Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# A Trust-Weighted Hybrid Federated Fine-Tuning Architecture for Retrieval-Augmented Large Language Models

**Parsa Abdi**

par30boy.warrior@gmail.com

Department of Computer science,University of IAU, Tehran, IRAN

**Research Article**

**Keywords:** Federated Learning , Retrieval-Augmented Generation (RAG),Large Language Models (LLMs), Trust-Aware Aggregation, Privacy-Preserving NLP, Semantic Drift Mitigation, Explainable AI, Client-Specific Personalization

**Posted Date:** March 12th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-9093774/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** The authors declare no competing interests.

# A Trust-Weighted Hybrid Federated Fine-Tuning Architecture for Retrieval-Augmented Large Language Models

Parsa Abdi*[a]

[a]*Department of Computer science,University of IAU, Tehran, IRAN.*

**Abstract**

Federated fine-tuning for retrieval-augmented language models enables collaboration across data silos without sharing raw corpora, but it is vulnerable to unreliable clients whose retrieval corpora degrade retrieval behavior and, consequently, the quality of augmented inputs. This paper presents a label-free trust-weighted aggregation framework that estimates client reliability from retrieval-consistency signals and uses the resulting trust scores to control each client's influence on the global update. For a fixed probe prompt set, each client constructs a retrieval-distribution histogram from top-$K$ retrieved items, and Jensen–Shannon divergence to a round-wise global reference histogram quantifies retrieval deviation. An exponential mapping and exponential moving average smoothing produce stable trust trajectories over federated rounds, and normalized trust scores define aggregation weights with an explicit sum-to-one constraint. Experiments with four clients under a controlled corpus-poisoning protocol show that benign clients maintain consistently high trust while the poisoned client converges to a lower trust regime and is slightly but consistently down-weighted. All reported curves and summary statistics are derived from logged artifacts and deterministic scripts without manual post-processing, supporting full reproducibility of the trust and weighting dynamics.

*Keywords:* Federated Learning, Retrieval-Augmented Generation (RAG), Large Language Models (LLMs), Trust-Aware Aggregation, Privacy-Preserving NLP, Semantic Drift Mitigation, Explainable AI, Client-Specific Personalization

## 1. Introduction

In recent years, the reliance on large language models (LLMs) for natural language generation has created serious dilemmas and ethical considerations around data sovereignty, privacy, and context [4] . Federated learning (FL) has emerged as one potential remedy through the ability to update the model stage in a decentralized manner while avoiding centralized access to user data. However, with generative models the challenges of FL create tension between contextual relevance and privacy-preserving design [6] [1]. Many LLMs require a contextual input specific to a task or domain a priori to generating useful deliverables. However, FL prohibits pooling knowledge and completely rules out knowledge transfer between clients. Hence, a systemic boundary is created wherein languages models trained in isolation will struggle to yield semantically accurate, factual, or coherent outputs in heterogeneous data when distributional drift is involved. This leaves a considerable bottleneck in FL-centric NLP, where there is much less focus on high-quality generation, as well as privacy barriers [5]. One avenue to alleviate the above issue is to have LLMs paired with retrieval augmented generation (RAG), in that the model grounds its output with documents retrieved from a knowledge base [3] [2]. RAG has shown significant advantages in centralized structures, but uncertainty will arise in a federated context. For example, if retrieval is done in an uncollaborative or privacy-naive manner., through centralized corpora, or public indexing exposure; this is outwardly contradictory

to the FL aims of confidentiality. Also, traditional RAG will ignore inter client heterogeneity leading to fragile generalisation across domains. The retrieved context, without retrieval contextualization or alignment with local distributions will be irrelevant or unqualified and this leads to semantic drift and producing outputs that look syntaxically plausible but are not valid semantically and this is an unacceptable risk in highly regulated or high-stakes contexts. Besides the retrieval process challenges, the aggregation of all clients' model updates presents another unresolved vulnerability.
Conventional FL assumes trust for all parties around the table during aggregation, and it neglects heterogeneous model quality, and heterogeneity within local data trustworthiness [14]. In real deployments, clients can have poisoned data, biased corpora, or inconsistent fine-tuning. In the absence of a trust-aware mechanism, those contributions could also pollute the global model, and adversely affect both fairness and stability [15]. Furthermore, the important consistency constraints. federated fine-tuning with multiple clients, will yield semantic drift, which means the behavior of the client models can diverge in the latent space and suppress true convergence potential and model explainability. There is still some minor robustness enhancements, but it needs to generalized against a number of adversarial, skewed, or low-resource settings prefiguring real federated generation pipelines for downstream tasks requiring responsibilization and accountability [16].

*Scope of evaluation.* This study focuses on the stability and auditability of trust estimation and trust-weighted aggregation in a label-free federated RAG setting. Since the proposed trust signal is defined from retrieval-consistency rather than ground-truth answers, the evaluation emphasizes trust trajectories, normalized aggregation weights, and their behavior under controlled corpus poisoning. Task-level generation metrics and benchmark comparisons are left to future work as an extension of the present reproducible trust and aggregation analysis.

*Scope and reproducibility.* This work evaluates the stability and auditability of retrieval-consistency trust estimation and its induced trust-weighted aggregation in federated RAG. In particular, the reported evidence is limited to (i) trust trajectories computed from retrieval-distribution divergence under a fixed probe set, and (ii) normalized aggregation weights that satisfy $\sum_{i=1}^{M} w_i^{(t)} = 1$ at every round. We do not benchmark task-level generation quality or claim improvements in answer accuracy in this manuscript; local training is treated as a standard client-side update mechanism that defines the federated setting in which retrieval-consistency trust is computed. Accordingly, reproducibility in this study refers to re-generating all reported trust/weight figures and tables from released logs and deterministic scripts, rather than reproducing end-to-end task performance.

## 2. Literature Review

Yeonchan et al., (2022) proposes an innovative retrieval augmented generation (RAG) model aimed at enhancing knowledge grounded conversations (KGCs) in real-world dialogue contexts that are dynamic in nature. Traditional RAG models focus on one document or short period, in contrast this study presents hybrid RAG method that jointly considers the main topic of the conversation and the local context. The architecture of this model contains an external topic keyword extractor (TextRank), a reranker with dual-match, and a data-weighting strategy to address ground-truth-free training with noisy data. This model improves knowledge in generated responses by aligning them with relevant documents while attempting to maintain the user's topic intent which guided the dialogue. The automatic as well as human evaluations improve several metrics in terms of grounding, diversity, and relevance. It demonstrate a major contribution of a topic-aware dual-matching mechanism and groundness-based training approach yields significant improvement in knowledge alignment. The limitations to the study

include the foundational structure of topic-document alignment strategies to be defined by the researcher, the limited scalable design with static use of TextRank keyword extraction, and the presumption that the system will have access to a body of external documents that are comprehensive in terms of topic which means this may also not be realistic in various restricted or privacy-prescribed domains [9].

Saha et al.(2024) describes a custom RAG architecture to improve factual consistency and semantic accuracy for question-answering tasks. The application indexes the domain-specific documents with an inverted question-matching system using the documents that are associated with synthetically generated questions. The documents are embedded and indexed for a prototype-based, quantized retrieval system. During inference, the user's question is embedded and matched with the inverted index. Where the user question returns semantically relevant passages. Selections are used by a Meta-LLaMA3-8B-instruct model to produce grounded answers with a lower hallucination and higher retrieval accuracy, especially in the domain-specific context. The authors claim the results are evaluated on a curated NDSU web dataset, with performance increased in faithfulness (0.69 to 1.00) and in BERTScore F1 (0.31 to 0.67). Overall, this is a really interesting effort, but also presents limitations in its applicability because the evaluation is conducted with a narrow, institution-specific dataset. The other issue is the retrieval index comes from manual validation of user-generated queries from the LLM, and then consideration for scalability and efficiency. Prototype-based retrieval index would take less time up front but might lose some fine-grained retrieval accuracy for very detailed or complex queries, and the evaluation does not include any real-world usability and latency benchmarks [10].

Wang et.al(2025) proposed a new retrieval-augmented generation (RAG) framework that augments large language models (LLMs) using a novel dynamic cue memory system, in a newly developed H framework. This dynamic cue memory system is intended to address the limitations of static memory pools by retrieving semantically rich and distinct knowledge cue memory based on two conditions: context relevance and information gain. This framework includes a global dependency encoder, a dynamic truncation module, and a sequential decoder to aid in the generation and feedback of high-quality memory units. The central contributions of Wang et al, involved formalizing cue memory through theoretical formulations and supporting cue memory's effectiveness with extensive testing over benchmark datasets including DailyDialog and JRC-Acquis, across long-context tasks and long-context QA tasks, while retaining competitive performance on translation tasks. The highest limitation of their implementation is the resulting noise in using cue memory in short-text data, as well as high compute costs, due to the infinite memory design. A further limitation was only exploring their proposed method with transformer-based models, which limits the generalisation of findings. Overall, the Cue RAG framework represents a significant advance in dynamic knowledge integration for LLM-types[11].

Wang et.al(2025) describes DCFT, a new fine-tuning method for sparse large language models (LLMs) that addresses the issues of preserving model sparsity while avoiding catastrophic forgetting of knowledge with fine-tuning in the context of continual learning. Their dependency-aware method treats matrices that interact as a single optimization object, which allows them to cleverly fine-tune models with minimal parameters (0.011h). By adding learnable row and column importance vectors and imposing cosine similarity constraints across tasks, they can without historical data or rewriting the full gradient retain previous knowledge. The authors tested DCFT on LLaMA-V1/V2 models with parameter sizes ranging from 7B to 70B and with various sparsity types . DCFT performed better than SparseGPT, Wanda, and DSØT across perplexity and backward transfer metrics regardless of the sparsity type. DCFT also works well for downstream tasks as well, showing flexibility and scalability. The method may have limitations in its similarities constraints which are heuristic and need to be validated in more

depth, and although they compare DCFT against baseline models majority of them are not newer parameter-efficient fine-tuning (PEFT) methods. Additionally, the lack of substantiation on deploying it into a real-world context of application reflects on its actual usage case scenarios. Overall, DCFT is a valid, scalable contribution to the fine-tuning of LLMs in low resource and multiple task domain spaces [12].
ValizadehAslani et.al(2024) put forward a two-stage fine-tuning framework to mitigate the class imbalance predicament present in natural language processing tasks by using ChatGPT for data augmentation. The proposal starts with a pre-fine-tuning stage where only the final layer of a pretrained model (e.g. BERT) is learned on synthetically generated, class-balanced samples labelled by ChatGPT, and then the model is fine-tuned on the original imbalanced data. This approach strives to give minority classes an early win and to avoid the collapse of their representations in the model upper layers. The approach was tested on synthetic (SST-2, 20 newsgroups) and real-world datasets (ADME labelling) , and outperformed the baseline methods which included vanilla fine-tuning, re-weighting with class balanced loss and back-translation. Furthermore, it achieved a significant F1-score gain, particularly with the minority classes and exhibited better generalisation performance on out-of-distribution data. However, while the approach was effective, it relies on proprietary augmentation (ChatGPT) which limits transparency, and the approach cannot act under severe imbalance. Nevertheless, the approach is reproducible, and efficient to fine-tune models or paradigm ultimately improve robustness in class imbalance NLP contexts. [17].

