

# 000 DYNAMIC PARAMETRIC RETRIEVAL AUGMENTED GEN- 001 002 ERATION 003 004

005 **Anonymous authors**

006 Paper under double-blind review

## 007 008 ABSTRACT 009

010 Retrieval-augmented generation (RAG) enhances large language models (LLMs)  
011 by injecting externally retrieved documents into the input context. It significantly  
012 increases inference costs and introduces knowledge conflicts, primarily caused by  
013 the lack of corresponding parametric knowledge in LLMs. Recently, Parametric  
014 RAG (PRAG) proposed to overcome these limitations by embedding symbolic  
015 documents into LLMs parameters, effectively reducing the inference costs and  
016 conflicts through offline training. However, PRAG needs to convert all documents  
017 into parameters in advance, which incurs high training and storage costs and renders  
018 it difficult to generalize to unseen documents. To address these challenges, we  
019 propose **Dynamic Parametric RAG (DyPRAG)**, a novel framework that leverages  
020 a lightweight parameter translator model to efficiently convert symbolic documents  
021 into parametric knowledge online. Specifically, the parameter translator employs  
022 several linear layers to convert document embeddings into LoRA modules of  
023 feed-forward networks of LLMs directly. DyPRAG achieves test-time parametric  
024 knowledge enhancement by dynamically generating the requisite parameters, which  
025 not only reduces the inference cost and mitigates knowledge conflicts inherent  
026 in RAG, but also lowers the training and storage overhead of PRAG. Extensive  
027 experiments on multiple datasets demonstrate the effectiveness and generalization  
028 capabilities of DyPRAG. Furthermore, the combination of contextual knowledge  
029 with test-time generated parametric knowledge offers a practical and more powerful  
030 RAG paradigm which updates parametric knowledge adaptively, enables superior  
031 knowledge fusion and alleviates knowledge conflicts in real-world applications.  
032 Our code is available at [https://anonymous.4open.science/r/DyPRAG\\_ICLR](https://anonymous.4open.science/r/DyPRAG_ICLR).  
033

## 034 1 INTRODUCTION 035

036 Large language models (LLMs) have demonstrated remarkable capabilities across diverse tasks (Liu  
037 et al., 2023), yet their performance on knowledge-intensive applications (Frisoni et al., 2024), such  
038 as question answering, remains constrained by limited access to up-to-date or domain-specific  
039 knowledge and a tendency to hallucination (Joshi et al., 2017; Kwiatkowski et al., 2019). To address  
040 this gap, retrieval-augmented generation (RAG) (Guu et al., 2020a) has emerged as a widely adopted  
041 approach which retrieves documents from external sources (e.g., Wikipedia) and injects them into the  
042 context, referred to as **in-context injection** (Izacard & Grave, 2021) (as shown in Figure 1 (a)).

043 While RAG mitigates knowledge gaps (Brown et al., 2020), this in-context injection strategy suffers  
044 from several limitations. As more documents are retrieved, inference costs increase rapidly due to  
045 elongated input sequences. More critically, knowledge conflicts (also known as RAG hallucination)  
046 often occur when external content contradicts the internal parametric knowledge of the LLM. It  
047 is mainly caused by low knowledge overlap, leading to erroneous outputs even in the presence of  
048 relevant documents (Zhang et al., 2024; Sun et al., 2024).

049 Parametric RAG (PRAG) employs another way of injecting knowledge, which integrates external  
050 knowledge directly into the parameters of LLMs to address these problems, known as **parameter**  
051 **injection** (as shown in Figure 1 (b)). The workflow of PRAG is divided into two stages. During the  
052 offline phase, PRAG first augments the retrieved documents to facilitate memorization and manip-  
053 ulation of knowledge (Allen-Zhu & Li, 2023a;b). Then, the augmented documents are fine-tuned  
with LoRA (Hu et al., 2022), encoding contextual knowledge directly into parameters. In the online

phase, retrieved documents are replaced with loadable parameters. Despite its benefit, PRAG faces critical limitations. Augmentation, training, and storing parameters for each retrieved document incur high computational and storage costs, coupled with non-generalization which struggles to adapt to unseen documents, severely limiting scalability in real-world applications (e.g., in frequent knowledge updates domains). This presents a critical challenge for PRAG: *How to achieve more efficient and generalizable test-time parametric knowledge enhancement with comparable performance?*

To address aforementioned challenges, we propose **Dynamic Parametric RAG (DyPRAG)**, a novel lightweight framework that enables on-the-fly parameter injection at test time (Figure 1 (c)). Rethinking PRAG, its intrinsic goal is to obtain an underlying mapping function  $\mathcal{F}$  which transforms external documents into parameters by statically fine-tuning on each document. Instead of this, DyPRAG introduces the parameter translator  $\mathcal{F}'_\phi$ , a small hypernetwork trained offline to learn this generalized mapping. This model, once trained, can dynamically generate document-related parameters at inference time to enhance the parametric knowledge of LLMs. After a detailed analysis of computation and storage overhead, our method significantly reduces the high inference costs of traditional RAG while eliminating the rigid training and storage costs of PRAG.

Through extensive experiments, we derive two key findings: **1) DyPRAG enhances the test-time parametric knowledge of LLMs effectively.** During evaluation, DyPRAG outperforms standard RAG across different scales of LLMs, demonstrating its ability to enhance the internal knowledge of LLMs. Furthermore, although PRAG learns  $\mathcal{F}$  by training separately for each document, DyPRAG achieves comparable or even better performance in various scenarios with significantly lower costs. **2) Combining test-time generated parametric knowledge with contextual knowledge leads to superior knowledge fusion.** Following Su et al. (2025), we further investigate the combination of in-context injection and parameter injection, referred to as **DyPRAG-Combine**. In both independent identically distributed and out-of-distribution settings, DyPRAG-Combine achieves the best results, showing strong generalization ability. Additionally, we find that DyPRAG-Combine effectively relieves knowledge conflicts by first injecting context-related parameters to fill the absent parametric knowledge gap. Based on the experimental results on RAGTruth (Niu et al., 2023) benchmark, we observe that DyPRAG-Combine enables LLMs to better internalize contextual knowledge, even on unseen documents. We further provide an in-depth analysis of how the parameter translator maps knowledge from different sources and how to interpret model performance based on internal signals. These findings suggest that integrating parametric and contextual knowledge using DyPRAG could be a promising approach for building a powerful and robust RAG system in real-world applications. We summarize our contributions as follows:

- We propose **Dynamic Parametric RAG (DyPRAG)**, a novel lightweight framework that efficiently converts symbolic documents into parameters at test-time. To the best of our knowledge, DyPRAG is the first approach in the RAG field to enable online transformation of symbolic knowledge into parametric representations, thereby eliminating the need for offline pre-conversion and storage of parameterized documents.
- We further develop a practical and powerful RAG paradigm **DyPRAG-Combine**, which effectively integrates symbolic documents with dynamically generated parametric knowledge, enabling the supplementation of requisite parametric knowledge in advance.
- Experimental results demonstrate that DyPRAG not only significantly outperforms in generalization but also efficiently enhances parametric knowledge and seamlessly integrates contextual knowledge,

108 boosting performance while reducing knowledge conflicts. As a result, DyPRAG provides a more  
 109 powerful and robust RAG paradigm for real-world applications.  
 110

111

## 112 2 RELATED WORK

113

114 2.1 RETRIEVAL AUGMENTED GENERATION

115

116 Large language models (LLMs) have demonstrated remarkable performance across diverse applica-  
 117 tions. However, their inherent knowledge often falls short in handling knowledge-intensive tasks,  
 118 highlighting the need for external knowledge integration to ensure robust performance in such con-  
 119 texts. A prominent approach to bridging this gap is retrieval-augmented generation (RAG), which  
 120 augments LLMs by incorporating relevant external knowledge sources (Borgeaud et al., 2022; Wang  
 121 et al., 2024a;b; Guu et al., 2020b). The retrieved documents are appended to the LLM’s input context,  
 122 enabling it to leverage knowledge beyond its training data (Lewis et al., 2020). However, this approach  
 123 leads to high inference costs as the number and length of retrieved documents increase (Xiong et al.,  
 124 2023). To address this issue, a recent study introduces Parametric RAG (PRAG) (Su et al., 2025), a  
 125 paradigm that fine-tunes the model on augmented documents, encoding useful information into pa-  
 126 rameters. While PRAG mitigates the inference cost, it introduces additional training and storage costs  
 127 due to the need to obtain and store LoRA parameters. Our proposed method significantly reduces  
 128 the high costs associated with standard RAG and PRAG while achieving superior generalization. By  
 129 combining contextual knowledge with test-time generated parametric knowledge via DyPRAG, our  
 130 approach enables better knowledge fusion and effectively mitigating knowledge conflicts.

131 2.2 CONTEXT COMPRESSION

132

133 Context compression is widely adopted to improve the efficiency of LLMs in processing contextual  
 134 knowledge. Recent studies propose condensing long contexts into soft prompts, allowing LLMs to  
 135 utilize information more effectively (Mu et al., 2023; Ge et al., 2023). Meanwhile, other works focus  
 136 on transforming context chunks into LoRA modules to improve the understanding ability of extended  
 137 contexts (Mao et al., 2024; Wang et al., 2024c; Charakorn et al., 2025). xRAG (Cheng et al., 2024)  
 138 integrates context compression by mapping documents into a compact token representation. Similarly,  
 139 AAG (Liao et al., 2024b) draws inspiration from human cognition, retrieving and recalling relevant  
 140 knowledge to compensate for knowledge gaps. This approach activates relevant information within  
 141 LLMs without relying on external resources. Building upon these advancements, we present the first  
 142 in-depth investigation into transforming symbolic documents into model parameters within RAG  
 143 systems. Our study demonstrates that this approach effectively unifies the contextual and parametric  
 144 knowledge, making it highly suitable for the RAG domain. This unification significantly mitigates  
 145 knowledge conflicts and enhances overall performance in RAG systems.

146

## 147 3 METHODOLOGY

148

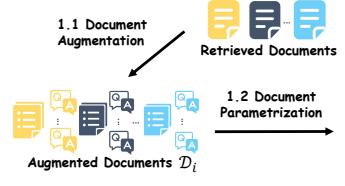
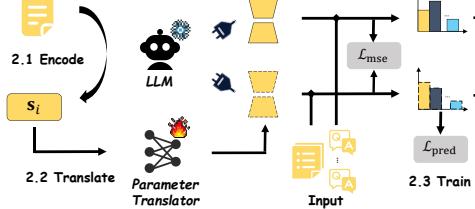
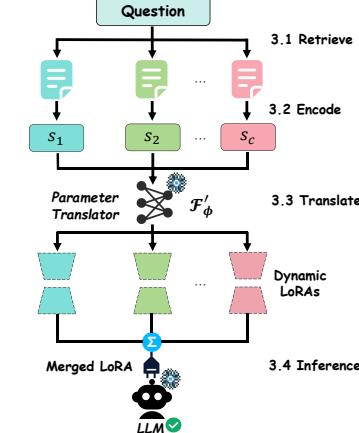
149 In this section, we introduce the Dynamic Parametric RAG framework, as shown in Figure 2. We first  
 150 formulate the problem and review the previous PRAG framework. Specifically, we revisit the offline  
 151 document parameterization process, which transforms documents into parametric representations  
 152 through **Document Augmentation** and **Document Parameterizing**, following Su et al. (2025).  
 153 Subsequently, we present our **Parameter Translation** process, which learns the underlying function to  
 154 map document embeddings into feed-forward networks (FFN) parameters via LoRA (Hu et al., 2022).  
 155 Once the translator is well optimized, retrieved documents can be directly converted into parametric  
 156 representations online. These parameters can be efficiently integrated into LLMs, enhancing model  
 157 parametric knowledge while reducing inference, training, and storage costs at test-time.

158

### 159 3.1 PRELIMINARY OF PARAMETRIC RAG

160

161 This subsection introduces the problem formulation of the RAG task and outlines the Parametric  
 162 RAG pipeline proposed in Su et al. (2025).

162  
163  
164165 **A. DyPRAG Offline Phase**166 **Stage 1: Doc-Param Pairs Collection**173 **Stage 2: DyPRAG Training**165 **B. DyPRAG Online Phase**166 **Stage 3: DyPRAG Inference**

183 Figure 2: An illustration of the DyPRAG framework. The offline phase consists of two stages:  
184 **Stage 1** follows the same parameterization process as PRAG to collect document-parameter (Doc-  
185 Param) pairs. In **Stage 2**, a parameter translator  $\mathcal{F}'_\phi$  is trained to learn a generalizable mapping from  
186 documents to corresponding parametric representations. During the online phase, **Stage 3** leverages  
187 the learned translator  $\mathcal{F}'_\phi$  to dynamically generate LoRA modules for any document at test-time. This  
188 enables DyPRAG to enhance LLMs with external parametric knowledge on demand.

191 **Standard RAG.** Let  $\mathcal{M}$  denote a large language model (LLM) with base parameters  $\Theta$ . Given a  
192 user query  $q$ , the task of LLM is to generate an accurate response augmented by an external corpus  $\mathcal{C}$ ,  
193 expressed as  $\mathcal{C} = \{d_1, d_2, \dots, d_N\}$ . Each element  $d_i$ , referred to as a document, represents a text  
194 chunk retrieved from external sources (Izacard & Grave, 2021). To achieve this, a retrieval module  
195  $\mathcal{R}$  is employed to compute relevance scores between  $q$  and the documents in  $\mathcal{C}$ . Traditional RAG  
196 approaches select the top- $c$  documents with the highest similarity scores and concatenates them with  
197 the query to form the extended input context. Based on this augmented input,  $\mathcal{M}$  generates the  
198 response by leveraging both the query and the retrieved documents. This procedure, referred to as  
199 **In-context Injection**, significantly increases the inference costs as the context length grows.

201 **Parametric RAG.** In contrast, Parametric RAG (PRAG) integrates documents directly into the  
202 parameters of  $\mathcal{M}$  to reduce the cost associated with long contexts. Each document  $d_i \in \mathcal{C}$  is  
203 transformed offline into a parametric representation  $\mathbf{P}_i = \mathcal{F}(d_i)$ , where  $\mathcal{F}$  is an underlying mapping  
204 function that converts each document  $d_i$  into its corresponding parameters  $\mathbf{P}_i$ . To achieve  
205 a more effective mapping, PRAG employs **Document Augmentation**, inspired by Allen-Zhu &  
206 Li (2023a;b), to help the model memorize and manipulate the information contained in document  
207  $d_i$ . Specifically, PRAG uses  $\mathcal{M}$  to rewrite  $d_i$  into multiple variations, resulting in  $\{d_i^1, d_i^2, \dots, d_i^n\}$ .  
208 Additionally, for each original document  $d_i$ , PRAG prompts  $\mathcal{M}$  to generate  $m$  question-answer (QA)  
209 pairs:  $\{(q_i^1, a_i^1), (q_i^2, a_i^2), \dots, (q_i^m, a_i^m)\}$ , where  $n$  and  $m$  are hyperparameters. This augmented set  
210 of documents preserves the factual content of the original document while incorporating diverse  
211 linguistic variations, expressed as:

$$\mathcal{D}_i = \{(d_i^k, q_i^j, a_i^j) \mid k \in [1, n], j \in [1, m]\}, \quad (1)$$

212 where each triple  $(d_i^k, q_i^j, a_i^j)$  is then concatenated to a training sample  $x = [d_i^k \oplus q_i^j \oplus a_i^j]$ . For  
213 **Document Parameterizing**, PRAG utilizes LoRA (Hu et al., 2022) to encode parametric knowledge

216  $\mathbf{P}_i$  for each  $\mathcal{D}_i$  where the overall goal is to optimize:  
 217

$$218 \quad \min_{\mathbf{P}_i} \sum_{(d_i^k, q_i^j, a_i^j) \in \mathcal{D}_i} \sum_{t=1}^T -\log P_{\Theta+\mathbf{P}_i}(x_t | x_{<t}), \quad (2)$$

221 where  $\mathbf{P}_i$  is the trainable low-rank matrix and only apply to feed-forward network (FFN). During  
 222 the inference phase, PRAG directly incorporates the obtained parametric representation  $\mathbf{P}_i$  into the  
 223 model parameters. We refer to this approach as **Parameter Injection**. Notably, although this method  
 224 eliminates the use of documents as context, it further introduces significant training cost and storage  
 225 cost, which will be analyzed in Section A.  
 226

