Privacy-Preserving Post Deployment Model Calibration with Fully Homomorphic Encryption

Shadman Mahmood Khan Pathan*, Qianlong Wang*, Jonathan Takeshita†, Sakan Binte Imran‡, Sachin Shetty*
*Department of Electrical and Computer Engineering, Old Dominion University, USA
†Department of Computer Science and School of Cybersecurity, Old Dominion University, USA
‡Department of Medicine, Sir Salimullah Medical College Mitford Hospital, Bangladesh
Email: spath004@odu.edu, q1wang@odu.edu, jtakeshi@odu.edu, sakanbinteimran.ssmc@gmail.com, sshetty@odu.edu

***Abstract*—Privacy rules often prohibit sharing raw data or even model internals across collaborating sites, yet deployed models must still cope with distribution shift. In this paper, we introduce a peer-to-peer protocol for post-deployment encrypted calibration that adjusts only the final linear layer of a classifier using the CKKS homomorphic encryption scheme [1]. The update is driven by misclassified samples whose penultimate features are shared only as ciphertext; to remain Homomorphic Encryption (HE) friendly, we replace softmax with a low-degree Chebyshev approximation, so all computations use additions, multiplications, and rotations on encrypted vectors. No raw features, logits, gradients, or plaintext parameters are exchanged, and no central server is required. We evaluate the method on MNIST and PathMNIST under a non-independent and identically distributed (non-IID) class skew with both identical and heterogeneous backbones connected through a shared feature interface. The results demonstrate that a single encrypted, per-sample calibration can reliably reallocate probability mass toward the correct class and often flip the decision. The approach provides a practical, privacy-preserving mechanism for targeted post-deployment adaptation and complements federated learning and secure multiparty training.**

***Index Terms*—Encrypted calibration, homomorphic encryption, federated learning, Chebyshev polynomial, non-IID data, privacy-preserving AI**

I. Introduction

Modern machine learning increasingly requires cross-institution collaboration under strict privacy rules (e.g., HIPAA, GDPR). Federated learning (FL) [2] trains without pooling raw data, yet client updates and metadata can still leak through gradient inversion or membership inference [3], [4]. A core challenge is data heterogeneity: partners hold non-IID data [5], which may slow convergence and harm generalization.

Homomorphic encryption (HE) enables computation on ciphertext [1]. Prior HE systems (CryptoNets [6], nGraph HE [7]) target fixed-model inference. We instead propose post-deployment encrypted calibration that adjusts only the final linear layer using signals formed entirely in cipher. To remain HE-friendly, we replace softmax with a low-degree Chebyshev approximation so inference or updates use only additions, multiplications, and rotations.

We propose a fully encrypted calibration framework under non-IID splits. In particular, misclassified samples trigger a short correction based on encrypted penultimate features. No features, logits, gradients, or plaintext parameters are revealed during model calibration. The protocol is serverless and supports identical or heterogeneous backbones. We assume an honest-but-curious peer seeking information about inputs/gradients/parameters. We limit exposure by keeping features/logits encrypted end-to-end, updating only the last layer, and using polynomial approximations; implementation uses CKKS approximate arithmetic [1]. Our loop performs targeted fixes using ciphertext-only evidence, avoiding raw-data exchange and reducing operational risk. In many cross-silo deployments, both data heterogeneity and model heterogeneity are unavoidable. Hospitals, banks, or agencies collect data from different populations and under different acquisition protocols, leading to non-IID distributions across sites. At the same time, institutions often deploy architectures that reflect their own constraints, so even when a common feature interface is agreed upon, the backbones need not match. These heterogeneous models and datasets must still maintain strict privacy rules that restrict sharing raw samples, gradients, or weights. To address this, we seek a lightweight, post-deployment calibration mechanism that can operate across heterogeneous datasets and models while keeping all shared signals encrypted. **Contributions.** (i) A peer-to-peer encrypted model-calibration protocol that adapts post-deployment by updating only the final layer, without aggregation or gradient sharing. (ii) Operation under non-IID splits with identical or heterogeneous backbones via a shared feature interface; tiny ciphertext-only updates correct errors. (iii) Practical runtime: encrypted inference plus at most 20 steps in $\sim$2–3 s per sample on CPU. Overall, this enables secure post-deployment adaptation and complements FL, secure MPC, and DP; the protocol can be applicable for broader settings with distribution shift and strict privacy constraints.

