FusePRAG: A Fuse-then-Project Paradigm for Parametric Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) enhances Large Language Models (LLMs) but suffers from high inference latency and noise sensitivity. While Parametric RAG (PRAG) addresses these issues by encoding retrieved knowledge into adapter parameters, existing methods adhere to a "Project-then-Fuse" paradigm, processing documents in isolation. This strategy incurs linear costs and severs cross-document dependencies essential for reasoning. To address this, we propose FusePRAG, a novel framework shifting the paradigm to "Fuse-then-Project". By employing a query-guided fusion module to synthesize semantic logic before projection, FusePRAG generates adapter parameters in a single pass, achieving constant complexity during the projection phase. This architecture constructs a holistic representation of reasoning chains, naturally complementing the fine-grained details of explicit context. Experiments on four benchmarks demonstrate that FusePRAG exhibits robust generalization and yields substantial synergy with standard RAG, achieving superior performance in the hybrid setting.

1 Introduction

Large Language Models (LLMs) have demonstrated exceptional generalization capabilities across diverse natural language processing tasks (Brown et al., 2020; Chowdhery et al., 2023; Fang et al., 2024). However, despite their extensive pre-training, LLMs often struggle with knowledge-intensive reasoning due to the static nature of their parametric memory and the tendency to hallucinate when recalling obscure facts (Mallen et al., 2023). To mitigate these limitations, **Retrieval-Augmented Generation (RAG)** (Lewis et al., 2020) has emerged as a standard paradigm (Figure 1(a)). By retrieving relevant documents and concatenating them into the input context (In-Context Injection), RAG provides the model with the necessary evidence. However, this approach faces significant scalability challenges: extending context length linearly increases inference latency and memory consumption (Liu et al., 2023). Moreover, injecting noisy or conflicting documents into the context can confuse the model, leading to degradation in reasoning performance (Shi et al., 2023).

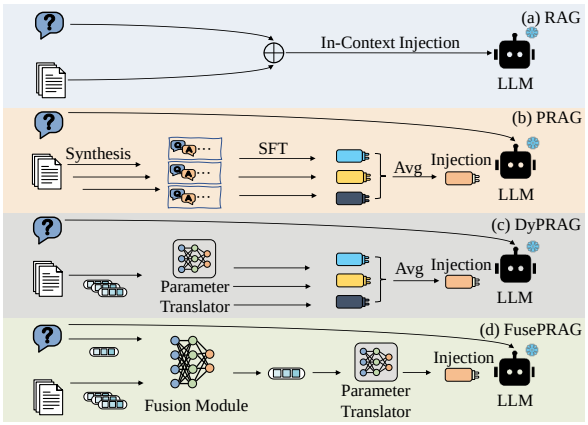

Figure 1: Comparison of RAG Paradigms. (a) **Standard RAG** performs in-context injection. (b) **PRAG** encodes documents into parameters offline. (c) **DyPRAG** dynamically projects retrieved documents into parameters independently. (d) **FusePRAG** (Ours) fuses cross-document semantics *before* projection.

Recently, **Parametric RAG (PRAG)** (Su et al., 2025) offered a promising alternative by shifting the locus of knowledge integration from the input context to adapter parameters (e.g., via LoRA (Hu et al., 2022)) offline (Figure 1(b)). To further improve adaptability, **DyPRAG** (Tan et al., 2025) introduced a hypernetwork framework that dynamically projects retrieved documents into adapter parameters online (Figure 1(c)). However, DyPRAG adheres to a **"Project-then-Fuse"** paradigm, where each document is independently projected into adapter parameters and aggregated subsequently. We identify two structural limitations in this design. First, projecting documents in isolation fundamen-

tally precludes modeling cross-document dependencies at the semantic level, thereby breaking the compositional evidence chains required for multi-hop reasoning. Second, this workflow incurs a linear computational burden, as the projection step must be repeated for each document prior to fusion.

To address this, we propose **FusePRAG**, a novel framework that shifts the paradigm from "Project-then-Fuse" to "Fuse-then-Project" (Figure 1(d)). Instead of processing documents in isolation, FusePRAG employs a lightweight query-guided fusion module to synthesize cross-document semantics *before* projection. This strategy enables the model to (1) distill critical evidence from noise via attention mechanisms and (2) construct a holistic representation of reasoning chains in the semantic space. Crucially, the projection maps this fused representation into adapter parameters in a single pass, ensuring constant projection complexity ($O(1)$) while preserving the logical dependencies required for complex queries. Moreover, this design is naturally complementary to standard RAG: the adapter parameters effectively augment the fine-grained details of explicit context, consistently yielding superior performance in the hybrid setting. Our contributions are summarized as follows:

- We propose FusePRAG, a "Fuse-then-Project" framework that employs a query-guided fusion module to synthesize cross-document semantics *before* projection, enabling robust reasoning with constant projection complexity ($O(1)$).
- We introduce a reasoning-aware training strategy that aligns the fusion module with synthesized summaries, ensuring generated parameters capture holistic reasoning chains.
- Extensive experiments show that FusePRAG effectively synthesizes distributed evidence, exhibits robust generalization, and yields substantial synergistic gains when augmenting standard RAG.

2 Related Work

2.1 Retrieval-Augmented Generation

Standard RAG conditions generation on retrieved documents (Guu et al., 2020; Lewis et al., 2020; Jiang et al., 2022). While recent works have optimized retrieval timing (Jiang et al., 2023; Su et al., 2024), refined reasoning prompts (Trivedi et al., 2023), or integrated structured knowledge (Peng et al., 2024), they predominantly rely on in-context injection, where documents are directly appended to the input. This paradigm faces inherent scalability hurdles: extending context length significantly increases inference latency (Dao et al., 2022; Liu et al., 2023) and degrades information access due to the "Lost-in-the-Middle" phenomenon (Liu et al., 2024). These structural limitations motivate Parametric RAG, which shifts knowledge integration from the input context to model parameters.