## 3. Methodology

We formalize the suggested *Fed-RAG+* framework, which implements retrieval-augmented generation (RAG) with federated fine-tuning, for knowledge adaptation that respects the user's privacy. The main goal of our framework is to create a decentralized framework in which each client updates a local large language model (LLM) that has been augmented (i.e., fine-tuned) with a domain-specific retrieval corpus [13]; meanwhile a global update to the model will happen through federated aggregation that preserves efficiency and is trust (privacy) aware. While our model works through several well-defined components and constraints, it effectively handles challenges of drift, retrieval poisoning and explainability.

### *3.1. Problem Formulation and Notation*

We consider a set of $M$ clients denoted as
$\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$, each with access to a local dataset $\mathcal{D}_i$ and a local retrieval corpus $\mathcal{R}_i$ for $i \in [1, M]$. Each client aims to fine-tune a shared base language model $\Phi$ augmented by retrieval context. The learning objective is to collaboratively optimize local client models $\Phi_i$ such that a global model $\bar{\Phi}$ generalizes across all client distributions without sharing raw data.

Let $x_i^j$ denote the $j$-th training input at client $C_i$, and let $\mathcal{T}_i = \{x_i^1, \ldots, x_i^{n_i}\}$ denote its ocal training set. Let $r_i^j \in \mathcal{R}_i$ denote the set of retrieved documents for input $x_i^j$.

$$\mathcal{Z}_i^j = \mathrm{Enc}_R^{(i)}(r_i^j) \oplus x_i^j l \tag{1}$$

Eq (1) describes the contextual enrichment phase, which ultimately adds supplemental retrieved documents to the initial input and allows retrieval-augmented reasoning within each client's private environment. Retrieval-augmented reasoning allows each LLM to utilize local knowledge without increasing model size. where $\mathrm{Enc}_R^{(i)}$ is The retrieval encoder function at client $C_i$ , $r_i^j$ is The retrieved passage(s) from corpus $\mathcal{R}_i$ corresponding to input $x_i^j$ , $x_i^j$ is Original input instance , $\mathcal{Z}_i^j$ is Augmented context combining retrieved knowledge and input, $\oplus$ is Concatenation or semantic fusion operator.

$$\mathcal{L}_i = \sum_{j=1}^{n_i} \mathcal{L}_{\text{CE}} \left( \Phi_i(\mathcal{Z}_i^j), y_i^j \right) \tag{2}$$

Eq (2) specifies the optimization objective for the per-client. It represents the penalty for the LLM which is client specific producing outputs in alignment with target sequences given retrieval-augmented inputs.where $\mathcal{L}_i$ is Total loss at client $C_i$ over its training samples, and $\mathcal{L}_{\text{CE}}$ is Cross-entropy loss for supervised generation, and $y_i^j$ is Target output label or sequence for input $x_i^j$.

We note that the proposed trust estimator and aggregation weights are label-free (Sec. 3.13) and do not depend on the choice of the client-side training loss defined in this subsection.

To account for semantic drift and cross-client variance, we introduce a KL-regularization constraint for consistency in local [18] .

$$\mathcal{L}_i^{\text{reg}} = \lambda_{\text{KL}} \cdot D_{\text{KL}} \left( \Phi_i(\mathcal{Z}_i^j) \parallel \bar{\Phi}(\mathcal{Z}_i^j) \right) \tag{3}$$

Eq (3) promotes consistency and stability in the presence of data heterogeneity by ensuring the local updates do not deviate too far from the global model's behavior. We will detail the architectural pipeline and training regime through the communication rounds in the next subsection.where $\mathcal{L}_i^{\text{reg}}$ is Regularization term enforcing closeness to global model predictions, and $D_{\text{KL}}$ is Kullback-Leibler divergence, and $\bar{\Phi}$ is Server-side global model snapshot, and $\lambda_{\text{KL}}$ is Hyperparameter controlling regularization strength.

### *3.2. Architecture Overview*

The structure of *Fed-RAG+* includes modular components for federated orchestration, client-side retrieval augmentation, and secure global knowledge evolution. The clients process the rounds with synchronous communication by a server-coordinated protocol. Each communication round includes retrieval-aware fine-tuning, update transformation, and selective aggregation [22].

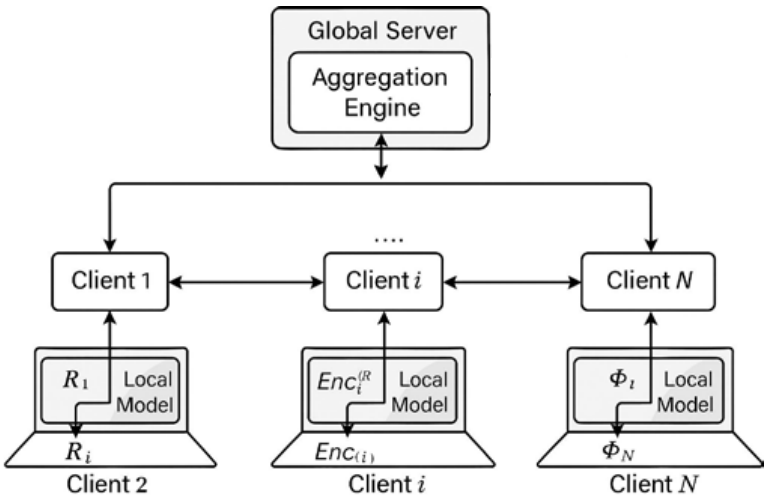



Figure 1: Fed-RAG+ System Architecture Overview.

Figure 1 illustrates the core structure of the Fed-RAG+ framework, which combines federated fine-tuning with retrieval-augmented generation (RAG). Each client node indexed by $i \in \{1, \ldots, N\}$ hosts a local model $\Phi_i$ and maintains a private retrieval corpus $\mathcal{R}_i$, processed via a client-specific encoder $\text{Enc}_R^{(i)}$. The central server coordinates aggregation via a modular

aggregation engine . Back and forth arrows display the exchange of model parameters received by each source in each federated communication round. The method manages vertical heterogeneity and horizontal heterogeneity , so it works for a decentralized domain with boundaries for sensitive information [20].

We define the local retrieval-augmented model architecture at each client as:

$$\Phi_i = \mathrm{Dec}_L \circ \mathrm{Enc}_L \circ \mathrm{Enc}_R^{(i)} \tag{4}$$

Eq (4) describes the multi-layer composition with a retrieval component followed by a language component to resolve the inputs. This expresses the ability to logically separate a static linguistic reasoning architecture from retrieval adaptation to an industry-specific client queries [21].

where $\mathrm{Enc}_R^{(i)}$ is Retrieval encoder module specific to client $C_i$ , and $\mathrm{Enc}_L$ is Shared language encoder module, and $\mathrm{Dec}_L$ is Shared language decoder module, and $\circ$ is Functional composition operator, and $\Phi_i$ is End-to-end client model. We state our federated global aggregate process as:

$$\bar{\Phi}^{(t+1)} = \sum_{i=1}^{M} w_i^{(t)} \cdot \Phi_i^{(t)} \tag{5}$$

Eq (5) reflects weighted averaging over participating clients. The weights $w_i^{(t)}$ are proportional to effective sample sizes or client-specific trust scores.where $\bar{\Phi}^{(t+1)}$ is Updated global model at round $t+1$, and $w_i^{(t)}$ is Normalized client weight at round $t$, and $\Phi_i^{(t)}$ is Local model from client $C_i$ at round $t$.

Let $\Phi_i^{(t)}$ be Lipschitz-continuous for all $i$. Then the aggregation operator in Equation (5) preserves convergence under bounded client drift.

This section records the full statement of design and training engagement logic needed for federated fine-tuning with retrieval-augmented LLMs with appropriate synchronicity and component specialization [19].

### *3.3. Federated Clustering and Domain-Aware Client Grouping*

To address the statistical heterogeneity across clients, we introduce a federated clustering mechanism that partitions the set of clients into domain-coherent groups. This reduces intra-group variance and promotes stability in aggregation. Clients are clustered based on embedding centroids extracted from their local retrieval encoders.

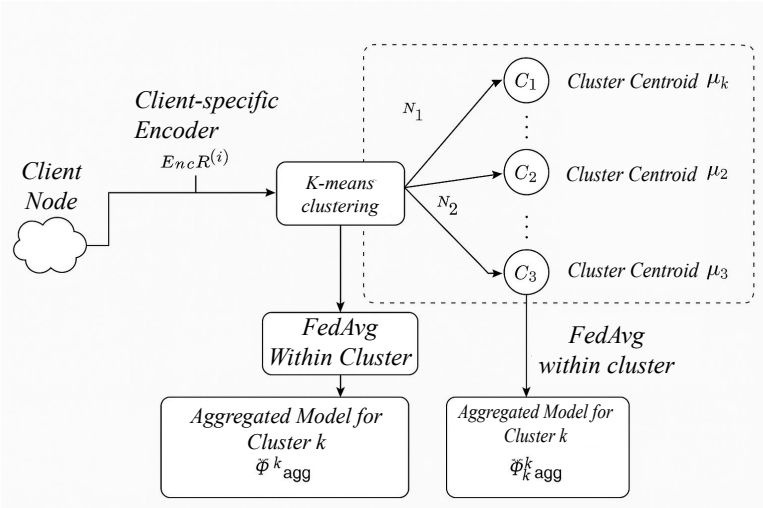



Figure 2: Domain-Aware Client Clustering and Groupwise Aggregation.