II. Related Work

Privacy-preserving collaborative learning spans several major threads. *Federated learning* (FL) avoids raw data sharing by averaging client updates on a coordinator [2]. A large body of work addresses statistical heterogeneity (client drift under non-IID data) via regularization or control terms (e.g., Fed-

Prox, FedDyn, MOON) [8]–[10]. Architecture heterogeneity is tackled by ensembling and teacher–student transfers (e.g., FedDF, FedGKT) and recent personalization for transformers (e.g., FedTP) [11]–[13]. Despite this progress, most FL methods still exchange model updates or logits and rely on a server. Our work targets a different point: a post-deployment, ciphertext-only, one-hop calibration that updates the last linear layer, without gradients, plaintext logits, or aggregation.

Homomorphic encryption (HE) enables computation on encrypted vectors; CKKS supports approximate arithmetic with packed SIMD [1]. Early systems such as CryptoNets and nGraph-HE demonstrated encrypted inference for fixed models [6], [7], but did not address adaptation after deployment. To keep multiplicative depth small, we replace softmax with a low-degree Chebyshev surrogate, allowing both inference and the corrective step to use only additions, multiplications, and rotations under CKKS.

Complementary defenses control leakage during training. *Differential privacy* perturbs updates or statistics to bound contributions of single examples, but noise can hurt accuracy under severe non-IID splits [14]. *Secure aggregation* and related cryptographic protocols protect client updates during coordination [15], yet typically require synchronous rounds and orchestration. Our approach avoids added noise and central coordination by performing tiny, on-demand fixes in cipher.

Encrypted learning beyond one-time inference is emerging. Some methods integrate encryption with attribution or filtering to curb harmful updates [16], but they do not provide a lightweight, ciphertext-only *last-layer* correction triggered by individual errors. In contrast, we show a case where misclassified penultimate features drive a short encrypted update loop that never reveals features, logits, gradients, or parameters. For context, we compare against softmax/temperature scaling (plaintext calibration baselines) and a small FedAvg baseline to illustrate communication–accuracy trade-offs.

III. Methodology

We introduce a secure *post deployment* collaborative learning framework evaluated on data under *non IID* splits. The pipeline has four stages: (i) simulation of data class skew across sites; (ii) local pretraining with a Chebyshev aware output head; (iii) encrypted inference with encrypted error signaling; and (iv) encrypted last layer calibration. All computations keep features and logits encrypted with the CKKS scheme [1].

Design goals.

- G1: No central aggregator and no plaintext exchange.
- G2: Compatibility with homomorphic encryption using only additions, multiplications, and rotations.
- G3: One hop peer to peer communication.
- G4: Bounded compute and communication per corrected sample.
- G5: Clear failure modes under severe class skew.

A. Threat Model and Privacy Goals

We consider two or more honest-but-curious sites that collaborate after deployment without revealing raw data or the internal states of their models. Each site holds its own private dataset and a locally trained model; datasets are generally non-IID across sites, and model architectures may also differ as long as they expose a shared penultimate feature dimension. The parties follow the protocol faithfully, but may attempt to infer additional information from any ciphertexts they receive. Our privacy and robustness goals are:

