FD-RAG: Federated Dual-System Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) has emerged as an effective paradigm for grounding large language models in external knowledge, yet most existing RAG systems assume centralized knowledge access and ample computation. These assumptions break down in edge environments, where knowledge is fragmented across devices, raw data cannot be shared, and repeated LLM calls are prohibitively expensive. We propose FD-RAG, a federated dual-system RAG framework that decouples lightweight memory access from on-demand LLM reasoning for decentralized deployment. Specifically, FD-RAG learns semantic-aware adaptive hypergraphs over local corpora and distills them into compact QA memories. At inference time, it answers well-covered queries via direct memory matching and invokes LLM-based reasoning only when necessary, while tracing retrieved memories to hypergraph-grounded evidence. To mitigate cross-device knowledge fragmentation, FD-RAG further aggregates anonymized memories across devices without exposing raw documents. Experiments on standard QA benchmarks show that FD-RAG improves accuracy by up to 7.8% while reducing latency by 8.4× compared with strong local and federated baselines. We also provide theoretical analysis establishing an $\mathcal{O}(1/\epsilon^2)$ convergence rate for the proposed hypergraph learning, supporting its tractable deployment in edge settings.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020b) has emerged as a standard paradigm for grounding large language models (LLMs) in external knowledge, achieving strong performance on knowledge-intensive tasks. However, its effectiveness implicitly relies on a centralized setting where both knowledge and computation are readily accessible (Oche et al., 2025). This assumption rarely holds in practice. In real-world domains such as healthcare, finance, and law, knowledge is inherently distributed across institutions and edge devices, data sharing is restricted by privacy and regulation (Xu et al., 2025), and inference must operate under stringent constraints on computation and latency. These challenges necessitate a new paradigm in which RAG operates directly on edge devices while respecting data locality.

Extending RAG to such environments introduces fundamental challenges. Existing approaches (Luo et al., 2025; Edge et al., 2024) rely heavily on LLMs throughout the pipeline, using them not only for reasoning but also for knowledge construction and query understanding. This design becomes fragile when deployed with small language models (SLMs), which are typical in edge settings: knowledge construction degrades (Fan et al., 2025), complex queries are handled less reliably, and repeated model invocations incur prohibitive latency (Liu et al., 2024). Moreover, most RAG frameworks assume access to a unified knowledge repository (Gao et al., 2023). In decentralized environments, knowledge is fragmented across devices, leaving each node with incomplete coverage, particularly for queries requiring evidence from multiple sources (Chakraborty et al., 2025). As a result, existing systems struggle to simultaneously achieve efficiency and completeness in edge scenarios.

We argue that these limitations stem from treating retrieval and reasoning as a monolithic process mediated by language models, without explicitly accounting for their distinct computational roles. Inspired by Dual-Process Theory (Kahneman, 2003), which distinguishes fast, memory-based responses (System 1) from slower, deliberative reasoning (System 2), we propose to decouple these two modes in RAG. This leads to **Federated Dual-System RAG (FD-RAG)**, a unified framework that separates lightweight memory access from selective reasoning, enabling efficient and expressive knowledge utilization under resource constraints.

At the knowledge construction level, FD-RAG

constructs a lightweight yet expressive knowledge structure via semantic-aware hypergraph learning, capturing higher-order relations without expensive extraction. This structure is further distilled into question–answer (QA) memories, serving as a compact interface for efficient access. At the inference level, FD-RAG introduces two complementary modules: a *Memorizer*, which directly resolves queries well covered by QA memory through efficient matching, and a *Cognizer*, which selectively invokes LLM-based reasoning over hypergraph-grounded evidence for more complex queries. To address knowledge fragmentation, FD-RAG further incorporates a federated memory aggregation mechanism, enabling multiple devices to collaboratively improve coverage without sharing raw data. Our main contributions are as follows:

- We propose FD-RAG, a dual-system RAG framework for edge environments that decouples memory-based retrieval and selective LLM reasoning, enabling collaborative knowledge utilization across distributed devices.
- We introduce a semantic-aware hypergraph learning approach for constructing lightweight yet expressive knowledge structures, and derive compact QA memories for efficient inference under resource constraints.
- Experiments on standard QA benchmarks show that FD-RAG improves accuracy by up to 7.8% while reducing latency by a factor of $8.4\times$. We further prove that the proposed hypergraph learning procedure achieves an $\mathcal{O}(1/\epsilon^2)$ convergence rate, providing theoretical guarantees for its efficiency and stability.

2 Related Work

Retrieval-Augmented Generation. Recent RAG research has moved beyond the standard retrieve-then-generate pipeline to better support complex question answering, primarily through iterative retrieval and structured knowledge modeling (Jin et al., 2024; Li et al., 2024b). Iterative retrieval methods improve evidence coverage by interleaving retrieval, generation, and query reformulation, but their repeated reliance on large language models often incurs substantial latency and computational cost (Liu et al., 2025; Asai et al., 2023). Another line of work organizes evidence with explicit structures, including trees (Sarthi et al., 2024), graphs (Gutiérrez et al., 2025; Li et al., 2024a), and more recently hypergraphs (Luo et al., 2025). Compared with trees and graphs, hypergraphs can capture higher-order semantic relations beyond pairwise dependencies, making them particularly appealing for complex reasoning. However, existing structured RAG methods typically rely on expensive knowledge construction, large context windows, or strong semantic reasoning capabilities, which limits their suitability for small models in resource-constrained edge settings. Although efficiency-oriented methods such as EfficientRAG reduce part of the computational burden (Zhuang et al., 2024), *how to preserve the benefits of structured knowledge while enabling efficient, low-latency inference on edge devices remains largely underexplored.*

Federated Retrieval-Augmented Generation. Federated RAG extends RAG (Lewis et al., 2020a) to settings where knowledge is inherently distributed across silos and cannot be centralized due to privacy or regulatory constraints. Existing work mainly explores two complementary directions. One line of research focuses on federated retrieval, enabling cross-silo access via routing or aggregation mechanisms (Wang et al., 2024; Guerraoui et al., 2025; Xu, 2024; Zhao, 2024), but still relies on centralized coordination, limiting system autonomy and flexibility. Another line integrates RAG with federated learning, jointly optimizing model parameters across clients (He et al., 2025; Liang et al., 2026; Fajardo et al., 2025), yet incurs substantial communication and computation overhead due to frequent parameter exchange. *Despite these advances, prior work largely overlooks collaboration at the knowledge level, hindering support for dynamic, unstructured, and semantically rich RAG scenarios in fully decentralized environments.*

3 Preliminaries

3.1 Hypergraph

A hypergraph is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ denotes the set of nodes, and $\mathcal{E} = \{e_1, \ldots, e_M\}$ denotes the set of hyperedges, each connecting a subset of nodes in $\mathcal{V}$.

The structure of the hypergraph is represented by an incidence matrix $H \in \mathbb{R}^{N\times M}$, where H_{nm} indicates the membership of node v_n in hyperedge e_m:

$$H_{nm} = \begin{cases} 1, & \text{if } v_n \in e_m, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

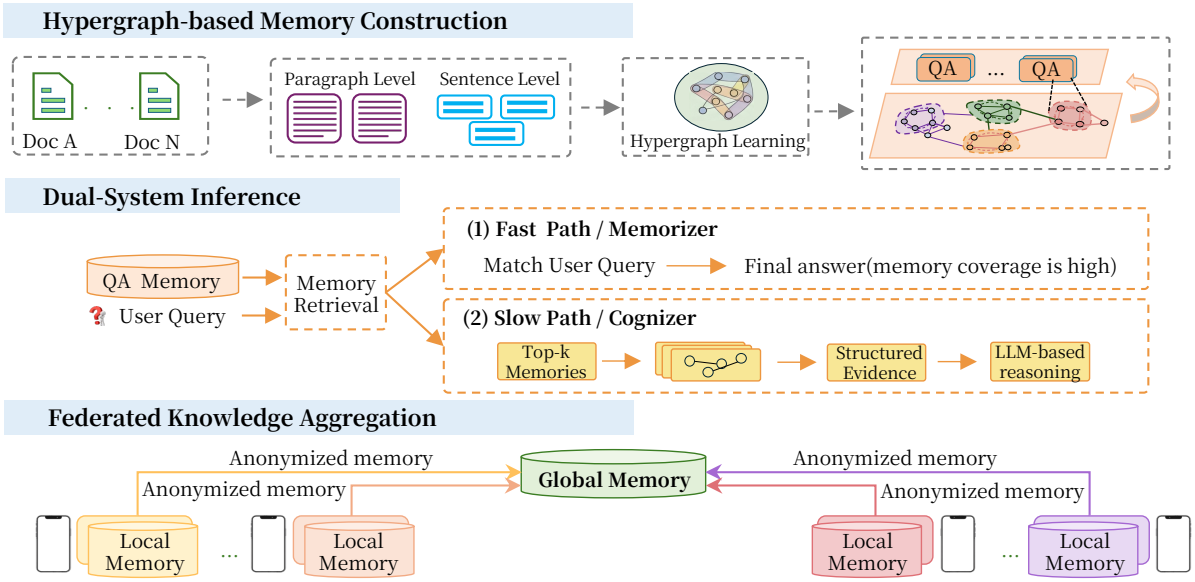

Figure 1: Overview of FD-RAG. We organize each local corpus into a semantic hypergraph and convert hyperedges into grounded QA memories. At inference time, the *Memorizer* answers high-confidence queries via direct memory matching, while the *Cognizer* retrieves structured evidence from supporting hyperedges and performs LLM-based reasoning. Local memories can be anonymized and aggregated across devices under federated constraints.

