# Privacy-Preserving Multi-Institutional Clinical Prediction via Multiparty Encrypted Newton-CG Logistic Regression

Xiaokang Dai[a,b], Weijie Miao[a,c], Wenyuan Wu[a,*], Yong Feng[a]

[a] *Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China*
[b] *School of Computer Science and Engineering, Chongqing Sanxia University of Science and Technology, Chongqing 404100, China*
[c] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China*

**Abstract**

Multi-institutional clinical prediction can benefit from larger and more diverse patient populations, but privacy and data-governance constraints often prevent pooling patient-level records. This study aims to enable collaborative logistic-regression training while keeping such records local. We developed a framework combining multiparty Cheon–Kim–Kim–Song (CKKS) homomorphic encryption with Newton–conjugate gradient optimization. Each institution computes local gradient and damped Hessian contributions, while only encrypted fixed-dimensional statistics are aggregated. The global Newton system is solved by encrypted preconditioned conjugate gradient with diagonal preconditioning and delayed division, making the dominant homomorphic cost depend mainly on model dimension. We evaluated the framework on nine clinical-prediction tasks from NHANES 2017–2018 and on the PIMA Diabetes and Breast Cancer Wisconsin datasets. Across the nine NHANES tasks, mean absolute deviations from pooled-data plaintext Newton training were below 0.002 for area under the receiver operating characteristic curve, area under the precision–recall curve, and average precision. Five-fold cross-validation accuracy was (77.21 ± 1.28)% on PIMA Diabetes and (96.42 ± 1.44)% on Breast Cancer Wisconsin. Under a common 20-iteration budget, the proposed method achieved approximately 3.85× faster training and nearly 90% lower peak memory use than a reproduced multiparty encrypted first-order baseline, while attaining a lower final relative gradient and closer agreement with plaintext Newton. These results show that exploiting locally computed Newton statistics can preserve predictive utility while substantially reducing encrypted training cost, supporting privacy-preserving multi-institutional clinical prediction without pooling patient-level data.

*Keywords:* privacy-preserving clinical prediction, multi-institutional learning, multiparty homomorphic encryption, logistic regression, Newton-CG, CKKS

## 1. Introduction

Machine-learning models are increasingly investigated for clinical decision support using patient measurements and medical images. Kerezis et al. [1] developed a support-vector-machine framework to prioritize long-COVID patients for specialist consultations. Kıvrak et al. [2] combined deep image features with conventional machine-learning classifiers to classify pulmonary hypertension from routine chest radiographs. These studies illustrate the potential of predictive models to support patient assessment and clinical decision-making.

For binary outcomes, logistic regression, rooted in classical analysis [3], remains attractive because of its interpretability, probabilistic output, and low model complexity. Its biomedical relevance is illustrated by Yang et al. [4], who combined convolutional neural networks with a logistic-regression classifier to identify mycobacteria in digitized tissue slides. In their pipeline, logistic regression classifies whole slides using features derived from neural-network-generated probability maps, providing a concrete example of its use in computer-assisted diagnosis.

Extending predictive modeling across healthcare institutions can broaden the available evidence, but requires addressing both data heterogeneity and restrictions on data sharing. Seoni et al. [5] reviewed image harmonization in multi-center and multi-device studies, highlighting the importance of accounting for acquisition variability when analyzing medical data from multiple sources. However, privacy concerns, institutional policies, and data-governance requirements often restrict patient-level pooling. Alzubaidi et al. [6] proposed ATD Learning, a decentralized approach that transfers models to local data for collaborative training while retaining raw data at each node. Multi-institutional learning must therefore exploit distributed records without exposing them. Recent studies of personalized clinical prediction and collaborative medical imaging further illustrate this need [7, 8].

Federated learning retains raw data locally, but exchanged intermediate values may reveal sensitive information without additional protection. Homomorphic encryption (HE) protects such values by supporting computation directly on ciphertexts. Gentry's 2009 construction established the feasibility of fully homomorphic computation [9]. Subsequent encrypted logistic-

*Corresponding author
*Email addresses:* daixiaokang1993@gmail.com (Xiaokang Dai), weijie.miao@connect.polyu.hk (Weijie Miao), wuwenyuan@cigit.ac.cn (Wenyuan Wu), yongfeng@cigit.ac.cn (Yong Feng)

regression methods have used sigmoid approximation, ciphertext packing, accelerated gradients, and simplified Hessians to reduce computation [10, 11, 12], including applications to healthcare prediction [13]. Nevertheless, repeated ciphertext operations make iterative encrypted training expensive, particularly when many optimization rounds are required.

Multiparty frameworks including SPINDLE [14], PPMLR [15], MK-HELR [16], and Geva et al.'s oncological-analysis framework [17] support collaborative logistic regression without directly sharing participants' records. These systems predominantly use first-order optimization, whose simple arithmetic is compatible with HE but may require many encrypted iterations to approach the optimum. Reducing global optimization rounds is therefore complementary to reducing the cost of individual ciphertext operations. The issue is especially relevant when a common clinical model must be fitted across several independent institutions.

Second-order methods exploit curvature to improve optimization progress. Inexact Newton methods replace an exact Newton solve with an approximate linear-system solve [18]; conjugate gradient (CG) solves symmetric positive-definite systems iteratively [19]. Miao and Wu [20] combined HE and Newton-CG for horizontally partitioned logistic regression, avoiding explicit encrypted Hessian inversion and reducing outer iterations. Their protocol, however, uses an asymmetric two-user setting: one participant owns the complete encryption/decryption key pair and performs interactive ciphertext management. It does not directly provide a general multiparty architecture in which independent institutions jointly conduct second-order training without assigning complete decryption capability to one participant.

Our framework uses the known additive decomposition of Newton statistics under horizontal partitioning [21, 20] to separate local patient-dependent computation from global encrypted optimization. Each institution computes its score vector and Hessian locally and encrypts only these fixed-dimensional statistics. After aggregation, the dominant homomorphic workload depends primarily on model dimension rather than patient count. This avoids repeated ciphertext-domain processing of individual records, reducing encrypted operations and multiplicative depth. The decomposition itself is not a new mathematical result; its role here is to define the computational boundary of a general multi-institutional training architecture.

We implement this architecture for $K$ institutions using multiparty CKKS [22]. Local statistics are aggregated under a collective public key, and ciphertext-domain preconditioned CG (PCG) solves the global Newton system. Diagonal preconditioning improves the linear-system conditioning [23], while delayed division reduces serial encrypted inversion stages. Collective key generation and distributed decryption use existing OpenFHE primitives [24]. The contribution is their integration with local Newton-statistic computation and encrypted second-order optimization, rather than a new cryptographic primitive. Efficiency is evaluated against a reproduced multiparty encrypted first-order baseline.

The main contributions are as follows.

- **A multi-institutional formulation based on encrypted local Newton statistics.** We use the established additive structure of the score and Hessian to organize $K$-institution training. Patient-dependent operations remain local, and only encrypted fixed-dimensional statistics enter global optimization. The resulting ciphertext workload depends primarily on model dimension, permitting a shallower circuit and more compact CKKS parameters while reducing runtime, memory, and accumulated approximation error.

- **Multiparty encrypted Newton-CG training under a collective key.** The global Newton system is solved using ciphertext-domain PCG with diagonal preconditioning and delayed division. This integrates second-order optimization with a general multiparty architecture, extending the asymmetric two-user setting considered by prior encrypted Newton-CG work. Distributed decryption prevents any individual institution from independently decrypting collaborative ciphertexts.

- **Clinical evaluation of predictive fidelity and computational efficiency.** Across nine NHANES 2017–2018 tasks, the mean absolute deviations from pooled plaintext Newton are $8.52\times10^{-4}$, $1.86\times10^{-3}$, and $1.79\times10^{-3}$ for ROC-AUC, PR-AUC, and average precision, respectively. These quantify agreement on the training cohorts; held-out utility is assessed separately by five-fold cross-validation on PIMA Diabetes and Breast Cancer Wisconsin. On seven matched NHANES tasks, a same-platform comparison with Geva et al. [17] under a common 20-iteration budget reduces the corresponding predictive deviations by 95.4%, 89.7%, and 90.4%. The proposed method attains a final mean relative gradient approximately 43 times smaller, a 3.85× speedup in average per-iteration and iterative training time, and 89.2% lower peak memory consumption.

Section 2 reviews related approaches. Section 3 defines the model and notation where needed and presents the multiparty encrypted Newton-CG protocol. Section 4 evaluates scalability, predictive agreement, and efficiency; Section 5 discusses implications and limitations. Section 6 concludes the paper.

## 2. Related Work

We review encrypted logistic regression, multiparty biomedical learning, and encrypted second-order optimization to distinguish the proposed architecture from existing approaches.

### *2.1. Homomorphic Encryption-Based Logistic Regression*

HE-based training requires repeated nonlinear evaluations and model updates, making it more demanding than encrypted inference. Bos et al. [25] used leveled lattice-based HE for clinical prediction in 2014, evaluating an existing model on encrypted records rather than training it.

Five representative schemes illustrate the development of encrypted logistic-regression training. In 2016, Aono et al. [26] approximated the log-likelihood by a low-degree polynomial and aggregated data-dependent coefficients using additive HE,

with Paillier, LWE, and ring-LWE instantiations. Optimization was then performed on decrypted aggregate coefficients. In 2018, Miran Kim et al. [27] combined CKKS, least-squares sigmoid approximation, and packing for ciphertext-domain gradient descent. Andrey Kim et al. [10] added Nesterov acceleration and packing to reduce iteration cost. Bonte and Vercauteren [11] used the Fan–Vercauteren (FV) scheme, a simplified fixed Hessian, and low-degree sigmoid approximation to avoid repeated Hessian updates and inversion. Chen et al. [12] implemented FV-based gradient descent in SEAL, using bootstrapping and homomorphic scaling down to extend training beyond a fixed depth budget. These developments shifted the workload from encrypted coefficient aggregation to ciphertext-domain optimization while addressing nonlinear approximation, iteration cost, and available depth.

Recent methods continue to reduce encrypted training cost. Wei et al. [28] combined low-degree sigmoid approximation with momentum-based stochastic gradient descent. Naresh and Reddi [13] investigated CKKS-based heart-disease prediction and reported predictive performance close to plaintext logistic regression. In a complementary inference setting, Bai et al. [29] used packing and masked nonlinear evaluation for privacy-preserving medical-image CNN inference. These biomedical applications emphasize the need to adapt numerical workloads to HE. Most logistic-regression training schemes nevertheless retain first-order or simplified optimization, making the number of encrypted rounds an important remaining cost.

### 2.2. Privacy-Preserving Multiparty and Multi-Institutional Learning

Distributed logistic regression predates modern federated learning. GLORE, introduced by Wu et al. in 2012 [21], combines locally computed derivatives without transferring patient records, but exchanging statistics differs from cryptographically protecting them. Nardi et al. [30] developed a secure multiparty protocol that protects intermediate values and reveals the final result.

Biomedical federated approaches address complementary collaboration problems. Yang et al. [7] proposed hierarchical reweighting for personalized multi-center clinical prediction; Ciupek et al. [8] reviewed multi-institutional medical-imaging applications. Pham et al. [31] combined multi-key HE and federated 3D CNNs for physical-activity recognition. These studies concern clinical collaboration, heterogeneous data, and encrypted learning beyond the logistic-regression setting considered here.

SPINDLE [14] supports multiparty encrypted logistic regression through cooperative gradient optimization and evaluates PIMA Diabetes and Breast Cancer Wisconsin. SCALR [32] combines Lagrange coded computing and secret sharing to reduce online communication while retaining gradient descent. PPMLR [15] distributes additive-HE logistic-regression computation among data owners, a service provider, and a cloud server. MK-HELR [16] uses multi-key CKKS and Nesterov acceleration with independent participant keys and dynamic membership. These systems improve communication, coordination, or key management while predominantly using first-order optimization.

Geva et al. [17] developed a multiparty FHE framework for collaborative oncological analysis, using polynomial sigmoid approximation and Nesterov gradient descent for logistic regression. Its medical application and multiparty design make it the first-order baseline for our controlled evaluation. Together, these approaches motivate examining whether second-order optimization can improve progress within a fixed encrypted-iteration budget.

### 2.3. Second-Order Optimization for Privacy-Preserving Logistic Regression

Curvature can improve optimization progress, but encrypted second-order methods must construct and solve the Hessian system securely. Bonte and Vercauteren [11] simplified this task using a fixed Hessian, without dynamically solving the Newton system.