- **No raw data exposure.** Images, labels, and feature vectors remain in plaintext only inside the originating site. Any features sent across sites are always encrypted with CKKS.
- **No gradient or weight sharing.** We never transmit plaintext gradients, logits, or parameters. Only encrypted penultimate features and encrypted error signals are exchanged, and each site updates only its own last layer.
- **Support for data and model heterogeneity.** Sites may train on highly skewed class distributions (data heterogeneity) and use different backbone architectures (model heterogeneity). The protocol must function reliably when both distributions and architectures differ, provided a common final-layer feature dimension is agreed upon.
- **Limited cryptographic key sharing.** Each site keeps its CKKS secret key private. Relinearization and rotation keys are shared only when necessary for packed matrix–vector operations and only over authenticated channels. Keys can be rotated or refreshed by policy.
- **Ciphertext-only calibration.** The calibration loop performs all computations— logits, error computation, and parameter updates— entirely in ciphertext using additions, multiplications, and rotations. No intermediate plaintext logits, gradients, or activations are revealed.

We do not attempt to defend against timing or side-channel attacks, nor adversaries who deviate from the protocol or inject malformed ciphertexts. These adversarial models are left for future work.

B. Collaborative learning setting

We consider two autonomous sites A and B that each hold private data and train a local model. There is no central server. The only messages that cross sites are (i) encrypted feature vectors for selected samples and (ii) encrypted calibration signals. Raw examples and model states remain inside the local boundary.

Roles.

- **Sites A and B:** Train a compact CNN with a Chebyshev compatible output head. Run inference and perform light weight last layer calibration from encrypted signals.
- **Channel:** A mutually authenticated Transport Layer Security link that carries CKKS ciphertexts. Rotation and relinearization keys that are needed for packed dot products are shared one way from the site that performs the update. Secret keys never leave the site. There is no global aggregation.

High level loop.

1) Each site trains on its own non IID digit split.
2) When B observes a misclassified test image, it encrypts the penultimate feature and sends it to A.
3) The receiver computes encrypted logits and a Chebyshev based softmax, forms an encrypted error signal, and applies a few last layer updates while everything remains encrypted.
4) Only the updated local parameters are kept. Nothing is revealed in plaintext. The procedure is a calibration step after deployment and not federated training.

C. Notation and CKKS parameters

Let $\phi(\cdot) \in \mathbb{R}^d$ be the penultimate feature map, $W \in \mathbb{R}^{d\times C}$ and $b \in \mathbb{R}^C$ the final linear layer, with $C{=}10$ classes for MNIST. Parties exchange $f^e = \texttt{Enc}(\phi(x))$ using CKKS with coefficient modulus levels $[60, 40, 40, 60]$, polynomial modulus degree $N = 2^{15}$, and global scale 2^{40}. Features are packed into a single ciphertext using single instruction multiple data and dot products are implemented with rotations.

We monitor the ciphertext budget so that multiplicative depth remains below three for matrix vector multiplication, the Chebyshev transform, and the update. Secret keys never cross sites. Public relinearization and rotation keys are exchanged once.

Chebyshev softmax. We approximate exp on $[-1, 1]$ by $P(x) = \sum_{k=0}^{n} c_k T_k(x)$ with degree $n \le 4$ to control noise growth. Centering and min to max scaling are applied in cipher using plaintext constants so that logits fall in $[-1, 1]$.

Per-sample HE cost and payload.: With packed CKKS, computing logits $z^e{=}W^\top f^e$ costs $\mathcal{O}(Cd)$ multiplications/additions and $\mathcal{O}(C \log d)$ rotations. The degree-n Chebyshev softmax $\widetilde{S}$ adds $\mathcal{O}(Cn)$ multiplications/additions with no rotations. The last-layer update in (1) again incurs $\mathcal{O}(Cd)$ multiplications/additions and $\mathcal{O}(C \log d)$ rotations. The one-hop ciphertext payload is comparable to the size of the encrypted feature f^e (tens of KB for $d \le 1024$). We adopt SEAL-compatible CKKS parameters, providing $\ge$ 128-bit classical security. A polynomial modulus degree of $N = 2^{15}$ and a coefficient modulus chain $[60, 40, 40, 60]$ support ciphertext–ciphertext multiplications in the Chebyshev polynomial and accommodate the depth of the encrypted calibration loop. SIMD packing allows a full 64-dimensional feature vector to be encoded per ciphertext, matching the feature interface in both identical and heterogeneous models. All experiments remain within the CKKS noise budget without bootstrapping.

