PRIVACY-PRESERVING LOGISTIC REGRESSION TRAINING WITH A FASTER GRADIENT VARIANT

Anonymous authors
Paper under double-blind review

ABSTRACT

Training logistic regression over encrypted data has emerged as a prominent approach to addressing security concerns in recent years. In this paper, we introduce an efficient gradient variant, termed the *quadratic gradient*, which is specifically designed for privacy-preserving logistic regression while remaining equally effective in plaintext optimization. By incorporating this quadratic gradient, we enhance Nesterov's Accelerated Gradient (NAG), Adaptive Gradient (AdaGrad), and Adam algorithms. We evaluate these enhanced algorithms across various datasets, with experimental results demonstrating state-of-the-art convergence rates that significantly outperform traditional first-order gradient methods. Furthermore, we apply the enhanced NAG method to implement homomorphic logistic regression training, achieving comparable performance within only four iterations. The proposed quadratic-gradient approach offers a unified framework that synergizes the advantages of first-order gradient-based methods and second-order Newton-type methods, suggesting broad applicability to diverse numerical optimization tasks.

1 INTRODUCTION

1.1 BACKGROUND

Given a patient's longitudinal healthcare data, one can train a logistic regression (LR) model to predict the likelihood of disease onset. However, personal health information is inherently sensitive, and privacy concerns remain a significant barrier to the large-scale sharing and aggregation of biomedical data. To address this, Homomorphic Encryption (HE) offers a robust security paradigm: data can be transformed into ciphertexts and outsourced to the cloud for computation without ever being decrypted, thereby preventing unauthorized access by the service provider.

The annual iDASH competition has become a benchmark for evaluating cryptographic technologies in biomedical settings. Since 2014, iDASH has promoted the development of privacy-preserving solutions for genomic and healthcare data analysis. In particular, the 2017 (Track 3) and 2018 (Track 2) competitions focused on efficient homomorphic-encryption-based training of logistic regression models, highlighting the practical importance of scalable encrypted learning techniques.

1.2 RELATED WORK

Several studies have explored the implementation of logistic regression models within homomorphic encryption frameworks. Aono et al. (2016b;a) proposed one of the earliest homomorphic encryption-based frameworks for privacy-preserving logistic regression over distributed datasets, demonstrating the feasibility of secure collaborative model training. Kim et al. (2018b) addressed the challenges of encrypted LR training by employing full-batch gradient descent and utilizing least-squares approximation for the sigmoid function. During the 2017 iDASH competition, this problem was further investigated by (Bonte & Vercauteren, 2018), (Kim et al., 2018a), (Chen et al., 2018), and (Crawford et al., 2018). In the subsequent 2018 iDASH competition, Kim et al. (2019b) and Blatt et al. (2019) introduced efficient ciphertext packing techniques and semi-parallel algorithms to enhance computational throughput.

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

Recent studies have explored privacy-preserving logistic regression training through various cryptographic techniques, including CKKS-based SIMD acceleration (Bergamaschi et al., 2019), HE combined with differential privacy (Kim et al., 2019a), optimized MPC protocols (De Cock et al., 2019; 2021), and circuit-level optimization for reducing homomorphic computation costs (Crockett, 2020; Li & Huang, 2020). Hybrid HE frameworks integrating TFHE, HEAAN, and MPC have further improved encrypted training efficiency (Carpov et al., 2020; Han et al., 2020). Other works investigated HE-friendly reformulations and extended encrypted learning to inference and neural network training (Byun et al., 2021; Hong et al., 2024; Montero et al., 2024).

Existing encrypted logistic regression approaches mainly rely on first-order optimizers, which require learning rate tuning that is difficult in the ciphertext domain. The SFH method (Bonte & Vercauteren, 2018) eliminates this requirement using a fixed Hessian approximation, but suffers from limitations on sparse datasets. Although Ogilvie et al. (2020) extended SFH to CKKS-based regression, it still relies on data normalization. We address these limitations by developing a generalized fixed-Hessian framework and leveraging ciphertext packing techniques (Kim et al., 2018a) for efficient homomorphic optimization.

