parameters can still be further exploited.

A.5.2 OOD Results

Table 5 shows the full OOD results in QA. It can be observed that our method has the best OOD generalization ability on all three benchmarks. Although LoRA performs well on the in-distribution part, its performance is generally poor on OOD, with some even showing negative performance. This highlights the importance of the domain adaptability of the implicit imagination HyperNetwork in our method, which generates LoRA adapter weights based on input.

A.5.3 Zero-Shot results Best_Subspan EM

LLMs have limited capacity to utilize extensive context effectively and are prone to generating illusions and redundant content. Best_subspan EM assesses whether the answer is included in the output. Previous studies have corroborated that LLMs encapsulate a considerable volume of knowledge and exhibit robust performance in QA.

Here, we report the Best_Subspan_EM values of Llama2-7B and Llama2-13B on three QA datasets. From Figure 7 and Figure 8, it can be observed that Best_Subspan_EM significantly improves, but the EM values are relatively small. This indicates that LLMs may not effectively utilize retrieval documents and are prone to outputting a lot of irrelevant information. Therefore, there is an urgent need to explore efficient techniques that leverage external information and internal knowledge.

However, the model did exhibit a weak adherence to instructions, often failing to output the exact answer. Remarkably, Llama2-13B displayed a decline in EM with an increase in document length on the WQ dataset, whereas the Best_Subspan_EM value augmented. Contrarily, our method excelled in extracting key information by using text imagination during the compression phase.

A.5.4 OOD and Ablation Experiment Results

Here, we supplement the experimental results of LoRA and IMcQA under supervised fine-tuning in closed-book settings and the ablation results of feedforward neural network (FFN) and Long Context Distillation (LCD). It can be observed that our method like LoRA, belongs to parameter-efficient fine-tuning, and because we share the HyperNetwork to generate LoRA adapter weights, we fine-tune fewer parameters.

From Table 6, it can be seen that releasing FFN can bring more performance improvement, possibly because adding LoRA in Attention cannot fully utilize enough knowledge (Yao et al., 2022). With the support of LCD, performance is further improved, with an average increase in EM of +5%. This also proves the effectiveness of our proposed LCD. In comparison with IMcQA and LoRA, it becomes more evident that LoRA tends to transfer knowledge to the LoRA module, resulting in low

Models	# Documents	Trainable Params	NQ			TQA			WQ		
			NQ	TQA	WQ	NQ	TQA	WQ	NQ	TQA	WQ
T5	0	220M	22.16	3.18	4.12	2.65	21.8	3.15	0.88	2.95	28.3
LoRA-Base	0	28.3M	5.43	3.15	4.02	0.00	9.60	0.00	0.22	1.77	20.47
w FFN	0	141.5M	16.17	4.71	6.89	3.15	21.16	0.00	1.33	3.04	26.38
w FFN & LCD	0	141.5M	21.37	2.82	6.89	1.99	17.94	3.74	0.00	2.82	32.50
IMcQA	0	26.1M	5.31	3.82	5.71	0.22	10.34	2.12	0.55	2.30	16.58
w FFN	0	139.3M	21.05	4.52	6.50	3.51	19.08	3.15	2.11	3.84	28.17
w FFN & LCD	0	141.5M	23.89	6.21	10.94	5.31	22.69	6.30	3.23	5.10	30.31
T5-l	0	770M	28.5*	3.18	4.12	2.65	28.7*	3.15	0.88	2.95	30.6*
LoRA-l	0	42.5M	4.42	6.50	7.87	3.98	10.03	3.94	1.99	6.71	18.11
w FFN	0	445.1M	17.70	7.49	8.66	3.54	23.87	4.72	0.00	5.65	29.13
w FFN & LCD	0	445.1M	28.32	4.52	10.94	5.31	25.71	6.12	1.75	4.52	29.92
IMcQA-l	0	34.8M	7.08	8.90	9.45	4.42	13.14	8.66	2.43	10.17	17.72
w FFN	0	437.5M	23.01	8.33	11.02	3.51	20.08	3.15	3.51	5.65	31.50
w FFN & LCD	0	437.5M	29.32	10.17	14.06	7.02	30.11	7.81	2.65	7.06	32.68

Table 6: OOD and ablation experiment results. * denotes the results are from the existing papers and LCD denotes Long Context Distillation.

generalization. Our method enhances knowledge activation through dynamic generation, showing significant effects not only in-distribution but also in OOD.

Methods	Prompt
CBQA	Please write a high-quality answer for the given question using your knowledge. Only give me the answer and do not output any other words. Question: {question} Answer:
Retrieval	Please write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant). Only give me the answer and do not output any other words. Context: {context} Answer the question based on the given passages. Question: {question} Answer:
Imagine	Please write a high-quality answer for the given question using your knowledge and the provided imagined compressed results (some of which might be irrelevant). Only give me the answer and do not output any other words. Imagined Context: {context} Answer the question based on your knowledge and the given imagined context. Question: {question} Answer:

Table 7: Prompts for different methods on Zero-Shot setting. **CBQA** denotes closed-book QA that just prompts the model with the question.

Models	Reader Params	# Documents	NQ	TriviaQA	WebQ
# closed-book setting					
T5 (Roberts et al., 2020a)	220M	0	25.9*	23.8*	27.9*
T5-l (Roberts et al., 2020a)	770M	0	28.5*	28.7*	30.6*
T5-xl	3b	0	28.30	33.92	34.43
LoRA-Base (Hu et al., 2021)	220M	0	16.17	21.16	26.38
LoRA-l (Hu et al., 2021)	770M	0	17.70	23.87	29.13
LoRA-xl	3b	0	23.15	32.16	35.24
IMcQA (Ours)	220M	0	23.89	22.69	30.31
IMcQA-l (Ours)	770M	0	29.32	30.11	32.68
IMcQA-xl (Ours)	3b	0	29.59	35.71	37.40
# Retrieval Augmented Generation					
DPR (Karpukhin et al., 2020)	110M	100	41.5*	56.8*	41.1*
RAG (Lewis et al., 2020)	400M	10	44.5*	56.1*	45.2*
FiD (Izacard and Grave, 2021)	220M	100	48.2*	65.0*	46.71
FiD-l	770M	100	51.4*	67.6*	50.52
FiD-xl	3b	20	**55.18**	**72.92**	**52.85**
FiD-l	770M	10	46.7*	61.9*	48.1*
FiD-xl	3b	10	50.1*	66.3*	50.8*
# Generation Augmented Generation					
GENREAD-l (Yu et al., 2023)	770M	10†	40.3*	67.8*	51.5*
GENREAD-xl	3b	10†	42.6*	69.6*	52.6*
# Our proposed method					
LoRA-Base	220M	1†	34.51	54.05	32.28
LoRA-l	770M	1†	40.05	62.81	43.70
LoRA-xl	3b	1†	44.15	66.92	48.23
IMcQA	220M	1†	40.14	60.75	41.73
IMcQA-l	770M	1†	42.32	65.48	45.28
IMcQA-xl	3b	1†	46.51	68.38	50.45
IMcQA	220M	10	47.01	64.95	46.36
IMcQA-l	770M	10	49.92	69.67	51.52
IMcQA-xl	3b	5‡	**50.87**	**70.34**	**52.78**

Table 8: Full QA performances (%) of different methods on three datasets. The first part (closed-book setting) indicates that explicit documentation was not utilized; The latter three parts utilize explicit augmented documents. The best results are in bold. * means that those results are from existing papers, † denotes that the number of documents is generated (‡ indicates that the number of documents is reduced due to insufficient memory for distillation).