D. Collaborative learning versus federated learning

Collaborative learning is a post-deployment, peer-to-peer scheme with no server or global averaging: each site trains locally and, when a peer flags an error, applies an encrypted last-layer correction. In contrast, FL aggregates client updates on a coordinator via synchronous rounds to build a global model. Our protocol exchanges only CKKS-encrypted penultimate features (optionally an encrypted head) and performs short, on-demand calibration, reducing the privacy surface (ciphertexts only; last-layer updates) and avoiding orchestration and drift control.

Fig. 1. Encrypted calibration protocol. The figure shows non IID partitioning of MNIST, local training with a Chebyshev head, feature encryption with CKKS, encrypted inference, error detection in cipher, and the short encrypted update loop.

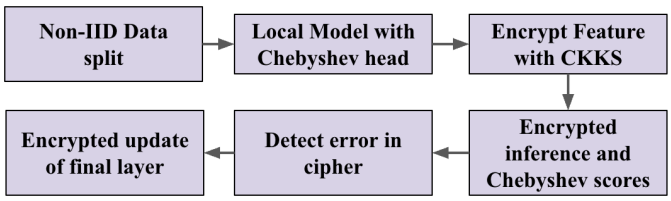

Fig. 2. End-to-end encrypted calibration pipeline from non-IID simulation to encrypted model correction.

E. Decentralized collaborative learning flow

We consider a multi party setting with the same steps at each site. When Site B encounters a misclassified sample it invites Site A to perform an encrypted calibration of A's head. No raw images, logits, or gradients are shared.

Step by step process for A and B.

1) **Local pretraining.** Train a CNN with a Chebyshev head on the local non IID data split.
2) **Encrypted feature sharing (B→A).** Extract the penultimate feature f_B, encrypt it to f_B^e with CKKS, and send f_B^e and an encrypted one hot label or an encrypted mismatch bit to A.
3) **Encrypted inference and error at A.** Compute $\hat{y}^e = \texttt{ChebyshevSoftmax}(W^\top f_B^e + b)$ and the encrypted error $\delta^e = y - \hat{y}^e$.
4) **Encrypted last layer update at A.** For $i = 1, \ldots, C$ update
$$W_i \leftarrow W_i + \eta\, \delta_i^e\, f_B^e, \qquad b_i \leftarrow b_i + \eta\, \delta_i^e,$$
using only additions and multiplications supported by CKKS.
5) **Return calibrated head (A→B).** Send the calibrated (W, b) encrypted under the public key of B or send an encrypted delta $(\Delta W, \Delta b)$. B replaces its local head and retests.

F. Simulating data heterogeneity

We induce cross-site imbalance by splitting digits into a primary set $\mathcal{C}_p = \{0, 1, 2, 3, 4\}$ and secondary set $\mathcal{C}_s = \{5, 6, 7, 8, 9\}$, keeping all samples for $\mathcal{C}_p$ and a fraction $\alpha = 0.2$ for $\mathcal{C}_s$:

$$n_j = \begin{cases} N_j, & j \in \mathcal{C}_p, \\ \alpha N_j, & j \in \mathcal{C}_s. \end{cases}$$

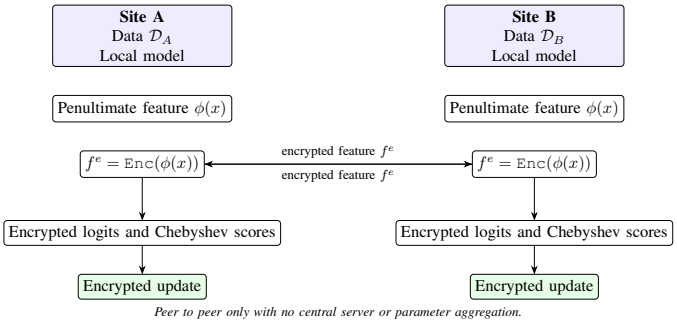

Fig. 3. Two-site encrypted collaborative calibration. Only CKKS-encrypted penultimate features are exchanged. Each peer updates its head locally in cipher.