2.2 Parametric Knowledge Injection

Parametric RAG seeks to internalize external knowledge into model weights. PRAG (Su et al., 2025) introduced the concept of encoding documents into independent LoRAs offline, but it incurs prohibitive storage costs for large corpora. DyPRAG (Tan et al., 2025) mitigates this by dynamically generating parameters at inference time via a hypernetwork. However, it inherently adopts a "Project-then-Fuse" paradigm, treating documents as isolated instances aggregated via linear combinations. This independence assumption severs cross-document semantic dependencies, failing to capture the reasoning chains required for multi-hop queries. In contrast, FusePRAG shifts to "Fuse-then-Project", synthesizing document interactions *before* parameterization to encode holistic logic.

2.3 Context Compression and Fusion

Context compression techniques, which condense long documents into compact vectors (Ge et al., 2023), soft prompts (Chevalier et al., 2023; Mu et al., 2023), or selected token subsets (Jiang et al., 2024; Li et al., 2023), have proven effective for reducing computational overhead. While sharing the efficiency goal, FusePRAG differentiates itself by shifting the integration target to parametric weights (LoRA) for deeper model integration. Furthermore, we redefine the compression objective from generic reconstruction to logical fusion. Instead of optimizing for surface-level text restoration, we supervise the fusion module with reasoning-aware summaries, explicitly forcing the model to synthesize scattered evidence into cohesive reasoning chains.

3 Methodology

In this section, we present our framework (illustrated in Figure 2), designed to overcome the limitations of independent document projection in parametric retrieval. We first define the problem scope

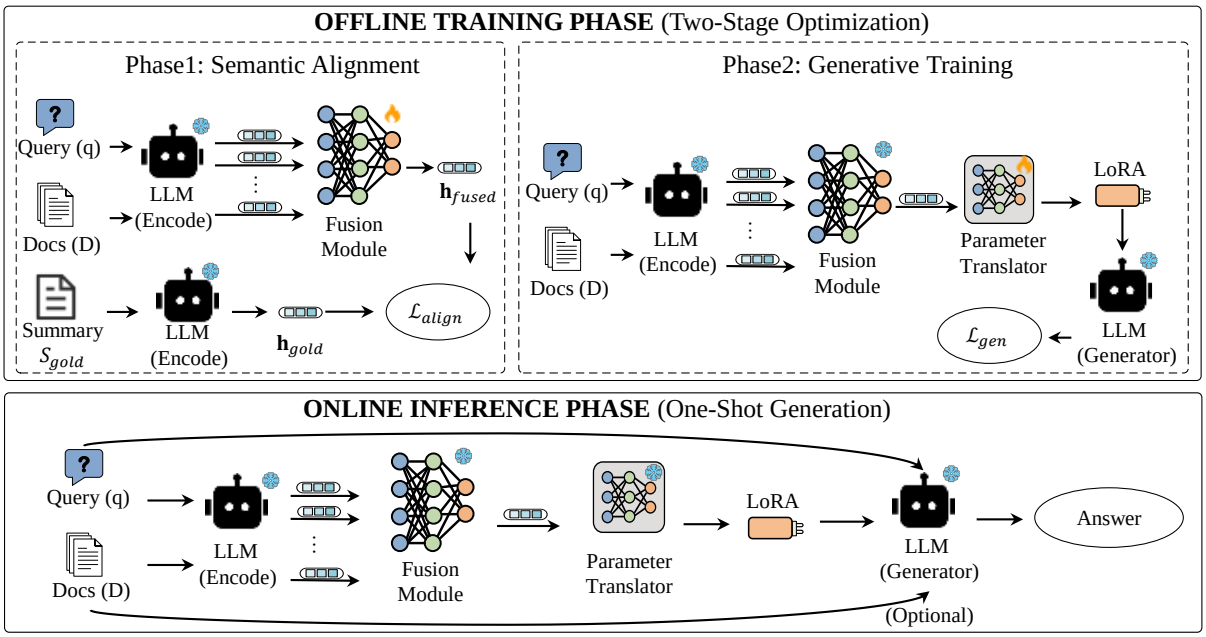

Figure 2: Overview of the FusePRAG framework. **Top (Offline Phase):** We employ a two-stage optimization strategy. Phase 1 aligns the fusion module ($\mathcal{M}_\psi$) with reasoning summaries to capture logical dependencies. Phase 2 freezes the fusion module and optimizes the translator ($\mathcal{F}_\phi$) to map the fused representation into LoRA parameters ($\Delta\Theta$). **Bottom (Online Inference):** The model performs *one-shot generation*, synthesizing all retrieved documents into a single parameter update via a single forward pass.

and revisit the standard Parametric RAG formulation. Subsequently, we introduce our core contribution: the **Cross-Document Semantic Fusion** mechanism. This module establishes a "Fuse-then-Project" paradigm, synthesizing multi-document contexts into a unified semantic representation before parameter generation. Finally, we detail our two-phase training strategy, which aligns this fused representation with reasoning-aware summaries to ensure robust logical deduction.

3.1 Preliminaries and Problem Formulation

Parametric RAG. Given a user query q and a large-scale external knowledge corpus $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$, a retrieval module $\mathcal{R}$ retrieves a subset of top-k relevant documents, denoted as $\mathcal{D} = \{d_1, d_2, \ldots, d_k\} \subset \mathcal{C}$, where $k \ll N$. Diverging from standard RAG, which explicitly concatenates $\mathcal{D}$ into the textual input context of the LLM, Parametric RAG (PRAG) seeks to integrate external knowledge directly into the model's parametric space. Let Θ represent the pre-trained parameters of the LLM. The objective is to learn an optimal context-dependent parameter update, denoted as $\Delta\Theta^*$, that maximizes the log-likelihood of generating the ground-truth response y:

$$\Delta\Theta^* = \arg\max_{\Delta\Theta} \log P(y \mid q; \Theta + \Delta\Theta). \quad (1)$$

In practice, $\Delta\Theta$ is typically implemented via parameter-efficient fine-tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA), to ensure computational feasibility (Hu et al., 2022).