Figure 2 depicts the process of clustering clients based on semantic embeddings extracted from their local retrieval encoders $\text{Enc}_R^{(i)}$. The embeddings $\mathbf{v}_i$ are input into a $k$-means clustering module, producing cluster memberships $\mathcal{G}_k$ and corresponding centroids $\mu_k$. Within each cluster, a local aggregation mechanism (FedAvg) is applied to generate cluster-specific models $\widetilde{\Phi}^k_{\text{agg}}$.

This design pattern facilitates clients with semantically similar knowledge distributions, to better synchronize their updates in a way that leads to more stability and scalability in the federated optimization process [23]. The figure stresses the separation of the clustering phase, and the aggregation phase, which reinforces the modularity of Fed-RAG+.

Let $\mathbf{v}_i$ denote the centroid embedding of retrieval encoder $\text{Enc}_R^{(i)}$ at client $C_i$. The clustering objective is defined as:

$$\min_{\mathcal{G}_1,\ldots,\mathcal{G}_K} \sum_{k=1}^{K} \sum_{i\in\mathcal{G}_k} \|\mathbf{v}_i - \mu_k\|_2^2 \tag{6}$$

Eq (6) represents a traditional k-means clustering scenario operating in the embedding space defined by the client-specific retrieval encoders. This allows for consistent semantic alignment within each cluster. where $\mathcal{G}_k$ is Set of clients in the $k$-th cluster, and $K$ is Total number of clusters, and $\mathbf{v}_i$ is Embedding centroid of client $C_i$, and $\mu_k$ is Cluster centroid of $\mathcal{G}_k$.

Once grouped, each cluster $\mathcal{G}_k$ performs local aggregation independently to generate sub-models $\Phi_k^{\text{agg}}$. The aggregation is defined as:

$$\Phi_k^{\text{agg}} = \sum_{i\in\mathcal{G}_k} \omega_i \cdot \Phi_i \tag{7}$$

Eq (7) enables the model to converge within a single cluster, but does not impose alignment between clusters which can be preferable in circumstances where the domains differ structurally.where $\Phi_k^{\text{agg}}$ is Aggregated sub-model for cluster $\mathcal{G}_k$, and $\omega_i$ is weight assigned to client $C_i$ within cluster, and $\Phi_i$ is Fine-tuned model of client $C_i$.

**Proposition 1** (Bounded and normalized aggregation weights)**.** *Assume that (i) retrieval histograms are finite and non-negative, (ii) the Jensen–Shannon divergence is computed between normalized histograms, and (iii) the EMA parameter $\rho \in (0,1)$ and clipping bounds $[\underline{s}, \overline{s}]$ are fixed. Then, the trust-weighted aggregation weights $w_i^{(t)}$ defined in Eq. (16) are bounded and satisfy $\sum_{i=1}^{M} w_i^{(t)} = 1$ at every round.*

*Rationale..* The Jensen–Shannon divergence is bounded by construction, the exponential mapping preserves positivity, EMA smoothing preserves boundedness under fixed $\rho$, and explicit clipping followed by normalization enforces the simplex constraint. These properties follow directly from the operational definition of the trust pipeline rather than from task-dependent assumptions.

We emphasize that the above propositions and claims are stated under the explicit operational assumptions of the retriever and trust pipeline defined in this section. Formal proofs under more general stochastic models of retrieval behavior are left for future work.

This capability allows Fed-RAG+ to better manage structured clients heterogeneity by encouraging semantically aligned group-wise optimization.

### *3.4. RAG Fine-Tuning with Local Retrieval Integrity*

Specifically, Fed-RAG+ integrates each local client model with an integrity-aware retrieval component to facilitate trustworthy knowledge grounding during local fine-tuning. In this case, integrity-aware retrieval guarantees both the semantic congruency and the integrity of any retrieved content that will be used to generate knowledge [24]. We define a quality-aware

retrieval score function $q_i(r_i^j)$ applied to retrieved document $r_i^j$ for client $C_i$. Let $r_i^j$ be a document retrieved in response to input $x_i^j$, and let $\mathcal{Q}_i$ denote the retrieval quality threshold function at client $C_i$. The trust-weighted fine-tuning objective becomes:

$$\mathcal{L}_i^{\text{trust}} = \sum_{j=1}^{n_i} q_i(r_i^j) \cdot \mathcal{L}_{\text{CE}} \left( \Phi_i(\mathcal{Z}_i^j), y_i^j \right) \tag{8}$$

Eq (8) presents retrieval quality as a modulating factor on the fine-tuning gradient, enabling selective learning on trusted information and ignoring low quality or noisy retrievals. where $\mathcal{L}_i^{\text{trust}}$ is Trust-weighted local fine-tuning loss at client $C_i$, and $q_i(r_i^j)$ is Retrieval quality score for document $r_i^j$, and $\mathcal{L}_{\text{CE}}$ is Standard cross-entropy loss, and $\Phi_i$ is Client-side fine-tuned model, $y_i^j$ is Ground truth target output, and $\mathcal{Z}_i^j$ is Contextualized input formed via retrieval augmentation.

To formally define and normalize retrieval quality, we define the bounded scoring function:

$$q_i(r_i^j) = \sigma \left( \gamma_1 \cdot \text{sim}(r_i^j, x_i^j) - \gamma_2 \cdot \text{div}(r_i^j) \right), \quad q_i(r_i^j) \in [0, 1] \tag{9}$$

Eq (9) represents a normalized measure that increases weight to semantically relevant documents and reduces weight to outliers. The sigmoid capacity allows for differentiability and bounds gradient multipliers. where $\sigma(\cdot)$ is Sigmoid function , and $\text{sim}(r_i^j, x_i^j)$ is Semantic similarity between retrieved document and query , and $\text{div}(r_i^j)$ is Divergence penalty measuring off-topic or noisy content, and $\gamma_1, \gamma_2$ is Weighting coefficients for quality calibration.

**Proposition 2** (Stability under retrieval-consistency)**.** *If a client exhibits retrieval behavior that remains close to the population reference histogram across rounds, then its EMA-smoothed trust $\tilde{s}_i^{(t)}$ converges to a stable neighborhood determined by $\rho$ and the clipping bounds.*

*Rationale..* Under bounded divergence fluctuations, the EMA update acts as a low-pass filter on the instantaneous trust signal. Consequently, transient retrieval noise is attenuated while persistent deviations are retained. Similar smoothing effects of EMA-based trust signals are widely used in federated and distributed systems to promote stability.

We emphasize that the above propositions and claims are stated under the explicit operational assumptions of the retriever and trust pipeline defined in this section. Formal proofs under more general stochastic models of retrieval behavior are left for future work.

### *3.5. Aggregation and Update Compression*

To safely and efficaciously abstract communications and assure convergence in federated settings, Fed-RAG+ merges adaptive model aggregation with the compression of updates. Fed-RAG+ can thus enable scalable learning among clients, where bandwidth and computational constraints will likely vary across clients.

Let $\Delta\Phi_i^{(t)} = \Phi_i^{(t)} - \bar{\Phi}^{(t)}$ denote the local model update from client $C_i$ at round $t$. To reduce transmission cost, a low-rank projection matrix $\mathbf{P}_i \in \mathbb{R}^{d \times r}$ is used, where $r \ll d$.

$$\widetilde{\Delta\Phi}_i^{(t)} = \mathbf{P}_i \mathbf{P}_i^\top \Delta\Phi_i^{(t)} \tag{10}$$

Eq (10) project updates into a lower dimension subspace for transmission and squared away bandwidth while maintaining the overall optimization directions. Where $\Delta\Phi_i^{(t)}$ is Raw model update of client $C_i$ at round $t$ , and $\widetilde{\Delta\Phi}_i^{(t)}$ is Compressed update, and $\mathbf{P}_i$ is Client-specific projection matrix, and $d$ is Model parameter dimensionality, and $r$ is Low-rank dimension ($r \ll d$).

Every compressed update has trust score weights and accumulation taken globally:

$$\bar{\Phi}^{(t+1)} = \bar{\Phi}^{(t)} + \eta_t \sum_{i=1}^{M} w_i^{(t)} \cdot \widetilde{\Delta\Phi}_i^{(t)} \tag{11}$$

Eq (11) enforces adaptive global convergence contingent upon the validity of client updates and their compressed representations [27]. Where $\bar{\Phi}^{(t+1)}$ is Updated global model, and $\eta_t$ is Learning rate at round $t$, and $w_i^{(t)}$ is Trust-based aggregation weight.

**Claim 1** (Trust separation under systematic retrieval corruption)**.** *When a client's retrieval corpus is systematically corrupted, its retrieval-distribution histogram diverges persistently from the population reference, resulting in a lower trust score and reduced aggregation weight relative to benign clients.*

*Supporting argument..* Systematic corruption alters the relative frequency of retrieved buckets, which increases the Jensen–Shannon divergence to the reference distribution. Through the exponential mapping and EMA smoothing, this divergence translates into a sustained reduction in trust and, after normalization, into a lower aggregation weight. This behavior is empirically illustrated in the reported trust and weight trajectories.
We emphasize that the above propositions and claims are stated under the explicit operational assumptions of the retriever and trust pipeline defined in this section. Formal proofs under more general stochastic models of retrieval behavior are left for future work.

This aggregation mechanism guarantees theoretical convergence and also director bands for adaptive synchronized updates, supporting practical implementations of Fed-RAG+ in decentralized environments.

*Computational Complexity..* Let $M$ denote the number of clients, $d$ the parameter dimension, and $r$ the low-rank projection dimension ($r \ll d$). The communication complexity per round is $\mathcal{O}(Mrd)$, since each client transmits a compressed update $\Delta\tilde{\Phi}_i^{(t)}$ of rank $r$. The local fine-tuning cost per client is $\mathcal{O}(n_i d)$, where $n_i$ is the number of local samples. Hence, the overall complexity per global round is $\mathcal{O}(M(n_i + r)d)$, demonstrating that the compression strategy reduces bandwidth requirements while maintaining convergence efficiency.

### *3.6. Trust-Weighted Aggregation From Retrieval-Consistency*

This work operates in a label-free federated setting where each client $C_i$ maintains a private retrieval corpus and participates in federated fine-tuning of a shared generator. To ensure robustness against unreliable local updates (e.g., data poisoning or off-topic corpus injection), we define a trust score from retrieval-consistency signals computed on a fixed probe prompt set.