This yields a controlled non-IID split for Site A; other sites use symmetric/complementary splits. A five-client FL baseline (two digits per client) is included to stress aggregation.

G. Pretraining with a polynomial-compatible output

Each site trains a compact CNN and replaces softmax with a CKKS-friendly Chebyshev surrogate

$$\tilde{S}_i(x) = \frac{P(x_i)}{\sum_j P(x_j)}, \quad P(x) = \sum_{k=0}^{n} c_k T_k(x),$$

which approximates exp, avoids divisions/exponentials under HE, and behaves like a probability vector. We vary $n \in \{2, 3, 4, 5\}$ to balance shaping quality vs. multiplicative depth/noise/runtime (higher n improves shaping but increases depth).

H. Encrypted inference and misclassification detection

Using TenSEAL/CKKS, the penultimate feature $f = \phi(x)$ is encrypted to $f^e = \texttt{Enc}(f) \in \mathbb{C}^d$. Encrypted logits and scores are

$$z_i^e = W_i^\top f^e + b_i, \quad i = 1, \ldots, C, \qquad \hat{y}^e = \widetilde{S}(z^e).$$

A sample is flagged when $\arg\max(\hat{y}^e) \neq y$. No plaintext logits/confidences are revealed; centering and min–max scaling of z^e use plaintext constants and do not increase depth.

I. Encrypted model calibration (lightweight)

Given one-hot $y \in \mathbb{R}^C$, set $\delta^e = y - \hat{y}^e$ and update only the head with plaintext step size η:

$$W_i^{(t+1)} = W_i^{(t)} + \eta\, \delta_i^e\, f^e, \qquad b_i^{(t+1)} = b_i^{(t)} + \eta\, \delta_i^e, \;\; i = 1, ..C \tag{1}$$

Packed CKKS computes $W_i^\top f^e$ via rotations and additions; no exponentials or divisions. Stop when $\|\delta^{e,(t)}\|_2 \leq \epsilon$ or $t \geq T_{\max}$ (with $\epsilon = 10^{-3}$, $T_{\max} = 20$); empirically $\|\delta^{e,(t)}\|_2^2$ decreases. Communication is ciphertext-only: a packed f^e to the peer and, in return, either (W, b) re-encrypted for the requester or encrypted deltas $(\Delta W, \Delta b)$; no raw inputs, logits, gradients, or plaintext parameters leave a site.

Algorithm 1 Encrypted calibration protocol for one MNIST sample

Require: Encrypted feature x^e, weights W, bias b, one hot label y, learning rate η

$\hat{y}^e \leftarrow \texttt{ChebyshevSoftmax}(W^\top x^e + b)$
$\delta \leftarrow y - \hat{y}^e$
for $i = 1$ to C **do**
  $W_i \leftarrow W_i + \eta \cdot \delta_i \cdot x^e$
  $b_i \leftarrow b_i + \eta \cdot \delta_i$
end for
if $\|\delta\|_2 \leq \epsilon$ **or** $t \geq T_{\max}$ **then**
  break
end if
return Updated (W, b)

Algorithm and calibration protocol: HE compatibility note. All steps use additions and multiplications between ciphertexts and plaintext scalars, which aligns with CKKS capabilities. The Chebyshev approximation removes unsupported operations.

Per-sample HE cost and payload.: With packed CKKS, computing logits $z^e{=}W^\top f^e$ requires $\mathcal{O}(Cd)$ multiplications/additions and $\mathcal{O}(C \log d)$ rotations. The degree-n Chebyshev softmax $\widetilde{S}$ adds $\mathcal{O}(Cn)$ multiplications/additions with no rotations. The last-layer update in (1) again incurs $\mathcal{O}(Cd)$ multiplications/additions and $\mathcal{O}(C \log d)$ rotations. The one-hop ciphertext payload is roughly the size of the encrypted feature f^e (tens of KB for $d{\leq}1024$).