The DyPRAG Baseline. To circumvent the prohibitive cost of online training for every query, DyPRAG (Tan et al., 2025) introduces a lightweight *Parameter Translator* (or Hypernetwork), denoted as $\mathcal{F}_\phi$. This module adopts an *Independent Projection* strategy, learning a mapping function that converts the semantic embedding $\mathbf{e}_{d_i}$ of a single document d_i directly into the parameter update space. Formally, DyPRAG generates a specific update $\Delta\Theta_i = \mathcal{F}_\phi(\mathbf{e}_{d_i})$. When handling multiple retrieved documents ($k > 1$), it follows a "Project-then-Fuse" paradigm, employing a linear aggregation in the parameter space:

$$\Delta\Theta_{\text{DyPRAG}} = \sum_{i=1}^{k} \lambda_i \cdot \mathcal{F}_\phi(\mathbf{e}_{d_i}), \quad (2)$$

where λ_i represents the aggregation weight (e.g., $\lambda_i = 1/k$ for simple averaging).

The Independence Assumption Gap. While the formulation in Eq. (2) ensures computational efficiency, we argue that it relies on a strong *Independence Assumption*—that each document con-

tributes to the target knowledge representation independently. We identify two critical limitations inherent in this paradigm: (1) **Neglected Cross-Document Interactions.** Complex queries often require multi-hop reasoning, where the answer is derived by synthesizing information across documents (e.g., $d_i \rightarrow$ intermediate fact $\rightarrow d_j$). The independent projection $\mathcal{F}_\phi(\mathbf{e}_{d_i})$ isolates the semantic encoding of each document, preventing the model from capturing these non-linear logical dependencies. (2) **Noise Sensitivity.** The linear aggregation operates indiscriminately on the retrieved set $\mathcal{D}$. When $\mathcal{D}$ contains irrelevant or noisy passages (distractors), the erroneous parameter updates $\Delta\Theta_{\text{noise}}$ are inevitably mixed into the final $\Delta\Theta$, diluting the parametric knowledge derived from gold passages.

The Fuse-then-Project Formulation. To address these limitations, we propose to shift the integration process from the *parameter space* to the *semantic space*. We reformulate the parameter generation process under a "Fuse-then-Project" paradigm. Specifically, we define a Cross-Document Semantic fusion module, $\mathcal{M}_\psi$, which accepts the query embedding $\mathbf{e}_q$ and the set of document embeddings $\{\mathbf{e}_{d_1}, \ldots, \mathbf{e}_{d_k}\}$ as joint inputs to explicitly model document interactions. The final parameter update is generated from this fused semantic representation:

$$\mathbf{h}_{\text{fused}} = \mathcal{M}_\psi(\mathbf{e}_q, \{\mathbf{e}_{d_1}, \ldots, \mathbf{e}_{d_k}\}), \tag{3}$$

$$\Delta\Theta_{\text{Ours}} = \mathcal{F}_\phi(\mathbf{h}_{\text{fused}}). \tag{4}$$

By adhering to Eq. (4), we enable the translator $\mathcal{F}_\phi$ to condition on a unified, noise-filtered semantic vector $\mathbf{h}_{\text{fused}}$, theoretically encapsulating the complete reasoning chain required for accurate parameter generation.

3.2 Cross-Document Semantic Fusion

The core component of our architecture is the Cross-Document Semantic fusion module ($\mathcal{M}_\psi$), designed to synthesize information from multiple retrieved documents into a unified, noise-filtered semantic representation.

Input Construction. Instead of concatenating raw tokens which incurs excessive context length, we perform fusion directly in the dense embedding space. Following DyPRAG (Tan et al., 2025), we employ the frozen backbone LLM as the feature extractor. This ensures that the extracted features are aligned with the target parametric space. Let $\mathcal{E}_{\text{LLM}}(\cdot)$ denote the encoding function of the backbone LLM. We first encode the query q and each retrieved document $d_i \in \mathcal{D}$ into dense vectors:

$$\mathbf{e}_q = \mathcal{E}_{\text{LLM}}(q), \quad \mathbf{e}_{d_i} = \mathcal{E}_{\text{LLM}}(d_i), \tag{5}$$

where $\mathbf{e}_q, \mathbf{e}_{d_i} \in \mathbb{R}^d$. We then construct the input sequence $\mathbf{H}^0$ for the fusion network by concatenating these embeddings. Crucially, we prepend the query embedding to the sequence to serve as the global information anchor:

$$\mathbf{H}^0 = [\mathbf{e}_q, \mathbf{e}_{d_1}, \mathbf{e}_{d_2}, \ldots, \mathbf{e}_{d_k}] \in \mathbb{R}^{(k+1)\times d}, \tag{6}$$

where $[\cdot]$ denotes concatenation along the sequence dimension. During training, $\mathcal{D}$ comprises a mixture of *gold* and *distractor* passages, forcing the model to discern relevant features within the sequence.

Lightweight Transformer Fusion. We employ a lightweight Transformer encoder to process the sequence $\mathbf{H}^0$. Its self-attention mechanism serves two pivotal roles: (1) **Denoising via Query-Document Attention**: By leveraging the query anchor $\mathbf{e}_q$ as a reference, the model dynamically reweights document embeddings $\mathbf{e}_{d_i}$, suppressing distractors. (2) **Reasoning via Document-Document Attention**: It facilitates interactions between documents (e.g., $d_i \leftrightarrow d_j$), capturing multi-hop logical dependencies in the semantic space.

