DyPRAG: Dynamic Parametric Retrieval-Augmented Generation for On-the-Fly Knowledge Adaptation

Anonymous ACL submission

Abstract

Once Large Language Models (LLMs) complete their training, the intrinsic parametric knowledge encoded within the model becomes fixed. Retrieval-Augmented Generation (RAG) alleviates this by supplying external documents as context, but a fundamental tension remains: the model's parameters are unchanged, often leading to conflicts where the model's outdated internal parameters struggle to synergize with the fresh retrieved information. In this paper, we propose **Dynamic Parametric RAG (DyPRAG)**, a novel framework that leverages a lightweight hypernetwork, termed parameter translator, to efficiently convert symbolic documents into parametric knowledge at inference-time. Specifically, the parameter translator maps documents directly into Low-Rank Adaptation (LoRA) weights for the Feed-Forward Networks (FFNs) of the LLM, enabling on-the-fly knowledge adaptation. Extensive experiments across diverse datasets demonstrate the superior effectiveness and generalization capability of DyPRAG. Crucially, our analysis confirms that DyPRAG harmonizes the model's internal and external knowledge sources, leading to a measurable reduction in knowledge conflicts and a more effective synthesis of information. Our code is available at https://anonymous.4open.science/r/DyPRAG-715.

1 Introduction

Large Language Models (LLMs) have emerged as the cornerstone of modern Natural Language Processing (NLP), demonstrating remarkable capabilities in text comprehension and generation (Brown et al., 2020; Touvron et al., 2023). This success is largely attributed to pretraining on massive-scale corpora, which allows models to encode rich linguistic patterns and world knowledge, thereby facilitating robust generalization across diverse tasks (Achiam et al., 2023; Dubey et al., 2024).

However, the intrinsic knowledge of LLMs becomes static after pre-training, restricting their knowledge to the training corpus (Kim et al., 2024). Consequently, despite their impressive general capabilities, performance on knowledge-intensive applications is constrained by a lack of access to up-to-date or domain-specific information (Joshi et al., 2017; Kwiatkowski et al., 2019; Frisoni et al., 2024) (in fig. 1 (a)). For instance, querying an LLM released in 2024 with "*Who were the recipients of the 2025 Nobel Prize in Chemistry?*" results in failure, as the model suffers from knowledge deficiency regarding unseen events.

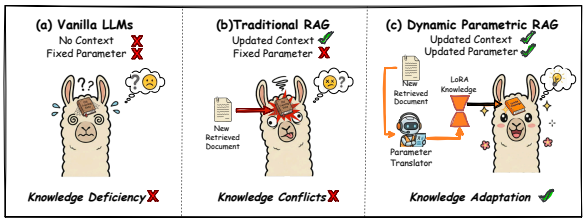

Figure 1: Comparison of vanilla LLMs, RAG, and DyPRAG. (a) Vanilla LLMs rely on static training data, leading to knowledge deficiency. (b) RAG supplements models with retrieved text but risks knowledge conflicts. (c) DyPRAG employs a parameter translator to convert symbolic documents into LoRA weights, achieving complementary contextual and parametric knowledge.

To address this, Retrieval-Augmented Generation (RAG) (Guu et al., 2020a) has emerged as a prominent approach. It retrieves new knowledge from external sources (e.g., Wikipedia) and provides it as context to the LLM (in fig. 1 (b)). However, while RAG mitigates knowledge deficiency, the model's parameters remain frozen. This creates a low overlap between the static internal knowledge and the dynamic retrieved information, as illustrated in fig. 2 (a). For example, a frozen LLM's outdated knowledge about the 2025 Nobel Prize has minimal commonality with up-to-date retrieved documents. This discrepancy renders the two information sources susceptible to conflict, causing the model to hallucinate or misapply retrieved context

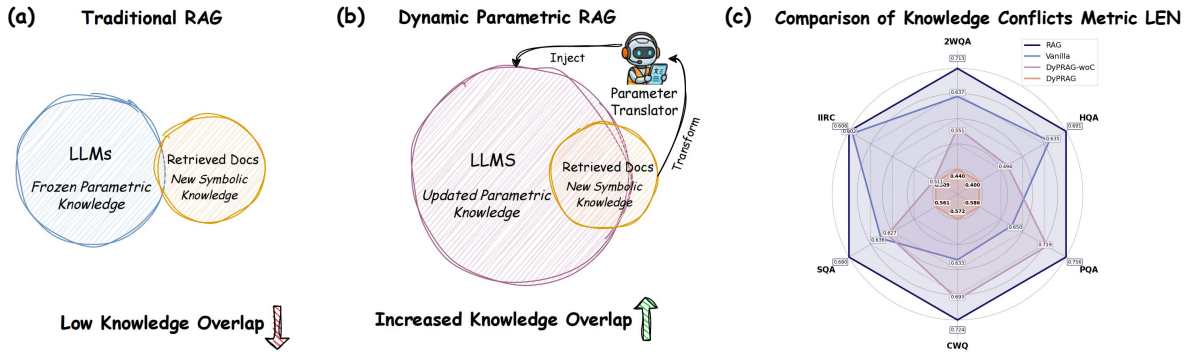

Figure 2: Overview of the knowledge overlap between parametric and symbolic knowledge. (a) In RAG, newly retrieved documents exhibit low overlap with frozen parametric knowledge. (b) In DyPRAG, the parameter translator converts documents into updated parameters, increasing the overlap. (c) Knowledge conflict measured by Length Normalized Entropy (LEN) (lower is better) decreases with updated parameters, indicating reduced conflicts.

and thereby increasing the likelihood of erroneous outputs, even when relevant documents are available (Zhang et al., 2024; Sun et al., 2024).

Parametric RAG (PRAG) (Su et al., 2025) updates the internal knowledge by encoding each document into a dedicated Low-Rank Adaptation (LoRA) (Hu et al., 2022) module via offline training. However, PRAG suffers from two major drawbacks. First, the per-document fine-tuning process incurs substantial computational and storage costs. Second, it is restricted to trained documents and fails to generalize to unseen ones. These limitations substantially undermine its scalability and practicality in real-world applications.

To address these challenges, we propose ***Dynamic Parametric RAG (DyPRAG)***, a novel and lightweight framework for on-the-fly knowledge adaptation (in fig. 1 (c)). DyPRAG is motivated by the hypothesis that a document and its corresponding parametric representation reside in a shared latent knowledge space, suggesting the existence of an implicit mapping function $\mathcal{F}$ that can transform documents directly into parametric form. Building on this assumption, DyPRAG introduces a parameter translator $\mathcal{F}'_\phi$, a compact hypernetwork trained offline to approximate this generalized mapping. At inference time, the translator dynamically generates document-specific parameters, enabling lightweight updates to the LLM's parametric knowledge with minimal overhead. Importantly, these inference-time updates increase the overlap between the internal parameters and the retrieved documents, allowing the LLM to better internalize the provided context, as illustrated in fig. 2 (b).

Extensive experiments demonstrate the effectiveness of DyPRAG, showing superior performance and strong generalization to unseen documents across both in-domain (ID) and out-of-domain (OOD) tasks. We further conduct in-depth analyses to verify that DyPRAG achieves genuine on-the-fly knowledge adaptation. As illustrated in fig. 2(c), conventional RAG increases the likelihood of knowledge conflicts by appending retrieved documents directly to the input context (Niu et al., 2023; Zhang et al., 2024; Tao et al., 2024). In contrast, DyPRAG addresses this issue by updating the model's parametric space. After loading the parameters generated by the parameter translator, the resulting parametric knowledge exhibits substantially higher overlap with the retrieved symbolic knowledge. This improved alignment mitigates conflicts and renders the two information sources highly complementary. Finally, comprehensive ablation studies and additional analyses further corroborate the effectiveness of our approach. We summarize our contributions as follows:

- We propose **Dynamic Parametric RAG (DyPRAG)**, the first lightweight framework to enable on-the-fly parametric knowledge adaptation for RAG system by converting symbolic documents into parametric representations.
- We empirically demonstrate that DyPRAG achieves superior performance and robust generalization across both in-domain and out-of-domain tasks, confirming its effectiveness and practical applicability in real-world scenarios.
- We provide an in-depth analysis revealing that DyPRAG significantly increases knowledge overlap between the parameters and retrieved content, leading to a reduction in knowledge conflicts and better synthesis of information.

2 Related Work

2.1 Retrieval Augmented Generation

Large language models (LLMs) excel broadly but lack sufficient knowledge for intensive tasks, highlighting the need for external knowledge integration (in fig. 1 (a)). A prominent solution is retrieval-augmented generation (RAG), which enhances LLMs by incorporating relevant external knowledge sources (Guu et al., 2020b; Borgeaud et al., 2022; Wang et al., 2024a,b). Parametric RAG (PRAG) (Su et al., 2025) further advances this by fine-tuning the model on augmented documents, encoding useful information directly into the model parameters. However, these approaches suffer from static parameters at inference-time or incur prohibitive costs when training on each document separately (in fig. 1 (b)). Our proposed DyPRAG enables on-the-fly knowledge adaptation by modeling the underlying mapping function from documents to parameters, thereby increasing the knowledge overlap between two distinct knowledge types and making them complementary (in fig. 1 (c)).