### 227 3.2 DYNAMIC PARAMETRIC RAG

228 In this section, we describe the detailed process of our proposed **Dynamic Parametric RAG**  
 229 (**DyPRAG**) paradigm. Rethinking PRAG, its intrinsic goal is to obtain a document-specific mapping  
 230 function  $\mathcal{F}$  through repeated augmentation and training for each document  $d_i$  separately. However,  
 231 this process is computationally intensive and impractical in real-world applications, where new  
 232 documents require retraining from scratch. We argue that the key to optimization lies in addressing  
 233 the question: *How to obtain a generalized mapping function  $\mathcal{F}$ ?* To this end, we propose a three-  
 234 stage framework designed to enable parameter injection in an effective and efficient manner, which  
 235 eliminates the need to pre-convert and store parameters for all documents offline.  
 236

237 **Doc-Param Pairs Collection.** To derive the general mapping function  $\mathcal{F}$ , we start by collecting  
 238 a set of document-parameter (Doc-Param) pairs using the method described in Sec. 3.1. For each  
 239 document  $d_i$ , we collect its corresponding parametric representation  $\mathbf{P}_i$ , forming the alignment set  
 240  $\mathcal{K} = \{(d_1, \mathbf{P}_1), (d_2, \mathbf{P}_2), \dots, (d_N, \mathbf{P}_N)\}$ .  
 241

242 **Dynamic Parametric RAG Training.** After obtaining the alignment set  $\mathcal{K}$ , we utilize the original  
 243 LLM  $\mathcal{M}$  to encode textual documents into embeddings. For a given document  $d_i$ , we extract the  
 244 last hidden state  $\mathbf{s}_i \in \mathbb{R}^h$  at the last token position before transforming it into the vocabulary space,  
 245 where  $h$  represents the hidden dimension. To model the implicit transformation, we design a simple  
 246 hypernetwork called **Parameter Translator**  $\mathcal{F}'_\phi$  to translate  $\mathbf{s}_i$  into parametric representation  $\mathbf{P}_i$ .  
 247 This hypernetwork consists of several linear layers parameterized by a base parameter  $\phi$ . As an  
 248 example, consider the up-project module in FFN. The standard LoRA process as follows:  
 249

$$249 \quad \mathbf{W}' = \mathbf{W} + \Delta \mathbf{W} = \mathbf{W} + \mathbf{B} \mathbf{A} \quad (3)$$

250 where  $\mathbf{W} \in \mathbb{R}^{h \times k}$ ,  $\mathbf{B} \in \mathbb{R}^{h \times r}$  and  $\mathbf{A} \in \mathbb{R}^{r \times k}$ .  $k$  represents the intermediate dimension of FFN and  
 251  $r$  is the controllable LoRA rank. At training phase,  $\mathcal{F}'_\phi$  performs separately on  $\mathbf{B}$  and  $\mathbf{A}$ . Formally:  
 252

$$252 \quad \mathbf{B}^l = \text{Reshape}(\mathbf{W}_{\text{up}}^{l,B} \text{Relu}(\mathbf{W}_{\text{down}}^{l,B}(\mathbf{s}_i \oplus \text{idx}^l))) \quad (4)$$

253 where  $\mathbf{W}_{\text{down}}^{l,B} \in \mathbb{R}^{p \times (h+1)}$  and  $\mathbf{W}_{\text{up}}^{l,B} \in \mathbb{R}^{hr \times p}$ . Here,  $p$  represents the tunable intermediate dimension of the MLP module in  $\mathcal{F}'_\phi$ , and  $\text{Reshape}(\cdot)$  reshapes the output vector into the shape of  $\mathbf{B}$ .  
 254 This process is applied at each layer  $l$ , so we concatenate the layer index with  $\mathbf{s}_i$ . We provide the  
 255 visualization of this workflow in Appendix J. A similar procedure is followed for matrices  $\mathbf{A}$  and in  
 256 other modules of FFN. The parametric representation generated by  $\mathcal{F}'_\phi$  is denoted as  $\mathbf{P}'_i$ . Our goal is  
 257 for it to perform as effectively as  $\mathbf{P}_i$ .  
 258

259 To align with PRAG (Su et al., 2025), we utilize the augmented dataset  $\mathcal{D}_i$  and the same objective  
 260 function as presented in Eq. 2 to optimize  $\mathcal{F}'_\phi$  which corresponds to  $\mathcal{L}_{\text{pred}}$ . Additionally, for the target  
 261 LoRA adapter  $\mathbf{P}_i$ , we employ  $\mathcal{L}_{\text{mse}}$  to compute the difference between the generated parameters and  
 262 the target parameters. The Kullback-Leibler divergence (Polzehl & Spokoiny, 2006), denoted as  $\mathcal{L}_{\text{kl}}$ ,  
 263 quantifies the discrepancy in word probability distributions between the two models, with the model  
 264 using  $\mathbf{P}_i$  serving as the target distribution to be imitated. The overall formulation is given by:  
 265

$$265 \quad \mathcal{L}_{\text{mse}} = \text{MSE}(\mathbf{P}_i, \mathcal{F}'_\phi(d_i)) \quad (5)$$

$$266 \quad \mathcal{L}_{\text{kl}} = \text{KL}(P_{\Theta+\mathbf{P}_i}(x | D_i), P_{\Theta+\mathcal{F}'_\phi(d_i)}(x | D_i)) \quad (6)$$

$$268 \quad \mathcal{L}_{\text{align}} = \mathcal{L}_{\text{pred}} + \lambda_1 \mathcal{L}_{\text{mse}} + \lambda_2 \mathcal{L}_{\text{kl}} \quad (7)$$

269 where we calculate the overall alignment loss for each document  $d_i$ ,  $\lambda_1$  and  $\lambda_2$  are tunable hyper-  
 270 parameter which set to 100 and 0.01 separately to make loss range similar.  
 271

270 **Dynamic Parametric RAG Inference.** During the inference stage, once a well-trained parameter  
 271 translator  $\mathcal{F}'_\phi$  is obtained, we can efficiently perform parameter injection which significantly reduces  
 272 the inference costs. For a test query  $q^t$ , we rerun the retrieval process using the retrieval module  $\mathcal{R}$  to  
 273 select the most relevant documents. For each selected document  $d_i^t$ , we derive its embedding  $s_i^t$  and  
 274 input it into  $\mathcal{F}'_\phi$  to obtain the dynamic LoRA adapter  $\mathbf{P}_i^{t, \prime}$ , which encodes the relevant information  
 275 from the document in parameter. We then merge this as the LoRA parameter for inference, resulting  
 276 in low inference costs without requiring the concatenated documents.  
 277

## 278 4 EXPERIMENTS

### 280 4.1 EXPERIMENTS DETAILS

282 **Datasets.** We validate our approach using various benchmarks to evaluate distinct reasoning  
 283 abilities, including multi-hop reasoning and commonsense inference. The selected datasets are  
 284 **2WikiMultihopQA (2WQA)** (Ho et al., 2020), **HotpotQA (HQA)** (Yang et al., 2018), **PopQA**  
 285 (**PQA**) (Mallen et al., 2022) and **ComplexWebQuestions (CWQ)** (Talmor & Berant, 2018). We  
 286 provide detailed information about these datasets in Appendix B.1.

287 **Evaluation Metrics.** For evaluation, we use the Exact Match (EM) score (%) to compare the  
 288 extracted answer with the reference answer at the exact match level. Additionally, we employ the  
 289 F1 score (%), which balances precision and recall by considering partially correct answers. Both  
 290 2WQA and HQA categorize questions by reasoning type, with 2WQA having four categories and  
 291 HQA two. To compare DyPRAG with other RAG baselines across reasoning tasks, we use the first  
 292 300 questions from each sub-dataset for evaluation.  
 293

294 **Implementation Details.** To ensure broad effectiveness across models, we select LLMs of  
 295 varying scales and series, including Qwen2.5-1.5B-Instruct (Yang et al., 2024), LLaMA-3.2-1B-  
 296 Instruct (Meta, 2024b) and LLaMA-3-8B-Instruct (Meta, 2024a). For our base experiments, we  
 297 collect 200 additional questions from each non-overlapping sub-dataset. The number of retrieved  
 298 documents  $c$  is set to 3, resulting in a alignment set  $\mathcal{K}$  of 4,800 samples. The intermediate size  $p$  is  
 299 set to 32. All experiments were conducted using PyTorch on NVIDIA A100 GPUs (80GB). Please  
 300 refer to Appendix B.1 for more detailed settings.

### 301 4.2 BASELINES

303 We select the following baselines to compare with our proposed **DyPRAG**, detailed in Appendix B.1:

- 305 • **Vanilla** represents the answer from original LLMs without any external knowledge.
- 306 • **RAG** appends top-retrieved documents to the LLM’s input prompt, explicitly instructing the model  
 307 to reference them when answering.
- 308 • **PRAG** injects relevant documents into the LLM’s parameters via offline parameterization, reducing  
 309 reliance on retrieved documents.
- 310 • **SFT** fine-tunes LLMs with same setting in DyPRAG to encode all knowledge without context.
- 311 • **Context-DPO** (Bi et al., 2024) aligns LLMs through direct preference optimization (DPO) (Rafailov  
 312 et al., 2023) to enhance context-faithfulness of LLMs and inference with retrieved documents.

314 Following the approach in Su et al. (2025), we conduct experiments that combine both in-context and  
 315 parameter injection to explore their interaction. Specifically, the retrieved documents are appended to  
 316 the input context, and their corresponding parametric representations are integrated into the model.  
 317 This results in two additional baselines, referred to as **PRAG-Combine** and **DyPRAG-Combine**.

### 319 4.3 MAIN RESULTS

321 In this section, we present the main experimental results and a detailed analysis of DyPRAG in com-  
 322 parison with the selected baselines. Additionally, we provide efficient RAG baselines in Appendix C.  
 323 Notably, the vanilla model occasionally outperforms RAG in certain situations. We analyze the  
 reasons for this in Appendix H and confirm that it won’t affect the subsequent analysis.

324  
 325 Table 1: The experimental results of DyPRAG are compared with parametric RAG, standard RAG  
 326 and two training-based methods. **All metrics are reported as EM scores (%) and F1 scores (%).** The  
 327 best performance is bolded, while the second-best is underlined. The **Avg** is the average performance  
 328 over all tasks.

329 330 331 332 333 334	335 336 337 338 339	340 341 342 343	344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359	360 361 362 363 364 365 366 367 368 369	Base LLM	Method	2WQA		HQA		PQA		CWQ		Avg		
					EM	F1	EM	F1	EM	F1	EM	F1	EM	F1			
329 330 331 332 333 334	335 336 337 338 339	340 341 342 343	344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359	360 361 362 363 364 365 366 367 368 369	360 361 362 363 364 365 366 367 368 369	360 361 362 363 364 365 366 367 368 369	Vanilla	17.47	22.87	18.56	24.10	0.67	2.26	23.67	34.94	16.74	21.04
							SFT	8.67	11.25	1.67	2.96	0.00	1.33	7.67	12.77	5.60	7.92
							Context-DPO	19.33	24.14	17.00	23.35	4.00	12.79	7.67	13.00	15.93	21.66
							RAG	17.93	24.77	21.44	30.33	9.67	17.65	25.67	37.39	18.93	26.99
							PRAG	20.13	25.92	19.00	25.35	12.00	23.58	26.00	35.86	19.57	26.51
							PRAG-Combine	20.60	26.94	23.33	<b>30.81</b>	<u>20.33</u>	<b>31.07</b>	<u>28.33</u>	<b>39.63</b>	<u>22.17</u>	<b>29.78</b>
							DyPRAG (ours)	24.27	<u>29.91</u>	19.56	25.97	7.33	11.33	<u>28.33</u>	36.86	21.57	27.57
329 330 331 332 333 334 335 336 337 338 339	335 336 337 338 339	340 341 342 343	344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359	360 361 362 363 364 365 366 367 368 369	360 361 362 363 364 365 366 367 368 369	360 361 362 363 364 365 366 367 368 369	DyPRAG-Combine (ours)	<b>26.33</b>	<u>32.53</u>	<b>23.33</b>	<u>30.80</u>	<b>21.33</b>	<u>29.93</u>	<b>29.33</b>	38.96	<b>25.23</b>	<b>31.80</b>
							Vanilla	20.87	27.20	14.78	23.13	0.67	2.87	18.00	26.47	16.74	25.79
							SFT	18.60	22.61	8.78	13.63	0.00	6.95	4.67	13.96	12.40	17.49
							Context-DPO	17.60	24.35	15.00	24.35	0.33	<u>14.18</u>	12.33	19.20	14.57	22.82
							RAG	16.33	23.89	14.89	24.68	0.67	9.97	18.64	28.23	14.56	23.17
							PRAG	<b>21.93</b>	<b>29.38</b>	16.00	24.04	1.33	3.87	22.31	30.82	18.13	25.37
							PRAG-Combine	19.07	27.29	<u>19.33</u>	<b>26.15</b>	2.67	12.61	21.67	<u>32.13</u>	17.77	<b>25.96</b>
							DyPRAG (ours)	21.87	<u>28.46</u>	17.11	24.93	<u>3.00</u>	6.64	<u>22.67</u>	31.94	<u>18.64</u>	25.56
							DyPRAG-Combine (ours)	18.87	25.87	<u>20.67</u>	<b>30.13</b>	<u>7.33</u>	<b>22.69</b>	<u>23.67</u>	<u>33.57</u>	<b>18.74</b>	<b>27.60</b>
							Vanilla	30.00	36.43	19.89	28.64	4.67	7.96	<u>30.00</u>	42.44	24.43	31.85
							SFT	1.53	13.09	0.33	2.19	0.00	0.00	5.92	0.86	7.80	
							Context-DPO	14.93	24.42	12.45	21.67	4.33	18.68	8.00	13.81	12.43	21.96
							RAG	28.40	34.20	19.13	28.67	5.67	16.13	25.33	35.45	23.04	30.86
							PRAG	33.20	40.54	<u>35.55</u>	<u>45.88</u>	<u>20.33</u>	26.13	<b>32.67</b>	<b>43.54</b>	<u>32.57</u>	41.00
							PRAG-Combine	<u>34.47</u>	<u>42.20</u>	<b>40.11</b>	<b>50.82</b>	11.33	<u>26.23</u>	28.00	36.41	<b>33.20</b>	<u>42.61</u>
							DyPRAG (ours)	32.07	39.17	24.67	37.33	11.00	13.60	<b>32.67</b>	41.87	27.80	36.23
							DyPRAG-Combine (ours)	<b>36.33</b>	<b>47.68</b>	33.22	43.22	<b>21.00</b>	<u>32.86</u>	<u>29.67</u>	39.07	<u>33.20</u>	<b>43.69</b>

**Overall Analysis.** Since PRAG learns the mapping function  $\mathcal{F}$  by training separately for each document, it can be considered as a upper bound of DyPRAG. Remarkably, our proposed DyPRAG achieves comparable or even superior results across various tasks, as shown in Table 1. **For instance, using LLaMA3.2-1B, DyPRAG achieves an average score of 27.57% (21.57%), surpassing PRAG by 1.06% (2.00%), RAG by 0.58% (2.64%) and vanilla by 5.18% (4.83%) in F1 (EM) scores.** This demonstrates that our method learns more useful information when trained on diverse datasets. We also compare DyPRAG with efficient RAG baselines, including FLARE (Jiang et al., 2023) and DRAGIN (Su et al., 2024). As shown in Table 8, both DRAGIN and FLARE outperform standard RAG in most 2WQA settings. However, DyPRAG achieves even better results, demonstrating its superiority. For example, when using LLaMA3-8B as the base model, DyPRAG outperforms DRAGIN and FLARE by 1.56% and 2.63% on 2WQA in F1 scores, respectively. However, Context-DPO proves less effective in resolving knowledge conflicts, while SFT experiences severe collapse, failing to encode such a large amount of knowledge, which leads to significant performance degradation. These results highlight the consistent performance improvements offered by DyPRAG over all baselines, underscoring its effectiveness for test-time parametric knowledge enhancement.