Fused Representation. After L layers of Transformer processing, we extract the updated vector at the first position (corresponding to the query anchor) as the final fused representation:

$$\mathbf{h}_{\text{fused}} = \text{Transformer}(\mathbf{H}^0)_0 \in \mathbb{R}^d. \tag{7}$$

This vector $\mathbf{h}_{\text{fused}}$ serves as a compressed semantic summary of the query-aware knowledge, ready for the subsequent parameter projection.

3.3 Parameter Projection

We employ an MLP-based Parameter translator $\mathcal{F}_\phi$ to translate the fused semantic knowledge into context-dependent updates. Diverging from the independent projection in DyPRAG (Tan et al., 2025), our translator $\mathcal{F}_\phi$ conditions directly on the unified vector $\mathbf{h}_{\text{fused}}$. For a target weight $W \in \mathbb{R}^{m\times n}$, it generates low-rank matrices $A \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{m\times r}$ ($r \ll \min(m, n)$):

$$\{A, B\} = \mathcal{F}_\phi(\mathbf{h}_{\text{fused}}), \quad \Delta\Theta = BA. \tag{8}$$

The generated $\Delta\Theta$ is then added to the frozen LLM for final generation.

3.4 Two-Phase Training Strategy

Direct end-to-end optimization yields unconstrained latent representations, often failing to distill precise reasoning chains from noise. To address this, we propose a two-phase training strategy leveraging **reasoning-aware summaries** as an intermediate supervision signal.

Phase 1: Semantic Alignment. The goal of this phase is to force the fusion module to "distill" the reasoning chain from mixed documents. Instead of relying solely on the final short answer, we introduce a dense supervision signal: the **Evidence Summary**. We assume access to a ground-truth summary S_{gold} derived from the gold passages (construction details are provided in Section 4.1). To anchor the fused representation within the native semantic space of the backbone LLM, we employ the same frozen backbone LLM $\mathcal{E}_{\text{LLM}}$ (defined in Section 3.2) to encode the summary. Specifically, we obtain the target embedding $\mathbf{h}_{\text{gold}} = \mathcal{E}_{\text{LLM}}(S_{\text{gold}})$. We optimize the fusion parameters ψ by minimizing a hybrid objective comprising Mean Squared Error (MSE) and Cosine Embedding Loss:

$$\begin{aligned}\mathcal{L}_{\text{align}} = &\|\mathbf{h}_{\text{fused}} - \mathbf{h}_{\text{gold}}\|_2^2 \\ &+ \lambda \cdot (1 - \cos(\mathbf{h}_{\text{fused}}, \mathbf{h}_{\text{gold}})),\end{aligned} \tag{9}$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity and λ is a balancing hyperparameter (set to 0.5 in our experiments). This dual constraint aligns $\mathbf{h}_{\text{fused}}$ with the summary in both magnitude and semantic direction, effectively capturing the reasoning essence independent of the downstream generation task.

Phase 2: End-to-End Generative Training. In the second phase, we integrate the translator with the LLM to perform end-to-end training. The objective is to translate the aligned semantic knowledge into context-dependent parameter updates. To facilitate convergence, we initialize the translator using the pre-trained weights from DyPRAG (Tan et al., 2025). We freeze the LLM backbone parameters Θ to preserve its general capabilities. To maintain the semantic alignment learned in Phase 1, we also freeze the fusion module parameters ψ, exclusively optimizing the translator parameters ϕ. The model is trained to maximize the probability of the ground-truth answer y:

$$\mathcal{L}_{\text{gen}} = -\sum_{t=1}^{|y|} \log P_{\text{LLM}}(y_t \mid q, y_{<t}; \Theta + \Delta\Theta_{\text{Ours}}). \tag{10}$$

During this process, the translator learns to map the dense semantic vector $\mathbf{h}_{\text{fused}}$, which has been semantically aligned in Phase 1, into the specific low-rank weight perturbations $\Delta\Theta_{\text{Ours}}$ required to generate the precise textual response.

3.5 Inference Process

Given a query q and the retrieved set $\mathcal{D}$, we input them jointly into the fusion module to obtain $\mathbf{h}_{\text{fused}}$. The translator then generates a single set of LoRA parameters $\Delta\Theta$. Finally, the LLM executes generation using the updated weights $\Theta + \Delta\Theta$.

4 Experiments

4.1 Experimental Setup

Training Data Construction. We constructed a compact, balanced dataset derived from **2WikiMultihopQA (2WQA)** (Ho et al., 2020) and **HotpotQA (HQA)** (Yang et al., 2018). Specifically, we sampled 200 instances per sub-category, totaling **1,200 samples** (randomly split into 960 training, 240 validation). To simulate realistic retrieval noise and provide intermediate supervision, we employed **GPT-3.5-Turbo** (Ouyang et al., 2022) to synthesize hard negative distractors and reasoning-aware summaries (prompts in Appendix A). We resolved ambiguity in original synonym lists by designating the single answer most semantically consistent with gold passages as the **training label**. Crucially, this curated dataset is strictly disjoint from the official evaluation sets.

Evaluation Protocols and Metrics. Adhering to the protocols of PRAG (Su et al., 2025) and DyPRAG (Tan et al., 2025), we evaluate on four datasets: **2WQA**, **HQA**, **PopQA (PQA)** (Mallen et al., 2023), and **ComplexWebQuestions (CWQ)** (Talmor and Berant, 2018). We report performance on the same evaluation subsets (same instances) defined in these baselines to ensure a fair comparison. Crucially, since our model is trained exclusively on samples from 2WQA and HQA, we classify these as **In-Distribution (ID)** benchmarks. Conversely, CWQ and PQA serve as **Out-of-Distribution (OOD)** tests to assess generalization capabilities. We adopt the **F1 Score** as the primary metric.