2.2 Context Parametrization

Context parametrization is widely used to convert contextual knowledge into parametric forms that LLMs can more easily digest. Recent studies have proposed condensing long contexts into soft prompts, enabling LLMs to leverage information more effectively (Mu et al., 2023; Ge et al., 2023). Meanwhile, other works focus on transforming context chunks into LoRA modules to enhance the model's ability to understand extended contexts (Mao et al., 2024; Wang et al., 2024c; Charakorn et al., 2025). For example, xRAG (Cheng et al., 2024) integrates context compression by mapping documents into compact token representations. Unlike previous studies, we present the first investigation of transforming symbolic documents into model parameters within RAG systems at inference-time. Our DyPRAG demonstrates that it effectively unifies contextual and parametric knowledge, significantly increases knowledge overlap, reduces conflicts, and improves overall performance in RAG systems.

3 Preliminary

Standard RAG. We first introduces the problem formulation of the standard RAG task (Guu et al., 2020b). Let $\mathcal{M}$ denote a large language model (LLM) with base parameters $\mathbf{\Theta}$. Given a user query q, the task of LLM is to generate an accurate response augmented by an external corpus $\mathcal{C}$, expressed as $\mathcal{C} = \{d_1, d_2, \ldots, d_N\}$. Each element d_i, referred to as a document, represents a text chunk retrieved from external sources (Izacard and Grave, 2021). To achieve this, a retrieval module $\mathcal{R}$ is employed to compute relevance scores between q and the documents in $\mathcal{C}$. Traditional RAG approaches select the top-c documents with the highest similarity scores and concatenates them with the query to form the extended input context. Based on this augmented input, $\mathcal{M}$ generates the response by leveraging both the query and the retrieved documents. This procedure leverages the ability of in-context learning (Brown et al., 2020) while the whole parameters remain unchanged.

Parametric RAG. Parametric RAG (PRAG) (Su et al., 2025) encodes each document $d_i \in \mathcal{C}$ directly into the parameters of model $\mathcal{M}$. This is achieved by training a parametric representation $\mathbf{P}_i$ for each document offline using LoRA (Hu et al., 2022). To ensure $\mathbf{P}_i$ captures fine-grained information, PRAG first performs data augmentation (Allen-Zhu and Li, 2023). For each document d_i, this process generates: (1) n paraphrased versions $\{d_i^1, \ldots, d_i^n\}$, and (2) m question-answer (QA) pairs $\{(q_i^1, a_i^1), \ldots, (q_i^m, a_i^m)\}$. These augmentations preserve the original factual content while introducing varied expressions. The augmented data is then structured into a training set $\mathcal{D}_i$ for each document:

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

where each training sample x is a concatenation of a triplet from this set: $x = [d_i^k \oplus q_i^j \oplus a_i^j]$. The overall goal is to optimize:

$$\min_{\mathbf{P}_i} \sum_{x \in D_i} \sum_{t=1}^{T} -\log\ P_{\mathbf{\Theta}+\mathbf{P}_i}\left(x_t \,\middle|\, x_{<t}\right), \quad (2)$$

where $\mathbf{P}_i$ represents the trainable low-rank matrices, which are applied exclusively to the feed-forward network (FFN) layers of the model. During inference, the trained parametric representation $\mathbf{P}_i$ is directly merged with the base parameters. However, this approach incurs substantial training and storage overhead, as a unique set of parameters $\mathbf{P}_i$ must be trained and stored for each document.

4 Methodology of DyPRAG

While PRAG considers updating parametric knowledge in RAG rather than relying solely on static

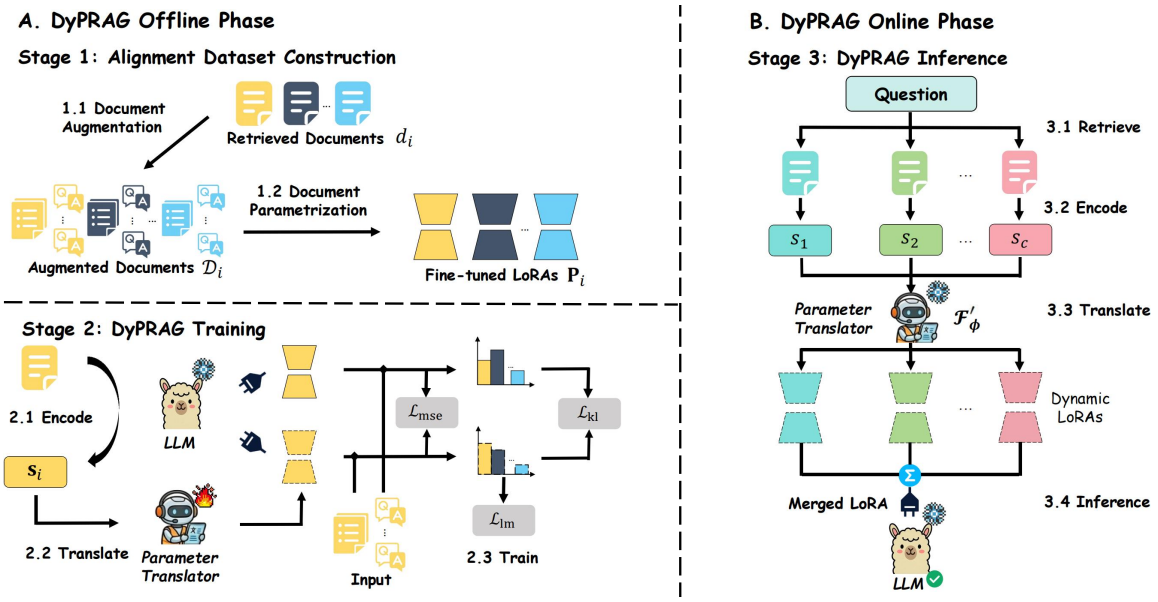

Figure 3: An illustration of the DyPRAG framework. The offline phase consists of two stages: **Stage 1** conducts parameterization process to collect document-parameter pairs. In **Stage 2**, a parameter translator $\mathcal{F}'_\phi$ is trained to learn a generalizable mapping from documents to corresponding parametric representations. During the online phase, **Stage 3** leverages the well-aligned $\mathcal{F}'_\phi$ to dynamically generate LoRA modules for any document at inference-time.

model parameters, it fails to support efficient on-the-fly deployment. Its heavyweight design incurs substantial time and storage overhead, which becomes prohibitive in scenarios involving frequently updated documents, thereby severely limiting its practical utility. In this section, we introduce **Dynamic Parametric RAG (DyPRAG)**. DyPRAG is based on the hypothesis that a document d_i and its well-trained parametric representation $\mathbf{P}_i$ lie in a shared underlying knowledge space. Consequently, there exists a latent mapping function $\mathcal{F}$ that can directly transform an arbitrary document into its corresponding parametric representation.

4.1 Training Parameter Translator for General Mapping Function

Alignment Dataset Construction. To derive the general mapping function $\mathcal{F}$, we first construct a collection of document–parameter (Doc–Param) pairs, in which the mapping function is implicitly embedded, following the procedure described in section 3. Specifically, for each document d_i, we obtain its corresponding parametric representation $\mathbf{P}_i$, thereby forming the alignment dataset $\mathcal{K} = \{(d_1, \mathbf{P}_1), (d_2, \mathbf{P}_2), \ldots, (d_N, \mathbf{P}_N)\}$.

Design of Parameter Translator. To model the implicit transformation, we introduce a lightweight hypernetwork, termed the **Parameter Translator** $\mathcal{F}'_\phi$, which maps the representation $\mathbf{s}_i$ to a parametric form $\mathbf{P}'_i$, i.e., $\mathbf{P}'_i = \mathcal{F}'_\phi(\mathbf{s}_i)$. For a given document d_i, we extract the last hidden state $\mathbf{s}_i \in \mathbb{R}^h$ at the final token position prior to projection into the vocabulary space, where h denotes the hidden dimension. The parameter translator is implemented as a stack of linear layers parameterized by a shared base parameter set ϕ. After constructing the alignment set $\mathcal{K}$, we employ the original large language model $\mathcal{M}$ to encode textual documents into dense representations. As a concrete example, we consider the up-proj module in the feed-forward network (FFN). The standard LoRA procedure is formulated as follows:

$$\mathbf{W}' = \mathbf{W} + \mathbf{\Delta W} = \mathbf{W} + \mathbf{BA} \tag{3}$$

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 r is the controllable LoRA rank. At training phase, $\mathcal{F}'_\phi$ performs separately on $\mathbf{B}$ and $\mathbf{A}$. Formally:

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

where $\mathbf{W}^{l,B}_{\text{down}} \in \mathbb{R}^{p\times(h+1)}$ and $\mathbf{W}^{l,B}_{\text{up}} \in \mathbb{R}^{hr\times p}$. Here, p represents the tunable intermediate dimension of the MLP module in $\mathcal{F}'_\phi$, and Reshape$(\cdot)$ transforms the output vector into the shape of $\mathbf{B}$. This process is applied at each layer l, so we concatenate the layer index with $\mathbf{s}_i$. We provide the detailed visualization of this workflow in appendix I.

A same procedure is followed for matrices $\mathbf{A}$ and in other modules of FFN (i.e., down-proj and gate-proj). The goal of $\mathbf{P}'_i$ generated by $\mathcal{F}'_\phi$ is to fully memorize the document information within its parameters and perform as effectively as $\mathbf{P}_i$.