J. Communication footprint per correction

Packed CKKS transmits one vector ciphertext per misclassified sample (optionally an encrypted head). For $d \leq 1024$ with modulus levels $[60, 40, 40, 60]$, each direction is only tens of kilobytes, so a single encrypted calibration fits comfortably on cross-site links and amortizes cost over occasional errors—unlike synchronous FL rounds that repeatedly broadcast full models.

IV. Experiments and Results

We evaluate a serverless, peer-to-peer collaboration between two sites (A and B) on non-IID data. Only penultimate features and calibration signals are exchanged as CKKS ciphertexts; raw images and model states never leave a site. We consider (i) heterogeneous backbones with a shared 64-D feature interface and (ii) identical backbones trained on different non-IID splits. Calibration is triggered by any misclassified test sample: the peer sends the encrypted penultimate feature and an encrypted label (or mismatch bit) for a short in-cipher last-layer update.

A. Dataset, Models, and Calibration Loop

Dataset. *MNIST* (10 classes, 28×28 grayscale) is split non-IID with `primary_ratio` $= 0.8$: a site retains all samples from its *primary* digits and a small subset from *secondary* digits. We use A: $\{0, 1, 2, 3, 4\}$, B: $\{5, 6, 7, 8, 9\}$; images are normalized to $[0, 1]$.

Dataset. *PathMNIST* (9 classes, 28×28×3 RGB) extends MNIST-like classification to histopathology tiles, providing a realistic medical benchmark for privacy-preserving inference. Because PathMNIST preserves MNIST-scale resolution and uses simple convolutional backbones, it enables a drop-in replacement for our 64-D feature interface without modification of the CKKS pipeline or the Chebyshev head. We adopt the standard MedMNIST train/val/test split and apply the same non-IID partitioning strategy (A: classes 0–3, B: classes 4–8), normalizing all channels to $[0, 1]$.

Backbones and head. Each site trains a compact CNN. In the heterogeneous regime the CNNs differ but expose a common 64-D penultimate feature; in the identical regime the backbones match and only data splits differ. The classifier replaces softmax with a Chebyshev surrogate,

$$\tilde{S}_i(x) = \frac{P(x_i)}{\sum_j P(x_j)}, \quad P(x) = \sum_{k=0}^{n} c_k T_k(x), \quad n \in \{3, 4, 5\},$$

and logits are centered and min–max scaled in cipher using plaintext constants.

Calibration loop. On a flagged error we run at most $T_{\max}=20$ encrypted last-layer updates with $\eta \in \{0.05, 0.1\}$ and early stop if $\|\delta^{(t)}\|_2 \leq 10^{-3}$. The loop uses only ciphertext–plaintext additions/multiplications; exponentials/divisions are avoided by the polynomial head.

B. HE Parameters and Runtime Overhead

Experiments run on a 32-core AMD EPYC 7502 CPU with 128 GB RAM, no GPU. CKKS uses coefficient modulus levels `[60,40,40,60]` and global scale 2^{40} with packed SIMD. Secrets never leave a site; rotation/relinearization keys are exchanged once over a mutually authenticated channel. For each misclassified sample, encrypted inference plus 20 updates completes in $\sim 2.0\,\mathrm{s}$ on MNIST. All runs fix `numpy`/`tensorflow` seeds to 42 and use CPU-only deterministic execution.