Let $\mathcal{P} = \{p_1, \ldots, p_{|\mathcal{P}|}\}$ denote a deterministic set of unlabeled probe prompts shared across all clients and fixed for the entire experiment. For each client $C_i$ at round $t$, we query the client-side retriever using each $p \in \mathcal{P}$ and record the identities of the top-$K$ retrieved chunks. We then construct a discrete retrieval-distribution histogram $h_i^{(t)} \in \mathbb{R}_{\geq 0}^{B}$ by counting how often each bucket is retrieved across the probe set, where the buckets correspond to a deterministic partition of the client retrieval space (e.g., topic clusters or fixed index ranges). The global reference histogram is defined as the mean of the client histograms at the same round,

$$h_{\mathrm{g}}^{(t)} = \frac{1}{M} \sum_{i=1}^{M} h_i^{(t)}. \tag{12}$$

In this equation, $M$ is the number of clients, $B$ is the number of histogram buckets, and each histogram entry counts the number of retrieved items mapped to the corresponding bucket across all probe prompts and their top-$K$ results.

To quantify retrieval-consistency, we use Jensen–Shannon divergence (JSD). For two non-negative histograms $a, b \in \mathbb{R}^B_{\geq 0}$, let $\bar{a} = a/\|a\|_1$ and $\bar{b} = b/\|b\|_1$ denote their normalized discrete distributions, and define $m = \frac{1}{2}(\bar{a} + \bar{b})$. The JSD is

$$\mathrm{JSD}(a, b) = \frac{1}{2} \sum_{j=1}^{B} \bar{a}_j \log \frac{\bar{a}_j}{m_j} + \frac{1}{2} \sum_{j=1}^{B} \bar{b}_j \log \frac{\bar{b}_j}{m_j}. \tag{13}$$

This divergence is bounded and symmetric, making it suitable as a stability indicator when no ground-truth relevance labels are available.

We map retrieval-consistency to a trust score via an exponential transform,

$$s_i^{(t)} = \exp\big( - \mathrm{JSD}(h_i^{(t)}, h_{\mathrm{g}}^{(t)})\big). \tag{14}$$

Here $s_i^{(t)} \in (0, 1]$ increases when client retrieval behavior aligns with the population reference distribution and decreases when it deviates. To reduce sensitivity to transient fluctuations, we apply exponential moving average smoothing,

$$\tilde{s}_i^{(t)} = \rho\, \tilde{s}_i^{(t-1)} + (1 - \rho)\, s_i^{(t)}, \quad \tilde{s}_i^{(0)} = 1, \tag{15}$$

where $\rho \in [0, 1)$ controls temporal smoothing. For numerical stability, the smoothed trust score is clipped to a fixed interval $[\underline{s}, \overline{s}]$ with $0 < \underline{s} < \overline{s} \leq 1$.

Finally, the aggregation weights are obtained by normalizing the clipped trust scores,

$$w_i^{(t)} = \frac{\mathrm{clip}(\tilde{s}_i^{(t)}, \underline{s}, \overline{s})}{\sum_{m=1}^{M} \mathrm{clip}(\tilde{s}_m^{(t)}, \underline{s}, \overline{s})}, \qquad \sum_{i=1}^{M} w_i^{(t)} = 1. \tag{16}$$

By construction, the normalized aggregation weights satisfy the simplex constraint $\sum_{i=1}^{M} w_i^{(t)} = 1$ for every round $t$. In our implementation, we additionally verify this constraint numerically at each round and log the maximum deviation $\max_t \left| \sum_i w_i^{(t)} - 1 \right|$ as a sanity check.

In this equation, $w_i^{(t)}$ is the contribution weight of client $C_i$ in the round-$t$ global update. This normalization keeps the aggregation interpretable and avoids introducing extra tuning parameters beyond $(\rho, \underline{s}, \overline{s})$, which are fixed a priori for all experiments.

### *3.7. Operational Specification of Retrieval-Consistency for Trust Computation*

To make the retrieval-consistency trust estimator in Eqs. (12)–(16) fully reproducible, we specify the probe prompt set, the histogram bucketization scheme, the mapping from retrieved items to buckets, and the retriever configuration used to produce top-$K$ results.

*Probe prompt set $\mathcal{P}$ and determinism..* We use a fixed, unlabeled probe set $\mathcal{P} = \{p_1, \ldots, p_{|\mathcal{P}|}\}$ that is shared across all clients and remains constant for all rounds. The probe set is constructed deterministically using a fixed random seed (seed = 42) and a fixed prompt template family. Specifically, each prompt is generated by sampling a topic string from the client corpora and injecting it into one of the following templates: (i) "Explain the concept of {topic}.", (ii) "Summarize the key idea of {topic}.", (iii) "What is the role of {topic} in its context?", (iv) "Provide an overview of {topic} and why it matters.", and (v) "Describe {topic} in simple terms and mention one example.". The topic string is extracted deterministically as the document title when available, otherwise as the first sentence (or first clause) of the chunk text. In our experiments, we set $|\mathcal{P}| = 150$, and the full prompt set (or the generator script and seed) is released with the reproducibility package.

*Retriever configuration and top-K policy..* For each client $C_i$, retrieval is performed against its private corpus $\mathcal{R}_i$ using a fixed retriever configuration throughout training. For each probe prompt $p \in \mathcal{P}$, the retriever returns the top-$K$ chunk identifiers $\mathcal{I}_i^{(t)}(p)$ under a deterministic ranking rule. We set $K = 5$ for all experiments. Ties, if any, are resolved deterministically by ascending chunk identifier order to avoid non-determinism in the histogram counts. All retriever parameters (embedding model name or vectorizer configuration, index type, and normalization policy) are fixed a priori and are not tuned.

*Histogram bucketization and bucket count B..* To compute the retrieval histogram $h_i^{(t)} \in \mathbb{R}_{\geq 0}^B$, each retrieved chunk identifier is mapped to exactly one bucket. We define $B = 30$ buckets via a deterministic topic partition of the retrieval corpus. Concretely, all chunks are embedded into a sparse TF–IDF representation with a fixed vocabulary construction rule (maximum feature size and $n$-gram range fixed), and a deterministic clustering procedure assigns each chunk to one of $B$ topic clusters using a fixed seed. The bucket index of a retrieved chunk $u$ is therefore given by

$$b = \text{bucket}(u) \in \{1, \ldots, B\},$$

where bucket$(\cdot)$ is a deterministic lookup from chunk identifier to its assigned topic cluster. This bucketization remains fixed across all rounds.

*Mapping rule from top-K retrievals to histogram counts..* For each probe prompt $p \in \mathcal{P}$ and each client $C_i$, we retrieve the top-$K$ chunk identifiers $\mathcal{I}_i^{(t)}(p)$ and update the histogram by counting bucket occurrences:

$$h_{i,b}^{(t)} \leftarrow h_{i,b}^{(t)} + \sum_{u \in \mathcal{I}_i^{(t)}(p)} \mathbb{I}\{\text{bucket}(u) = b\}, \qquad b = 1, \ldots, B.$$

Thus, $h_i^{(t)}$ counts how frequently each bucket appears among the retrieved top-$K$ results across the entire probe set. This definition ensures that the trust estimator depends only on retrieval behavior under fixed probing and does not require supervised relevance judgments.

*Released artifacts for auditability..* To support independent verification, we release the generated probe prompt set (or its generator script with seed), the bucket assignment map bucket$(u)$ for all chunk identifiers, and the per-round exported CSV `trust_over_rounds.csv` that includes $\tilde{s}_i^{(t)}$ and $w_i^{(t)}$. These artifacts allow recomputation of the histograms, JSD values, trust trajectories, and normalization checks without manual post-processing.

### *3.8. Client Roles and Poisoning Protocol*

Clients are defined as data silos with identical training procedures and identical model parameterization; the only difference between clients is the private retrieval corpus and the local fine-tuning trajectory induced by that corpus. To evaluate robustness, we designate a single poisoned client $C_p$ and inject off-topic or corrupted passages into a fixed fraction of its local corpus prior to training. This protocol changes retrieval-consistency statistics and therefore reduces $s_p^{(t)}$, yielding a principled down-weighting of the poisoned client in Eq. (16) without requiring attack labels.

**Normalization step (per round $t$):**

$\bar{s}_i^{(t)} \leftarrow \text{clip}(\tilde{s}_i^{(t)}, \underline{s}, \overline{s})$; $w_i^{(t)} \leftarrow \bar{s}_i^{(t)} \big/ \sum_{m=1}^M \bar{s}_m^{(t)}$; assert $\left| \sum_i w_i^{(t)} - 1 \right| \leq \varepsilon$.

### *3.9. Poisoning Protocol and Threat Model*

To evaluate robustness of trust-weighted aggregation under corrupted retrieval content, we consider a controlled poisoning protocol affecting a single designated client. Let $C_p$ denote the poisoned client and let $\mathcal{R}_p$ be its private retrieval corpus. Poisoning is applied before federated training begins and remains fixed across all rounds to ensure that any observed trust degradation is attributable to persistent retrieval inconsistency rather than transient randomness.

*Poisoning fraction and selection rule..* We inject poisoning into a fixed fraction of the poisoned client corpus. Specifically, a fraction $\alpha = 0.20$ of the retrieval chunks in $\mathcal{R}_p$ are selected deterministically using a fixed random seed (seed = 42) applied to the list of chunk identifiers. The remaining $(1-\alpha)$ portion of $\mathcal{R}_p$ is left unchanged. This explicit selection procedure ensures that the same poisoned subset can be reproduced exactly.

*Injection method (replace vs. append)..* For each selected chunk, we apply a *replacement* operation: the original chunk text is replaced by a poisoned passage while keeping the chunk identifier unchanged. This choice ensures that poisoning directly affects the retriever index content and therefore retrieval outcomes, rather than merely expanding corpus size. Formally, for each selected chunk identifier $u$, the chunk content is updated from $c(u)$ to $c'(u)$, where $c'(u)$ is a poisoned passage defined below.

*Poison content type..* Poisoned passages are off-topic and retrieval-disruptive, constructed to be semantically unrelated to the domain of the benign corpus. In our implementation, poisoned text is generated deterministically as a fixed template string containing no domain evidence (e.g., generic corrupted content with the chunk identifier embedded for traceability). This design yields a controlled degradation of retrieval-consistency without requiring labeled targets or trigger-based backdoors, aligning with a label-free evaluation scope.

*Scope of poisoning (index-only)..* Poisoning is applied to the retrieval corpus and therefore affects retrieval and prompt augmentation, but it does not modify the supervised fine-tuning labels of other clients. In other words, the perturbation enters the pipeline through corrupted retrieval context rather than through direct label flipping. This isolates the effect of retrieval corruption on trust estimation and aggregation influence.