3.2 Problem Formulation

Let $\mathcal{Z} = \{Z_k\}_{k=1}^{K}$ be a set of edge devices, where each device Z_k holds a local corpus $\mathcal{T}_k$ and a collection of user queries $\mathcal{Q}_k = \{q_{kl}\}_{l=1}^{N_k}$. We aim to build a federated RAG system $\mathcal{M}$ that enables each device to produce accurate answers without exposing its local data or queries. Specifically, the answer to query q_{kl} is generated as:

$$a_{kl} = \mathcal{M}_k(q_{kl}; \mathcal{T}_k, \mathcal{G}), \quad \mathcal{G} = \bigcup_{k=1}^{K} \Gamma_k, \quad (2)$$

where Γ_k is an anonymized, shareable knowledge summary distilled from $\mathcal{T}_k$, and $\mathcal{G}$ aggregates cross-device knowledge without centralizing raw data.

We seek Pareto-optimal solutions that maximize answer accuracy $\mathcal{U} = \sum_{k,l} \mathrm{Acc}(a_{kl})$ and minimize end-to-end latency $\mathcal{L}$ under knowledge-sharing constraints:

$$\max_{\mathcal{M}, \{\Gamma_k\}} \quad [\mathcal{U}, \ -\mathcal{L}(\mathcal{M})]^\top \quad \text{s.t.} \quad \Gamma_k = \Phi(\mathcal{T}_k), \quad \forall k \in [K], \quad (3)$$

where $\Phi(\cdot)$ denotes local knowledge distillation. This formulation captures the accuracy–latency trade-off while restricting federation to distilled knowledge. The solutions lie on the Pareto frontier (Hochman and Rodgers, 1969), enabling principled trade-offs in resource-constrained edge environments.

4 Federated Dual-System RAG

In this section, we present FD-RAG, which consists of three components: Hypergraph-based Memory Construction (Section 4.1), Dual-System Inference (Section 4.2), and Federated Knowledge Aggregation (Section 4.3), as illustrated in Figure 1. Concretely, FD-RAG first constructs a semantic hypergraph over the local corpus and distills it into a set of hyperedge-grounded QA memories. During inference, the *Memorizer* resolves queries with sufficient memory coverage via direct matching, while the *Cognizer* handles harder queries by localizing relevant hyperedges and invoking LLM-based reasoning over the structured evidence. Cross-device knowledge gaps are addressed by sharing anonymized memories under federated constraints. We describe each component in detail below.

4.1 Hypergraph-based Memory Construction

Semantic-Aware Hypergraph Learning. To capture higher-order semantic relations across text units without relying on costly extraction procedures, we learn the corpus structure directly from dense semantic representations.

A central design consideration is representational granularity: sentence-level units provide precise, localized anchors suitable for fact-level memory, while paragraph-level units preserve broader discourse context necessary for multi-hop and com-

positional reasoning. To capture both, we segment the local corpus $\mathcal{T}$ into paragraph-level units $P = \{p_i\}_{i=1}^{I}$ and sentence-level units $S = \{s_j\}_{j=1}^{J}$, and encode them with a pre-trained dense encoder (e.g., BGE-M3 (Chen et al., 2024)), yielding embedding matrices $E_P \in \mathbb{R}^{I\times D}$ and $E_S \in \mathbb{R}^{J\times D}$, where D denotes the embedding dimension.

For each granularity $t \in \{p, s\}$, let $X^t \in \mathbb{R}^{N_t\times D}$ denote the node embedding matrix, where $X^p = E_P$ and $X^s = E_S$. We introduce M^t learnable hyperedges and optimize a soft incidence matrix $\hat{H}^t \in [0,1]^{N_t\times M^t}$, where $\hat{H}^t_{nm}$ measures the association degree between node n and hyperedge m. Each row is constrained to a probability simplex (Boyd and Vandenberghe, 2004) to ensure interpretable membership:

$$\sum_{m=1}^{M^t} \hat{H}^t_{nm} = 1, \qquad \hat{H}^t_{nm} \geq 0. \tag{4}$$

We then sparsify $\hat{H}^t$ by retaining only assignments above a threshold μ, yielding a compact incidence matrix H^t. The prototype of each hyperedge e^t_m is computed as the weighted mean of its incident node embeddings:

$$e^t_m = \frac{\sum_{n=1}^{N_t} H^t_{nm}\, x^t_n}{\sum_{n=1}^{N_t} H^t_{nm}} \in \mathbb{R}^D, \tag{5}$$

where x^t_n denotes the embedding of node n at granularity t.

We optimize the hyperedge assignments via two complementary objectives (Shang et al., 2024). The intra-hyperedge term enforces semantic compactness within each hyperedge:

$$L^t_{\text{intra}} = \frac{1}{M^t}\sum_{m=1}^{M^t} \frac{1}{|\mathcal{N}(e^t_m)|} \sum_{x^t_n\in\mathcal{N}(e^t_m)} \|x^t_n - e^t_m\|_2, \tag{6}$$

where $\mathcal{N}(e^t_m)$ denotes the node set incident to e^t_m. The inter-hyperedge term regulates the global geometry of hyperedge prototypes: rather than enforcing indiscriminate separation, it attracts semantically similar hyperedges while repelling dissimilar ones:

$$L^t_{\text{inter}} = \frac{1}{(M^t)^2}\sum_{i=1}^{M^t}\sum_{j=1}^{M^t}\Big(\rho_{ij}\|e^t_i - e^t_j\|_2 + (1-\rho_{ij})\max\big(\gamma - \|e^t_i - e^t_j\|_2,\, 0\big)\Big), \tag{7}$$

where ρ_{ij} denotes the cosine similarity (Salton, 1989) between hyperedge prototypes e^t_i and e^t_j, and γ is a margin hyperparameter. The overall objective balances local compactness and global discrimination:

$$L_{\text{total}} = \sum_{t\in\{p,s\}} \big(\lambda\, L^t_{\text{intra}} + (1-\lambda)\, L^t_{\text{inter}}\big). \tag{8}$$

Here, $\lambda \in [0,1]$ balances the intra-hyperedge and inter-hyperedge terms.

After optimization, paragraph-level hyperedges $\mathcal{E}^p$ and sentence-level hyperedges $\mathcal{E}^s$ jointly form the overall hyperedge set $\mathcal{E} = \mathcal{E}^p \cup \mathcal{E}^s$ with $M = |\mathcal{E}|$. Each hyperedge $e_m \in \mathcal{E}$ is associated with a set of text units $\mathcal{C}(e_m)$ from the original corpus, enabling the downstream inference framework to retrieve the corresponding textual evidence.

