TOWARDS PRACTICAL FHE-FEDERATED LLMS TRAINING: FHE-COMPATIBLE COMPRESSION TECHNIQUES AND VERSATILE ADVANCED AGGREGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) enables collaborative LLM training on decentralized data. However, plaintext updates remain vulnerable to privacy leakage, such as gradient inversion. Fully Homomorphic Encryption (FHE) can secure aggregation, but scaling FL-FHE to LLMs introduces two challenges. First, ciphertext expansion exacerbates communication overhead. Prior work applies Top-k sparsification to compress updates, but it conflicts with FHE's SIMD packing. This forces a trade-off: coarse ciphertext-level sparsification prunes excess information, while element-wise sparsification unions Top-k that sacrifice compression efficiency. Second, costly non-linear FHE operations limit aggregation to linear FedAvg, excluding advanced aggregators essential for statistical heterogeneity. In this work, we present **FUSE** (FHE-compatible Unified Sparse-Encode), a unified, novel FHE-native compression suite with two core components. *(i) FUSE-SPA* resolves the SIMD-sparsification conflict through encrypted importance voting. This achieves fine-grained Top-k sparsification at exact cardinality, with a provable selection fidelity bound. *(ii) FUSE-RDX* avoids pruning-induced information loss by preserving full density through radix encoding. It introduces an aggregation-aware radix condition to ensure aggregation correctness. To extend FL-FHE beyond FedAvg for statistical heterogeneity, we also design a homomorphic Norm Bounding circuit that reformulates costly non-linear comparison into polynomial arithmetic. Experiments demonstrate that FUSE achieves communication reductions of $3.46\times$ for GPT-2 and $3.87\times$ for Llama-7B. This yields end-to-end WAN latency speedups of $3.24\times$ and $3.88\times$ with $< 1\%$ perplexity degradation relative to FHE baselines.

1 INTRODUCTION

Federated Learning (FL) has emerged as a paradigm for collaborative fine-tuning of Large Language Models (LLMs) on decentralized proprietary data, enabling participants to train shared models while preserving data locality (Ye et al., 2024; Kuang et al., 2024). However, transmitting plaintext updates to an untrusted central server for aggregation risks leaking sensitive client information through update-based attacks (e.g., gradient inversion (Petrov et al., 2024; Zhu et al., 2019)). In regulated, high-stakes domains (e.g., healthcare, finance), where data privacy has become a critical regulatory mandate, securing federated LLM training beyond data locality becomes essential. Several protection paradigms have been used to secure FL. Differential privacy (DP) perturbs updates with calibrated noise (Sun et al., 2024; Tran et al., 2025), offering statistical guarantees. Yet it incurs a privacy–utility trade-off that sharpens with composition loss over rounds (Du et al., 2023; Wen et al., 2025; Xu et al., 2025). Secure aggregation (SecAgg) relies on additive secret masking (Ma et al., 2025; So et al., 2022). This limits advanced, non-linear aggregation for non-IID settings, and eventually exposing the plaintext aggregate to the server. Fully Homomorphic Encryption (FHE) addresses these limitations by providing cryptographic confidentiality, enabling computation on encrypted updates throughout the process (Sav et al., 2021; 2023; Pan et al., 2024; Xue et al., 2025). Despite these advantages, scaling the FL-FHE paradigms to LLMs reveals two key challenges:

Prohibitive Communication Overhead. Communication efficiency has been a central challenge in federated LLM training (Qin et al., 2024; Shu et al., 2025; Meng et al., 2026), and FHE protection exacerbates it due to ciphertext expansion. Prior FL-FHE work has explored several strategies to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

mitigate this overhead. The first narrows *what is encrypted*: Selective HE (SHE) encrypts only update subsets deemed sensitive, yet transmitting unselected updates in plaintext relaxes the threat model compared to full encryption (Jin et al., 2023; Liu et al., 2026; Hu & Li, 2025). The second narrows *what is trained*: parameter-efficient fine-tuning (e.g., LoRA) reduces trainable parameter counts (Wang et al., 2024; Bai et al., 2024). However, even lightweight updates (e.g., $\approx 0.25\%$ parameters of Llama-7B (Ye et al., 2024)) yield multi-gigabyte encrypted updates per client round.

The third narrows *what is transmitted*: Some studies compress the updates before encryption through quantized integer packing (Zhang et al., 2020) or dictionary decomposition and pruning (Xue et al., 2025). However, their methods are designed on linear aggregation, thereby limiting advanced aggregation under FHE. Alternatively, Top-k sparsification selects the k most significant updates independently of the aggregator, thus preserving algorithmic flexibility. Yet its native adoption under FHE is non-trivial. Top-k sparsification requires slot-wise index alignment, but this conflicts with the vectorized batching of FHE with cryptographically opaque slots. Hence, compact FHE packing disrupts the slot alignment essential for correct homomorphic sparse aggregation. Existing studies attempt to reconcile this conflict but force a trade-off: coarse-level sparsification degrades convergence due to excess pruning (Yan et al., 2024), while element-wise sparsification sacrifices compression ratio to union disparate Top-k indices (Wei et al., 2026). These gaps motivate FHE-native compression that balances granularity with compression efficiency, while preserving correctness and flexibility to support advanced aggregators.

Table 1: Qualitative comparison of privacy-preserving FL work.

Method	Server Conf.	Non-linear Agg.	Model Scale	Update Compression
SHE-LoRA (Liu et al., 2026)	◐[a]	✗	LLM	–
Lancelot (Jiang et al., 2025)	◐[b]	✓	CNN	–
BatchCrypt (Zhang et al., 2020)	●	✗	CNN/ LSTM	Integer batching *(additive-only HE)*
FedSHE-CQ (Wei et al., 2026)	●	✗	CNN	Union Top-k + quantization *(union exceed k)*
FedPHE (Yan et al., 2024)	●	✗	CNN	Ciphertext-level sparsification *(excess pruning)*
DictPFL (Xue et al., 2025)	●	✗	LLM	Dictionary + history pruning *(mask from past global gradients)*
FUSE (This work)	●	✓	**LLM**	**Element-wise exact-k sparsification + radix encoding** *(consensus approx. local Top-k)*

*Note: **Server conf.** denotes update confidentiality throughout secure aggregation; ●: ciphertext-only, ◐[a]: partial plaintext (SHE), ◐[b]: intermediate decryption via non-colluding party. **Non-linear agg.** indicates support for encrypted non-linear aggregation. **Update compression** denotes HE-compatible compression of updates for aggregation.*

Constrained Algorithmic Versatility in FL-FHE. Many existing FL-FHE implementations adopt FedAvg (McMahan et al., 2017) because FHE supports linear operations efficiently (Xue et al., 2025; Pan et al., 2024; Yan et al., 2024; Wei et al., 2026). However, its simplicity limits its effectiveness under statistical heterogeneity, motivating substantial research into non-IID federated learning (Yan et al., 2025; Bai et al., 2024; Wang et al., 2024). Many advanced algorithms (see Appendix A for more algorithms) addressing this often involve complex, non-linear operations (e.g., comparisons, clipping) that are expensive to compute under FHE. Existing efforts toward encrypted non-linear aggregation fall short by sacrificing different properties. First, prior fully encrypted non-linear aggregation studies were evaluated on small-scale CNNs/RNNs architectures (Sav et al., 2021; 2023; Jiang et al., 2025). Second, recent FL-FHE studies at LLM scale remain confined to linear aggregation (Xue et al., 2025; Mia & Amini, 2025). Third, alternative solutions circumvent the cost of homomorphic non-linear operations by exposing intermediate values to non-colluding servers for plaintext computation (Zhao et al., 2024; Chen et al., 2024). This weakens the threat model compared to a single-server, pure FHE pipeline (Li et al., 2025), forcing a server-privacy trade-off for algorithmic advancement.

Together, these gaps raise a central question: *can communication be reduced without sacrificing algorithmic flexibility for non-IID encrypted federated LLM training?*. To answer this, we demonstrate that FHE-compatible compression and advanced encrypted aggregation can coexist within a single FL-FHE pipeline. Our main contributions are summarized below.

• **FHE-Compatible Communication-Efficient Suite (FUSE).** We propose FUSE, a unified compression suite designed to address the communication bottleneck in secure federated LLM training. FUSE consists of three components: *(i) FUSE-SPA*, an FHE-aware sparsification that achieves unified, fine-grained, exact-cardinality Top-k updates for FHE aggregation through two mechanisms: binned index ranking and a compact index exchange protocol. It is formalized by a selection fidelity bound to quantify ranking consistency, alongside an overflow-prevention threshold B_{max}. *(ii) FUSE-RDX*, a tailored radix encoding scheme that optimizes ciphertext spatial utilization by pairing updates per slot. We derive aggregation-aware radix conditions to ensure overflow-free aggregation, achieving density-

preserving compression without the information loss inherent in sparsification. *(iii) FUSE-Fusion*, an integration of these techniques to compound compression benefits. FUSE achieves significant communication reductions compared to the FHE baseline: reducing Llama-7B updates from 11.87 GB to 3.07 GB (3.87× reduction) and up to 3.46× for GPT-2. This yields round latency speedups of up to 3.88× under WAN environments. All FUSE variants exhibit minimal fidelity impact with <1% perplexity degradation across all evaluated datasets.

• **Versatile Homomorphic Aggregation.** To advance FL-FHE computation beyond FedAvg, we carefully design the homomorphic circuit for Norm Bounding. It enables encrypted norm clipping to stabilize non-IID convergence, a non-linear capability that FedAvg's linear aggregation falls short. We introduce *Composite Sign-Gating*, a homomorphic multiplexer that transforms non-native comparison into logic-gate-inspired arithmetic via composite polynomial approximation. When integrated with FUSE, the round latency for Norm Bounding on Llama-7B reduces from ≈ 35 minutes to ≈ 9 minutes. By leveraging dual-context FHE parameterization, FUSE amplifies the communication efficiency up to 6.59× for Llama-7B on Norm Bounding, from 11.87 GB to 1.80 GB.

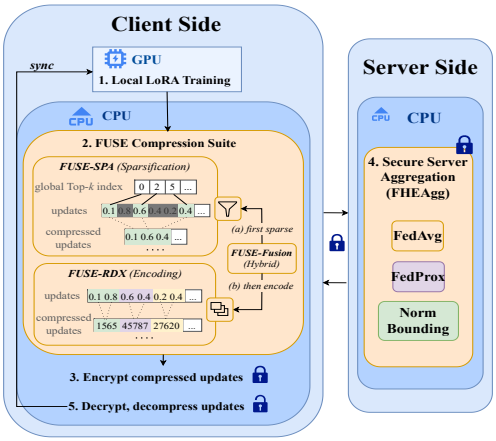


Figure 1: Overall architecture of the proposed FL-FHE with compression process.

• **Architectural Interoperability and Analytical Heuristics.** FUSE offers an interoperable compression architecture fully compatible with orthogonal techniques like Low-Rank Adaptation (LoRA), establishing a practical pathway for FL-FHE LLM training. Analysis also reveals that the optimal FL-FHE compression is governed by CKKS computational overhead associated with aggregation complexity rather than model size.

2 PRELIMINARY

Fully Homomorphic Encryption (FHE) enables computation directly on ciphertexts without decryption. Among existing schemes, the Cheon–Kim–Kim–Song (CKKS) scheme (Cheon et al., 2017) supports approximate fixed-point arithmetic over real numbers, well-suited for federated LLMs with real-valued model updates (refer Appendix A for more background). CKKS supports SIMD (Single Instruction, Multiple Data) by packing up to $N/2$ values into a single ciphertext. It is defined by the following algorithms:

• $\mathsf{KeyGen}(1^\lambda) \to (sk, pk, evk)$: Given a security parameter λ, outputs a secret key sk, a public key pk, and an evaluation key evk.

• $\mathsf{Ecd}(z, \Delta_{\mathrm{CKKS}}) := m$: Encodes a message vector $z \in \mathbb{R}^{N/2}$ to a plaintext polynomial m in a ring of degree N, scaled by Δ_{CKKS}.

• $\mathsf{Enc}_{pk}(m) \to ct$: Encrypts a plaintext polynomial m to a ciphertext ct using the public key pk.

• $\mathsf{Eval}_{evk}(f, ct) := ct_f$: Applies a function f to the encrypted inputs using evk. In our construction, f encompasses standard arithmetic $(+, \times)$ and polynomial approximations $(\mathsf{sgn}_{\mathrm{poly}})$.

• $\mathsf{Dec}_{sk}(ct) := m'$: Recovers the plaintext polynomial m' from ct using the secret key sk.

• $\mathsf{Dcd}(m', \Delta_{\mathrm{CKKS}}) := z'$: Decodes the plaintext polynomial m' back into a real-valued vector $z' \in \mathbb{R}^{N/2}$.

3 METHODOLOGY

Overview Workflow. Figure 1 depicts the proposed FUSE workflow. In each round, selected clients train locally using LoRA (Hu et al., 2022), aligning with the established paradigm for distributed LLM

training (Ye et al., 2024; Kuang et al., 2024). The clients then apply FUSE compression: FUSE-SPA (fine-grained Top-k sparsification), FUSE-RDX (density-preserving encoding), or FUSE-Fusion (hybrid compression). The compressed updates are encrypted and sent to the server for homomorphic aggregation (FedAvg, FedProx, Norm Bounding) to derive a global model. Finally, clients receive, decrypt, decompress, and synchronize the updates for subsequent rounds.

Threat Model. We consider a federated semi-honest (honest-but-curious) environment, where the server adheres to the protocol but attempts to infer sensitive client information. Through FHE, the server aggregates on ciphertexts and never gains access to plaintext updates, thereby precluding the reconstruction of individual local gradients. In FUSE-SPA, clients decrypt only the aggregated importance score map s_{glob} to derive consensus Top-k indices. This is a deliberate, bounded disclosure of coarse, index-level saliency under the semi-honest, non-collusion setting (refer to Appendix B.2 for detailed privacy analysis). This provides robust input privacy, while complementary mechanisms for output privacy (e.g., differential privacy) remain orthogonal and integrable. Under this semi-honest setting, clients are assumed to follow the prescribed federation protocol. While the model ensures strict confidentiality, enforcing computational integrity against potential active protocol deviations remains an orthogonal research challenge. To support deployment scenarios that demand stronger verification of client-side operations, it can be extended with complementary integrity mechanisms (e.g., TEEs) without modifying the core privacy-preserving aggregation.

