FedMosaic: Federated Retrieval-Augmented Generation via Parametric Adapters

Zhilin Liang
SKLCCSE Lab
Beihang University
Beijing, China
zlliang@buaa.edu.cn

Yuxiang Wang
SKLCCSE Lab
Beihang University
Beijing, China
yuxiangwang@buaa.edu.cn

Zimu Zhou
Department of Data Science
City University of Hong Kong
Hong Kong, China
zimuzhou@cityu.edu.hk

Hainan Zhang
Beijing Advanced Innovation Center
Beihang University
Beijing, China
zhanghainan1990@163.com

Boyi Liu
SKLCCSE Lab
Beihang University
Beijing, China
boyliu@buaa.edu.cn

Yongxin Tong*
SKLCCSE Lab
Beihang University
Beijing, China
yxtong@buaa.edu.cn

Abstract

Retrieval-Augmented Generation (RAG) enhances Large Language Models (LLMs) by grounding generation in external knowledge to improve factuality and reduce hallucinations. Yet most deployments assume a centralized corpus, which is infeasible in privacy-aware domains where knowledge remains siloed. This motivates federated RAG (FedRAG), where a central LLM server collaborates with distributed silos without sharing raw documents. In-context RAG violates this requirement by transmitting verbatim documents, whereas parametric RAG encodes documents into lightweight adapters that merge with a frozen LLM at inference, avoiding raw-text exchange. We adopt the parametric approach but face two unique challenges induced by FedRAG: high storage and communication from per-document adapters, and destructive aggregation caused by indiscriminately merging multiple adapters. We present FedMosaic, the first federated RAG framework built on parametric adapters. FedMosaic clusters semantically related documents into multi-document adapters with document-specific masks to reduce overhead while preserving specificity, and performs selective adapter aggregation to combine only relevance-aligned, non-conflicting adapters. Experiments show that FedMosaic achieves an average 10.9% higher accuracy than state-of-the-art methods in four categories, while lowering storage costs by 78.8% to 86.3% and communication costs by 91.4%, and never sharing raw documents[1].

CCS Concepts

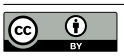
• **Computing methodologies** → **Learning paradigms**.

Keywords

Retrieval-Augmented Generation; Parametric Adapters; Federated Computing

*Corresponding author.

[1]We have open-sourced the code at: https://github.com/lewellin727/FedMosaic



SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809655

ACM Reference Format:
Zhilin Liang, Yuxiang Wang, Zimu Zhou, Hainan Zhang, Boyi Liu, and Yongxin Tong. 2026. FedMosaic: Federated Retrieval-Augmented Generation via Parametric Adapters. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3805712.3809655

1 Introduction

Retrieval-Augmented Generation (RAG) [12] has emerged a core technique for enhancing Large Language Models (LLMs) by grounding their responses in *dynamic external knowledge* rather than static pre-training alone. By retrieving relevant documents and conditioning generation on them, RAG improves factual accuracy and mitigates hallucination, driving its adoption in applications such as conversational search [43], enterprise knowledge assistants [6], and domain-specific Q&A [50]. However, most RAG deployments assume access to a *centralized* corpus (*e.g.* Wikipedia, Common Crawl, or enterprise repositories). While feasible in open-domain settings, this assumption fails in vertical domains such as healthcare and finance, where critical knowledge remains locked in institutional silos due to privacy and compliance constraints.

This limitation motivates *Federated RAG* (FedRAG), where a central LLM server collaborates with multiple distributed silos, each maintaining its own local corpus and serving as a knowledge provider without sharing its raw documents, *i.e.*, the *locality constraint* (see Fig. 1). In rare-disease diagnosis, for instance, a medical LLM could generate well-informed suggestions by leveraging clinical narratives (*e.g.* pathology reports, triage notes, genomic records) from multiple hospitals. No single hospital offers sufficient coverage for reliable decisions [3], and regulations such as HIPAA [8] and GDPR [33] prohibit centralizing sensitive records [26]. FedRAG allows the server to integrate knowledge across hospitals without exposing raw documents, supporting robust diagnosis while ensuring compliance. More broadly, it extends RAG to distributed, privacy-conscious environments that increasingly characterize modern web-of-silos information systems.

Despite rapid progress in RAG, existing methods cannot be directly extended to the federated setting because they violate the

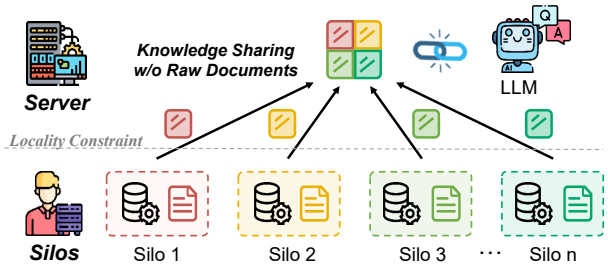


Figure 1: Federated RAG with locality constraint.

locality constraint. Conventional RAG depends on *in-context* integration, where retrieved passages are inserted into the LLM's prompt [10, 14, 19, 29, 43]. In a federated environment, this requires the server to fetch verbatim documents from silos, which inherently violates locality. Other efforts pursue privacy-preserving prompt engineering by injecting noise into raw documents, but our empirical study shows that these approaches suffer severe accuracy degradation in FedRAG settings (see Sec. 4.3.3). Noise injected independently by silos accumulates at the server, leading to performance even worse than single-silo retrieval.

Recent advances in parametric RAG [30] offer a promising alternative. Instead of raw text, documents are encoded into lightweight adapters (*e.g.* LoRA) that can be merged with the frozen base LLM model at inference. It avoids raw-text exchange and naturally satisfies the locality constraint, making it an attractive candidate for FedRAG. In practice, each silo can encode its local documents into adapters and upload relevant ones to the server for adapter aggregation and response generation. Yet directly extending parametric RAG to federated environments is far from trivial due to two challenges. *(i) Storage and communication overhead.* Training one adapter per document leads to an explosion in adapter count, imposing unsustainable storage demands at silos and heavy communication costs. A natural remedy is to encode *multiple* documents per adapter, but naive grouping of heterogeneous documents causes *intra-silo adapter interference*. *(ii) Destructive adapter aggregation*: As in federated learning, the standard adapter aggregation strategy is to average adapters across *all* silos. While this initially broadens knowledge coverage, *indiscriminate* aggregation causes *inter-silo adapter interference*. Irrelevant adapters inject noise, and parameter averaging creates conflicts.

To address these challenges, we propose FedMosaic, an efficient and accurate federated RAG framework built upon parametric adapters. FedMosaic enforces the locality constraint while reducing overhead and mitigating interference through two innovations. *(i) Multi-Document Parametric Adapters.* It clusters *semantically related* documents into a single adapter while learning *document-specific masks* to gate parameters. This preserves per-document specificity, prevents conflicting signals within an adapter, and substantially reduces storage and communication costs. *(ii) Selective Adapter Aggregation.* It ensures that only relevant and non-conflicting adapters are combined across silos. For each query, silos re-rank their local documents and share only relevance scores and masks. The server then selects globally relevant adapters with minimal parameter conflict and aggregates them to generate responses. Analogous to arranging tiles into a coherent mosaic, FedMosaic composes only the most relevant knowledge across silos, achieving locality-preserving, relevance-aware, and efficient federated RAG.

Our main contributions are summarized as follows.

- We introduce the first framework for federated RAG that enforces the locality constraint, enabling multi-silo retrieval without transmitting raw documents.
- We propose FedMosaic, a unified design that tackles the unique efficiency and accuracy challenges in federated parametric RAG. It introduces multi-document parametric adapters with document-specific masks to reduce storage and communication overhead while preserving per-document specificity, and selective adapter aggregation to mitigate inter-silo conflicts by aggregating only relevant, non-overlapping adapters.
- Extensive evaluations on four datasets show that FedMosaic achieves an average 10.9% higher accuracy than state-of-the-art methods in four categories: local RAG [29, 38, 43], in-context FedRAG [1, 13, 28, 49], federated fine-tuning [37, 48], and parametric RAG [30], while enforcing locality constraint and yielding a reduction of 78.8%to 86.3% in storage and 91.4% in communication cost.