Baselines. Following the setup in (Su et al., 2025; Tan et al., 2025), we compare our method with the following baselines: (1) **Vanilla LLM**: Generates

Base LLM	Method	2WQA				HQA		PQA	CWQ	Avg
		Compare	Bridge	Inference	Compose	Bridge	Compare			
LLaMA3.2-1B	**Vanilla**	42.89	24.17	16.91	7.87	13.25	40.26	2.26	34.94	22.82
	Standard RAG	41.23	26.78	22.51	10.21	21.38	42.46	17.65	37.39	27.45
	PRAG	50.20	24.34	19.11	8.24	13.65	40.90	23.58	35.86	26.99
	PRAG-Combine	40.50	31.30	22.85	9.77	22.56	41.55	32.59	39.63	30.09
	DyPRAG	48.75	49.09	17.42	7.26	13.75	41.06	8.91	35.57	27.73
	DyPRAG-Combine	50.99	42.31	23.21	11.73	22.75	42.49	30.52	38.58	32.82
	FusePRAG (ours)	**55.02**	47.35	12.38	13.09	9.17	43.03	12.67	38.19	28.86
	FusePRAG-Combine (ours)	45.82	**53.46**	**25.07**	**14.22**	**23.12**	**44.18**	**38.65**	**43.97**	**36.06**
Qwen2.5-1.5B	**Vanilla**	45.74	39.06	17.04	7.27	12.18	39.46	2.87	26.47	23.76
	Standard RAG	38.75	38.84	11.87	5.68	16.19	37.13	9.97	28.23	23.33
	PRAG	44.96	43.96	**19.29**	**11.14**	13.27	40.42	21.55	30.82	28.18
	PRAG-Combine	40.50	44.00	16.30	8.17	18.86	36.49	23.43	32.13	27.49
	DyPRAG	43.03	47.20	17.04	8.55	13.72	41.39	6.64	31.94	26.16
	DyPRAG-Combine	35.83	44.89	14.81	8.64	21.56	41.25	22.69	33.57	27.91
	FusePRAG (ours)	**50.99**	**52.20**	14.11	8.04	10.45	**45.83**	6.81	**35.78**	28.03
	FusePRAG-Combine (ours)	49.07	48.53	15.84	10.49	**21.96**	43.64	**29.32**	35.74	**31.82**
LLaMA3-8B	**Vanilla**	54.90	55.20	24.59	14.43	19.00	45.63	7.96	42.44	33.02
	Standard RAG	58.43	47.77	19.20	11.07	19.68	42.10	16.13	35.45	31.23
	PRAG	57.78	**58.93**	27.61	19.17	33.68	65.88	26.13	43.54	41.59
	PRAG-Combine	60.13	56.69	32.71	20.91	**39.41**	**68.22**	26.23	36.41	42.59
	DyPRAG	58.15	54.86	24.33	15.92	18.73	57.29	12.44	41.41	35.39
	DyPRAG-Combine	62.58	51.54	**36.16**	26.08	32.44	58.36	33.35	40.24	42.59
	FusePRAG (ours)	**63.80**	55.00	17.36	13.58	16.62	58.26	17.14	**48.78**	36.32
	FusePRAG-Combine (ours)	60.40	53.02	35.80	**27.47**	35.57	60.21	**35.17**	42.11	**43.72**

Table 1: Performance comparison of FusePRAG against Standard RAG and parametric baselines on four QA benchmarks across three LLM backbones. All metrics are reported as F1 scores (%). **Combine** denotes the hybrid setting integrating the parametric adapter with explicit retrieved context. **Avg** denotes the average performance across all sub-tasks. The best and second-best results are **bolded** and underlined, respectively.

responses using only the frozen backbone parameters without external retrieval. (2) **Standard RAG**: Concatenates retrieved documents into the input context and prompts the model to answer based on the provided evidence. (3) **PRAG**: Encodes documents into independent LoRA modules via offline training, thereby shifting knowledge injection from the context to the parameter space. (4) **DyPRAG**: Projects document embeddings independently into parameters at test-time via a hypernetwork, avoiding the storage overhead of individual adapters.

Implementation Details. We conduct experiments on three diverse LLMs: **Llama-3.2-1B-Instruct**, **Qwen-2.5-1.5B-Instruct**, and **Llama-3-8B-Instruct**. The fusion module is implemented as a lightweight 2-layer Transformer Encoder, with the hidden dimension d set to match the backbone LLM. We employ the AdamW optimizer for the two-phase optimization. In **Phase 1**, we train the fusion module for 3 epochs (lr=1e-5) to align semantic representations. In **Phase 2**, we freeze both the backbone and the fusion module, exclusively optimizing the translator for 3 epochs (lr=1e-6). Following Tan et al. (2025), we uniformly use BM25 to retrieve the top-3 documents for all methods, ensuring a fair comparison. Further details are provided in Appendix B.

4.2 Main Results

Table 1 presents the performance across three LLM backbones. The "Avg" column underscores our robust performance: FusePRAG consistently surpasses DyPRAG, validating our architectural design, while FusePRAG-Combine achieves the highest overall F1 across all backbones (e.g., +3.24% on Llama-3.2-1B). We analyze the results regarding parametric knowledge fusion (ID) and generalization robustness (OOD).

Performance on ID Benchmarks. In the parametric-only regime, FusePRAG generally surpasses DyPRAG on reasoning-heavy tasks (e.g., 2WQA), validating our fusion mechanism, though it may occasionally lag in precise lexical extraction. Crucially, the hybrid setting yields the most

Method	F1	Δ
Ours (Full Model)	**43.97**	-
w/o Fusion Module (Mean Pool)	40.86	-3.11
w/o Phase 1 (End-to-End)	36.63	-7.34

Table 2: **Ablation studies on the CWQ dataset (Combine setting).** We evaluate the fusion module and the Two-Phase training strategy using Llama-3.2-1B. Both components are essential for optimal performance.