**DyPRAG-Combine Leads to Superior Performance.** By combining in-context injection with parameter injection, DyPRAG-Combine achieves the best performance across all models, outperforming all baselines. **For instance, DyPRAG-Combine outperforms PRAG-Combine by 2.02% (3.06%) on LLaMA3.2-1B, 0.55% (0.17%) on Qwen2.5-1.5B and 1.08% (0.00%) on LLaMA3-8B on average in F1 (EM) scores.** Moreover, combining these two types of knowledge results in shorter responses, effectively reducing costs due to improved knowledge internalization ability, as shown in Figure 4. These results demonstrate the dynamic parameters generated by our approach effectively intergrade with contextual knowledge, enabling these two information sources to complement each other.

#### 4.4 OUT-OF-DISTRIBUTION PERFORMANCE

To further demonstrate the generalization ability of the DyPRAG method, we evaluate it in the out-of-distribution (OOD) scenario. Notably, PRAG can not handle this OOD scenario without additional offline training. We conduct the OOD performance on commonsense datasets: StrategyQA (**SQA**) (Geva et al., 2021), IIRC (Ferguson et al., 2020), and OpenBookQA (**OBQA**) (Mihaylov et al., 2018). Additionally, MedMCQA (**MQA**) (Pal et al., 2022) focuses on a completely unseen domain about medical. All OOD datasets are provided with ground-truth passages.

As shown in Table 2, the vanilla model performs poorly due to a lack of sufficient relevant knowledge, particularly in the IIRC dataset. DyPRAG effectively enhances parametric knowledge, resulting

378  
379 Table 2: The OOD performance on three open-  
380 domain datasets for  $\mathcal{F}'_\phi$  trained on  $\mathcal{K}$  is reported.  
381

Base Model	Method	IIRC	SQA	OBQA	MedQA	Avg
LLaMA3.2-1B	Vanilla	10.99	21.67	40.33	39.00	28.00
	SFT	2.83	0.00	0.00	0.00	0.71
	RAG	40.38	27.67	<b>52.00</b>	50.33	42.60
	DyPRAG	14.04	39.67	43.00	40.67	34.35
Qwen2.5-1.5B	DyPRAG-Combine	<b>41.91</b>	<b>50.33</b>	<b>52.00</b>	<b>52.67</b>	<b>49.23</b>
	Vanilla	8.78	1.00	40.09	33.67	20.89
	SFT	7.39	0.00	9.67	0.00	4.27
	RAG	30.52	39.00	45.00	<b>52.67</b>	41.80
LLaMA3-8B	DyPRAG	10.23	15.67	43.38	34.67	25.99
	DyPRAG-Combine	<b>38.25</b>	43.33	<b>48.57</b>	<b>52.67</b>	<b>45.71</b>
	Vanilla	13.23	33.33	52.33	55.00	38.47
	SFT	2.42	0.00	0.00	0.00	0.61
LLaMA3-8B	RAG	43.27	45.67	60.00	55.67	51.15
	DyPRAG	18.16	45.67	53.00	55.00	42.96
	DyPRAG-Combine	<b>57.90</b>	<b>58.67</b>	<b>60.67</b>	<b>56.67</b>	<b>58.48</b>

385 Table 3: Ablation study of alignment loss. The backbone model is the LLaMA3.2-1B.  
386

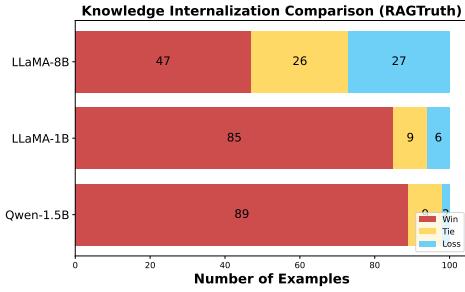
Method	2WQA					HQA			PQA	CWQ	Avg
	Compare	Bridge	Inference	Compose	Total	Bridge	Compare	Total			
DyPRAG	51.25	<b>48.15</b>	<b>17.35</b>	7.54	25.31	<b>14.05</b>	<b>43.9</b>	<b>19.97</b>	8.37	36.86	25.28
<i>Ablation Study</i>											
w/o $\mathcal{L}_{kl}$	29.54	37.74	12.94	5.78	20.27	7.35	35.28	13.12	1.93	22.85	18.68
w/o $\mathcal{L}_{mse}$	<b>56.06</b>	36.96	17.29	<b>8.40</b>	<b>27.28</b>	12.78	42.11	17.65	5.94	32.98	23.38
w/o $\mathcal{L}_{kl}, \mathcal{L}_{mse}$	45.23	24.84	16.74	7.48	23.43	12.66	39.46	18.26	2.42	34.92	22.54

399 in a moderate improvement in performance. However, when the model heavily relies on critical  
400 information from documents to answer questions, DyPRAG struggles to accurately reconstruct this  
401 information. This loss of information primarily stems from the encoding and translation processes,  
402 which contributes to the model’s suboptimal performance (e.g., in IIRC). Notably, DyPRAG-Combine  
403 which incorporates golden passages with document-related parametric knowledge leads to deeper  
404 knowledge fusion, achieving best performance across all scenarios, even in hardest IIRC task. For  
405 example, DyPRAG-Combine improves performance on SQA (MQA) by 22.66% (2.34%) using  
406 LLaMA3.2-1B, on IIRC by 13.63% using LLaMA3-8B and on OBQA by 3.57% using Qwen2.5-  
407 1.5B. We believe that the observed performance gain comes from the coarse-grained parametric  
408 transformation of document knowledge. This transformation increases the overlap between the two  
409 distinct types of knowledge, thereby improving the model’s ability to understand unseen documents.  
410 We further investigate DyPRAG’s performance on non-QA tasks in **Appendix C**. Additionally,  
411 **Appendix F** presents textual similarity analyses across datasets to validate a reliable OOD setting,  
412 while **Appendix G** examines the generated LoRA matrices to explore the parameter translator’s  
413 underlying generalization ability.

## 4.5 ABLATION STUDY

416 **Effect of Alignment Loss.** The alignment loss  $\mathcal{L}_{align}$  is composed of three components:  $\mathcal{L}_{pred}$ ,  $\mathcal{L}_{mse}$ ,  
417 and  $\mathcal{L}_{kl}$ . We investigated which component contributes the most to the effectiveness of DyPRAG. As  
418 shown in Table 3, removing any single loss component negatively impacts the model’s performance.  
419 For instance, when  $\mathcal{L}_{kl}$  is removed, the model’s performance drops significantly, demonstrating that  
420 aligning with the target output distribution is an effective strategy (Liao et al., 2024a). While removing  
421  $\mathcal{L}_{mse}$  has the smallest impact, ensuring that  $\mathcal{F}'_\phi$  generates  $\mathbf{P}'$  values as close as possible to the trained  
422  $\mathbf{P}$  still proves beneficial. Furthermore, even when only  $\mathcal{L}_{pred}$  is retained, DyPRAG maintains stable  
423 performance, indicating that the  $\mathcal{L}_{pred}$  loss plays a central role in the overall alignment. We further  
424 present several ablation studies, including the effects of training dataset size, injected documents  
425 number to performance, intermediate dimension  $p$ , different retrievers, and data augmentation, as  
426 detailed in Appendix D.

427  
428 **Effect of Data Augmentation.** In Section 3, we introduce data augmentation to improve the  
429 model’s ability to memorize and process information from documents. To assess the impact of data  
430 augmentation on the DyPRAG method, we remove it during the Doc-Param pair collection phase and  
431 compare the results with those of the original method. The results in Table 4 indicate that removing

385 Figure 3: Comparison of knowledge internalization between DyPRAG-Combine vs RAG  
386 judged by GPT-4o.  
387

432 Table 4: **Ablation study of effectiveness in data augmentation.** All metrics are reported as F1 scores  
 433 (%). The backbone model is the Qwen2.5-1.5B.

Method	2WQA (Total)	HQA (Total)	PQA	CWQ	IIRC	SQA	OBQA	MQA
<b>Vanilla</b>	26.87	17.76	2.87	26.47	8.78	1.00	40.09	33.67
<b>RAG</b>	24.31	20.73	9.97	28.23	30.52	39.00	45.00	52.67
<b>PRAG-Combine</b>	27.49	23.10	23.43	32.13	—	—	—	—
w/o Aug	22.79	19.00	10.74	28.54	—	—	—	—
Change	<b>-17.1%</b>	<b>-17.7%</b>	<b>-54.2%</b>	<b>-11.2%</b>	—	—	—	—
<b>DyPRAG</b>	26.46	19.67	6.64	31.94	10.23	15.67	43.38	34.67
w/o Aug	28.36	15.71	3.35	28.04	8.49	0.30	38.36	22.94
Change	<b>+7.2%</b>	<b>-20.1%</b>	<b>-49.5%</b>	<b>-12.2%</b>	<b>-17.0%</b>	<b>-98.1%</b>	<b>-11.6%</b>	<b>-33.8%</b>
<b>DyPRAG-Combine</b>	25.18	27.57	22.69	33.57	38.25	43.33	48.57	52.67
w/o Aug	23.00	19.88	9.84	27.97	29.41	30.67	43.90	34.03
Change	<b>-8.7%</b>	<b>-27.9%</b>	<b>-56.6%</b>	<b>-16.7%</b>	<b>-23.1%</b>	<b>-29.2%</b>	<b>-9.6%</b>	<b>-35.4%</b>

445  
 446 data augmentation greatly diminishes the quality of offline parameterization, which in turn affects  
 447 the parameter translator’s ability to convert documents into parametric knowledge. This degradation  
 448 results in a significant performance drop for both PRAG, which relies on offline parameterization,  
 449 and DyPRAG, which dynamically converts parameters.

450  
 451  
 452 **Effect of Parameter Translators Size.** As illustrated in Section A, the total storage cost for  $\mathcal{F}'_\phi$   
 453 is  $3L(phr + 2p(h + 1) + pkr)$ , which scales linearly with  $p$ . Therefore, we conducted an ablation  
 454 study on  $p$ . As shown in Table 11, DyPRAG consistently outperforms both standard RAG and PRAG.  
 455 Surprisingly,  $p = 2$  achieves the second-best performance with a storage cost of only 7.71MB. In  
 456 contrast, PRAG requires 9.33GB to store data for all test questions in main experiments, resulting in  
 457 a significant overhead. The experiments demonstrate that our proposed DyPRAG not only drastically  
 458 reduces storage costs but also enhances performance, showcasing exceptional robustness. Notably,  
 459 DyPRAG significantly reduces the inference cost compared to RAG, while introducing only minimal  
 460 overhead (i.e., encode and translate processes). We further present an end-to-end latency analysis in  
 461 Table 12, demonstrating that both DyPRAG and DyPRAG-Combine achieve faster inference. This  
 462 improvement is attributed to shorter responses and the asynchronous mode, which mitigates the  
 463 latency introduced by the encoding and translation processes in real-world scenarios.

#### 4.6 ANALYSIS OF CONTEXTUAL AND PARAMETRIC KNOWLEDGE CONFLICTS AND FUSION

464  
 465  
 466  
 467 **Pre-inject Converted Parameters Enhances Knowledge Overlap.** When LLMs struggle to  
 468 identify the more reliable information source (Tao et al., 2024; Zhang et al., 2024), it is primarily  
 469 due to conflicts between contextual knowledge and parametric knowledge, which fundamentally  
 470 stem from low overlap between these two types of knowledge. We further investigate how internal  
 471 signals, such as entropy, can detect RAG conflicts and how dynamic parametrization effectively  
 472 mitigates these issues. As shown in Table 5, we leverage Entropy (EN), Length Normalized Entropy  
 473 (LEN) (Malinin & Gales, 2020), and Lexical Similarity (LS) (Lin et al., 2022) to evaluate the  
 474 likelihood of knowledge conflicts. Our findings indicate that EN and LEN increase, while LS  
 475 decreases, when in-context injection is applied, suggesting that retrieved passages in RAG systems  
 476 often exhibit low overlap with the model’s internal knowledge that increasing the uncertainty during  
 477 generation. Notably, a comparison between RAG and DyPRAG-Combine shows that employing  
 478 the parameter translator to inject converted parametric knowledge significantly reduces knowledge  
 479 conflicts, underscoring the effectiveness of DyPRAG.

480 In contrast, DyPRAG-Combine effectively integrates contextual knowledge with transformed para-  
 481 metric knowledge, enabling it to provide correct answers and demonstrating its ability to leverage  
 482 both types of knowledge effectively. Compared to RAG, DyPRAG-Combine transforms the retrieved  
 483 documents into parameters before concatenated into the input prompt. This approach ensures that the  
 484 LLM already contains relevant knowledge when answering the questions, mitigating the well-known  
 485 conflicts issues (Sun et al., 2024).

486  
 487 Table 5: We present the experimental results for the knowledge conflicts metrics of DyPRAG and  
 488 DyPRAG-Combine, in comparison with Vanilla and Standard RAG. In these metrics,  $\uparrow$  indicates  
 489 that higher values are better, while  $\downarrow$  indicates the opposite. The best performance for each metric is  
 highlighted in bold. The backbone model is the LLaMA3.2-1B.

Metric	Method	2WQA (total)	HQA (total)	PQA	CWQ	SQA	IRC
EN $\downarrow$	Vanilla	3.187	3.176	3.251	3.163	3.178	3.011
	DyPRAG (ours)	<b>2.199</b>	<b>1.999</b>	<b>1.757</b>	<b>2.860</b>	<b>2.805</b>	<b>2.544</b>
	RAG	3.565	3.453	3.778	3.619	3.398	3.030
LEN $\downarrow$	DyPRAG-Combine (ours)	<b>2.755</b>	<b>2.470</b>	<b>3.584</b>	<b>3.467</b>	<b>3.136</b>	<b>2.555</b>
	Vanilla	0.637	0.635	0.650	0.633	0.636	0.602
	DyPRAG (ours)	<b>0.440</b>	<b>0.400</b>	<b>0.586</b>	<b>0.572</b>	<b>0.561</b>	<b>0.509</b>
LS $\uparrow$	RAG	0.713	0.691	0.756	0.724	0.680	0.606
	DyPRAG-Combine (ours)	<b>0.551</b>	<b>0.494</b>	<b>0.719</b>	<b>0.693</b>	<b>0.627</b>	<b>0.511</b>
	Vanilla	<b>0.923</b>	<b>0.936</b>	0.723	0.730	0.497	0.963
LS $\uparrow$	DyPRAG (ours)	0.915	0.933	<b>0.842</b>	<b>0.859</b>	<b>0.527</b>	<b>0.966</b>
	RAG	0.945	0.956	0.936	0.962	0.812	0.966
	DyPRAG-Combine (ours)	<b>0.953</b>	<b>0.959</b>	<b>0.966</b>	<b>0.988</b>	<b>0.853</b>	<b>0.975</b>

502  
 503 **DyPRAG Enables LLMs to Internalize Un- 504 seen Knowledge.** The retrieved documents 505 in our experiments are primarily sourced from 506 Wikipedia, which are already encountered by 507 LLMs during pre-training. In this section, we 508 further investigate how DyPRAG performs on 509 unseen documents using the RAGTruth bench- 510 mark (Niu et al., 2023). Specifically, we ran- 511 domly sample 100 examples from the QA-type 512 sub-dataset, which presents greater challenges 513 (e.g., the required answers are only accessible 514 in carefully crafted context). As shown in Fig- 515 ure 3, DyPRAG-Combine significantly outper- 516 forms RAG. This demonstrates that DyPRAG ef- 517 fectively enables LLMs to better internalize con- 518 textual knowledge and mitigate conflicts, even 519 when handling unseen data. Additionally, we 520 present a further detailed analysis of contextual 521 and parametric knowledge in **Appendix I**.

522 These experiments further validate our explanation in Section 4.4, demonstrating that DyPRAG 523 struggles with fine-grained information reconstruction due to the inherent constraints of compression. 524 However, the current results are sufficient to show that injecting transformed parametric knowledge 525 increases its overlap with contextual knowledge while effectively mitigating the issue of uncertain 526 responses caused by knowledge conflicts.