2 Preliminaries

2.1 Federated RAG

Problem Setting. Consider a federation with a central LLM server $\mathcal{G}$ and M data silos $\{\mathcal{S}_1, \ldots, \mathcal{S}_M\}$ (*e.g.* distinct organizations or geo-distributed data centers). Each silo $\mathcal{S}_m$ maintains a local knowledge corpus $\mathcal{D}_m = \{d_m^1, \ldots, d_m^{N_m}\}$, where each d_m^i is a document. The silos act as distributed external knowledge providers, while the server is responsible for generating responses.

Given a user query q, federated retrieval-augmented generation aims to produce an answer a using knowledge across $\{\mathcal{D}_m\}_{m=1}^M$ under the *locality constraint*: no plaintext document d_m^i may leave its silo $\mathcal{S}_m$. The objective is to maximize answer *accuracy* while keeping *communication* overhead low.

Limitations of Prior Arts. Existing federated RAG frameworks [1, 13, 28, 35, 49] rely on *in-context* integration, where the server fetches *verbatim* documents from silos and inserts them into the LLM context, which violates the *locality constraint* by nature.

2.2 Parametric Adapters

Parametric RAG [30] offers a promising alternative by integrating external knowledge *without* in-context documents. It converts documents into *parametric adapters* that directly augment the LLM. The pipeline consists of two steps.

- *Document Augmentation.* For each document d^i, an LLM generates n rewrites and m question-answer (QA) pairs, forming an augmented set

$$D^i = \{(d^{i,k}, q^{i,j}, a^{i,j}) \mid 1 \le k \le n,\ 1 \le j \le m\}, \tag{1}$$

where $d^{i,k}$ is a rewrite and $(q^{i,j}, a^{i,j})$ is a QA pair.

Algorithm 1: Naive Federated Parametric RAG

Input: user query q, LLM server $\mathcal{G}$, data silos $\{\mathcal{S}_m\}_{m=1}^{M}$
Output: answer a

```
// Offline Stage
for each silo S_m do
    Train an independent LoRA adapter for each document
// Online Stage
for each silo S_m do
    Retrieve top-k documents based on q and upload the
      corresponding averaged adapters to the server
Server aggregates multiple adapters by parameter
  summation and generates answer a
return a
```

- *Parametric Encoding*. Using LoRA adapters [16], each D^i is encoded into a parametric form by minimizing the next-token loss:

$$\min_{\Delta\theta} \sum_{x \in D^i} \sum_{t=1}^{T} -\log P_{\theta+\Delta\theta}(x_t \mid x_{<t}), \tag{2}$$

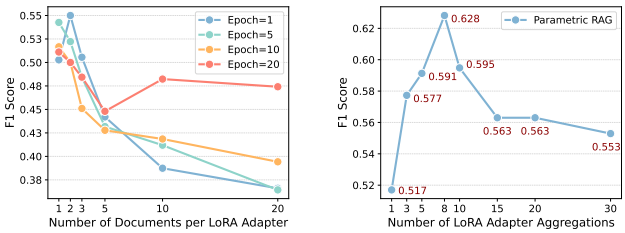
where θ is the frozen LLM parameters, $\Delta\theta$ is the trainable LoRA parameters, x_t is the t-th token in sequence x, and T is the sequence length.

At inference, relevant adapters are composed with θ rather than sending raw text. This *parametric* integration naturally enforces *locality*, making it an attractive candidate for federated RAG. Algorithm 1 illustrates the straightforward extension of parametric RAG to federated scenarios.

2.3 Challenges

Although parametric RAG is attractive, directly extending it to the federated setting introduces two challenges.

- **Storage and Communication Overhead**. Training one adapter *per document* causes an explosion in the number of adapters, leading to excessive storage requirements at silos and high silo-server communication costs. A natural remedy is to encode *multiple documents* into a single adapter. Yet experiments show that naive grouping of heterogeneous documents severely degrades accuracy. As shown in Fig. 2a, when the number of documents per adapter increases from 1 to 20, the F1 score of parametric RAG at a single silo drops sharply[2]. The degradation persists across training epochs, indicating that heterogeneous documents within a shared adapter cause *intra-silo adapter interference*.
- **Destructive Adapter Aggregation**. In federated RAG, a straightforward strategy is to aggregate adapters from *all* silos by averaging their parameters. While this can initially improve accuracy by incorporating *diverse* knowledge, *indiscriminate* aggregation ultimately harms accuracy. As shown in Fig. 2b, the F1 score improves when a small number of relevant adapters are averaged, but declines as more are included[3]. This degradation arises from *inter-silo adapter interference*, caused by two factors: *(i)* not all silo documents are equally relevant to a given query, so aggregating too many introduces noise, and *(ii)* parameter averaging can create parameter conflicts among adapters [17, 21, 23, 44]. Together, these effects lead to destructive aggregation and reduced answer quality.



(a) Grouped Documents (b) Multiple LoRA Aggregation

Figure 2: Accuracy curves of parametric RAG for (a) grouped documents under a LoRA adapter across different training epochs, and (b) aggregation of multiple LoRA adapters.

Objectives. These challenges motivate the design of a federated RAG framework that: *(i)* preserves locality by exchanging only parametric adapters, *(ii)* reduces storage and communication costs by supporting multi-document adapters without intra-silo interference, and *(iii)* mitigates destructive aggregation through selective adapter combination that avoids inter-silo interference. These objectives underpin the design of FedMosaic.

3 Method

3.1 FedMosaic Overview

We present FedMosaic, an efficient and accurate federated RAG framework built upon parametric adapters (see Fig. 3). FedMosaic addresses the challenges in Sec. 2.3 by *(i)* encoding multiple semantically related documents into shared adapters, and *(ii)* selectively aggregating relevant, non-conflicting adapter parameters across silos. Analogous to a mosaic that arranges relevant tiles into a coherent image, FedMosaic groups documents into parametric adapters and assembles only the most relevant ones across silos to support efficient and accurate federated RAG. To our knowledge, FedMosaic is the first federated RAG scheme that ensures the *locality constraint*.

Architecture. FedMosaic consists of two functional modules.

- **Multi-Document Parametric Adapters** (Sec. 3.2). To reduce storage and communication overhead, FedMosaic shares one adapter across a cluster of semantically coherent documents. This is feasible because a single document primarily activates a small subset of adapter parameters. Accordingly, FedMosaic *(i)* groups semantically similar documents and trains one cluster-level adapter per cluster, and *(ii)* learns

[2]Results are measured on 2WikiMultihopQA Bridge dataset. We randomly group a fixed number of documents to each adapter at different epochs. At inference time, the response is generated using the adapter most relevant to the query.

[3]Results are measured on 2WikiMultihopQA Comparison dataset. Each document has an independent adapter trained for 3 epochs. At inference time, we increase the number of averaged adapters for the same query from 1 to 30.

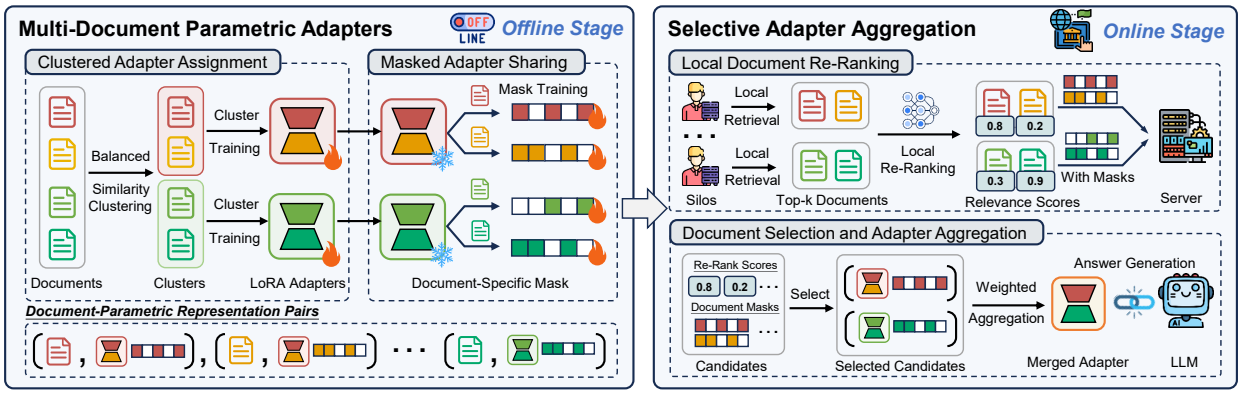


Figure 3: FedMosaic architecture and workflow.

document-specific masks that gate adapter parameters during adapter aggregation and answer generation. This design preserves per-document specificity and mitigates intra-silo adapter interference during adapter sharing.