3.1 FHE-COMPATIBLE UNIFIED SPARSE-ENCODE (FUSE)

FUSE Sparse-Pack-Aggregate (FUSE-SPA). Unlike other privacy-preserving FL compression schemes, Top-k sparsification selects which coordinates are transmitted without constraining the aggregation rule. However, implementing it with encryption-based FL presents a structural conflict: FHE leverages SIMD packing to batch values into position-static ciphertext slots, while sparsification is index-dynamic, as clients select disparate parameter subsets. Hence, the server cannot correctly sum them from different clients without knowing their index alignment. This forces two unsatisfactory options: *(1) Compact Packing,* transmitting only Top-k values in a densely packed ciphertext achieves compression but breaks aggregation correctness; and *(2) Naive Alignment*, zero-padding non-selected Top-k indices preserves correctness but nullifies compression.

Algorithm 1 FUSE-SPA

Input: $\{u_c\}_{c\in\mathcal{C}}$: Local updates of client c ($u_c \in \mathbb{R}^d$); n_c: Data weights; l: Slot capacity; k: Top-k ratio; B: number of bins.
Output: Aggregated global update Δu_{glob}.
Determine digit-width per score $ISlot$ by (2);
Derive packing factor $\rho \leftarrow \lfloor l/ISlot \rfloor$;
for each client $c \in C$ **in parallel do**
  Find Top-k on u_c with ratio k;
  Find bin width $w_c \leftarrow (k_c^{\max} - k_c^{\min})/B$;
  Construct B intervals $\{Q_b\}_{b=1}^{B}$ partitioning $[k_c^{\min}, k_c^{\max}]$;
  $s_{c,j} \leftarrow b$ if $|u_{c,j}| \in Q_b$, else $s_{c,j} \leftarrow 0$, for $j \in \{1,\ldots,d\}$; $s_c \leftarrow (s_{c,1},\ldots,s_{c,d}) \in \{0,\ldots,B\}^d$;
  Partition $\mathbf{s}_c$ into segments of ρ scores;
  Pack each segment via $ISlot$-wide digit-shifting; CKKS encode and encrypt into $\mathbf{ct}_c^{idx}$;
end for
Server Index Agg: $\mathbf{ct}_{idx} \leftarrow \sum_{c\in C} \mathbf{ct}_c^{idx}$;
for each client $c \in C$ **in parallel do**
  $z_{glob} \leftarrow \lfloor \mathsf{Dcd}(\mathsf{Dec}_{sk}(ct_{\text{idx}}), \Delta_{\text{CKKS}}) \rceil$;
  $s_{glob} \leftarrow \mathsf{Unpack}(z_{glob})$;
  Select $\hat{\mathbf{u}}_c \leftarrow \{u_{c,j}\}_{j\in\mathcal{I}_{glob}}$, $\mathcal{I}_{glob}$: the sorted, set of k largest indices in s_{glob};
  $ct_c \leftarrow \mathsf{Enc}_{pk}(\mathsf{Ecd}(\hat{u}_c, \Delta_{\text{CKKS}}))$;
end for
Server Update Agg: $\mathbf{ct}_{glob} \leftarrow \mathsf{FHEAgg}(\{\mathbf{ct}_c\}_{c\in\mathcal{C}}, \{n_c\}_{c\in\mathcal{C}})$; (Alg. 3)
return $\Delta u_{glob} \leftarrow \mathsf{Dcd}(\mathsf{Dec}_{sk}(\mathbf{ct}_{glob}), \Delta_{\text{CKKS}})$

Existing solutions provide only partial remedies. FedPHE (Yan et al., 2024) pruned coarsely at the ciphertext-block level via mean thresholds. This lack of granularity incurs information loss that prolongs convergence, which is critical for large models with costly training epochs. FedSHE-CQ (Wei et al., 2026) maintains alignment by homomorphically summing CPIQP-packed masks by retaining every selected index from any client. However, it unions all clients' Top-k sets, expanding the final parameter set far beyond the target sparsity k. Consequently, achieving truly fine-grained Top-k sparsification under FL-FHE remains a non-trivial challenge. To address this, we propose FUSE-SPA that decouples index selection from aggregation by reframing Top-k sparsification as an encrypted importance-voting problem. This ensures that all clients converge on a consensus set of Top-k before encrypted aggregation. Alg. 1 (refer to Table 3 for notations) depicts the execution flow, introducing two core mechanisms:

• *Binned Index Ranking.* This mechanism quantifies update significance into bounded importance scores to derive a consensus Top-k update set for homomorphic aggregation. Direct ordinal ranking (ranks 1 to k) preserves optimal intra-client importance but risks arithmetic overflow during homomorphic summation. FUSE-SPA mitigates this by mapping the local Top-k range into B uniform bins of width $w_c \leftarrow (k_c^{\max} - k_c^{\min})/B$, assigning each update bounded score $s_{c,j} \in \{0, \ldots, B\}$ corresponding to its bin, while indices not in Top-k receive a score of 0. This produces an importance vector in which higher scores indicate greater local significance (Alg. 1, lines 6-9). This mechanism ensures overflow safety during compact index exchange (Alg. 1, lines 10-13) while enabling the server to synthesize a *global importance map* that captures cross-client contribution density. Finally, clients sort and select from the global importance map to synchronize a unified Top-k set (Alg. 1, lines 14-19), enforcing the k-cardinality budget while maximizing representational fidelity. The selectivity of local binning introduces a controlled selection error. We formalize this via a *Top-k selection fidelity bound*, which defines the maximum possible magnitude by which a rejected gradient index can exceed a selected one. Appendix B.1 details the formal proof.

Theorem 1 (Top-k Selection Fidelity Bound)**.** *For each client $c \in \mathcal{C}$, let $|u_{c,j}|$ denote the true update magnitude of index $j \in \{1, \ldots, d\}$ and $s_{c,j} \in \{0, \ldots, B\}$ its binned importance score, where $s_{c,j} = 0$ denotes exclusion from the local Top-k set. Let $w_c = (k_c^{\max} - k_c^{\min})/B$ be the local bin width. For any selected index i and rejected index i' satisfying $\sum_{c\in\mathcal{C}} s_{c,i} \geq \sum_{c\in\mathcal{C}} s_{c,i'}$, the true aggregate magnitudes $T_i = \sum_{c\in\mathcal{C}} |u_{c,i}|$ and $T_{i'} = \sum_{c\in\mathcal{C}} |u_{c,i'}|$ satisfy:*

$$T_{i'} - T_i \;\leq\; \sum_{c\in\mathcal{C}} \max\big(w_c,\, k_c^{\min}\big) \;+\; \underbrace{\sum_{c\in\mathcal{C}} w_c \,(s_{c,i'} - s_{c,i})}_{\textit{Heterogeneity Gap}}. \tag{1}$$

This bound consists of two terms: *(1)* cumulative discretization width across all clients, and *(2)* a heterogeneity gap that quantifies ranking discrepancies from aggregating unweighted scores across clients of disparate local scale. As B increases, the discretization error shrinks and the scores more precisely represent true magnitudes, allowing the aggregate selection to converge to the near-global Top-k. This enables a tunable trade-off between selection fidelity and communication efficiency.

• *Compact Index Exchange Protocol.* To aggregate scores under FHE with minimal overhead, we design a dynamic digit-packing technique. This technique encodes multiple scores into a single slot using a $base$ representation, ensuring arithmetic isolation between indices. To ensure overflow safety during homomorphic summation across C clients, we define the digit width per importance score as:

$$ISlot = \lceil \log_D(C) \rceil + 1 \tag{2}$$

Given per-slot decimal digit capacity l (expressed in base-D digits), we derive a packing factor $\rho = \lfloor l/ISlot \rfloor$ to compress ρ independent scores into a single slot. This bounds the maximum per-index score to $B_{max} := \lfloor D^{ISlot}/C \rfloor - 1$, ensuring that the global sum satisfies $B_{max} \cdot C < D^{\text{ISlot}}$, preventing carry-over errors between index placeholders. We adopt $D = 10$ as a design standard to provide dynamic range for supporting larger B_{max} values, preserving selection fidelity. The slot digit capacity $l = 16$ is determined by the CKKS parameters to fully utilize the capacity per ciphertext slot. The efficacy of FUSE-SPA is governed by the arithmetic feasibility bounds and cryptographic parameters. This requirement induces a representation-efficiency trade-off: increasing B improves selection fidelity (Theorem 1), but requires a larger $ISlot$, which reduce the packing factor ρ and increases index exchange overhead. This capacity is also tied to the CKKS parameterization, where the polynomial degree (N) determines the total available slots per ciphertext.

FUSE Radix Encoding Scheme (FUSE-RDX). While FUSE-SPA optimizes communication through sparsification, it inevitably discards update information, which may impede convergence in LLM training. To resolve this, we propose *FUSE-RDX*, an information-preserving compression technique that doubles data density without discarding any updates. FUSE-RDX is inspired by Kronecker Substitution (KS) (Harvey, 2009), which uses mixed-radix encoding for coefficient packing (Bos et al., 2022; Wang et al., 2022). However, direct application of KS to FL is non-trivial: standard KS assumes exact integer convolution, whereas FL-LLM workflows operate on slot-wise aggregation over floating-point values. FUSE-RDX bridges this gap with an *aggregation-aware radix encoding* protocol depicted in Alg. 2. It first maps floating-point updates into the integer domain via a tunable scaling factor Δ_{FUSE}. The core innovation is the derivation of the aggregation-aware radix condition, where the encoding base B_{rdx} satisfies:

$$B_{rdx} > 2 \cdot A \cdot W_{\text{total}} \tag{3}$$

where A is the largest absolute magnitude after Δ_{FUSE} scaling, and W_{total} is introduced to capture the worst-case aggregation multiplier from the aggregation (e.g., $W_{\text{total}} \approx 1$ for FedAvg). By projecting pairs of updates into a packed integer $z_c = \tilde{u}_{c,a} \cdot B_{rdx} + \tilde{u}_{c,b}$, FUSE-RDX isolates updates into higher- and lower-order positions to ensure integers do not overlap. The encoding base must satisfy the radix constraint while still within the CKKS scaling limit (e.g., 2^{60} in OpenFHE) to guarantee error-free integer packing for aggregation. The encoded integers are encrypted and sent to the server for aggregation. Finally, the aggregated results are decoded through base-B_{rdx} decomposition to recover components, which are then rescaled by $\Delta_{\text{FUSE}}^{-1}$ to restore the floating-point values.

FUSE-Fusion. A hybrid compression that combines FUSE-SPA and FUSE-RDX is proposed. FUSE-SPA first performs sparsification to derive global Top-k parameters, followed by FUSE-RDX aggregation-aware radix projection to derive the final compression set. This synergistic integration delivers a compounding compression effect by combining dimensionality reduction with spatial packing. The correctness of FUSE-Fusion requires satisfying both the index overflow-protection constraint of FUSE-SPA and the radix condition of FUSE-RDX under the given CKKS parameterization, ensuring error-free homomorphic aggregation. Appendix C detailed the noise analysis for FUSE.

Algorithm 2 FUSE-RDX

Input: $\{u_c\}_{c\in\mathcal{C}}$: Local updates of client c in C client subset; n_c: Data weights; Δ_{FUSE}: RDX scaling factor; W_{total}: Worst-case multiplier.
Output: Aggregated global update Δu_{glob}.
Derive B_{rdx} as (3);
for each client $c \in \mathcal{C}$ **do**
  Scale input to an integer $\tilde{u}_c \leftarrow \lfloor u_c \cdot \Delta_{\text{FUSE}} \rfloor$;
  Construct pairs $(\tilde{u}_{c,a}, \tilde{u}_{c,b})$ by grouping adjacent elements in $\tilde{u}_c$;
  Project pairs into large integer $z_c \leftarrow \tilde{u}_{c,a} \cdot B_{rdx} + \tilde{u}_{c,b}$;
  $\mathbf{ct}_c \leftarrow \mathsf{Enc}_{pk}(\mathsf{Ecd}(z_c, \Delta_{\text{CKKS}}))$;
end for
Server Update Agg: $\mathbf{ct}_{glob} \leftarrow \mathsf{FHEAgg}(\{\mathbf{ct}_c\}_{c\in\mathcal{C}}, \{n_c\}_{c\in\mathcal{C}})$; (Alg. 3)
for each client $c \in \mathcal{C}$ **do**
  $Z_{agg} \leftarrow \mathsf{Dcd}(\mathsf{Dec}_{sk}(\mathbf{ct}_{glob}), \Delta_{\text{CKKS}})$;
  Recover signed components $U_b \leftarrow Z_{agg} \pmod{B_{rdx}} \in (-\frac{B_{rdx}}{2}, \frac{B_{rdx}}{2}]$ and $U_a \leftarrow (Z_{agg} - U_b)/B_{rdx}$;
end for
return $\Delta u_{glob} \leftarrow$ Rescale all (U_a, U_b) with $\Delta_{\text{FUSE}}^{-1}$ and W_{total}

3.2 Expanding FHE-Compatible Advanced Aggregation

FedAvg dominates in FL-FHE systems for its linearity but struggles under statistical heterogeneity. Advanced encrypted aggregators remain challenging under FHE because they require expensive non-linear operations (e.g., comparisons, square roots). Existing work resolves this tension only partially: FHE-based non-linear aggregation is demonstrated largely at CNN/RNN scale (Sav et al., 2023; Jiang et al., 2025), while scaling to LLM fall back to linear aggregation (Xue et al., 2025; Mia & Amini, 2025). Alternatively, recent studies sidestep homomorphic non-linearity by relaxing the threat model through intermediate decryption (Chen et al., 2024). To close these gaps, this work extends FL-FHE beyond FedAvg, supporting FedProx and Norm Bounding, entirely within a single-server FHE pipeline.

Encrypted FedProx. This stabilizes non-IID convergence via a proximal regularizer $\frac{\mu}{2}\|u_c - u_{glob}\|_2^2$. After local training, clients apply FUSE and encrypt their updates for server-side weighted aggregation (Alg. 3, line 12).