Proposition 1. *(Stationarity of Hyperedge Assignment Learning.) Under L-smoothness of L_{total}, the simplex-constrained optimization of the soft incidence matrix $\hat{H}^t$ attains an ϵ-stationary point in $\mathcal{O}(1/\epsilon^2)$ iterations. The proof, tailored to the probability-simplex constraints of our hyperedge assignment formulation, is provided in Appendix A.*

QA Memory Construction. To expose the hypergraph as an efficient memory interface, we convert each hyperedge into a set of retrieval-oriented QA memories. For each hyperedge $e_m \in \mathcal{E}$, we first derive a typed fact set from its associated context $\mathcal{C}(e_m)$ using a lightweight SpaCy-based extractor (Honnibal et al., 2020), providing a comprehensive factual basis for subsequent QA generation:

$$\mathcal{F}_m = \langle (u_1, v_1),\, (u_2, v_2),\, \ldots,\, (u_{T_m}, v_{T_m})\rangle, \tag{9}$$

where u_t denotes a textual fact span and v_t its semantic type. This design delegates raw-corpus fact extraction to an efficient traditional NLP pipeline, preserving localized evidence while reducing the burden on the language model. Conditioned on $\mathcal{F}_m$ and $\mathcal{C}(e_m)$, we then use a local language model to synthesize hyperedge-grounded QA memory items (see Appendix E for the prompt template). This keeps the offline memory construction stage lightweight and well suited to edge deployment. Formally,

$$\Gamma = \bigcup_{m=1}^{M} \{\gamma^r_m = (q^r_m,\, a^r_m,\, \mathcal{S}^r_m)\}_{r=1}^{R_m}, \tag{10}$$

where q^r_m and a^r_m denote the r-th question–answer pair grounded in hyperedge e_m, and $\mathcal{S}^r_m \subseteq \mathcal{E}$ is the supporting hyperedge set, with $e_m \in \mathcal{S}^r_m$ required. Additional hyperedges are included when

the question involves cross-hyperedge composition, naturally accommodating both fact-level memories ($|\mathcal{S}_m^r| = 1$) and multi-hop memories ($|\mathcal{S}_m^r| > 1$).

The resulting memory Γ serves as a structured access layer over the underlying semantic hypergraph rather than an independent synthetic QA pool. Each memory item maintains explicit links to its supporting hyperedges, preserving traceability to the original graph structure. During downstream inference, the model can answer directly from memory when the stored information is sufficient, or retrieve structured evidence from the hypergraph when more elaborate reasoning is required.

4.2 Dual-System Inference Framework

Queries exhibit heterogeneous reasoning demands: well-covered queries can be answered via direct memory matching, while complex ones require structured retrieval and LLM-based reasoning. Uniform LLM usage incurs unnecessary latency, whereas LLM-free methods sacrifice accuracy. To balance this trade-off, we propose a dual-system inference framework that routes queries based on their coverage under the QA memory Γ.

Unified Matching Score. To quantify memory coverage, we score each candidate $\gamma_r = (q_r, a_r, \mathcal{S}_r) \in \Gamma$ against the user query q by combining two complementary signals. Dense semantic similarity alone may miss structural alignment on entity-centric queries, while structural overlap alone is brittle when surface forms vary. Their combination yields a robust, low-cost coverage estimate:

$$\begin{aligned}\text{Score}(q, \gamma_r) = \alpha\ \text{Sim}\big(f(q), f(q_r)\big) \\ + (1-\alpha)\ \text{Cover}(q, \mathcal{S}_r)\end{aligned} \tag{11}$$

where $f(\cdot)$ is a dense text encoder, $\alpha \in [0, 1]$ balances the two signals, and

$$\text{Cover}(q, \mathcal{S}_r) = \frac{2|\mathcal{A}_q \cap \mathcal{A}_r|}{|\mathcal{A}_q| + |\mathcal{A}_r|} \tag{12}$$

measures the Dice overlap between the query anchor set $\mathcal{A}_q$ (salient named entities, noun phrases, and typed concept spans extracted from q) and the support anchor set $\mathcal{A}_r = \bigcup_{e \in \mathcal{S}_r} \mathcal{A}(e)$ derived from the typed facts of each supporting hyperedge.

Memorizer: Memory-Triggered Fast Thinking. For queries that are well-covered by memory, repeated LLM invocation is wasteful: the answer is already latent in the stored QA pairs. The *Memorizer* therefore identifies the best-matched item

$$r^* = \arg\max_r \text{Score}(q, \gamma_r), \tag{13}$$

and directly returns a_{r^*} whenever $\text{Score}(q, \gamma_{r^*}) \geq \delta$, bypassing LLM inference entirely. This fast path eliminates the dominant source of latency for the majority of queries while preserving answer fidelity, as the retrieved answer is grounded in hyperedge-verified evidence from the construction phase.

Cognizer: Hyperedge-Grounded Slow Thinking. When $\text{Score}(q, \gamma_{r^*}) < \delta$, direct memory matching is insufficient as the query may require compositional reasoning across multiple evidence pieces that no single memory item can cover. Rather than falling back to full-corpus retrieval, the *Cognizer* reuses the memory layer as a *localization interface*: the top-K items $\mathcal{R}_q = \text{TopK}(q, \Gamma)$ are retrieved, and their supporting hyperedges are aggregated as

$$\mathcal{E}_q = \bigcup_{\gamma_r \in \mathcal{R}_q} \mathcal{S}_r. \tag{14}$$

For each $e \in \mathcal{E}_q$, the source context $\mathcal{C}(e)$ and typed fact set $\mathcal{F}(e)$ are assembled into structured evidence units $z_e = \big(\mathcal{C}(e), \mathcal{F}(e)\big)$, over which the LLM reasons using the prompt in Appendix E:

$$\begin{aligned}\mathcal{Z}_q &= \{z_e \mid e \in \mathcal{E}_q\}, \\ \hat{a} &= \text{LLM}\big(q, \mathcal{Z}_q\big).\end{aligned} \tag{15}$$

This design is critical for edge deployment: by confining LLM reasoning to a small, hyperedge-selected evidence set rather than the full corpus, the slow path achieves targeted inference without sacrificing compositional reasoning capability.

4.3 Federated Knowledge Aggregation

Local corpora on individual devices are inherently incomplete, and evidence for a query may be distributed across devices rather than available to any single one. To mitigate this knowledge fragmentation, FD-RAG performs federation at the memory level: each device shares QA memories distilled from its local hypergraph, while raw corpora and full hypergraph structures remain on device.

Local Memory Export. Each device k constructs a local semantic hypergraph $\mathcal{H}_k$ and QA memory Γ_k from its corpus $\mathcal{T}_k$ (Section 4.1). For federation, the device uploads a shareable memory view derived from Γ_k. In privacy-sensitive settings, sensitive entities in the typed facts (Eq. 9) are perturbed via randomized response (Warner, 1965) to satisfy ϵ-LDP (Dwork, 2008). To preserve semantic utility, each entity e is replaced with a surrogate sampled from a candidate set W ($|W| = c$) containing e and $c - 1$ semantically similar alternatives.

Proposition 2. *(ϵ-LDP Perturbation Mechanism.) Given a sensitive entity e with candidate set $W = \{e, w_1, \ldots, w_{c-1}\}$ of semantically similar alternatives, the perturbation mechanism*

$$\Pr[e' \mid e] = \begin{cases} \dfrac{e^{\epsilon}}{e^{\epsilon} + c - 1}, & e' = e, \\ \dfrac{1}{e^{\epsilon} + c - 1}, & e' \in W,\ e' \neq e, \end{cases} \tag{16}$$

satisfies ϵ-local differential privacy. Proof in Appendix B.

This mechanism preserves utility while obfuscating device-specific identifiers. The anonymized memory is defined as

$$\widetilde{\Gamma}_k = \text{Anonymize}(\Gamma_k). \tag{17}$$

Global Memory Fusion. Given uploads $\{\widetilde{\Gamma}_k\}_{k=1}^{K}$, the server aggregates them into a global memory bank:

$$\Gamma^g = \bigcup_{k=1}^{K} \widetilde{\Gamma}_k. \tag{18}$$

Because different devices often contain complementary evidence, Γ^g extends the coverage of any single device. It improves the *Memorizer* by increasing the chance of high-confidence matches, and assists the *Cognizer* by providing cross-device cues for localizing the hyperedges most likely to contain the required evidence. Federation therefore serves primarily as knowledge-level collaboration to reduce fragmentation, while privacy protection remains an enabling safeguard for sensitive deployments.

5 Experiments

We evaluate FD-RAG from three complementary perspectives. **RQ1:** How does FD-RAG compare with representative baselines in overall answer quality and efficiency under both local and federated settings? **RQ2:** How does the proposed Dual-System inference mechanism balance the Memorizer and the Cognizer, and does it provide a better accuracy–efficiency trade-off than always using either path alone? **RQ3:** Are the main design choices in FD-RAG all necessary, and how does removing each component affect answer quality and latency?

5.1 Experimental Setup

Benchmarks. We follow the benchmark selection used in Jiang et al. (2024), evaluating our method on HotPotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). For consistency in evaluation protocol, we adopt the experimental settings introduced in LongBench (Bai et al., 2024).