Noise Level (N_{noise})	DyPRAG (Combine)	FusePRAG (Combine)	Gain (Δ F1)
0	38.58	**43.97**	**+5.39**
1	38.43	**43.69**	**+5.26**
3	37.09	**43.09**	**+6.00**
5	37.67	**42.47**	**+4.80**
8	36.83	**41.79**	**+4.96**

Table 3: **Robustness to Retrieval Noise.** F1 scores on CWQ (Llama-3.2-1B) across varying numbers of noise documents (N_{noise}). Our method maintains a consistent performance margin over the baseline even under high-noise conditions.

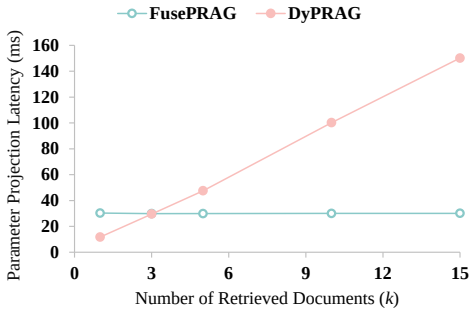

Figure 3: **Parameter Projection Latency.** Wall-clock time to transform retrieved context embeddings into LoRA parameters. Crucially, measurements for FusePRAG explicitly include the query encoding latency. Despite this overhead, our approach maintains constant projection complexity ($O(1)$), demonstrating superior scalability over the linear ($O(k)$) DyPRAG baseline as retrieval depth increases.

robust gains. FusePRAG-Combine demonstrates superior synergy, achieving substantial improvements on both ID datasets (2WQA and HQA) across all backbones. This confirms our hypothesis: the fused adapter provides a high-level reasoning blueprint, while explicit context supplies necessary fine-grained details.

Generalization on OOD Benchmarks. On the unseen datasets (PQA and CWQ), our method exhibits superior OOD generalization. While parametric baselines often struggle with domain shifts due to overfitting specific factual distributions, our approach adapts dynamically to new contexts via the retrieval-augmented fusion mechanism. Notably, FusePRAG substantially surpasses Vanilla LLMs on CWQ across all backbones (e.g., **+6.34%** on Llama-3-8B). This indicates that our two-phase training learns a task-agnostic *reasoning pattern*, defined as the capacity to synthesize conflicting or disjoint retrieved information, rather than merely memorizing domain-specific facts.

4.3 Ablation Study

We validate our core components via ablation studies on CWQ using Llama-3.2-1B, comparing FusePRAG against two variants: (1) **w/o Fusion Module**: We replace the fusion module with mean pooling over document embeddings. This removes cross-document interaction, reducing the integration strategy to linear aggregation. (2) **w/o Phase 1 (End-to-End)**: We skip the alignment phase and jointly train the randomly initialized fusion module with the pre-trained translator in an end-to-end manner. We train this variant for the same total number of steps to ensure a fair comparison. Results are shown in Table 2.

Effect of Semantic Fusion. Replacing the fusion module with mean pooling results in a performance decline of **3.11 points**. This drop underscores the limitation of simple linear aggregation. Mean pooling treats all retrieved documents equally and inevitably dilutes the semantic signals from relevant evidence with noise. In contrast, our fusion module effectively weighs the input documents and acts as a parametric filter. It synthesizes critical information into a unified representation before parameter generation, which is essential for handling the complex reasoning required in CWQ.

Necessity of Two-Phase Alignment. Skipping the Phase 1 alignment strategy causes a substantial drop of **7.34 points**. This result indicates that directly optimizing the mapping from document embeddings to LoRA parameters is highly unstable due to the complexity of the parametric space. The summary-driven alignment in Phase 1 provides a critical inductive bias. It forces the translator to learn a robust semantic representation, serving as a warm-up to constrain the optimization landscape for the subsequent generation task.

Question: Who did Warren Moon play for whose fight song was "Skol, Vikings" ?		
Ground truth: Minnesota Vikings		
Retrieved Documents: **[Doc 1]** Pro Football Hall of Fame quarterback Warren Moon served as the starting signal-caller for the NFL franchise whose fight song is "Skol, Vikings" for three seasons from 1994 to 1996. **[Doc 2]** "Skol, Vikings" is the fight song of the Minnesota Vikings of the National Football League. It was introduced...		
Method	**Answer**	**Status**
DyPRAG-Combine	Warren Moon	✗
FusePRAG-Combine	Minnesota Vikings	✓

Table 4: Case study about multi-hop reasoning in CWQ. The backbone model is the LLaMA3.2-1B. We highlight key entities used for reasoning bridges in yellow. Green and red backgrounds indicate correct and incorrect answers, respectively.

4.4 Analysis and Discussion

In this subsection, we validate the efficacy and efficiency of FusePRAG through three lenses: robustness against retrieval noise, inference scalability, and qualitative reasoning capabilities.

Robustness to Retrieval Noise. We introduce distractor documents ($N_{noise} \in \{0..8\}$) alongside the top-3 retrieved contexts. Table 3 shows that FusePRAG maintains a stable margin of ∼5.0 points over DyPRAG. Crucially, even in the high-noise regime ($N_{noise} = 8$), our method achieves **41.79**, substantially surpassing DyPRAG's noise-free baseline (38.58). This confirms that while the baseline dilutes signals via indiscriminate aggregation, our fusion module leverages attention mechanisms to suppress noise prior to projection, preserving the integrity of parametric updates.