- **Selective Adapter Aggregation** (Sec. 3.3). To mitigate inter-silo adapter interference during adapter averaging, FedMosaic aggregates only adapters associated with the most relevant documents and least conflicting parameters. This is enabled by the document-specific masks from the multi-document adapters. For each query, silos re-rank their local documents and upload only the relevance scores and corresponding masks. The server then selects the globally most relevant masks while minimizing parameter conflicts, measured by overlap among mask positions. The corresponding adapters are requested, gated by the selected masks, and averaged using standard parameter aggregation. This yields a relevance- and conflict-aware aggregation that preserves locality and improves answer quality.

Workflow. We assume each silo has the same base LLM as the server and employs a homogeneous re-ranking model $\mathcal{M}_r$. These assumptions are standard in cross-silo federated learning, where participants typically have sufficient resources to fine-tune large models locally. Under this setting, FedMosaic operates in two stages.

- **Offline Stage**. Each silo clusters its local documents, trains a cluster-level adapter for each cluster, and learns a binary mask for every document. The resulting adapters and masks are stored locally as the parametric representation of the silo's corpus.
- **Online Stage**. When a query arrives, the server broadcasts it to all silos. Each silo performs local retrieval, applies the re-ranking model to compute relevance scores, and uploads the scores together with the masks of the retrieved documents. The server then selects top-ranked documents, requests their adapters from silos, aggregates the masked adapters, and composes the result with the base LLM to generate the final answer.

Algorithm 2 illustrates the overall workflow. In the *offline* stage, each silo first performs balanced clustering on its local document corpus and trains an adapter for each cluster (line 4-5), then train a mask for each document (lines 7). This stage is executed only once. In the *online* stage, the server broadcasts the user query to all silos. Each silo retrieves top-k documents and computes their relevance scores (lines 11-13). The server then selects documents based on the relevance scores and corresponding masks (lines 15-16), aggregates the associated LoRA adapters weighted by the scores, and generates the final answer a (line 18-19).

Algorithm 2: FedMosaic

Input: user query q, LLM server $\mathcal{G}$, data silos $\{\mathcal{S}_m\}_{m=1}^{M}$
Output: answer a

1 // Offline Stage
2 **for** *each silo* $\mathcal{S}_m$ **do**
3 // Clustered Adapter Assignment
4 Obtain clusters $\{C_m^1, C_m^2, \ldots, C_m^t\}$ via Eq. (3)
5 Train adapters $\{(A_m^i, B_m^i)\}_{i=1}^{t}$ per cluster via Eq. (2)
6 // Masked Adapter Sharing
7 Train mask per document via Eq. (7)
8 // Online Stage
9 // Local Document Re-Ranking
10 **for** *each silo* $\mathcal{S}_m$ **do**
11 Retrieve top-k documents $\{d_m^{r_1}, \cdots, d_m^{r_k}\}$ based on q
12 Compute relevance scores $\{s_m^{r_1}, \cdots, s_m^{r_k}\}$ via $\mathcal{M}_r$
13 Upload the relevance scores and the corresponding document masks $\{(s_m^{r_i}, M_m^{r_i})\}_{i=1}^{k}$ to the server
14 // Conflict-Aware Document Selection
15 Server selects k' documents via Eq. (11) and notifies silos
16 Silos upload corresponding LoRA adpters $\{(B_i, A_i)\}_{i=1}^{k'}$
17 // Masked Adapter Aggregation
18 Server aggregates LoRA adapters ΔW_{merge} via Eq. (13)
19 LLM generates answer $a \leftarrow \mathcal{G}(q, \Delta W_{\text{merge}})$
20 **return** a

3.2 Multi-Document Parametric Adapters

To reduce silo-side storage and silo–server communication while avoiding *intra-silo adapter interference*, FedMosaic replaces per-document adapters with *multi-document* adapters. The design has

two components: *(i) clustered adapter assignment*, which groups semantically similar documents and trains one adapter per cluster (Sec. 3.2.1); and *(ii) masked adapter sharing*, which learns document-specific binary masks to activate distinct parameter subsets within a shared adapter (Sec. 3.2.2). To our knowledge, this is the first *multi-document* parametric RAG scheme that exploits the sparsity of LoRA adapters.

3.2.1 Clustered Adapter Assignment. This module partitions a silo's corpus into balanced clusters of semantically related documents and trains one adapter per cluster.

Balanced Document Clustering. Each document is embedded into a vector representation, and semantic similarity is measured in the embedding space using *e.g.* Euclidean distance.

To control *intra-silo adapter interference*, we restrict the maximum number of documents assigned to each cluster. This is because grouping semantically related documents improves adapter sharing, but overly large clusters reintroduce interference.

In FedMosaic, we adopt the constrained k-means clustering algorithm [4] with an empirically chosen maximum cluster size (5 to 10, see Sec. 4.4.1). Consequently, each silo $\mathcal{S}_m$ obtains a balanced partition of its local corpus

$$C_m = \{C_m^1, C_m^2, \ldots, C_m^t\}, \tag{3}$$

where t is the number of clusters, and C_m^i represents the set of documents assigned to the i-th cluster.

Cluster-Level Parametric Adapters. After clustering, adapters are trained at the level of clusters rather than individual documents as in Sec. 2.2. For each cluster C_m^i in silo $\mathcal{S}_m$, we construct an augmented document set D_m^i following Eq. (1). This set is encoded into a LoRA adapter (A_m^i, B_m^i) via Eq. (2), where $B_m^i \in \mathbb{R}^{d\times r}$ and $A_m^i \in \mathbb{R}^{r\times d}$ define the low-rank update $\Delta W = B_m^i A_m^i$, with d representing the LLM layer size and $r \ll d$ the adapter rank. Thus, each silo $\mathcal{S}_m$ obtains t cluster-level adapters $\{(A_m^1, B_m^1), \ldots, (A_m^t, B_m^t)\}$, reducing the number of adapters compared to per-document training while retaining semantic coherence within clusters.

3.2.2 Masked Adapter Sharing. This module aims to learn *document-specific binary masks* that activate distinct subsets of a shared adapter, thereby reducing *intra-silo adapter interference*. The design is motivated by two observations: *(i)* LoRA adapters contain redundancy [40], and *(ii)* fine-tuning typically affects only subsets of model parameters, which differ across tasks [11, 44]. We hypothesize that document-specific knowledge can also be captured by distinct subsets of LoRA parameters. Accordingly, FedMosaic learns binary masks over a *frozen* cluster-level adapter so that each document activates a tailored subspace. These masks also assist in selective adapter aggregation (Sec. 3.3).

Adapter Mask Designation. Let (A_m^i, B_m^i) be the LoRA adapter for cluster C_m^i, which encodes documents $\{d_m^{i_1}, \ldots, d_m^{i_n}\}$. Applying a binary mask directly to the full update $\Delta W = B_m^i A_m^i \in \mathbb{R}^{d\times d}$ would require $O(d^2)$ mask parameters, which is both storage- and computation-intensive. Instead, we adopt a lightweight *row-wise* masking scheme on $B_m^i \in \mathbb{R}^{d\times r}$, requiring only $O(d)$ parameters.

For document $d_m^{i_j}$, we define a binary vector $M_m^{i_j} \in \{0,1\}^d$ and apply it row-wise to B_m^i:

$$\tilde{B}_m^{i_j} = M_m^{i_j} \circ B_m^i, \tag{4}$$

where $\circ$ denotes the Hadamard product with broadcasting across columns. To stabilize the magnitude of masked updates, we introduce a rescale factor:

$$\lambda_m^{i_j} = \frac{d}{\|M_m^{i_j}\|_1}, \tag{5}$$

and define the masked low-rank update as

$$\Delta\tilde{W}_m^{i_j} = \lambda_m^{i_j} \cdot \tilde{B}_m^{i_j} A_m^i. \tag{6}$$

This ensures stable masked updating while allowing each document to activate only a subset of adapter rows.

Adapter Mask Training. During training, the adapter parameters (A_m^i, B_m^i) remain frozen and only the masks are optimized. This preserves shared cluster-level knowledge while enabling each document to learn a sparse, specialized activation pattern.