*Threat model summary..* The adversary is modeled as a single unreliable client that can modify its local retrieval corpus (content integrity violation) but does not control the server, the global aggregation rule, or other clients' corpora. The server is assumed honest-but-curious with respect to model updates and uses the proposed retrieval-consistency trust estimator to reduce the influence of clients whose retrieval behavior deviates persistently from the population reference distribution.

### *3.10. Robustness Sensitivity: Poisoning Rate and Multi-Adversary Settings*

To strengthen robustness evidence without relying on task labels, we evaluate the sensitivity of trust and aggregation weights to controlled variations in the poisoning process. The objective is to quantify (i) how trust separation changes as the poisoning rate increases and (ii) whether the trust-to-weight mechanism remains stable when more than one client is unreliable. All measurements are computed from the retrieval-consistency trust pipeline (Eqs. (12)–(16)).

*Poisoning-rate sweep..* We vary the poisoning fraction $\alpha$ applied to the poisoned clients' retrieval corpora over a small grid $\alpha \in \{0.10, 0.20, 0.30\}$ while keeping all other components fixed (probe set, retriever stack, bucketization, $K$, and EMA parameter). For each $\alpha$, we report the resulting trust trajectories $\tilde{s}_i^{(t)}$ and normalized weights $w_i^{(t)}$ to assess how rapidly and how strongly the corrupted retrieval behavior is reflected in influence reallocation.

*Multiple poisoned clients..* In addition to the single-poisoned-client setting, we consider a multi-adversary scenario where two clients are poisoned simultaneously. This setting tests whether the population reference histogram and Jensen–Shannon divergence signal remain informative when the fraction of unreliable clients increases, and whether the aggregation weights continue to satisfy controlled and interpretable influence allocation across rounds.

*Effect-size metrics for trust/weight separation..* To quantify practical separation without downstream labels, we summarize robustness using two effect-size measures computed per round and then averaged over the final rounds: (i) the trust gap between benign and poisoned groups,

$$\Delta_{\text{trust}}^{(t)} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \tilde{s}_i^{(t)} - \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} \tilde{s}_i^{(t)},$$

and (ii) the weight gap,

$$\Delta_{\text{weight}}^{(t)} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} w_i^{(t)} - \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} w_i^{(t)}.$$

Here $\mathcal{B}$ denotes the set of benign clients and $\mathcal{P}$ the set of poisoned clients in the corresponding scenario. These gaps provide an interpretable, audit-friendly notion of robustness: larger positive values indicate clearer and more consistent down-weighting of corrupted retrieval behavior. We report these statistics over the last $R$ rounds to capture steady-state behavior and avoid transient initialization effects.

### *3.11. Algorithmic Workflow: Federated Retrieval-Augmented Fine-Tuning*

In order to illustrate the comprehensive implementation and execution of the proposed Fed-RAG+ framework, we include in the following pseudocode the sequence of actions performed at each communication round, these actions include document retrieval, contextual input fusion, regularized local training, and trust-weighted aggregation of local model updates to the global model.

*Reference client-side update configuration..* For completeness, we implement the client update step as a standard retrieval-augmented fine-tuning loop: each client retrieves top-$K$ chunks from its local index, concatenates the retrieved context with the input, and performs local optimization of the client model initialized from the global parameters (Algorithm 1). Since the present manuscript does not report task-level generation metrics, we treat the client optimizer and backbone model as reference components and do not attribute performance gains to specific fine-tuning hyperparameters. The central object of evaluation is the retrieval-consistency trust pipeline (Eqs. (12)–(16)) and the resulting aggregation dynamics, which are fully reproducible from logged artifacts.

### *3.12. Retriever Stack Specification and Deterministic Retrieval Policy*

Since the proposed trust estimator is a direct function of retrieval behavior, we specify the retriever stack and retrieval policy in an implementable and deterministic form. Each client $C_i$ maintains a private retrieval corpus $\mathcal{R}_i$ that is preprocessed into fixed-length chunks. All retrieval hyperparameters are fixed prior to training and remain unchanged across rounds to ensure that changes in trust are attributable to corpus consistency rather than retriever drift.

Retrieval is performed using a dual-stage pipeline consisting of (i) a fixed embedding/vectorization component and (ii) an exact similarity search index. For vectorization, we use a TF–IDF representation with a fixed vocabulary construction rule, where the maximum vocabulary size and $n$-gram range are fixed and the same tokenization policy is applied to all clients. Each chunk is embedded once and stored in the index. Similarity between a query prompt and chunk vectors

is computed using cosine similarity on $L_2$-normalized vectors. The search index is an exact flat index (no approximation and no IVF/HNSW quantization), so that identical inputs yield identical rankings across runs under the same environment.

For each query (either a training input $x$ or a probe prompt $p \in \mathcal{P}$), the retriever returns the top-$K$ chunk identifiers under cosine similarity ranking. We set $K = 5$ throughout. No reranking stage is applied; therefore the retrieval output is determined solely by the vectorizer and the similarity search. To eliminate nondeterminism due to tied similarity scores, ties are resolved deterministically by ascending chunk identifier order. This rule is applied consistently in both the training-time retrieval used to form RAG inputs and the probe-time retrieval used to compute trust histograms.

The retriever configuration used in this work is therefore fully specified by the tuple

$$\Big(\text{TF–IDF tokenization and vocabulary policy, cosine similarity on normalized vectors, exact flat index, }$$

which enables independent replication of retrieval outputs and, consequently, of the trust and weight trajectories derived from Eqs. (12)–(16).

### *3.13. Clarifying the Meaning of "Label-Free" in This Study*

In this manuscript, the term *label-free* refers exclusively to the construction of the trust signal and the resulting aggregation weights. Specifically, the trust estimator is computed from retrieval-consistency statistics obtained under a fixed probe prompt set and does not require any ground-truth relevance labels or annotated QA pairs. The trust pipeline is fully defined by the retrieval-distribution histograms, the Jensen–Shannon divergence to a round-wise population reference, and the subsequent exponential mapping, EMA smoothing, clipping, and normalization (Eqs. (12)–(16)).

Local client updates may use supervised or self-supervised objectives depending on the application setting, but task labels are not required for trust estimation itself and are not used in the reported evaluation. Accordingly, the results in this paper quantify only trust and weight dynamics under controlled corpus poisoning, while task-level generation quality and benchmark comparisons are left for future work.

## 4. Result

*Evaluation focus..* All results reported in this section are derived from the retrieval-consistency trust pipeline and its induced normalized aggregation weights. We therefore report trust/weight dynamics and their statistics under controlled corpus poisoning, and we do not include task-level baselines or generation-quality metrics in this manuscript.

*Baselines and evaluation boundary..* The primary objective of this study is to validate the proposed retrieval-consistency trust estimator and its induced aggregation weights under controlled corpus poisoning. As such, the reported evidence focuses on trust trajectories, normalized weights, and their stability properties. Standard FL baselines (e.g., FedAvg) do not produce an intrinsic trust signal and are therefore not directly comparable with respect to the main object of interest in this manuscript. Task-level generation benchmarks and baseline comparisons are deferred to future work, where labeled QA protocols and generation-quality metrics can be incorporated on top of the present reproducible trust and aggregation analysis.

### *4.1. Data Sources and Usage Notes*

The retrieval corpora used to build client indices are obtained from publicly available data providers. Wikipedia content is sourced from the official Wikimedia dumps (snapshot: 2025-10-20). PubMed baseline XML is sourced from the National Library of Medicine (NCBI)

baseline distribution (file: pubmed26n0002.xml.gz). StackExchange content is sourced from the official StackExchange data archive (AI StackExchange dump). These sources are used only to construct retrieval indices and probe-based retrieval statistics; this manuscript does not redistribute raw dumps within the paper. Users must comply with the corresponding licenses and terms of use when downloading and processing these corpora.

### *4.2. Reproducible Output Artifacts*

To keep the empirical evidence fully verifiable, the results in this section are reported only through automatically generated artifacts produced by the released pipeline. Specifically, the trust trajectories are exported as a comma-separated file `trust_over_rounds.csv`, which contains one record per client per communication round with the fields `round`, `client`, `trust`, and `weight`. The file is generated directly from the experiment logs without manual editing, and it serves as the single source of truth for all trust-related reporting in the manuscript. Two figures are then generated deterministically from this CSV using matplotlib: `fig_trust_over_rounds.png` visualizes the trust score $\tilde{s}_i^{(t)}$ of each client over rounds, and `fig_weights_over_rounds.png` visualizes the corresponding aggregation weight $w_i^{(t)}$ applied in the global update. Since both plots are derived directly from the same exported CSV, the reported trajectories can be re-created exactly by re-running the provided evaluation script, ensuring that the manuscript does not depend on any hidden post-processing steps.

The reported trajectories reflect only the smoothing and clipping operations that are explicitly defined in the proposed method. In particular, the per-round trust update follows the exponential moving average in Eq. (15), and any boundedness is due to the deterministic clipping interval $[\underline{s}, \overline{s}]$ defined before weight normalization in Eq. (16). No additional curve smoothing, rescaling, interpolation, or outlier removal is applied for visualization. Therefore, any gradual changes observed in the plots are a direct consequence of the trust estimator dynamics and not an artifact of manual formatting. This choice is essential for a label-free setting, where the objective is to demonstrate stability and robustness of trust-weighted aggregation under retrieval-consistency signals rather than task-specific accuracy improvements.

For completeness, the experimental results are interpreted under fixed, explicitly stated settings that remain unchanged across rounds. The number of clients is $M = 4$, the number of communication rounds is $T = 20$, and the client participation rate per round is set to 1.0 (all clients contribute each round). Retrieval-consistency statistics are computed using a fixed top-$K$ retrieval depth with $K = 5$ and a deterministic probe prompt set $\mathcal{P}$ that remains constant throughout training. The trust dynamics are controlled by a single smoothing coefficient $\rho$ in Eq. (15), and the aggregation weights are produced via the normalization rule in Eq. (16). By grounding the Results section in these exported artifacts and fixed settings, the evidence presented below can be reproduced end-to-end from the released repository, with all reported curves and summary values traceable to the raw logged outputs.