Encrypted Norm Bounding. This enforces a magnitude constraint on client updates via server-side clipping under a confidential threshold, stabilizing convergence. However, its homomorphic implementation is non-trivial: it requires conditional clipping based on the ℓ_2-norm, applying a factor of $\{1, \frac{\tau}{\ell_2}\}$ depending on whether the norm exceeds τ that holds in the server. Though conceptually simple, this involves four expensive FHE primitives: *(1) square root* for ℓ_2-norm computation, *(2) division*, *(3) comparison* between ℓ_2-norm and τ threshold, and *(4) conditional clipping* to apply

Algorithm 3 Versatile Server Homomorphic Aggregation (V-SHAgg)

Input: $\{\mathbf{ct}_{u,c}\}_{c\in C}$: Encrypted updates; $\{\mathbf{ct}_{\rho,c}\}_{c\in C}$: Auxiliary Reciprocal Norm; $\{n_c\}$: Data weights; τ: Clipping threshold.
Output: Aggregated global ciphertext $\mathbf{ct}_{glob}$.
if Protocol is *Secure Norm-Bounding* **then**
  for each client $c \in C$ **do**
    $\mathbf{ct}_{x,c} \leftarrow \mathsf{sgn}_{\text{poly}}(\tau^{-1} - \mathbf{ct}_{\rho,c})$;
    $\mathbf{ct}_{\sigma,c} \leftarrow 0.5 \cdot (\mathbf{ct}_{x,c} + 1)$;
    $\mathbf{ct}_{S,c} \leftarrow \mathbf{ct}_{\sigma,c} \cdot (\tau \cdot \mathbf{ct}_{\rho,c} - 1) + 1$;
    $\mathbf{ct}_{g,c} \leftarrow \mathbf{ct}_{S,c} \cdot \mathbf{ct}_{u,c}$;
  end for
  $\mathbf{ct}_{glob} \leftarrow \sum_{c\in C} n_c \cdot \mathbf{ct}_{g,c}$;
else
  $\mathbf{ct}_{glob} \leftarrow \sum_{c\in C} n_c \cdot \mathbf{ct}_{u,c}$;
end if
return $\mathbf{ct}_{glob}$

the factor. All require expensive approximation computation. To compute homomorphic Norm Bounding efficiently, we derive two mechanisms as follows:

• *Composite Sign-Gating.* Emulating "if-else" branching in the encrypted domain requires high-degree polynomial approximations, which are computationally expensive. Thus, we design a *homomorphic multiplexer* that utilizes a composite polynomial approximation to reformulate logical branching into a logic-gate arithmetic circuit as:

$$\mathbf{ct}_{S,c} \approx \mathbf{ct}_{\sigma,c} \cdot (\tau \cdot \mathbf{ct}_{\rho,c} - 1) + 1 \tag{4}$$

where $ct_{\sigma,c}$ is derived from $1/\tau - ct_{\rho,c}$ via a composition of minimax polynomials $f^{(k)}(x)$ (Cheon et al., 2020) approximating the sign-indicator function $[-1, 1] \to [0, 1]$ (Alg 3, lines 5-6). In this circuit, $ct_{\sigma,c} \approx 1$ (when $\ell_2 > \tau$) "activates" the clipping factor, while $ct_{\sigma,c} \approx 0$ "bypasses" it by reducing the expression to 1 (Alg. 3, lines 7–8). This transforms conditional clipping into efficient linear arithmetic under FHE. Each client's encrypted update is multiplied by their clipping factor, and the server performs the weighted-sum aggregation (Alg 3, lines 8-10). To control sign-approximation error that affects the clipping precision, we sharpen the polynomial transition near the boundaries to minimize error propagation. Aggregation across clients may amplify these deviations, but experiments show minimal impact with perplexity degradation $< 1\%$. Error analysis is provided in Appendix D.

• *Auxiliary Reciprocal-Norm (ARN).* Each client computes its reciprocal norm $\rho_c = \frac{1}{\ell_2}$ and transmits it as encrypted auxiliary data along with the encrypted updates. The server consumes encrypted ρ_c only within the homomorphic circuit (Alg. 3, lines 4-8) without decryption. It learns neither $\|u_c\|_2$ nor any client's clipping outcome, bypassing costly homomorphic square-root and division. Under the semi-honest setting, it enables efficient norm-based scaling to stabilize convergence under heterogeneous update settings. For stronger client-side integrity, this approach is orthogonal with trusted execution environment (TEE) for remote attestation on norm computation, proving validity to the server without exposing plaintext.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

We conduct experiments on GPT-2 and Llama-7B, utilizing OpenFedLLM (Ye et al., 2024) for federated orchestration and OpenFHE Al Badawi et al. (2022) as the FHE backend (CKKS, $N = 2^{16}$). It is trained on OpenWebText and evaluated on Lambada, WikiText-103, and One Billion Words (1WB). This study evaluate proposed FUSE across three representative FL algorithms: FedAvg (FA), FedProx (FP), and Norm Bounding (NB). All experiments, including baseline, employ LoRA ($rank = 32$) under non-IID partitioning (Dirichlet, $\alpha = 0.5$) over 150 rounds per experiment. We compare FUSE against a standard FHE-encrypted aggregation baseline (without FUSE), measuring model utility and communication-latency performance on GPT-2 family, and perform scalability experiments on Llama-7B. Detailed setup configurations can be found in Appendix E.

4.1 HOW MUCH COMMUNICATION DOES FUSE REDUCE ACROSS FHE-BASED AGGREGATION?

Communication Efficiency and Cryptographic Synergy. Communication analysis of GPT-2 (Table 2) reveals a trade-off dictated by CKKS parameterization. In lightweight CKKS configurations (FA & FP), FUSE-SPA (50%) achieves $1.57\times$ reduction, outperforming FUSE-RDX ($1.49\times$) due to lower per-ciphertext cost in shallow circuits. However, this trend reverses in NB, where non-linear operations need deeper circuits ($depth = 21$), which inflate ciphertext sizes to ≈ 24.07 MB. This makes FUSE-RDX superior ($1.93\times$ vs. $1.62\times$ for FUSE-SPA-50%). It reduces the overall ciphertext count (FUSE-RDX's without index exchange), yielding large absolute communication reduction gains. This reflects the trade-off between fewer (but larger) ciphertexts and more (but smaller) ciphertexts as CKKS parameters change. By synergizing both primitives, FUSE-Fusion achieves the highest reduction (between $1.14\times$ and $3.46\times$), lowering GPT-2's NB overhead from 833 MB to 242 MB and from 113 MB to 44 MB for FA & FP (see Appendix F).

Dual-context Optimization for High-depth FL-FHE. Prior observation discover that a single CKKS-parameter design forces all ciphertexts to adopt the highest required parameter setting, leading to substantial bandwidth waste. Thus, we leverage a dual CKKS-parameter technique for NB that exploits the linearity of the index-exchange to decouple sparsification from aggregation. We perform index aggregation using a lightweight parameter while reserving a heavy parameter for encrypted aggregation. This reduces index-related ciphertext size by over $7\times$, reducing the total communication

Table 2: Performance of FUSE variants across models and FL algorithms (single CKKS context): communication efficiency and latency analysis.

	k		GPT-2								Llama-7B							
			Comm.	Time (s)				Latency (s)			Comm.	Time (s)				Latency (s)		
			MB	C	S	LAN	WAN	T_{LAN}	T_{WAN}		GB	C	S	LAN	WAN	T_{LAN}	T_{WAN}	
FA	100%	BL	113	1.00×	1.73	1.28	1.81	18.33	4.82	1.00×	21.24	1.00×	1.62	1.00×	35.34	10.06	25.97	259.93	71.37	1.00×	305.33	1.00×
	75%	FS	101	1.11×	2.25	0.5	1.61	16.3	4.36	1.11×	19.05	1.11×	1.41	1.15×	33.54	8.28	22.69	227.10	64.51	1.11×	268.92	1.14×
	75%	FF	82	1.38×	2.35	0.28	1.31	13.29	3.94	1.22×	15.92	1.33×	1.07	1.51×	26.12	3.96	17.10	171.16	47.18	1.51×	201.24	1.52×
	50%	FR	76	1.49×	**1.18**	**0.37**	1.21	12.28	2.76	1.75×	13.83	1.54×	1.07	1.51×	14.75	5.07	17.07	170.87	36.89	1.93×	190.69	1.60×
	50%	FS	72	1.57×	1.85	0.35	1.59	11.7	3.79	1.27×	13.97	1.52×	1.01	1.60×	26.16	5.78	16.21	162.24	48.15	1.48×	194.18	1.57×
	50%	FF	61	1.85×	1.98	0.19	0.96	9.93	3.13	1.54×	12.1	1.76×	0.80	2.03×	21.81	2.72	12.83	128.50	37.36	1.91×	153.03	2.00×
	25%	FS	45	2.51×	1.44	0.18	0.72	7.40	**2.34**	2.06×	**9.02**	2.35×	0.61	2.66×	19.10	3.11	9.72	97.34	31.93	2.24×	119.55	2.55×
	25%	FF	**44**	2.57×	1.74	0.11	0.71	7.25	2.56	1.88×	9.10	2.33×	**0.53**	3.06×	**16.80**	**1.44**	8.50	86.13	**26.74**	2.67×	**104.37**	2.93×
FP	100%	BL	113	1.00×	1.77	1.2	1.81	18.33	4.78	1.00×	21.3	1.00×	1.62	1.00×	34.14	10.87	25.97	259.93	70.98	1.00×	304.94	1.00×
	75%	FS	101	1.11×	2.28	0.49	1.61	16.3	4.38	1.09×	19.07	1.12×	1.41	1.15×	33.82	8.51	22.69	227.10	65.02	1.09×	269.43	1.13×
	75%	FF	82	1.38×	2.36	0.29	1.31	13.29	6.53	1.21×	18.51	1.15×	1.07	1.51×	25.86	3.97	17.10	171.16	46.93	1.51×	200.99	1.52×
	50%	FR	76	1.49×	**1.14**	**0.36**	1.21	12.28	2.71	1.76×	13.78	1.55×	1.07	1.51×	14.92	5.42	17.07	170.87	37.41	1.90×	191.21	1.59×
	50%	FS	72	1.57×	1.83	0.34	1.59	11.77	3.76	1.27×	13.94	1.53×	1.01	1.60×	26.07	5.55	16.21	162.24	47.83	1.48×	193.86	1.57×
	50%	FF	61	1.85×	2.00	0.19	0.96	9.93	3.15	1.52×	12.12	1.76×	0.80	2.03×	21.66	2.69	12.83	128.50	37.18	1.91×	152.85	2.00×
	25%	FS	45	2.51×	1.40	0.20	0.72	7.40	**2.32**	2.06×	**9.00**	2.37×	0.61	2.66×	18.34	2.79	9.72	97.34	30.85	2.30×	118.47	2.57×
	25%	FF	**44**	2.57×	1.73	0.18	0.71	7.25	3.42	1.40×	9.96	2.14×	**0.53**	3.06×	**17.25**	**1.37**	8.50	86.13	**27.12**	2.62×	**104.75**	2.91×
NB	100%	BL	833	1.00×	5.35	10.22	13.33	133.53	28.9	1.00×	149.1	1.00×	11.87	1.00×	77.84	83.94	189.87	1898.87	351.65	1.00×	2060.65	1.00×
	75%	FS	732	1.14×	4.87	9.11	11.71	117.33	25.69	1.12×	131.31	1.14×	10.26	1.16×	68.90	68.28	164.27	1642.87	301.45	1.17×	1780.05	1.16×
	75%	FF	454	1.83×	4.42	7	7.26	72.83	18.68	1.55×	84.25	1.77×	6.17	1.92×	51.77	23.20	98.67	986.87	173.64	2.03×	1061.84	1.94×
	50%	FR	433	1.93×	2.99	6.69	6.93	69.53	16.61	1.74×	79.21	1.88×	6.17	1.92×	36.63	28.00	98.67	986.86	163.30	2.15×	1051.49	1.96×
	50%	FS	515	1.62×	3.62	7.6	8.25	82.66	19.47	1.48×	93.88	1.59×	7.30	1.63×	49.66	46.85	116.80	1168.20	213.31	1.65×	1264.71	1.63×
	50%	FF	338	2.46×	3.49	5.41	5.41	54.33	14.31	2.02×	63.23	2.36×	4.63	2.56×	41.10	17.15	74.14	741.53	132.39	2.66×	799.78	2.58×
	25%	FS	315	2.64×	2.6	5.73	5.05	50.66	13.38	2.16×	58.99	2.53×	4.37	2.72×	32.40	26.99	69.86	698.87	129.25	2.72×	758.26	2.72×
	25%	FF	**242**	3.44×	**2.79**	**4.48**	3.86	38.79	**11.13**	2.60×	**46.06**	3.24×	**3.07**	3.87×	**29.54**	**11.28**	49.07	490.87	**89.89**	3.91×	**531.69**	3.88×

Interpretation: *FUSE variants consistently outperform the baseline (BL). FUSE-Fusion (FF) achieves the highest communication reduction and dominates in Norm Bounding (NB). In FedAvg (FA) and FedProx (FP), FUSE-RDX (FR) minimizes operation time, while FUSE-SPA (FS) optimizes end-to-end latency.* ***Legend:*** *k: retention ratio; Comm: per-client/round transfer; C/S: client (FUSE+crypto) / server (secure aggregation) compute; LAN/WAN: network transmission costs; T_{LAN} / T_{WAN}: end-to-end latency ($C + S$ + network). Blue: best experiment; green: speedup vs. BL.*

for NB up to 5.78×. Despite a one-time setup overhead of ≈ 82.5 MB, the significant per-round reductions render it an essential optimization for long-horizon LLM training (see Appendix F.2).