Training Objectives. To learn a well-aligned parameter translator $\mathcal{F}'_\phi$ that ideally matches the behavior of the general mapping function $\mathcal{F}$, we design the training objective from three complementary perspectives. (1) *Language Modeling Loss*: We optimize $\mathcal{F}'_\phi$ using the augmented dataset $\mathcal{D}_i$ and the same language modeling objective as in eq. (2) to predict the next-token correctly, denoted as $\mathcal{L}_{\text{lm}}$. (2) *Parameter Discrepancy Loss*: For the target LoRA adapter $\mathbf{P}_i$, we introduce a mean squared error (MSE) loss to measure the discrepancy between the generated parameters and the target parameters, denoted as $\mathcal{L}_{\text{mse}}$. (3) *Distribution Alignment Loss*: We further employ the Kullback–Leibler divergence (Polzehl and Spokoiny, 2006), denoted as $\mathcal{L}_{\text{kl}}$, to align the word-level probability distributions of the two models, where the model equipped with $\mathbf{P}_i$ serves as the target distribution. The overall training objective is formulated as:

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

$$\mathcal{L}_{\text{kl}} = \text{KL}(P_{\mathbf{\Theta}+\mathbf{P}_i}(x), P_{\mathbf{\Theta}+\mathcal{F}'_\phi(s_i)}(x)) \tag{6}$$

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{lm}} + \lambda_1 \mathcal{L}_{\text{mse}} + \lambda_2 \mathcal{L}_{\text{kl}} \tag{7}$$

where we calculate the overall alignment loss for each document d_i, λ_1 and λ_2 are tunable hyperparameter which set to 100 and 0.01 separately.

4.2 Inference with Knowledge Adaptation

In fig. 2, the RAG system suffers from static parameters, which hinder LLMs from effectively utilizing updated context due to the low knowledge overlap between these two distinct knowledge types. In contrast, once a well-trained parameter translator $\mathcal{F}'_\phi$ is obtained in DyPRAG, knowledge adaptation can be efficiently performed with parametrized document information at inference-time, significantly increasing the knowledge overlap.

Specifically, given a test query q^t, we rerun the retrieval process using the retrieval module $\mathcal{R}$ to select the most relevant documents. For each selected document d_i^t, we extract its embedding $\mathbf{s}_i^t$ and feed it into $\mathcal{F}'_\phi$ to obtain the dynamic LoRA adapter $\mathbf{P}_i^{t,\prime}$, which encodes the relevant document information in parametric form. These adapters are then averaged as the LoRA parameter for inference, enabling updated parametric knowledge during inference with minimal computational cost. The entire process of DyPRAG is illustrated in fig. 3.

5 Experiments

5.1 Experiments Details

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

Evaluation Metrics. For evaluation, we use the Exact Match (EM) score (%) to compare the extracted answer with the reference answer at the exact match level. Additionally, we employ the F1 score (%), which balances precision and recall by considering partially correct answers.

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

5.2 Baselines

We compare **DyPRAG** with a set of baselines that either leverage retrieved context or without external retrieval. For methods with retrieved context, **RAG** appends retrieved documents to the input and explicitly instructs the model to reference them, while **Context-DPO** (Bi et al., 2024) applies direct preference optimization (DPO) (Rafailov et al., 2023) to improve context faithfulness. **PRAG** transforms documents into parameters through offline parameterization, and **DyPRAG** further extends this idea by dynamically translating retrieved documents into parameters at inference-time. For methods without retrieved context, **Vanilla** denotes the base

Method	2WQA		HQA		PQA		CWQ		Avg	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
LLaMA3.2-1B										
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
PRAG-woC	20.13	25.92	19.00	25.35	12.00	23.58	26.00	35.86	19.57	26.51
DyPRAG-woC	24.27	29.91	19.56	25.97	7.33	11.33	28.33	36.86	21.57	27.57
RAG	17.93	24.77	21.44	30.33	9.67	17.65	25.67	37.39	18.93	26.99
Context-DPO	19.33	24.14	17.00	23.35	4.00	12.79	7.67	13.00	15.93	21.66
PRAG	20.60	26.94	23.33	**30.81**	20.33	**31.07**	28.33	**39.63**	22.17	29.78
DyPRAG	**26.33**	**32.53**	**23.33**	30.80	**21.33**	29.93	**29.33**	38.96	**25.23**	**31.80**
Qwen2.5-1.5B										
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
PRAG-woC	**21.93**	**29.38**	16.00	24.04	1.33	3.87	22.31	30.82	18.13	25.37
DyPRAG-woC	21.87	28.46	17.11	24.93	3.00	6.64	22.67	31.94	18.64	25.56
RAG	16.33	23.89	14.89	24.68	0.67	9.97	18.64	28.23	14.56	23.17
Context-DPO	17.60	24.35	15.00	24.35	0.33	14.18	12.33	19.20	14.57	22.82
PRAG	19.07	27.29	19.33	26.15	2.67	12.61	21.67	32.13	17.77	25.96
DyPRAG	18.87	25.87	**20.67**	**30.13**	**7.33**	**22.69**	**23.67**	**33.57**	**18.74**	**27.60**
LLaMA3-8B										
Vanilla	30.00	36.43	19.89	28.64	4.67	7.96	30.00	42.44	24.43	31.85
SFT	1.53	13.09	0.33	2.19	0.00	0.00	0.00	5.92	0.86	7.80
PRAG-woC	33.20	40.54	35.55	45.88	20.33	26.13	**32.67**	**43.54**	32.57	41.00
DyPRAG-woC	32.07	39.17	24.67	37.33	11.00	13.60	**32.67**	41.87	27.80	36.23
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	34.47	42.20	**40.11**	**50.82**	11.33	26.23	28.00	36.41	**33.20**	42.61
DyPRAG	**36.33**	**47.68**	33.22	43.22	**21.00**	**32.86**	29.67	39.07	**33.20**	**43.69**

Table 1: In-domain experimental results. All metrics are reported as EM scores (%) and F1 scores (%). The best performance is **bolded**, while the second-best is underlined. The **Avg** is the average performance over all tasks.

LLM without external knowledge, **SFT** fine-tunes the LLM under the same setting as DyPRAG to encode all knowledge into parameters, and **PRAG-woC** and **DyPRAG-woC** are ablated variants that remove retrieved context.

5.3 Main Results

In this section, we present the main experimental results and a detailed analysis of DyPRAG compared to the selected baselines. Detailed cost analysis is provided in appendix A. Additional experiments are provided in appendix C. Notably, the vanilla model occasionally outperforms RAG in certain cases, we analyze the reasons in appendix G and confirm it does not impact the subsequent analysis.

In-Domain Results. As shown in table 1, by integrating symbolic passages with inference-time updated parametric knowledge, DyPRAG consistently achieves the best performance across all evaluated models, outperforming all baselines. Specifically, DyPRAG surpasses PRAG by 2.02% (3.06%) on LLaMA3.2-1B, 1.64% (0.97%) on Qwen2.5-1.5B, and 1.08% (0.00%) on LLaMA3-8B on average in F1 (EM) scores. Moreover, the parameter representations transformed by $\mathcal{F}'_\phi$ remain effective even without retrieved context. For example, on LLaMA3.2-1B, DyPRAG-woC achieves an average score of 27.57% (21.57%), outperforming PRAG-woC by 1.06% (2.00%), RAG by 0.58% (2.64%), and the vanilla model by 5.18% (4.83%) in F1 (EM) scores. In contrast, Context-DPO is less effective at resolving knowledge conflicts, while SFT suffers from severe collapse when encoding large-scale knowledge, leading to substantial performance degradation. Overall, these results demonstrate the consistent advantages of DyPRAG over all baselines, highlighting its effectiveness for inference-time parametric knowledge adaptation.

Out-of-Domain Results. To further assess the generalization capability of DyPRAG, we evaluate its performance under out-of-domain (OOD) settings. Notably, PRAG cannot be directly applied to OOD scenarios without additional offline training. We conduct OOD evaluations on several commonsense benchmarks, including StrategyQA (**SQA**) (Geva et al., 2021), **IIRC** (Ferguson et al., 2020), and OpenBookQA (**OBQA**) (Mihaylov et al., 2018). In addition, MedMCQA (**MQA**) (Pal et al., 2022) represents a completely

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	**52.00**	50.33	42.60
DyPRAG-woC	14.04	39.67	43.00	40.67	34.35
DyPRAG	**41.91**	**50.33**	**52.00**	**52.67**	**49.23**
Qwen2.5-1.5B					
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	**52.67**	41.80
DyPRAG-woC	10.23	15.67	43.38	34.67	25.99
DyPRAG	**38.25**	43.33	**48.57**	**52.67**	**45.71**
LLaMA3-8B					
Vanilla	13.23	33.33	52.33	55.00	38.47
SFT	2.42	0.00	0.00	0.00	0.61
RAG	43.27	45.67	60.00	55.67	51.15
DyPRAG-woC	18.16	45.67	53.00	55.00	42.96
DyPRAG	**57.90**	**58.67**	**60.67**	**56.67**	**58.48**

Table 2: Out-of-domain experimental results. All datasets are provided with ground-truth documents.

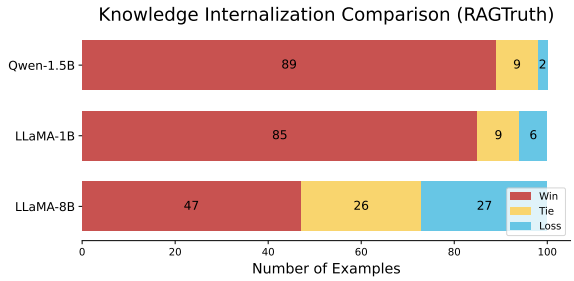

Figure 4: Comparison of knowledge internalization between DyPRAG vs RAG judged by GPT-4o.