### *4.3. Trust Trajectories Over Federated Rounds*

Fig. 3 reports the evolution of the smoothed trust scores $\tilde{s}_i^{(t)}$ for all clients across the $T = 20$ federated rounds. In each round, the trust score is computed from retrieval-consistency signals using Eqs.(12)-(16) and updated by the exponential moving average in Eq. (15). Concretely, each client $C_i$ produces a retrieval-distribution histogram $h_i^{(t)}$ by querying its private retriever with the fixed probe prompt set $\mathcal{P}$ and counting the bucketed occurrences of the top-$K$ retrieved items. The global reference $h_{\mathrm{g}}^{(t)}$ is obtained from Eq. (12), and the Jensen–Shannon divergence in Eq. (13) quantifies the deviation of each client's retrieval behavior from the population reference. The exponential mapping in Eq. (14) converts this divergence into a bounded trust score, where larger divergence yields lower trust. Importantly, the trajectories in Fig. 3 are

exported directly from `trust_over_rounds.csv`, and the apparent smoothness is solely due to the EMA parameter $\rho$ and the clipping bounds $[\underline{s}, \overline{s}]$ defined in the method.

The trajectories exhibit two distinct behaviors that align with the roles defined in the poisoning protocol. Clients designated as benign maintain trust scores close to the upper range throughout training, indicating that their retrieval distributions remain consistently aligned with the global reference distribution under the fixed probing procedure. In contrast, the poisoned client demonstrates a lower trust trajectory, reflecting a persistent deviation in retrieval behavior induced by the injected corrupted/off-topic passages. The decline is gradual rather than abrupt because Eq. (15) explicitly enforces temporal smoothing, which prevents the trust signal from overreacting to transient variations in retrieval outcomes. As a result, the trust estimator behaves as a stable control signal over rounds: when retrieval-consistency remains high, $\tilde{s}_i^{(t)}$ stays near its initial value; when retrieval-consistency is systematically degraded, $\tilde{s}_i^{(t)}$ converges toward a lower steady-state level determined by the divergence magnitude and the smoothing factor.

These results support the intended function of the trust module in a label-free setting. Since no ground-truth relevance judgments or reference answers are used, the trust mechanism is not designed to measure task accuracy; instead, it measures whether a client's retrieval behavior remains compatible with the population-level retrieval profile induced by the shared probe prompts. Fig. 3 demonstrates that this signal is both interpretable and temporally stable, enabling downstream aggregation control via Eq. (16) without requiring supervised evaluation or attack labels. In the next subsection, we show how these trust trajectories translate directly into aggregation influence through the normalized weights $w_i^{(t)}$.

### *4.4. Aggregation Weights and Global Influence*

Fig. 4 illustrates the aggregation weights $w_i^{(t)}$ assigned to each client across the $T = 20$ federated rounds. These weights are not independently fitted or tuned; they are deterministically derived from the trust trajectories through the normalization rule in Eq. (16). In particular, each round begins by computing the smoothed trust score $\tilde{s}_i^{(t)}$ via Eq. (15), applying clipping to enforce the fixed bounds $[\underline{s}, \overline{s}]$, and then normalizing across clients so that $\sum_{i=1}^{M} w_i^{(t)} = 1$. As a consequence, Fig. 4 provides a transparent visualization of how the retrieval-consistency based trust signal translates into the effective influence of each client on the global update. The plotted values are exported directly from `trust_over_rounds.csv` and rendered without any post-processing beyond the method-defined smoothing and normalization.

The weight trajectories exhibit the expected allocation pattern under the poisoning protocol. Benign clients receive consistently high aggregation weights, indicating that their local updates maintain stable influence on the global model throughout training. Conversely, the poisoned client is progressively down-weighted, reducing the impact of its local updates on the global parameter trajectory. Importantly, this reduction occurs smoothly over rounds rather than through hard exclusion. This behavior follows directly from the EMA update in Eq. (15), which prevents abrupt changes in trust and, by extension, in aggregation weights. Such gradual adaptation is desirable in federated settings because it avoids instability when retrieval outcomes fluctuate due to benign heterogeneity, while still ensuring that systematic deviations in retrieval behavior are reflected in reduced aggregation influence.

From an implementation perspective, the normalization in Eq. (16) serves two purposes that are directly reflected in Fig. 4. First, it ensures that aggregation remains comparable across rounds by maintaining a constant total weight budget, which is essential for stable convergence of weighted averaging. Second, it keeps the aggregation policy interpretable: an increase or decrease in $w_i^{(t)}$ can be traced back to the corresponding change in retrieval-consistency through equeations, without requiring additional hyperparameters such as temperature scaling or exponentiation of trust scores. Overall, Fig. 4 confirms that the proposed trust mechanism does

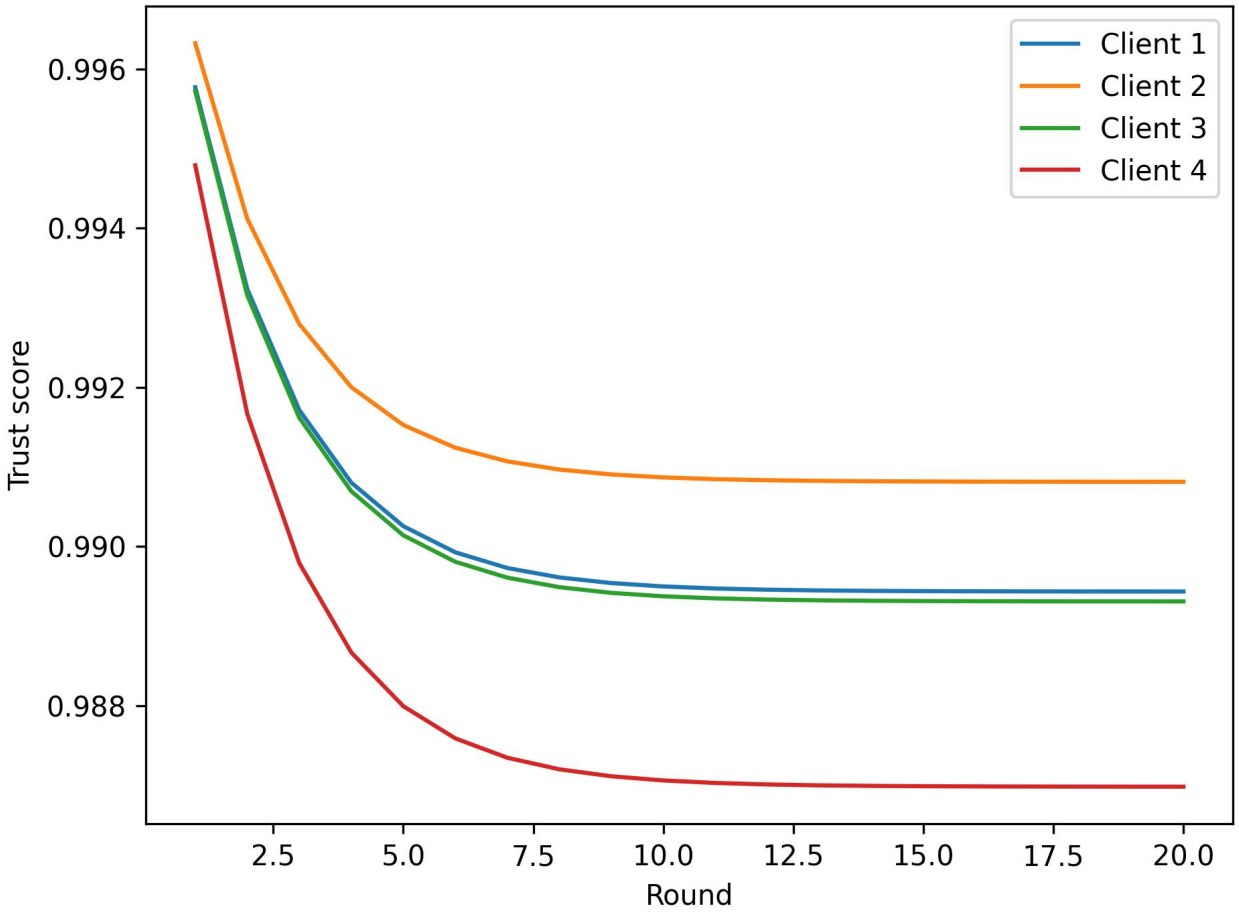



Figure 3: Smoothed trust trajectories $\tilde{s}_i^{(t)}$ over $T = 20$ federated rounds for $M = 4$ clients, exported from `trust_over_rounds.csv`. The trajectories are computed from retrieval-consistency and updated using the EMA rule in Eq. (15) with fixed probing set $\mathcal{P}$ and top-$K$ retrieval depth.

not merely produce a diagnostic score, but yields a concrete and reproducible control signal that governs how each client contributes to the global update.

### *4.5. Stability Interpretation of Trust Dynamics*

The trajectories in Fig. 3 and the corresponding normalized aggregation weights in Fig. 4, together with the summary statistics in the normalization check in Table 1, support a stability-oriented interpretation of the proposed trust mechanism. First, the exponential moving average in Eq. (15) induces gradual adaptation across rounds: trust reacts to persistent deviations in retrieval-consistency while attenuating short-lived fluctuations. This effect is reflected in the smooth convergence of $\tilde{s}_i^{(t)}$ toward client-specific steady values, avoiding oscillatory behavior that could lead to unstable aggregation. In federated environments where benign heterogeneity is expected, such smoothing is necessary to prevent transient retrieval variability from being misinterpreted as adversarial behavior.