Dataset Construction. We evaluate all methods under two settings. (1) Local setting: Each query is assigned to a home client whose local document store contains all of its gold supporting documents $D^+(q)$, so that no cross-silo retrieval is required to answer it. Since conventional RAG baselines are not designed with a decentralized protocol, evaluating them under the federated setting would introduce an inherent architectural disadvantage; we therefore restrict their evaluation to this setting to ensure a fair and controlled comparison. (2) Federated setting: Following prior work (Wang et al., 2024), we partition the corpus into disjoint subsets assigned to different clients, simulating a multi-silo environment. For each query q, we define it as *local* if all documents in $D^+(q)$ reside on its home client, and *cross-silo* otherwise. This protocol preserves the original QA pairs while introducing controlled distribution of supporting evidence across clients. Unless otherwise specified, all methods share the same document partition and query assignment on each dataset. Detailed dataset statistics are provided in Appendix C.

Evaluation Metrics. We report accuracy (ACC), F1, and average end-to-end latency per query as primary metrics. Latency is measured on device from query receipt to final answer generation and averaged over the evaluation set. We further report the average number of LLM calls per query to characterize inference cost. In federated settings, metrics are computed per client and averaged across clients.

Baselines. We compare FD-RAG against three groups of baselines. (1) **RAG baselines**: Vanilla RAG (Lewis et al., 2020b), LongRAG (Jiang et al., 2024), IterDRAG (Yue et al., 2025), EfficientRAG (Zhuang et al., 2024), RAPTOR (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), and HyperGraphRAG (Luo et al., 2025). (2) **Baselines under the federated setting**: LOCAL-RAG: a non-collaborative federated baseline where each client performs RAG using only its own local data, RAGRoute (Guerraoui et al., 2025) and C-FedRAG (Xu, 2024). (3) **FD-RAG variants**: FD-RAG-LOCAL, the single-device version without federated aggregation, and FD-RAG (W/O FUSION),

Table 1: Main results on three benchmarks under both local and federated settings. We report F1, accuracy (ACC), latency (Lat), and average LLM calls (Calls) per query. Best results in each column are shown in **bold**.

Method	HotPotQA				2WikiMQA				MuSiQue			
	F1	ACC	Lat	Calls	F1	ACC	Lat	Calls	F1	ACC	Lat	Calls
Local Setting												
Vanilla RAG (Lewis et al., 2020b)	44.3	49.6	2.4s	1.00	43.5	45.2	2.6s	1.00	11.0	9.2	1.5s	1.00
IterDRAG (Yue et al., 2025)	47.4	44.4	5.2s	2.47	38.8	43.8	6.37s	3.02	17.5	12.2	4.3s	2.13
LongRAG (Jiang et al., 2024)	57.3	53.0	10.4s	6.23	58.1	54.0	12.3s	7.35	35.5	31.0	9.2s	5.48
EfficientRAG (Zhuang et al., 2024)	52.1	50.0	3.3s	2.65	46.3	44.2	3.9s	3.22	18.3	17.0	3.6s	2.86
RAPTOR (Sarthi et al., 2024)	63.5	59.0	4.5s	1.00	61.2	50.6	5.9s	1.00	36.8	30.2	5.6s	1.00
GraphRAG (Edge et al., 2024)	55.1	50.8	6.9s	1.00	58.9	52.7	7.4s	1.00	34.2	29.8	7.1s	1.00
HyperGraphRAG (Luo et al., 2025)	60.7	58.4	4.2s	1.00	61.8	57.1	4.9s	1.00	37.4	34.8	4.1s	1.00
HippoRAG 2 (Gutiérrez et al., 2025)	63.2	60.4	3.8s	1.00	**63.5**	57.9	3.9s	1.00	**40.3**	36.2	3.4s	1.00
FD-RAG-Local (Ours)	**73.4**	**68.2**	**0.45s**	**0.32**	63.1	**59.2**	**0.62s**	**0.45**	39.1	**38.0**	**0.54s**	**0.52**
Federated Setting												
LOCAL-RAG	39.2	44.5	2.0s	1.00	41.3	43.3	2.2s	1.00	9.7	7.9	1.3s	1.00
C-FedRAG (Xu, 2024)	49.2	46.1	4.9s	2.84	50.6	47.8	5.3s	2.97	24.9	22.0	4.2s	2.73
RAGRoute (Guerraoui et al., 2025)	53.4	50.9	3.0s	1.31	55.9	52.8	3.2s	1.38	28.7	26.0	2.7s	1.24
FD-RAG (w/o fusion)	65.0	61.2	1.42s	0.80	60.3	56.7	1.58s	0.86	35.6	32.8	1.46s	0.82
FD-RAG (Ours)	**68.9**	**64.5**	**1.14s**	**0.62**	**62.6**	**58.9**	**1.28s**	**0.68**	**38.3**	**35.7**	**1.21s**	**0.65**

Table 2: Dual-system inference decomposition on HotPotQA. Fast-path coverage is the fraction of queries resolved by the Memorizer; fast/slow ACC are conditional accuracies on each subset; Oracle is an upper bound only.

Method	Fast Cover.	Fast ACC	Slow ACC	Overall ACC	Avg. Lat.
FD-RAG (full)	68.0%	77.7	48.0	68.2	0.45s
Mem.-only	100.0%	38.2	—	38.2	0.16s
Cog.-only	0.0%	—	59.3	59.3	2.13s
Oracle (upper)	80.0%	—	—	75.0	0.55s

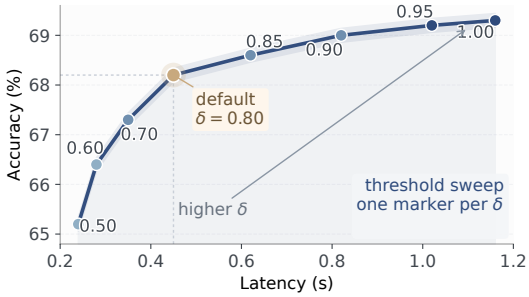

Figure 2: Pareto frontier of accuracy and latency on HotPotQA under varying confidence threshold δ. Each marker corresponds to one threshold setting, and the highlighted marker denotes the default $\delta = 0.8$.

which keeps the federated deployment protocol but removes the global memory fusion stage.

Implementation Details. We use $\mu = 0.5$, $K = 5$, $\lambda = 0.6$, and $\delta = 0.8$ across all datasets and clients. For privacy-preserving memory sharing, we set the local differential privacy budget to $\epsilon = 1.0$ with candidate set size $c = 5$. The semantic-aware hypergraph objective is optimized for 300 steps with a learning rate of 0.05, using a fixed hyperparameter configuration throughout. We adopt BGE-M3 (Chen et al., 2024) as the dense encoder and `Llama-3.1-8B` (Grattafiori et al., 2024) (INT4 quantized) as the language model. All federated experiments use five clients. FD-RAG is deployed on an NVIDIA Jetson Orin Nano 8GB (NVIDIA Corporation, 2022), with end-to-end latency measured on device using batch size 1. Additional deployment details and fairness controls are provided in Appendix C.2.

5.2 Main Results (RQ1)

Table 1 shows that FD-RAG consistently achieves a stronger accuracy–efficiency frontier than both conventional RAG baselines and federated competitors. In the local setting, FD-RAG-LOCAL achieves the best ACC on all benchmarks and attains 68.2 on HotPotQA, outperforming HippoRAG 2 by 7.8% ACC while reducing latency by 8.4×. The gain is not merely due to using structured retrieval, since FD-RAG-LOCAL also surpasses GraphRAG and HyperGraphRAG while keeping the average number of LLM calls below 0.6 across datasets. This suggests that the benefit comes from exposing the learned structure as QA memory and invoking LLM reasoning only when memory coverage is insufficient. In the federated setting, the drop of LOCAL-RAG to 44.5 ACC on HotPotQA highlights the

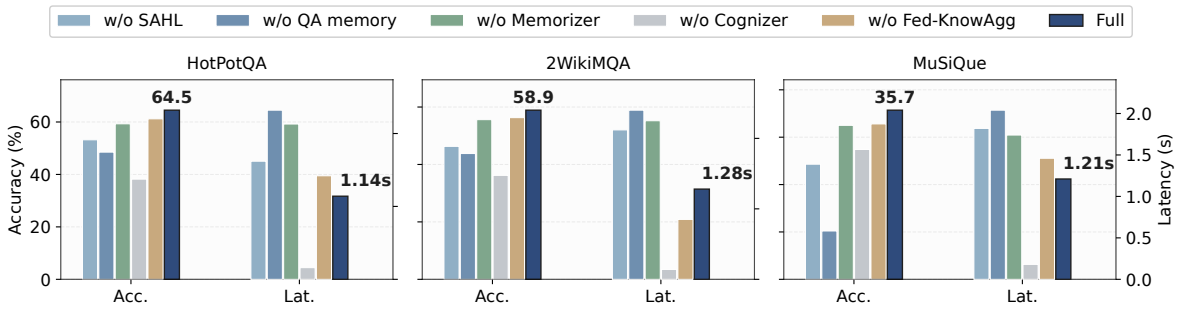

Figure 3: Ablation study on HotPotQA, 2WikiMQA, and MuSiQue. Each panel reports accuracy (Acc.) and latency (Lat.) for the full model and five ablated variants.

severity of cross-silo knowledge fragmentation. FD-RAG substantially recovers this gap, outperforming RAGRoute by 13.6 ACC points while remaining 2.6× faster, and it consistently improves over FD-RAG (w/o fusion) across all benchmarks. Together, these results indicate that memory-level federation effectively expands coverage for distributed evidence while preserving the lightweight inference profile of the local model.