Method	Retriever	Avg	
		EM	F1
Vanilla	None	24.43	31.85
RAG	BM25	**23.04**	**30.86**
	all-MiniLM-L6-v2	17.70	24.95
	Qwen3-Embedding-0.6B	16.63	23.82
DyPRAG-woC	BM25	**27.80**	**36.23**
	all-MiniLM-L6-v2	19.67	26.46
	Qwen3-Embedding-0.6B	19.56	26.54
DyPRAG	BM25	**33.20**	**43.69**
	all-MiniLM-L6-v2	18.70	26.17
	Qwen3-Embedding-0.6B	19.07	26.98

Table 3: Ablation study of retriever averaged on ID datasets. The backbone model is the LLaMA3-8B.

unseen domain focused on medical knowledge. All OOD datasets are provided with ground-truth passages, with details described in appendix B.1.

As shown in table 2, the vanilla model performs poorly due to insufficient domain-relevant knowledge. DyPRAG-woC improves performance by enhancing parametric knowledge, yielding moderate gains across most datasets. However, when answering questions that critically depend on document-specific information, DyPRAG-woC struggles to accurately reconstruct the required content. This limitation primarily arises from information loss during the encoding and translation processes. In contrast, DyPRAG integrates golden passages with dynamic parametric knowledge, substantially increases knowledge overlap and achieves the best performance across all OOD scenarios.

Furthermore, the retrieved documents in main experiments are primarily sourced from Wikipedia, which is largely covered during pre-training. To examine performance on truly unseen documents, we further evaluate DyPRAG on the RAGTruth benchmark (Niu et al., 2023), which poses greater challenges as the correct answers are only accessible through carefully constructed contexts. As shown in fig. 4, DyPRAG significantly outperforms RAG, demonstrating its ability to better internalize contextual knowledge and mitigate knowledge conflicts even when handling unseen data.

5.4 Ablation Study

Ablation of Different Retriever. Retrieval quality is critical to RAG, as it determines whether the model accesses the information required to answer a question. We compare lexical retrieval and dense retrieval within the DyPRAG framework. Specifically, we evaluate BM25 as a representative lexical retriever (Robertson et al., 2009) and two dense retrievers, all-MiniLM-L6-v2 and Qwen3-Embedding-0.6B. As shown in table 3, BM25 consistently outperforms dense retrieval methods across all evaluated datasets, despite the strong performance of dense retrievers in general IR tasks. These results are consistent with prior studies (Ram et al., 2023; Su et al., 2024), reaffirming BM25 as a robust and effective baseline for RAG.

Ablation of Parameter Translator Size. As shown in table 4, the storage cost of $\mathcal{F}'_\phi$ is $3L(phr + 2p(h+1) + pkr)$ and scales linearly with p. Results in table 7 show that under no-context scenario, DyPRAG outperforms RAG and PRAG consistently, with $p = 2$ achieving the second-best performance at only 7.71 MB of storage. In contrast, PRAG requires 9.33 GB to store all queries, incurring substantial overhead. Overall, DyPRAG markedly reduces cost with improved performance.

Ablation of Retrieved Number. As shown in fig. 10, performance in both RAG and DyPRAG-woC fluctuates with the number of retrieved documents, with the best results generally achieved at $c = 3$. This indicates that introducing less relevant context can degrade model performance.

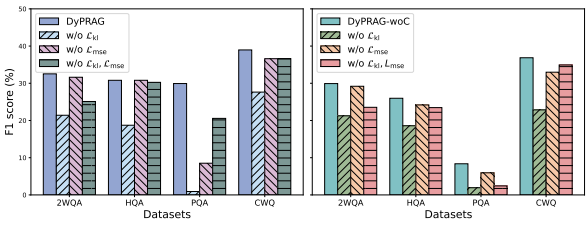

Figure 5: Ablation study of alignment loss. The backbone model is the LLaMA3.2-1B.

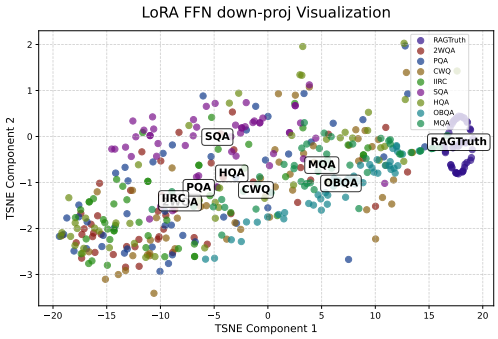

Figure 6: t-SNE visualization of down-proj average vector in LoRA FFN generated by parameter translator.

Since DyPRAG achieves better performance than RAG, inference time becomes a critical dimension. As shown in table 9, DyPRAG demonstrates higher inference efficiency than RAG, particularly as the number of documents increases. While RAG suffers from substantial latency growth with more documents, DyPRAG maintains lower inference time by enabling rapid loading of document-specific LoRA modules with special code design and producing shorter responses, as shown in fig. 7.

Although DyPRAG introduces additional document encoding and parameter translation costs that scale with the number of documents, these operations are not yet fully optimized. In practical deployments, RAG systems are often implemented in asynchronous architectures (e.g., message-queue systems (Kreps et al., 2011)), where document encoding and translation can be executed in parallel during queue waiting time. Under this setting, DyPRAG incurs no additional online latency, allowing it to achieve superior performance while remaining more inference-efficient than RAG.

Ablation of Alignment Loss. The alignment loss consists of three terms: $\mathcal{L}_{\text{lm}}$, $\mathcal{L}_{\text{mse}}$, and $\mathcal{L}_{\text{kl}}$. As shown in fig. 5, removing any component consistently degrades performance. Notably, excluding $\mathcal{L}$kl results in a substantial drop, underscoring the importance of aligning the model with the target output distribution. Overall, these results validate the effectiveness of our training objectives. Additional ablation studies are provided in appendix D.

5.5 In-depth Analysis

Updated Parametric Space Enhances Knowledge Overlap. When LLMs fail to identify reliable information sources (Tao et al., 2024; Zhang et al., 2024), the root cause is often conflicts between context and parameter arising from low knowledge overlap, which supports by the LEN trends in fig. 2. We further analyze more internal signals to verify this pattern. As shown in table 10, Entropy (EN) and Lexical Similarity (LS) (Lin et al., 2022) exhibit trends consistent with prior observations: RAG increases uncertainty, while DyPRAG consistently lowers EN and increases LS. These results confirm that dynamic parametrization in DyPRAG effectively increases knowledge overlap and alleviates knowledge conflicts. More analysis is placed in appendix E.

Analysis of Parametric Representations. Since the hypernetwork is difficult to interpret directly, we examine whether it generates document-specific parameters. As shown in fig. 6, the down-proj weights in the FFN exhibit substantial diversity across input documents. Moreover, similarity analysis fig. 11 conducted separately on the document representations s_i and the translated parameters $\mathbf{P}_i'$ shows significantly lower similarity on OOD datasets than on ID datasets, suggesting that DyPRAG generalizes effectively with various input. Additional analysis is provided in appendix F.1.

6 Conclusion

In this work, we propose Dynamic Parametric RAG (DyPRAG), a novel framework that successfully learns the underlying mapping function from documents to parameters by leveraging a hypernetwork, enabling effective parametric knowledge adaptation at inference-time. Extensive experiments across diverse datasets demonstrate the effectiveness and generalization capability of DyPRAG. Moreover, our in-depth analysis shows that DyPRAG substantially increases the overlap between the model's parametric knowledge and the retrieved symbolic information, thereby mitigating knowledge conflicts and enabling more effective knowledge fusion with minimal cost, highlighting its potential for real-world RAG applications.

Limitations

While our framework demonstrates strong performance on standard RAG systems, it has not yet been validated in broader settings. In particular, with the increasing prevalence of agentic RAG, the effectiveness of dynamic parameterization in such multi-step, decision-driven pipelines remains unexplored. Moreover, although the parameter translator can be viewed as a form of parametric memory, its behavior under continual learning scenarios has not been empirically evaluated, leaving its robustness in long-term or incremental update settings unverified.

Ethical considerations

Our approach does not introduce ethical concerns. The datasets we used are public, and there are no privacy issues.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Zeyuan Allen-Zhu and Yuanzhi Li. 2023. Physics of language models: Part 3.1, knowledge storage and extraction. arXiv preprint arXiv:2309.14316.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, and 1 others. 2024. Context-dpo: Aligning language models for context-faithfulness. arXiv preprint arXiv:2412.15280.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In International conference on machine learning, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. 2025. Text-to-lora: Instant transformer adaption. arXiv preprint arXiv:2506.06105.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. arXiv preprint arXiv:2405.13792.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. arXiv preprint arXiv:1903.00161.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. arXiv e-prints, pages arXiv–2407.

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. IIRC: A dataset of incomplete information reading comprehension questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1137–1147, Online. Association for Computational Linguistics.

Giacomo Frisoni, Alessio Cocchieri, Alex Presepi, Gianluca Moro, and Zaiqiao Meng. 2024. To generate or to retrieve? on the effectiveness of artificial contexts for medical open-domain question answering. arXiv preprint arXiv:2403.01924.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. arXiv preprint arXiv:2307.06945.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. Transactions of the Association for Computational Linguistics, 9:346–361.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

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

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

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

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

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

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

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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

Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. 2024. Knowledge entropy decay during language model pretraining hinders new knowledge acquisition. arXiv preprint arXiv:2410.01380.