Miao and Wu [20] developed the most closely related Newton-CG approach. They compute local score and Hessian contributions for horizontally partitioned data, then solve the aggregated system through encrypted CG with diagonal preconditioning. This reduces outer iterations without explicit encrypted Hessian inversion. Their use of additive Newton statistics is therefore established prior work, not a new result claimed here.

Our distinction is the collaboration architecture: their asymmetric two-user protocol assigns the complete key pair to one participant, whereas our framework supports $K$ institutions using collective CKKS key generation and distributed decryption. Table 1 compares representative systems. The proposed framework combines multiparty encrypted learning and ciphertext-domain Newton-CG within one multi-institutional protocol.

Table 1: Comparison of representative privacy-preserving logistic-regression training frameworks.

| Work | Collaboration setting | Privacy mechanism | Optimizer | Protected optimization workload | Key / coordination structure | Medical evaluation |
|---|---|---|---|---|---|---|
| SPINDLE [14] | Multiple distributed providers | Multiparty HE | Cooperative gradient descent | Encrypted distributed gradient computation | Collective multiparty HE | Yes; PIMA and BCW |
| SCALR [32] | Multiple users | Lagrange coded computing + secret sharing | Gradient descent | Coded / secret-shared gradient computation | Decentralized collaboration | No; general ML benchmarks |
| PPMLR [15] | Multiple data owners, cloud-assisted | Additive HE | Gradient descent | Protected gradient computation | Service provider + cloud server | Yes; two UCI medical datasets |
| MK-HELR [16] | Multiple data contributors | Multi-key CKKS | Nesterov accelerated gradient | Encrypted iterative gradient training | Independent keys + dynamic membership | No; predictive-maintenance data |
| Geva et al. [17] | Multiple institutions | Multiparty FHE | Nesterov gradient descent | Encrypted sigmoid and gradient updates | Multiparty FHE | Yes; oncological data |
| Miao and Wu [20] | Two users, horizontal partition | CKKS | Newton-CG / PCG | Local Newton statistics + encrypted PCG | One party owns complete key pair | Yes; public medical benchmarks |
| **Our work** | **$K$ institutions, horizontal partition** | **Multiparty CKKS** | **Newton-CG / PCG** | **Local Newton statistics + encrypted aggregation and PCG** | **Collective key + distributed decryption** | **Yes; NHANES + PIMA + BCW** |

## 3. Multiparty Encrypted Newton-CG Logistic Regression

We define the distributed learning problem and information-flow boundary, then present key generation, local statistics, encrypted aggregation, PCG, and model updating. Patient-dependent computation remains local; collective optimization operates on encrypted statistics.

### 3.1. System Model, Problem Formulation, and Security Assumptions

We consider $K$ healthcare institutions $P_1, \ldots, P_K$ and a computation server $S$. Institution $P_i$ holds a horizontally partitioned dataset $\mathcal{D}_i = (X_i, y_i)$, where $X_i \in \mathbb{R}^{n_i \times d}$ contains $n_i$ patient records in a common feature space and $y_i \in \{0,1\}^{n_i}$ contains their labels. The dimension $d$ includes an intercept column. The conceptual global population contains $n = \sum_i n_i$ records, but features and labels remain at their originating institutions. The server coordinates aggregation and homomorphic evaluation using the collective public and evaluation keys; it holds no secret-key share. Let $x_{ij} \in \mathbb{R}^d$ and $y_{ij}$ denote the $j$-th feature vector and label at $P_i$. The positive-class probability is

$$p_{ij} = \sigma(x_{ij}^\top w) = \frac{1}{1 + \exp(-x_{ij}^\top w)}, \tag{1}$$

where $w \in \mathbb{R}^d$ is the model vector. The local summed binary cross-entropy is

$$J_i(w) = -\sum_{j=1}^{n_i} \left[ y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij}) \right]. \tag{2}$$

The shared training objective decomposes as

$$J(w) = \sum_{i=1}^{K} J_i(w), \tag{3}$$

and the local score vectors and damped Newton matrices satisfy

$$g = \sum_{i=1}^{K} g_i, \qquad H = \sum_{i=1}^{K} H_i. \tag{4}$$

Here $g_i = -\nabla J_i(w)$ is the negative loss gradient and $H_i = \nabla^2 J_i(w) + \rho_i I_d$, where $I_d$ is the identity matrix and $\rho_i > 0$ follows the local trace-based rule in Eq. (9). Thus $H = \nabla^2 J(w) + (\sum_i \rho_i) I_d$. Damping stabilizes the Newton system without adding a penalty to $J$ or changing the score. In exact arithmetic, aggregation recovers the pooled-data score and Hessian with the same total damping, rather than an undamped Newton step. At outer iteration $t$, Newton-CG approximately solves

$$H_t \Delta w_t = g_t, \qquad w_{t+1} = w_t + \Delta w_t. \tag{5}$$

For inner approximation $\Delta w_{t,k}$, define the residual $r_{t,k} = g_t - H_t \Delta w_{t,k}$. Within one Newton solve, the outer index is suppressed and this is written $r_k$, or $r$ when the inner index is unnecessary. The solver uses matrix–vector products, inner products, and vector updates instead of explicit Hessian inversion. The plaintext delayed-division PCG formulation is given in Section S7 of the Supplementary Information. The goal is to obtain

$$w^* = \arg\min_w \sum_{i=1}^{K} J_i(w), \tag{6}$$

without exposing local records or plaintext institution-specific statistics to other institutions or the server. Each institution evaluates patient-dependent nonlinear functions and forms its Newton statistics locally, encrypting them before transmission. Only the encrypted statistics and subsequent optimization intermediates enter cross-institutional computation. This boundary avoids ciphertext-domain processing of the full patient-level dataset while retaining one shared global objective.

Institutions and the server are assumed *semi-honest*: they follow the protocol but may seek information from available messages. Patient records, local Newton statistics, the Newton direction, and optimization intermediates are sensitive. Synchronized model iterates are legitimate shared training state. Feature-wise training-set minima and maxima are explicitly disclosed to establish common local scaling; their concealment is not claimed. The plaintext Newton direction and convergence scalars are transient institution-side intermediates, never plaintext outputs to the server. For optional stopping checks, the server receives only a continue/stop signal.

OpenFHE supplies the multiparty CKKS key-generation and distributed-decryption primitives [24]. Institutions retain separate secret-key shares, and all $K$ participants must contribute to decryption in the adopted configuration. Neither an individual institution nor the server can independently decrypt collaborative ciphertexts. These existing primitives are building blocks rather than cryptographic contributions of this work.

Interactive inversion and ciphertext refreshing use the randomized masking techniques of Miao and Wu [20], adapted to distributed decryption under the collective key. Values are masked before temporary decryption and recovered after re-encryption. The cited work provides the procedures and masking analysis; we introduce no new inversion or refresh primitive. The protocol assumes non-collusion: institutions do not share secret-key shares, masks, or private protocol states, and the server does not collude with an institution.

Consequently, local records remain private, statistics are encrypted before aggregation, and interactive values are masked. Institutions jointly recover the Newton direction for synchronized local updates. They also recover $g^\top g$ and $r^\top r$, the squared score and residual norms, and send only an optional stopping decision to the server. The server receives neither plaintext local statistics, the Newton direction, nor unmasked interactive values. This information-flow protection is considered under the stated semi-honest, non-colluding model. Malicious statistics, malformed ciphertexts, incorrect partial decryptions, poisoning, Byzantine behavior, and denial-of-service attacks are outside scope.

### 3.2. Multiparty Key Generation

The institutions first agree on common CKKS parameters and jointly generate a collective public key, retaining only their own secret-key shares. They also generate the evaluation keys for ciphertext multiplication and SIMD slot rotations. These keys establish a common context in which local statistics can be encrypted and combined across institutions; no participant possesses the complete secret key.

CKKS supports approximate arithmetic over packed real or complex values [22]. SIMD packing processes multiple entries within one ciphertext, and rotations support vector inner products

and matrix–vector products. Because encrypted multiplication and rescaling introduce approximation error and consume modulus levels, the subsequent solver limits multiplicative depth and uses ciphertext refreshing. This numerical setting motivates both the layouts in Section 3.4 and the PCG formulation in Section 3.5.

Figure 1 illustrates initialization. Key generation occurs once before iterative training; subsequent Newton steps reuse the context, public key, and evaluation keys. Collective decryption requires the participating institutions' shares as specified in Section 3.1. OpenFHE provides these procedures, while our contribution is the organization of encrypted second-order training on top of them.

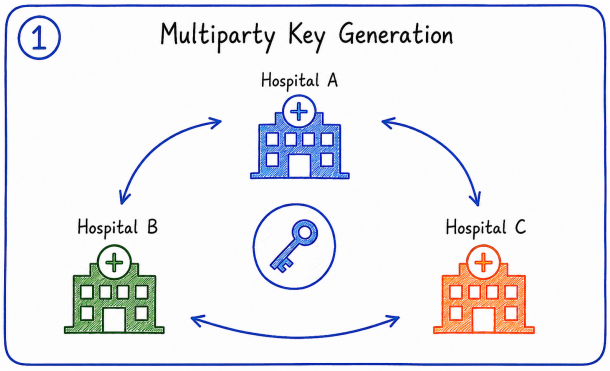



Figure 1: Multiparty key-generation workflow.

### 3.3. Local Computation of Newton Statistics

At each Newton iteration, institutions use the synchronized model $w$ to compute $p_i = \sigma(X_i w) \in \mathbb{R}^{n_i}$ locally, applying the sigmoid element-wise. They then form the score and unshifted Hessian:

$$H_i^{\text{raw}} = X_i^\top W_i X_i, \quad g_i = X_i^\top (y_i - p_i) \tag{7}$$

Here $W_i = \text{Diag}(p_i \odot (\mathbf{1}_{n_i} - p_i))$, $\mathbf{1}_m$ is the length-$m$ all-ones vector, Diag constructs a diagonal matrix, and $\odot$ denotes element-wise multiplication. The score is the negative gradient of $J_i$, and the raw Hessian is positive semidefinite. Before encryption, each institution applies a positive diagonal shift:

$$H_i = H_i^{\text{raw}} + \rho_i I_d, \tag{8}$$

The coefficient follows the local mean diagonal curvature:

$$\rho_i = \frac{\tau}{K} \frac{\text{Tr}(H_i^{\text{raw}})}{d}, \qquad \tau = 10^{-3}, \tag{9}$$

where Tr denotes trace. The relative parameter $\tau$ is fixed, while $\rho_i$ is recomputed from the current Hessian at each outer iteration. This rule scales damping to local curvature using plaintext computation. For a nonempty dataset and finite logistic scores, the intercept and positive logistic weights imply $\text{Tr}(H_i^{\text{raw}}) > 0$, hence $\rho_i > 0$ in exact arithmetic. Institutions encrypt the damped matrices, whose aggregate is

$$H = \sum_{i=1}^{K} H_i = \sum_{i=1}^{K} H_i^{\text{raw}} + \left( \sum_{i=1}^{K} \rho_i \right) I_d. \tag{10}$$

with total shift

$$\rho_{\text{tot}} = \sum_{i=1}^{K} \rho_i = \frac{\tau}{Kd} \text{Tr}\left( \sum_{i=1}^{K} H_i^{\text{raw}} \right). \tag{11}$$

For any nonzero $v$, $v^\top H v \geq (\sum_i \rho_i)\|v\|_2^2 > 0$ in exact arithmetic. The Newton system is therefore positive definite even for rank-deficient features, without global singular value decomposition (SVD) or patient-level pooling. This is damping of the Newton step, rather than L2-penalized logistic regression: $g_i$ and $J(w)$ remain unchanged. Institution and iteration dependence of the coefficients is retained, although outer indices are suppressed.

Figure 2 illustrates this stage. Local sigmoid evaluation avoids the polynomial approximation and multiplicative depth required for patient-level CKKS evaluation. Matrix construction also remains plaintext, so sample-dependent work is completed before encryption. The additive structure in Eq. (4) makes $g_i$ and $H_i$ sufficient for constructing the global Newton system; no underlying records need to be aggregated. These statistics are not transmitted in plaintext: Section 3.4 encodes and encrypts them under the collective key. Thus the encrypted domain is reserved for operations requiring cross-institutional collaboration.