To enable gradient-based optimization, the binary mask is relaxed via a sigmoid parameterization σ. Let $\hat{M}_m^{i_j} \in \mathbb{R}^d$ be trainable logits. During training we use $\sigma(\alpha\hat{M}_m^{i_j})$ with sharpening factor $\alpha > 0$, and impose ℓ_1 sparsity, The total loss function is defined as

$$\min_{\hat{M}_m^{i_j}} \mathcal{L}(\hat{M}_m^{i_j}; \boldsymbol{\theta}_F) = \mathcal{L}_{\text{next}}(\hat{M}_m^{i_j}; \boldsymbol{\theta}_F) + \lambda_{\ell_1} \cdot \left\| \sigma\left(\alpha \cdot \hat{M}_m^{i_j}\right) \right\|_1, \tag{7}$$

where $\boldsymbol{\theta}_F$ is the frozen base and adapter parameters, $\mathcal{L}_{\text{next}}$ is the next-token loss, and λ_{ℓ_1} balances task accuracy and mask sparsity.

After training, the masks are binarized via thresholding:

$$M_m^{i_j} = \mathbb{I}\left[\hat{M}_m^{i_j} > 0\right], \tag{8}$$

where $\mathbb{I}[\cdot]$ is the indicator function.

This procedure is applied independently for each document in the cluster, producing a set of sparse, document-specific masks $\left\{M_m^{i_j}\right\}_{j=1}^n$ that *(i)* mitigate intra-adapter interference by isolating document-specific subspaces, and *(ii)* provide conflict indicators for selective aggregation across silos.

To further reduce the storage overhead introduced by the binary mask parameters, we adopt a bit-packing scheme that encodes each d-dimensional binary mask vector $M_m^{i_j} \in \{0,1\}^d$ into $\lceil d/8 \rceil$ bytes by packing 8 bits per byte.

$$\mathbf{b}_i = \sum_{j=1}^{8} m_{8(i-1)+j} \cdot 2^{j-1}, \quad \text{for } i = 1, \ldots, \left\lceil \frac{d}{8} \right\rceil, \tag{9}$$

where $\mathbf{b}_i$ denotes the i-th packed byte and $m_{8(i-1)+j}$ denotes the $8(i-1)+j$-th bit in $M_m^{i_j}$. This reduces storage by 8× over naive byte-aligned `uint8` representations and supports encoding and decoding in $O(d)$ time via simple linear scans of the mask entries.

3.3 Selective Adapter Aggregation

As discussed in Sec. 2.3, naive aggregation of adapters from all silos introduces *inter-silo adapter interference*, since irrelevant documents introduce noise and overlapping parameters cause conflicts. To overcome this, FedMosaic introduces *selective adapter aggregation*, which aggregates only query-relevant documents while suppressing parameter conflicts. The process consists of three steps: *(i) local document re-ranking*, each silo re-ranks its local documents for the query and uploads only their relevance scores and masks (Sec. 3.3.1); *(ii) conflict-aware document selection*, the server selects a globally relevant subset under a conflict-aware criterion (Sec. 3.3.2); and *(iii) masked adapter aggregation*, the selected adapters are aggregated

in a mask-gated, relevance-weighted manner (Sec. 3.3.3). These designs address the unique challenges on accuracy when adapting parametric RAG to federated scenarios.

3.3.1 Local Document Re-Ranking. Upon receiving a query, the server broadcasts it to all silos. Each silo $\mathcal{S}_m$ retrieves k candidate documents $\{d_m^{r_1}, \ldots, d_m^{r_k}\}$ and applies a re-ranking model $\mathcal{M}_r$, which is the same across silos, to assign normalized relevance scores $\{s_m^{r_1}, \ldots, s_m^{r_k}\}$. For each candidate, the silo collects its associated mask $M_m^{r_i}$ over the cluster-level adapter and uploads only the pairs $\{(s_m^{r_i}, M_m^{r_i})\}_{i=1}^{k}$, while keeping the raw documents and LoRA adapters local.

3.3.2 Conflict-Aware Document Selection. The server collects the relevance scores and masks of all kM candidates and selects a subset S of k' documents that maximize relevance while minimizing parameter conflicts among their masks, given that relevance-based re-ranking alone is insufficient for RAG [18]. We measure conflicts between two masks M_i and M_j as their overlap:

$$\text{overlap}(M_i, M_j) = \frac{\langle M_i, M_j \rangle}{d}, \tag{10}$$

where d is the mask dimension.

The selection objective is formulated as

$$\max_{S \subseteq [n],\, |S|=k'} \left(\sum_{i \in S} \frac{s_i}{k'} - 2\lambda_{\text{ol}} \cdot \frac{\sum_{i,j \in S, i<j} \text{overlap}(M_i, M_j)}{k' \cdot (k'-1)} \right), \tag{11}$$

where $[n]$ is the index set of all candidate documents, s_i is the relevance score of document i, M_i is its mask, and $\lambda_{\text{ol}} > 0$ balances relevance against conflicts. We prove the above selection problem is NP-hard in Appendix A.

We solve Eq. (11) via a greedy algorithm. At each iteration, we select the candidate with the highest marginal gain

$$\Delta_i = s_i - \lambda_{\text{ol}} \cdot \frac{1}{|S|} \sum_{j \in S} \text{overlap}(M_i, M_j), \tag{12}$$

until k' documents are chosen. To avoid noisy selections, a score threshold τ is applied. Only candidates with $s_i > \tau$ are considered, and the procedure terminates early if none remain.

3.3.3 Masked Adapter Aggregation. After selection, the server requests only the necessary information from silos. For each chosen document $i \in S$, this includes its mask M_i and the associated cluster adapter (B_i, A_i). Let $w_i = s_i / \sum_{j \in S} s_j$ be normalized relevance weights. The aggregated LoRA adapter is then computed as

$$\Delta W_{\text{merge}} = \sum_{i \in S} w_i \, (M_i \circ B_i) A_i, \tag{13}$$

which is combined with the base LLM $\mathcal{G}$ to generate the final answer. This aggregation strategy ensures query relevance, suppresses parameter conflicts, and maintains locality of raw documents.

4 Experiments

4.1 Experimental Setup

Baselines. We compare FedMosaic with four categories of methods: local RAG (Standard RAG, CoTRAG [38], ReAct [43], Dargin [29]), in-context FedRAG (FRAG [1, 49], MKPQA [28], RAGRoute [13]), federated fine-tuning (FedIT [48], FLora [37]), and parameterized RAG (PRAG [30]). All baselines are adapted to the federated setting for fair comparison. Methods based on privacy-preserving prompt engineering are not included as baselines due to their severe accuracy degradation (see Sec. 4.3.3).

Datasets and Models. We experiment on HotpotQA (HQA) [42], 2WikiMultihopQA (2WQA) [15], PopQA (PQA) [22], and ComplexWebQuestions (CWQ) [31] with different question types (*e.g.* Bridge). By default, each dataset is subsampled to 300 Q&A instances and a document corpus is constructed following [30]. This corpus is then partitioned into silos based on underlying topics using a Dirichlet-based allocation strategy with $\alpha = 0.1$. We also conduct privacy evaluations on Enron Emails and WikiText datasets following [46, 47]. LLaMA3.2-1B-Instruct and LLaMA3-8B-Instruct are used as backbone LLMs.

Metrics. We report four metrics: (1) *Accuracy*: correctness of server LLM responses to user queries measured by F1 score; (2) *Privacy*: average success protection rate against targeted attack [46] and prefix attack [5] at server-side inference; (3) *Communication Efficiency*: average parameters transmitted per query from silo to server; (4) *Storage Overhead*: extra silo-side storage for parametric adapters.

4.2 Main Results

Table. 1 reports the overall F1 score of FedMosaic and baselines on four datasets with the LLaMA3.2-1b-instruct backbone. Local RAG methods show unstable performance, since each silo can only rely on its own documents without cross-silo knowledge. In-context FedRAG baselines attempt to integrate cross-silo information but violate the locality constraint and often introduce noisy documents, limiting their gains. FedIT and FLoRA achieve relatively high scores on some subsets (*e.g.* 0.4250 on 2WQA Bridge) and exhibit more stable performance overall, but require 10× higher training budgets. Naive federated parametric RAG performs well on certain subsets (*e.g.* 0.4805 on 2WQA Compare), but suffers severe drops on others (*e.g.* 0.2810 on CWQ), consistent with the challenges in Sec. 2.3.