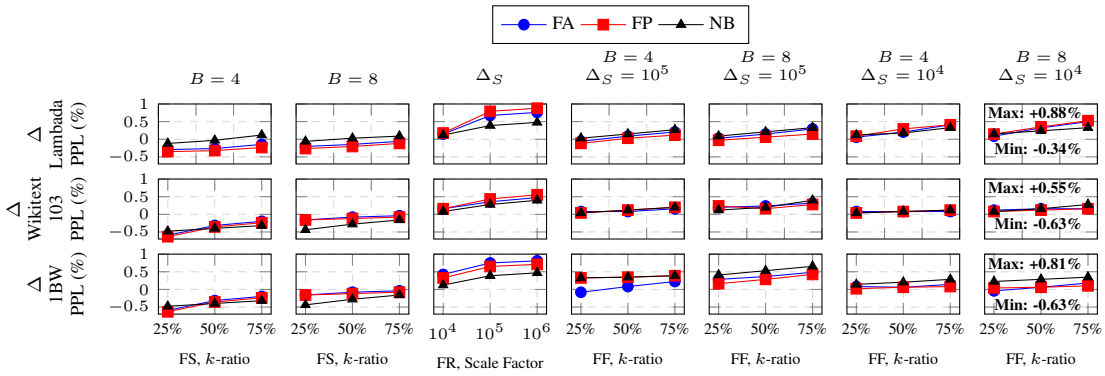
Scalability in Frontier Parameter Spaces. Table 2 demonstrates that FUSE scales with model dimensionality on a Llama-7B, in which a slightly higher efficiency gains is observed. This is because the auxiliary overheads are more effectively amortized across a larger parameter space (≈ 14× increase over GPT-2). In NB, FUSE-Fusion ($k = 25\%$) reduces per-client communication from 11.87 GB to 3.07 GB in a single-context setup, and further compresses up to 1.80 GB (6.59× reduction) via dual-context technique (Table 6). The scalability analysis in Appendix G demonstrates that FUSE-SPA sustains over 2× communication savings even at scale ($C = 100$).

4.2 When do communication savings become latency gains?

Algorithmic Efficiency and Computational Divergence. The GPT-2 analysis shows divergent latency performance driven by the compression overhead and ciphertext processing. In lightweight settings (FA & FP), FUSE-RDX delivers optimal time efficiency (client C + server S operations) at 50% sparsity, since the encrypted index-exchange overhead is avoided. Although FUSE-Fusion reduces communication further, its exchange overhead outweighs the gains when baseline computation is already minimal. The trade-off reverses under NB, where deep circuits increase per-ciphertext cost, making FUSE-Fusion ($k = 50\%$) preferable, lowering the total compute time ($C + S$) to 8.90s. This shows that aggressive ciphertext reduction is essential for scaling FHE-FL to complex, non-linear aggregation (see Appendix H for detailed analysis).

From Bandwidth-Bound to Execution-Balanced Regimes. The GPT-2 comparison between localized operation time ($C + S$) and E2E latency reveals a significant latency amplification effect under T_{WAN}. While FUSE-Fusion ($k = 25\%$) reduces cryptographic operation time by 2.14× under NB (up to 2.33× with dual CKKS-parameter), the total E2E latency reduction scales up to 3.24×. This amplification arises because WAN transmission dominates the NB FHE baseline, accounting for ≈ 89% (≈ 133.53s) of total round latency. FUSE-Fusion ($k = 25\%$) reduces this to ≈ 38.79s, achieving a more execution-balanced workflow. For lightweight CKKS (FA/FP), WAN stabilizes FUSE-SPA ($k = 25\%$) gains up to 2.37× speedup.

Network Resilience. The Llama-7B latency analysis in Table 2 highlights FUSE's impact on large-model deployment. The FHE baseline for NB aggregation incurs 2060.65s (≈ 34 minutes) per round WAN latency. FUSE-Fusion ($k = 25\%$) reduces the massive 11.87 GB communication



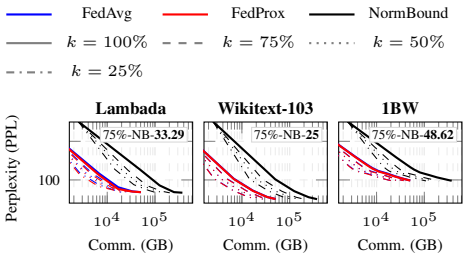
Notes: *Results show perplexity change (Δ PPL %) relative to FHE baseline for FUSE-SPA (FS), FUSE-RDX (FR), and FUSE-Fusion (FF);* ***(+ve)*** *values indicate improvement (lower PPL);* ***(-ve)*** *values indicate degradation; B: FS bin count; Δ_S: FR scale factor; k-ratio: top-k retention parameters; Rightmost annotations report best & worst PPL change.*

Figure 2: Ablation of FUSE variants over FHE baseline performance across Lambada, Wikitext-103, and 1BW datasets.

to 3.07 GB. This improvement is coupled with a speedup of up to 3.91× in LAN and 3.88× in WAN. In the lightweight scenario (FP/FA), FUSE-Fusion ($k = 25\%$) lower the E2E WAN time from $\approx$ 5 minutes to $\approx$ 1.75 minutes, achieving up to 2.93× speedup. This demonstrates that FUSE's compression benefits become increasingly significant as model complexity grows, shifting the viability of privacy-preserving LLM training.

4.3 How FUSE affects model utility across encrypted aggregators?

Referring to Figure 2 the FUSE suite achieves communication reductions while confining perplexity (PPL) fluctuations to a narrow $[-0.64\%, +0.88\%]$ band relative to the FHE baseline on GPT-2 family across different model scales (see Appendix J for full perplexity results). We observed that higher FUSE-SPA binning ($B = 8$) enhances rank-consensus resolution to accelerate convergence, yet it still incurs minor utility degradation from information pruning. Conversely, FUSE-RDX preserves utility by retaining full model updates but its scaling exhibits diminishing returns beyond $\Delta_S = 10^5$, suggesting the optimal numerical headroom for structural recovery. Notably, FUSE hyperparameter tuning preserves constant per-round overhead, as update granularity is adjusted within a fixed ciphertext volume. Figure 3 plots PPL against cumulative communication, demonstrating that FUSE-Fusion ($B = 8, \Delta_S = 10^5$) bridges FUSE approaches by maintaining stable performance even under aggressive compression. Empirically, NB-75% achieves the lowest final perplexity overall. Moreover, our FHE-FL pipeline preserves perplexity relative to the official GPT-2 baselines (Radford et al., 2019), with no degradation observed. In contrast, FedPHE (Yan et al., 2024) incurs ≈1.58% accuracy loss on ResNet-20 FedAvg (Yan et al., 2024), where FUSE's perplexity degradation on encrypted federated-LLM remains below 1%. Detailed comparisons are in Appendix I.



Notes: *Binning $B = 8$; FUSE scaling $\Delta_S = 10^5$; k : parameter ratio; Annotated values: the optimal PPL achieved.*

Figure 3: Perplexity vs. cumulative communication for FUSE-Fusion.

5 Conclusion

In this study, we present FUSE, which reconciles the tension between update sparsity and ciphertext packing, together with a sign-gating circuit that extends encrypted aggregation beyond FedAvg to Norm Bounding. A central insight is that the optimal compression technique depends on the CKKS computational regime: FUSE-RDX optimizes operational and time efficiency, whereas FUSE-Fusion preserves utility under deep parameter constraints. A notable limitation in FUSE is the high overhead due to FHE sign approximation. In the future, we plan to co-develop more FHE-friendly approximations, extend the threat model to malicious clients, and explore hardware-accelerated FHE kernels. This positions FUSE as a scalable and resilient foundation for secure federated systems.

AI USE STATEMENT

We only used generative AI tools for writing assistance (grammar checking and formatting) and literature search. All AI-assisted content was manually reviewed and verified.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, We provided the source code as part of the supplementary materials (https://anonymous.4open.science/r/FUSE-Versatile-Advanced-Aggregation-C325), which includes all necessary scripts to reproduce the experiments. Detailed experimental setups are provided in Appendix E.

REFERENCES

Ahmad Al Badawi, Jack Bates, Flavio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, R. V. Saraswathy, Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. OpenFHE: Open-source fully homomorphic encryption library. In *Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pp. 53–63. ACM WAHC, 2022. doi: 10.1145/3560827.3563379.

Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, et al. Homomorphic Encryption Security Standard. Technical report, Homomorphic Encryption Standardization, Toronto, Canada, 2018.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 14457–14483. Curran Associates, Inc., 2024. doi: 10.52202/079017-0461.

Joppe W. Bos, Joost Renes, and Christine van Vredendaal. Post-Quantum cryptography with contemporary Co-Processors: Beyond kronecker, schönhage-strassen & nussbaumer. In *Proceedings of the 31st USENIX Security Symposium*, pp. 3683–3697, USENIX Security, 2022.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Leveled fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 309–325. ACM ITCS, 2012. doi: 10.1145/2090236.2090262.

Dong Chen, Hongyuan Qu, and Guangwu Xu. Aegisfl: efficient and flexible privacy-preserving byzantine-robust cross-silo federated learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology – ASIACRYPT 2017*, pp. 409–437. Springer, 2017. doi: 10.1007/978-3-319-70694-8_15.

Jung Hee Cheon, Dongwoo Kim, and Duhyeong Kim. Efficient homomorphic comparison methods with optimal complexity. In *Advances in Cryptology – ASIACRYPT 2020*, pp. 221–256. Springer, 2020. doi: 10.1007/978-3-030-64834-3_8.

Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1):34–91, 2020. doi: 10.1007/s00145-019-09319-x.

Pedro Correia, Ivan Costa, Ivone Amorim, Eva Maia, and Isabel Praça. Federated learning: An approach with hybrid homomorphic encryption. In Romain Laborde, Joaquin Garcia-Alfaro, Abbas Yazdinejad, Gregory Epiphaniou, Habtamu Abie, Silvio Ranise, Michał Choraś, Michał Woźniak, Yuko Hara, Jan Tobias Mühlberg, Claudia Greco, and Kim-Kwang Raymond Choo (eds.), *Computer Security. ESORICS 2025 International Workshops*, pp. 254–273, Cham, 2026.

Minxin Du, Xiang Yue, Sherman S. M. Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, pp. 2665–2679, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700507. doi: 10.1145/3576915.3616592.

Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *Cryptology ePrint Archive, Paper 2012/144*, 2012.

David Harvey. Faster polynomial multiplication via multipoint kronecker substitution. *Journal of Symbolic Computation*, 44(10):1502–1510, 2009. doi: 10.1016/j.jsc.2009.05.004.

Chenghao Hu and Baochun Li. Maskcrypt: Federated learning with selective homomorphic encryption. *IEEE TDSC*, 22(1):221–233, January 2025. ISSN 1545-5971. doi: 10.1109/TDSC.2024.3392424.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, ICLR, 2022.

Siyang Jiang, Hao Yang, Qipeng Xie, Chuan Ma, Sen Wang, Zhe Liu, Tao Xiang, and Guoliang Xing. Towards compute-efficient Byzantine-robust federated learning with fully homomorphic encryption. *Nature Machine Intelligence*, 7:1657–1668, 2025. doi: 10.1038/s42256-025-01107-6.

Weizhao Jin, Yuhang Yao, Shanshan Han, Carlee Joe-Wong, Srivatsan Ravi, Salman Avestimehr, and Chaoyang He. FedML-HE: An efficient homomorphic-encryption-based privacy-preserving federated learning system. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS*, 2023.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5260–5271. ACM KDD, 2024. doi: 10.1145/3637528.3671573.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of the 3rd MLSys Conference*, pp. 429–450. MLSys, 2020.

Yanli Li, Zhongliang Guo, Nan Yang, Huaming Chen, Dong Yuan, and Weiping Ding. Threats and defenses in the federated learning life cycle: A comprehensive survey and challenges. *IEEE Transactions on Neural Networks and Learning Systems*, 36(9):15643–15663, 2025. doi: 10.1109/TNNLS.2025.3563537.

Jianmin Liu, Li Yan, Borui Li, Lei Yu, and Chao Shen. SHE-loRA: Selective homomorphic encryption for federated tuning with heterogeneous loRA. In *The 14th International Conference on Learning Representations*, 2026.

Yiping Ma, Yue Guo, Harish Karthikeyan, and Antigoni Polychroniadou. Armadillo: Robust single-server secure aggregation for federated learning with input validation. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, CCS '25, pp. 2219–2233, 2025. ISBN 9798400715259. doi: 10.1145/3719027.3765216.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

Chuiyang Meng, Ming Tang, and Vincent Wong. Florg: Federated fine-tuning with low-rank gram matrices and procrustes alignment. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, 2026.

Md Jueal Mia and M. Hadi Amini. Fedshield-llm: A secure and scalable federated fine-tuned large language model. *arXiv preprint arXiv:2506.05640*, 2025.

Yao Pan, Zheng Chao, Wang He, Yang Jing, Li Hongjia, and Wang Liming. Fedshe: privacy preserving and efficient federated learning with adaptive segmented ckks homomorphic encryption. *Cybersecurity*, 7(1):40, 2024.

Ivo Petrov, Dimitar Iliev Dimitrov, Maximilian Baader, Mark Niklas Mueller, and Martin Vechev. DAGER: Exact gradient inversion for large language models. In *The 38th Conference on Neural Information Processing Systems*, 2024.

Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41473–41497. PMLR, 21–27 Jul 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sinem Sav, Apostolos Pyrgelis, Juan Ramón Troncoso-Pastoriza, David Froelicher, Jean-Philippe Bossuat, Joao Sa Sousa, and Jean-Pierre Hubaux. POSEIDON: Privacy-preserving federated learning with fully homomorphic encryption. In *Proceedings of the 28th Annual Network and Distributed System Security Symposium*. NDSS, 2021.

Sinem Sav, Abdulrahman Diaa, Apostolos Pyrgelis, Jean-Philippe Bossuat, and Jean-Pierre Hubaux. Privacy-preserving federated recurrent neural networks. In *Proceedings on Privacy Enhancing Technologies*, volume 2023, pp. 500–521, 2023. doi: 10.56553/popets-2023-0122.

Yao Shu, Wenyang Hu, See-Kiong Ng, Bryan Kian Hsiang Low, and Fei Yu. Ferret: Federated full-parameter tuning at scale for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55379–55402. PMLR, 13–19 Jul 2025.