1.3 Contributions

Our specific contributions in this paper are summarized as follows:

1. We introduce a novel gradient variant, termed the *quadratic gradient*, which bridges first-order gradient-based methods and the second-order Newton-type algorithms. This enables a unified framework that synergistically leverages the computational efficiency of first-order methods and the rapid convergence of second-order approaches.
2. We develop three enhanced optimization algorithms by integrating the quadratic gradient into existing frameworks. Experimental results demonstrate that these enhanced algorithms achieve state-of-the-art performance in terms of convergence rates across diverse datasets.
3. We implement privacy-preserving logistic regression training using the enhanced Nesterov's Accelerated Gradient (NAG) method. To the best of our knowledge, our implementation strikes an optimal balance between computational efficiency and storage overhead without significant performance degradation.

2 Preliminaries

In the following discussion, we employ square brackets $[\cdot]$ to denote the indexing of vectors and matrices. Specifically, for a vector $\boldsymbol{v} \in \mathbb{R}^n$ and a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, $\boldsymbol{v}[i]$ (or $\boldsymbol{v}_i$) denotes the i-th element of $\boldsymbol{v}$, while $\mathbf{M}[i][j]$ (or $\mathbf{M}_{i,j}$) refers to the element in the i-th row and j-th column of $\mathbf{M}$.

2.1 Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) is a cryptographic primitive that enables an arbitrary number of additions and multiplications directly on ciphertexts. While Gentry proposed the first feasible FHE scheme using a *bootstrapping* operation in 2009 (Gentry, 2009), such schemes remain computationally intensive. Beyond the inherent complexity of FHE, the efficiency of homomorphic computation is significantly influenced by the choice of dataset encoding and the management of plaintext magnitudes (Jäschke & Armknecht, 2016). To address the latter, Cheon et al. (2017) introduced an HE scheme featuring a *rescaling* procedure that effectively mitigates this technical bottleneck.

In our implementation of homomorphic LR algorithms, we adopt the open-source library `HEAAN`, which realizes this rescaling functionality. Furthermore, leveraging ciphertext packing is essential to achieve a superior amortized runtime. `HEAAN` supports Single Instruction, Multiple Data (`SIMD`) operations (Smart & Vercauteren, 2014), allowing multiple complex numbers to be packed into the slots of a single ciphertext polynomial and enabling efficient rotation operations across these slots. The mathematical foundations of the `HEAAN` scheme are detailed in (Kim et al., 2018a;b; Han et al., 2019), with the underlying abstract algebra further elaborated in (Artin, 2011).

2.2 DATABASE ENCODING METHOD

Kim et al. (2018a) proposed a highly efficient database encoding method that leverages `SIMD` batching to optimize both computational and storage resources. Given a training dataset $\mathbf{Z}$ containing n samples, each with $d+1$ covariates, the authors packed the entire matrix into a single ciphertext in a row-major format, thereby maximizing the utilization of available ciphertext slots.

Under this encoding paradigm, the data matrix $\mathbf{Z}$ can be manipulated directly through homomorphic operations on its ciphertext representation, $\text{Enc}(\mathbf{Z})$. This is achieved using a minimal set of primitive operations: rotation, addition, and multiplication. For example, to isolate the first column of $\text{Enc}(\mathbf{Z})$ while masking others, one can multiply the ciphertext by a constant matrix $\mathbf{Z}$ (where the first column contains ones and all other elements are zeros).

Han et al. (2019) delineated several fundamental yet critical operations employed in the implementation by (Kim et al., 2018a), such as the `SumColVec` procedure for computing column-wise summations of a matrix. By composing these primitive operations, more sophisticated computations—most notably the gradient evaluation required for logistic regression—can be efficiently realized in the encrypted domain.

2.3 LOGISTIC REGRESSION MODEL

Logistic regression is widely used in binary classification tasks to infer whether a binary-valued variable belongs to a certain class or not. LR generalizes linear regression (Murphy, 2012) by mapping the linear predictor $\boldsymbol{\beta}^\top \mathbf{x}$ to the interval $(0, 1)$ via the sigmoid function $\sigma(z) = (1 + \exp(-z))^{-1}$. Here, $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$ denotes the model parameters, and $\mathbf{x} = (1, x_1, \ldots, x_d)^\top \in \mathbb{R}^{d+1}$ represents the input covariate vector. For a class label $y \in \{+1, -1\}$, the conditional probabilities are formulated as:

$$\Pr(y = +1 \mid \mathbf{x}, \boldsymbol{\beta}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\beta}^\top \mathbf{x}}},$$
$$\Pr(y = -1 \mid \mathbf{x}, \boldsymbol{\beta}) = 1 - \sigma(\boldsymbol{\beta}^\top \mathbf{x}) = \frac{1}{1 + e^{\boldsymbol{\beta}^\top \mathbf{x}}}.$$

A classification decision is typically made by comparing the output probability against a predefined threshold (e.g., 0.5).