In contrast, FedMosaic consistently achieves state-of-the-art performance across all datasets, demonstrating its effectiveness in mitigating both intra- and inter-silo adapter interference. Specifically, compared to the strongest baseline, FedMosaic improves the average F1 scores on HQA, 2WQA, and PQA by 10.57%, 13.08%, and 10.03%, respectively. These results highlight the robustness of FedMosaic and its ability to leverage distributed knowledge while preserving locality constraints.

4.3 Micro-Benchmarks

4.3.1 Privacy Evaluation. The experiment justifies the necessity of the locality constraint for FedRAG, and further quantifies privacy benefits of FedMosaic under the locality constraint versus in-context FedRAG. We consider two data extraction attacks: *targeted* and *prefix* [5, 46]. We construct 300 privacy-sensitive samples following [47], and then apply retrieval-data attacks to in-context FedRAG and training-data attacks to FedMosaic. The in-context FedRAG (*i.e.*, IC-FedRAG) adopts a minimal FRAG [1, 49] design, capturing key characteristics of MKPQA [28] and RAGRoute [13].

Table. 2 reports the attack success rates (lower is better) of FedMosaic and in-context FedRAG, showing that FedMosaic provides

Table 1: Overall accuracy measured by F1 score for FedMosaic and four types of baseline methods.

Type	Method	HotpotQA		2WikiMultihopQA				PopQA	CWQ
		Bridge	Compare	Bridge	Compare	Inf.	Compose	Total	Total
Local	Standard RAG	0.1474	0.3602	0.2838	0.2837	0.1549	0.0748	0.1366	0.2804
	CoTRAG [38]	0.0808	0.2632	0.3821	0.3444	0.1380	0.0377	0.0444	0.2378
	React [43]	0.1731	0.3103	0.3108	0.2608	0.1528	0.0590	0.1135	0.2443
	Dargin [29]	0.1257	0.3540	0.3271	0.3786	0.1473	0.0766	0.0895	0.3662
FedRAG	FRAG [1, 49]	0.1317	0.3390	0.3357	0.2964	0.1527	0.0265	0.1207	0.1946
	MKPQA [28]	0.1562	0.2757	0.2831	0.2679	0.2112	0.0510	0.1329	0.2008
	RAGRoute [13]	0.1137	0.2825	0.2239	0.2472	0.1351	0.0293	0.0380	0.1767
FedFT	FedIT [48]	0.1107	0.4188	0.4250	0.4377	0.1996	0.0692	0.2152	0.3687
	FLoRA [37]	0.1030	0.4259	0.3625	0.4038	0.1864	0.0770	0.1917	0.3497
Param.	PRAG [30]	0.0983	0.4161	0.3875	0.4805	0.2138	0.0462	0.0759	0.2810
Ours	FedMosaic	**0.1980**	**0.4547**	**0.4453**	**0.5275**	**0.2474**	**0.0940**	**0.2368**	**0.3841**

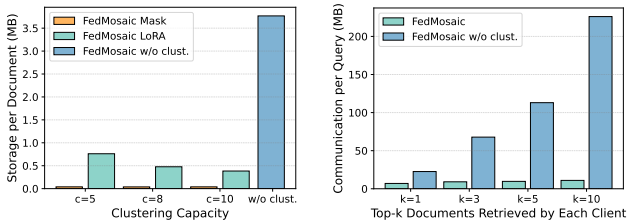
Table 2: Privacy evaluation over target and prefix attack.

Attack	Method	Target Prompts	Target Info.	Repeat Prompts	Repeat Length
Target	IC-FedRAG	138.93	102.20	131.33	21.34
	FedMosaic	**0**	**0**	**0**	**0**
Prefix	IC-FedRAG	52,86	46.66	55.93	23.58
	FedMosaic	**44.00**	**29.53**	**22.86**	**22.22**

Table 3: LLaMA3-8B compared with competitive baselines.

Type	Method	2WikiMultiHopQA			
		Bridge	Compare	Inf.	Compose
Local	ReAct	0.3432	0.4022	0.2093	0.0746
FedRAG	MKPQA	0.2238	0.4011	0.2797	0.0932
FedFT	FedIT	0.4995	0.6105	0.2677	0.1311
FedFT	FLoRA	0.4701	0.5714	0.2355	0.1138
Param.	PRAG	0.1812	0.2527	0.0829	0.0271
Ours	FedMosaic	**0.5207**	**0.6237**	**0.2859**	**0.1584**

stronger resistance to data extraction attacks. Under targeted attacks, FedMosaic is almost immune (0% across all metrics) since the absence of explicit context makes it difficult to trigger adapter-encoded knowledge, whereas in-context FedRAG prompts are more easily repeated by the LLM. Under prefix attacks, FedMosaic also outperforms in-context FedRAG (*e.g.* a 59% reduction in repeated prompts and a 36.7% reduction in target information), benefiting from training on rewritten data (see Eq. (1)), where sensitive content can be filtered using simple prompts (*e.g.* Ensure the revision has no emails). Furthermore, successful attacks require knowledge of the actual training data, which further increases attack difficulty and demonstrates the higher degree of privacy achieved by FedMosaic.



(a) Per-Document Storage **(b) Per-Query Communication**

Figure 4: Storage and communication overhead.

4.3.2 Performance with Larger Backbones. To evaluate the scalability of FedMosaic, we extend the backbone to LLaMA3-8B-Instruct. For a comprehensive comparison, we select the most competitive baselines reported in Sec. 4.2 and conduct experiments on the 2WQA dataset, as it contains the most diverse set of Q&A types.

Table. 3 reports the F1 scores of FedMosaic and the competitive baselines. We observe that PRAG exhibits notable performance degradation due to accumulated noise and parameter conflicts. In contrast, FedMosaic consistently maintains state-of-the-art performance, demonstrating strong scalability to larger model sizes. For example, on the Bridge and Compose datasets, FedMosaic outperforms the strongest baselines by 4.2% and 20.8%, respectively.

4.3.3 Compared to Privacy-Preserving Prompt Engineering Methods. Privacy-preserving prompt engineering seeks to protect sensitive information by introducing noise through anonymization or differential privacy. Representative approaches include differential privacy-based methods (DP-Prompt [34], AUGPR [39]) and anonymization-based methods (Sage [47]).

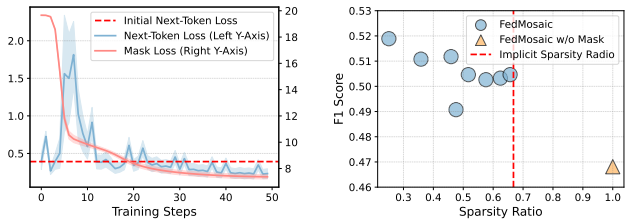
Table. 4 reports the F1 scores of these methods, together with their performance drops relative to FedMosaic. All methods show severe accuracy degradation, especially differential privacy approaches (*e.g.* AUGPR), which perturb the global distribution by introducing noise that obscures document-specific information, and, even when applied document-wise (*e.g.* DP-Prompt), can damage fine-grained knowledge crucial for accurate RAG. Similarly,

Table 4: Performance of privacy-preserving prompt engineering methods.