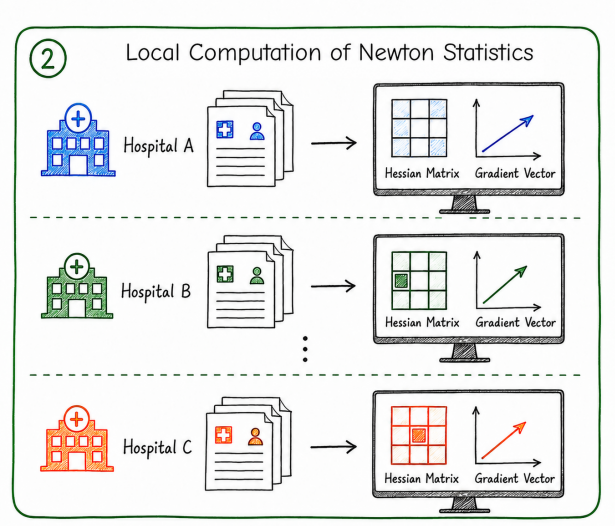



Figure 2: Local computation of Newton statistics.

### 3.4. Encoding, Encryption, and Homomorphic Aggregation

Each institution encodes its local statistics, encrypts them under the collective key, and submits ciphertexts for aggregation (Fig. 3). We write ciphertexts as $[\![\cdot]\!]$, with subscripts std and HS identifying two complementary slot layouts.

The standard layout packs $v = (v_0, \ldots, v_{d-1})$ into consecutive CKKS slots. Its ciphertext $[\![v]\!]_{\text{std}}$ supports element-wise multiplication, inner products, and vector updates. Hessian–vector products use a layout based on the Halevi–Shoup diagonal method [33]:

$$\mathcal{V}(v) = v^{(0)} \| v^{(1)} \| \cdots \| v^{(d-1)}, \tag{12}$$

Here $v_j^{(s)} = v_{(j+s) \bmod d}$ for $j, s = 0, \ldots, d-1$, and $\|$ denotes concatenation. Encoding and encryption give $[\![v]\!]_{\text{HS}}$. The

two layouts represent the same vector but match different homomorphic operations. For $H_i$, define the cyclic diagonal $(D_s(H_i))_j = (H_i)_{j,(j+s) \bmod d}$ and pack

$$\mathcal{D}(H_i) = D_0(H_i) \| D_1(H_i) \| \cdots \| D_{d-1}(H_i), \tag{13}$$

The resulting $[\![H_i]\!]_{\mathrm{HS}}$ aligns with the HS vector representation, allowing matrix–vector multiplication through SIMD products, rotations, and additions. Section 3.5 uses this operation in the global PCG solve. The local score is encoded in both layouts as $[\![g_i]\!]_{\mathrm{std}}$ and $[\![g_i]\!]_{\mathrm{HS}}$.

Given a common initial Newton approximation $\Delta w_0$, institution $P_i$ forms $r_i = g_i - H_i \Delta w_0$ locally and encrypts it in both layouts. No homomorphic matrix–vector product is needed to initialize this residual because its inputs are available locally. Standard packing supports inner products and updates, while HS packing supports Hessian–vector products involving the residual or derived search direction. Maintaining both forms avoids repeated slot rearrangement during iterative computation.

Each institution also constructs the damping-adjusted diagonal $\widetilde{\delta}_i = \mathrm{diag}(H_i) = \mathrm{diag}(H_i^{\mathrm{raw}}) + \rho_i \mathbf{1}_d$, where diag extracts diagonal entries. The same damped $H_i$ is used in the residual, preconditioner, and Hessian–vector products: the shift is applied to the entire Newton matrix before encryption, rather than only to the preconditioner. The diagonal is encrypted in standard packing. Patient records and plaintext statistics remain local throughout; the common context and public key allow ciphertexts with matching layouts to be directly combined.

The server aggregates statistics independently by layout using CKKS additions:

$$\begin{aligned} [\![H]\!]_{\mathrm{HS}} &= \sum_{i=1}^{K} [\![H_i]\!]_{\mathrm{HS}}, & [\![g]\!]_{\mathrm{std}} &= \sum_{i=1}^{K} [\![g_i]\!]_{\mathrm{std}}, \\ [\![r]\!]_{\mathrm{std}} &= \sum_{i=1}^{K} [\![r_i]\!]_{\mathrm{std}}, & [\![r]\!]_{\mathrm{HS}} &= \sum_{i=1}^{K} [\![r_i]\!]_{\mathrm{HS}}, \\ [\![\widetilde{\delta}]\!]_{\mathrm{std}} &= \sum_{i=1}^{K} [\![\widetilde{\delta}_i]\!]_{\mathrm{std}}. \end{aligned} \tag{14}$$

Because institutions share $\Delta w_0$, the initial aggregate satisfies $r = \sum_i r_i = g - H\Delta w_0$. It equals the score when $\Delta w_0 = 0$; later Newton steps may instead use a nonzero warm start. Aggregation requires no ciphertext multiplication and adds no multiplicative depth. The aggregate diagonal $\widetilde{\delta} = \sum_i \widetilde{\delta}_i$ corresponds to the same damped global matrix, whose positive local shifts remove the need for an encrypted rank test. The diagonal preconditioner is

$$M^{-1} = \mathrm{Diag}\left(\widetilde{\delta}_0^{-1}, \ldots, \widetilde{\delta}_{d-1}^{-1}\right), \tag{15}$$

Its reciprocal is computed after aggregation using the secure inversion technique of Miao and Wu [20], since CKKS does not directly provide an efficient reciprocal operation. The protocol returns $[\![M^{-1}]\!]_{\mathrm{std}}$ and $[\![M^{-1}]\!]_{\mathrm{HS}}$ without revealing the diagonal to the server. These symbols denote the reciprocal-diagonal vector $m = (\widetilde{\delta}_0^{-1}, \ldots, \widetilde{\delta}_{d-1}^{-1})$ packed in the two layouts, rather than a dense matrix. Preconditioning is therefore element-wise multiplication by $m$. The encrypted Hessian, residuals, and preconditioner now supply the inputs for ciphertext PCG.

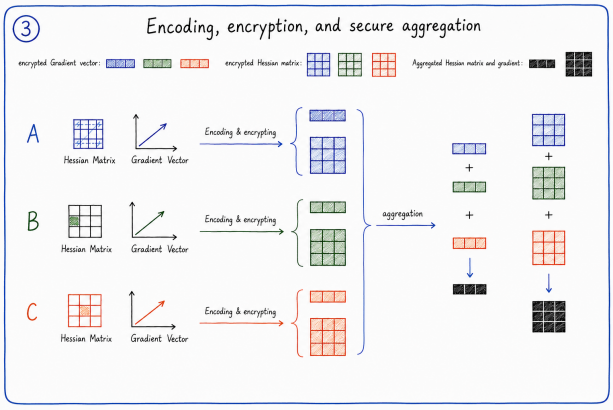



Figure 3: Encoding, encryption, and homomorphic-aggregation workflow.

### 3.5. *Ciphertext PCG Training with Delayed Division*

The server solves the Newton system using $[\![H]\!]_{\mathrm{HS}}$, together with the initial residual and diagonal preconditioner in both standard and HS layouts. Here $r_0$ is the initial residual aggregated above. Let $L$ be the maximum inner-iteration count and $\tau_r$ an optional squared-residual threshold. Section S7 of the Supplementary Information relates delayed division to conventional PCG; this subsection presents its encrypted realization (Fig. 4).

For inner iteration $k$, $p_k$ is the search direction, $z_k = M^{-1} r_k$ the preconditioned residual, and $\mu_k = r_k^\top z_k$. These search-direction vectors are distinct from prediction probabilities $p_{ij}$ and $p_i$. Initially $q_0 = Hp_0$, $\eta_0 = p_0^\top q_0$, and $\alpha_0 = \mu_0/\eta_0$. For $k \geq 1$, delayed division forms $f_k = \mu_{k-1} z_k + \mu_k p_{k-1}$, followed by $q_k = Hf_k$, $\eta_k = f_k^\top q_k$, and $\widetilde{\alpha}_k = \mu_k/\eta_k$. The monitored scalar is $s_{r,k+1} = r_{k+1}^\top r_{k+1}$; $q_k$ always denotes a matrix–vector product, while $w$ denotes model parameters.

For ciphertexts, $\langle \cdot, \cdot \rangle$ denotes homomorphic inner-product evaluation, with scalar outputs replicated across active slots for scalar–vector multiplication. HMult denotes encrypted Hessian–vector multiplication; SecInv and Refresh are the masked interactive inversion and refresh techniques adopted from Miao and Wu [20]. Their detailed masking procedures remain in the cited work. Algorithm 1 gives the complete encrypted procedure.

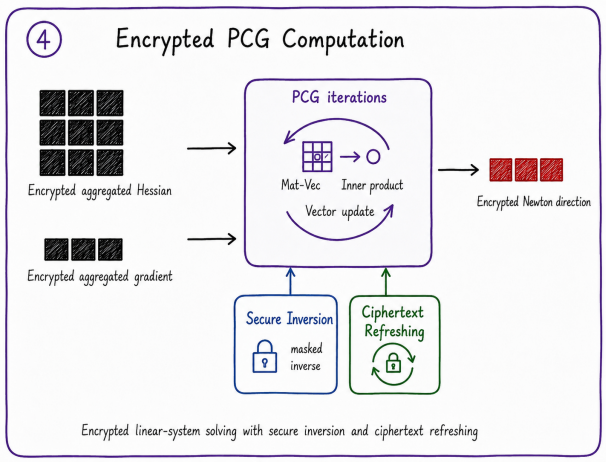



Figure 4: Encrypted PCG-computation workflow.

**Algorithm 1** Ciphertext PCG with Delayed Division

**Input:** $[\![H]\!]_{\mathrm{HS}}$, $[\![r_0]\!]_{\mathrm{std}}$, $[\![r_0]\!]_{\mathrm{HS}}$, $[\![M^{-1}]\!]_{\mathrm{std}}$, $[\![M^{-1}]\!]_{\mathrm{HS}}$, $\Delta w_0$, $L$, and optional residual threshold $\tau_r$.
**Output:** $[\![\Delta w_L]\!]_{\mathrm{std}}$.