Efficiency and Scalability Analysis. To empirically validate the computational advantage of FusePRAG, we conduct a comparative latency analysis against DyPRAG on CWQ (Llama-3.2-1B). We strictly isolate the **Adapter Generation Latency**, defined as the wall-clock time to transform retrieved context embeddings into LoRA parameters, to decouple the parametric integration mechanism from universal text encoding costs (see Appendix C.1 for details).

Figure 3 reveals distinct scaling behaviors across retrieval depths $k \in \{1, \ldots, 15\}$. DyPRAG exhibits linear growth ($O(k)$) in latency as its independent projection paradigm requires executing the hypernetwork k times followed by aggregation. Conversely, FusePRAG achieves **constant projection complexity** ($O(1)$). By synthesizing multi-document contexts into a unified representation within the fusion module, we perform parameter projection only once, effectively decoupling the generation cost from the retrieval volume. Further inspection reveals a performance crossover at $k \approx 3$. At $k = 1$, FusePRAG incurs a marginal initialization cost (∼18ms overhead) attributable to query encoding and fusion. However, this fixed cost is rapidly amortized as k increases. In high-recall regimes (e.g., $k = 15$), FusePRAG achieves a substantial speedup of approximately **5**× (∼30ms vs. ∼150ms). This confirms that our architecture is highly scalable and particularly well-suited for evidence-intensive applications.

Case Study. Table 4 illustrates a scenario from CWQ where evidence is distributed across disjoint documents. Resolving the query requires bridging the fact that *Warren Moon* played for the team with the fight song "*Skol, Vikings*" (Doc 1) to the entity "*Minnesota Vikings*" (Doc 2). DyPRAG fails to traverse this chain, as its independent projection paradigm isolates these semantic links, severing the critical transitive relationship during parameter aggregation. Conversely, FusePRAG successfully performs this deduction. By enabling semantic interaction *prior* to projection, our model identifies the shared bridge entity and captures the critical cross-document dependency ($Moon \rightarrow Song \rightarrow Team$) within the generated adapter.

5 Conclusion

This work introduces **FusePRAG** to overcome the semantic fragmentation inherent in the "Project-then-Fuse" paradigm of Parametric RAG. By synthesizing cross-document evidence *prior* to parameterization, our "Fuse-then-Project" architecture preserves holistic reasoning chains while reducing adapter generation complexity from linear to constant time ($O(1)$). Experiments across diverse benchmarks confirm that FusePRAG effectively models multi-hop reasoning dependencies and exhibits superior robustness against retrieval noise. Furthermore, our findings highlight a potent synergy where dynamic parameters provide high-level reasoning blueprints to complement the fine-grained explicit context of standard RAG, enabling scalable and resilient knowledge integration.

Limitations

Despite the advantages of FusePRAG, we acknowledge four primary limitations. (1) **The Parametric Gap:** While FusePRAG outperforms independent projection baselines, our experiments (Table 1) reveal that the pure parametric setting still lags behind the hybrid "Combine" setting. This indicates that encoding discrete textual evidence into continuous LoRA parameters inherently involves **information loss**, particularly regarding fine-grained lexical details (e.g., exact entity strings). Realizing a "Pure Parametric RAG" that fully matches the precision of explicit context injection remains a grand challenge for future research. (2) **Retriever Dependency:** Our fusion mechanism is designed to filter noise, not to hallucinate missing facts. The model's reasoning capability remains strictly bounded by the recall quality of the upstream retrieval system ("garbage-in, garbage-out"). (3) **Task Scope:** Our experiments are confined to knowledge-intensive QA tasks. The efficacy of this "Fuse-then-Project" paradigm in long-form generation or multimodal contexts remains to be verified. (4) **Interpretability:** Compared to the explicit context injection of standard RAG, the parametric updates in FusePRAG are implicit. Tracing specific reasoning steps within the generated LoRA parameters poses a significant interpretability challenge.

References

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.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 3829–3846. Association for Computational Linguistics (ACL).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling laws for dense retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1339–1349.

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*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. 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.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677.

Zhengbao Jiang, Luyu Gao, Zhiruo Wang, Jun Araki, Haibo Ding, Jamie Callan, and Graham Neubig. 2022. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2336–2349.

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*.

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. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 6769–6781. Association for Computational Linguistics (ACL).

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 6342–6353.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

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

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: dynamic retrieval augmented generation based on the information needs of large language models. *arXiv preprint arXiv:2403.10081*.

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*.

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

Yuqiao Tan, Shizhu He, Huanxuan Liao, Jun Zhao, and Kang Liu. 2025. Dynamic parametric retrieval augmented generation for test-time knowledge enhancement. *arXiv preprint arXiv:2503.23895*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.

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*.

A Data Construction Details

To enhance reproducibility, we provide the specific prompts used to instruct GPT-3.5-Turbo for synthesizing the training data.

A.1 Reasoning-Aware Summary Generation

In Phase 1 (Semantic Alignment), we require a condensed summary that captures the multi-hop reasoning chain. The prompt template used to generate these summaries is presented in Figure 4.

A.2 Hard Negative Distractor Generation

To simulate realistic retrieval noise, we synthesize "hard negative" documents that share high lexical overlap with the query but contain no relevant answer. The prompt is shown in Figure 5.

Human Verification. To ensure the quality of the synthesized data, we conducted a manual review on a randomly sampled subset of 100 instances. For *evidence summaries*, we verified that they accurately retained the reasoning chain without introducing external hallucinations. For *hard negative distractors*, we confirmed that they shared high lexical overlap with the query while strictly excluding the ground-truth answer. Our review indicated a pass rate of over 95%, confirming the reliability of our prompt engineering pipeline.

B Implementation Details

B.1 Computing Infrastructure

All experiments are implemented using **PyTorch** and the **HuggingFace Transformers** library. Training and inference are conducted on NVIDIA A100 GPUs (80GB) and V100 (32GB) GPUs.