Jinhyun So, Chaoyang He, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E. Ali, Basak Guler, and Salman Avestimehr. Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning. In D. Marculescu, Y. Chi, and C. Wu (eds.), *Proceedings of Machine Learning and Systems*, volume 4, pp. 694–720, 2022.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, 2024.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

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

Linh Tran, Wei Sun, Stacy Patterson, and Ana Milanova. Privacy-preserving personalized federated prompt learning for multimodal large language models. In *The 13th International Conference on Learning Representations*, 2025.

Na Wang, Wei Yan, Sian-Jheng Lin, and Yuliang Huang. An entropy coding based on binary encoding for mixed-radix digits. In *2022 Data Compression Conference*, pp. 488–488. DCC, 2022. doi: 10.1109/DCC52660.2022.00099.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 22513–22533. Curran Associates, Inc., 2024. doi: 10.52202/079017-0708.

Xiyi Wei, Ruwei Huang, Lei Lei, and Jiayi Wu. Fedshe–cq: A communication-efficient homomorphic encryption framework for federated learning. *Computer Networks*, 274:111813, 2026. ISSN 1389-1286. doi: 10.1016/j.comnet.2025.111813.

Ming Wen, Jiaqi Zhu, Yuedong Xu, Yipeng Zhou, and DINGDING HAN. Differentially private federated low rank adaptation beyond fixed-matrix. In *The 39th Annual Conference on Neural Information Processing Systems*, 2025.

Shuangqing Xu, Yifeng Zheng, and Zhongyun Hua. Harnessing sparsification in federated learning: A secure, efficient, and differentially private realization. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, CCS '25, 2025. doi: 10.1145/3719027.3765044.

Jiaqi Xue, Mayank Kumar, Yuzhang Shang, Shangqian Gao, Rui Ning, Mengxin Zheng, Xiaoqian Jiang, and Qian Lou. DictPFL: Efficient and private federated learning on encrypted gradients. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Nan Yan, Yuqing Li, Jing Chen, Xiong Wang, Jianan Hong, Kun He, and Wei Wang. Efficient and straggler-resistant homomorphic encryption for heterogeneous federated learning. In *Proceedings of the IEEE Conference on Computer Communications*, pp. 791–800. IEEE INFOCOM, 2024.

Yunlu Yan, Chun-Mei Feng, Wangmeng Zuo, Lei Zhu, Rick Siow Mong, and Yong Liu. Federated residual low-rank adaption of large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, 2025.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. OpenFedLLM: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6137–6147. ACM KDD, 2024.

Farjana Yesmin. Medhe: Communication-efficient privacy-preserving federated learning with adaptive gradient sparsification for healthcare. *arXiv preprint arXiv:2511.09043*, 2025.

Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, pp. 493–506. USENIX Association, July 2020. ISBN 978-1-939133-14-4.

Yulin Zhao, Hualin Zhou, and Zhiguo Wan. SuperFL: Privacy-preserving federated learning with efficiency and robustness. *Cryptology ePrint Archive, 2024/081*, 2024.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, 2019.

A Backgrounds

A.1 Federated Learning (FL)

Federated learning (McMahan et al., 2017) is a distributed training paradigm in which a central server coordinates training across clients without accessing their local datasets. In each FL round, a subset of clients is selected for local training on their respective datasets and send the resulting model updates to the server. The server then aggregates these updates with a designated FL algorithm to generate a new global model, which is then broadcast to all clients. Finally, the client synchronizes its locally trained model for the next round FL. This iterative process repeats until convergence or a preset number of rounds.

The aggregation algorithm is the defining component of an FL algorithm. FedAvg (McMahan et al., 2017) is one popular FL algorithm due to its simplicity. The global model update (u_{glob}) is computed as a weighted average over selected clients (u_c) as follow:

$$u_{glob} = \sum_{c \in C} \frac{n_c}{N} \cdot u_c, \tag{5}$$

where $c \in C$ is the subset of clients and $\frac{n_c}{N}$ is their weight proportion. It has been widely used in FL research (Xue et al., 2025; Zhang et al., 2020; Jin et al., 2023; Yan et al., 2024) due to its linear operations. However, this simplicity limits its performance in complex real-world scenarios, such as statistical heterogeneity (non-IID), motivating the development of more advanced federated algorithms.

FedProx (Li et al., 2020) stabilizes non-IID convergence by introducing a proximal term to client's objective to penalize deviation from the previous global model (u^{t-1}):

$$\frac{\mu}{2} \left\| u_c - u^{t-1} \right\|_2^2, \tag{6}$$

where μ controls the regularization strength. While Norm Bounding Sun et al. (2019) provides an alternative to reduce client drift by constraining clients' influence through input clipping as follow:

$$\tilde{u}_c = u_c \cdot \left(\frac{\tau}{\|u_c\|_2} \right), \text{when } \|u_c\|_2 > \tau \tag{7}$$

For clients whose ℓ_2-norm ($\|u_c\|_2$) exceeds a threshold τ, their input are scaled, otherwise remaining unchanged.

A.2 Fully Homomorphic Encryption

Modern FHE schemes are generally categorized into three generations based on their underlying mathematical structures: *(i) Integer Schemes* such as BGV Brakerski et al. (2012) and BFV Fan & Vercauteren (2012), which facilitate modular arithmetic over finite fields; *(ii) Torus-based Schemes* such as TFHE Chillotti et al. (2020), which are optimized for fast boolean gate evaluations and lookup tables; and *(iii) Fixed-point Arithmetic Schemes* such as CKKS Cheon et al. (2017), which support approximate computations over real or complex numbers. The CKKS scheme is essential for federated LLM, where model updates are often represented as floating-point values. A CKKS ciphertext encrypts a plaintext polynomial that encodes a message vector via SIMD packing, allowing parallel slot-wise operations. The scheme natively supports arithmetic primitives (addition, subtraction, multiplication), while non-linear functions (e.g., comparison, max, min, etc) must be approximated using polynomial circuits. These approximations often introduce both numerical errors and greater computational latency. This computational disparity thus defines a key challenge for implementing FHE-compatible robust aggregation.

CKKS Encoding and Packing. The CKKS scheme enables efficient parallel computation via batching, a technique that encodes a vector of real or complex values into a single plaintext polynomial. Computation is performed over the cyclotomic ring $\mathcal{R} = \mathbb{Z}[X]/(X^N+1)$, where N is a power-of-two. A message vector $z \in \mathbb{R}^{N/2}$ is first scaled by a factor Δ_{CKKS}, then mapped via the inverse canonical embedding $\sigma^{-1} : \mathbb{C}^N \to \mathcal{R}$ to a polynomial, rounded to the nearest element of $\mathcal{R}$. This encoding enables SIMD execution, allowing up to $N/2$ independent values to be packed into independent ciphertext "slots" for parallel, slot-wise computation. Rearranging data across slots is achieved through Galois automorphisms (rotations), which induce cyclic rotations of the slot contents without decryption.

CKKS Parameters and Noise Management. The operational feasibility of CKKS is governed by multiplicative depth (L), numerical precision (scaling factor Δ_{CKKS}), and ring degree. Each ciphertext carries a noise budget that accumulates with each homomorphic operation. To prevent decryption failure, it must accommodate noise growth by ensuring the top modulus Q of the chain satisfies $\log Q \approx (L+1) \log \Delta_{\text{CKKS}}$, with N scaled accordingly to maintain security. Since each homomorphic multiplication consumes one level from this chain, non-linear operations that require approximation would require deeper circuits and a larger modulus chain, which makes computation costly.

B FUSE-SPA (Extended)

B.1 Fidelity Bound Analysis

In this section, we provide the formal proof for the Top-k Selection Fidelity Bound presented in the main text.

Table 3: Summary of notation.

Symbol	Description
Federated Learning	
C	Subset of active clients in an FL round.
c	Index of a single client, $c \in C$.
u_c	Local model update from client c.
u_{glob}	Global model update.
n_c	Data weight (e.g., local sample count) for client c.
Secure Aggregation	
pk, sk, evk	Public key, secret key, and evaluation key.
$\mathsf{sgn}_{\text{poly}}(\cdot)$	Polynomial approximation of the sign function.
τ	Clipping threshold for secure Norm Bounding.
$\mathsf{FHEAgg}(\cdot)$	Secure homomorphic aggregation protocol (Alg. 3).
FUSE-SPA	
r	Sparsification ratio (Top-k percentage).
B	Number of bins for importance scores.
$ISlot$	Digit-width per importance score, used to prevent overflow (Eq. 2).
ρ	Packing factor: number of scores per ciphertext slot.
$\mathcal{I}_{\text{glob}}$	Set of globally significant indices corresponding to the top $r\%$ aggregate importance scores.
$\mathsf{Unpack}(\cdot)$	Mapping of aggregated sparse updates to their original indices for model synchronization.
FUSE-RDX	
Δ_{FUSE}	FUSE scaling factor for integer projection.
B_{rdx}	Radix encoding base (Eq. 3).
W_{total}	Worst-case aggregation multiplier.
$\tilde{u}_{c,a}, \tilde{u}_{c,b}$	Constituent integer pairs derived from the scaled u_c for mixed-radix packing.

Theorem 1 (Restated). For each client $c \in \mathcal{C}$, let $|u_{c,j}|$ denote the true update magnitude of index $j \in \{1, \ldots, d\}$ and $s_{c,j} \in \{0, \ldots, B\}$ its binned importance score, where $s_{c,j} = 0$ denotes exclusion from the local Top-k set. Let $w_c = (k_c^{\max} - k_c^{\min})/B$ be the local bin width. For any selected index i and rejected index i' satisfying $\sum_{c\in\mathcal{C}} s_{c,i} \geq \sum_{c\in\mathcal{C}} s_{c,i'}$, the true aggregate magnitudes $T_i = \sum_{c\in\mathcal{C}} |u_{c,i}|$ and $T_{i'} = \sum_{c\in\mathcal{C}} |u_{c,i'}|$ satisfy:

$$T_{i'} - T_i \;\leq\; \sum_{c\in\mathcal{C}} \max\big(w_c,\, k_c^{\min}\big) \;+\; \underbrace{\sum_{c\in\mathcal{C}} w_c\,(s_{c,i'} - s_{c,i})}_{\text{Heterogeneity Gap}}$$

Proof. Fix a client $c \in \mathcal{C}$. Since $\{Q_b\}_{b=1}^{B}$ partitions $[k_c^{\min}, k_c^{\max}]$ into intervals of width w_c, the binning scheme bounds magnitude values as follows:

- **Top-k set** ($s_{c,j} \geq 1$): $k_c^{\min} + (s_{c,j} - 1)\, w_c \;\leq\; |u_{c,j}| \;< k_c^{\min} + s_{c,j}\, w_c$;
- **Rejected set** ($s_{c,j} = 0$): $0 \leq |u_{c,j}| < k_c^{\min}$.

We evaluate the per-client magnitude difference $\delta_c := |u_{c,i'}| - |u_{c,i}|$ by considering the score of the selected index i:

Case 1 ($s_{c,i} \geq 1$)**.** Applying the upper bound to i' and the lower bound to i,

$$\delta_c \;\leq\; \big(k_c^{\min} + s_{c,i'} w_c\big) - \big(k_c^{\min} + (s_{c,i} - 1) w_c\big) \;=\; w_c\,(s_{c,i'} - s_{c,i}) + w_c.$$

Case 2 ($s_{c,i} = 0$)**.** Here $|u_{c,i}| \geq 0$, so

$$\delta_c \;\leq\; k_c^{\min} + s_{c,i'} w_c \;=\; w_c\,(s_{c,i'} - s_{c,i}) + k_c^{\min},$$

where the equality uses $s_{c,i} = 0$.

Since $w_c \le \max(w_c, k_c^{\min})$ and $k_c^{\min} \le \max(w_c, k_c^{\min})$, both cases give

$$\delta_c \;\le\; w_c\,(s_{c,i'} - s_{c,i}) + \max\big(w_c,\, k_c^{\min}\big).$$

Summing over all $c \in \mathcal{C}$ and noting $\sum_c \delta_c = T_{i'} - T_i$,

$$T_{i'} - T_i \;\le\; \sum_{c\in\mathcal{C}} \max\big(w_c,\, k_c^{\min}\big) + \sum_{c\in\mathcal{C}} w_c\,(s_{c,i'} - s_{c,i}),$$

This completes the proof. □

Remark 1 (Saturation of the discretisation term). The first term is governed by each client's local bin width only while that width exceeds the smallest selected magnitude. Increasing B therefore tightens this term in inverse proportion to B. The point at which this tightening stops is determined by the ratio between the largest and smallest magnitudes in each client's local Top-k set. Beyond that point, further binning brings no additional benefit, and the term settles at the sum of the clients' smallest selected magnitudes.

When bin widths differ, the Heterogeneity can be positive even when the unweighted score sums favour the selected index. A client with large w_c can dominate the weighted term. This term captures the ranking discrepancy from aggregating unweighted scores across heterogeneous clients. Increasing B narrows every bin width and so continues to shrink this term, even after the first term has saturated.

B.2 Privacy of the Global Importance Map.

Under our semi-honest setting, the privacy of the global importance map s_{glob} is rooted in its role as a quantized metadata aggregate, which decouples index-level saliency from raw gradient magnitudes. For any index $j \in \{1, \ldots, d\}$, the server computes on aggregate $s_{glob,j} = \sum_{c\in\mathcal{C}} s_{c,j} \in [0,\, C \cdot B]$, where each local contribution $s_{c,j}$ is obtained via a surjective, many-to-one binning function of width w_c. This induces binning ambiguity, whereby a single discrete score corresponds to a range of underlying magnitudes, and multiple distinct local updates map to the same value. In addition, aggregation-induced obfuscation arises from summing C bounded contributions, such that any observed $s_{\text{glob},j}$ admits a vast set of valid decompositions. This combinatorial uncertainty scales with the client population C and bin count B, rendering the decomposition of the aggregate into individual client contributions underdetermined. This ambiguity is weakest near the boundary values ($s_{glob,j} \approx 0$ or $s_{glob,j} \approx C \cdot B$), which reveal near-unanimous consensus across clients rather than any individual magnitude. Thus, only coarse, index-level saliency is exchanged rather than precise per-client updates. This reduction in information granularity prioritizes communication efficiency over fine-grained observability. For deployments requiring stronger protection against auxiliary inference, techniques like differential privacy (e.g., injecting calibrated discrete noise before packing) can be integrated as an orthogonal extension, because the sensitivity per score is limited by the bound $[0, B]$.