5.3 Dual-System Inference Analysis (RQ2)

Table 2 provides a direct view of how the two inference paths divide labor. At the default threshold $\delta = 0.8$, the Memorizer resolves 68.0% of queries with 77.7 conditional ACC, so most inputs can terminate on the fast path without triggering full LLM reasoning. At the same time, the large gap between the full model and *Mem.-only* (68.2 vs. 38.2 ACC) shows that direct memory matching alone cannot absorb the compositional burden of multi-hop QA. The comparison with *Cog.-only* is equally revealing: although it invokes the LLM on every query, it still trails the full system by 8.9 points and incurs nearly 5× higher latency. This indicates that QA memory contributes more than routing. The top-K retrieved items act as structured hyperedge pointers that narrow the evidence space before LLM inference, improving slow-path reasoning quality in addition to reducing cost. Figure 2 further shows a smooth Pareto frontier as δ varies, with the default setting lying near the knee of the curve. The routing gate therefore functions as a controlled mechanism for balancing accuracy and latency, rather than a brittle heuristic.

5.4 Ablation Study (RQ3)

Figure 3 shows that the gain of FD-RAG is distributed across representation, inference, and federation modules rather than dominated by a single design choice. Removing semantic-aware hypergraph learning (*w/o SAHL*) consistently lowers accuracy and increases latency, indicating that the learned hypergraph is not an auxiliary construction step but the structural basis for precise memory formation and evidence localization. Removing QA memory (*w/o QA memory*) is even more damaging on both axes, since QA memory is the interface shared by the Memorizer and the Cognizer: once it is removed, the model loses both efficient direct matching and focused grounding for the slow path. The two inference ablations reveal complementary failure modes. *w/o Memorizer* mainly hurts efficiency by forcing all queries through LLM-based reasoning, whereas *w/o Cognizer* causes the sharpest accuracy collapse, showing that memory matching alone cannot support compositional reasoning. Finally, the drop of *w/o Fed-KnowAgg* confirms that global memory fusion remains important when supporting evidence is distributed across clients. Additional experimental results are provided in Appendix D.

6 Conclusion

We presented FD-RAG, a federated dual-system RAG framework for edge environments. By organizing local corpora into semantic hypergraphs and distilling them into QA memories, FD-RAG enables efficient memory-based response for well-covered queries while reserving LLM-based reasoning for more complex cases. This fast–slow decoupling, together with federated memory aggregation, provides a practical way to improve knowledge utilization under data locality and resource constraints. Empirical results validate the effectiveness of FD-RAG in balancing accuracy and efficiency, while the proposed hypergraph learning objective admits convergence guarantees. We hope FD-RAG offers a principled foundation for RAG deployment in real-world edge scenarios.

7 Limitations

FD-RAG demonstrates strong performance in edge-device application scenarios; however, its adaptation to entirely new domains or tasks still relies heavily on an offline construction and optimization process. This limitation arises because the system's knowledge base, memory organization, and reasoning patterns are built from existing data during offline preparation. As a result, when a new domain introduces unfamiliar concepts, terminologies, or relational structures, the system typically requires offline rebuilding, retraining, or re-indexing before it can achieve strong performance, rather than adapting immediately during deployment. While this design helps keep online inference efficient, it also limits the speed and flexibility of domain transfer. Future work will therefore focus on developing more incremental and adaptive mechanisms to reduce the cost of offline reconstruction. In particular, transfer learning and lightweight continual updating may help transfer accumulated knowledge from known domains to new ones more efficiently, thereby improving adaptation speed and system robustness.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, and Lei Hou. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Abhijit Chakraborty, Chahana Dahal, and Vivek Gupta. 2025. Federated retrieval-augmented generation: A systematic mapping study. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7362–7374, Suzhou, China. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Cynthia Dwork. 2008. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Val Andrei Fajardo, David B. Emerson, Amandeep Singh, Veronica Chatrath, Marcelo Lotif, Ravi Theja Desetty, Chi Ho Cheung, and Izuki Matsuba. 2025. Fedrag: A framework for fine-tuning retrieval-augmented generation systems. *arXiv preprint arXiv:2506.09200*.

Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. 2025. Minirag: Towards extremely simple retrieval-augmented generation. *arXiv preprint arXiv:2501.06713*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Rachid Guerraoui, Anne-Marie Kermarrec, Diana Petrescu, Rafael Pires, Mathis Randl, and Martijn de Vos. 2025. Efficient federated search for retrieval-augmented generation. *arXiv preprint arXiv:2502.19280*. To appear in EuroMLSys'25 proceedings.

Gauthier Guinet, Behrooz Omidvar-Tehrani, Anoop Deoras, and Laurent Callot. 2024. Automated evaluation of retrieval-augmented language models with task-specific exam generation. *arXiv preprint arXiv:2405.13622*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*.

Hangyu He, Xin Yuan, Kai Wu, Ren Ping Liu, and Wei Ni. 2025. pfedrag: A personalized federated retrieval-augmented generation system with depth-adaptive tiered embedding tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

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*.

Harold M Hochman and James D Rodgers. 1969. Pareto optimal redistribution. *The American economic review*, 59(4):542–557.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*.

Daniel Kahneman. 2003. Maps of bounded rationality: Psychology for behavioral economics. *American economic review*, 93(5):1449–1475.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, and Naman Goyal. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, and Tim Rocktäschel. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, and Wanli Ouyang. 2024a. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12758–12786.

Xinze Li, Yixin Cao, Yubo Ma, and Aixin Sun. 2024b. Long context vs. rag for llms: An evaluation and revisits. *arXiv preprint arXiv:2501.01880*.

Zhilin Liang, Yuxiang Wang, Zimu Zhou, Hainan Zhang, Boyi Liu, and Yongxin Tong. 2026. Fedmosaic: Federated retrieval-augmented generation via parametric adapters. *arXiv preprint arXiv:2602.05235*.

Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. *arXiv preprint arXiv:2502.12442*.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning*.

Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, and Yifan Zhu. 2025. Hypergraphrag: Retrieval-augmented generation with hypergraph-structured knowledge representation. *arXiv preprint arXiv:2503.21322*.

NVIDIA Corporation. 2022. Nvidia jetson orin. Accessed: 2026-03-24.

Agada Joseph Oche, Ademola Glory Folashade, Tirthankar Ghosal, and Arpan Biswas. 2025. A systematic review of key retrieval-augmented generation (rag) systems: Progress, gaps, and future directions. *arXiv preprint arXiv:2507.18910*.

Gerard Salton. 1989. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 169.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Zongjiang Shang, Ling Chen, Binqing Wu, and Dongliang Cui. 2024. Ada-mshyper: adaptive multi-scale hypergraph transformer for time series forecasting. *Advances in Neural Information Processing Systems*, 37:33310–33337.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. 2024. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 763–773.

Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American statistical association*, 60(309):63–69.

Chenhao Xu, Longxiang Gao, Yuan Miao, and Xi Zheng. 2025. Distributed retrieval-augmented generation. *arXiv preprint arXiv:2505.00443*.

Ziyue Xu. 2024. C-fedrag: A confidential federated retrieval-augmented generation system. *arXiv preprint arXiv:2412.13163*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, and Ziyu Jiang. 2024. Crag–comprehensive rag benchmark. *arXiv preprint arXiv:2406.04744*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*.

Dongfang Zhao. 2024. Frag: Toward federated vector database management for collaborative and secure retrieval-augmented generation. *arXiv preprint arXiv:2410.13272*.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. Efficientrag: Efficient retriever for multi-hop question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3411.