Jay Kreps, Neha Narkhede, Jun Rao, and 1 others. 2011. Kafka: A distributed messaging system for log processing. In Proceedings of the NetDB, volume 11, pages 1–7. Athens, Greece.

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

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

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

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

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

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

Meta. 2024a. Llama-3.2-1b-instruct. Accessed: 2024-09.

Meta. 2024b. Meta-llama-3-8b-instruct. Accessed: 2024-04.

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

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

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

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

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Proceedings of the Conference on Health, Inference, and Learning, volume 174 of Proceedings of Machine Learning Research, pages 248–260. PMLR.

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

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

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

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

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval, 3(4):333–389.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

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

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

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

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

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

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

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

Yan Wang, Dongyang Ma, and Deng Cai. 2024c. With greater text comes greater necessity: Inference-time training helps long text generation. In First Conference on Language Modeling.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115.

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

Hao Zhang, Yuyang Zhang, Xiaoguang Li, Wenxuan Shi, Haonan Xu, Huanshuo Liu, Yasheng Wang, Lifeng Shang, Qun Liu, Yong Liu, and 1 others. 2024. Evaluating the external and parametric knowledge fusion of large language models. arXiv preprint arXiv:2405.19010.

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).

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 $\mathbf{P}_i$. 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 4.

Inference Cost. We first analyze the inference cost across three baselines. For DyPRAG, there is additional cost incurred for encoding and translating. The encoding cost is $c \times (|d|^2 \times \text{ATTN} + |d| \times \text{FFN})$, as each document should be encoded separately. As shown in table 9, the encoding time is significantly lower than the inference time because encoding requires only a single forward pass. Additionally, the translation time is also negligible. Moreover, the response length $|R|$ exhibits a linear relationship with the LLM inference cost. As illustrated in fig. 7, the response length decreases when DyPRAG is employed, enabling LLMs to better internalize knowledge. Notably, DyPRAG achieves much shorter response lengths, significantly reducing inference costs compared to standard RAG. We further discuss this in section 5.4.

Training Cost. PRAG (Su et al., 2025) introduces further training for each document to obtain corresponding LoRA parameters. In appendix A, we hypothesize that after augmentation, there are 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 and $M \times (9|d|^2 \times \text{ATTN} + 3|d| \times \text{FFN})$ for PRAG, where N represents the size of the training dataset $\mathcal{K}$ and M denotes the size of the test set. The common divisor of offline parametrization is $E_1 \times (81|d|^2 \times \text{ATTN} + 9|d| \times \text{FFN})$, where E_1 is the number of epochs for LoRA training.

Additionally, to train our $\mathcal{F}'_\phi$ for E_2 epochs, we need to perform both forward and backward passes (the backward pass requires twice the cost of the forward pass) on one QA pair and its corresponding document in each step. This results in a cost of $N \times E_2 \times 9(|qa|+|d|)^2 \times \text{ATTN} + 3(|qa|+|d|) \times \text{FFN}$, with a negligible cost for translation. As shown in fig. 8 and fig. 9, our DyPRAG achieves stable results with as few as 480 examples (even fewer is powerful), while $M = 3000$ in our experiments, and this value would be significantly larger in real-world applications.

For instance, using LLaMA3-8B as the backbone, producing a $\mathbf{P}_i$ requires 88 seconds, while one step for $\mathcal{F}'_\phi$ only takes an average of 15 seconds. Therefore, the total cost for training (excluding augmentation) is $M \times 88s$ in PRAG and $N \times 103s$

Method	Inference Cost	Training Cost	Storage Cost
RAG	$\|R\| \times ((c\|d\| + \|q\|)^2 \times \text{ATTN} + (c\|d\| + \|q\|) \times \text{FFN})$	-	-
PRAG	$\|R\| \times ((c\|d\| + \|q\|)^2 \times \text{ATTN} + (c\|d\| + \|q\|) \times \text{FFN})$	$M \times (9\|d\|^2 \times \text{ATTN} + 3\|d\| \times \text{FFN}) +$ $M \times E_1 \times (81\|d\|^2 \times \text{ATTN} + 9\|d\| \times \text{FFN})$	$M \times 3Lr(h+k)$
DyPRAG	$c \times (\|d\|^2 \times \text{ATTN} + \|d\| \times \text{FFN}) +$ $c \times O(p(h+1+hr)) +$ $\|R\| \times ((c\|d\| + \|q\|)^2 \times \text{ATTN} + (c\|d\| + \|q\|) \times \text{FFN})$	$N \times (9\|d\|^2 \times \text{ATTN} + 3\|d\| \times \text{FFN}) +$ $N \times E_1 \times (81\|d\|^2 \times \text{ATTN} + 9\|d\| * \text{FFN}) +$ $N \times E_2 \times (9(\|qa\| + \|d\|)^2 \times \text{ATTN} + 3(\|qa\| + \|d\|) \times \text{FFN}) + O(p(h+1+hr))$	$\cancel{N * 3Lr(h+k)} +$ $3L(phr + 2p(h+1) + pkr)$

Table 4: Comparison of cost metrics for different baselines. ATTN denotes the time complexity of the self-attention module as $O(|I|^2 h)$, and FFN represents the FFN with $O(|I|h^2)$, where context length $|I| = 1$ and $|R|$ denotes the response length. ■ indicates high cost, _ denotes negligible cost, and ／ represents temporal storage.

in DyPRAG. Assuming $N = 480$ and $M = 3000$, DyPRAG is 5.34x faster than PRAG. The low requirement for a large N makes DyPRAG highly effective and generalizable for real-world scenarios, with extremely low costs that can be handled during offline training.

Storage Cost. As illustrated in appendix A, each $\mathbf{P}_i$ requires 3.36MB for PRAG using Qwen2.5-1.5B, resulting in a total storage cost of 9.33GB in our main experiment. However, we significantly reduce this cost by imitating the underlying function between the document and parameters. Notably, the cost for $\mathbf{P}_i$ is a temporary cost in DyPRAG, which can be removed after collecting data or training one $\mathbf{P}_i$ and then updating $\mathcal{F}'_\phi$ by one step. Consequently, the overall cost of DyPRAG is substantially lower than that of PRAG (e.g., DyPRAG achieve better performance with only 7.71MB of storage as shown in table 7).

B Experiment Setup

B.1 Implementation Details

In-domain QA Datasets. To ensure a comprehensive evaluation, we assess our method using the following datasets:

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

Both 2WQA and HQA categorize questions by reasoning type, with 2WQA having four categories and HQA two. To compare DyPRAG with other RAG baselines across reasoning tasks, we use the first 300 questions from each sub-dataset for evaluation.

Offline Doc-Param Pairs Collection. Following (Jiang et al., 2023; Su et al., 2025), we utilize Wikipedia dumps as the external knowledge corpus, adopting the dataset proposed by DPR (Karpukhin et al., 2020). For document augmentation, each document is rewritten once, and three QA pairs are generated based on the document. Unless explicitly stated otherwise, the downstream LLM is used for this purpose. During LoRA fine-tuning, the learning rate was set to 3×10^{-4}, and training was conducted for a single epoch (except PQA for 2). The LoRA modules were integrated exclusively into the feed-forward network (FFN) matrices, while the query, key, and value (QKV) matrices were excluded. The scaling factor α was set to 32, the LoRA rank r was configured to 2, and no dropout was applied to ensure training stability and maximize parameter updates. The LoRA weights were randomly initialized following the settings outlined in the original LoRA paper (Hu et al., 2022).

Baselines Implementation. To conduct comprehensive experiments, we compare our DyPRAG with two commonly used baselines: SFT and Context-DPO, alongside parametric and non-parametric RAG baselines. For SFT, widely regarded as a standard approach for adapting models to various downstream tasks, is included to evaluate the generalization ability of DyPRAG. Specifically, we use the exact same hyperparameters as

DyPRAG, setting the learning rate to 3×10^{-4}, fine-tuning on the same dataset (i.e., 36,000 samples) with a batch size of 1 for 1 epoch. For Context-DPO, we follow the implementation described in Bi et al. (2024). To ensure a fair comparison, we configure the trainable LoRA modules for both methods to match those in DyPRAG, maintaining equivalent parameter learning capacity. The LoRA modules are integrated exclusively into the FFN, while the query, key, and value matrices are excluded. The scaling factor α is set to 32, and the LoRA rank r is configured as 2.

Inference Settings. All experiments use the publicly available Hugging Face implementations of LLaMA and Qwen. To ensure fairness, DyPRAG and all baselines share the same prompt template in fig. 13 and fig. 14 following Su et al. (2025) and adopt of greedy decoding for result reproducibility. The max number of new tokens is set to 128.