Prompt for generating reasoning-aware summaries

User:
You will receive a Question and its Golden Passages. Your task is to extract a compact summary containing only the information necessary to answer the Question.

Requirements:
1. The summary must be strictly based on the Golden Passages. Do NOT add any new facts or outside knowledge.
2. Keep only the essential information required to answer the Question.
3. Compress the content into 2--4 concise sentences.
4. Maintain objective and factual style.
5. Output format must be EXACTLY:
Summary: <your concise extracted paragraph>

Question: {question}
Golden Passages: {golden_passages}
Answer: {answer}

Figure 4: The prompt template used for generating reasoning-aware summaries (S_{gold}).

Prompt for generating hard negative distractors

User:
You are a data construction assistant for retrieval-augmented QA.
[Task]
Given a question, its answer, and several gold relevant passages that are enough to answer the question, generate additional NOISE passages.
Noise passages must:
1. Look topically or entity-wise related to the question (for example, same domain, similar type of entity).
2. Not be useful to answer the question or derive the correct answer.
3. Do not repeat or closely paraphrase the input passages.
4. Not contain the answer text or obvious paraphrases of the answer.
5. Each passage should be about 80--200 words, in one plain paragraph (no lists).
[Input]
Question: {question}
Answer: {answer}
Gold relevant passages: {gold_passages}
[Output format]
Return ONLY a JSON array of strings, each string is one noise passage.

Figure 5: The prompt template used for generating synthetic hard negative distractors (noise passages).

B.2 Retrieval Implementation

For retrieval, motivated by recent findings that BM25 performs on par with state-of-the-art dense models while offering superior efficiency (Su et al., 2025; Tan et al., 2025), we adopt BM25 implemented via Elasticsearch. Regarding the retrieval corpus, we utilize the standard English Wikipedia dump pre-processed by DPR (Karpukhin et al., 2020). We specifically use the `psgs_w100.tsv` dump, consistent with Tan et al. (2025), serving as the unified external knowledge source for all comparative methods.

B.3 Hyperparameter Configuration

Document and Query Encoding. Following DyPRAG (Tan et al., 2025), we use the frozen backbone LLM as the encoder. For each query q and retrieved document d_i, we extract the last hidden state at the final token position before the vocabulary projection as the dense representation. This design ensures that the semantic embeddings are naturally aligned with the parametric space of the generator.

Fusion Module Architecture. The cross-document fusion module is implemented as a lightweight Transformer encoder with 2 layers and 8 attention heads. The hidden dimension matches that of the backbone LLM, and the feed-forward dimension is set to twice the hidden size. Given the concatenated sequence of the query embedding and document embeddings, the output representation corresponding to the query position is used as the fused semantic vector.

Parameter Translator and LoRA Configuration. We adopt the same parameter translator architecture as DyPRAG (Tan et al., 2025) to ensure a fair comparison. LoRA modules are integrated exclusively into the feed-forward network (FFN) matrices of the backbone LLM, while the query, key, and value (QKV) projections are excluded. The LoRA rank is set to $r = 2$ with a scaling factor $\alpha = 32$, and no dropout is applied. To ensure stable optimization, we initialize the parameter translator with pre-trained weights from DyPRAG, thereby facilitating convergence and bypassing the instability of random initialization. Subsequently, during the generation phase (Phase 2), the translator parameters are fine-tuned to adapt the mapping function to the fused cross-document semantic representations.

Context Truncation Strategy. For the hybrid "Combine" setting, we adopt a unified truncation protocol to ensure strict fairness. Following Tan et al. (2025), the maximum context length is set to **3000 tokens**. We prioritize preserving the query and truncate the concatenated retrieved documents from the tail to fit this limit. This constraint is applied uniformly across all context-augmented baselines (Standard RAG, DyPRAG-Combine, and FusePRAG-Combine), ensuring that performance gains are attributed to our fusion logic rather than variations in effective context window size.

C Detailed Evaluation Setup

C.1 Efficiency Analysis Protocol

In Section 4.4, we specifically isolate the *Adapter Generation Latency* to ensure a fair algorithmic comparison. The total inference latency T_{total} typically comprises document encoding (T_{enc}), parameter projection (T_{proj}), and generation (T_{gen}).

Although our retrieval setup utilizes BM25, which returns textual documents, the parametric integration modules (in both baselines and our method) operate on dense representations. This implies a theoretical online encoding cost $T_{\text{enc}} \propto k$. However, we exclude T_{enc} from our reported metric for two reasons:

1. **Universality**: Both DyPRAG and FusePRAG require mapping retrieved text to dense embeddings. This pre-processing cost is identical across methods and does not reflect the algorithmic efficiency of the parametric projection architecture itself.

2. **Cacheability**: In standard dense retrieval production environments (e.g., DPR), document embeddings are pre-computed and indexed offline, rendering the online encoding cost negligible ($T_{\text{enc}} \approx 0$).

Therefore, we focus exclusively on T_{proj}.

- **DyPRAG**: $T_{\text{proj}} \approx k \cdot T_{\text{network}}$ (Linear scaling).
- **FusePRAG**: $T_{\text{proj}} \approx T_{\text{query\_enc}} + T_{\text{fuse}} + 1 \cdot T_{\text{network}}$ (Constant projection complexity).

This isolation strictly evaluates the efficiency of the "Fuse-then-Project" paradigm against the "Project-then-Fuse" baseline.

D Supplement Experiment Results

Method	F1	Δ
Ours (Full Model)	**38.19**	-
w/o Fusion Module (Mean Pool)	36.57	-1.62
w/o Phase 1 (End-to-End)	34.77	-3.42

Table 5: **Ablation studies on the CWQ dataset (Parametric-Only setting).** Unlike Table 2, these results are evaluated **without explicit context**, demonstrating that our proposed components yield robust gains in the pure parametric regime.