Encrypt $\Delta w_0$ as $[\![\Delta w_0]\!]_{\mathrm{std}}$.
**for** $k = 0, \ldots, L-1$ **do**
  $[\![z_k]\!]_{\mathrm{std}} \leftarrow [\![M^{-1}]\!]_{\mathrm{std}} \odot [\![r_k]\!]_{\mathrm{std}}$
  $[\![z_k]\!]_{\mathrm{HS}} \leftarrow [\![M^{-1}]\!]_{\mathrm{HS}} \odot [\![r_k]\!]_{\mathrm{HS}}$
  $[\![\mu_k]\!]_{\mathrm{std}} \leftarrow \langle [\![r_k]\!]_{\mathrm{std}}, [\![z_k]\!]_{\mathrm{std}} \rangle$
  **if** $k = 0$ **then**
    $[\![p_0]\!]_{\mathrm{std}} \leftarrow [\![z_0]\!]_{\mathrm{std}}, \quad [\![p_0]\!]_{\mathrm{HS}} \leftarrow [\![z_0]\!]_{\mathrm{HS}}$
    $[\![q_0]\!]_{\mathrm{std}} \leftarrow \mathrm{HMult}([\![H]\!]_{\mathrm{HS}}, [\![p_0]\!]_{\mathrm{HS}})$
    $[\![\eta_0]\!]_{\mathrm{std}} \leftarrow \langle [\![p_0]\!]_{\mathrm{std}}, [\![q_0]\!]_{\mathrm{std}} \rangle$
    $[\![\eta_0^{-1}]\!]_{\mathrm{std}} \leftarrow \mathrm{SecInv}([\![\eta_0]\!]_{\mathrm{std}})$
    $[\![\alpha_0]\!]_{\mathrm{std}} \leftarrow [\![\mu_0]\!]_{\mathrm{std}} \odot [\![\eta_0^{-1}]\!]_{\mathrm{std}}$
    $[\![\Delta w_1]\!]_{\mathrm{std}} \leftarrow [\![\Delta w_0]\!]_{\mathrm{std}} + [\![\alpha_0]\!]_{\mathrm{std}} \odot [\![p_0]\!]_{\mathrm{std}}$
    $[\![r_1]\!]_{\mathrm{std}} \leftarrow [\![r_0]\!]_{\mathrm{std}} - [\![\alpha_0]\!]_{\mathrm{std}} \odot [\![q_0]\!]_{\mathrm{std}}$
  **else**
    $[\![f_k]\!]_{\mathrm{std}} \leftarrow [\![\mu_{k-1}]\!]_{\mathrm{std}} \odot [\![z_k]\!]_{\mathrm{std}} + [\![\mu_k]\!]_{\mathrm{std}} \odot [\![p_{k-1}]\!]_{\mathrm{std}}$
    $[\![f_k]\!]_{\mathrm{HS}} \leftarrow [\![\mu_{k-1}]\!]_{\mathrm{std}} \odot [\![z_k]\!]_{\mathrm{HS}} + [\![\mu_k]\!]_{\mathrm{std}} \odot [\![p_{k-1}]\!]_{\mathrm{HS}}$
    $[\![q_k]\!]_{\mathrm{std}} \leftarrow \mathrm{HMult}([\![H]\!]_{\mathrm{HS}}, [\![f_k]\!]_{\mathrm{HS}})$
    $[\![\eta_k]\!]_{\mathrm{std}} \leftarrow \langle [\![f_k]\!]_{\mathrm{std}}, [\![q_k]\!]_{\mathrm{std}} \rangle$
    $\left([\![\eta_k^{-1}]\!]_{\mathrm{std}}, [\![\mu_{k-1}^{-1}]\!]_{\mathrm{std}}\right) \leftarrow \mathrm{SecInv}([\![\eta_k]\!]_{\mathrm{std}}, [\![\mu_{k-1}]\!]_{\mathrm{std}})$
    $[\![\widetilde{\alpha}_k]\!]_{\mathrm{std}} \leftarrow [\![\mu_k]\!]_{\mathrm{std}} \odot [\![\eta_k^{-1}]\!]_{\mathrm{std}}$
    $[\![p_k]\!]_{\mathrm{std}} \leftarrow [\![f_k]\!]_{\mathrm{std}} \odot [\![\mu_{k-1}^{-1}]\!]_{\mathrm{std}}$
    $[\![p_k]\!]_{\mathrm{HS}} \leftarrow [\![f_k]\!]_{\mathrm{HS}} \odot [\![\mu_{k-1}^{-1}]\!]_{\mathrm{std}}$
    $[\![\Delta w_{k+1}]\!]_{\mathrm{std}} \leftarrow [\![\Delta w_k]\!]_{\mathrm{std}} + [\![\mu_{k-1}]\!]_{\mathrm{std}} \odot [\![\widetilde{\alpha}_k]\!]_{\mathrm{std}} \odot [\![f_k]\!]_{\mathrm{std}}$
    $[\![r_{k+1}]\!]_{\mathrm{std}} \leftarrow [\![r_k]\!]_{\mathrm{std}} - [\![\mu_{k-1}]\!]_{\mathrm{std}} \odot [\![\widetilde{\alpha}_k]\!]_{\mathrm{std}} \odot [\![q_k]\!]_{\mathrm{std}}$
  **end if**
  $[\![s_{r,k+1}]\!]_{\mathrm{std}} \leftarrow \langle [\![r_{k+1}]\!]_{\mathrm{std}}, [\![r_{k+1}]\!]_{\mathrm{std}} \rangle$
  Participating institutions recover $s_{r,k+1}$ by masked distributed decryption
  **if** early stopping is enabled and institutions signal $s_{r,k+1} < \tau_r$ **then**
    **return** $[\![\Delta w_{k+1}]\!]_{\mathrm{std}}$
  **end if**
  $([\![r_{k+1}]\!]_{\mathrm{std}}, [\![p_k]\!]_{\mathrm{std}}) \leftarrow \mathrm{Refresh}([\![r_{k+1}]\!]_{\mathrm{std}}, [\![p_k]\!]_{\mathrm{std}})$
  Regenerate $[\![r_{k+1}]\!]_{\mathrm{HS}}$ and $[\![p_k]\!]_{\mathrm{HS}}$.
**end for**
**return** $[\![\Delta w_L]\!]_{\mathrm{std}}$

After each inner update, the server computes the encrypted squared residual norm and randomizes its ciphertext before distributed decryption. Institutions recover only the scalar, not the residual vector. If early stopping is enabled, they compare it with $\tau_r$ and return only a continue/stop decision. Under a fixed iteration budget, it is recorded solely for convergence monitoring and does not change the execution schedule.

Delayed division absorbs the search-direction denominator into $f_k$ before the expensive Hessian–vector product. The reciprocals of $\eta_k = f_k^\top H f_k$ and $\mu_{k-1}$ then become available together and can be evaluated within one interactive inversion stage. Two serial divisions are thus reorganized into one interaction containing two independent reciprocals. After these reciprocals are obtained, the server recovers the encrypted search direction and updates the Newton increment and residual without exposing their plaintext values.

Repeated CKKS multiplications consume modulus levels and accumulate approximation error. Residual and search-direction ciphertexts are therefore refreshed after each PCG iteration through masked distributed decryption under the collective key. Their standard representations are regenerated in HS form for the next Hessian–vector product. Throughout the inner solve, the global Hessian, residual vectors, search directions, and Newton increment remain encrypted; only the designated convergence scalar is recovered institution-side.

### *3.6. Distributed Decryption, Convergence Check, and Model Update*

After PCG, the server holds $[\![\Delta w^{(t)}]\!]_{\mathrm{std}}$. Superscripts $(t)$ identify outer iterates, with $w^{(t)} = w_t$, $g^{(t)} = g_t$, and $\Delta w^{(t)} = \Delta w_t$ as defined in Section 3.1. Institutions contribute partial decryption shares and combine them institution-side to recover the same Newton direction. The server has no secret-key share and never receives this plaintext intermediate. Figure 5 illustrates recovery, convergence checking, and updating.

The server evaluates the encrypted squared score norm, $[\![s_g^{(t)}]\!]_{\mathrm{std}} = \langle [\![g^{(t)}]\!]_{\mathrm{std}}, [\![g^{(t)}]\!]_{\mathrm{std}} \rangle$. After masking and distributed decryption, institutions recover $s_g^{(t)} = (g^{(t)})^\top g^{(t)}$ and return only the optional stopping decision. If early stopping is enabled and $s_g^{(t)} < \theta$, with configurable threshold $\theta$, the current model $w^{(t)}$ is retained. Otherwise, all institutions update $w^{(t+1)} = w^{(t)} + \Delta w^{(t)}$ and use the synchronized model to recompute local statistics. Neither the score nor residual vector is decrypted for these checks. For convergence plots in Section 4.4, we report

$$\rho_g^{(t)} = \sqrt{\frac{s_g^{(t)}}{s_g^{(0)}}} = \frac{\|g^{(t)}\|_2}{\|g^{(0)}\|_2}. \tag{16}$$

This quantity is defined for $s_g^{(0)} > 0$ and is diagnostic only. The optional stopping rule uses the absolute squared score norm, rather than $\rho_g^{(t)}$; fixed-budget experiments use neither quantity to terminate training.

Key generation is performed once, whereas local statistics, encrypted aggregation, PCG, and distributed model updating repeat until an enabled stopping criterion or the outer-iteration limit is reached. The latest model becomes the trained logistic-regression model. Figure 6 combines the stages in Sections 3.2–3.6: if another Newton step is required, the protocol returns to local statistic construction with the updated shared parameters. Thus each outer round constructs one global Newton system and one collective update rather than separately fitting and averaging institutional models.

### *3.7. Computational and Communication Complexity Analysis*

Let $T$ and $L$ denote the outer Newton and inner PCG iteration counts, and $B_H$ and $B_v$ the CKKS block counts for a Hessian and a $d$-dimensional vector. At each outer iteration, local matrix construction dominates the plaintext work at institution $P_i$, with cost $O(n_i d^2)$. Institutions can execute this stage independently and in parallel. Patient-level data are reduced to fixed-dimensional Newton statistics before entering the encrypted domain, so the subsequent homomorphic workload does not directly scale with total record count.

Each PCG iteration requires a constant number of Hessian–vector products, inner products, element-wise multiplications, and vector updates. Under the packing in Section 3.4, the dominant encrypted cost per Newton step is

$$O(L(B_H + B_v)\,[C_{\mathrm{mul}} + \log d\, C_{\mathrm{rot}}]) \tag{17}$$

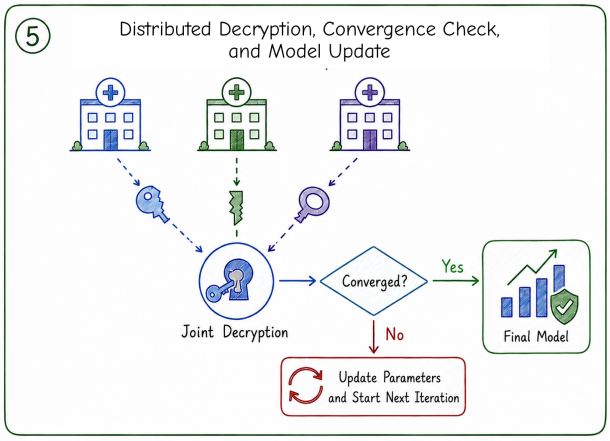



Figure 5: Distributed decryption, convergence checking, and model-update workflow.

where $C_{\text{mul}}$ and $C_{\text{rot}}$ are the costs of CKKS ciphertext multiplication and rotation. The model dimension and packing determine $B_H$ and $B_v$. If the Hessian exceeds the available slot capacity, additional blocks increase runtime and memory; encrypted cost is therefore not independent of feature dimension.

Before PCG, institutions upload encrypted Hessian and vector statistics for server-side addition. Interactive inversion, refreshing, and distributed recovery require contributions from all institutions, giving approximately linear communication growth with $K$. Across training, the dominant communication is

$$O(TK(B_H + LB_v)|ct|)\,, \tag{18}$$

where $|ct|$ is the ciphertext size at the corresponding modulus level. After aggregation, however, the global PCG solve runs once per Newton step rather than once per institution. Increasing $K$ adds submissions and distributed-decryption work without duplicating the expensive encrypted linear solve.

This separation distinguishes three scaling factors. Larger cohorts primarily increase parallelizable local plaintext work. More institutions increase encrypted-statistic transmission, aggregation, and distributed interaction. More features enlarge the Hessian quadratically and can change the ciphertext block count, directly affecting the encrypted solver. Consequently, moderate-dimensional clinical models over large distributed cohorts are a natural application of this architecture, while high-dimensional models require substantially more ciphertext storage and computation. The next section evaluates these effects empirically alongside predictive fidelity and optimization cost.

## 4. Experimental Evaluation

We evaluate scalability, predictive agreement, convergence, and computational efficiency. After specifying the common environment and protocol, we examine institution count, sample size, and feature dimension. Nine NHANES tasks assess agreement with pooled plaintext Newton, while PIMA and BCW assess held-out utility. A same-platform comparison with Geva et al. [17] evaluates predictive deviations, gradient reduction, runtime, and memory under a common iteration budget. These experiments test whether local patient-dependent computation and encrypted global optimization preserve predictive behavior while reducing computational cost.

### 4.1. Experimental Setup

Experiments ran on a server with two Intel Xeon Gold 6530 processors (64 physical cores), 1 TB RAM, and 128 OpenMP threads, using Ubuntu 24.04.3 LTS and g++ 13.3.0. The proposed C++ implementation uses OpenFHE 1.2.3 and multiparty CKKS, with default ring dimension $N = 32{,}768$ and total ciphertext modulus $\log_2 Q \approx 414$ bits, where $Q$ denotes the modulus.

Both methods target at least 128-bit classical security. The HomomorphicEncryption.org guidelines for Gaussian error $\sigma = 3.19$ and ternary secrets recommend maximum $\log_2 Q$ values of approximately 868 for $N = 32{,}768$ and 1747 for $N = 65{,}536$ at this security category [34]. The proposed and reproduced Geva configurations use 414 and 886 bits, respectively, below these limits. The smaller proposed parameters therefore reflect a shallower encrypted circuit, rather than a lower target security level.

All proposed-method runs use Eq. (9) with $\tau = 10^{-3}$, recomputing each local shift at every Newton step. This parameter controls damping, not stopping. The default uses three institutions; scalability runs vary $K$ from 3 to 9 with the same global population, $\tau$, and other configured settings. The total shift remains $K$-dependent according to Eq. (11).