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

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

Implementation of OOD Experiment. To evaluate the generalization ability of our proposed DyPRAG, we select to out-of-domain (OOD) datasets to conduct:

- **StrategyQA (SQA)** (Geva et al., 2021): A QA benchmark where reasoning steps are implicit in the question and must be inferred through strategic reasoning, including human-curated evidence paragraphs from Wikipedia.
- **IIRC** (Ferguson et al., 2020): A dataset comprising over 13,000 questions based on English Wikipedia paragraphs that provide only partial information and supplemented with samples from SQuAD 2.0 (Rajpurkar et al., 2016) and DROP (Dua et al., 2019), requiring retrieval of missing details from linked documents.
- **OpenBookQA (OBQA)** (Mihaylov et al., 2018): A multiple-choice QA dataset derived from a subset of WorldTree (Jansen et al., 2018), mainly focus on common knowledge.
- **MedMCQA (MQA)** (Pal et al., 2022): A multiple-choice QA dataset designed to address real-world medical domain entrance exam questions.

For each dataset, we select the first 300 examples for testing and evaluate performance using F1 score for IIRC, Accuracy for SQA, Recall for OBQA and MQA as metrics. Both datasets provide with ground-truth passages which indicate a more rigorous evaluation setting. For IIRC, we adopt the few-shot prompts from Su et al. (2024), while SQA, OBQA and MQA are evaluated in a zero-shot setting. Notably, the same prompt format (in fig. 13 and fig. 14) from the main experiment is used to ensure a fair comparison.

Implementation of RAGTruth Experiment. RAGTruth (Niu et al., 2023) is a benchmark dataset designed to evaluate the extent of hallucination in models. For our evaluation, we randomly select 100 QA-type subsets from RAGTruth, ensuring alignment with the training data of $\mathcal{F}'_\phi$. Notably, some questions in RAGTruth require the provided documents to be answerable which are more difficult. Interestingly, during evaluation, we observe that $\mathcal{F}'_\phi$ with fewer trained parameters perform better in such scenarios. Specifically, we train only 480 examples for LLaMA3.2-1B and Qwen2.5-1.5B, and 240 examples for LLaMA3-8B. We use GPT-4o as judge using prompt template in fig. 15.

C Supplement Experiment Results

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

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

Base LLM	Method	2WQA	HQA	Avg
		Total	Total	
LLaMA3.2-1B	RAG	23.12	**27.14**	25.13
	DRAGIN	21.73	12.50	17.12
	FLARE	21.55	19.38	20.47
	DyPRAG-woC	25.31	19.97	22.64
	DyPRAG	**29.18**	26.58	**27.88**
Qwen2.5-1.5B	RAG	24.31	20.73	22.52
	DRAGIN	25.01	8.51	16.76
	FLARE	21.56	7.97	14.77
	DyPRAG-woC	**26.46**	19.67	23.07
	DyPRAG	25.18	**27.57**	**26.38**
LLaMA3-8B	RAG	34.55	24.23	29.39
	DRAGIN	35.69	12.16	23.93
	FLARE	34.62	29.43	32.03
	DyPRAG-woC	37.25	22.55	29.90
	DyPRAG	**45.17**	**38.35**	**41.76**

Table 5: The experimental results of DyPRAG are compared with other effective RAG methods. All metrics are reported as F1 scores (%). The best performance is bolded, while the second-best is underlined. The evaluation is conducted on 2WQA and HQA datasets, focusing exclusively on the total sub-task.

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

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

Comparison of Response Length. Notably, we consider only the context length when calculating inference cost. However, in practice, the response length from LLMs also affects inference time. As shown in fig. 7, we compare DyPRAG with RAG across four benchmarks, considering the average response length. DyPRAG significantly reduces response length, by 20% in 2WQA and up to 90% in CWQ. This demonstrates that DyPRAG can answer questions correctly with fewer tokens, thereby lowering inference costs and avoiding redundant information.

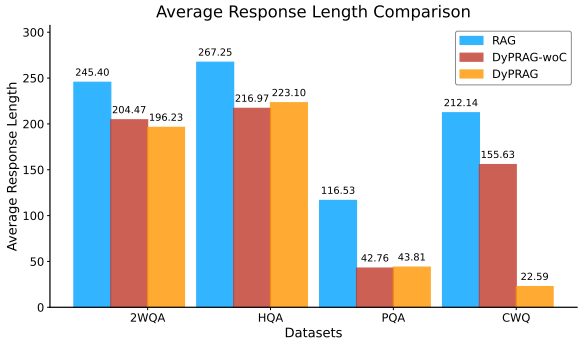

Figure 7: Comparison of response length across various datasets. The backbone model is the Qwen2.5-1.5B.

Performance of DyPRAG on Non-Instruct Models. With the rapid advancement of reinforcement learning, a growing number of long-context models, referred to as large reasoning models (LRMs) have emerged (Guo et al., 2025; Yang et al., 2025). Our goal is to evaluate whether the current design of DyPRAG can adapt effectively to such up-to-date models. For this purpose, we selected Qwen3-8B[1] (a reasoning model) and Qwen3-4B-Instruct[2] (an instruct model) for experiments. As shown in table 6, the performance of Qwen3-8B decreases significantly when DyPRAG generated parameters are applied. This decline is primarily due to differences in answer patterns. LRMs tend to generate extremely lengthy reasoning trajectories, whereas our method only augments simple and short QA pairs. In contrast, the results for the instruct model, Qwen3-4B-Instruct, align with our main experiments, demonstrating that the current method is well-suited for instruct models. To enable compatibility with LRMs, the parameter translation process needs to be integrated into the reinforcement learning training pipeline. Addressing this challenge will be a focus of our future work.

Base LLM	Method	2WQA (Total)		HQA (Total)		PQA		CWQ		Avg	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Qwen3-8B	Vanilla	24.67	31.33	21.00	28.12	0.00	0.40	22.33	**36.01**	17.00	23.97
	RAG	**35.33**	**42.26**	**32.33**	**44.00**	**0.33**	**9.17**	**23.00**	35.79	**22.75**	**32.81**
	DyPRAG-woC	21.00	27.94	20.33	27.82	0.00	0.46	17.67	29.12	14.75	21.34
	DyPRAG	31.00	38.37	29.67	39.81	**0.33**	4.54	20.00	31.37	20.25	28.52
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	25.33	36.62	18.67	26.32	2.00	7.36	17.92	25.78
	DyPRAG-woC	27.00	35.44	16.33	24.00	10.00	13.49	0.33	4.47	13.42	19.35
	DyPRAG	**31.00**	**38.37**	**29.67**	**39.81**	**20.67**	**27.33**	**8.67**	**19.14**	**22.50**	**31.16**

Table 6: The experimental results of DyPRAG are compared with standard RAG based on Qwen3-8B and Qwen3-4b-Instruct. All metrics are reported as EM scores (%) and F1 scores (%). The best performance is bolded, while the second-best is underlined. The **Avg** is the average performance over all tasks.

Method	CWQ F1	Storage Cost (MB)
Vanilla	26.47	-
RAG	28.32	-
PRAG-woC	30.82	19107.84 (1x)
DyPRAG-woC ($p = 2$)	32.66	7.71 (0.04%x)
DyPRAG-woC ($p = 4$)	33.26	15.42 (0.08%x)
DyPRAG-woC ($p = 16$)	32.08	61.70 (0.32%x)
DyPRAG-woC ($p = 32$)	31.94	123.39 (0.64%x)

Table 7: Ablation study of intermediate dimension p of $\mathcal{F}'_\phi$. The backbone model is the Qwen2.5-1.5B.

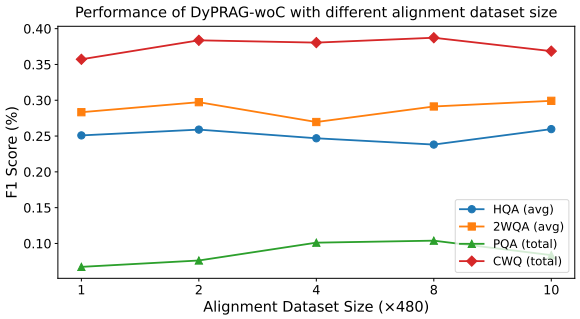

Figure 8: Ablation study of varying training dataset size for DyPRAG. The backbone model is the LLaMA3.2-1B.

D Additional Ablation Experiment Results

Effect of Training Dataset Size. We adjust the pre-selected size of the training dataset composed of Doc-Param pairs, increasing it from 480 to 4800. As shown in fig. 8 and fig. 9, DyPRAG achieves strong performance even with just 480 training examples. The performance remains remarkably stable 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.

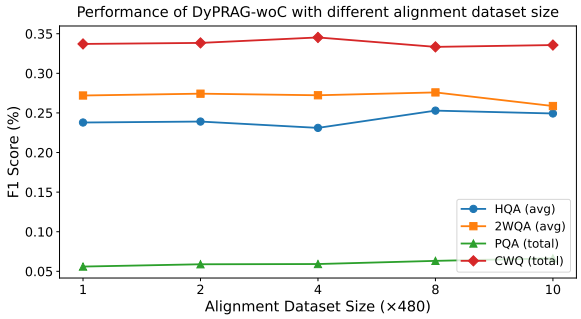

Figure 9: Ablation study of varying training dataset size for DyPRAG. The backbone model is the Qwen2.5-1.5B.

Effect of Data Augmentation. In section 4, we introduce data augmentation to improve the model's ability to memorize and process information from documents. To assess the impact of data augmentation on the DyPRAG method, we remove it during the alignment dataset construction phase and compare the results with those of the original method. The results in table 8 indicate that removing data augmentation greatly diminishes the quality of offline parameterization, which in turn affects the parameter translator's ability to convert documents into parametric knowledge. This degradation results in a significant performance drop for both PRAG, which relies on offline parameterization, and DyPRAG, which dynamically converts parameters.