Training an LR model is framed as a Maximum Likelihood Estimation (MLE) problem, seeking $\boldsymbol{\beta}$ that maximizes the likelihood $L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \Pr(y_i \mid \mathbf{x}_i, \boldsymbol{\beta})$. For computational tractability, we maximize the log-likelihood function $l(\boldsymbol{\beta})$:

$$l(\boldsymbol{\beta}) = \ln L(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \ln(1 + e^{-y_i \boldsymbol{\beta}^\top \mathbf{x}_i}),$$

where n denotes the number of training samples. Since $l(\boldsymbol{\beta})$ lacks a closed-form solution, parameters are typically estimated using iterative optimization: (a) first-order methods, such as gradient descent; and (b) second-order methods, such as Newton's method. The gradient and Hessian of $l(\boldsymbol{\beta})$ are given by:

$$\nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) = \sum_{i=1}^{n} (1 - \sigma(y_i \boldsymbol{\beta}^\top \mathbf{x}_i)) y_i \mathbf{x}_i, \nabla^2_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) \quad = -\mathbf{X}^\top \mathbf{S} \mathbf{X},$$

where $\mathbf{S}$ is a diagonal matrix with entries $\mathbf{S}_{ii} = \sigma(y_i \boldsymbol{\beta}^\top \mathbf{x}_i)(1 - \sigma(y_i \boldsymbol{\beta}^\top \mathbf{x}_i))$, and $\mathbf{X} \in \mathbb{R}^{n \times (d+1)}$ denotes the dataset matrix.

The log-likelihood $l(\boldsymbol{\beta})$ is a concave function with at most one unique global maximum (Allison, 2008), where its gradient is zero. Newton's method, as a second-order technique, finds the optimal $\boldsymbol{\beta}$ by iteratively solving for the roots of the optimality condition $\nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) = 0$.

3 METHODOLOGY

Newton's method requires recomputing the Hessian matrix and its inverse at each iteration, a process that is often computationally prohibitive. To mitigate this burden, the *fixed Hessian Newton method* (also known as the *lower bound principle*) replaces the varying Hessian with a constant matrix $\bar{\mathbf{H}}$, thereby enhancing computational efficiency.

Böhning & Lindsay (1988) established that the convergence of Newton's method is guaranteed provided that $\bar{\mathbf{H}} \preceq \nabla^2_{\boldsymbol{\beta}} l(\boldsymbol{\beta})$, where $\bar{\mathbf{H}}$ is a symmetric negative-definite matrix independent of $\boldsymbol{\beta}$. Here, "$\preceq$" denotes the Loewner partial order, implying that the difference $\nabla^2_{\boldsymbol{\beta}} l(\boldsymbol{\beta}) - \bar{\mathbf{H}}$ is positive semi-definite. By employing such a fixed Hessian $\bar{\mathbf{H}}$, the Newton iteration simplifies to the following update rule:

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \bar{\mathbf{H}}^{-1} \nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}_t).$$

Böhning and Lindsay further suggest that the fixed matrix $\bar{\mathbf{H}} = -\frac{1}{4}\mathbf{X}^\top \mathbf{X}$ serves as an effective lower bound for the Hessian of the log-likelihood function $l(\boldsymbol{\beta})$ in logistic regression.

3.1 SIMPLIFIED FIXED HESSIAN

Bonte & Vercauteren (2018) further simplify the fixed Hessian $\bar{\mathbf{H}}$ to facilitate its inversion within the encrypted domain. Specifically, they approximate $\bar{\mathbf{H}}$ with a diagonal matrix $\mathbf{B}$, where each diagonal element b_{ii} is defined as the sum of the corresponding row in $\bar{\mathbf{H}}$. Furthermore, they propose an optimized computational schedule to construct $\mathbf{B}$ more efficiently. This diagonal approximation is formulated as follows:

$$\mathbf{B} = \begin{bmatrix} \sum_{j=0}^{d} \bar{h}_{0j} & 0 & \dots & 0 \\ 0 & \sum_{j=0}^{d} \bar{h}_{1j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sum_{j=0}^{d} \bar{h}_{dj} \end{bmatrix},$$

where $\bar{h}_{kj}$ denotes the (k, j)-th entry of $\bar{\mathbf{H}}$.