A Proof of Proposition 1

A.1 Problem Setup

We analyze the convergence of the parameter $H^t \in \mathbb{R}^n$ in an hypergraph learning framework. The variable H^t is constrained to lie on the probability simplex Δ^n, defined as follows.

Definition 1 (Probability Simplex)**.** The n-dimensional probability simplex is $\Delta^n := \{H \in \mathbb{R}^n \mid H_i \geq 0 \text{ for all } i,\ \sum_{i=1}^n H_i = 1\}$.

We consider the optimization problem $\min_{H^t \in \Delta^n} L_{\text{total}}(H^t)$, where $L_{\text{total}} : \mathbb{R}^n \to \mathbb{R}$ is a differentiable objective function.

Assumption 1 (Smoothness)**.** The objective function L_{total} is L-smooth, i.e., $\|\nabla L_{\text{total}}(x) - \nabla L_{\text{total}}(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^n$.

We employ Projected Gradient Descent (PGD) to solve the above problem.

Definition 2 (Projected Gradient Descent (PGD))**.** Given a step size $\eta > 0$, the PGD update rule is

$$H_{k+1}^t = \text{Proj}_{\Delta^n}\left(H_k^t - \eta \nabla L_{\text{total}}(H_k^t)\right), \quad (19)$$

where $\text{Proj}_{\Delta^n}(\cdot)$ denotes the Euclidean projection onto Δ^n.

Definition 3 (Gradient Mapping)**.** The gradient mapping at iteration k is $G_k^t := \frac{H_k^t - H_{k+1}^t}{\eta}$.

A.2 Descent Property and Convergence

We now establish the descent property of the PGD updates.

Lemma 1 (Descent Lemma for PGD)**.** *Let $\eta = \frac{1}{L}$. Then, for each iteration k, the following holds:*

$$L_{total}(H_{k+1}^t) \leq L_{total}(H_k^t) - \frac{1}{2L}\|G_k^t\|^2. \quad (20)$$

Proof. For brevity, let $L_k := L_{\text{total}}(H_k^t)$ and $g_k := \nabla L_{\text{total}}(H_k^t)$. By L-smoothness,

$$L_{k+1} \leq L_k + g_k^\top (H_{k+1}^t - H_k^t) + \frac{L}{2}\|H_{k+1}^t - H_k^t\|^2. \quad (21)$$

Substituting $H_{k+1}^t - H_k^t = -\eta G_k^t$, we obtain

$$L_{k+1} \leq L_k - \eta g_k^\top G_k^t + \frac{L\eta^2}{2}\|G_k^t\|^2. \quad (22)$$

Next, the optimality condition of Euclidean projection gives

$$\left\langle H_k^t - \eta g_k - H_{k+1}^t,\ y - H_{k+1}^t \right\rangle \leq 0, \quad \forall y \in \Delta^n. \quad (23)$$

Setting $y = H_k^t$ and using the definition of G_k^t, we get

$$g_k^\top G_k^t \geq \|G_k^t\|^2. \quad (24)$$

Substituting this bound back yields

$$L_{k+1} \leq L_k - \eta\|G_k^t\|^2 + \frac{L\eta^2}{2}\|G_k^t\|^2. \quad (25)$$

Choosing $\eta = \frac{1}{L}$, we obtain $L_{k+1} \leq L_k - \frac{1}{2L}\|G_k^t\|^2$, which is exactly the desired inequality. □

Restatement of Proposition 1. Let $H_*^t \in \Delta^n$ be the optimal solution and define $\Delta := L_{\text{total}}(H_0^t) - L_{\text{total}}(H_*^t)$. Then, after T iterations of PGD with $\eta = \frac{1}{L}$, the minimum norm of the gradient mapping satisfies:

$$\min_{0 \leq k < T} \|G_k^t\|^2 \leq \frac{2L\Delta}{T}. \quad (26)$$

Consequently, to achieve $\|G_k^t\| \leq \epsilon$, the number of iterations required is:

$$T \geq \frac{2L\Delta}{\epsilon^2}, \quad (27)$$

i.e., the iteration complexity is $\mathcal{O}(1/\epsilon^2)$.

Proof. Summing the descent inequality from Lemma 1 over $k = 0$ to $T - 1$, we obtain $\sum_{k=0}^{T-1} \frac{1}{2L}\|G_k^t\|^2 \leq L_{\text{total}}(H_0^t) - L_{\text{total}}(H_T^t) \leq \Delta$, and hence $\sum_{k=0}^{T-1} \|G_k^t\|^2 \leq 2L\Delta$. Therefore, $\min_{0 \leq k < T} \|G_k^t\|^2 \leq \frac{1}{T}\sum_{k=0}^{T-1} \|G_k^t\|^2 \leq \frac{2L\Delta}{T}$. Solving for T such that the right-hand side is at most ϵ^2 gives the desired result. □

B Proof of Proposition 2

B.1 Preliminaries

Definition 4 (Local Differential Privacy)**.** A randomized mechanism $\mathcal{M}\colon \mathcal{X} \to \mathcal{Y}$ satisfies ϵ-local differential privacy (ϵ-LDP) if, for all inputs $x, x' \in \mathcal{X}$ and all outputs $y \in \mathcal{Y}$, it holds that

$$\frac{\Pr[\mathcal{M}(x) = y]}{\Pr[\mathcal{M}(x') = y]} \leq e^{\epsilon}.$$

This condition ensures that the mechanism's output does not reveal significant information about any individual input, thereby preserving privacy in the local model.

B.2 Restatement of Proposition 2

Proposition 2. Let e be a sensitive entity and W a candidate set with $|W| = c$. Consider the perturbation mechanism defined by the conditional distribution

$$\Pr[e' \mid e] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+c-1}, & \text{if } e' = e, \\ \frac{1}{e^{\epsilon}+c-1}, & \text{if } e' \in W,\ e' \neq e. \end{cases}$$

Then, this mechanism satisfies ϵ-local differential privacy.

B.3 Proof of Proposition 2

We prove the proposition by verifying the ϵ-LDP condition for all possible inputs $e, e^* \in \{e\} \cup W$ and outputs $e' \in \{e\} \cup W$. Specifically, we must show:

$$\frac{\Pr[e' \mid e]}{\Pr[e' \mid e^*]} \leq e^{\epsilon}.$$

Proof. We consider the following exhaustive cases:

- **Case 1:** $e' = e = e^*$**.**

$$\frac{\Pr[e' \mid e]}{\Pr[e' \mid e^*]} = \frac{\frac{e^{\epsilon}}{e^{\epsilon}+c-1}}{\frac{e^{\epsilon}}{e^{\epsilon}+c-1}} = 1 \leq e^{\epsilon}.$$

- **Case 2:** $e' = e \neq e^*$**.**

$$\frac{\Pr[e' \mid e]}{\Pr[e' \mid e^*]} = \frac{\frac{e^{\epsilon}}{e^{\epsilon}+c-1}}{\frac{1}{e^{\epsilon}+c-1}} = e^{\epsilon}.$$

- **Case 3:** $e' = e^* \neq e$**.**

$$\frac{\Pr[e' \mid e]}{\Pr[e' \mid e^*]} = \frac{\frac{1}{e^{\epsilon}+c-1}}{\frac{e^{\epsilon}}{e^{\epsilon}+c-1}} = \frac{1}{e^{\epsilon}} \leq e^{\epsilon}.$$

- **Case 4:** $e' \neq e$ **and** $e' \neq e^*$**.**

$$\frac{\Pr[e' \mid e]}{\Pr[e' \mid e^*]} = \frac{\frac{1}{e^{\epsilon}+c-1}}{\frac{1}{e^{\epsilon}+c-1}} = 1 \leq e^{\epsilon}.$$

In all cases, the privacy condition $\frac{\Pr[e'|e]}{\Pr[e'|e^*]} \leq e^{\epsilon}$ is satisfied. Therefore, the mechanism guarantees ϵ-local differential privacy. □

C Experiments Details

C.1 Statistics of Datasets

Table 3 provides detailed statistics of the datasets used in our experiments, including HotPotQA, 2WikiMQA, and MuSiQue. These datasets vary in size and complexity, offering a comprehensive evaluation framework for multi-hop question answering models.

C.2 Implementation Details

We use $\mu = 0.5$, $K = 5$, $\lambda = 0.6$, and $\delta = 0.8$ across all datasets and clients. For privacy-preserving memory sharing, we set the local differential privacy budget to $\epsilon = 1.0$ with candidate set size $c = 5$. We optimize the semantic-aware hypergraph learning objective for 300 steps using a learning rate of 0.05, and keep the same hyperparameter configuration throughout training. We use BGE-M3 (Chen et al., 2024) as the dense encoder and `Llama-3.1-8B` (Grattafiori et al., 2024) in INT4 quantized form as the language model. To ensure a consistent federated protocol, all federated experiments use five clients.