E Exploring Metrics for Knowledge Conflicts Detection

Research has shown that generating multiple outputs for a single input improves estimation of sequence-level uncertainty. Accordingly, we set temperature=1.0, top_p=0.95, and top_k=20, and

[1]https://huggingface.co/Qwen/Qwen3-8B
[2]https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507

Method	2WQA (Total)	HQA (Total)	PQA	CWQ	IIRC	SQA	OBQA	MQA
Vanilla	26.87	17.76	2.87	26.47	8.78	1.00	40.09	33.67
RAG	24.31	20.73	9.97	28.23	30.52	39.00	45.00	52.67
PRAG	27.49	23.10	23.43	32.13	–	–	–	–
w/o Aug	22.79	19.00	10.74	28.54	–	–	–	–
Change	-17.1%	-17.7%	-54.2%	-11.2%	–	–	–	–
DyPRAG-woC	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	+7.2%	-20.1%	-49.5%	-12.2%	-17.0%	-98.1%	-11.6%	-33.8%
DyPRAG	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	-8.7%	-27.9%	-56.6%	-16.7%	-23.1%	-29.2%	-9.6%	-35.4%

Table 8: Ablation study of effectiveness in data augmentation. All metrics are reported as F1 scores (%). The backbone model is the Qwen2.5-1.5B.

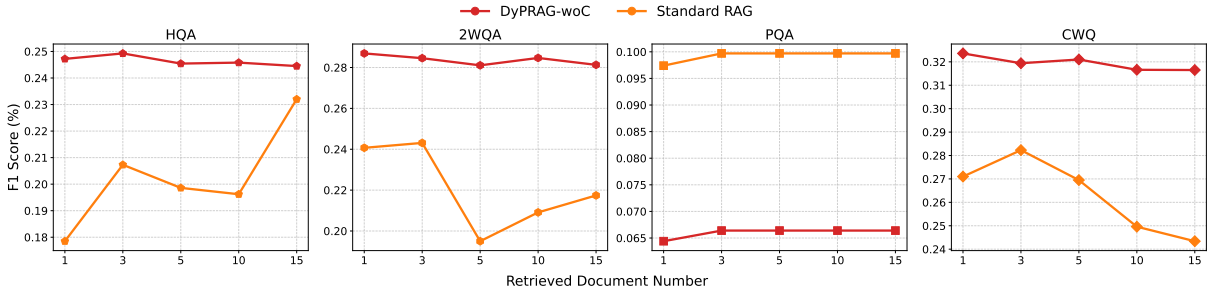

Figure 10: Ablation study of varying number of retrieved documents to RAG and DyPRAG's performance. The backbone model is the Qwen2.5-1.5B.

generated five responses to compute Entropy (EN), length-normalized entropy (LEN) (Malinin and Gales, 2020), and Lexical Similarity (LS) (Lin et al., 2022) for assessing the probability of knowledge conflicts[3]. As shown in table 10, our approach reduces knowledge conflicts in most settings, with the largest improvement achieved by our DyPRAG.

We further observe that appending retrieved documents increases EN and LEN while decreasing LS, indicating that retrieved passages in RAG systems often conflict with the model's parametric knowledge. By contrast, updating converted parametric knowledge via DyPRAG substantially lowers the likelihood of such conflicts, demonstrating the effectiveness of DyPRAG.

F Detailed Analysis of Parametric Representations

F.1 Embedding similarity across datasets

To train our parameter translator, we utilized datasets, including 2WikiMultihopQA, HotpotQA, PopQA, and ComplexWebQuestions. To evaluate

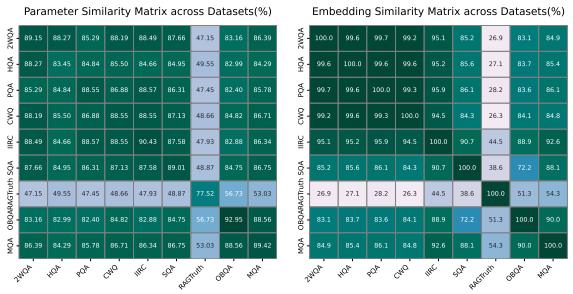

Figure 11: Analysis of input and output similarity in parameter translator

generalization, we conducted OOD experiments on datasets such as IIRC, StrategyQA, RAGTruth and OpenBookQA.

For ID datasets, the documents were retrieved exclusively from Wikipedia. In contrast, the OOD datasets exhibit diverse sources: IIRC primarily draws from English Wikipedia, supplemented with samples from SQuAD 2.0 (Rajpurkar et al., 2016) and DROP (Dua et al., 2019). StrategyQA includes human-curated evidence paragraphs from Wikipedia. RAGTruth is based on the QA set of MS MARCO (Nguyen et al., 2016), which origi-

[3]We used the implementation available at https://github.com/alibaba/eigenscore.

Documents	Method	Inference Time	Loading Time	Translate Time	Encode Time
3	DyPRAG-woC	**0.84**	0.0037	0.056	0.132
	RAG	1.23	-	-	-
	DyPRAG	**0.36**	0.0037	0.055	0.132
10	DyPRAG-woC	**0.80**	0.0044	0.185	0.433
	RAG	1.54	-	-	-
	DyPRAG	**0.78**	0.0045	0.185	0.432
20	DyPRAG-woC	**0.80**	0.0057	0.361	0.862
	RAG	1.74	-	-	-
	DyPRAG	**1.40**	0.0057	0.361	0.862
30	DyPRAG-woC	**0.80**	0.0067	0.545	1.295
	RAG	2.18	-	-	-
	DyPRAG	**1.96**	0.0067	0.545	1.294

Table 9: Ablation study of varying number of retrieved documents to computation cost. The backbone model is the Qwen2.5-1.5B.

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

Table 10: We present the experimental results for the knowledge conflicts metrics of DyPRAG-woC and DyPRAG, in comparison with Vanilla and Standard RAG. In these metrics, ↑ indicates that higher values are better, while ↓ indicates the opposite. The best performance for each metric is highlighted in **bold**. The backbone model is the LLaMA3.2-1B.

nates from Bing search results. OpenBookQA is a multiple-choice QA dataset derived from a subset of WorldTree (Jansen et al., 2018).

To quantify the differences across datasets, we computed the vector similarity of the mean hidden states (i.e., the last-layer outputs of the final token) across them as shown in right part of fig. 11. As expected, the ID datasets exhibit extremely high similarity (>99%) due to their shared reliance on Wikipedia. In contrast, the OOD datasets show significantly lower similarity with the ID datasets. Although StrategyQA and IIRC primarily depend on Wikipedia, they include additional samples from other sources or incorporate human-curated content, which reduces their similarity to the ID datasets. Notably, RAGTruth demonstrates particularly low similarity, as its samples are carefully selected from MS MARCO to focus exclusively on content related to daily life. This underscores the substantial differences between the training corpora and our OOD evaluation datasets.

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

F.2 Parameter similarity across datasets

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

To investigate this, we collect 20 generated parameters across all datasets and compute the inter-average and intra-average parameter similarity. Since the parameter space itself is non-semantic, we measure similarity using the Frobenius norm: $1 - \frac{||A-B||_F}{\max(||A||_F, ||B||_F)}$. As shown in left part of fig. 11, the similarity of the generated

LoRA parameters strongly correlates with the textual similarity of the inputs. In particular, the model produces significantly different outputs when exposed to distinct contexts, even from the same dataset. Although hypernetwork still lacks well-established interpretability methods, this simple comparison provides evidence that the hypernetwork is indeed mapping from different textual embeddings to diverse parameter space. We hope that future research will develop more comprehensive approaches to explain hypernetwork behavior.

G Why Vanilla Outperforms RAG Occasionally?

In this section, we provide a detailed analysis of why the vanilla model occasionally outperforms RAG. As shown in table 1, the vanilla model surpasses RAG most significantly in 2WQA, as the results vary across different models. For instance, the vanilla model outperforms RAG by 2.62% and 0.99% in Qwen2.5-1.5B and LLaMA3-8B on average in F1, respectively. After analyzing the cases, we identify two key issues that most affect RAG's performance: **1) Poor Retrieval Resulsts.** Following Su et al. (2025), we use BM25 as the retriever which is better than dense retriever in out setting (in table 3). However, in many cases, the retrieved documents contain only similar words rather than relevant content. This results in the provided content being unhelpful or even detrimental to LLMs. **2) Already Seen Data.** During the pre-training stages of the selected LLMs (Yang et al., 2024; Meta, 2024b,a), the external source we use (i.e., Wikipedia) has already been seen. This allows LLMs to answer certain questions independently, especially in simpler tasks like 2WQA. Moreover, the inclusion of incorrect or irrelevant context further degrades the performance.

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 lower than that of RAG, which aligns with our hypothesis. For instance, the vanilla model achieves only 8.78% and 1.00% accuracy on Qwen2.5-1.5B for IIRC and SQA, respectively. In contrast, DyPRAG-woC demonstrates a notable improvement in test-time knowledge, achieving 10.23% and 15.67% accuracy on Qwen2.5-1.5B for IIRC and SQA, respectively. These results highlight the pivotal role of RAG and demonstrate that our proposed DyPRAG can seamlessly incorporate OOD knowledge. Moreover, DyPRAG establishes a stronger RAG paradigm, achieving highest performance under these more challenging conditions. In summary, we argue that this more rigorous experimental setting provides stronger evidence for the effectiveness of our method.