This diagonal structure significantly simplifies the matrix representation, allowing $\mathbf{B}$ to be efficiently derived from $\bar{\mathbf{H}}$. The inverse $\mathbf{B}^{-1}$ can be homomorphically approximated by computing the reciprocal of each diagonal element via an iterative Newton-Raphson method with a carefully chosen initial value. Consequently, the SFH update rule is formulated as:

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \mathbf{B}^{-1} \nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}_t) \quad = \boldsymbol{\beta}_t - \begin{bmatrix} b_{00}^{-1} & 0 & \dots & 0 \\ 0 & b_{11}^{-1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_{dd}^{-1} \end{bmatrix} \begin{bmatrix} \nabla_0 \\ \nabla_1 \\ \vdots \\ \nabla_d \end{bmatrix} = \boldsymbol{\beta}_t - \begin{bmatrix} b_{00}^{-1} \cdot \nabla_0 \\ b_{11}^{-1} \cdot \nabla_1 \\ \vdots \\ b_{dd}^{-1} \cdot \nabla_d \end{bmatrix},$$

where b_{ii}^{-1} denotes the reciprocal of $\sum_{j=0}^{d} \bar{h}_{ij}$, and ∇_i represents the i-th element of the gradient vector $\nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta}_t)$.

We characterize $\mathbf{B}^{-1} \nabla_{\boldsymbol{\beta}} l(\boldsymbol{\beta})$ as an *augmented gradient* and incorporate a dedicated learning rate to capture its underlying dynamics. Provided this learning rate decays from a value exceeding unity (e.g., 2.0) toward 1.0 within a finite number of iterations, the fixed-Hessian principle theoretically ensures the convergence of our proposed scheme.

Despite its advantages, the SFH method (Bonte & Vercauteren, 2018) suffers from two primary limitations. First, its convergence is strictly guaranteed only when all entries of the symmetric matrix $\bar{\mathbf{H}}$ are non-positive. While feature normalization (e.g., scaling to $[0, 1]$) often satisfies this condition in machine learning, it may not hold in general numerical optimization. Second, both the simplified Hessian $\mathbf{B}$ and the fixed Hessian $\bar{\mathbf{H}} = -\frac{1}{4}\mathbf{X}^\top \mathbf{X}$ are prone to singularity, particularly in high-dimensional sparse settings such as the MNIST dataset.

To address these issues, we generalize the SFH framework to ensure the invertibility of the simplified Hessian under arbitrary conditions. Building on this generalization, we introduce an accelerated gradient variant, termed the *quadratic gradient*.

3.2 QUADRATIC GRADIENT: DEFINITION

Suppose a differentiable scalar-valued function $F(\mathbf{x})$ has a gradient $\boldsymbol{g}$ and a Hessian matrix $\mathbf{H}$. For the maximization task, let $\bar{\mathbf{H}}$ be any matrix satisfying the condition $\bar{\mathbf{H}} \preceq \mathbf{H}$ in the Loewner partial ordering. We represent these quantities as:

$$\boldsymbol{g} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_d \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \nabla^2_{00} & \nabla^2_{01} & \dots & \nabla^2_{0d} \\ \nabla^2_{10} & \nabla^2_{11} & \dots & \nabla^2_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^2_{d0} & \nabla^2_{d1} & \dots & \nabla^2_{dd} \end{bmatrix}, \quad \bar{\mathbf{H}} = \begin{bmatrix} \bar{h}_{00} & \bar{h}_{01} & \dots & \bar{h}_{0d} \\ \bar{h}_{10} & \bar{h}_{11} & \dots & \bar{h}_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{h}_{d0} & \bar{h}_{d1} & \dots & \bar{h}_{dd} \end{bmatrix},$$

where $\nabla^2_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}$. We construct a diagonal matrix $\tilde{\mathbf{B}}$, referred to as the *diagonal Hessian approximation*, where each diagonal entry $\tilde{B}_{kk}$ is defined as:

$$\tilde{B}_{kk} = -\epsilon - \sum_{i=0}^{d} |\bar{h}_{ki}|.$$

Formally, $\tilde{\mathbf{B}}$ is structured as:

$$\tilde{\mathbf{B}} = \begin{bmatrix} -\epsilon - \sum_{j=0}^{d} |\bar{h}_{0j}| & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -\epsilon - \sum_{j=0}^{d} |\bar{h}_{dj}| \end{bmatrix},$$

where $\epsilon > 0$ is a small constant (typically set to $1\text{e}{-8}$) introduced for numerical stability, such as ensuring strict negative definiteness and preventing division by zero during inversion.