## 5 CONCLUSION

527  
 528 In this work, we propose Dynamic Parametric RAG (DyPRAG), a novel framework that addresses the 529 high inference cost of RAG, the high training and storage costs of parametric RAG, while effectively 530 mitigating knowledge conflicts. DyPRAG successfully learns the underlying mapping function from 531 documents to parameters by leveraging a hypernetwork, enabling effective parametric knowledge 532 enhancement at test-time. Extensive experiments conducted on multiple datasets demonstrate the 533 superior performance, flexibility, and practicality of DyPRAG. By dynamically combining test-time 534 generated parametric knowledge with contextual knowledge, DyPRAG enables adaptive parametric 535 knowledge updates, superior knowledge fusion, and effective mitigation of knowledge conflicts. 536 These advantages establish DyPRAG as a powerful and cost-efficient framework, highlighting its 537 potential for real-world RAG applications.

538 Table 6: Case study about contextual and para- 539 metric knowledge conflicts in 2WQA where only DyPRAG-Combine answers correctly (11.33%). The backbone model is the LLaMA3.2-1B. : deficiency in parametric knowledge, : knowledge conflicts, : successful knowledge manipulation.

540 **Question:** Which film whose director was born first,  
 541 The Snake Brothers or Olympus Has Fallen ?

542 **Ground truth:** Behind Prison Gates

543 **Retrieved top-1 document:** Roman Waugh  
 544 was announced as director for the film.  
 Olympus Has Fallen (film series)...

Method	Answer	Status
Vanilla	David R	
RAG	The Snake Brothers	
DyPRAG (ours)	The Snake Brothers	
DyPRAG-Combine (ours)	Olympus Has Fallen	

540 REFERENCES  
541

542 Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and  
543 extraction. [arXiv preprint arXiv:2309.14316](https://arxiv.org/abs/2309.14316), 2023a.

544 Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation.  
545 [arXiv preprint arXiv:2309.14402](https://arxiv.org/abs/2309.14402), 2023b.

546

547 Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui  
548 Mei, Junfeng Fang, Zehao Li, Furu Wei, et al. Context-dpo: Aligning language models for  
549 context-faithfulness. [arXiv preprint arXiv:2412.15280](https://arxiv.org/abs/2412.15280), 2024.

550 Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican,  
551 George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al.  
552 Improving language models by retrieving from trillions of tokens. In [International conference on  
553 machine learning](https://proceedings.mlr.press/v158/borgeaud22a.html), pp. 2206–2240. PMLR, 2022.

554 Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal,  
555 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are  
556 few-shot learners. [Advances in neural information processing systems](https://proceedings.neurips.cc/paper/2020/file/1457db555f70e2932bcb2ef8f004c51-Paper.pdf), 33:1877–1901, 2020.

557 Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-lora: Instant  
558 transformer adaption. [arXiv preprint arXiv:2506.06105](https://arxiv.org/abs/2506.06105), 2025.

559

560 Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside:  
561 Llms’ internal states retain the power of hallucination detection. [arXiv preprint arXiv:2402.03744](https://arxiv.org/abs/2402.03744),  
562 2024.

563

564 Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and  
565 Dongyan Zhao. xrag: Extreme context compression for retrieval-augmented generation with one  
566 token. [arXiv preprint arXiv:2405.13792](https://arxiv.org/abs/2405.13792), 2024.

567 Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner.  
568 Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. [arXiv  
569 preprint arXiv:1903.00161](https://arxiv.org/abs/1903.00161), 2019.

570

571 James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. IIRC:  
572 A dataset of incomplete information reading comprehension questions. In Bonnie Webber,  
573 Trevor Cohn, Yulan He, and Yang Liu (eds.), [Proceedings of the 2020 Conference on Empirical  
574 Methods in Natural Language Processing \(EMNLP\)](https://aclanthology.org/2020.emnlp-main.86), pp. 1137–1147, Online, November 2020.  
575 Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.86. URL  
576 <https://aclanthology.org/2020.emnlp-main.86/>.

577 Giacomo Frisoni, Alessio Cocchieri, Alex Presepi, Gianluca Moro, and Zaiqiao Meng. To generate or  
578 to retrieve? on the effectiveness of artificial contexts for medical open-domain question answering.  
579 [arXiv preprint arXiv:2403.01924](https://arxiv.org/abs/2403.01924), 2024.

580

581 Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for  
582 context compression in a large language model. [arXiv preprint arXiv:2307.06945](https://arxiv.org/abs/2307.06945), 2023.

583

584 Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle  
585 use a laptop? a question answering benchmark with implicit reasoning strategies. [Transactions of  
586 the Association for Computational Linguistics](https://aclanthology.org/2021.tacl-1.31), 9:346–361, 2021.

587

588 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,  
589 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms  
via reinforcement learning. [arXiv preprint arXiv:2501.12948](https://arxiv.org/abs/2501.12948), 2025.

590 Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented  
591 language model pre-training. In Hal Daumé III and Aarti Singh (eds.), [Proceedings of the 37th  
592 International Conference on Machine Learning](https://proceedings.mlr.press/v119/guu20a.html), volume 119 of [Proceedings of Machine Learning  
593 Research](https://proceedings.mlr.press/v119/guu20a.html), pp. 3929–3938. PMLR, 13–18 Jul 2020a. URL <https://proceedings.mlr.press/v119/guu20a.html>.

594 Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented  
 595 language model pre-training. In *International conference on machine learning*, pp. 3929–3938.  
 596 PMLR, 2020b.

597 Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop  
 598 qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*,  
 599 2020.

600 Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,  
 601 Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

602 Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open  
 603 domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings*  
 604 of the 16th Conference of the European Chapter of the Association for Computational Linguistics:  
 605 Main Volume, pp. 874–880, Online, April 2021. Association for Computational Linguistics. doi:  
 606 10.18653/v1/2021.eacl-main.74. URL <https://aclanthology.org/2021.eacl-main.74>.

607 Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. Worldtree:  
 608 A corpus of explanation graphs for elementary science questions supporting multi-hop inference.  
 609 *arXiv preprint arXiv:1802.03052*, 2018.

610 Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang,  
 611 Jamie Callan, and Graham Neubig. Active retrieval augmented generation. *arXiv preprint*  
 612 *arXiv:2305.06983*, 2023.

613 Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly  
 614 supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan  
 615 (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*  
 616 (Volume 1: Long Papers), pp. 1601–1611, Vancouver, Canada, July 2017. Association for Compu-  
 617 tational Linguistics. doi: 10.18653/v1/P17-1147. URL <https://aclanthology.org/P17-1147>.

618 Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi  
 619 Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv*  
 620 *preprint arXiv:2004.04906*, 2020.

621 Jay Kreps, Neha Narkhede, Jun Rao, et al. Kafka: A distributed messaging system for log processing.  
 622 In *Proceedings of the NetDB*, volume 11, pp. 1–7. Athens, Greece, 2011.

623 Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris  
 624 Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion  
 625 Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav  
 626 Petrov. Natural questions: A benchmark for question answering research. *Transactions of the*  
 627 *Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl\_a\_00276. URL  
 628 <https://aclanthology.org/Q19-1026>.

629 Quinn Leng, Jacob Portes, Sam Havens, Matei Zaharia, and Michael Carbin. Long context rag  
 630 performance of large language models. *arXiv preprint arXiv:2411.03538*, 2024.

631 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal,  
 632 Heinrich Kütter, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented genera-  
 633 tion for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:  
 634 9459–9474, 2020.

635 Huanxuan Liao, Shizhu He, Yao Xu, Yuanzhe Zhang, Yanchao Hao, Shengping Liu, Kang Liu,  
 636 and Jun Zhao. From instance training to instruction learning: Task adapters generation from  
 637 instructions. *Advances in Neural Information Processing Systems*, 37:45552–45577, 2024a.

638 Huanxuan Liao, Shizhu He, Yao Xu, Yuanzhe Zhang, Kang Liu, Shengping Liu, and Jun Zhao.  
 639 Awakening augmented generation: Learning to awaken internal knowledge of large language  
 640 models for question answering. *arXiv preprint arXiv:2403.15268*, 2024b.

641 Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization*  
 642 *branches out*, pp. 74–81, 2004.

648 Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. Towards collaborative neural-symbolic graph semantic  
 649 parsing via uncertainty. *Findings of the Association for Computational Linguistics: ACL 2022*,  
 650 2022.

651

652 Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig.  
 653 Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language  
 654 processing. *ACM Comput. Surv.*, 55(9), jan 2023. ISSN 0360-0300. doi: 10.1145/3560815. URL  
 655 <https://doi.org/10.1145/3560815>.

656

657 Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction.  
 arXiv preprint arXiv:2002.07650, 2020.

658

659 Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi.  
 660 When not to trust language models: Investigating effectiveness of parametric and non-parametric  
 661 memories. arXiv preprint arXiv:2212.10511, 2022.

662

663 Yansheng Mao, Jiaqi Li, Fanxu Meng, Jing Xiong, Zilong Zheng, and Muhan Zhang. Lift: Improving  
 664 long context understanding through long input fine-tuning. arXiv preprint arXiv:2412.13626,  
 2024.

665

666 Meta. Meta-llama-3-8b-instruct, 2024a. URL <https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>. Accessed: 2024-04.

667

668 Meta. Llama-3.2-1b-instruct, 2024b. URL <https://huggingface.co/meta-llama/Llama-3-2-1B-Instruct>. Accessed: 2024-09.

669

670 Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct  
 671 electricity? a new dataset for open book question answering. arXiv preprint arXiv:1809.02789,  
 672 2018.

673

674 Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *Advances  
 675 in Neural Information Processing Systems*, 36:19327–19352, 2023.

676

677 Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and  
 678 Li Deng. Ms marco: A human-generated machine reading comprehension dataset. 2016.

679

680 Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and  
 681 Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented  
 682 language models. arXiv preprint arXiv:2401.00396, 2023.

683

684 Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-  
 685 subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H  
 686 Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference  
 687 on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*,  
 688 pp. 248–260. PMLR, 07–08 Apr 2022. URL <https://proceedings.mlr.press/v174/pal22a.html>.

689

690 Jörg Polzehl and Vladimir Spokoiny. Propagation-separation approach for local likelihood estimation.  
 Probability Theory and Related Fields, 135:335–362, 2006.

691

692 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea  
 693 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances  
 694 in neural information processing systems*, 36:53728–53741, 2023.

695

696 Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for  
 697 machine comprehension of text. arXiv preprint arXiv:1606.05250, 2016.

698

699 Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and  
 700 Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association  
 701 for Computational Linguistics*, 11:1316–1331, 2023.

702

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and  
 703 Peter J Liu. Out-of-distribution detection and selective generation for conditional language models.  
 arXiv preprint arXiv:2209.15558, 2022.

702 Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond.  
 703 *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.  
 704

705 Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael  
 706 Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In  
 707 *International Conference on Machine Learning*, pp. 31210–31227. PMLR, 2023.

708 Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval  
 709 augmented generation based on the real-time information needs of large language models. In  
 710 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting  
 711 of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12991–13013,  
 712 Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/  
 713 2024.acl-long.702. URL <https://aclanthology.org/2024.acl-long.702/>.

714 Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia  
 715 Zhou, and Yiqun Liu. Parametric retrieval augmented generation. *arXiv preprint arXiv:2501.15915*,  
 716 2025.

717 Zhongxiang Sun, Xiaoxue Zang, Kai Zheng, Yang Song, Jun Xu, Xiao Zhang, Weijie Yu, and Han Li.  
 718 Redep: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability.  
 719 *arXiv preprint arXiv:2410.11414*, 2024.

720

721 Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions.  
 722 *arXiv preprint arXiv:1803.06643*, 2018.

723

724 Yufei Tao, Adam Hiatt, Erik Haake, Antonie J Jetter, and Ameeta Agrawal. When context leads but  
 725 parametric memory follows in large language models. *arXiv preprint arXiv:2409.08435*, 2024.

726

727 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz  
 728 Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing  
 729 systems*, 30, 2017.

730

731 Changyue Wang, Weihang Su, Qingyao Ai, and Yiqun Liu. Knowledge editing through chain-of-  
 732 thought. *arXiv preprint arXiv:2412.17727*, 2024a.

733

734 Changyue Wang, Weihang Su, Hu Yiran, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min  
 735 Zhang, and Shaoping Ma. Lekube: A legal knowledge update benchmark. *arXiv preprint  
 736 arXiv:2407.14192*, 2024b.

737

738 Yan Wang, Dongyang Ma, and Deng Cai. With greater text comes greater necessity: Inference-time  
 739 training helps long text generation. In *First Conference on Language Modeling*, 2024c. URL  
 740 <https://openreview.net/forum?id=dj9x6JuID5>.

741

742 Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. *arXiv  
 743 preprint arXiv:2409.07429*, 2024d.

744

745 Zhepei Wei, Wei-Lin Chen, and Yu Meng. Instructrag: Instructing retrieval-augmented generation  
 746 via self-synthesized rationales. *arXiv preprint arXiv:2406.13629*, 2024.

747

748 Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin,  
 749 Rashi Rungta, Karthik Abinav Sankararaman, Barlas Ooguz, et al. Effective long-context scaling of  
 750 foundation models. *arXiv preprint arXiv:2309.16039*, 2023.

751

752 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,  
 753 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint  
 754 arXiv:2412.15115*, 2024.

755

756 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang  
 757 Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*,  
 758 2025.

759

760 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov,  
 761 and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question  
 762 answering. *arXiv preprint arXiv:1809.09600*, 2018.

756 Hao Zhang, Yuyang Zhang, Xiaoguang Li, Wexuan Shi, Haonan Xu, Huanshuo Liu, Yasheng Wang,  
757 Lifeng Shang, Qun Liu, Yong Liu, et al. Evaluating the external and parametric knowledge fusion  
758 of large language models. [arXiv preprint arXiv:2405.19010](https://arxiv.org/abs/2405.19010), 2024.  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809

810  
 811 Table 7: Comparison of cost metrics for different baselines. ATTN denotes the time complexity of  
 812 the self-attention module as  $O(|I|^2h)$ , and FFN represents the FFN with  $O(|I|h^2)$ , where context  
 813 length  $|I| = 1$  and  $|R|$  denotes the response length. █ indicates significantly high cost,   denotes  
 814 negligible cost, and   represents temporal storage.

Method	Inference Cost	Training Cost	Storage Cost
RAG	<span style="background-color: red; color: white; padding: 2px 5px;">█</span> $ R  \times ((c d  +  q )^2 \times \text{ATTN} + (c d  +  q ) \times \text{FFN})$	$M \times (9 d ^2 \times \text{ATTN} + 3 d  \times \text{FFN}) +$	<u> </u>
PRAG	$ R  \times ( q ^2 \times \text{ATTN} +  q  \times \text{FFN})$	<span style="background-color: red; color: white; padding: 2px 5px;">█</span> $M \times E_1 \times (81 d ^2 \times \text{ATTN} + 9 d  \times \text{FFN})$	<span style="background-color: red; color: white; padding: 2px 5px;">█</span> $M \times 3Lr(h + k)$
DyPRAG	$c \times ( d ^2 \times \text{ATTN} +  d  \times \text{FFN}) +$ $c \times O(p(h + 1 + hr)) +$ $ R  \times ( q ^2 \times \text{ATTN} +  q  \times \text{FFN})$	$N \times (9 d ^2 \times \text{ATTN} + 3 d  \times \text{FFN}) +$ $N \times E_1 \times (81 d ^2 \times \text{ATTN} + 9 d  \times \text{FFN}) +$ $N \times E_2 \times (9( qa  +  d )^2 \times \text{ATTN} + 3( qa  +  d ) \times \text{FFN}) + O(p(h + 1 + hr))$	$\frac{N}{3} \times 3Lr(h + k) +$ $3L(phr + 2p(h + 1) + pkr)$

## A COMPUTATION AND STORAGE COST ANALYSIS

We present an initial pilot analysis and a broad evaluation of computation and storage costs across three baseline methods. More detailed analysis of time complexity is provided in Appendix A.1.

**Computation Cost.** The computation cost in RAG is primarily the inference cost, whereas PRAG introduces additional training and inference costs due to augmentation and offline training. Suppose the average token count of document  $d$  is  $|d|$ . As noted in Su et al. (2025), the augmentation process typically generates about  $2|d|$  tokens, leading to an augmentation cost of  $3|d|$ . When training the target LoRA, a forward pass over  $3|d|$  tokens and a backward pass over  $6|d|$  tokens (typically twice the forward cost) result in a total training cost of  $9|d|$ . Although these tasks can be performed offline, it still requires a long time and do not generalize to new questions with unseen documents. In contrast, DyPRAG offers a more practical solution by requiring only  $N$  Doc-Param pairs while even a small  $N$  can achieve powerful performance, significantly reducing costs for augmentation and training. The cost of MLP-based  $\mathcal{F}'_\phi$  is negligible compared to transformer-based LLMs (Vaswani et al., 2017).