C. Model and Protocol Details

In the identical-model setting, both sites use a compact CNN with a 3×3 convolution (32 filters), ReLU, max-pooling, flatten, a 64-unit dense layer, and a 10-class Chebyshev head. In the different-model setting, Site A uses the same shallow CNN, while Site B uses two conv blocks (5×5 with 16 filters and 3×3 with 32 filters) with pooling, then flatten, a 64-unit dense layer, and the Chebyshev head. Both expose a shared 64-D penultimate feature for encrypted calibration. The local model: Conv2D(32, 3×3)+ReLU+MaxPool, Flatten, Dense(64, ReLU), Dense(10), Chebyshev head

$$\tilde{S}_i(x) = \frac{P(x_i)}{\sum_j P(x_j)}, \quad P(x) = \sum_{k=0}^{n} c_k T_k(x).$$

Encrypted calibration steps: (1) select a misclassified (x_{test}, y); (2) compute penultimate $f = \phi(x_{\text{test}})$; (3) encrypt $f^e = \text{Enc}_{\text{CKKS}}(f)$; (4) scores $\hat{y}^e = \tilde{S}(W^\top f^e + b)$; (5) error $\delta = y - \hat{y}^e$; (6) update $W_i \leftarrow W_i + \eta\,\delta_i\,f^e$, $b_i \leftarrow b_i + \eta\,\delta_i$ for $i = 1, \ldots, 10$.

D. Metrics and Baselines

We report top-1 accuracy; micro/macro precision–recall–F1; macro OvR AUROC; per-sample calibration success (true-class probability increase and label flip); encrypted $\|\delta^{(t)}\|_2$ across steps; and wall-clock per corrected sample. Baselines include a standard softmax head, temperature scaling, and a five-client FedAvg configuration (two digits per client) to contextualize communication versus accuracy/AUROC.

E. Results Discussion

Effect under data heterogeneity: We first evaluate two sites with identical CNN backbones but non-IID class partitions (0–4 vs. 5–9). The encrypted calibration increases cross-site accuracy and AUROC despite the severe distribution shift.

Effect under combined data and model heterogeneity: We then evaluate two sites with different CNN backbones sharing only a 64-dimensional penultimate feature interface. Even under architectural mismatch, the ciphertext-only calibration raises the true-class probability from as low as 0.01–0.02 to approximately 0.30 and flips the final decision in many cases.

Interpretation: Across both heterogeneity regimes, the improvement arises from a targeted redistribution of probability mass in response to hard, non-IID examples. Despite the absence of shared weights, gradients, or logits, the encrypted last-layer update behaves like a lightweight, privacy-preserving fine-tuning step for the final classifier head.

As summarized in Table I, our method improves AUROC under both regimes. Across both settings, the ciphertext-only calibration loop improves true-class probability and flips misclassifications without exposing features, logits, gradients, or plaintext parameters. Improvements are modest for a single calibrated example under severe non-IID shift, as expected, while communication and runtime remain lightweight.

We also evaluate the encrypted calibration loop on PathMNIST to demonstrate generality beyond digit classification. Despite the increased visual complexity and 9-way imbalance, ciphertext-only calibration remains effective: on a representative hard sample (class 6 misclassified as class 2), the true-class score increases from $0.073 \to 0.241$ after 20 encrypted steps, with a clean label flip. Cross-site transfer (B→A) improves from Acc $0.152 \to 0.224$ and AUROC $0.612 \to 0.673$ under identical models. Under heterogeneous CNNs, the transfer improves from Acc $0.167 \to 0.238$ and AUROC $0.587 \to 0.645$. These results show that the ciphertext-only calibration mechanism generalizes to clinical-style datasets without modifying the HE pipeline or backbone architecture.

F. Communication footprint and head comparison

Per-sample encrypted calibration communicates 0.21–0.26 MB, versus 26.5 MB for a 5-round, 2-client FedAvg run. At site A, softmax: Acc 0.9790, AUROC 0.9992, NLL 0.0763, ECE 0.0036; Chebyshev (HE-friendly): Acc 0.9646, AUROC 0.9932, NLL 0.7796, ECE 0.4902. Softmax calibrates better in plaintext, while the Chebyshev surrogate enables the fully encrypted loop. One-shot payload includes a 64-D encrypted feature (∼16–64 KB) and one encrypted head return

TABLE I
SUMMARY ACROSS HETEROGENEITY SETTINGS; CKKS WITH CHEBYSHEV UPDATES (20 STEPS); PER-SAMPLE RUNTIME $\sim$2 S.