C Decryption Correctness and Noise Analysis

Let $\|a\|_\infty^{\text{can}} = \|\sigma(a)\|_\infty$ denote the infinity norm of a under the canonical embedding σ, and let c_M be the ring expansion factor satisfying $\|a\|_\infty \le c_M \|a\|_\infty^{\text{can}}$. To guarantee correct decryption within the CKKS framework for the compact index packing in FUSE-SPA and the integer components of FUSE-RDX, the encoded plaintext polynomial m must satisfy $\|m\|_\infty < q/2$ to prevent modular wrap-around modulo q. Let $T = \|m\|_\infty^{\text{can}}$ denote the canonical-embedding norm of the encoded, scaled and aggregated plaintext. The expansion bound then gives $\|m\|_\infty \le c_M \cdot T$, so a sufficient condition for correct decryption is

$$c_M \cdot T < \frac{q}{2},$$

This condition ensures that the aggregated messages in FUSE-SPA (governed by total score magnitude) and FUSE-RDX (governed by the packed integer magnitude) do not overflow the ciphertext modulus. Our parameter selection satisfies this inequality for the defined ranges of B and B_{rdx}. For remaining notation details, please refer to Section 2.2 of the CKKS paper (Cheon et al., 2017).

Therefore, in the FUSE-RDX encoding, the largest integer that can be safely encoded is approximately T/Δ_{CKKS}, with $T < q/(2c_M)$, where Δ_{CKKS} is the scaling factor. To ensure the accumulated encryption noise does not corrupt the lowest-order digit in FUSE-RDX, Δ_{CKKS} must dominate the aggregated noise. Following Lemma 1 of (Cheon et al., 2017), for summation across C clients, the scaling factor must satisfy:

$$\Delta_{\mathrm{CKKS}} > 2\,C \cdot B_{\mathrm{enc}}.$$

If the aggregation involves constant multiplication with a maximum magnitude (W_{total}), the condition is:

$$\Delta_{\mathrm{CKKS}} > 2\,C \cdot W_{total} \cdot B_{\mathrm{enc}}.$$

This ensures that $\lfloor (\Delta_{\mathrm{CKKS}} \cdot z + e)/\Delta_{\mathrm{CKKS}} \rceil = z$ yields the correct integer without interference from the accumulated noise.

D Error Analysis for Encrypted Norm Bounding

Approximation Error Bound. This analyzes the approximation error introduced by the encrypted norm-bounding mechanism. Let $f_{\mathrm{comp}}(g)$ denote the composite polynomial used to approximate the sign function $\mathrm{sgn}(g)$ over the normalized domain $g \in [-1, 1]$. Define the worst-case error outside the transition region $[-\varepsilon, \varepsilon]$ as

$$\delta \coloneqq \sup_{x \in [-1,1] \setminus [-\varepsilon,\varepsilon]} |\mathrm{sgn}(x) - f_{\mathrm{comp}}(g)|$$

To derive the boolean indicator $\sigma_c \in [0, 1]$, we apply the linear mapping $\sigma_c = 0.5(f_{\mathrm{comp}}(g) + 1)$, resulting in an indicator error bounded by $\delta/2$ relative to the ideal indicator $I_c \in \{0, 1\}$. The encrypted clipping factor is:

$$S_c^{\text{FHE}} = \sigma_c \cdot \left(\frac{\tau}{\|\mathbf{u}_c\|_2} - 1 \right) + 1$$

The worst-case error per client is:

$$|S_c^{\text{FHE}} - S_c^{\text{ideal}}| = |(\sigma_c - I_c)| \cdot \left| \frac{\tau}{\|\mathbf{u}_c\|_2} - 1 \right| \leq \frac{\delta}{2} \cdot \left| \frac{\tau}{\|\mathbf{u}_c\|_2} - 1 \right|.$$

The clipping threshold τ is set to bound only unusually large updates. It is reasonable to assume all client updates satisfy$\|\mathbf{u}_c\|_2 \geq \tau/\beta$ for some $\beta > 1$. Under this assumption, $\left| \frac{\tau}{\|\mathbf{u}_c\|_2} - 1 \right| \leq \beta - 1$. Thus, the final bound is:

$$|S_c^{\text{FHE}} - S_c^{\text{ideal}}| \leq \frac{\delta}{2} \cdot (\beta - 1)$$

For $\delta \approx 10^{-4}$ and $\beta = 2$, the maximum deviation is 5×10^{-5}. This theoretical upper bound on per-client scaling error is negligible, which aligns with the observed stability from our empirical result of $< 1\%$ perplexity degradation in Section 4.3.

E Setup Configurations

Network and configuration. FUSE is evaluated under two network profiles. The *LAN* environment (1 Gbps, 1 ms RTT) reflects data-center clusters, while the *WAN* environment (100 Mbps, 100 ms RTT) simulates cross-region cloud coordination. All experiments are conducted on an Intel Xeon W5-2465X workstation with 512 GB RAM and an RTX 5090 GPU. All CKKS configurations are instantiated at 128-bit classical security according to the *Homomorphic Encryption Security Standard* Albrecht et al. (2018), with $N = 2^{16}$.

Table 4: Experimental configuration: parameters for CKKS, federated learning, and FUSE variants.

Parameter Category	**Value / Configuration**
CKKS FHE Parameters	
Polynomial degree (N)	65,536
Baseline & FUSE-SPA (scale / depth)	**FA & FP**: 50 / 1; **NB**: 50 / 20
FUSE-RDX & FUSE-Fusion (scale / depth)	**FA & FP**: 55 / 2; **NB**: 55 / 21
Federated Learning Parameters	
FL rounds	150
Local epochs per round	1
Client participation	3 selected from 10 per round
Batch size	16
Initial learning rate	5×10^{-5}
LoRA rank (r)	32
Dirichlet concentration (α)	0.5
Algorithm-Specific Hyperparameters	
FedProx proximal term (μ)	0.01
Norm Bounding threshold (τ)	5
FUSE-Specific Parameters	
Top-k sparsification ratios	75%, 50%, 25%
Bin count B (FUSE-SPA / FUSE-Fusion)	$\{4, 8\}$
Scaling factor Δ_{FUSE} (FUSE-RDX / FUSE-Fusion)	$\{10^4, 10^5, 10^6\}$

Backbone model and evaluation. For the architectural backbone, we employ LoRA ($R = 32$) for all experiments, including the FHE baseline. This configuration results in approximately 16.1 million trainable parameters for Llama-7B (Touvron et al., 2023) , 9.83 million and 1.18 million for GPT-2 (Radford et al., 2019). Consistent with established language modeling benchmarks, we evaluate and report zero-shot perplexity following the protocol established for GPT-2.

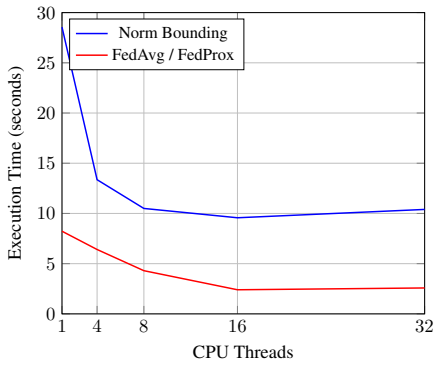


Figure 4: Effect of CPU parallelism on total execution time for per FL round.

CPU-thread parallelism on homomorphic operations This experiment exploits hybrid parallelism: CPU parallelism for encryption operations in OpenFHE, and GPU parallelism for local model training. As shown in Figure 4, the total execution time for encrypted full-model update operations (covering encryption, decryption, and aggregation), the CPU-parallelism performance gain saturates beyond 16 threads, where the runtime reduction becomes marginal (14.56s for NormBound, 2.4s for FedProx, and FedAvg), thus the experiments use 16 CPU threads for all homomorphic operations.

F COMMUNICATION ANALYSIS (EXTENDED)

F.1 DEPTH-DEPENDENT COMMUNICATION DYNAMICS AND PARAMETER SCALING.

Table 5: Evaluation of per-ciphertext size and context overhead (MB) across various CKKS configurations ($N = 2^{16}$).

Set	Scale	Depth	Per-Ciphertext Size (MB)	Context Size (MB)
Parameter for FedAvg & FedProx				
1	50	1	3.14	16.00
2	55	2	4.19	16.60
Parameter for Norm Bounding				
3	50	20	23.14	62.79
4	55	21	24.07	66.50

The comparative efficiency of FUSE-SPA and FUSE-RDX is governed by a trade-off between ciphertext density and per-ciphertext volume, modulated by the multiplicative depth of homomorphic circuit. As observed in our experiments, at a nominal sparsity of 50%, FUSE-SPA outperforms FUSE-RDX under FedAvg and FedProx. However, FUSE-RDX demonstrates superior performance under NB compared to FUSE-SPA. This shift is driven by the CKKS parameter requirements:

- **Lightweight Circuits (FedAvg/FedProx)**: FUSE-RDX requires slightly larger parameters (scale 55, depth 2) to support large integer packing. This results in a larger per-ciphertext size (≈ 4.19 MB) compared to FUSE-SPA (≈ 3.14 MB), as detailed in Table 5. Consequently, at the same sparsity ratio, FUSE-SPA yields greater communication reduction because the marginal gain from FUSE-RDX does not yet outweigh the increased cost of its heavier ciphertext base.
- **Deep Circuits (Norm Bounding):** Implementing secure norm-bounding requires a higher CKKS parameter (depth $20/21$) to accommodate the polynomial compositions used for sign-gating. While this depth requirement inflates per-ciphertext size to ≈ 24 MB.

The deep circuit for Norm Bounding creates a crossover point in the compression technique. FUSE-SPA requires transmitting additional ciphertexts to exchange index masks for consensus Top-k selection. The substantial size of these auxiliary ciphertexts incurs significant communication overhead, leading to diminishing returns. In contrast, FUSE-RDX prioritizes minimizing the total ciphertext count by maximizing spatial utilization within the CKKS slots. Thus, achieving greater communication efficiency without transmitting expensive auxiliary masks. In summary, FUSE-RDX encoding performs better than FUSE-SPA for deep circuits because it treats the expensive ciphertexts as the primary resource to be conserved, whereas FUSE-SPA's index-exchange mechanism is better suited for lightweight circuits where individual ciphertexts are relatively inexpensive.

F.2 DUAL-CONTEXT OPTIMIZATION FOR NORM BOUNDING.

We introduce a dual-context mixed-parameter technique to further enhance Norm Bounding compression efficiency. This optimization is predicated on the observation that the index-exchange protocol in sparsification-based FUSE involves only linear homomorphic additions, whereas Norm Bounding's clipping requires polynomial evaluations that demand a much larger CKKS parameter setting. In a single-context setup, all ciphertexts will be instantiated with the heavy parameters needed for sign approximation, resulting in significant bandwidth waste. Therefore, this technique decouples these operations into two distinct homomorphic contexts: update context uses full-depth parameters (Table 5 Set 3/4) to support the non-linear clipping operations of Norm Bounding; index context uses minimal-depth parameters (Table 5 Set 1) solely for the linear aggregation of index masks. By segregating these homomorphic operations, this technique elevates the total end-to-end communication reduction for NB up to $5.78\times$ for GPT-2 and $6.59\times$ for Llama-7B, deriving an additional $\approx 1.7\times$ reduction compared to single context CKKS (refer to Table 6)

Table 6: Detailed per-client per-round communication cost of FUSE across FedAvg (FA), FedProx (FP), and Norm Bounding (NB) on GPT-2 and Llama-7B.

k	Exp.	GPT-2 (MB)									Llama-7B (GB)								
		FA & FP			NB						FA & FP			NB					
		IE	SA	T_S	IE	SA	T_S	**Ratio**	T_D	**Ratio**	IE	SA	T_S	IE	SA	T_S	**Ratio**	T_D	**Ratio**
100%	BL	–	113	113	–	833	833	1.00×	–	–	–	1.62	1.62	–	11.87	11.87	1.00×	–	–
75%	FS	16	85	101	115	617	732	1.14×	633	1.32×	0.20	1.21	1.41	1.40	8.86	10.26	1.16×	9.06	1.31×
75%	FF	23	59	82	121	333	454	1.83×	356	2.34×	0.27	0.80	1.07	1.53	4.63	6.16	1.93×	4.90	2.42×
50%	FR	–	76	76	–	433	433	1.92×	–	–	–	1.07	1.07	–	6.17	6.17	1.92×	–	–
50%	FS	16	57	73	115	400	515	1.62×	416	2.00×	0.20	0.81	1.01	1.40	5.90	7.30	1.63×	6.10	1.95×
50%	FF	23	38	61	121	218	339	2.46×	241	3.46×	0.27	0.53	0.80	1.53	3.10	4.63	2.56×	3.37	3.52×
25%	FS	16	29	45	115	200	315	2.64×	216	3.86×	0.20	0.41	0.61	1.40	2.97	4.37	2.72×	3.17	3.74×
25%	FF	23	21	**44**	121	121	**242**	3.44×	**144**	**5.78**×	0.27	0.27	**0.54**	1.53	1.53	**3.07**	3.87×	**1.80**	**6.59**×

Notes: *k: retention ratio; BL: FHE baseline. Exp. denotes the experimental variant: BL (FHE baseline), FR (FUSE-RDX), FS (FUSE-SPA), and FF (FUSE-Fusion). IE/SA: Index Exchange and Server Aggregation communication; T_S and T_D: single and dual CKKS (leverage lightweight parameter for IE) parameter settings. Bold blue: best. Green: reduction vs. BL.*

Table 7: Scalability and communication analysis for FHE Baseline and FUSE-SPA.