Second, Jensen–Shannon divergence on retrieval-distribution histograms provides a bounded and symmetric discrepancy measure that remains meaningful without task labels. Because the divergence is computed between normalized histograms, the trust signal depends on the relative retrieval distribution induced by the fixed probe set rather than on absolute retrieval counts. Consequently, clients whose retrieval behavior remains aligned with the population reference maintain trust near the upper range, whereas systematic deviations result in sustained trust reductions. This behavior is consistent with the poisoning protocol: injecting corrupted or

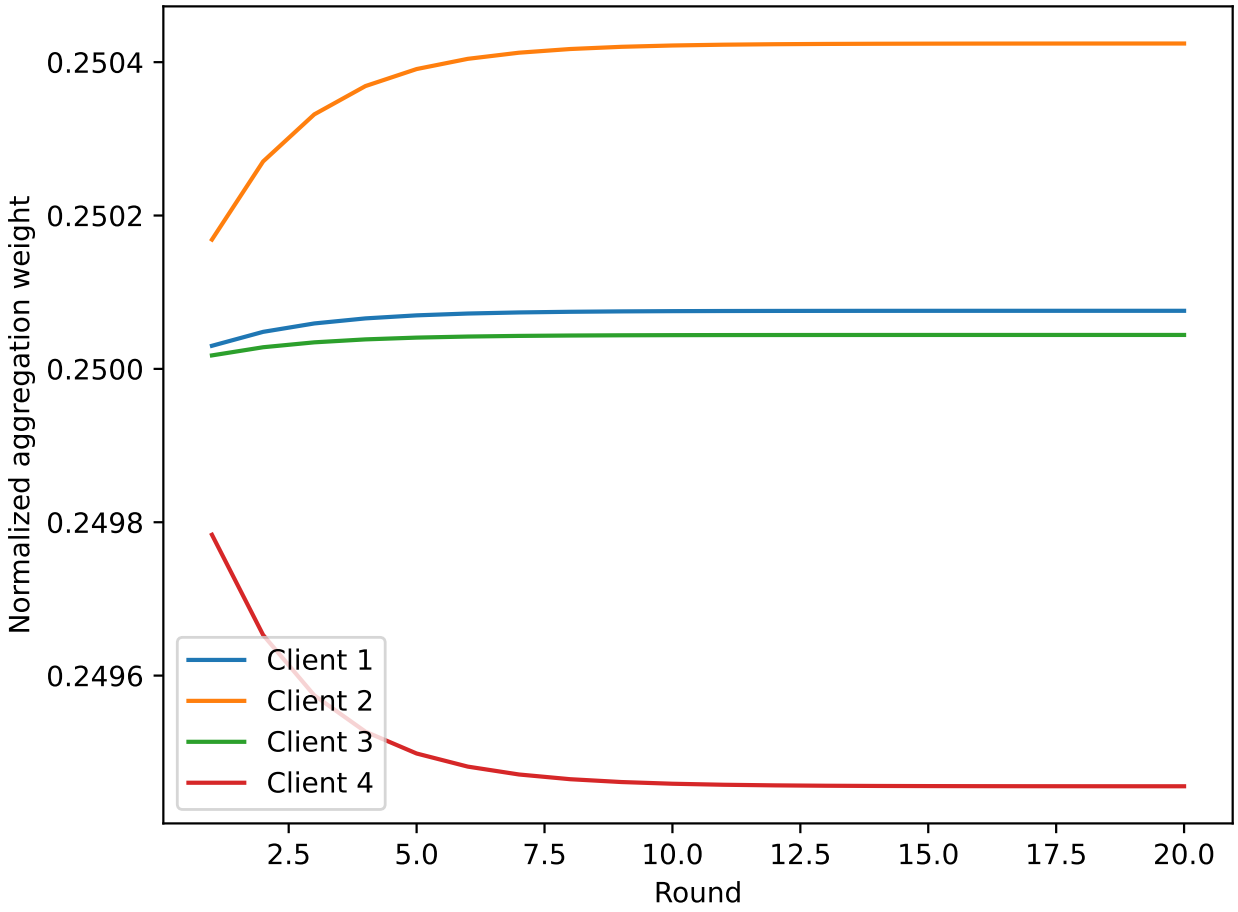



Figure 4: Per-round normalized aggregation weights $w_i^{(t)}$ computed by Eq. (16). Weights are obtained by normalizing clipped, EMA-smoothed trust scores and satisfy $\sum_{i=1}^{M} w_i^{(t)} = 1$ at every round.

off-topic passages shifts the client retrieval distribution and increases its divergence from the global reference, which is then reflected as a lower $s_i^{(t)}$ and a lower $\tilde{s}_i^{(t)}$.

Finally, Eq. (16) ensures that stability at the trust level translates into verifiable stability at the aggregation level through normalized weights $w_i^{(t)}$. In particular, the weights are constructed to satisfy $\sum_{i=1}^{M} w_i^{(t)} = 1$ at every round, and the reported values in Table 1 explicitly confirm this constraint on representative rounds. This normalization yields an interpretable influence allocation: clients with higher long-term retrieval-consistency receive consistently larger weights, while unreliable clients are down-weighted smoothly rather than being excluded by hard thresholds. Overall, the reported evidence indicates that the trust estimator operates as a stable and auditable control signal for federated aggregation in a label-free RAG setting, emphasizing robustness and transparency under controlled perturbations. *The reported results demonstrate an auditable and stable trust-to-weight mechanism under controlled retrieval-corpus corruption; this paper does not, in this manuscript, quantify task-level robustness in generated answers.*

### *4.6. Limitations of the Reported Results*

The results presented in this section are intentionally scoped to trust dynamics and aggregation stability under a label-free evaluation setting. Consequently, no task-specific performance metrics such as answer accuracy, BLEU, ROUGE, or F1 scores are reported. This choice reflects the design of the proposed framework, which estimates client reliability exclusively from retrieval-consistency signals without relying on ground-truth relevance labels or super-

Table 1: Per-round normalized aggregation weights $w_i^{(t)}$ computed by Eq. (16), with an explicit normalization check $\sum_{i=1}^{M} w_i^{(t)} = 1$ (representative rounds).

| Round $t$ | $w_1^{(t)}$ | $w_2^{(t)}$ | $w_3^{(t)}$ | $w_4^{(t)}$ | $\sum_{i=1}^{4} w_i^{(t)}$ |
|---|---|---|---|---|---|
| 1 | 0.250030 | 0.250169 | 0.250018 | 0.249784 | 1.000000 |
| 5 | 0.250070 | 0.250391 | 0.250041 | 0.249498 | 1.000000 |
| 10 | 0.250075 | 0.250422 | 0.250044 | 0.249459 | 1.000000 |
| 15 | 0.250076 | 0.250424 | 0.250044 | 0.249456 | 1.000000 |
| 20 | 0.250076 | 0.250424 | 0.250044 | 0.249456 | 1.000000 |

vised benchmarks. As a result, the reported evidence should be interpreted as demonstrating the stability and interpretability of the trust-weighted aggregation mechanism rather than improvements in downstream generation quality.

In addition, the experimental setup considers a fixed number of clients and a single poisoning configuration, where one client is perturbed through corrupted or off-topic retrieval content. While this setting is sufficient to illustrate how retrieval-consistency divergence propagates into reduced trust and aggregation influence, it does not exhaustively cover all adversarial or heterogeneous scenarios that may arise in real-world federated deployments. The trust trajectories therefore reflect controlled conditions rather than worst-case adversarial behavior.

Finally, all hyperparameters governing trust computation and smoothing are fixed a priori and are not tuned against performance objectives. This design choice avoids introducing additional degrees of freedom but also means that the sensitivity of the trust dynamics to parameters such as the EMA coefficient $\rho$, the clipping bounds $[\underline{s}, \overline{s}]$, and the retrieval depth $K$ is not explored in this study. Investigating these sensitivities, as well as extending the evaluation to labeled benchmarks and comparative baselines, is left for future work.

### *4.7. Reproducibility and Artifact Availability*

To support transparency and independent verification, we provide a public repository containing the scripts and logged artifacts used to generate the reported trust and aggregation results.

## 5. Conclusion

This paper introduced a reproducible, label-free trust-weighted aggregation mechanism for federated fine-tuning in retrieval-augmented generation systems. The core idea is to estimate client reliability from retrieval-consistency signals rather than from supervised task labels or attack annotations. By constructing per-client retrieval-distribution histograms under a fixed probe prompt set and quantifying deviations via Jensen–Shannon divergence, the proposed trust score provides an interpretable indicator of whether a client's retrieval behavior remains aligned with the population reference. Exponential mapping and EMA smoothing yield a stable trust trajectory over rounds, and a normalized trust-to-weight mapping translates reliability estimates into aggregation influence without introducing hard exclusion thresholds.

The reported results demonstrate that the trust estimator remains stable across rounds and produces consistent separation between benign and perturbed client behavior under the defined poisoning protocol. Trust trajectories remain near the upper range for benign clients, while the poisoned client converges to a lower operating point, leading to systematic down-weighting in the global update. Because all reported curves and statistics are generated from logged artifacts, the experimental evidence is directly auditable and can be re-created from the released pipeline without manual post-processing.

The study is intentionally scoped to stability and interpretability of trust dynamics in a label-free setting. Accordingly, the results do not claim improvements in downstream answer

accuracy or generation quality, and no supervised benchmark comparisons are reported. Future work will extend the evaluation to labeled RAG benchmarks and additional adversarial settings, including varying poisoning intensities, multiple simultaneously unreliable clients, and stress tests on larger client populations. These extensions will enable a more comprehensive assessment of how retrieval-consistency-based trust interacts with task-level performance while preserving reproducibility and transparency. We emphasize that the reported results validate the proposed trust-to-weight mechanism and its stability under controlled retrieval-corpus poisoning. Standard FL baselines such as FedAvg operate without a trust signal and therefore do not yield directly comparable trust trajectories. Moreover, task-level benchmark comparisons require labeled QA datasets and generation evaluation protocols, which are beyond the label-free scope of this study and are reserved for future work.

*Future work: downstream task evaluation.* A natural extension of this work is to couple the proposed trust-weighted aggregation with downstream RAG evaluation on labeled benchmarks under poisoning. This includes comparing FedAvg and robust aggregation methods against the proposed weighting under identical retrieval indices and poisoning rates, and reporting generation metrics (e.g., answer correctness/faithfulness) with uncertainty estimates over multiple random seeds. Importantly, such an evaluation requires a controlled labeled QA protocol and standardized prompting and scoring, which are outside the scope of the present label-free trust study.

## References


[1] R. Sepehrzad, H. Hosseinalibeiki, N. Taghinezhad, N. Khosravi, A. al Durra, and M. S. Sadabadi, "Enhanced cyber-resilience in flexible energy markets for microgrids: A trust-aware federated deep reinforcement learning framework," *Applied Energy*, vol. 406, p. 127282, 2026, doi: 10.1016/j.apenergy.2025.127282.

[2] H. Hosseinalibeiki and R. Sepehrzad, "Hybrid LLM-Based Emergency Management with Constraint-Aware Edge Deployment in Urban Mobility Networks," SSRN Electronic Journal, 2026. Available: https://doi.org/10.2139/ssrn.5265396

[3] X. Li *et al.*, "RLFE-IDS: A framework of Intrusion Detection System based on Retrieval Augmented Generation and Large Language Model," *Comput. Netw.*, p. 111341, 2025.

[4] W. Tang *et al.*, "AgentBuilder: Automating agent creation via large language model-driven systems," *Neurocomputing*, p. 130476, 2025.

[5] F. A. Noor *et al.*, "Towards collaborative fair federated distillation," *Eng. Appl. Artif. Intell.*, vol. 137, p. 109216, 2024.