Setting	Val. A (%)	Val. B (%)	Transfer gain	Calibration highlight
Identical models	91.27	94.03	Acc $1.02 \rightarrow 10.28$, AUROC $0.404 \rightarrow 0.474$	True class $0.1100 \rightarrow 0.3183$ with flip; monotone loss decrease
Different backbones	96.39	95.51	Acc $11.64 \rightarrow 18.69$, AUROC $0.3122 \rightarrow 0.4165$	True class $0.0126 \rightarrow 0.3070$ with flip; smooth loss descent

TABLE II
PATHMNIST SUMMARY UNDER IDENTICAL AND HETEROGENEOUS BACKBONES.

Setting	Val. A (%)	Val. B (%)	Transfer gain	Calibration highlight
Identical models	82.41	84.77	Acc $15.2 \rightarrow 22.4$, AUROC $0.612 \rightarrow 0.673$	True-class $0.073 \rightarrow 0.241$ with flip
Different backbones	85.93	83.22	Acc $16.7 \rightarrow 23.8$, AUROC $0.587 \rightarrow 0.645$	True-class $0.054 \rightarrow 0.228$ with flip

($\sim$200 KB). FedAvg accounting counts one global broadcast and two client uploads per round.

V. CONCLUSION AND FUTURE WORK

We introduced a peer-to-peer, ciphertext-only calibration framework that updates only the final classifier layer under CKKS with a Chebyshev surrogate head, revealing no plaintext features, logits, gradients, or weights. On MNIST, a single encrypted correction typically lifts the true-class probability from ~ 0.11 to ~ 0.30, increases cross-site accuracy from $\sim$0.10 to $\sim$0.19, and improves AUROC under both identical and heterogeneous backbones. Each fix runs in $\sim$2 s on CPU and communicates only $\sim$0.21–0.26 MB, dramatically smaller than even modest FL baselines.

We demonstrate that the same ciphertext-only calibration generalizes to PathMNIST, a 9-class medical histopathology benchmark. Despite higher visual complexity and a multi-class imbalance, encrypted updates raise the true-class score (e.g., $0.073 \rightarrow 0.241$), flip misclassifications, and improve B$\rightarrow$A transfer accuracy and AUROC under both identical and heterogeneous backbones without modifying the HE pipeline or feature interface. This establishes that the proposed mechanism is not tied to digit data but applies to realistic, privacy-sensitive domains.

ACKNOWLEDGMENTS

This project is based on work in part supported by the National Science Foundation (NSF) under Grant No. 2447364 and COVA CCI under Grant No. C-3Q25-ODU-01. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and COVA CCI.

REFERENCES

[1] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International conference on the theory and application of cryptology and information security*. Springer, 2017, pp. 409–437.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[4] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[6] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *International conference on machine learning*. PMLR, 2016, pp. 201–210.

[7] F. Boemer, V. Costan, R. Cammarota *et al.*, "ngraph-he: A graph compiler for deep learning on homomorphically encrypted data," *Privacy Enhancing Technologies*, 2019.

[8] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[9] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.

[10] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 713–10 722.

[11] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *Advances in neural information processing systems*, vol. 33, pp. 2351–2363, 2020.

[12] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, p. 2032, 2022.

[13] H. Li, Z. Cai, J. Wang, J. Tang, W. Ding, C.-T. Lin, and Y. Shi, "Fedtp: Federated learning by transformer personalization," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 10, pp. 13 426–13 440, 2023.

[14] T. Fukami, T. Murata, K. Niwa, and I. Tyou, "Dp-norm: Differential privacy primal-dual algorithm for decentralized federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5783–5797, 2024.

[15] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[16] X. Chen, H. Yu, X. Jia, and X. Yu, "Apfed: Anti-poisoning attacks in privacy-preserving heterogeneous federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5749–5761, 2023.