Method	HotpotQA		2WikiMultihopQA				PopQA	CWQ
	Bridge	Compare	Bridge	Compare	Inf.	Compose	Total	Total
DP_Prompt	$0.0228_{\downarrow 88.5\%}$	$0.2179_{\downarrow 52.1\%}$	$0.0785_{\downarrow 82.4\%}$	$0.1305_{\downarrow 75.3\%}$	$0.0385_{\downarrow 84.4\%}$	$0.0114_{\downarrow 87.9\%}$	$0.0060_{\downarrow 97.5\%}$	$0.0778_{\downarrow 79.7\%}$
Sage (Attr)	$0.0791_{\downarrow 60.1\%}$	$0.3286_{\downarrow 27.7\%}$	$0.1808_{\downarrow 59.4\%}$	$0.2218_{\downarrow 58.0\%}$	$0.1318_{\downarrow 46.7\%}$	$0.0222_{\downarrow 76.4\%}$	$0.0676_{\downarrow 71.5\%}$	$0.1217_{\downarrow 68.3\%}$
Sage (Agent)	$0.0923_{\downarrow 53.4\%}$	$0.3314_{\downarrow 27.1\%}$	$0.2898_{\downarrow 34.9\%}$	$0.2645_{\downarrow 49.9\%}$	$0.1343_{\downarrow 45.7\%}$	$0.0231_{\downarrow 75.4\%}$	$0.0560_{\downarrow 76.4\%}$	$0.1485_{\downarrow 61.3\%}$
AUGPE	$0.0079_{\downarrow 96.0\%}$	$0.0514_{\downarrow 88.7\%}$	$0.0153_{\downarrow 96.6\%}$	$0.0283_{\downarrow 94.6\%}$	$0.0225_{\downarrow 90.9\%}$	$0.0032_{\downarrow 96.6\%}$	$0.0004_{\downarrow 99.8\%}$	$0.0247_{\downarrow 93.6\%}$

Table 5: Accuracy improvement and robustness of selective aggregation under different selection k.

Dataset	Type	w/o Sel.	k=1	k=3	k=5	k=10	k=15	k=30
2WQA	Bridge	0.4224	$0.4552_{\uparrow 7.77\%}$	$0.4447_{\uparrow 5.28\%}$	$0.4492_{\uparrow 6.34\%}$	$0.4525_{\uparrow 7.13\%}$	$0.4525_{\uparrow 7.13\%}$	$0.4525_{\uparrow 7.13\%}$
	Compare	0.5001	$0.4940_{\downarrow 1.22\%}$	$0.5108_{\uparrow 2.14\%}$	$0.5142_{\uparrow 2.82\%}$	$0.5142_{\uparrow 2.82\%}$	$0.5142_{\uparrow 2.82\%}$	$0.5142_{\uparrow 2.82\%}$
	Inf.	0.2137	$0.2316_{\uparrow 8.38\%}$	$0.2581_{\uparrow 20.7\%}$	$0.2455_{\uparrow 14.8\%}$	$0.2445_{\uparrow 14.4\%}$	$0.2422_{\uparrow 13.3\%}$	$0.2422_{\uparrow 13.3\%}$
	Compose	0.0862	$0.0750_{\downarrow 12.9\%}$	$0.0869_{\uparrow 0.81\%}$	$0.0953_{\uparrow 10.5\%}$	$0.0953_{\uparrow 10.5\%}$	$0.0953_{\uparrow 10.5\%}$	$0.0953_{\uparrow 10.5\%}$



(a) Training Loss **(b) Mask to Accuracy**

Figure 5: Impact of document mask.

anonymization-based methods such as Sage also lose substantial information, with an average 53% degradation, indicating that key evidence for Q&A is often compromised during anonymizatio. Overall, the poor utility of these approaches in RAG tasks makes them unsuitable as primary baselines in our evaluation.

4.4 Ablation Studies

We conduct a thorough ablation study to isolate and verify the contribution of each component in FedMosaic. Specifically, we analyze *multi-document parametric adapters* in terms of overhead reduction (Sec. 4.4.1) and mask effectiveness (Sec. 4.4.2), while *selective adapter aggregation* is evaluated for the impact of the selection k (Sec. 4.4.3) and the retrieval k (Sec. 4.4.4) on accuracy.

4.4.1 Overhead Reduction by Clustering. Fig. 4 compares the storage and communication overhead of FedMosaic with the w/o clustering variant. We choose $C \in \{5, 8, 10\}$ and $k \in \{1, 3, 5, 10\}$ for silo retrieval in our experiments. Fig. 4a shows that adapter size inversely proportional decreases with clustering (*e.g.* 12.67% of no-clustering at $c = 8$), while per-document masks stay constant at $\sim 1.03\%$. The total storage is reduced to 11.23% of the w/o clustering variant at $c = 10$. Fig. 4b shows that per-query communication increases linearly in the w/o clustering variant, whereas clustering keeps it stable; at $k = 10$, the cost is reduced to 4.86% of the variant.

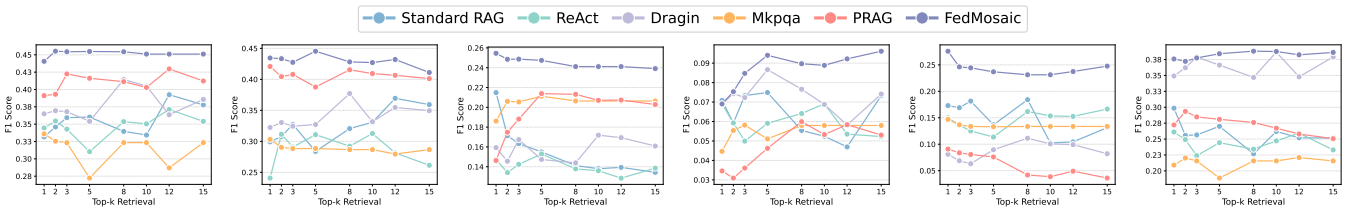
4.4.2 Contributions of Document-Specific Mask. Fig. 5 demonstrates the role of the document-specific mask in alleviating *intra-silo adapter interference* and its effect on model accuracy. We select the number of mask training epochs from $\{3, 5, 10, 20\}$ with varying learning rate. Fig. 5a shows training curves of next-token and mask losses, where both prediction accuracy and mask sparsity improve simultaneously, supporting the hypothesis in Sec. 3.2.2. Fig. 5b further evaluates accuracy under different sparsity settings, where FedMosaic consistently outperforms the w/o mask variant. Notably, setting $\lambda_{\ell_1} = 0$ in Eq. (7) drives mask learning solely via next-token loss, yielding implicit sparsity and indicating that document-specific knowledge can be captured by distinct subsets of LoRA parameters.

4.4.3 Effectiveness of Selective Aggregation. Table. 5 presents the effectiveness of selective aggregation on the 2WQA dataset, which is selected for its diversity, under different values of selection k. FedMosaic outperforms its w/o selection variant as k increases (*e.g.* 20.7% higher in Inf.), demonstrating its ability to filter irrelevant documents and mitigate parameter conflicts. Moreover, its performance stabilizes with larger k, highlighting strong robustness.

4.4.4 Impact of Top-k Retrieval. Fig. 6 compares FedMosaic with the most competitive FedRAG baseline across all types of datasets under different top-k retrieval settings. Other methods show fluctuating performance, indicating sensitivity to retrieval noise. With selective aggregation, FedMosaic achieves consistently higher and more stable accuracy, *e.g.* achieving up to a 10.17% average improvement on HQA Compare over the strongest baseline.

5 Related Work

Retrieval-Augmented Generation (RAG). RAG [12] enhances LLMs with external knowledge, reducing hallucinations and compensating for outdated parametric memory. The mainstream is *in-context RAG* [10, 14, 19], and it can be improved through retrieval optimization [20, 24, 25, 41], retriever-LLM alignment [7, 27, 36, 45], and multi-round reasoning [9, 29, 43]. However, in-context RAG is unfit for the federated settings by design, since it requires transmitting verbatim documents to the server, which violates the locality constraint.



(a) HQA Compare (b) 2WQA Bridge (c) 2WQA Inference (d) 2WQA Compose (e) PQA Total (f) CWQ Total

Figure 6: Performance of federated RAG under varying top-k retrieval settings.

Parametric RAG [30] offers an alternative by compiling knowledge into model parameters rather than context tokens through training. DyPRAG [32] trains a parameter translator model to convert documents into parametric knowledge. By encoding knowledge in reusable parameters rather than prompts, parametric RAG allows documents to remain local while uploading only adapterss, making it suited to federated scenarios. Our work is built upon parametric RAG, and focuses on the unique efficiency and accuracy challenges when adapting it to federated environments.

Federated Retrieval-Augmented Generation (FedRAG). FedRAG extends RAG to settings where knowledge is distributed across silos that cannot share raw documents. Recent efforts have explored various strategies for federated search, yet rely on *in-context* RAG. For example, FeB4RAG [35] provides a benchmark for federated search pipelines, MKPQA [28] enables probabilistic cross-domain search, and RAGRoute [13] improves efficiency via query routing. Other variants target aggregation or search security. For example, C-FedRAG [1] prevent the leakage of raw documents during aggregation and FRAG [49] leverage encrypted similarity search. Despite these advances, existing frameworks still transmit raw documents across silos, violating locality constraints.