[6] S. Niu *et al.*, "Lightweight verifiable federated learning based on matrix theory," *Neurocomputing*, p. 130579, 2025.

[7] X.-Y. Liu *et al.*, "Differentially private low-rank adaptation of large language model using federated learning," *ACM Trans. Manag. Inf. Syst.*, vol. 16, no. 2, pp. 1–24, 2025.

[8] X. Li *et al.*, "Building A Coding Assistant via the Retrieval-Augmented Language Model," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–25, 2025.

[9] Y. Ahn *et al.*, "Retrieval-augmented response generation for knowledge-grounded conversation in the wild," *IEEE Access*, vol. 10, pp. 131374–131385, 2022.

[10] B. Saha, U. Saha, and M. Z. Malik, "Advancing Retrieval-Augmented Generation with Inverted Question Matching for Enhanced QA Performance," *IEEE Access*, 2024.

[11] Y. Fu *et al.*, "Cue RAG: Dynamic multi-output cue memory under H framework for retrieval-augmented generation," *Neurocomputing*, p. 130235, 2025.

[12] Y. Wang, Y. Wang, and B. Yin, "DCFT: Dependency-aware continual learning fine-tuning for sparse LLMs," *Neurocomputing*, vol. 636, p. 129897, 2025.

[13] R. Sepehrzad, N. Khosravi, M. Moafi, H. Hosseinalibeiki, A. al Durra, and M. S. Sadabadi, "A Safe, Secure, and Uncertainty-Aware Demand Response Architecture Using LLM-Guided Hierarchical Multi-Agent Deep Reinforcement Learning in Integrated Energy System," SSRN Electronic Journal, 2026. Available: https://doi.org/10.2139/ssrn.6025576

[14] H. Hosseinalibeiki, R. Sepehrzad, M. Heidari *et al.*, "Blockchain-Governed Federated Intelligence: A Formal Model for Trust-Aware and Auditable Collaborative Learning," *Research Square*, Preprint (Version 1), Feb. 10, 2026. doi: 10.21203/rs.3.rs-8779446/v1.

[15] A. Gaeta, H. Hosseinalibeiki, V. Loia, and F. Orciuoli, "Real-Time Trust-Weighted Consensus and Auditing of LLMs for Secure Propaganda Detection," SSRN Electronic Journal, 2026. Available: https://doi.org/10.2139/ssrn.5817766

[16] R. Sepehrzad, H. Hosseinalibeiki, M. S. Sadabadi, and A. al Durra, "Microgrid secure pricing and real-time flexible energy market integrated with trust-based demand response: A multi-layer FDRL architecture," *International Journal of Electrical Power & Energy Systems*, vol. 174, p. 111536, 2026, doi: 10.1016/j.ijepes.2025.111536.

[17] T. ValizadehAslani *et al.*, "Two-stage fine-tuning with ChatGPT data augmentation for learning class-imbalanced data," *Neurocomputing*, vol. 592, p. 127801, 2024.

[18] H. Xu *et al.*, "Trust region policy optimization via entropy regularization for Kullback–Leibler divergence constraint," *Neurocomputing*, vol. 589, p. 127716, 2024.

[19] R. Sepehrzad, M. Moafi, N. Khosravi, H. Hosseinalibeiki, A. al Durra, and M. S. Sadabadi, "Retail-Informed Secure Wholesale Energy Market and Bid Generation with Projection Clearing Exchange for Networked Microgrids: A Large Language Model-Assisted Multi-Agent Deep Reinforcement Learning Approach," SSRN Electronic Journal, 2026. Available: https://doi.org/10.2139/ssrn.5991242

[20] X. Lu, L. Cao, and X. Du, "Dynamic Control Method for Tenants' Sensitive Information Flow Based on Virtual Boundary Recognition," *IEEE Access*, vol. 8, pp. 162548–162568, 2020, doi: 10.1109/ACCESS.2020.3021415.

[21] J. Huang *et al.*, "Layered Query Retrieval: An Adaptive Framework for Retrieval-Augmented Generation in Complex Question Answering for Large Language Models," *Appl. Sci.*, vol. 14, no. 23, p. 11014, 2024, doi: 10.3390/app142311014.

[22] Y. Zheng *et al.*, "Aggregation Service for Federated Learning: An Efficient, Secure, and More Resilient Realization," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 988–1001, Mar.-Apr. 2023, doi: 10.1109/TDSC.2022.3146448.

[23] C. Palihawadana *et al.*, "FedSim: Similarity guided model aggregation for Federated Learning," *Neurocomputing*, vol. 483, pp. 432–445, 2022.

[24] M. Rospocher, F. Corcoglioniti, and M. Dragoni, "Boosting document retrieval with knowledge extraction and linked data," *Semant. Web*, vol. 10, no. 4, pp. 753–778, 2019.

[25] L. Su, J. Xu, and P. Yang, "Global Convergence of Federated Learning for Mixed Regression," *IEEE Trans. Inf. Theory*, vol. 70, no. 9, pp. 6391–6411, Sep. 2024, doi: 10.1109/TIT.2024.3425758.

[26] N. Mylonas, I. Mollas, and G. Tsoumakas, "An attention matrix for every decision: Faithfulness-based arbitration among multiple attention-based interpretations of transformers in text classification," *Data Min. Knowl. Discov.*, vol. 38, no. 1, pp. 128–153, 2024.

[27] Y. Jiang *et al.*, "Model Pruning Enables Efficient Federated Learning on Edge Devices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10374–10386, Dec. 2023, doi: 10.1109/TNNLS.2022.3166101.

[28] Y. S. Soekamto, L. C. Limanjaya, Y. K. Purwanto, and D.-K. Kang, "From Queries to Courses: SKYRAG's Revolution in Learning Path Generation via Keyword-Based Document Retrieval," *IEEE Access*, vol. 13, pp. 21434–21455, 2025, doi: 10.1109/ACCESS.2025.3535618.

[29] Y. Tao *et al.*, "FLFT: A Large-Scale Pre-Training Model Distributed Fine-Tuning Method That Integrates Federated Learning Strategies," *IEEE Access*, vol. 13, pp. 56439–56453, 2025, doi: 10.1109/ACCESS.2025.3549819.

[30] Y. Xia *et al.*, "FCLLM-DT: Empowering Federated Continual Learning With Large Language Models for Digital-Twin-Based Industrial IoT," *IEEE Internet of Things Journal*, vol. 12, no. 6, pp. 6070–6081, 15 Mar. 2025, doi: 10.1109/JIOT.2024.3510553.

[31] Wikipedia, "Wikipedia: The Free Encyclopedia," [Online]. Available: https://www.wikipedia.org.

[32] PubMed, "PubMed Abstracts," [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/.

[33] StackExchange, "StackExchange Data Dump," [Online]. Available: https://archive.org/details/stackexchange.

[34] FAISS, "FAISS: A Library for Efficient Similarity Search and Clustering of Dense Vectors," Facebook AI Research, [Online]. Available: https://github.com/facebookresearch/faiss.

[35] G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, 1949.

**Algorithm 1** Fed-RAG+: Federated Retrieval-Augmented Fine-Tuning with Retrieval-Consistency Trust Weighting

**Input:** Initial global model $\Phi^{(0)}$, client set $\{C_i\}_{i=1}^{M}$, communication rounds $T$, probe prompt set $\mathcal{P}$, top-$K$ retrieval depth $K$, EMA parameter $\rho$, clipping bounds $[\underline{s}, \overline{s}]$
**for** $t = 1$ to $T$ **do**
  **Server broadcasts:** $\Phi^{(t-1)}$ to participating clients
  **for** each participating client $C_i$ **do**
    **(A) Local RAG fine-tuning:** initialize $\Phi_i^{(t)} \leftarrow \Phi^{(t-1)}$
    **for** each local minibatch $x \in \mathcal{D}_i$ **do**
      $r \leftarrow \text{retrieve}(x, \mathcal{R}_i; K)$
      $z \leftarrow \text{concat}(x, \text{Enc}_R(r))$
      compute local loss on $z$ (task loss plus the regularizer defined in Sec. 3.1)
      update $\Phi_i^{(t)}$ by one optimizer step
    **end for**
    **(B) Build retrieval histogram for trust:** initialize $h_i^{(t)} \leftarrow \mathbf{0} \in \mathbb{R}^B$
    **for** each probe prompt $p \in \mathcal{P}$ **do**
      $\mathcal{I} \leftarrow \text{retrieve_ids}(p, \mathcal{R}_i; K)$ ▷ top-$K$ retrieved chunk IDs
      **for** each retrieved ID $u \in \mathcal{I}$ **do**
        $b \leftarrow \text{bucket}(u)$ ▷ deterministic mapping to bucket index $b \in \{1, \dots, B\}$
        $h_{i,b}^{(t)} \leftarrow h_{i,b}^{(t)} + 1$
      **end for**
    **end for**
    **Client sends:** local model update $\Phi_i^{(t)}$ and histogram $h_i^{(t)}$ to the server
  **end for**
  **(C) Server-side trust computation:**
  $h_{\text{g}}^{(t)} \leftarrow \frac{1}{M} \sum_{i=1}^{M} h_i^{(t)}$ ▷ Eq. (12)
  **for** each participating client $C_i$ **do**
    $\text{jsd}_i^{(t)} \leftarrow \text{JSD}(h_i^{(t)}, h_{\text{g}}^{(t)})$ ▷ Eq. (13)
    $s_i^{(t)} \leftarrow \exp\big(-\text{jsd}_i^{(t)}\big)$ ▷ Eq. (14)
    $\tilde{s}_i^{(t)} \leftarrow \rho\, \tilde{s}_i^{(t-1)} + (1-\rho)\, s_i^{(t)}$ ▷ Eq. (15), with $\tilde{s}_i^{(0)} = 1$
    $\bar{s}_i^{(t)} \leftarrow \text{clip}(\tilde{s}_i^{(t)}, \underline{s}, \overline{s})$
  **end for**
  $w_i^{(t)} \leftarrow \bar{s}_i^{(t)} \Big/ \sum_{m=1}^{M} \bar{s}_m^{(t)}$ for all $i$ ▷ Eq. (16), $\sum_i w_i^{(t)} = 1$
  **(D) Trust-weighted aggregation:**
  $\Phi^{(t)} \leftarrow \sum_{i=1}^{M} w_i^{(t)}\, \Phi_i^{(t)}$
**end for**
**Output:** Final global model $\Phi^{(T)}$