For the multi-granularity hypergraph module, we set the numbers of candidate hyperedges as $M^p = \lceil N_p/4 \rceil$ and $M^s = \lceil N_s/4 \rceil$, where N_p and N_s denote the numbers of paragraph- and sentence-level nodes, respectively. Each soft incidence matrix $\hat{H}^t$ is initialized with positive random values and row-wise normalized onto the probability simplex, after which the paragraph- and sentence-level hypergraphs are jointly optimized under the unified objective in Eq. (7), while maintaining separate incidence matrices and hyperedge prototypes for the two granularities.

We deploy FD-RAG on a real NVIDIA Jetson Orin Nano 8GB edge device. Table 4 summarizes the evaluation hardware configuration. We follow the official device specifications and report end-to-end latency measured on this platform with batch size 1 under a 15 W power limit (NVIDIA Corporation, 2022).

Table 3: Statistics of Datasets.

Dataset	Avg #Tokens	Max #Tokens	#Samples
HotPotQA	9.1k	12.7k	500
2WikiMQA	9.2k	12.3k	500
MuSiQue	11.1k	17.3k	500

Table 4: Evaluation Hardware.

Component	Specification
CPU	Cortex-A78AE, 1.2 GHz, 6-core
GPU	Ampere, 1024 CUDA cores, 32 Tensor cores, 625 MHz
Power Limit	15 W
DRAM	8 GiB LPDDR5-4250
Storage	512 GB SD Card, UHS

C.3 Fairness of Runtime Comparison.

All runtime comparisons follow a shared deployment protocol. Methods requiring answer generation use the same generator backbone (`Llama-3.1-8B`, INT4), the same dense encoder (`BGE-M3`), and the same Jetson Orin Nano 8GB environment with batch size 1 under a 15 W power limit. We also standardize the decoding setup, prompt/output length budget, and final evidence budget passed to the generator; single-stage retrievers use top-5 retrieval, while multi-stage methods retain their original pipelines but are constrained to the same final evidence budget. Non-federated baselines are reproduced as faithfully as possible from the original papers and public implementations, with only hardware-compatible adjustments that do not alter their core algorithmic workflow.

C.4 Baselines

We compare FD-RAG against representative baselines for multi-hop question answering under both local and federated settings. Methods without a native federated design are evaluated only in the local setting, while federated comparisons are restricted to approaches with an explicit decentralized protocol, to avoid introducing ad hoc adaptations.

- **Vanilla RAG** (Lewis et al., 2020b): Integrates a retriever with a generator, retrieving the top-5 relevant documents to augment context for improved answer generation.
- **IterDRAG** (Yue et al., 2025): Segments complex queries into sub-queries, utilizing iterative retrieval and in-context learning to refine answers progressively through a reasoning chain.
- **LongRAG** (Jiang et al., 2024): Employs a dual-perspective approach, combining an information extractor, chain-of-thought-guided filter, and generator to address challenges in processing long texts and identifying fine-grained factual details.
- **EfficientRAG** (Zhuang et al., 2024): Trains lightweight models to iteratively generate queries and filter irrelevant information, enhancing retrieval efficiency and performance in multi-hop question answering.
- **RAPTOR** (Sarthi et al., 2024): Constructs a hierarchical summary tree through recursive embedding, clustering, and summarization, retrieving insights across abstraction levels to handle long documents effectively.
- **GraphRAG** (Edge et al., 2024): GraphRAG is a graph-based RAG method that constructs an entity knowledge graph and leverages community-level summaries to generate and aggregate responses for global queries.
- **HippoRAG 2** (Gutiérrez et al., 2025): HippoRAG 2 builds a passage-linked knowledge graph and performs retrieval with Personalized PageRank, improving multi-hop evidence association while preserving strong factual recall.
- **HyperGraphRAG** (Luo et al., 2025): HyperGraphRAG is a hypergraph-based RAG method that models n-ary relations via hyperedges, enabling more accurate and efficient retrieval and generation than standard and graph-based RAG.
- **C-FedRAG** (Xu, 2024): C-FedRAG is a federated RAG framework that leverages confidential computing to enable secure and scalable retrieval across decentralized data sources.

(a) HotPotQA (b) 2WikiMQA (c) MuSiQue

Figure 4: Training Loss Curves on Three QA Datasets: HotPotQA, 2WikiMQA, MuSiQue

- **RAGRoute** (Guerraoui et al., 2025): Introduces a routing mechanism to dynamically select the most appropriate retrieval and generation strategies based on query characteristics.

D Additional Experimental Results

D.1 Training Convergence Results

We evaluated the convergence performance of our method across three distinct datasets. As illustrated in Figure 4 , the loss curves of these datasets demonstrate the following trends: 1) Rapid Convergence: Across all datasets, the loss values sharply decline during the initial stages of training (within the first 50 epochs), indicating that the model swiftly captures thematic information and optimizes effectively early on. 2) Stability: As training progresses, the loss values stabilize after 100 epochs, suggesting that the model has largely converged without significant oscillations, which indicates the robustness of hypergraph learning across different datasets. 3) Consistency: The uniformity of the loss trends across the three datasets further supports the applicability of this method to various types of question-answering datasets, highlighting its adaptability and generalization capacity.

D.2 Offline Construction Cost Analysis

We evaluate the offline construction cost of FD-RAG on HotPotQA and compare it with RAPTOR, GraphRAG, HippoRAG 2, and HyperGraphRAG under the same deployment setting as the main experiments. All methods use the same dense encoder (`BGE-M3`) and local generator (`Llama-3.1-8B`, INT4), and all measurements are collected on Jetson Orin Nano 8GB.

We report three metrics: LLM output tokens per 1k corpus tokens (**OutTok/1kT**), wall-clock construction time per 1k corpus tokens (**TP1kT**), and peak memory during construction. For FD-RAG, only QA-memory synthesis contributes to LLM token usage; SPACY-based fact extraction and hypergraph optimization are counted in runtime but not in token usage. Lower is better for all metrics.

As shown in Table 6, FD-RAG achieves the lowest cost on all three metrics, with 58 OutTok/1kT, 7.1 seconds TP1kT, and 5.5 GB peak memory. Compared with HippoRAG 2, FD-RAG reduces token usage, construction time, and peak memory by 67.0%, 58.0%, and 27.6%, respectively. Compared with HyperGraphRAG, it reduces construction time from 19.3 to 7.1 seconds and peak memory from 7.2 to 5.5 GB. These results indicate that FD-RAG offers a more efficient offline construction profile and is better suited to resource-constrained edge deployment.

D.3 Privacy Protection Evaluation

We assess whether memories sanitized before federation still leak recoverable sensitive entities. To this end, we extract person, organization, and location mentions from shared QA pairs and supporting-fact tuples, and perform an *LLM restoration attack* with `gpt-4o-mini`: given a sanitized item, the model is prompted to recover the original entity from context. We compare three sharing strategies: NO PROTECTION, which shares raw memories; TYPE MASKING, which replaces each sensitive entity with a coarse placeholder such as `[PERSON]`; and FD-RAG PRIVACY (ours), which applies the proposed semantic candidate perturbation under ϵ-LDP. We use Restoration Acc@1 to measure privacy leakage and downstream QA accuracy to measure utility. The $\epsilon = 1.0$ setting for FD-RAG PRIVACY is the default federated configuration and therefore matches the results reported in Table 1.

Table 5 reveals a clear privacy–utility trade-off. NO PROTECTION yields restoration accuracy above 89 on all three benchmarks, indicating that raw shared memories leak entity identity almost directly. TYPE MASKING minimizes leakage, reducing average Restoration Acc@1 to 3.4, but substantially de-

Table 5: Privacy protection evaluation under an LLM restoration attack in the federated setting. FD-RAG Privacy uses the default budget $\epsilon = 1.0$ with candidate set size $c = 5$. Restoration Acc@1 measures how often the attacker correctly recovers the original sensitive entity from a shared QA/fact item; lower is better. QA ACC measures downstream utility after sharing; higher is better. Best results in each column are **bolded**; second best are underlined.