Other efforts adopt privacy-preserving prompt engineering by injecting noise into documents. For example, DP-Prompt [34] and AUGPE [39] adopt differential privacy strategies. Sage [47] adopt anonymization strategies to achieve data desensitization. However, these approaches suffer severe accuracy degradation in FedRAG settings, limiting their applicability.

Another relevant solution to FedRAG is federated fine-tuning, as explored in FedIT [48] and FLora [37]. Compared to FedRAG, federated fine-tuning is less flexible because new documents demand costly retraining and it cannot selectively activate relevant silos.

In contrast, our FedMosaic introduces the first *federated parametric RAG* framework. By encoding documents into local adapters and composing them across silos at inference, FedMosaic adheres to the locality constraint while providing high flexibility. Its optimizations ensure low storage and communication overhead as well as high accuracy, making it a practical and scalable solution for federated knowledge-intensive generation.

6 Conclusion

In this work, we explore federated RAG where a central LLM collaborates with distributed knowledge silos without sharing their raw documents. We propose FedMosaic, the first federated RAG framework built on parametric adapters. By clustering semantically related documents into multi-document adapters with document-specific masks, FedMosaic drastically reduces storage and communication overhead while maintaining per-document specificity. Furthermore, its selective adapter aggregation mechanism ensures that only relevance-aligned, non-conflicting adapters are combined, mitigating destructive aggregation across silos. Extensive experiments on four datasets show that FedMosaic consistently outperforms state-of-the-art methods across four categories including local RAG, in-context FedRAG, federated fine-tuning, and parametric RAG, while enforcing locality constraints and significantly reducing storage and communication overhead. We envision FedMosaic as a foundation for scalable, privacy-preserving knowledge integration, and an important step toward deploying RAG systems in real-world distributed information ecosystems.

Acknowledgments

This work was partially supported by National Science Foundation of China (NSFC) (Grant Nos. 62425202, 62336003), the Beijing Natural Science Foundation (Z230001), and the State Key Laboratory of Complex & Critical Software Environment (SKLCCSE). Zimu Zhou's research is partially supported by the RGC of Hong Kong SAR, China (Project No. CityU 11206425).

A Proof of NP-Hardness

We show that the conflict-aware document selection optimization problem is NP-hard by through a reduction from CLIQUE problem [2]. The proof is given in two steps.

First, We can formally cast the conflict-aware document selection problem as a weighted subgraph selection problem over a complete graph, defined as follows.

Definition 1 (Weighted Subgraph Selection Problem). *Let $G = (V, E)$ be the complete graph on n vertices. Each vertex $v \in V$ is associated with a positive vertex weight $a_v > 0$, and each unordered pair $\{u, v\} \subseteq V$ is assigned an edge weight $b_{uv} \in \mathbb{R}$. For any subset $S \subseteq V$ with cardinality $|S| = k$, the total weight of S is defined as*

$$W(S) = \sum_{v \in S} a_v - \sum_{\{u,v\} \subseteq S} b_{uv}. \tag{14}$$

The Weighted Subgraph Selection Problem asks: given a weighted complete graph with vertex weights $\{a_v\}_{v \in V}$, edge weights $\{b_{uv}\}_{\{u,v\} \subseteq V}$, and an integer k, find a subset $S \subseteq V$ of size k that maximizes $W(S)$.

Then, we reduce from the classical CLIQUE problem, which is known to be NP-complete [2]. Given a graph $G' = (V', E')$ and an integer q, the CLIQUE problem asks whether there exists a

subset $C \subseteq V'$ with $|C| = q$ such that every pair of vertices in C is connected by an edge in E'. From an instance (G', q) we construct an instance of the Weighted Subgraph Selection Problem as follows:

- The vertex set of the new instance is $V := V'$.
- For each $v \in V$, set the vertex weight $a_v := 1$.
- For each unordered pair $\{u, v\} \subseteq V$, define the edge weight

$$b_{uv} = \begin{cases} 0, & \{u, v\} \in E', \\ B, & \{u, v\} \notin E', \end{cases} \tag{15}$$

  where B is a sufficiently large constant (e.g., $B > q$).
- The target subset size is set to $k := q$.

Consider any subset $S \subseteq V$ with $|S| = q$. Its weight is $W(S) = q - B \cdot t(S)$, where $t(S)$ denotes the number of non-edges induced by S, i.e.,

$$t(S) = \left|\{\{u, v\} \subseteq S : \{u, v\} \notin E'\}\right|. \tag{16}$$

By construction, if S is a clique of size q, then $t(S) = 0$ and $W(S) = q$. Otherwise, $t(S) \geq 1$ and thus $W(S) \leq q - B$. Since $B > q$, the global maximum of $W(S)$ equals q if and only if G' contains a clique of size q. Hence, deciding whether G' admits a clique of size q can be answered by solving the weighted subgraph selection problem on the constructed instance. This establishes a polynomial-time reduction

$$\text{CLIQUE} \leq_p \text{Weighted Subgraph Selection}. \tag{17}$$

Since CLIQUE is NP-complete, it follows immediately that the weighted subgraph selection problem, and consequently the conflict-aware document selection problem, is NP-hard.

B Supplementary Experiment

B.1 Experimental Settings

B.1.1 Experiment Configuration. We conduct our experiments on a machine equipped with an Intel Xeon Gold 6230R CPU and four NVIDIA A100 GPUs, each providing 40 GB of memory for computation. The fundamental experimental configurations of FedMosaic adopt mask training for 10 epochs with learning rate 3×10^{-1}, $\lambda_{l1} = 10^{-4}$, $\lambda_{ol} = 0.8$, $C \in \{5, 8\}$, and LoRA training for 3 or 5 epochs. The setups for document augmentation and adapter training follow [30].

B.1.2 Hyperparameters. We set the default retrieval to the top-5 documents per silo for each query. Other hyperparameters of the baselines are detailed below.

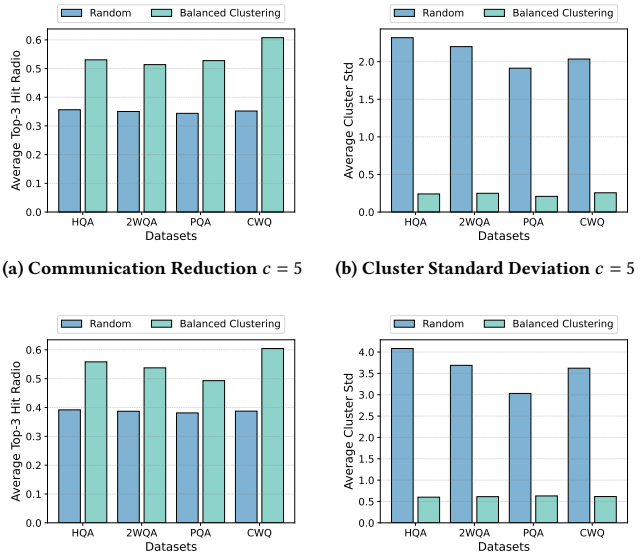
- **CoTRAG** [38]: We set the few-shot number to 11.
- **ReAct** [43]: We set the maximum attempts per step to 10.
- **MKPQA** [28]: We set the threshold to $\tau = 400$.
- **RAGRoute** [13]: We train the router on 300 generated samples for 150 epochs.
- **FedIT** [48], **FLora** [37]: We perform 10 local epochs per silo and 3 rounds of server aggregation.
- **Dragin** [29], **PRAG** [30]: We use the default hyperparameters from the original papers.

For in-context integration methods, we design distinct prompts for the silo side and the server side, since silos only upload candidate answers to the server.

B.2 Additional Experiments

B.2.1 Impact of Balanced Document Clustering. Fig. 7 evaluates balanced document clustering on four datasets at $c = 5, 15$, reporting the average hit ratio (*i.e.*, the proportion of top-3 documents in the same cluster) and the cluster size deviation. Compared with random clustering, the balanced strategy more effectively controls cluster sizes and ensures stability across different datasets, *e.g.* reducing the deviation by 85.2% on HQA at $c = 15$. It also consistently groups semantically related documents into the same adapter, thereby increasing the likelihood that retrieved candidates share an adapter and significantly lowering communication cost, *e.g.* the average top-3 hit ratio improves from 0.352 to 0.607 in CWQ at $c = 5$.