Every proposed-method experiment executes 20 outer Newton iterations and 12 encrypted PCG iterations per step. The optional stopping rules in Sections 3.5 and 3.6 are disabled. Squared residual norms are recorded after inner iterations; squared score norms and relative gradients are recorded for outer iterations. Their institution-side recovery is used only for monitoring, without decrypting score or residual vectors or changing the fixed schedule. This makes proposed-method runtime and convergence measurements comparable across runs.

The primary evaluation comprises nine binary tasks from NHANES 2017–2018 [35]. Common demographic, anthropometric, blood-pressure, physical-activity, and questionnaire predictors, together with task-specific missingness indicators, yield 65–67 features. Outcome-defining variables and obvious direct disease proxies are excluded to avoid target leakage. Supplementary Information documents outcome definitions and the final feature dictionary.

Categorical responses are encoded locally as binary or one-hot indicators under a common schema. For continuous feature $j$, institution $P_i$ discloses training-set extrema $a_{ij}$ and $b_{ij}$, giving common bounds $a_j = \min_i a_{ij}$ and $b_j = \max_i b_{ij}$. Institutions apply the same local transformation:

$$x'_j = \frac{x_j - a_j}{b_j - a_j} - \frac{1}{2}, \qquad b_j > a_j. \tag{19}$$

Binary and one-hot indicators retain their encoding. Only extrema are disclosed; records and labels are not exchanged. This is range-based normalization, not global z-score standardization, and the disclosed extrema are part of the information boundary

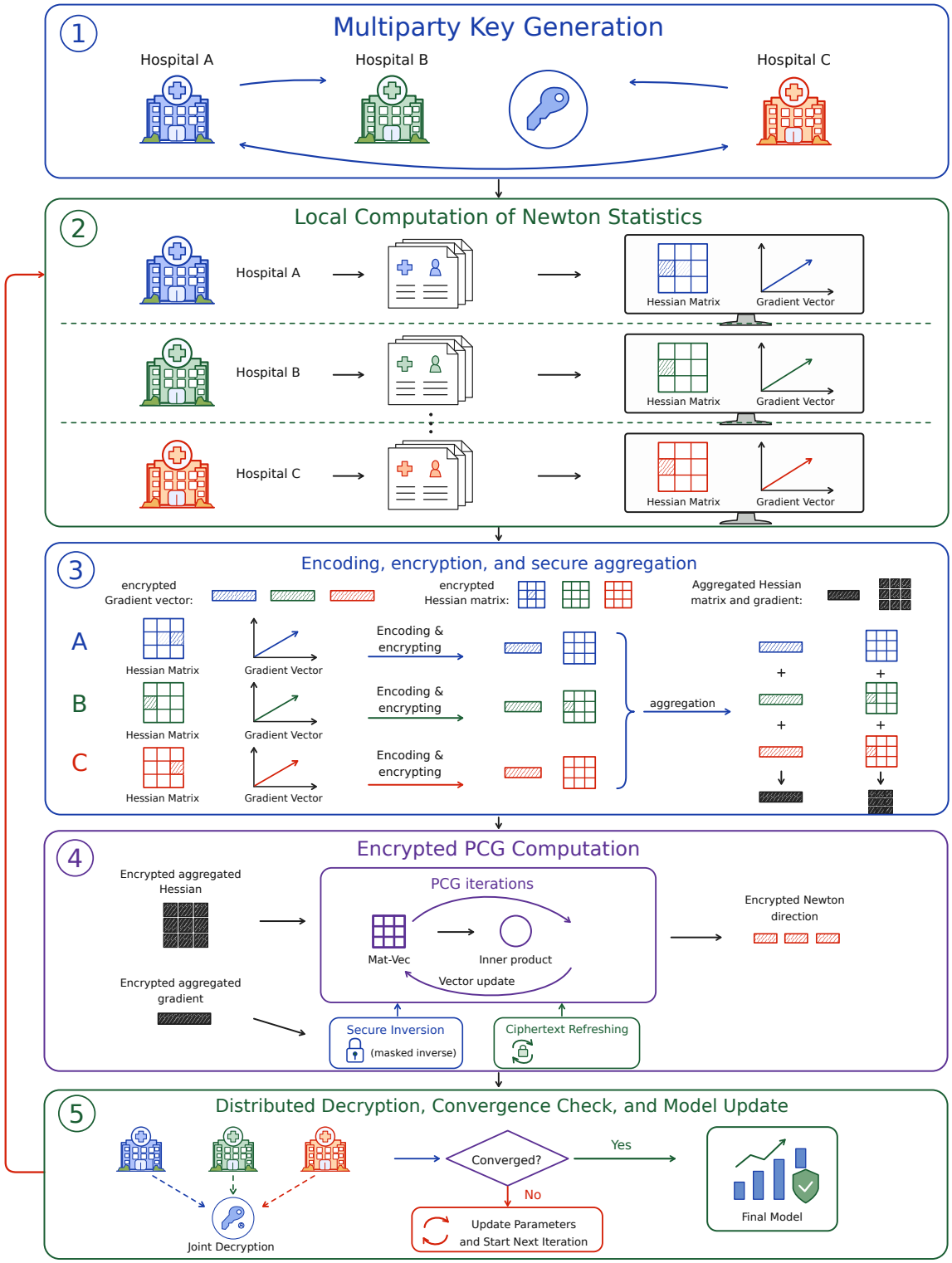



Figure 6: Overall workflow of the proposed privacy-preserving multi-institutional logistic regression framework. Multiparty key generation is performed once during system initialization. Each subsequent Newton iteration consists of local computation of Newton statistics, encoding and homomorphic aggregation, ciphertext PCG computation, and distributed decryption followed by convergence checking and model updating. If the convergence criterion is not satisfied, the updated model is returned to the participating institutions and Steps 2–5 are repeated until an enabled stopping criterion is satisfied or the maximum outer-iteration budget is reached.

in Section 3.1. No global SVD rank correction is performed: positive diagonal damping is applied to local Hessians before encryption.

For held-out evaluations, bounds are fitted on distributed training records and applied unchanged to test records. NHANES evaluates the training cohorts and measures predictive agreement

rather than out-of-sample clinical performance. PIMA and BCW use the five-fold protocol below. Labels are 1 for the positive outcome and 0 otherwise. Encrypted and plaintext methods use identical feature representations and evaluation records, so differences reflect their end-to-end training procedures under the common representation.

Logical institutions run on one physical server with a common CKKS context. This controls hardware and network variability when evaluating multiparty overhead; wall-clock time primarily measures computation and protocol processing rather than wide-area latency. Key generation and one-time initialization are excluded from iterative training time and reported separately in scalability results.

The per-iteration timer starts before local statistic construction and ends after distributed model updating and post-update metric evaluation. It includes local computation, encryption, aggregation, PCG, interactive inversion/refresh, recovery of the model increment, updating, and diagnostic evaluation and logging. Data loading and one-time key initialization are excluded. Both methods use the same process-level peak-memory monitor, which includes cryptographic context, evaluation keys, and iterative working memory whenever they contribute to the process maximum.

### *4.2. Multi-Institutional and Data-Scale Scalability*

We examine three scaling factors with different computational effects: institution count adds multiparty operations, sample count increases local plaintext statistic construction, and feature dimension determines the encrypted Newton-system size and ciphertext block count.

**Number of institutions.** The same global population is repartitioned among 3–9 logical institutions, keeping CKKS parameters, iteration budget, $\tau$, and the learning objective fixed. The total diagonal shift varies with $K$ under the stated trace-based rule, so this experiment measures scaling with that damping configuration.

Figure 7 shows that mean per-Newton runtime increases from 12.63 s at three institutions to 15.35 s at nine, a 21.5% increase despite tripling the participant count. With a fixed 20-step budget, iterative training time increases from 252.58 s to 306.92 s. The dominant encrypted PCG solve runs once on the aggregate system, while additional institutions contribute statistic submissions, additions, and distributed-decryption operations.

Peak memory increases from 1.365 GB to 1.404 GB (2.9%). Key generation and initialization increase more strongly, from 2.38 s to 6.26 s, but occur once and remain small relative to iterative training. These results indicate moderate runtime growth and limited additional memory over the evaluated institution counts; they do not include geographically distributed network overhead.

**Number of training samples.** Sample-dependent scores and Hessians are constructed locally before encryption. To examine overall runtime across a broad sample range, we use benchmarks adopted by Miao and Wu [20], primarily from public UCI datasets: LBW, PCS, UIS, Splice, iDASH, and NHANES contain 189–15,649 samples. CHF from our clinical evaluation is also included. This approximately 83-fold range probes whether record count visibly dominates the measured per-Newton runtime.

Figure 8 normalizes per-iteration runtime by the mean across evaluated datasets. Values remain approximately 0.97–1.03, without a systematic increase with sample count. Because these are different datasets rather than controlled subsamples with a fixed feature space, the result corroborates the complexity analysis but does not strictly isolate the effect of sample size.

The local cost still grows with $n_i$: institutions must evaluate predictions and construct $g_i$ and $H_i$ from their records. After their aggregation in Eq. (4), however, the encrypted solver uses fixed-dimensional statistics. In the evaluated range, local plaintext work remains small relative to cryptographic computation, explaining why overall iteration time shows little record-count sensitivity. Sample-size independence is therefore a property of the dominant encrypted workload, not of every component of the training pipeline.

**Feature dimension.** Our ring dimension $N = 32{,}768$ supplies $N/2 = 16{,}384$ complex slots. The Hessian layout requires approximately $d^2$ packed positions, so $d \leq 128$ fits within the same ciphertext-block configuration. Increasing dimension in this regime adds no blocks, and the dominant encrypted workload changes only slightly. This statement concerns the packing regime rather than an unrestricted claim of dimension-independent runtime.

Beyond the slot boundary, additional Hessian blocks cause stepwise increases in encrypted matrix–vector work, rotations, storage, and communication. Feature scaling is therefore packing-dependent rather than a smooth function of $d$. Miao and Wu [20] likewise observed increased computational and communication costs on higher-dimensional datasets with a smaller-capacity packing configuration. Thus the clinical setting of many records and a moderate feature count is particularly favorable, while larger feature spaces directly increase encrypted cost.

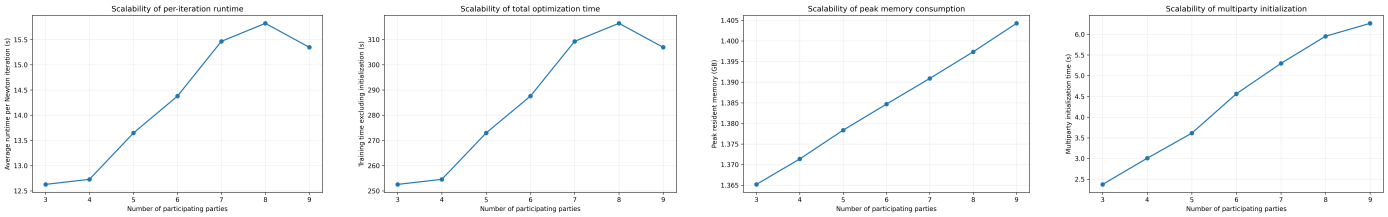
Together, the experiments support the architectural distinction: additional institutions mainly add multiparty overhead, larger cohorts add local plaintext work, and higher dimension can expand the ciphertext representation. Within the evaluated cohort sizes and fixed packing regime, the first two effects remain small relative to the global encrypted PCG solve.

### *4.3. Predictive Utility on Public Medical Datasets*

We assess training-cohort agreement with pooled plaintext Newton on nine NHANES tasks and held-out utility on PIMA Diabetes and Breast Cancer Wisconsin (BCW), benchmarks also used by SPINDLE. These complementary evaluations examine encrypted-training fidelity and generalization on public medical datasets separately.

The NHANES tasks cover asthma, cancer, coronary heart disease (CHD), congestive heart failure (CHF), chronic obstructive pulmonary disease (COPD), depression, kidney disease, myocardial infarction (MI), and stroke. Table 2 gives their characteristics.

For each task, three institutions hold horizontal partitions and apply local scaling using shared training extrema. Both methods




(a) Runtime per Newton iteration (b) Total training time (c) Peak memory consumption (d) Multiparty initialization time

Figure 7: Multi-institutional scalability of the proposed framework as the number of participating institutions increases from 3 to 9: (a) average runtime per Newton iteration, (b) total iterative training time, (c) peak memory consumption, and (d) multiparty key-generation and initialization time. Results are averaged over nine complete NHANES prediction tasks.