H Further Analysis of Contextual and Parametric Knowledge Conflicts

Updated Parameters Increase LLMs Certainty. As shown in table 11, while vanilla LLMs contain accurate parametric knowledge regarding which director was born later, the introduction of retrieved documents about each director causes contextual knowledge to mislead $\mathcal{M}$, resulting in the incorrect answer "William Lustig" while DyPRAG stays the same. This demonstrates that DyPRAG can effectively reduce the knowledge conflicts problem. In this case, standard RAG often introduces redundant or incorrect information from the context, a phenomenon commonly referred to as RAG hallucination (Sun et al., 2024). In contrast, our proposed DyPRAG effectively incorporates accurate information into parametric knowledge. This allows DyPRAG to align parametric knowledge with contextual knowledge, thereby reducing the likelihood of conflicts and enabling LLMs to rely more consistently on its own knowledge.

Dynamic Parametrization Enhances LLMs at Inference-time. Our DyPRAG serves as an effective plug-and-play technique for enhancing parametric knowledge during inference-time. As demonstrated in table 12, DyPRAG successfully manipulates the original parametric knowledge of LLMs in 14.67% of cases. Therefore, it can directly enhance the model's knowledge during inference without the need for further fine-tuning.

Proportion of Different Combinations. Furthermore, as shown in table 13, when both Vanilla LLMs and RAG give incorrect answers, DyPRAG-woC provides the correct answer 26.33% of the time. This indicates that DyPRAG-woC can effectively update missing parametric knowledge and outperforms RAG methods. Additionally, in cases where the vanilla LLM provides the correct answer (i.e., the model possesses accurate internal knowledge), RAG achieves a correct answer rate of 5.33%, while DyPRAG-woC performs better with a rate of 6.33%, showing that dynamic updated pa-

rameter leads to lower conflicts. Similar trend of DyPRAG is presented in table 14.

These results demonstrate that our proposed DyPRAG updates parametric knowledge at inference-time successfully and mitigates conflicts between internal parametric knowledge and external contextual knowledge through the loading of knowledgeable LoRA adapters.

Question: Which film has the director born later, Diary Of A Maniac or Return Of The Hero ?		
Ground truth: Return Of The Hero		
Retrieved top-1 document: Maniac (1980 film) Maniac is a 1980 American psychological slasher film directed by William Lustig and written by C. A. Rosenberg...		
Method	**Answer**	**Status**
Vanilla	Return Of The Hero	✓
RAG	William Lustig	✗
DyPRAG-woC	Return Of The Hero	✓
DyPRAG (ours)	Return Of The Hero	✓

Table 11: 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.

Question: Which film has the director born later, Miss Sloane or Time Changer ?		
Ground truth: Time Changer		
Retrieved top-1 document: production budget of $13 million. " Miss Sloane " is ranked number 75 by per-theater average on Box Office...		
Method	**Answer**	**Status**
Vanilla	John Frankenheimer	✗
RAG	Miss Sloane	✗
DyPRAG-woC	Time Changer	✓
DyPRAG	Time Changer	✓

Table 12: Case study about contextual and parametric knowledge conflicts in 2WQA (Bridge sub-task) where only DyPRAG-woC and DyPRAG answer wrongly (14.67%). The backbone model is the LLaMA3.2-1B. :deficiency in parametric knowledge, : knowledge conflicts, : successful knowledge manipulation

Vanilla	RAG	DyPRAG-woC	Ratio(%)
✓	✓	✓	4.67
✗	✗	✗	34.67
✓	✗	✓	**6.33**
✓	✓	✗	5.33
✗	✓	✓	8.33
✗	✗	✓	**26.33**
✗	✓	✗	7.67
✓	✗	✗	6.33

Table 13: Right/Wrong answer combinations of Vanilla, RAG, DyPRAG-woC 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(%)
✓	✓	✓	5.33
✗	✗	✗	35.00
✓	✗	✓	**6.33**
✓	✓	✗	4.67
✗	✓	✓	8.00
✗	✗	✓	**26.00**
✗	✓	✗	8.00
✓	✗	✗	6.67

Table 14: 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).

I Visualization of Parameter Translator Workflow.

To clearly illustrate the workflow of the parameter translator $\mathcal{F}'_\phi$, we use the up-proj module in the FFN as an example, as shown in fig. 12. This visualization demonstrates the transformation of document embeddings into dynamic LoRAs, consistent with eq. (4).

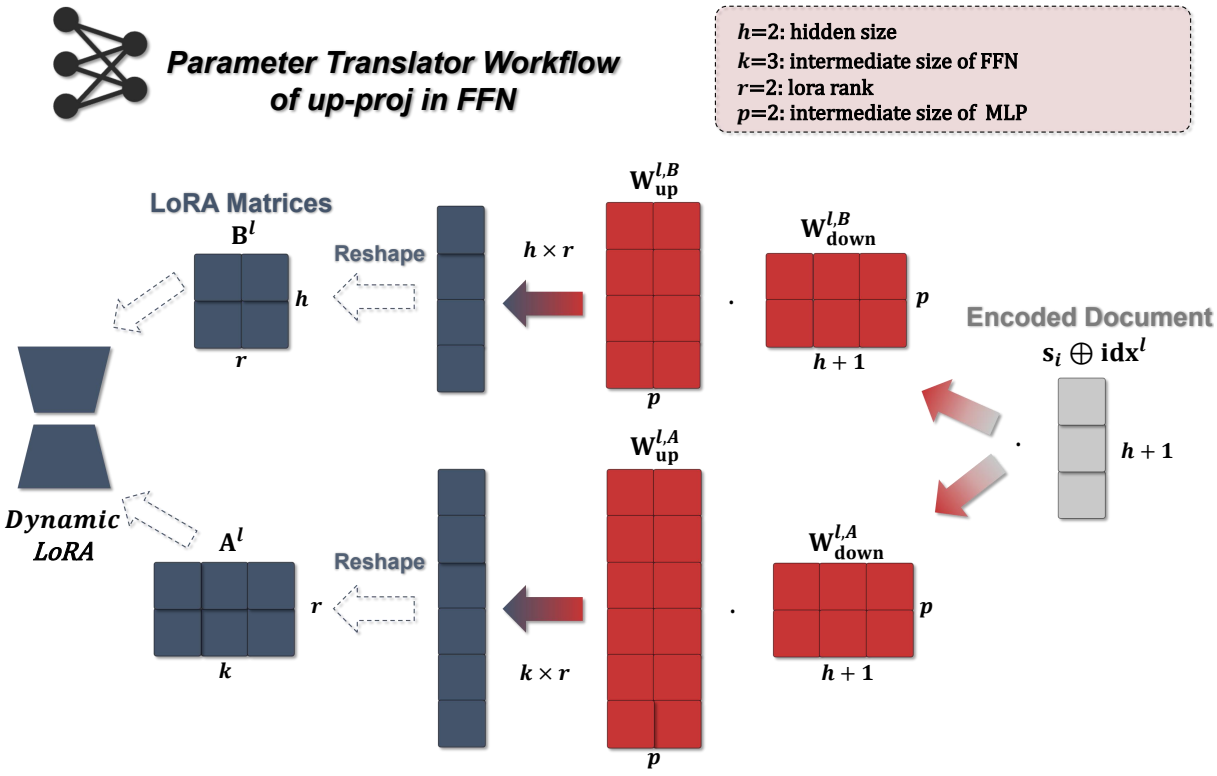

Figure 12: Visualization of the parameter translator workflow of up-proj in FFN. The overall process remains consistent with eq. (4).

J Prompt for Main Experiments Evaluation

In the main experiments, we used the following prompt to assess the performance of DyPRAG and other baseline models in fig. 13 and fig. 14:

K Prompt for Knowledge Internalization Evaluation

In the knowledge internalization experiments, we used the following prompt to assess the internalization ability of RAG generation from DyPRAG and RAG method evaluated by GPT-4o in fig. 15:

Prompt Format of No-CoT

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

Passage 1: {passages[0]}
Passage 2: {passages[1]}
Passage 3: {passages[2]}

Question: {question}
The answer is {answer}

Figure 13: Prompt format of No-CoT in our expriments.

Prompt Format of CoT

You should reference the knowledge provided below and combine it with your own knowledge to answer the question. Please follow the format of the example I provided above. Here are some examples about how to answer the questions.
Question: $\text{fewshot}_q[0]$
Answer: $\text{fewshot}_a[0]$
Question: $\text{fewshot}_q[1]$
Answer: $\text{fewshot}_a[1]$
Question: $\text{fewshot}_q[2]$
Answer: $\text{fewshot}_a[2]$
...

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

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

Figure 14: Prompt format of CoT in our expriments.

Prompt Format of Evaluate RAGTruth

Compare DyPRAG and RAG answers to assess which better internalizes knowledge—integrating its own knowledge with the given context for a natural, informed response.

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

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

Question: {question}
Context: {passages}
DyPRAG Answer: {dyprag_answer}
RAG Answer: {rag_answer}

Respond in the following format:
{{
"win model": "DyPRAG or RAG or Tie",
"reason": "Provide a concise explanation of why the selected answer demonstrates better knowledge integration, referencing the question, context, and specific details from both answers. If one answer has clear advantages in integration, explain them; if there are errors or weaknesses, specify them."
}}

Figure 15: Prompt format of evaluate RAGTruth using GPT-4o. We compare answer between standard RAG and DyPRAG.