The primary advantage of PRAG is the reduction of inference cost. Let  $|q|$  denotes the length of the question,  $c$  represents the number of retrieved documents. The inference context of PRAG and DyPRAG is  $|q|$ , whereas RAG requires  $c|d| + |q|$ . The parameterization process significantly reduces the inference cost, especially when  $|d|$  and  $c$  grow larger. Notably, the inference cost is also closely tied to the length of model response. DyPRAG demonstrates an improved ability to internalize knowledge, resulting in shorter responses that effectively reduce costs, as shown in Figure 4.

**Storage Cost.** One of the main shortcomings of PRAG is the storage cost associated with  $\mathbf{P}_i$ . Let  $r$  denote the LoRA rank,  $L$  the number of Transformer layers,  $h$  the hidden size, and  $k$  the intermediate size of the FFN. The number of parameters in the parametric representation of a document is  $3Lr(h + k)$ . For instance, in the Qwen2.5-1.5B model (which has 28 layers, a hidden dimension of 1536, and an intermediate size of 8960), setting  $r$  to 2 results in approximately 1.76M parameters, storing 3.36MB in 16-bit precision for each parametric representation. In our following experiments, we need to store 9.33GB offline parameters for Qwen2.5-1.5B, presenting a significant storage cost.

In contrast, our DyPRAG only needs to save the weights of  $\mathcal{F}'_\phi$ . As we set the intermediate size  $p$  of the  $\mathcal{F}'_\phi$  to 2, the total number of parameters for the Qwen2.5-1.5B model is  $3L(phr + 2p(h + 1) + pkr)$  as we configure separate translators for up-proj, down-proj, and gate-proj. This amounts to about 4.04M parameters, storing only 7.71MB (0.08% of PRAG) in 16-bit precision. The reduced storage cost makes it negligible compared to its generalization ability when used in real applications.

### A.1 DETAILED COST COMPARISON

In this section, we provide a detail comparison of several cost metrics for standard RAG, PRAG and our proposed DyPRAG, as shown in Table 7.

**Inference Cost.** We first analyze the inference cost across three baselines. Intuitively, the RAG method requires more resources for inference due to its context length of  $c|d| + |q|$ , compared to only  $|q|$  for PRAG and DyPRAG. In our experimental settings,  $|q|$  is usually less than 100, while  $|d|$  is typically larger than 600, with  $c$  set to 3. This results in an attention cost of at least 271x and a FFN cost of 19x for RAG. For DyPRAG, there is additional cost incurred for encoding and

864 translating. The encoding cost is  $c \times (|d|^2 \times \text{ATTN} + |d| \times \text{FFN})$ , as each document should be  
 865 encoded separately. As shown in Table 11, the encoding time is significantly lower than the inference  
 866 time because encoding requires only a single forward pass. Additionally, the translation time is also  
 867 negligible. Moreover, the response length  $|R|$  exhibits a linear relationship with the LLM inference  
 868 loss. As illustrated in Figure 4, the response length decreases when DyPRAG is employed, enabling  
 869 LLMs to better internalize knowledge. Notably, DyPRAG-Combine achieves much shorter response  
 870 lengths, significantly reducing inference costs compared to standard RAG.

871  
 872 **Training Cost.** PRAG (Su et al., 2025) introduces further training for each document to obtain  
 873 corresponding LoRA parameters. In Section A, we hypothesize that after augmentation, there are  
 874 a total of  $3|d|$  tokens, resulting in a cost of  $N \times (9|d|^2 \times \text{ATTN} + 3|d| \times \text{FFN})$  for DyPRAG  
 875 and  $M \times (9|d|^2 \times \text{ATTN} + 3|d| \times \text{FFN})$  for PRAG, where  $N$  represents the size of the training  
 876 dataset  $\mathcal{K}$  and  $M$  denotes the size of the test set. The common divisor of offline parametrization is  
 877  $E_1 \times (81|d|^2 \times \text{ATTN} + 9|d| \times \text{FFN})$ , where  $E_1$  is the number of epochs for LoRA training.  
 878

879 Additionally, to train our  $\mathcal{F}'_\phi$  for  $E_2$  epochs, we need to perform both forward and backward passes  
 880 (the backward pass requires twice the cost of the forward pass) on one QA pair and its corresponding  
 881 document in each step. This results in a cost of  $N \times E_2 \times 9(|qa| + |d|)^2 \times \text{ATTN} + 3(|qa| + |d|) \times \text{FFN}$ ,  
 882 with a negligible cost for translation. As shown in Figure 6 and 7, our DyPRAG achieves stable  
 883 results with as few as 480 examples (even fewer is powerful), while  $M = 3000$  in our experiments,  
 884 and this value would be significantly larger in real-world applications.  
 885

886 For instance, using LLaMA3-8B as the backbone, producing a  $\mathbf{P}_i$  requires 88 seconds, while one  
 887 step for  $\mathcal{F}'_\phi$  only takes an average of 15 seconds. Therefore, the total cost for training (excluding  
 888 augmentation) is  $M \times 88s$  in PRAG and  $N \times 103s$  in DyPRAG. Assuming  $N = 480$  and  $M = 3000$ ,  
 889 DyPRAG is 5.34x faster than PRAG. The low requirement for a large  $N$  makes DyPRAG highly  
 890 effective and generalizable for real-world scenarios, with extremely low costs that can be handled  
 891 during offline training.  
 892

893 **Storage Cost.** As illustrated in Section A, each  $\mathbf{P}_i$  requires 3.36MB for PRAG using Qwen2.5-1.5B,  
 894 resulting in a total storage cost of 9.33GB in our main experiment. However, we significantly reduce  
 895 this cost by imitating the underlying function between the document and parameters. Notably, the  
 896 cost for  $\mathbf{P}_i$  is a temporary cost in DyPRAG, which can be removed after collecting data or training  
 897 one  $\mathbf{P}_i$  and then updating  $\mathcal{F}'_\phi$  by one step. Consequently, the overall cost of DyPRAG is substantially  
 898 lower than that of PRAG (e.g., DyPRAG achieves better performance with only 7.71MB of storage as  
 899 shown in Table 11).  
 900

## 901 B EXPERIMENT SETUP

### 902 B.1 IMPLEMENTATION DETAILS

903 **QA Datasets.** To ensure a comprehensive evaluation, we assess our method using the following  
 904 datasets:

- 905 • **2WikiMultihopQA (2WQA)** (Ho et al., 2020) is designed to evaluate a model’s capability in  
 906 multi-hop reasoning by synthesizing information from multiple Wikipedia passages.
- 907 • **HotpotQA (HQA)** (Yang et al., 2018) similarly targets multi-hop reasoning, requiring models to  
 908 amalgamate information from various contexts to answer a single query.
- 909 • **PopQA (PQA)** (Mallen et al., 2022) focuses on factual question answering, posing challenges that  
 910 test the model’s ability to recall precise knowledge and navigate ambiguities in entity representation.
- 911 • **ComplexWebQuestions (CWQ)** (Talmor & Berant, 2018) entails answering complex, multi-step  
 912 questions sourced from the web, further challenging the model’s capacity to retrieve and reason  
 913 over extensive web content.

914 **Offline Doc-Param Pairs Collection.** Following (Jiang et al., 2023; Su et al., 2025), we utilize  
 915 Wikipedia dumps as the external knowledge corpus, adopting the dataset proposed by DPR (Karpukhin  
 916 et al., 2020). For document augmentation, each document is rewritten once, and three QA pairs are  
 917 generated based on the document. Unless explicitly stated otherwise, the downstream LLM is used

918 for this purpose. During LoRA fine-tuning, the learning rate was set to  $3 \times 10^{-4}$ , and training was  
 919 conducted for a single epoch (except PQA for 2). The LoRA modules were integrated exclusively  
 920 into the feed-forward network (FFN) matrices, while the query, key, and value (QKV) matrices were  
 921 excluded. The scaling factor  $\alpha$  was set to 32, the LoRA rank  $r$  was configured to 2, and no dropout  
 922 was applied to ensure training stability and maximize parameter updates. The LoRA weights were  
 923 randomly initialized following the settings outlined in the original LoRA paper (Hu et al., 2022).  
 924

925 **Baselines Implementation.** To conduct comprehensive experiments, we compare our DyPRAG  
 926 with two commonly used baselines: SFT and Context-DPO, alongside parametric and non-parametric  
 927 RAG baselines. For SFT, widely regarded as a standard approach for adapting models to various  
 928 downstream tasks, is included to evaluate the generalization ability of DyPRAG. Specifically, we  
 929 use the exact same hyperparameters as DyPRAG, setting the learning rate to  $3 \times 10^{-4}$ , fine-tuning  
 930 on the same dataset (i.e., 36,000 samples) with a batch size of 1 for 1 epoch. For Context-DPO, we  
 931 follow the implementation described in Bi et al. (2024). To ensure a fair comparison, we configure  
 932 the trainable LoRA modules for both methods to match those in DyPRAG, maintaining equivalent  
 933 parameter learning capacity. The LoRA modules are integrated exclusively into the FFN, while the  
 934 query, key, and value matrices are excluded. The scaling factor  $\alpha$  is set to 32, and the LoRA rank  $r$  is  
 935 configured as 2.  
 936

936 **Inference Settings.** All experiments use the publicly available Hugging Face implementations of  
 937 LLaMA and Qwen. To ensure fairness, DyPRAG and all baselines share the same prompt template in  
 938 Figure 11 and 12 following Su et al. (2025) and adopt of greedy decoding for result reproducibility.  
 939 The max number of new tokens is set to 128.  
 940

941 **Retrieval Module  $\mathcal{R}$ .** Recent research on retrieval-augmented generation (RAG) (Ram et al.,  
 942 2023) has shown that BM25 matches or even surpasses state-of-the-art dense retrieval models in  
 943 certain scenarios. Following Su et al. (2025), we adopt BM25 as the retriever in our approach and  
 944 Elasticsearch is used as the backend for implementing BM25.  
 945

946 **Training  $\mathcal{F}'_\phi$ .** Motivated by Liao et al. (2024a), we use simple MLP hypernetwork to transform  
 947 embedding into adapter parameters. Through cross validation, the learning rate was set to  $1 \times 10^{-5}$ ,  
 948 and the training epoch was set to 1 which making the overall alignment process quickly. The  
 949 truncation max length of text is set to 3000, which is larger than most retrieved documents. The  
 950 performance reports for Qwen2.5-1.5B and LLaMA3.2-1B in Table 1 are based on training with 4,800  
 951 examples, while LLaMA3-8B is trained on 2,400 examples (except for 480 examples on 2WQA).  
 952

952 **Implementation of OOD Experiment.** To evaluate the generalization ability of our proposed  
 953 DyPRAG, we select to out-of-distribution (OOD) datasets to conduct.  
 954

- 955 • **StrategyQA (SQA)** (Geva et al., 2021): A QA benchmark where reasoning steps are implicit in  
 956 the question and must be inferred through strategic reasoning, including human-curated evidence  
 957 paragraphs from Wikipedia.
- 958 • **IIRC** (Ferguson et al., 2020): A dataset comprising over 13,000 questions based on English  
 959 Wikipedia paragraphs that provide only partial information and supplemented with samples from  
 960 SQuAD 2.0 (Rajpurkar et al., 2016) and DROP (Dua et al., 2019), requiring retrieval of missing  
 961 details from linked documents.
- 962 • **OpenBookQA (OBQA)** (Mihaylov et al., 2018): A multiple-choice QA dataset derived from a  
 963 subset of WorldTree (Jansen et al., 2018), mainly focus on common knowledge.
- 964 • **MedMCQA (MQA)** (Pal et al., 2022): A multiple-choice QA dataset designed to address real-  
 965 world medical domain entrance exam questions.
- 966 • **CNNDailymail<sup>1</sup>**: A summarization datasets containing just over 300k unique news articles as  
 967 written by journalists at CNN and the Daily Mail.  
 968

969 For each dataset, we select the first 300 examples for testing and evaluate performance using F1  
 970 score for IIRC, Accuracy for SQA, Recall for OBQA and MQA and Rouge-L (Lin, 2004) for  
 971

<sup>1</sup>[https://huggingface.co/datasets/ccdv/cnn\\_dailymail](https://huggingface.co/datasets/ccdv/cnn_dailymail)

972  
 973 Table 8: The experimental results of DyPRAG are compared with other effective RAG methods. All  
 974 metrics are reported as F1 scores (%). The best performance is bolded, while the second-best is  
 975 underlined. The evaluation is conducted on 2WQA and HQA datasets, focusing exclusively on the  
 976 total sub-task.

977	978	Base LLM	Method	2WQA	HQA	Avg
				Total	Total	
979	980	LLaMA3.2-1B	RAG	23.12	<b>27.14</b>	<u>25.13</u>
			DRAGIN	21.73	12.50	17.12
			FLARE	21.55	19.38	20.47
			DyPRAG (ours)	<u>25.31</u>	19.97	22.64
981	982	Qwen2.5-1.5B	DyPRAG-Combine (ours)	<b>29.18</b>	<u>26.58</u>	<b>27.88</b>
			RAG	24.31	<u>20.73</u>	<u>22.52</u>
			DRAGIN	25.01	8.51	16.76
			FLARE	21.56	7.97	14.77
983	984	LLaMA3-8B	DyPRAG (ours)	<b>26.46</b>	19.67	23.07
			DyPRAG-Combine (ours)	<u>25.18</u>	<b>27.57</b>	<b>26.38</b>
			RAG	34.55	24.23	29.39
			DRAGIN	35.69	12.16	23.93
985	986	LLaMA3-8B	FLARE	34.62	<u>29.43</u>	<u>32.03</u>
			DyPRAG (ours)	<u>37.25</u>	22.55	29.90
			DyPRAG-Combine (ours)	<b>45.17</b>	<b>38.35</b>	<b>41.76</b>

991  
 992 Table 9: The experimental results of DyPRAG are compared with standard RAG based on Qwen3-8B  
 993 and Qwen3-4b-Instruct. All metrics are reported as EM scores (%) and F1 scores (%). The best  
 994 performance is bolded, while the second-best is underlined. The Avg is the average performance over  
 995 all tasks.

996	997	Base LLM	Method	2WQA (Total)		HQA (Total)		PQA		CWQ		Avg	
				EM	F1								
998	999	Qwen3-8B	Vanilla	24.67	31.33	21.00	28.12	0.00	0.40	<u>22.33</u>	<b>36.01</b>	17.00	23.97
			RAG	<b>35.33</b>	<u>42.26</u>	<u>32.33</u>	<b>44.00</b>	<b>0.33</b>	<b>9.17</b>	<u>23.00</u>	<u>35.79</u>	<b>22.75</b>	<b>32.81</b>
			DyPRAG	21.00	27.94	20.33	27.82	0.00	0.46	17.67	29.12	14.75	21.34
			DyPRAG-Combine	<u>31.00</u>	<b>38.37</b>	<u>29.67</u>	<u>39.81</u>	<b>0.33</b>	<u>4.54</u>	20.00	31.37	<u>20.25</u>	<b>28.52</b>
1000	1001	Qwen3-4B-Instruct	Vanilla	21.00	28.97	15.00	23.32	8.67	12.10	0.00	1.58	11.17	16.49
			RAG	25.67	32.81	<u>25.33</u>	<u>36.62</u>	<u>18.67</u>	<u>26.32</u>	<u>2.00</u>	<u>7.36</u>	<u>17.92</u>	<u>25.78</u>
			DyPRAG	<u>27.00</u>	<u>35.44</u>	16.33	24.00	10.00	13.49	0.33	4.47	13.42	19.35
			DyPRAG-Combine	<b>31.00</b>	<b>38.37</b>	<b>29.67</b>	<b>39.81</b>	<b>20.67</b>	<b>27.33</b>	<b>8.67</b>	<b>19.14</b>	<b>22.50</b>	<b>31.16</b>

1004 CNNDailymail (except for 100 examples) as metrics. Both datasets provide with ground-truth  
 1005 passages which indicate a more rigorous evaluation setting. For IIRC, we adopt the few-shot prompts  
 1006 from Su et al. (2024), while SQA, OBQA, MQA and CNNDailymail are evaluated in a zero-shot  
 1007 setting. Notably, the same prompt format (in Figure 11 and 12) from the main experiment is used to  
 1008 ensure a fair comparison, except CNNDailymail using summarization template in Figure 13.