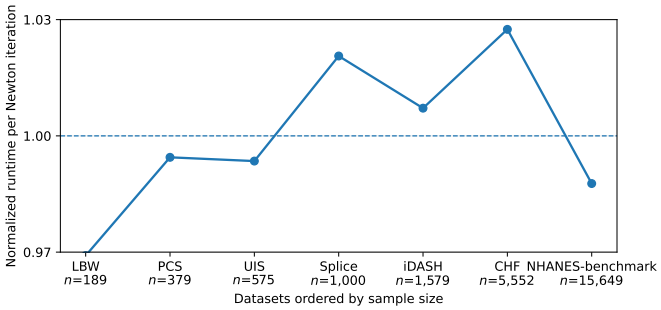



Figure 8: Empirical runtime sensitivity across benchmark datasets spanning an approximately 83-fold range in sample size. The benchmark collection follows Miao and Wu, with CHF added from our NHANES evaluation. Runtime per Newton iteration is normalized by the mean runtime across the evaluated datasets.

Table 2: Statistics of the NHANES 2017–2018 medical prediction tasks used for predictive evaluation. The reported feature dimension excludes the intercept term, and positive prevalence denotes the proportion of samples with outcome label 1.

| Task | Samples | Features | Positive (%) |
|---|---|---|---|
| Asthma | 8,888 | 67 | 14.91 |
| Cancer | 5,567 | 65 | 10.56 |
| CHD | 5,553 | 65 | 4.77 |
| CHF | 5,552 | 65 | 3.62 |
| COPD | 5,560 | 65 | 5.27 |
| Depression | 5,068 | 66 | 9.06 |
| Kidney disease | 5,560 | 65 | 4.01 |
| MI | 5,559 | 65 | 4.86 |
| Stroke | 5,559 | 65 | 4.91 |

Table 3: Five-fold predictive performance on public medical benchmarks also evaluated by SPINDLE. SPINDLE values are taken from the original publication, whereas the proposed method reports mean ± standard deviation over five held-out folds.

| Dataset | Method | Accuracy | F1 |
|---|---|---|---|
| BCW | SPINDLE | 0.962 | 0.944 |
| | Proposed | 0.9642 ± 0.0144 | 0.9478 ± 0.0214 |
| PIMA | SPINDLE | 0.780 | 0.677 |
| | Proposed | 0.7721 ± 0.0128 | 0.6368 ± 0.0318 |

use identical predictors and evaluate the same training records; outcome variables and direct proxies are excluded. Positive prevalence ranges from 3.62% to 14.91%, so ROC-AUC, PR-AUC, and average precision (AP) assess agreement under class imbalance. The NHANES results quantify fidelity on fitted cohorts rather than held-out clinical performance.

Figure 9 reports absolute metric deviations, with smaller values indicating closer agreement.

Mean ROC-AUC is 0.813351 for the encrypted model versus 0.814203 for plaintext Newton; mean PR-AUC is 0.258212 versus 0.259484, and AP is 0.259818 versus 0.261071. Mean absolute deviations are $8.52 \times 10^{-4}$, $1.86 \times 10^{-3}$, and $1.79 \times 10^{-3}$, respectively. The largest ROC-AUC deviation is 0.00393 on kidney disease; the largest PR-AUC and AP deviations are 0.00564 and 0.00559 on COPD. All remain below 0.006.

These differences measure the combined effects of damping, finite PCG iterations, and encrypted arithmetic, without isolating CKKS error. Together with the scalability results, they support end-to-end predictive agreement despite keeping patient records local and restricting encryption to global optimization.

PIMA contains 768 samples with eight features; BCW contains 699 samples with nine features after identifier removal. We use stratified five-fold cross-validation, training on approximately 80% and testing on 20% in each fold. Only training records are partitioned among three institutions. Preprocessing is fitted on each training fold and applied to its test fold.

Accuracy denotes the fraction correctly classified; F1 is the harmonic mean of positive-class precision and recall. Table 3 compares mean ± standard deviation across the five held-out folds with published SPINDLE results. The latter provide a shared-benchmark reference rather than a controlled system comparison.

BCW accuracy and F1 are 0.9642 and 0.9478, close to SPINDLE's 0.962 and 0.944. PIMA accuracy is 0.7721 versus the published 0.780, while its F1 is lower at 0.6368 versus 0.677. Discrimination remains strong: BCW ROC-AUC, PR-AUC, and AP are 0.9938 ± 0.0030, 0.9866 ± 0.0069, and 0.9867 ± 0.0068; PIMA values are 0.8323 ± 0.0093, 0.7130 ± 0.0203, and 0.7194 ± 0.0173. The fold-to-fold variation shows that these results are not based on one train/test partition.

Differences in preprocessing, folds, cryptographic implementations, hardware, and software prevent controlled efficiency comparisons with SPINDLE; no runtime or memory speedup is claimed. NHANES agreement and PIMA/BCW held-out results

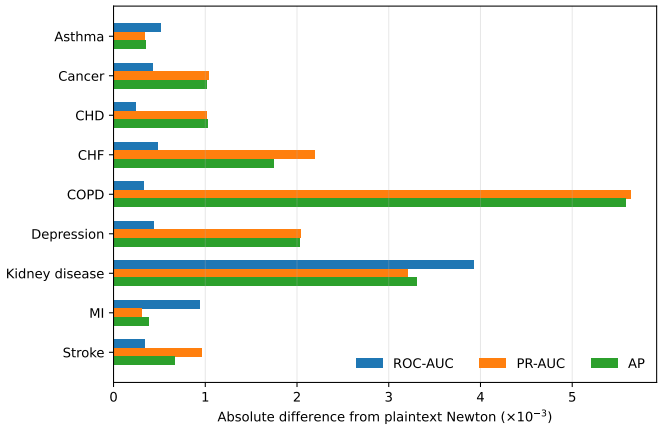



Figure 9: Absolute predictive-performance deviations between the proposed encrypted training method and pooled plaintext Newton across nine NHANES prediction tasks. Smaller values indicate closer agreement with plaintext training.

support technical feasibility but do not establish external clinical validation.

### *4.4. Controlled Comparison with Geva et al.*

We compare the proposed framework with the reproduced multi-party HE logistic-regression method of Geva et al. [17] on seven NHANES tasks: asthma, cancer, CHD, CHF, COPD, depression, and stroke. Reproduction was also attempted on kidney disease and MI. These runs failed during iterations 8 and 11, respectively, after a `NaN` intermediate caused overflow in CKKS packed encoding under 59-bit scaling. They are excluded because complete iteration-20 baseline models are unavailable, rather than because of their predictive results.

The seven completed tasks use the same server, OpenFHE-based environment, training and evaluation records, and preprocessing. Both methods use three institutions and 20 outer optimization iterations: Newton steps for the proposed method and first-order updates for Geva et al. All matched predictive, convergence, and timing results use this common window. Key initialization is excluded, and both methods share the timing and process-level memory scope defined above. Equal outer counts control the evaluation budget, although a Newton step and a first-order update perform different arithmetic workloads.

**Predictive agreement.** Both encrypted methods' iteration-20 models are compared with pooled plaintext Newton at iteration 40. The longer plaintext run supplies a common reference trajectory, without assigning extra iterations to either encrypted method. Figure 10 compares deviations on the same training records.

Mean ROC-AUC deviation is $3.99 \times 10^{-4}$ for the proposed method versus $8.74 \times 10^{-3}$ for Geva et al., a 95.4% reduction. The proposed method is closer on all seven tasks. Mean PR-AUC deviation decreases from 0.01845 to 0.00189 (89.7%), and AP deviation from 0.01853 to 0.00178 (90.4%); the proposed method is closer on six tasks for each metric. Its mean prediction-probability MAE is approximately 0.00164. These metrics quantify predictive agreement with the reference, rather than an isolated estimate of encryption error or external clinical performance.

**Convergence behavior.** Figure 11 gives the mean relative gradient over the seven matched tasks under the 20-iteration window.

The trajectories are similar initially and then separate. At iteration 5, mean relative gradients are $4.70 \times 10^{-3}$ for the proposed method and $5.69 \times 10^{-2}$ for Geva et al.; at iteration 10, they are $2.30 \times 10^{-4}$ and $7.03 \times 10^{-3}$. At iteration 20, the proposed method reaches $1.17 \times 10^{-4}$ versus $4.98 \times 10^{-3}$, approximately 43 times smaller. The reproduced Geva trajectory has entered a plateau by this stage, with little further progress beyond the reported window.

Using $10^{-3}$ only as a descriptive reference level, all seven proposed-method runs reach it within approximately 5–7 Newton iterations, compared with four Geva runs within 20 updates. It is not an experimental stopping threshold. The gradient reduction indicates greater progress toward stationarity within the common outer budget, complementing the predictive deviations without certifying a unique optimum.

Some encrypted trajectories show small late-stage fluctuations (Fig. 12).

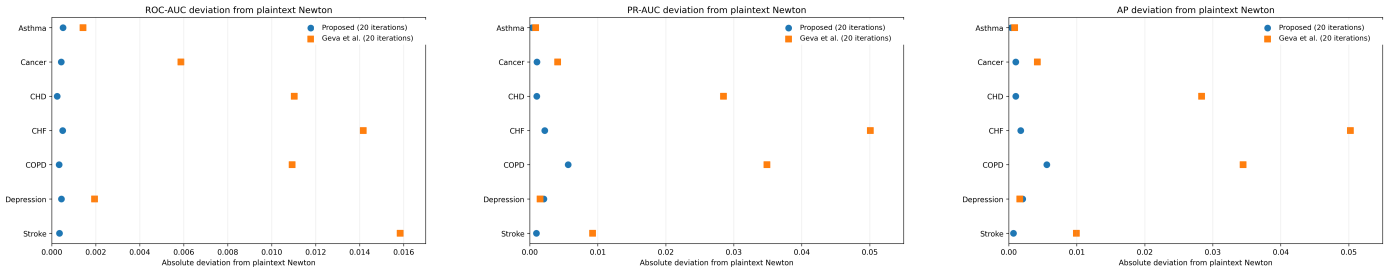
These occur after the gradient is already small and predictive metrics have stabilized. A plausible explanation is that the remaining optimization signal approaches the effective precision of CKKS arithmetic, repeated ciphertext operations, and finite-precision refreshing. Numerical uncertainty may then prevent monotonically decreasing recorded gradients. This is an empirical interpretation rather than a formal error bound; the observed fluctuations do not coincide with predictive instability or practical divergence.

**Computational efficiency.** Figure 13 compares per-iteration time, iterative training time, and peak memory on the seven completed tasks.

Average per-iteration runtime is 12.70 s for the proposed method versus 48.91 s for Geva et al., a 3.85× speedup; task-level speedups range from 3.09× to 4.38×. Under the common 20-iteration schedule, mean iterative training times are 253.91 s and 978.24 s, also a 3.85× ratio. The proposed method therefore reduces measured computation time while attaining a lower final relative gradient.

Mean peak memory is 1.365 GB versus 12.60 GB, a reduction of 89.2%; the baseline uses about 9.2 times as much memory. These measurements include process-level cryptographic context and evaluation keys as well as iterative working memory, providing a consistent comparison of the two implementations' memory requirements.

Geva et al. evaluate the nonlinear sigmoid approximation homomorphically and require a deeper circuit, using $N = 65{,}536$ and $\log_2 Q \approx 886$ here. Local patient-dependent computation allows our encrypted PCG solve to use $N = 32{,}768$ and $\log_2 Q \approx 414$. Both configurations meet the same at-least-128-bit classical-security target under the cited guidance [34]. Smaller ring dimension, shorter modulus chain, and lower encrypted depth jointly explain the runtime and memory differences; the advantage does not arise from reducing the target security level.




(a) ROC-AUC deviation (b) PR-AUC deviation (c) AP deviation

Figure 10: Absolute predictive-performance deviations from pooled plaintext Newton on seven matched NHANES prediction tasks for the proposed method and Geva et al. Smaller values indicate closer agreement with plaintext training.

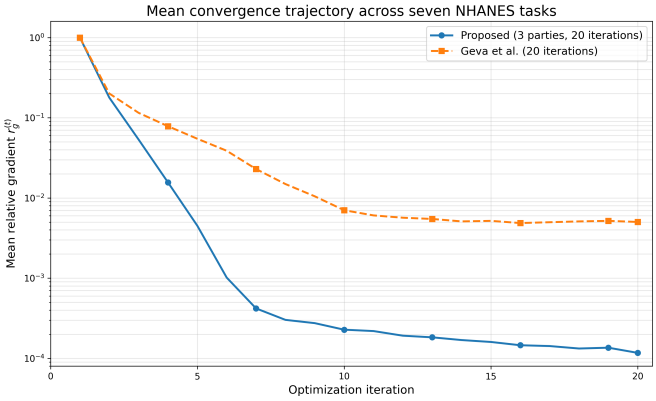



Figure 11: Mean relative-gradient convergence of the proposed method and Geva et al. over seven matched NHANES prediction tasks. Both methods are shown over a common 20-iteration optimization budget.