	Baseline		FUSE (Ours)	
Metrics	$C = 10$	$C = 100$	$C = 10$	$C = 100$
Scalability	$\mathcal{O}(C \cdot \lvert\theta\rvert)$		$\mathcal{O}\left(C \cdot \frac{\lvert\theta\rvert}{\log(C)}\right)$	
Packing factor (ρ)	–	–	8	5
Sparsity (k)	–	–	25%	25%
Communication (GB)				
Index Exchange	–	–	0.16	2.56
Update Aggregation	1.13	11.30	0.29	2.90
Total (GB)	1.13	11.30	0.45	5.46
Reduction	–	–	**2.51**×	**2.01**×

G Scalability Analysis Under High Parallelism

In FUSE-SPA, the digit-width (I_{Slot}) scales logarithmically with the number of clients C. Hence, the scalability complexity is $\mathcal{O}(C \cdot \frac{|\theta|}{\log C})$, while the baseline grows as $\mathcal{O}(C \cdot |\theta|$, where $|\theta|$ is the number of trainable parameters. Table 7 summarizes the communication scaling of FUSE-SPA (25%) relative to the FHE baseline under highly parallelized settings. When C increases from 10 to 100, I_{Slot} is expanded from 2 to 3, thereby reducing packing factor ρ from 8 to 5. This means that with 100 clients, each ciphertext slot can only pack 5 importance scores. Despite this, FUSE-SPA consistently achieves favourable communication reduction. For GPT-2, total communication is reduced from 1.13 GB to 0.45 GB at $C = 10$ (2.51× reduction). and from 11.30 GB to 5.46 GB at $C = 100$ (2.01× reduction) compared to the FHE baseline. Although the compression ratio decreases marginally due to reduced packing factor, FUSE-SPA maintains over 2× communication savings at scale.

A critical concern is whether the discretization of importance scores introduces cumulative noise that degrades Top-k selection as C increases. While the absolute error bound in Theorem 1 scales linearly with C, the relative selection fidelity remains stable. This stability arises because the true aggregate magnitudes (T_i, T_j) also grow linearly with C, the signal-to-noise ratio of the importance map is preserved. This allows the Top-k ordering to remain robust to discretization noise with stable relative fidelity when the network size grows.

H Time Component and Computational Efficiency Analysis

Table 8(a) provides a decomposition of the per-round execution time for GPT-2. In lightweight CKKS setting (FA/FP), FUSE-RDX yields the highest operational efficiency with of up to 1.98× speedup over the baseline. However, FUSE-Fusion emerges as the more efficient variant of up to 2.14× speedup in NB. This divergence arises from two key factors: FUSE overhead and ciphertext processing cost. FUSE-Fusion incurs compression overhead ($\approx$ 1s) from its index exchange stage, raising per FL round time to $1.85 - 2.63$s. In contrast, FUSE-RDX utilizes direct update pairing with

Table 8: Detailed per-round execution time breakdown (in seconds) of FUSE across FL algorithms on GPT-2 and Llama-7B.

		FedAvg (FA)						FedProx (FP)						Norm Bounding (NB)							
k	Exp.	F_{ie}	F_{pack}	C	S_{agg}	T_S	Speed	F_{ie}	F_{pack}	C	S_{agg}	T_S	Speed	F_{ie}	F_{pack}	C	S_{agg}	T_S	Speed	T_D	Speed
100%	BL	–	–	1.73	1.28	3.01	1.00×	–	–	1.77	1.20	2.97	1.00×	–	–	5.35	10.22	15.57	1.00×	–	–
75%	FS	0.98	0.02	1.27	0.48	2.75	1.09×	1.00	0.02	1.28	0.47	2.77	1.07×	1.55	0.02	3.38	9.03	13.98	1.11×	13.41	1.16×
75%	FF	1.00	0.34	1.02	0.27	2.63	1.14×	1.00	0.35	1.02	0.28	2.65	1.12×	1.60	0.36	2.55	6.92	11.43	1.36×	10.83	1.44×
50%	FR	–	0.20	0.98	0.37	**1.55**	1.94×	–	0.20	0.94	0.36	**1.50**	1.98×	–	0.20	2.79	6.69	9.68	1.61×	–	–
50%	FS	1.00	0.02	0.85	0.33	2.20	1.37×	0.99	0.02	0.84	0.32	2.17	1.37×	1.51	0.02	2.17	7.52	11.22	1.39×	10.71	1.45×
50%	FF	1.00	0.33	0.66	0.18	2.17	1.39×	1.00	0.34	0.67	0.18	2.19	1.36×	1.60	0.35	1.64	5.31	8.90	1.75×	8.30	1.88×
25%	FS	0.99	0.01	0.45	0.17	1.62	1.86×	0.97	0.01	0.43	0.19	1.60	1.86×	1.53	0.01	1.14	5.65	8.33	1.87×	7.79	2.00×
25%	FF	1.00	0.35	0.40	0.10	1.85	1.63×	1.00	0.34	0.40	0.13	1.87	1.59×	1.60	0.35	0.91	4.41	**7.27**	2.14×	**6.67**	2.33×

(a) GPT-2 Time Component Analysis

		FedAvg (FA)						FedProx (FP)						Norm Bounding (NB)							
k	Exp.	F_{ie}	F_{pack}	C	S_{agg}	T_S	Speed	F_{ie}	F_{pack}	C	S_{agg}	T_S	Speed	F_{ie}	F_{pack}	C	S_{agg}	T_S	Speed	T_D	Speed
100%	BL	–	–	35.34	10.06	45.40	1.00×	–	–	34.14	10.87	45.01	1.00×	–	–	77.84	83.94	161.78	1.00×	–	–
75%	FS	11.64	0.42	21.94	7.82	41.82	1.09×	11.66	0.42	22.20	8.05	42.33	1.06×	19.88	0.43	49.66	67.21	137.18	1.18×	128.94	1.25×
75%	FF	11.88	2.87	11.54	3.79	30.08	1.51×	11.76	2.85	11.42	3.80	29.83	1.51×	17.91	2.52	32.37	22.17	74.97	2.16×	68.94	2.35×
50%	FR	–	1.26	13.49	5.07	**19.82**	2.29×	–	1.36	13.56	5.42	**20.34**	2.21×	–	1.58	35.05	28.00	64.63	2.50×	–	–
50%	FS	11.96	0.29	14.37	5.32	31.94	1.42×	12.19	0.29	14.03	5.11	31.62	1.42×	17.76	0.28	32.70	45.77	96.51	1.68×	90.71	1.78×
50%	FF	12.06	1.57	8.35	2.55	24.53	1.85×	11.69	1.60	8.54	2.52	24.35	1.85×	18.17	1.70	22.28	16.10	58.25	2.78×	52.14	3.10×
25%	FS	12.32	0.15	7.09	2.65	22.21	2.04×	11.74	0.15	6.90	2.34	21.13	2.13×	17.48	0.14	15.83	25.94	59.39	2.72×	54.23	2.98×
25%	FF	11.09	0.88	5.00	1.27	18.24	2.49×	11.62	0.78	5.02	1.20	18.62	2.42×	18.55	0.94	11.09	10.24	**40.82**	3.96×	**33.36**	4.85×

(b) LLaMA Time Component Analysis

Notes: *k: parameter retention ratio. Exp.: experimental variant (BL: baseline, FR: FUSE-RDX, FS: FUSE-SPA, FF: FUSE-Fusion); F_{ie}: index exchange time (including cryptographic overheads + index server aggregation); F_{pack}: packing operations; C (cryptographic overhead) and S_{agg} (server aggregation) refer exclusively to the update aggregation phase; T_S / T_D: total runtime for single/dual context CKKS; Bold blue: fastest; green: speedup vs. BL.*

only 0.02s packing overhead, maintaining a leaner $1.50 - 1.55$s round time. However, under NB, the use of larger CKKS parameters increases per-ciphertext operation time, and expensive non-linear operations shifts the dominant cost to the update aggregation phase. Here, the overhead of index exchange for FUSE-Fusion diminishes to $< 20\%$ of total time. Thus, minimizing the number of ciphertexts becomes critical for achieving overall speedup. By leveraging a dual-CKKS context, we further reduce computational complexity, achieving a $2.33\times$ acceleration (6.67s/round) for the NB protocol. Table 8(b) confirms that this trend scales to Llama-7B, the acceleration factors consistently exceed those observed in GPT-2, with dual-context speedups reaching $4.85\times$. This demonstrates that FUSE's compression benefits become increasingly significant as model complexity and base communication costs grow, shifting the viability of privacy-preserving LLM training.

I Comparison with Other Work

By evaluating zero-shot perplexity on the GPT-2 benchmark (Radford et al., 2019), the FUSE method preserves model performance, achieving perplexities (Lambada $33.84 \pm 0.79\%$, WikiText-103 $25.19 \pm 0.43\%$, 1BW $49.16 \pm 0.93\%$) that are comparable to the official baseline (35.13, 37.50, 75.20). While several FL frameworks (e.g., OpenFedLLM (Ye et al., 2024), FederatedScope (Kuang et al., 2024)) support LLM compression but lack of FHE integration.

Recent efforts to bridge the FL-FHE efficiency gap, such as FedSHE-CQ (Wei et al., 2026) encrypts packed index masks to aggregate client updates over the selection union. While this guarantees local index retention, the union size may expand as $k \leq |U| \leq \min(Ck, d)$, depending on selection overlap from clients. Under our FedAvg setup ($N = 2^{16}$, $r = 25\%$) with its native 4-bit packing, their per-client communication spans between ≈ 0.53 GB (best-case) and 1.75 GB (worst-case) on Llama-7B. In contrast, FUSE-SPA achieves overlap-invariant bandwidth stability via exact-k consensus selection, transmitting ≈ 0.61 GB per client. This is slightly above their best-case but $\approx 2.86\times$ below its worst case at the cost of consensus Top-k (Theorem 1).

MedHE (Yesmin, 2025) demonstrate effective error-compensated top-k sparsification for moderate-scale full models on specific medical classification tasks. However, its reliance on packing 64 values per slot requires extreme bit-width reduction, which constrains per-parameter precision. This paradigm is less suited for the information-dense updates of PEFT like LoRA for more generic, complex generative tasks, where high numerical fidelity is essential to preserve the integrity of the low-rank manifold. In contrast, FUSE is architected as a more generic compression solution co-designed with LoRA, balancing ciphertext efficiency with the stability necessary for large-scale model convergence.

BatchCrypt (Zhang et al., 2020) batches r-bit two's-complement gradients into a single Paillier plaintext, relying on exact decryption. It prevents aggregate overflow by widening the quantisation range with the client count m. By contrast, FUSE-RDX instead absorbs the headroom into the encoding base within each CKKS SIMD slot via the radix condition $B_{\text{rdx}} > 2 \cdot A \cdot W_{\text{total}}$. This keeps encoded values per slot within the approximate CKKS modulus budget, with per-value resolution set by Δ_{FUSE}.

Other existing FL-FHE works exhibit disparities in security parameters, missing hyperparameters, and reporting metrics that complicate quantitative comparison. FedPHE (Yan et al., 2024) uses a degree of $N = 8,192$ that OpenFHE cannot replicate due to its enforced security requirement (ring degree $\geq 16,384$) (Al Badawi et al., 2022). Additionally, unreported key hyperparameters (mean-threshold) make a fair reproduction and quantitative comparison of communication and runtime infeasible. Although direct runtime comparison is challenging, the effect of compression on model accuracy can be evaluated. FedPHE incurs $\approx 1.58\%$ accuracy loss on ResNet-20 relative to plaintext, and about 2.5% compared to other CKKS-based methods under FedAvg. In comparison, FUSE exhibits a maximum perplexity degradation of $< 1\%$ across all datasets under FedAvg, demonstrating a more balanced trade-off between communication reduction and model performance preservation.

On a different front, Hybrid Homomorphic Encryption (HHE) (Correia et al., 2026) achieves a smaller per-ciphertext size (54.1 KB vs. FUSE's 3.14 MB) via transciphering (PASTA for transmission, BFV for computation), but incurs a prohibitive $\approx$ 62,748$\times$ ($5804s$ for four clients on FedAvg) slower per-round server aggregation, making the approach infeasible for LLM-scale FL. FUSE provides compression from a different aspect by reducing the number of encrypted updates, preserving time performance without altering the encryption scheme.

J Perplexity Analysis (Extended)

The empirical evaluation establishes clear performance hierarchies among aggregation protocols under FHE constraints. Across all benchmarks, FUSE maintains minimal perplexity (PPL) degradation of $< 1\%$ relative to baseline performance. Figures 5–7 illustrate the PPL relative to cumulative communication, revealing that the NB consistently achieves the lowest final perplexity across all benchmarks. The convergence trajectories for FedProx (red line) consistently shift to the left of the FedAvg (blue line) across all datasets. Since both algorithms incur identical per-round communication for a fixed k, this horizontal displacement shows that FP converge faster than FA. Ultimately, FUSE-Fusion demonstrates exceptional robustness to sparsification intensity (k), delivering stable performance even under aggressive compression. Notably, FUSE hyperparameter tuning preserves constant per-round communication and computational overhead, as update granularity is adjusted within a fixed ciphertext transmission volume.

To assess whether FUSE's compression-induced utility degradation scales with model size, we further evaluate perplexity on GPT-2-XL ($\approx 8\times$ larger than GPT-2-S) across all FL algorithms and FUSE variants under an aggressive compression setting ($k = 25\%$ for FS & FF). Table 9 reports communication, runtime, and PPL change for GPT-2-XL alongside GPT-2-S for comparison. The compression and latency reductions remain effective as the model scales, consistently exceeding the improvement of over $2\times$ relative to the FHE baseline. For perplexity, GPT-2-XL's baseline is lower than GPT-2-S's, consistent with established language model scaling laws. More importantly, PPL degradation under FUSE remains controlled even as model size increases. The perplexity changes for GPT-2-XL ranges from $[-0.60\%, +0.74\%]$, which falls within the full-spectrum range observed on GPT-2-S $[-0.64\%, +0.88\%]$. This supports that FUSE's compression-utility trade-off is governed by protocol design rather than model scale, evidenced by stable utility across a $8\times$ parameter increase under our most aggressive compression setting.

Table 9: Detailed per-round performance of FUSE across FL algorithms on GPT-2-XL.