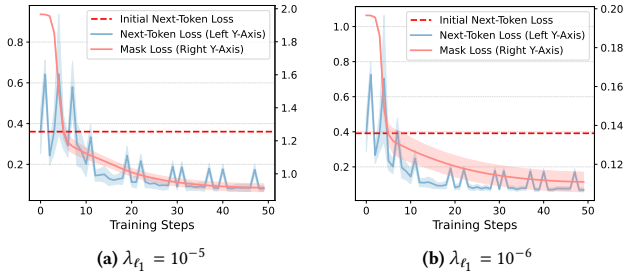
B.2.2 Impact of balance coefficient λ_{ℓ_1}. Fig. 8 reports the mask training loss curves obtained under different settings of λ_{ℓ_1}. Across varying λ_{ℓ_1}, both the next-token loss and the mask loss exhibit substantial reductions throughout the training process. Meanwhile, larger λ_{ℓ_1} values yield sparser masks, effectively enabling a trade-off between training accuracy and mask sparsity.



(a) Communication Reduction $c = 5$ (b) Cluster Standard Deviation $c = 5$

(c) Communication Reduction $c = 15$ (d) Cluster Standard Deviation $c = 15$

Figure 7: Balanced and random document clustering.



(a) $\lambda_{\ell_1} = 10^{-5}$ (b) $\lambda_{\ell_1} = 10^{-6}$

Figure 8: Mask training loss under varying λ_{ℓ_1}.

References

[1] Parker Addison, Minh-Tuan H Nguyen, Tomislav Medan, Jinali Shah, Mohammad T Manzari, Brendan McElrone, Laksh Lalwani, Aboli More, Smita Sharma, Holger R Roth, et al. 2024. C-fedrag: A confidential federated retrieval-augmented generation system. *arXiv preprint arXiv:2412.13163* (2024).

[2] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. 1999. The maximum clique problem. In *Handbook of Combinatorial Optimization: Supplement Volume A*. Springer, 1–74.

[3] Kym M Boycott and Roberto Giugliani. 2025. The RDI–Lancet Commission on Rare Diseases: improving visibility to address health-care disparities for 400 million people. *The Lancet* 405, 10479 (2025), 605–607.

[4] Paul S Bradley, Kristin P Bennett, and Ayhan Demiriz. 2000. Constrained k-means clustering. *Microsoft Research, Redmond* 20 (2000).

[5] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *ICLR*.

[6] Zhe Chen, Yusheng Liao, Shuyang Jiang, Pingjie Wang, YiQiu Guo, Yanfeng Wang, and Yu Wang. 2025. Towards omni-RAG: Comprehensive retrieval-augmented generation for large language models in medical applications. In *ACL*. 15285–15309.

[7] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self-memory. *NeurIPS* 36 (2023), 43780–43799.

[8] U.S. Congress. 1996. Health Insurance Portability and Accountability Act of 1996. https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf. Accessed: 2025-9-19.

[9] Chenxu Cui, Haihui Fan, Jinchao Zhang, Lin Shen, Bo Li, and Weiping Wang. 2025. CIRAG: Retrieval-Augmented Language Model with Collective Intelligence. In *SIGIR*. 1316–1326.

[10] Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Ji-Rong Wen, and Zhicheng Dou. 2025. Understand what LLM needs: Dual preference alignment for retrieval-augmented generation. In *WWW*. 4206–4225.

[11] Guodong Du, Zitao Fang, Jing Li, Junlin Li, Runhua Jiang, Shuyang Yu, Yifei Guo, Yangneng Chen, Sim Kuan Goh, Ho-Kin Tang, Daojing He, Honghai Liu, and Min Zhang. 2025. Neural Parameter Search for Slimmer Fine-Tuned Models and Better Transfer. In *ACL*. 32668–32687.

[12] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *SIGKDD*. 6491–6501.

[13] Rachid Guerraoui, Anne-Marie Kermarrec, Diana Petrescu, Rafael Pires, Mathis Randl, and Martijn de Vos. 2025. Efficient federated search for retrieval-augmented generation. In *Proceedings of the 5th Workshop on Machine Learning and Systems*. 74–81.

[14] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *ICML*. 3929–3938.

[15] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *COLING*. 6609–6625.

[16] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

[17] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024. Emr-merging: Tuning-free high-performance model merging. *NeurIPS* 37 (2024), 122741–122769.

[18] Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. 2025. Shifting from Ranking to Set Selection for Retrieval Augmented Generation. In *ACL*. 17606–17619.

[19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS* 33 (2020), 9459–9474.

[20] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *WWW*. 3497–3508.

[21] Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024. Twin-merging: Dynamic integration of modular expertise in model merging. *NeurIPS* 37 (2024), 78905–78935.

[22] 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 *ACL*. 9802–9822.

[23] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2023. Task arithmetic in the tangent space: Improved editing of pre-trained models. *NeurIPS* 36 (2023), 66727–66754.

[24] Tao Ouyang, Guihang Hong, Kongyange Zhao, Zhi Zhou, Weigang Wu, Zhaobiao Lv, and Xu Chen. 2025. AdaRAG: Adaptive Optimization for Retrieval Augmented Generation with Multilevel Retrievers at the Edge. In *INFOCOM*. IEEE, 1–10.

[25] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3 (2009), 333–389.

[26] Tianyi Shen, Yuxi Li, Yanlin Cao, Xin Du, Xinru Wang, Yajuan Zhang, and Yi Zhang. 2025. Rapid deployment of large language model DeepSeek in Chinese hospitals demands a regulatory response. *Nature Medicine* (2025), 1–6.

[27] Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval Augmented Generation with Collaborative Filtering for Personalized Text Generation. In *SIGIR*.

[28] Parshin Shojaee, Sai Sree Harsha, Dan Luo, Akash Maharaj, Tong Yu, and Yunyao Li. 2025. Federated retrieval augmented generation for multi-product question answering. In *COLING*, Vol. Industry Track. 387–397.

[29] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *ACL*. 12991–13013.

[30] Weihang Su, Yichen Tang, Qingyao Ai, Junxi Yan, Changyue Wang, Hongning Wang, Ziyi Ye, Yujia Zhou, and Yiqun Liu. 2025. Parametric retrieval augmented generation. In *SIGIR*. 1240–1250.

[31] Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL*. 641–651.

[32] 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* (2025).

[33] European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng. Accessed: 2025-9-19.

[34] Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally differentially private document generation using zero shot prompting. In *Findings of EMNLP*.

[35] Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. 2024. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. In *SIGIR*. 763–773.

[36] Yuhao Wang, Ruiyang Ren, Yucheng Wang, Wayne Xin Zhao, Jing Liu, Hua Wu, and Haifeng Wang. 2025. Unveiling Knowledge Utilization Mechanisms in LLM-based Retrieval-Augmented Generation. In *SIGIR*.

[37] Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. 2024. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. *NeurIPS* 37 (2024), 22513–22533.

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS* 35 (2022), 24824–24837.

[39] Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, et al. 2024. Differentially private synthetic data via foundation model apis 2: Text. In *ICML*.

[40] Jing Xu and Jingzhao Zhang. 2024. Random masking finds winning tickets for parameter efficient fine-tuning. In *ICML*. 55501–55524.

[41] Shicheng Xu, Liang Pang, Jun Xu, Huawei Shen, and Xueqi Cheng. 2024. List-aware reranking-truncation joint model for search and retrieval-augmented generation. In *WWW*. 1330–1340.

[42] 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 *EMNLP*. 2369–2380.

[43] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, and et al. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

[44] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *ICML*.

[45] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *ACL*. 2421–2436.

[46] Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, et al. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (RAG). In *Findings of ACL*. 4505–4524.

[47] Shenglai Zeng, Jiankun Zhang, Pengfei He, Jie Ren, Tianqi Zheng, Hanqing Lu, Han Xu, Hui Liu, Yue Xing, and Jiliang Tang. 2024. Mitigating the privacy issues in retrieval-augmented generation (rag) via pure synthetic data. *arXiv preprint arXiv:2406.14773* (2024).

[48] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. 2024. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP*. 6915–6919.

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

[50] Xuejiao Zhao, Siyan Liu, Su-Yin Yang, and Chunyan Miao. 2025. Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot. In *WWW*. 4442–4457.