1009  
 1010 **Implementation of RAGTruth Experiment.** RAGTruth (Niu et al., 2023) is a benchmark dataset  
 1011 designed to evaluate the extent of hallucination in models. For our evaluation, we randomly select 100  
 1012 QA-type subsets from RAGTruth, ensuring alignment with the training data of  $\mathcal{F}'_\phi$ . Notably, some  
 1013 questions in RAGTruth require the provided documents to be answerable which are more difficult.  
 1014 Interestingly, during evaluation, we observe that  $\mathcal{F}'_\phi$  with fewer trained parameters perform better in  
 1015 such scenarios. Specifically, we train only 480 examples for LLaMA3.2-1B and Qwen2.5-1.5B, and  
 1016 240 examples for LLaMA3-8B. We use GPT-40 as judge using prompt template in Figure 14.

## C SUPPLEMENT EXPERIMENT RESULTS

1018  
 1019 **Comparison with effective RAG baselines.** To compare our DyPRAG with effective RAG methods,  
 1020 we introduce two powerful baselines:

1021  
 1022 • FLARE (Jiang et al., 2023) is a multi-round retrieval augmentation method that triggers retrieval  
 1023 whenever it encounters an uncertain token. The query is defined as the last generated sentence  
 1024 excluding the uncertain tokens.

1026 • DRAGIN (Su et al., 2024) improves multi-round retrieval by triggering only when an uncertain  
 1027 token has semantic significance and strongly influences subsequent tokens. It formulates queries  
 1028 using the model’s internal state and preceding context.

1029 The experimental results are presented in Table 8. Compared to standard RAG, DRAGIN and  
 1030 FLARE do not demonstrate significant performance advantages when the model size is smaller  
 1031 (e.g., LLaMA3.2-1B and Qwen2.5-1.5B). However, as the model size increases (e.g., LLaMA3-8B),  
 1032 DRAGIN achieves the best performance on the 2WQA dataset, while FLARE performs best on  
 1033 the HQA dataset comparing with RAG baseline. This indicates that effective RAG methods are  
 1034 often constrained by the model’s inherent capabilities and lack robust generalization. In contrast,  
 1035 our proposed DyPRAG consistently delivers superior performance in most cases, demonstrating the  
 1036 effectiveness of our approach. Furthermore, when combined with in-context injection, DyPRAG  
 1037 achieves an average improvement of 6.54% over standard RAG, highlighting the substantial potential  
 1038 of integrating parametric knowledge with contextual knowledge.

1039 **OOD Performance in Summarization Task.** Since our training primarily focused on QA data,  
 1040 we are curious whether the parameter translator  $\mathcal{F}'_\phi$  can generalize effectively to other tasks. To  
 1041 evaluate DyPRAG’s performance on non-QA tasks, we conducted additional experiments using the  
 1042 CNNDailyMail dataset for a summarization task. This task employed the prompt template outlined  
 1043 in Figure 13 and was evaluated using the Rouge-L (Lin, 2004). As presented in Table 10, we  
 1044 utilized the parameter translator to first transform documents into parametric knowledge, enhancing  
 1045 the overlap between knowledge and model inputs. This step encouraged LLMs to better leverage  
 1046 document-specific knowledge. Across all model scales, our DyPRAG-Combine approach achieved  
 1047 an average improvement of 0.21 in performance, demonstrating that our method is effective beyond  
 1048 QA tasks. This result highlights the capability of our approach to perform general mapping from  
 1049 textual embeddings to the parametric space.

1050 **Comparison of Response Length.** Notably, we consider only the context length when calculating  
 1051 inference cost. However, in practice, the response length from LLMs also affects inference time. As  
 1052 shown in Figure 4, we compare DyPRAG-Combine with RAG across four benchmarks, considering  
 1053 the average response length. DyPRAG-Combine significantly reduces response length, by 20% in  
 1054 2WQA and up to 90% in CWQ. This demonstrates that DyPRAG-Combine can answer questions  
 1055 correctly with fewer tokens, thereby lowering inference costs and avoiding redundant information.

1056 **Performance of DyPRAG on Non-Instruct Models.** With the rapid advancement of reinforcement  
 1057 learning, a growing number of long-context models, referred to as large reasoning models (LRMs)  
 1058 have emerged (Guo et al., 2025; Yang et al., 2025). Our goal is to evaluate whether the current  
 1059 design of DyPRAG can adapt effectively to such up-to-date models. For this purpose, we selected  
 1060 Qwen3-8B<sup>2</sup> (a reasoning model) and Qwen3-4B-Instruct<sup>3</sup> (an instruct model) for experiments. As  
 1061 shown in Table 9, the performance of Qwen3-8B decreases significantly when DyPRAG generated  
 1062 parameters are applied. This decline is primarily due to differences in answer patterns. LRM  
 1063 tend to generate extremely lengthy reasoning trajectories, whereas our method only augments simple  
 1064 and short QA pairs. In contrast, the results for the instruct model, Qwen3-4B-Instruct, align with  
 1065 our main experiments, demonstrating that the current method is well-suited for instruct models. To  
 1066 enable compatibility with LRM, the parameter translation process needs to be integrated into the  
 1067 reinforcement learning training pipeline. Addressing this challenge will be a focus of our future work.

## 1073 D ADDITIONAL ABLATION EXPERIMENT RESULTS

1074 **Effect of Training Dataset Size.** We adjust the pre-selected size of the training dataset composed  
 1075 of Doc-Param pairs, increasing it from 480 to 4800. As shown in Figure 6 and 7, DyPRAG achieves  
 1076 strong performance even with just 480 training examples. The performance remains remarkably stable

1077 <sup>2</sup><https://huggingface.co/Qwen/Qwen3-8B>

1078 <sup>3</sup><https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507>

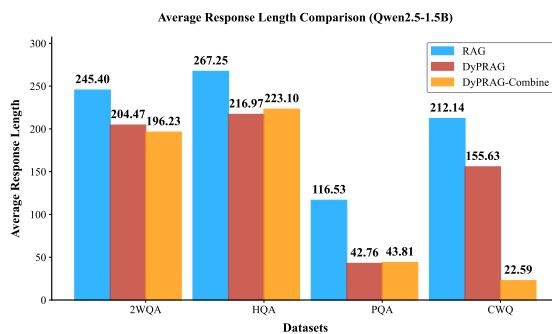


Figure 4: Comparison of response length across various datasets. The backbone model is the Qwen2.5-1.5B.

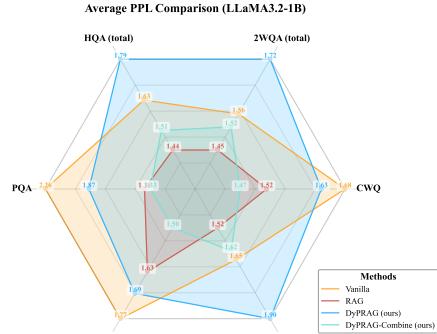


Figure 5: Comparison of average PPL. Smaller PPL means less conflicts. The backbone model is the LLaMA3.2-1B.

Table 10: The OOD performance on summarization dataset CNNDailymail. The metric is reported as Rouge-L (%) (Lin, 2004).

Base LLM	Method	CNNDailymail
LLaMA3.2-1B	RAG	21.09
	DyPRAG-Combine	<b>21.21</b>
Qwen2.5-1.5B	RAG	19.86
	DyPRAG-Combine	<b>20.38</b>
LLaMA3-8B	RAG	<b>22.34</b>
	DyPRAG-Combine	22.33

across different dataset sizes, indicating that our design,  $\mathcal{F}'_\phi$ , is capable of learning the underlying mapping between documents and parameters with minimal data.

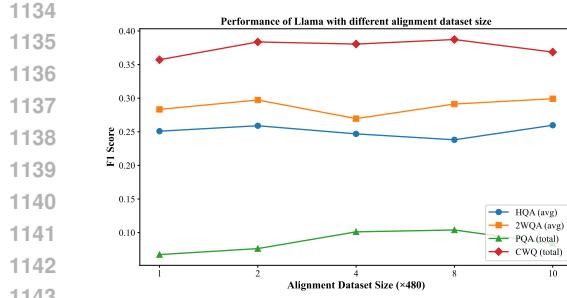
**Performance Effect of Retrieved Documents Number.** For standard RAG, the number of retrieved documents, denoted as  $c$ , is a crucial hyperparameter to tune. Recent studies (Leng et al., 2024; Wei et al., 2024) have investigated the impact of longer context lengths on standard RAG. **As shown in Figure 8, the performance fluctuates as the number of retrieved documents increases, with the best value generally achieved at  $c = 3$ . This demonstrates that introducing more less-relevant context can negatively impact the model’s ability to extract key information.**

However, the effect of the number of injected documents in parametric form remains underexplored. Our proposed DyPRAG framework can seamlessly adapt to this scenario due to its inherent flexibility. As shown in Figure 9, the performance of DyPRAG does not significantly improve as the number of injected documents increases. For instance, in the 2WQA and CWQ datasets, the best performance is achieved when using only the top-1 document. This indicates that the most relevant document, as determined by the retriever  $\mathcal{R}$ , is sufficient to provide the knowledge needed to answer the question effectively. On the other hand, in datasets such as HQA and PQA, the best performance is observed when  $c = 3$ , suggesting that when more relevant information is retrieved, simple averaging of LoRA parameters can effectively integrate the knowledge. Additionally, in three out of four datasets (except PQA), the model’s performance declines when too many documents are injected. This observation aligns with the findings in Shi et al. (2023), which suggest that task-irrelevant redundant information can degrade the model’s performance, especially the compression of documents is lossy.

**Computation Effect of Injected Documents Number.** We have specifically designed the code of DyPRAG to enable the rapid loading of document-specific LoRA modules, ensuring minimal delays during operation. As demonstrated in Table 12, DyPRAG achieves superior inference efficiency compared to standard RAG, particularly as the number of injected documents increases. While the inference time of standard RAG grows significantly with more injected documents, DyPRAG

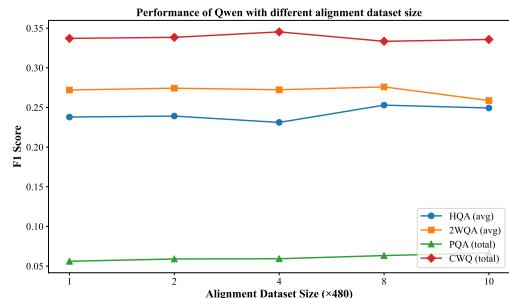
Table 11: Ablation study of intermediate dimension  $p$  of  $\mathcal{F}'_\phi$ . The backbone model is the Qwen2.5-1.5B. The inference time is computed by average time of CWQ with batch\_size of 1. The encode time is highlighted in red, while the translate time is marked in blue.

Method	CWQ F1	Inference Time (s)	Storage Cost (MB)
Vanilla	26.47	0.56 (0.47x)	-
RAG	28.32	1.20 (1x)	-
PRAG	30.82	0.56 (0.47x)	19107.84 (1x)
DyPRAG ( $p = 2$ )	32.66	0.56+0.13+0.060 (0.625x)	7.71 (0.04%)
DyPRAG ( $p = 4$ )	33.26	0.56+0.13+0.062 (0.627x)	15.42 (0.08%)
DyPRAG ( $p = 16$ )	32.08	0.56+0.13+0.055 (0.621x)	61.70 (0.32%)
DyPRAG ( $p = 32$ )	31.94	0.56+0.13+0.060 (0.625x)	123.39 (0.64%)



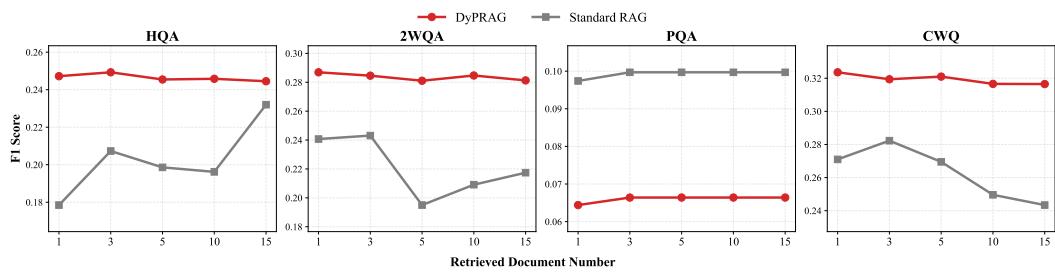
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157

Figure 6: Ablation study of varying training dataset size for DyPRAG. The backbone model is the LLaMA3.2-1B.



1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157

Figure 7: Ablation study of varying training dataset size for DyPRAG. The backbone model is the Qwen2.5-1.5B.



1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171

Figure 8: Ablation study of varying number of retrieved documents to RAG's performance. The backbone model is the Qwen2.5-1.5B.

1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179

Figure 9: Ablation study of varying number of injected documents to DyPRAG's performance. The backbone model is the Qwen2.5-1.5B.

Combine maintains consistently lower inference times due to its shorter response lengths (as shown in Figure 4). Furthermore, DyPRAG significantly outperforms standard RAG in inference time which passages are excluded from the context.

However, the design of DyPRAG introduces an increase in encoding and translation time as the injected documents number grows. Currently, the encoding and translation processes are not fully optimized. In real-world applications, RAG-based queries are typically managed via message queues (e.g., Kafka (Kreps et al., 2011)), which provide a natural asynchronous execution environment. By leveraging this architecture, document embeddings can be extracted and transformed during the waiting period using separate process and model instance. Consequently, when the query reaches the processing stage, there is no additional encoding or translation delay. This allows DyPRAG to outperform standard RAG in both performance and inference efficiency.

1188  
1189 Table 12: Ablation study of varying number of injected documents to computation cost. The backbone  
1190 model is the Qwen2.5-1.5B.

1191	Documents	Method	Inference Time	Loading Time	Translate Time	Encode Time
1192	3	DyPRAG	<b>0.84</b>	0.0037	0.056	0.132
		RAG	1.23	-	-	-
		DyPRAG-Combine	<b>0.36</b>	0.0037	0.055	0.132
1193	10	DyPRAG	<b>0.80</b>	0.0044	0.185	0.433
		RAG	1.54	-	-	-
		DyPRAG-Combine	<b>0.78</b>	0.0045	0.185	0.432
1194	20	DyPRAG	<b>0.80</b>	0.0057	0.361	0.862
		RAG	1.74	-	-	-
		DyPRAG-Combine	<b>1.40</b>	0.0057	0.361	0.862
1195	30	DyPRAG	<b>0.80</b>	0.0067	0.545	1.295
		RAG	2.18	-	-	-
		DyPRAG-Combine	<b>1.96</b>	0.0067	0.545	1.294

1203  
1204 Table 13: **Ablation study of retriever.** All metrics are reported as EM scores (%) and F1 scores (%).  
1205 The backbone model is the LLaMA3-8B.