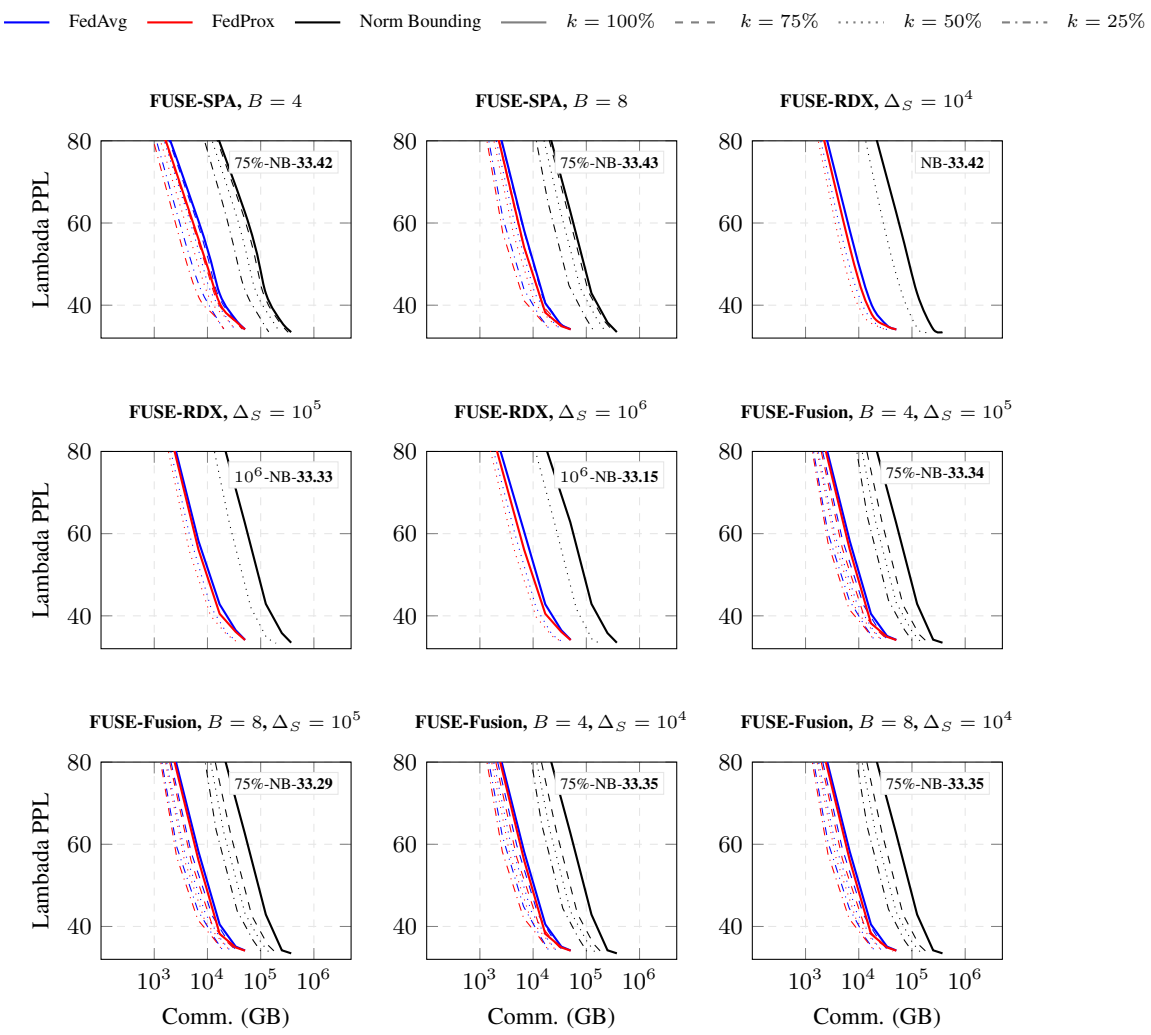
		FedAvg (FA)				**FedProx (FP)**				**Norm Bounding (NB)**			
		Time		Comm.		Time		Comm.		Time		Comm.	
k	**Exp.**	Sec	Speed	MB	Ratio	Sec	Speed	MB	Ratio	Sec	Speed	MB	Ratio
100%	BL	12.81	1.00×	933	1.00×	13.05	1.00×	933	1.00×	55.82	1.00×	5,633	1.00×
50%	FR	**4.32**	2.97×	433	2.15×	**3.75**	3.48×	433	2.15×	27.53	2.03×	2,400	2.35×
25%	FS	4.37	2.93×	236	3.95×	4.36	2.99×	236	3.95×	23.13	2.41×	1,733	3.25×
25%	FF	4.45	2.88×	**159**	5.87×	4.18	3.12×	**159**	5.87×	**17.36**	3.22×	**900**	6.26×

(a) Communication and latency

	GPT-2-S			**GPT-2-XL**		
Dataset	Baseline	FUSE	Δ PPL (Range)	Baseline	FUSE	Δ PPL (Range)
Lambada	33.81 ± 0.35	33.29 ± 0.14	$[-0.34\%, +0.88\%]$	7.92 ± 0.25	7.59 ± 0.37	$[-0.29\%, +0.74\%]$
WikiText-103	25.19 ± 0.09	25.05 ± 0.05	$[-0.64\%, +0.55\%]$	12.87 ± 0.09	12.56 ± 0.11	$[-0.41\%, +0.65\%]$
1BW	49.15 ± 0.21	48.78 ± 0.16	$[-0.63\%, +0.81\%]$	38.07 ± 0.18	37.53 ± 0.40	$[-0.60\%, +0.73\%]$

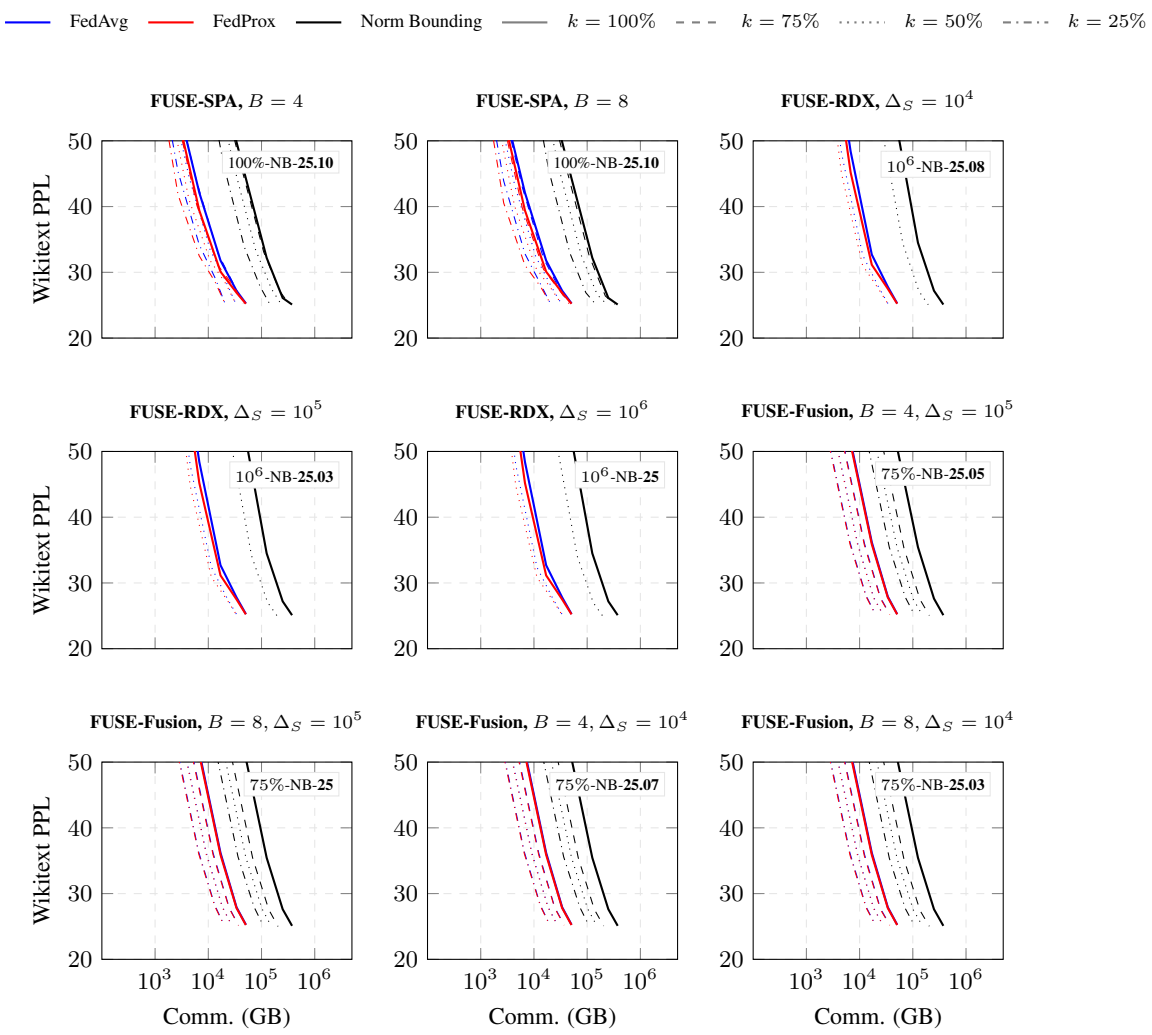
(b) Perplexity change (ΔPPL) of FUSE variants compared to respective FL baselines across GPT-2 models

Notes: *k: parameter retention ratio; Exp.: experimental variant (BL: baseline, FR: FUSE-RDX, FS: FUSE-SPA, FF: FUSE-Fusion); **Bold blue**: optimal value (lowest per-round latency / maximum compression ratio / lowest perplexity); green: relative speedup or compression factor vs. BL. GPT-2-XL evaluated under configuration $B = 8$, $\Delta_S = 10^5$.*



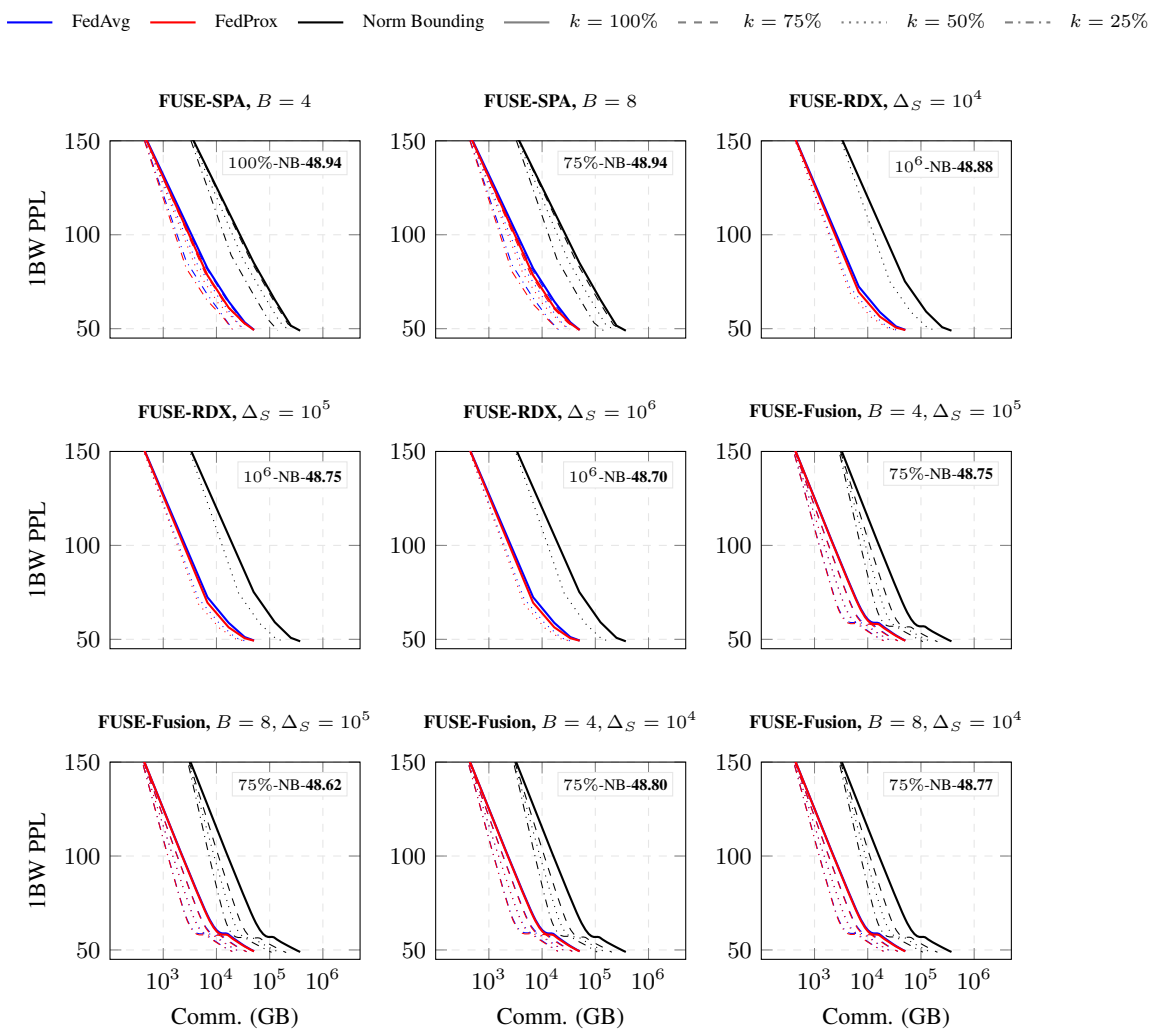
Notes: *This compare three variants: FUSE-SPA (bin size B), FUSE-RDX (scale factor Δ_S), and the hybrid FUSE-Fusion. The best converged PPL for each configuration is annotated in the top-right.*

Figure 5: Evaluation of lambada perplexity (PPL) over cumulative communication volume across FUSE variants on GPT-2s.



Notes: *This compare three variants: FUSE-SPA (bin size B), FUSE-RDX (scale factor Δ_S), and the hybrid FUSE-Fusion. The best converged PPL for each configuration is annotated in the top-right.*

Figure 6: Evaluation of Wikitext-103 perplexity (PPL) over cumulative communication volume across FUSE variants on GPT-2s.



Notes: *This compare three variants: FUSE-SPA (bin size B), FUSE-RDX (scale factor Δ_S), and the hybrid FUSE-Fusion. The best converged PPL for each configuration is annotated in the top-right.*

Figure 7: Evaluation of 1BW perplexity (PPL) over cumulative communication volume across FUSE variants on GPT-2s.