Method	HotPotQA		2WikiMQA		MuSiQue		Average	
	Rest.@1↓	ACC↑	Rest.@1↓	ACC↑	Rest.@1↓	ACC↑	Rest.@1↓	ACC↑
No Protection	95.1	**65.4**	92.8	**59.8**	89.4	**36.5**	92.4	**53.9**
Type Masking	**3.9**	59.0	**3.4**	53.7	**2.8**	31.9	**3.4**	48.2
FD-RAG Privacy (Ours)	9.6	64.5	8.9	58.9	7.8	35.7	8.8	53.0

Table 6: Offline construction cost on HotPotQA under our deployment setting. All methods use `BGE-M3` and `Llama-3.1-8B` (INT4) on Jetson Orin Nano 8GB. OutTok/1kT denotes the number of LLM output tokens generated during construction per 1k corpus tokens. TP1kT denotes wall-clock construction time per 1k corpus tokens. Peak Mem. is measured during the offline construction stage. Lower is better for all metrics.

Method	OutTok/1kT↓	TP1kT (s)↓	Peak Mem. (GB)↓
RAPTOR (Sarthi et al., 2024)	88	12.7	6.0
GraphRAG (Edge et al., 2024)	542	39.4	7.4
HippoRAG 2 (Gutiérrez et al., 2025)	176	16.9	7.6
HyperGraphRAG (Luo et al., 2025)	214	19.3	7.2
FD-RAG (ours)	**58**	**7.1**	**5.5**

Table 7: Privacy–utility trade-off of FD-RAG Privacy under different local privacy budgets ϵ in the federated setting. We fix the candidate set size to $c = 5$ and report averages over HotPotQA, 2WikiMQA, and MuSiQue. The default setting used in the main experiments is $\epsilon = 1.0$. Lower Restoration Acc@1 indicates less leakage; higher QA ACC indicates better utility.

Privacy Budget	Rest.@1↓	ACC↑
$\epsilon = 0.1$	4.7	49.4
$\epsilon = 0.5$	6.4	51.4
$\epsilon = 1.0$ **(default)**	8.8	53.0
$\epsilon = 2.0$	14.3	53.6

grades utility because coarse placeholders remove task-relevant semantics for retrieval and answer selection. In contrast, FD-RAG Privacy keeps the attacker success rate below 10% on every dataset while preserving nearly the same QA performance as raw sharing. Averaged across datasets, it reduces Restoration Acc@1 from 92.4 to 8.8, a 90.5% relative reduction in leakage compared with No Protection, while retaining 98.3% of its utility (53.0 vs. 53.9 average ACC). Relative to Type Masking, it improves downstream accuracy by 4.8 points on average at the cost of only a modest increase in restoration accuracy. These results show that FD-RAG Privacy offers a substantially better privacy–utility balance than either raw sharing or coarse masking.

Table 7 further examines this trade-off under different privacy budgets. As expected, larger ϵ weakens perturbation: the average restoration success rate rises from 4.7 at $\epsilon = 0.1$ to 14.3 at $\epsilon = 2.0$, while QA accuracy increases from 49.4 to 53.6. Notably, even under the strictest setting, FD-RAG Privacy still outperforms Type Masking in utility (49.4 vs. 48.2 average ACC), suggesting that semantically constrained perturbation preserves more task-relevant information than coarse placeholders. The default setting $\epsilon = 1.0$ provides a balanced operating point and matches the main federated results in Table 1: it keeps leakage an order of magnitude below raw sharing (8.8 vs. 92.4) while retaining nearly all downstream utility (53.0 vs. 53.9). Overall, FD-RAG supports a smooth and practical privacy–utility trade-off.

D.4 Analysis of Generated QA Memory Questions

Following prior work (Guinet et al., 2024), we conduct a statistical analysis of the generated QA-memory questions to assess their diversity and coverage. Concretely, we randomly sampled 100 question–document (context) pairs from the three datasets and analyzed the corresponding generated questions along two dimensions:

Question Type Analysis. Drawing on several prior research efforts (Guinet et al., 2024; Yang

Table 8: Number of Questions by Type Across Different Datasets

Question Type	HotPotQA	MuSiQue	2WikiMQA
Aggregation Questions	953	894	1188
Comparison Questions	910	844	1105
False Premise Questions	912	854	1097
Multi-hop Questions	909	862	1143
Post-processing Heavy Questions	930	876	1152
Set Questions	1469	1459	1562

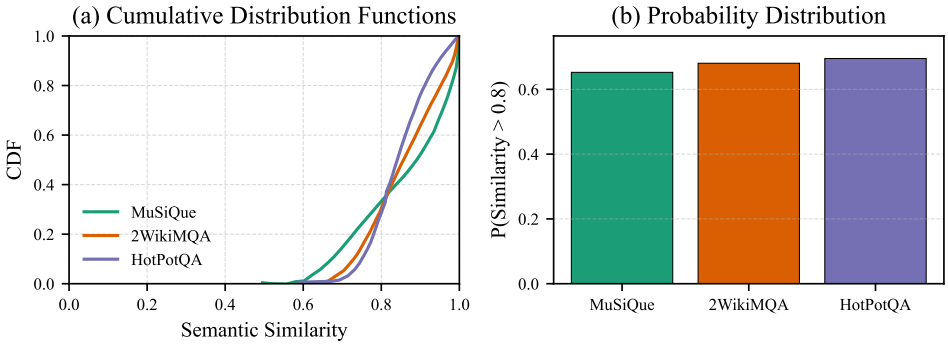

Figure 5: Performance evaluation of the generated questions with respect to semantic similarity.

et al., 2024), we further divided the complex questions generated by our QA memory construction pipeline into six categories: Set, Comparison, Aggregation, Multi-hop, Post-processing Heavy, and False Premise. This classification allows us to better assess the diversity and depth of QA memory question generation. As shown in Table 8, our approach consistently covers a wide range of question types across different benchmark datasets (HotPotQA, MuSiQue, and 2WikiMQA), demonstrating its strong generalization capability. In particular, the substantial presence of complex forms such as Set and Multi-hop questions suggests that the generated questions are not only varied in structure but also require multi-faceted reasoning and synthesis across information sources—an essential characteristic for evaluating evidence-grounded multi-hop QA.

Question Diversity and Coverage Analysis. As shown in Figure 5 (a), we present the cumulative distribution function (CDF) of semantic similarity between the original questions and the top-1 most semantically similar questions generated by our QA memory question generation method across three datasets: MuSiQue, 2WikiMQA, and HotPotQA. This distribution reflects how closely the generated questions align with the original questions at a semantic level. A wide and smooth distribution indicates that our method is capable of generating questions with varying degrees of similarity—ranging from highly similar to more diverse ones—demonstrating both strong semantic coverage and question diversity.

To quantify this, Figure 5 (b) shows the proportion of generated questions whose semantic similarity with the original question exceeds 0.8. In all three datasets, this proportion exceeds 60%, indicating that a substantial number of generated questions are semantically close to the originals. This high proportion suggests that our method effectively captures the core semantics of the input while also generating a broad range of diverse questions. Together, these results validate that our approach achieves a good balance between semantic fidelity and diversity, resulting in high-quality and broadly representative question generation.

E Prompt Templates

We provide the prompt templates used in our experiments. The prompts are designed to elicit specific information from the model, guiding it to generate accurate and relevant responses.

Prompt Template:
Role:
You are an advanced information system responsible for generating retrieval-oriented QA memory questions grounded in the provided atomic facts and original text.

Task:
Your task is to generate complex questions based on extracted atomic facts and the original text. The questions should be answerable using only the provided information and, when appropriate, require multi-fact integration (e.g., comparison, aggregation, or multi-hop reasoning) to support downstream retrieval and evidence-grounded answering.

Requirements:
Questions must strictly rely on the extracted atomic facts and original text, without introducing any external information.
Prefer questions that are specific, unambiguous, and informative for retrieval (avoid overly generic prompts).
Encourage compositional reasoning when supported by the facts (e.g., Set / Comparison / Aggregation / Multi-hop / Post-processing Heavy / False Premise).
Answers must accurately reflect the original content and refer to specific expressions in the text or atomic facts whenever possible.
Language must be clear and logically rigorous, avoiding ambiguity.

Output Format (Follow this format strictly):
Example:
{example}

Now, based on the following atomic facts and original paragraph, generate a complex question and its corresponding answer:
Question Type:
{type}
Extracted Facts:
{extracted_facts}
Original Text:
{text}

Table 9: QA Memory Question Generation Prompt

Prompt Template:
Role:
You are now an intelligent assistant tasked with answering the final question based on the provided reference question-answer pairs and context documents. Follow these rules strictly: Only output the final answer, without any explanation or additional content.

Reference Q&A Pairs: {context}
Context Document: {document}
Question: {question}
Answer:

Table 10: RAG Prompt