1206	Method	Retriever	2WQA		HQA		PQA		CWQ		Avg	
			EM	F1								
1207	Vanilla	None	30.00	36.43	19.89	28.64	4.67	7.96	30.00	42.44	24.43	31.85
1208	RAG	Sparse	<b>28.40</b>	<b>34.20</b>	<b>19.13</b>	<b>28.67</b>	5.67	<b>16.13</b>	<b>25.33</b>	35.45	<b>23.04</b>	<b>30.86</b>
		Dense	22.20	27.82	11.66	20.60	<b>6.33</b>	12.15	24.67	<b>36.48</b>	17.70	24.95
1209	DyPRAG	Sparse	<b>32.07</b>	<b>39.17</b>	<b>24.67</b>	<b>37.33</b>	<b>11.00</b>	<b>13.60</b>	<b>32.67</b>	<b>41.87</b>	<b>27.80</b>	<b>36.23</b>
		Dense	22.20	28.48	15.67	23.34	8.33	11.09	30.33	41.10	19.67	26.46
1210	DyPRAG-Combine	Sparse	<b>36.33</b>	<b>47.68</b>	<b>33.22</b>	<b>43.22</b>	<b>21.00</b>	<b>32.86</b>	<b>29.67</b>	<b>39.07</b>	<b>33.20</b>	<b>43.69</b>
		Dense	23.73	28.92	11.44	21.10	7.33	14.85	26.67	38.97	18.70	26.17

1214  
1215 **Effect of Different Retriever.** Retrieval plays a critical role in RAG by determining whether the  
1216 retrieved documents contain the necessary information to answer a given question. In the field of  
1217 information retrieval, the two dominant retrieval methods are lexical matching(Robertson et al., 2009)  
1218 and dense retrieval(Su et al., 2024). Among lexical matching techniques, BM25 stands out for its  
1219 widespread adoption and proven effectiveness. In contrast, despite advancements in dense retrieval  
1220 methods, none have achieved the same level of popularity or reliability as BM25. To explore the  
1221 performance of these approaches, we employ the well-known all-MiniLM-L6-v2<sup>4</sup> model as the  
1222 dense retriever which maps sentences into a 384-dimensional dense vector space, enabling dense  
1223 retrieval tasks. As shown in Table 13, our experiments reveal that BM25 consistently outperforms  
1224 dense retrieval methods across various datasets within the DyRPAG framework, despite the dense  
1225 retrieval methods often excel in many other information retrieval tasks. These findings align with prior  
1226 research (Su et al., 2024; Ram et al., 2023), which highlights BM25’s robustness and effectiveness in  
1227 RAG tasks. Despite significant advancements in dense retrieval technologies, our results reaffirm  
1228 that the simpler, lexicon-based BM25 algorithm remains a strong baseline for improving LLM  
1229 performance in RAG tasks.

1230  
1231 

## E EXPLORING METRICS FOR KNOWLEDGE CONFLICTS DETECTION

1232  
1233 **Can Perplexity Reflects Knowledge Conflicts?** Recent studies have explored methods to detect  
1234 hallucinations in LLMs and RAG systems by leveraging various metrics (Chen et al., 2024; Sun  
1235 et al., 2024). Among these, we first adopt the simplest yet effective metric which only need single  
1236 generation, Perplexity (PPL) (Ren et al., 2022), to evaluate knowledge conflicts. As illustrated in  
1237 Figure5, Vanilla and DyPRAG exhibit higher PPL, while DyPRAG-Combine and RAG demonstrate  
1238 significantly lower PPL. However, these results are inconsistent with the findings in Table 1 and  
1239 Table 2. For instance, although DyPRAG-Combine achieves the best performance on IIRC, its  
1240 calculated PPL suggests a higher probability of knowledge conflicts, which is clearly incorrect. We  
1241 hypothesize that this discrepancy primarily stems from variations in model parameters introduced  
1242 by parameter injection in DyPRAG, which cannot be detected using the simple PPL method. Given

<sup>4</sup><https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

1242 Table 14: Textual similarity matrix (%) across both IID and OOD datasets. This matrix is computed  
 1243 based on the hidden states of retrieved documents and is symmetrical. It exhibits a significantly  
 1244 different trend when comparing documents from two distinct sources.

Dataset	2WQA	HQA	PQA	CWQ	IIRC	SQA	RAGTruth	OBQA	CNNDailyMail	MQA
2WQA	100	99.6	99.7	99.2	95.1	85.2	26.9	83.1	87.9	84.9
HQA	—	100	99.6	99.6	95.2	85.6	27.1	83.7	88.1	85.4
PQA	—	—	100	99.3	95.9	86.1	28.2	83.6	88.7	86.1
CWQ	—	—	—	100	94.5	84.3	26.3	84.1	86.6	84.8
IIRC	—	—	—	—	100	90.7	44.5	88.9	91.9	92.6
SQA	—	—	—	—	—	100	38.6	72.2	95.9	88.1
RAGTruth	—	—	—	—	—	—	100	51.3	40.4	54.3
OBQA	—	—	—	—	—	—	—	100	74.6	90.0
CNNDailyMail	—	—	—	—	—	—	—	—	100	88.8
MQA	—	—	—	—	—	—	—	—	—	100

1257 that different tokens contribute unequally to the overall semantics of a sentence, the PPL, which is  
 1258 calculated as the average of token-level uncertainty, fails to effectively capture the uncertainty of the  
 1259 entire sequence.

1261 **Effective Detection with Sentence-Level Metrics.** Given the limitations of PPL, we decided to  
 1262 explore alternative metrics that leverage multiple generations. Research has shown that generating  
 1263 multiple outputs for a single input is beneficial for estimating sequence-level uncertainty. To this end,  
 1264 we set the temperature to 1.0, top\_p to 0.95, and top\_k to 20, generating five responses to calculate  
 1265 Entropy (EN), Length Normalized Entropy (LEN) (Malinin & Gales, 2020), and Lexical Similarity  
 1266 (LS) (Lin et al., 2022) to evaluate the probability of knowledge conflicts<sup>5</sup>. As shown in Table 5, our  
 1267 approach demonstrates reduced knowledge conflicts in most scenarios, especially in our strongest  
 1268 DyPRAG-Combine.

1269 We observed that both EN and LEN increase when in-context injection is applied, suggesting that in  
 1270 RAG systems, the retrieved passages often conflict with the model’s internal knowledge. In contrast,  
 1271 utilizing DyPRAG to inject converted parametric knowledge significantly reduces the likelihood of  
 1272 knowledge conflicts, demonstrating the effectiveness of DyPRAG. However, the LS results indicate  
 1273 that adding context reduces conflicts, which contradicts the established definition of knowledge  
 1274 conflicts. We argue that EN and LEN are more suitable for effective knowledge conflicts detection in  
 1275 DyPRAG settings. Exploring more effective detection methods remains an important direction for  
 1276 future work.

## 1278 F DIVING INTO GENERALIZATION ABILITY OF DYPRAG

1280 To train our parameter translator, we utilized datasets, including 2WikiMultihopQA, HotpotQA,  
 1281 PopQA, and ComplexWebQuestions. To evaluate generalization, we conducted OOD experiments on  
 1282 datasets such as IIRC, StrategyQA, RAGTruth, OpenBookQA, and CNNDailyMail.

1283 For IID datasets, the documents were retrieved exclusively from Wikipedia. In contrast, the OOD  
 1284 datasets exhibit diverse sources: IIRC primarily draws from English Wikipedia, supplemented with  
 1285 samples from SQuAD 2.0 (Rajpurkar et al., 2016) and DROP (Dua et al., 2019). StrategyQA includes  
 1286 human-curated evidence paragraphs from Wikipedia. RAGTruth is based on the QA set of MS  
 1287 MARCO (Nguyen et al., 2016), which originates from Bing search results. OpenBookQA is a  
 1288 multiple-choice QA dataset derived from a subset of WorldTree (Jansen et al., 2018). CNNDailyMail  
 1289 is a summarization dataset comprising unique news articles authored by journalists at CNN and the  
 1290 Daily Mail.

1291 To quantify the differences across datasets, we computed the vector similarity of the mean hidden  
 1292 states (i.e., the last-layer outputs of the final token) across them. As expected, the IID datasets exhibit  
 1293 extremely high similarity (>99%) due to their shared reliance on Wikipedia. In contrast, the OOD  
 1294 datasets show significantly lower similarity with the IID datasets. Although StrategyQA and IIRC

1295 <sup>5</sup>We use the implementation in <https://github.com/alibaba/eigenscore>

1296 Table 15: Parameter similarity matrix (%) across both IID and OOD datasets generated by parameter  
 1297 translator. This matrix is computed based on the generated parameters and is symmetrical. It exhibits  
 1298 a significantly different trend when comparing parameters from two distinct sources.

Dataset	2WQA	HQA	PQA	CWQ	IIRC	SQA	RAGTruth	OBQA	CNNDailyMail	MQA
<b>2WQA</b>	89.15	88.27	85.29	88.19	88.49	87.66	47.15	83.16	89.25	86.39
<b>HQA</b>	—	83.45	84.84	85.50	84.66	84.95	49.55	82.99	86.27	84.29
<b>PQA</b>	—	—	88.55	86.88	88.57	86.31	47.45	82.40	87.98	85.78
<b>CWQ</b>	—	—	—	88.55	88.55	87.13	48.66	84.82	88.89	86.71
<b>IIRC</b>	—	—	—	—	90.43	87.58	47.93	82.88	88.48	86.34
<b>SQA</b>	—	—	—	—	—	89.01	48.87	84.75	89.63	86.75
<b>RAGTruth</b>	—	—	—	—	—	—	77.52	56.73	48.89	53.03
<b>OBQA</b>	—	—	—	—	—	—	—	92.95	86.02	88.56
<b>CNNDailyMail</b>	—	—	—	—	—	—	—	—	91.97	88.47
<b>MQA</b>	—	—	—	—	—	—	—	—	—	89.42

1312 primarily depend on Wikipedia, they include additional samples from other sources or incorporate  
 1313 human-curated content, which reduces their similarity to the IID datasets. Notably, RAGTruth  
 1314 demonstrates particularly low similarity, as its samples are carefully selected from MS MARCO to  
 1315 focus exclusively on content related to daily life. This underscores the substantial differences between  
 1316 the training corpora and our OOD evaluation datasets.

1317 These findings further suggest that DyPRAG exhibits strong generalization capabilities, effectively  
 1318 adapting to the diverse characteristics of OOD datasets, as shown in Table 2.

## 1320 G DOES PARAMETER TRANSLATOR REALLY LEARN TO GENERALIZE?

1323 After obtaining the parameter translator, a natural question arises: does the parameter translator  $\mathcal{F}'_\phi$   
 1324 truly learn to generalize, or does it simply generate nearly identical LoRA matrices every time?

1325 To investigate this, we collect 20 generated parameters across all datasets and compute the inter-  
 1326 average and intra-average parameter similarity. Since the parameter space itself is non-semantic,  
 1327 we measure similarity using the Frobenius norm:  $1 - \frac{\|A - B\|_F}{\max(\|A\|_F, \|B\|_F)}$ . As shown in Table 15,  
 1328 the similarity of the generated LoRA parameters strongly correlates with the textual similarity  
 1329 of the inputs. In particular, the model produces significantly different outputs when exposed to  
 1330 distinct contexts, even from the same dataset. Although hypernetwork still lacks well-established  
 1331 interpretability methods, this simple comparison provides evidence that the hypernetwork is indeed  
 1332 mapping from different textual embeddings to diverse parameter space. We hope that future research  
 1333 will develop more comprehensive approaches to explain hypernetwork behavior.

## 1335 H WHY VANILLA OUTPERFORMS RAG OCCASIONALLY?

1337 In this section, we provide a detailed analysis of why the vanilla model occasionally outperforms  
 1338 RAG. As shown in Table 1, the vanilla model surpasses RAG most significantly in 2WQA, as the  
 1339 results vary across different models. For instance, the vanilla model outperforms RAG by 2.62% and  
 1340 0.99% in Qwen2.5-1.5B and LLaMA3-8B on average in F1, respectively. After analyzing the cases,  
 1341 we identify two key issues that most affect RAG’s performance: **1) Poor Retriever.** Following Su  
 1342 et al. (2025), we use BM25 as the retriever. However, in many cases, the retrieved documents  
 1343 contain only similar words rather than relevant content. This results in the provided content being  
 1344 unhelpful or even detrimental to LLMs. **2) Already Seen Data.** During the pre-training stages of  
 1345 the selected LLMs (Yang et al., 2024; Meta, 2024a;b), the external source we use (i.e., Wikipedia)  
 1346 has already been seen. This allows LLMs to answer certain questions independently, especially in  
 1347 simpler tasks like 2WQA. Moreover, the inclusion of incorrect or irrelevant context further degrades  
 1348 the performance, as observed in Table 1.

1349 A more rigorous evaluation setting should include ground-truth passages and ensure no or less data  
 leakage. Under this setting, as shown in Table 2, the performance of the vanilla model is significantly

1350 lower than that of RAG, which aligns with our hypothesis. For instance, the vanilla model achieves  
 1351 only 8.78% and 1.00% accuracy on Qwen2.5-1.5B for IIRC and SQA, respectively. In contrast,  
 1352 DyPRAG demonstrates a notable improvement in test-time knowledge, achieving 10.23% and 15.67%  
 1353 accuracy on Qwen2.5-1.5B for IIRC and SQA, respectively. These results underscore the critical  
 1354 role of RAG while showcasing the ability of our proposed DyPRAG to seamlessly enhance OOD  
 1355 knowledge effectively. Furthermore, DyPRAG-Combine establishes a superior RAG paradigm by  
 1356 delivering even better performance under these more challenging conditions. In summary, we believe  
 1357 that this more rigorous experimental setting better validates our proposed method.  
 1358

## 1359 I FURTHER ANALYSIS OF CONTEXTUAL AND PARAMETRIC KNOWLEDGE 1360 CONFLICTS

1361 **Parameter Injection Makes LLMs Trust Themselves.** As shown in Table 16, while vanilla LLMs  
 1362 contain accurate parametric knowledge regarding which director was born later, the introduction  
 1363 of retrieved documents about each director causes contextual knowledge to mislead  $\mathcal{M}$ , resulting  
 1364 in the incorrect answer "William Lustig" while DyPRAG stays the same. This demonstrates that  
 1365 DyPRAG can effectively reduce the knowledge conflicts problem. In this case, standard RAG often  
 1366 introduces redundant or incorrect information from the context, a phenomenon commonly referred to  
 1367 as RAG hallucination (Sun et al., 2024). In contrast, our proposed DyPRAG effectively incorporates  
 1368 accurate information into parametric knowledge. This allows DyPRAG-Combine to align parametric  
 1369 knowledge with contextual knowledge, thereby reducing the likelihood of conflicts and enabling  
 1370 LLMs to rely more consistently on its own knowledge.  
 1371

1372 **Dynamic Parametric Knowledge Enhances LLMs at Test-time.** Our DyPRAG serves as an  
 1373 effective plug-and-play technique for enhancing parametric knowledge during test-time. As demon-  
 1374 strated in Table 17, DyPRAG successfully manipulates the original parametric knowledge of LLMs in  
 1375 14.67% of cases. Therefore, it can directly enhance the model's knowledge during inference without  
 1376 the need for further fine-tuning.  
 1377

1378 **Proportion of Different Combinations.** Furthermore, as shown in Table 18, when both Vanilla  
 1379 LLMs and RAG give incorrect answers, DyPRAG provides the correct answer 26.33% of the time.  
 1380 This indicates that DyPRAG can effectively inject missing parametric knowledge and outperforms  
 1381 in-context injection methods. Additionally, in cases where the vanilla LLM provides the correct  
 1382 answer (i.e., the model possesses accurate internal knowledge), RAG achieves a correct answer rate  
 1383 of 5.33%, while DyPRAG performs better with a rate of 6.33%, showing that parameter injection  
 1384 leads to lower conflicts. Similar trend of DyPRAG-Combine is presented in Table 19.  
 1385

1386 These results demonstrate that our proposed DyPRAG injects parametric knowledge successfully  
 1387 and mitigates conflicts between internal parametric knowledge and external contextual knowledge  
 1388 through the injection of knowledgeable LoRA adapters.  
 1389  
 1390  
 1391  
 1392  
 1393  
 1394  
 1395  
 1396  
 1397  
 1398  
 1399  
 1400  
 1401  
 1402  
 1403

Table 16: Case study about contextual and parametric knowledge conflicts in 2WQA (Bridge sub-task) where only standard RAG answers wrongly (6.67%). The backbone model is the LLaMA3.2-1B. █: deficiency in parametric knowledge, █: knowledge conflicts, █: successful knowledge manipulation.

<b>Question:</b> Which film has the director born later, <span style="background-color: #ffff00;">Diary Of A Maniac</span> or <span style="background-color: #90ee90;">Return Of The Hero</span> ?		
<b>Ground truth:</b> <span style="background-color: #90ee90;">Return Of The Hero</span>		
<b>Retrieved top-1 document:</b> Maniac (1980 film) Maniac is a 1980 American psychological slasher film directed by <span style="background-color: #ffff00;">William Lustig</span> and written by C. A. Rosenberg...		
<b>Method</b>	<b>Answer</b>	<b>Status</b>
Vanilla	<span style="background-color: #90ee90;">Return Of The Hero</span>	<span style="color: green;">✓</span>
RAG	<span style="background-color: #ffff00;">William Lustig</span>	<span style="color: red;">✗</span>
DyPRAG (ours)	<span style="background-color: #90ee90;">Return Of The Hero</span>	<span style="color: green;">✓</span>
DyPRAG-Combine (ours)	<span style="background-color: #90ee90;">Return Of The Hero</span>	<span style="color: green;">✓</span>

Table 18: Right/Wrong answer combinations of Vanilla, RAG, DyPRAG and corresponding proportional distribution in 2WQA (Bridge Sub-task). The backbone model is the LLaMA3.2-1B. ✓ indicates a correct answer, while ✗ indicates an incorrect answer. The "Ratio (%)" column on the right represents the percentage of each combination across the dataset (300 examples).