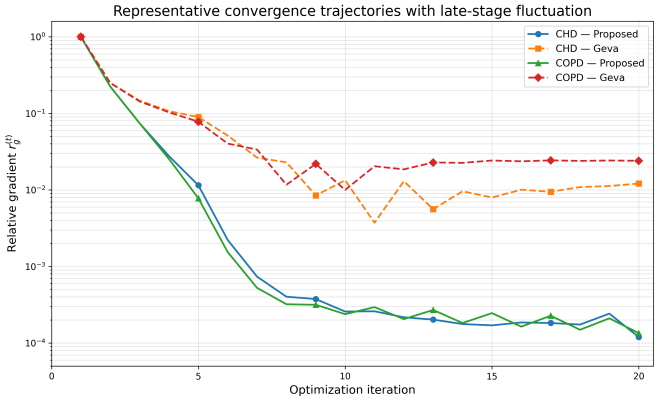



Figure 12: Representative late-stage relative-gradient fluctuations under encrypted computation.

The smaller conventional benchmarks show a different runtime pattern. On five fully matched datasets, mean per-iteration times are 11.58 s for the proposed method and 11.37 s for Geva et al. Fixed cryptographic overhead can therefore limit per-step acceleration on small problems, and 20-iteration training times are similar. Peak memory still decreases from 5.68 GB to 1.27 GB. Thus the observed memory advantage persists on these benchmarks, while the runtime advantage is most pronounced on the larger medical tasks.

Overall, the matched clinical experiments show closer predictive agreement, greater gradient reduction, and lower runtime and memory within the common evaluation window. These findings support the design choice of local nonlinear computation followed by encrypted aggregation and PCG. Their scope is the stated datasets, parameter configurations, and single-server simulation; they do not imply uniform per-iteration acceleration for every dataset or optimizer configuration.

## 5. Discussion

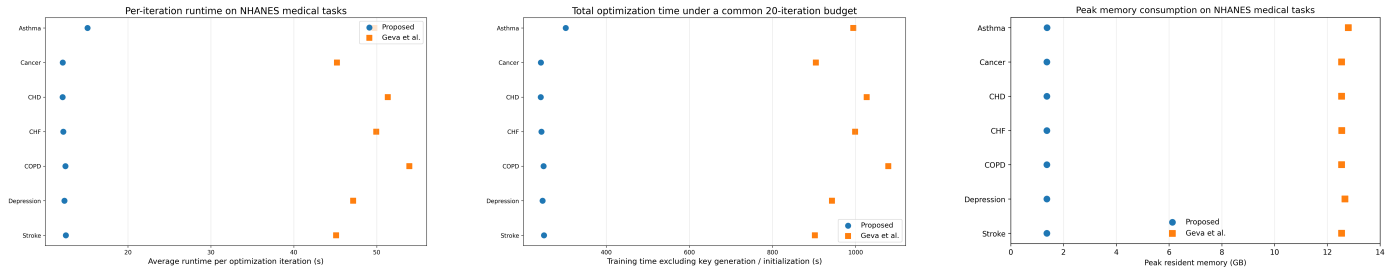
The findings support technical feasibility while identifying computational, numerical, and deployment limits of multiparty encrypted clinical prediction.

### *5.1. Clinical and Practical Implications*

The framework trains a shared logistic-regression model without pooling patient records. Institutions retain their datasets and perform patient-dependent operations locally; only encrypted Newton statistics enter collaborative optimization. This separation addresses privacy and governance constraints that restrict individual-record exchange in multi-institutional studies, while allowing all institutions to contribute to one model.

The statistical objective remains unchanged. Equation (4) reconstructs the pooled score and Hessian with the same total diagonal damping in exact arithmetic. Damping stabilizes the linear system without adding an L2 penalty to the objective. Institutions therefore optimize a global model from collective first- and second-order information, rather than fitting local models and averaging their parameters. The small NHANES metric deviations support agreement on training cohorts; PIMA and BCW cross-validation separately assess held-out utility. These forms of evidence should remain distinct when interpreting clinical relevance.

Patient records also remain local during preprocessing, although training-set feature extrema are disclosed for common scaling. This explicit metadata policy is narrower than concealment of every data-derived value. Similarly, model iterates are shared training state, whereas the server does not receive plaintext Newton statistics or model increments. The privacy claim must be understood within this information boundary and the




(a) Runtime per optimization iteration

(b) Total training time

(c) Peak memory consumption

Figure 13: Computational efficiency of the proposed method and Geva et al. on seven matched NHANES prediction tasks: (a) runtime per optimization iteration, (b) total iterative training time, and (c) peak memory consumption.

semi-honest, non-colluding threat model.

The separation of cohort size from encrypted-system size favors large patient populations with moderate numbers of predictors. After aggregation, the encrypted PCG solve runs once for the global system, limiting additional cost from more institutions. Collective CKKS distributes decryption capability and avoids assigning the complete key to one hospital, while local data control gives institutions a clear responsibility within the protocol.

Applications may include collaborative disease-risk prediction and retrospective cohort analysis, provided that institutions use harmonized features and a common training protocol. The present evidence establishes technical feasibility rather than clinical readiness. Deployment also requires governance agreements, secure communication, data harmonization, and external model validation; the experiments do not replace those requirements.

### 5.2. *Sources of Computational Efficiency and Scalability*

The main efficiency gain comes from reducing encrypted work. Local sigmoid evaluation removes homomorphic nonlinear approximation from the global circuit, which processes aggregated Newton statistics. The resulting parameters ($N = 32{,}768$, $\log_2 Q = 414$) are lighter than the reproduced Geva baseline's ($N = 65{,}536$, $\log_2 Q = 886$) at the same target security category. Smaller ciphertext representations and shorter modulus chains reduce both arithmetic cost and memory, explaining much of the difference in Section 4.4.

PCG provides a second source of efficiency. It replaces explicit encrypted Hessian inversion with packed matrix–vector products, inner products, and vector updates. A diagonal preconditioner avoids dense encrypted preconditioning, while delayed division groups reciprocal evaluations into one interactive stage. The method therefore improves optimization progress through curvature while expressing the encrypted solve with a limited set of CKKS operations. These benefits arise from the integration of workload design, packing, and optimization rather than one isolated low-level acceleration.

The comparison should be interpreted at both cryptographic and optimization levels. Under a common 20-iteration window, the proposed method attains a lower relative gradient and lower measured time on the matched clinical tasks. Each Newton step nevertheless contains an inner PCG solve, so equal outer counts do not imply identical arithmetic workloads. The measured per-iteration and total times capture that difference. On smaller benchmarks, fixed overhead limits per-step acceleration, although the memory reduction persists; the advantage is consequently workload-dependent.

Additional institutions submit and aggregate statistics and contribute decryption shares, but do not duplicate the global PCG solve. This explains the moderate runtime and memory growth observed across 3–9 participants. Larger cohorts primarily increase parallelizable local computation; their encrypted statistics keep the same dimension within a fixed feature space.

Feature count has a stronger direct effect: the Hessian grows quadratically and requires additional ciphertext blocks beyond the slot capacity. Thus the favorable cohort and institution scaling does not extend to unrestricted feature dimension. High-dimensional models remain more demanding in storage, homomorphic matrix–vector operations, and communication, even though they retain the same local/global computational boundary.

### 5.3. *Numerical Behavior under Approximate Homomorphic Encryption*

The relative gradient falls rapidly during early Newton iterations and may fluctuate after reaching a low magnitude. Inner PCG residuals can also vary non-monotonically. Strict stepwise monotonicity is therefore not preserved throughout encrypted training, although the observed overall trajectories show substantial gradient reduction and stable predictive metrics.

CKKS approximation offers a plausible explanation. Encoding, rescaling, ciphertext multiplication, and repeated operations introduce numerical perturbations. Early optimization signals are much larger, so their effect on the trajectory is limited. As gradients decrease, the remaining signal may approach the effective precision of encrypted arithmetic and refreshing, making temporary increases visible. The experiment does not isolate these contributions or establish a formal precision floor; this remains an empirical interpretation of the recorded behavior.

The late fluctuations do not coincide with unstable predictive metrics and should not automatically be read as divergence. Most visible optimization progress occurs beforehand, and additional encrypted iterations may yield little predictive improvement once numerical uncertainty dominates. However, small

gradients and stable AUC alone do not certify convergence to a unique model or establish external predictive validity.

Convergence assessment should therefore consider overall gradient and residual reduction together with predictive stability, rather than demand monotonic decrease to plaintext machine precision. We use 20 outer and 12 inner iterations in every proposed-method experiment and record diagnostics throughout. These fixed schedules avoid choosing an unexplained stopping tolerance and allow trajectories to be compared under a common budget. Optional encrypted stopping checks remain in the general protocol for deployments with justified application-specific thresholds.

A formal analysis relating CKKS approximation, circuit depth, damping, and finite PCG solves to attainable optimization accuracy is still needed. Such analysis could guide parameter selection and stopping policies. The current evidence supports the practical distinction between small late-stage numerical fluctuations and instability, without providing an encryption-only error estimate or a general convergence guarantee.

### *5.4. Limitations and Future Directions*

The security model is semi-honest and non-colluding. Malformed statistics or ciphertexts, incorrect shares, poisoned data, and Byzantine deviations require verification and robustness mechanisms beyond the present protocol, with additional cost. Consequently, the reported privacy and efficiency results should not be extended to malicious participants without further safeguards.

The adopted distributed decryption is $K$-out-of-$K$: all institutions must contribute. Interactive inversion and refresh have the same availability dependency, so a missing participant can stall a round. Threshold or dropout-tolerant decryption would improve deployment resilience and is an important extension.

Institutions are simulated on one server by partitioning a common dataset. This controls the comparison but omits wide-area latency, bandwidth limits, and hardware heterogeneity. It also does not reproduce site-specific prevalence, patient mix, or measurement practice. Evaluation across physically separate hospitals should therefore measure both real communication cost and heterogeneous institutional distributions.

NHANES results measure training-cohort agreement, with held-out utility assessed only on PIMA and BCW. Independent NHANES hold-outs and external cohorts are needed for clinical validation. Local extrema are intentionally disclosed for scaling; stricter metadata policies would require protected extrema computation rather than the present disclosure rule. Common feature definitions and consistent preprocessing remain necessary for meaningful collective statistics.

Finally, the explanation of CKKS-related fluctuations is empirical. No formal bound links approximation error, encrypted depth, and attained Newton-PCG accuracy. Damping makes the Newton matrix positive definite but does not make the original unpenalized objective strongly convex. Future analysis should retain this distinction when deriving convergence and parameter-selection criteria. Together, these limitations motivate stronger security and availability guarantees, real multi-site deployment, independent validation, and rigorous numerical analysis.

## 6. Conclusion

This work presented a privacy-preserving framework for multi-institutional clinical prediction based on multiparty homomorphic encryption and encrypted Newton-CG logistic regression. Rather than claiming the additive decomposition of Newton statistics as a new mathematical result, the framework uses this known structure as the architectural basis for general $K$-institution training, separating local patient-dependent computation from global encrypted optimization. Each institution computes its gradient and Hessian contributions locally, while only the resulting fixed-dimensional statistics are encrypted and homomorphically aggregated. As a result, the dominant homomorphic workload depends primarily on the model dimension rather than on the number of patient records, substantially reducing the ciphertext workload and multiplicative depth required for collaborative training.

To support second-order optimization across multiple independent institutions, the proposed framework integrates multiparty CKKS with encrypted Newton-CG/PCG computation under a collective key. The global Newton system is solved through ciphertext-domain PCG with diagonal preconditioning and delayed division. Distributed decryption keeps complete decryption capability out of any single institution and prevents the computation server from obtaining the plaintext Newton direction; the participating institutions jointly recover the direction only to update their synchronized model. This extends encrypted second-order logistic-regression training from previously studied two-user settings to a general $K$-institution collaboration architecture.