Vanilla	RAG	DyPRAG	Ratio(%)
<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	4.67
<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	34.67
<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<b>6.33</b>
<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	5.33
<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	8.33
<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<b>26.33</b>
<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	7.67
<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	6.33

Table 17: Case study about contextual and parametric knowledge conflicts in 2WQA (Bridge sub-task) where only DyPRAG and DyPRAG-Combine answer wrongly (14.67%). The backbone model is the LLaMA3.2-1B. █: deficiency in parametric knowledge, █: knowledge conflicts, █: successful knowledge manipulation

<b>Question:</b> Which film has the director born later, <span style="background-color: #ffff00;">Miss Sloane</span> or <span style="background-color: #90ee90;">Time Changer</span> ?		
<b>Ground truth:</b> <span style="background-color: #90ee90;">Time Changer</span>		
<b>Retrieved top-1 document:</b> production budget of \$13 million. " <span style="background-color: #ffff00;">Miss Sloane</span> " is ranked number 75 by per-theater average on Box Office...		
<b>Method</b>	<b>Answer</b>	<b>Status</b>
Vanilla	<span style="background-color: #f08080;">John Frankenheimer</span>	<span style="color: red;">✗</span>
RAG	<span style="background-color: #ffff00;">Miss Sloane</span>	<span style="color: red;">✗</span>
DyPRAG (ours)	<span style="background-color: #90ee90;">Time Changer</span>	<span style="color: green;">✓</span>
DyPRAG-Combine (ours)	<span style="background-color: #90ee90;">Time Changer</span>	<span style="color: green;">✓</span>

Table 19: Right/Wrong answer combinations of Vanilla, RAG, DyPRAG-Combine and corresponding proportional distribution in 2WQA (Bridge Sub-task). The backbone model is the LLaMA3.2-1B. ✓ indicates a correct answer, while ✗ indicates an incorrect answer. The "Ratio (%)" column on the right represents the percentage of each combination across the dataset (300 examples).

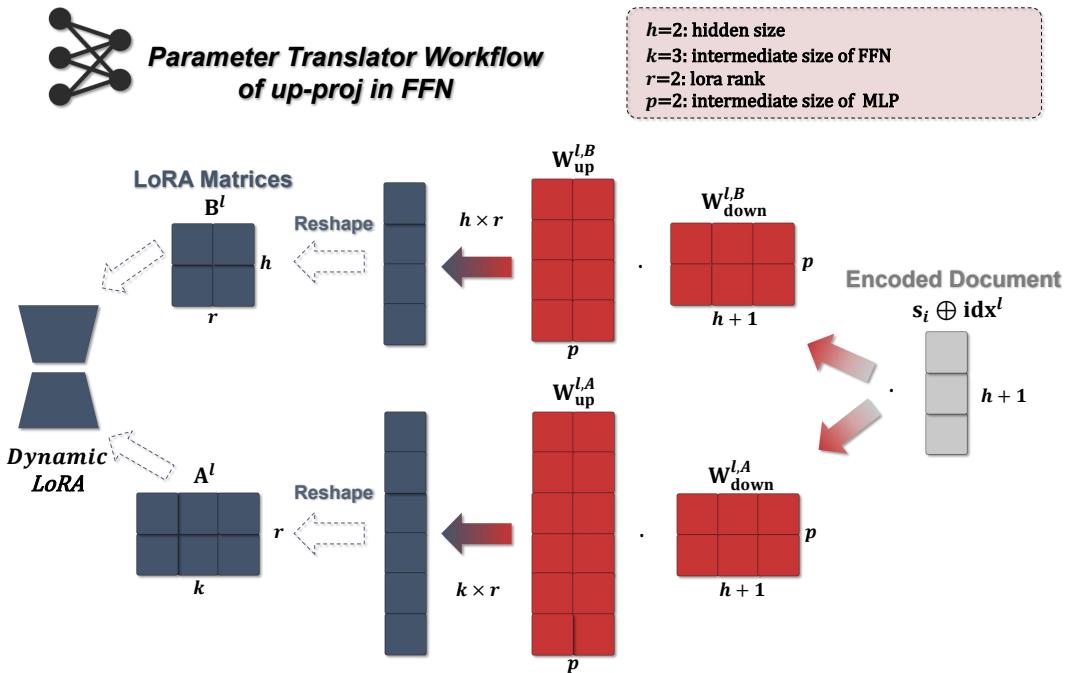
Vanilla	RAG	DyPRAG-Combine	Ratio(%)
<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	5.33
<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	35.00
<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<b>6.33</b>
<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	4.67
<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<span style="color: green;">✓</span>	8.00
<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<b>26.00</b>
<span style="color: red;">✗</span>	<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	8.00
<span style="color: green;">✓</span>	<span style="color: red;">✗</span>	<span style="color: red;">✗</span>	6.67

1458  
 1459  
 1460  
 1461  
 1462  
 1463  
 1464  
 1465  
 1466  
 1467  
 1468  
 1469  
 1470  
 1471  
 1472  
 1473  
 1474  
 1475

1476 Table 20: The experimental results of DyPRAG are compared with parametric RAG, standard RAG  
 1477 and two training-based methods. All metrics are reported as F1 scores (%). The best performance is  
 1478 bolded, while the second-best is underlined. The **Avg** is the average performance over all sub-tasks.

Base LLM	Method	2WQA					HQA			PQA	CWQ	Avg
		Compare	Bridge	Inference	Compose	Total	Bridge	Compare	Total			
LLaMA3.2-1B	Vanilla	42.89	24.17	16.91	7.87	22.52	13.25	40.26	18.79	2.26	34.94	22.39
	SFT	25.36	10.87	6.05	3.35	10.60	1.86	4.51	2.51	1.33	12.77	7.92
	Context-DPO	37.28	39.39	16.29	4.86	22.89	17.03	32.86	20.17	12.79	13.00	21.66
	RAG	41.23	26.78	22.51	<u>10.21</u>	23.12	21.38	42.46	<u>27.14</u>	17.65	37.39	26.99
	PRAG	50.20	24.34	19.11	8.24	27.73	13.65	40.90	21.50	23.58	35.86	26.51
	PRAG-Combine	40.50	31.30	<b>22.85</b>	9.77	<u>30.30</u>	<b>22.56</b>	41.55	<b>28.31</b>	<u>32.59</u>	<b>39.63</b>	<u>29.94</u>
	DyPRAG (ours)	<u>51.25</u>	<b>48.15</b>	17.35	7.54	25.31	14.05	<b>43.90</b>	19.97	11.33	36.86	27.57
Qwen2.5-1.5B	DyPRAG-Combine (ours)	<b>52.13</b>	<u>46.19</u>	<u>22.54</u>	<b>12.60</b>	<u>29.18</u>	<u>22.05</u>	43.78	26.58	<u>29.93</u>	<u>38.96</u>	<b>31.80</b>
	Vanilla	<b>45.74</b>	39.06	17.04	7.27	26.87	12.18	39.46	17.76	2.87	26.47	25.79
	SFT	37.98	43.44	9.06	3.83	18.75	5.82	26.21	8.85	6.95	13.96	17.49
	Context-DPO	35.01	40.59	<u>17.88</u>	6.51	21.78	19.12	31.41	22.51	14.18	19.20	22.82
	RAG	38.75	38.84	11.87	5.68	24.31	16.19	37.13	20.73	9.97	28.23	23.17
	PRAG	<u>44.96</u>	43.96	<b>19.29</b>	<b>11.14</b>	<b>27.55</b>	13.27	40.42	18.42	21.55	30.82	<u>27.14</u>
	PRAG-Combine	40.50	44.00	16.30	8.17	<u>27.49</u>	<u>18.86</u>	36.49	<u>23.10</u>	<b>23.43</b>	<u>32.13</u>	27.05
LLaMA3-8B	DyPRAG (ours)	43.03	<b>47.20</b>	17.04	8.55	26.46	13.72	<b>41.39</b>	19.67	6.64	31.94	25.56
	DyPRAG-Combine (ours)	35.83	44.89	14.81	<u>8.64</u>	25.18	<b>21.56</b>	41.25	<b>27.57</b>	22.69	<b>33.57</b>	<b>27.60</b>
	Vanilla	54.90	55.20	24.59	14.43	33.02	19.00	45.63	21.29	7.96	<b>42.44</b>	31.85
	SFT	9.66	26.26	16.79	1.12	11.63	2.04	2.48	2.05	0.00	5.92	7.80
	Context-DPO	46.90	25.57	20.83	6.81	21.98	13.86	32.97	18.17	18.68	13.81	21.96
	RAG	58.43	47.77	19.20	11.07	34.55	19.68	42.10	24.23	16.13	35.45	30.86
	PRAG	57.78	<u>58.93</u>	27.61	19.17	39.19	<u>33.68</u>	<u>65.88</u>	38.08	26.13	<b>43.54</b>	41.00
	PRAG-Combine	<u>60.13</u>	56.69	<u>32.71</u>	<u>20.91</u>	<u>40.55</u>	<b>39.41</b>	<b>68.22</b>	<b>44.84</b>	<u>26.23</u>	36.41	<u>42.61</u>
	DyPRAG (ours)	57.39	56.43	25.33	18.88	37.80	24.85	58.59	28.56	13.60	41.87	36.23
	DyPRAG-Combine (ours)	<b>66.00</b>	<b>59.46</b>	<u>35.78</u>	<b>26.90</b>	<b>50.24</b>	33.37	57.93	<u>38.35</u>	<b>32.86</b>	39.07	<b>43.69</b>

1495  
 1496  
 1497  
 1498  
 1499  
 1500  
 1501  
 1502  
 1503  
 1504  
 1505  
 1506  
 1507  
 1508  
 1509  
 1510  
 1511

1512 **J VISUALIZATION OF PARAMETER TRANSLATOR WORKFLOW.**  
15131514 To clearly illustrate the workflow of the parameter translator  $\mathcal{F}'_\phi$ , we use the up-proj module in the  
1515 FFN as an example, as shown in Figure 10. This visualization demonstrates the transformation of  
1516 document embeddings into dynamic LoRAs, consistent with Eq. 4.  
15171539 Figure 10: Visualization of the parameter translator workflow of up-proj in FFN. The overall process  
1540 remains consistent with Eq. 4.  
1541

1566 **K PROMPT FOR MAIN EXPERIMENTS EVALUATION**  
 1567

1568 In the main experiments, we used the following prompt to assess the performance of DyPRAG and  
 1569 other baseline models in Figure 11 and 12:  
 1570

1571 **Prompt Format of No-CoT**  
 1572

1573 You should answer the question by referring to the knowledge provided below and integrating your own  
 1574 knowledge.  
 1575

1576 Passage 1: {passages[0]}  
 1577 Passage 2: {passages[1]}  
 1578 Passage 3: {passages[2]}  
 1579

1580 Question: {question}  
 1581 The answer is {answer}

1582 **Figure 11: Prompt format of No-CoT in our experiments.**  
 1583

1584 **Prompt Format of CoT**  
 1585

1586 You should reference the knowledge provided below and combine it with your own knowledge to answer  
 1587 the question. Please follow the format of the example I provided above. Here are some examples about  
 1588 how to answer the questions.  
 1589

1590 Question:  $\text{fewshot}_q[0]$   
 1591 Answer:  $\text{fewshot}_a[0]$   
 1592 Question:  $\text{fewshot}_q[1]$   
 1593 Answer:  $\text{fewshot}_a[1]$   
 1594 Question:  $\text{fewshot}_q[2]$   
 1595 Answer:  $\text{fewshot}_a[2]$

1596 ...

1597 Here are some reference.  
 1598 Passage 1: {passages[0]}  
 1599 Passage 2: {passages[1]}  
 1600 Passage 3: {passages[2]}

1601 Let's think step by step. Answer the questions in the same format as above.  
 1602 Question: {question}  
 1603 Answer: {answer}

1604 **Figure 12: Prompt format of CoT in our experiments.**  
 1605

1606 In summarization experiment in CNNDailyMail, we used the following prompt to assess the performance  
 1607 of DyPRAG and other baseline models in Figure 13:  
 1608

1609 **Prompt Format of summarization**  
 1610

1611 Please summarize the main ideas from the content of Passage 1 in a clear and concise manner.  
 1612

1613 Passage 1: {passages[0]}  
 1614

1615 **Figure 13: Prompt format of summarization in our experiments.**  
 1616

1620 **L PROMPT FOR KNOWLEDGE INTERNALIZATION EVALUATION**  
16211622 In the knowledge internalization experiments, we used the following prompt to assess the internaliza-  
1623 tion ability of RAG generation from DyPRAG-Combine and RAG method evaluated by GPT-4o in  
1624 Figure 14:1625 Prompt Format of Evaluate RAGTruth  
16261627 Compare DyPRAG and RAG answers to assess which better internalizes knowledge—integrating its  
1628 own knowledge with the given context for a natural, informed response.

## 1629 Evaluation Criteria:

1630 1. Internalization: Does the answer go beyond repetition to integrate knowledge seamlessly?  
1631 2. Fluency: Is the response well-structured and readable?  
1632 3. Relevance: Does it stay on topic while demonstrating depth?

1633 Mark the Winner: Identify the superior response. If both are equally strong, mark it as a tie.

1634 Question: {question}

1635 Context: {passages}

1636 DyPRAG Answer: {dyprag\_answer}

1637 RAG Answer: {rag\_answer}

1638 Respond in the following format:

1639 {{

1640 "win model": "DyPRAG or RAG or Tie",

1641 "reason": "Provide a concise explanation of why the selected answer demonstrates better knowledge  
1642 integration, referencing the question, context, and specific details from both answers. If one answer has  
1643 clear advantages in integration, explain them; if there are errors or weaknesses, specify them."  
1644 }}1645  
1646  
1647 Figure 14: Prompt format of evaluate RAGTruth using GPT-4o. We compare answer between  
1648 standard RAG and DyPRAG-Combine.  
16491650 **M FUTURE DIRECTIONS**  
16511652 In this study, our proposed Dynamic Parametric RAG (DyPRAG) demonstrates superior performance  
1653 in both IID and OOD settings across various scales of LLMs. Developing and deploying such RAG  
1654 system in real-world applications is a promising and worthwhile avenue for future work. Moreover,  
1655 we believe the most promising direction for DyPRAG lies in the integration of memory (Wang et al.,  
1656 2024d), which is commonly implemented using external textual databases. This raises a fundamental  
1657 question: *Can a parameter translator convert any textual knowledge into parametric knowledge?* If  
1658 the answer is yes, it would enable the replacement of large, text-based memory banks with a simple,  
1659 plug-and-play memory translator. This approach opens up an exciting avenue for enhancing the  
1660 fine-grained knowledge manipulation capabilities of  $\mathcal{F}'_\phi$ , which we aim to explore in future work.1661  
1662 **N REPRODUCIBILITY**  
16631664 In this work, we use open-source LLMs and publicly available datasets to conduct our experiments.  
1665 To ensure reproducibility, we provide the implementation details in Section 4.1 and Appendix B.1.  
1666 Details of all prompts referenced in this paper are included in Appendix K and L. The full code  
1667 and a detailed reproduction procedure of DyPRAG, which can be accessed via the following link:  
1668 [https://anonymous.4open.science/r/DyPRAG\\_ICLR](https://anonymous.4open.science/r/DyPRAG_ICLR). We also provide a well-trained parameter  
1669 translator used in our experiments, available through the anonymous link.1670  
1671 **O THE USAGE OF LLMs**  
16721673 We used Large Language Models (LLMs) to perform minor language polishing and grammar  
refinement on select sections of the paper. The LLMs were not involved in generating core content,

1674 conducting research, or formulating ideas. All substantive contributions, including analysis, results,  
1675 and conclusions, were independently produced by the authors.  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727