Experiments on nine NHANES 2017–2018 clinical-prediction tasks together with the PIMA Diabetes and Breast Cancer Wisconsin benchmarks demonstrate that the proposed framework preserves predictive behavior close to pooled-data plaintext Newton on the NHANES training cohorts and provides held-out predictive results on PIMA and BCW. In the controlled same-platform comparison with Geva et al. [17], the proposed method reaches a final relative gradient approximately 43 times lower than the first-order encrypted baseline under the same 20-iteration budget. It also reduces average iterative training time by a factor of 3.85 and peak memory consumption by 89.2%, while generally providing closer predictive agreement with the plaintext Newton reference. The scalability experiments further show that increasing the number of participating institutions introduces only moderate additional overhead, while increasing the number of patient records has little effect on the dominant encrypted PCG workload.

Overall, the results indicate that combining local Newton-statistic computation with low-depth multiparty encrypted second-order optimization provides a practical route toward privacy-preserving multi-institutional clinical prediction. Future work will focus on deployment across physically distributed healthcare institutions, stronger adversarial security models, and a more rigorous analysis of the relationship between CKKS approximation error and the attainable convergence accuracy of encrypted Newton-PCG training.

## CRediT authorship contribution statement

**Xiaokang Dai:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Weijie Miao:** Methodology, Validation, Writing – review & editing. **Wenyuan Wu:** Conceptualization, Methodology, Supervision, Funding acquisition, Writing – review & editing. **Yong Feng:** Supervision, Resources, Funding acquisition, Writing – review & editing.


## Funding

This research is supported by the National Key Research Project of China under Grant 2025YFA1017201 and by the National Natural Science Foundation of China under Grants 12301650 and 12571553.


## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data Availability

The datasets used in this study are publicly available from their original repositories, including NHANES 2017–2018 and the public benchmark datasets described in Section 4. Detailed task construction and preprocessing procedures are described in the experimental section and accompanying Supplementary Information.

## Declaration of generative AI and AI-assisted technologies in the manuscript preparation process.

During the preparation of this work the author(s) used ChatGPT for language editing and flowchart preparation. After using this service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

## References


[1] A. Kerezis, A. Antonoglou, A. Asimakos, N. Athanasiou, S. Spetsioti, K. Dalakleidi, P. Asvestas, P. Katsaounou, S. Golemati, A machine learning framework for efficient triage of Long-COVID patients to specialists, in: 2025 47th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2025, pp. 1–4. doi:10.1109/EMBC58623.2025.11252756.

[2] T. Kıvrak, M. A. Gelen, O. Salkin, O. Karaca, P. D. Barua, S. Dogan, T. Tuncer, R.-S. Tan, M. Salvi, V. S. Subbhuraam, U. R. Acharya, Translational application of a self-organized deep feature engineering pipeline for non-invasive pulmonary hypertension classification from routine chest radiographs, Intelligent Medicine 6 (3) (2026) 272–284. doi:10.1016/j.imed.2025.06.003.

[3] D. R. Cox, The regression analysis of binary sequences, Journal of the Royal Statistical Society: Series B (Methodological) 20 (2) (1958) 215–232. doi:10.1111/j.2517-6161.1958.tb00292.x.

[4] M. Yang, K. Nurzynska, A. E. Walts, A. Gertych, A CNN-based active learning framework to identify mycobacteria in digitized Ziehl-Neelsen stained human tissues, Computerized Medical Imaging and Graphics 84 (2020) 101752. doi:10.1016/j.compmedimag.2020.101752.

[5] S. Seoni, A. Shahini, K. M. Meiburger, F. Marzola, G. Rotunno, U. R. Acharya, F. Molinari, M. Salvi, All you need is data preparation: A systematic review of image harmonization techniques in Multi-center/device studies for medical support systems, Computer Methods and Programs in Biomedicine 250 (2024) 108200. doi:10.1016/j.cmpb.2024.108200.

[6] L. Alzubaidi, S. A. Jebur, T. A. Jaber, M. A. Mohammed, H. A. Alwzwazy, A. Saihood, H. Gammulle, J. Santamaria, Y. Duan, C. Fookes, R. Jurdak, Y. Gu, ATD Learning: A secure, smart, and decentralised learning method for big data environments, Information Fusion 118 (2025) 102953. doi:10.1016/j.inffus.2025.102953.

[7] X. Yang, D. Wan, G. Han, W. Zhang, W. Tang, Personalized federated learning with hierarchical reweighting for multi-center clinical prediction, Computer Methods and Programs in Biomedicine 271 (2025) 109015. doi:10.1016/j.cmpb.2025.109015.

[8] D. Ciupek, M. Malawski, T. Pieciak, Federated learning: A new frontier in the exploration of multi-institutional medical imaging data, Computer Methods and Programs in Biomedicine 284 (2026) 109454. doi:10.1016/j.cmpb.2026.109454.

[9] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, ACM, 2009, pp. 169–178. doi:10.1145/1536414.1536440.

[10] A. Kim, Y. Song, M. Kim, K. Lee, J. H. Cheon, Logistic regression model training based on the approximate homomorphic encryption, BMC Medical Genomics 11 (Suppl 4) (2018) 83. doi:10.1186/s12920-018-0401-7.

[11] C. Bonte, F. Vercauteren, Privacy-preserving logistic regression training, BMC Medical Genomics 11 (Suppl 4) (2018) 86. doi:10.1186/s12920-018-0398-y.

[12] H. Chen, R. Gilad-Bachrach, K. Han, Z. Huang, A. Jalali, K. Laine, K. Lauter, Logistic regression over encrypted data from fully homomorphic encryption, BMC Medical Genomics 11 (Suppl 4) (2018) 81. doi:10.1186/s12920-018-0397-z.

[13] V. S. Naresh, S. Reddi, Exploring the future of privacy-preserving heart disease prediction: A fully homomorphic encryption-driven logistic regression approach, Journal of Big Data 12 (2025) 52. doi:10.1186/s40537-025-01098-6.

[14] D. Froelicher, J. R. Troncoso-Pastoriza, A. Pyrgelis, S. Sav, J. Sa Sousa, J.-P. Bossuat, J.-P. Hubaux, Scalable privacy-preserving distributed learning, Proceedings on Privacy Enhancing Technologies 2021 (2) (2021) 323–347. doi:10.2478/popets-2021-0030.

[15] H. Wang, T. Chen, Y. Ding, Y. Wang, C. Yang, Privacy-preserving multiparty logistic regression in cloud computing, Computer Standards & Interfaces 90 (2024) 103857. doi:10.1016/j.csi.2024.103857.

[16] D. H. E. Kang, D. Kim, Y. Song, D. Lee, H. Kwak, B. W. Anthony, Harnessing the potential of shared data in a secure, inclusive, and resilient manner via multi-key homomorphic encryption, Scientific Reports 14 (2024) 13626. doi:10.1038/s41598-024-63393-1.

[17] R. Geva, A. Gusev, Y. Polyakov, L. Liram, O. Rosolio, A. Alexandru, N. Genise, M. Blatt, Z. Duchin, B. Waissengrin, D. Mirelman, F. Bukstein, D. T. Blumenthal, I. Wolf, S. Pelles-Avraham, T. Schaffer, L. A. Lavi, D. Micciancio, V. Vaikuntanathan, A. Al Badawi, S. Goldwasser, Collaborative privacy-preserving analysis of oncological data using multiparty homomorphic encryption, Proceedings of the National Academy of Sciences of the United States of America 120 (33) (2023) e2304415120. doi:10.1073/pnas.2304415120.

[18] R. S. Dembo, S. C. Eisenstat, T. Steihaug, Inexact newton methods, SIAM Journal on Numerical Analysis 19 (2) (1982) 400–408. doi:10.1137/0719025.

[19] M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards 49 (6) (1952) 409–436. doi:10.6028/jres.049.044.

[20] W. Miao, W. Wu, Privacy-preserving logistic regression model training scheme by homomorphic encryption, in: Information and Communications Security: 26th International Conference, ICICS 2024, Mytilene, Greece, August 26–28, 2024, Proceedings, Part I, Vol. 15056 of Lecture Notes in Computer Science, Springer, Singapore, 2025, pp. 271–291, conference:

ICICS 2024. doi:10.1007/978-981-97-8798-2_14.

[21] Y. Wu, X. Jiang, J. Kim, L. Ohno-Machado, Grid binary logistic regression (GLORE): Building shared models without sharing data, Journal of the American Medical Informatics Association 19 (5) (2012) 758–764. doi:10.1136/amiajnl-2012-000862.

[22] J. H. Cheon, A. Kim, M. Kim, Y. Song, Homomorphic encryption for arithmetic of approximate numbers, in: Advances in Cryptology – ASIACRYPT 2017, Vol. 10624 of Lecture Notes in Computer Science, Springer, Cham, 2017, pp. 409–437. doi:10.1007/978-3-319-70694-8_15.

[23] M. Benzi, Preconditioning techniques for large linear systems: A survey, Journal of Computational Physics 182 (2) (2002) 418–477. doi:10.1006/jcph.2002.7176.

[24] A. Al Badawi, A. Alexandru, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, C. Pascoe, Y. Polyakov, I. Quah, Saraswathy R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, V. Zucca, OpenFHE: Open-source fully homomorphic encryption library, Cryptology ePrint Archive, Paper 2022/915 (2022).
URL https://eprint.iacr.org/2022/915

[25] J. W. Bos, K. Lauter, M. Naehrig, Private predictive analysis on encrypted medical data, Journal of Biomedical Informatics 50 (2014) 234–243. doi:10.1016/j.jbi.2014.04.003.

[26] Y. Aono, T. Hayashi, L. Trieu Phong, L. Wang, Scalable and secure logistic regression via homomorphic encryption, in: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, ACM, 2016, pp. 142–144. doi:10.1145/2857705.2857731.

[27] M. Kim, Y. Song, S. Wang, Y. Xia, X. Jiang, Secure logistic regression based on homomorphic encryption: Design and evaluation, JMIR Medical Informatics 6 (2) (2018) e19. doi:10.2196/medinform.8805.

[28] S. Wei, Z. Chen, X. Chen, B. Dong, Y. Ren, J. Fu, Secure logistic regression training based on fully homomorphic encryption, International Journal of Information and Computer Security 28 (4) (2025) 424–444. doi:10.1504/IJICS.2025.150024.

[29] Y. Bai, H. Zhao, X. Shi, L. Chen, Towards practical and privacy-preserving CNN inference service for cloud-based medical imaging analysis: A homomorphic encryption-based approach, Computer Methods and Programs in Biomedicine 261 (2025) 108599. doi:10.1016/j.cmpb.2025.108599.

[30] Y. Nardi, S. E. Fienberg, R. J. Hall, Achieving both valid and secure logistic regression analysis on aggregated data from different private sources, Journal of Privacy and Confidentiality 4 (1) (2012) 189–220. doi:10.29012/jpc.v4i1.617.

[31] C.-H. Pham, T. Huynh-The, E. Sedgh-Gooya, M. El-Bouz, A. Alfalou, Extension of physical activity recognition with 3D CNN using encrypted multiple sensory data to federated learning based on multi-key homomorphic encryption, Computer Methods and Programs in Biomedicine 243 (2024) 107854. doi:10.1016/j.cmpb.2023.107854.

[32] X. Lu, H. U. Sami, B. Güler, SCALR: Communication-efficient secure multi-party logistic regression, IEEE Transactions on Communications 72 (1) (2024) 162–178. doi:10.1109/TCOMM.2023.3308954.

[33] S. Halevi, V. Shoup, Algorithms in HElib, in: Advances in Cryptology – CRYPTO 2014, Vol. 8616 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2014, pp. 554–571. doi:10.1007/978-3-662-44371-2_31.

[34] J.-P. Bossuat, R. Cammarota, I. Chillotti, B. R. Curtis, W. Dai, H. Gong, E. Hales, D. Kim, B. Kumara, C. Lee, X. Lu, C. Maple, A. Pedrouzo-Ulloa, R. Player, Y. Polyakov, L. A. Ruiz Lopez, Y. Song, D. Yhee, Security guidelines for implementing homomorphic encryption, IACR Communications in Cryptology 1 (4) (2025). doi:10.62056/anxra69p1.

[35] National Center for Health Statistics, NHANES 2017–2018 overview, Centers for Disease Control and Prevention, National Health and Nutrition Examination Survey, 2017–2018 survey cycle; accessed 2026-10-01 (n.d.).
URL https://wwwn.cdc.gov/nchs/nhanes/continuousnhanes/overview.aspx?BeginYear=2017