For the approximation $\tilde{\mathbf{B}}$ to serve as a valid lower bound within the fixed-Hessian framework, it must satisfy the convergence criterion $\tilde{\mathbf{B}} \preceq \mathbf{H}$. Given our foundational assumption that $\bar{\mathbf{H}} \preceq \mathbf{H}$, this requirement is satisfied by establishing that $\tilde{\mathbf{B}} \preceq \bar{\mathbf{H}}$. Following the analytical approach detailed in (Bonte & Vercauteren, 2018), it can be shown that $\tilde{\mathbf{B}}$ consistently maintains this lower-bound property relative to $\bar{\mathbf{H}}$, thereby ensuring stable convergence in the encrypted domain without the computational overhead of the full Hessian.

Definition 1 (Quadratic Gradient) *Utilizing the diagonal approximation $\tilde{\mathbf{B}}$, we define the quadratic gradient as $\mathbf{G} = \bar{\mathbf{B}}\boldsymbol{g}$, where $\bar{\mathbf{B}}$ is a diagonal matrix with entries $\bar{B}_{kk} = 1/|\tilde{B}_{kk}|$. For the maximization of $F(\mathbf{x})$, we employ the iterative update rule:*

$$\mathbf{x}_{t+1} = \mathbf{x}_t + N_t \cdot \mathbf{G},$$

where $N_t \geq 1$ is a dynamic learning rate designed to decay toward 1.0 over a finite number of iterations.

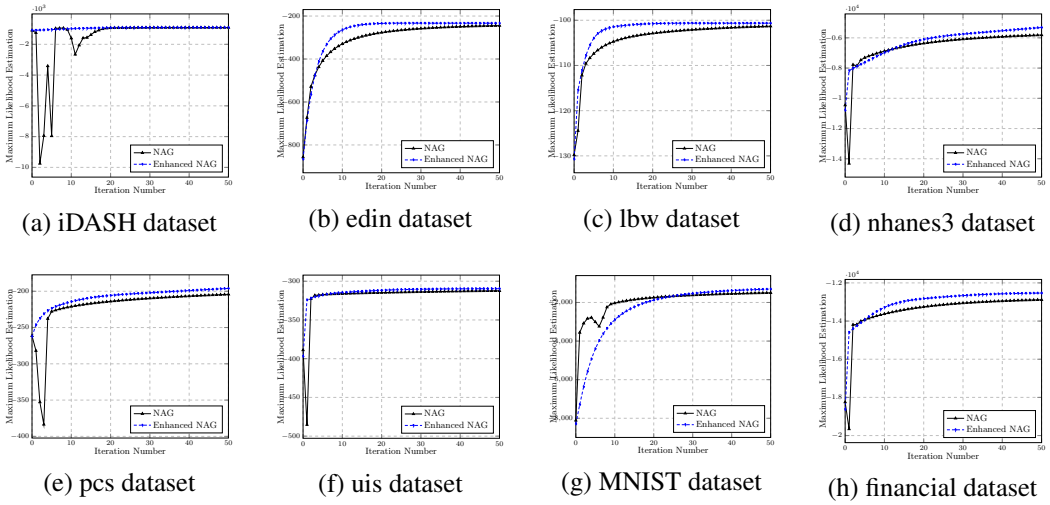
Minimizing $F(\mathbf{x})$ is equivalent to maximizing $-F(\mathbf{x})$. In this context, $\tilde{\mathbf{B}}$ can be constructed using either a lower bound of the Hessian of $-F(\mathbf{x})$ or an upper bound of the Hessian of $F(\mathbf{x})$. Notably, $\bar{\mathbf{H}}$ may be set to the Hessian $\mathbf{H}$ itself when available, bypassing the need for separate bound estimation.

3.3 QUADRATIC GRADIENT: ALGORITHMS

The quadratic gradient framework can be generalized to enhance various first-order optimization algorithms by incorporating second-order curvature information into their respective update rules:

1. **NAG:** NAG is a variant of the momentum method designed to provide the momentum term with greater prescience. The standard iterative formulas for NAG in a gradient *ascent* context are:

$$\mathbf{V}_{t+1} = \boldsymbol{\beta}_t + \eta_t \nabla J(\boldsymbol{\beta}_t), \tag{1}$$
$$\boldsymbol{\beta}_{t+1} = (1 - \gamma_t)\mathbf{V}_{t+1} + \gamma_t \mathbf{V}_t, \tag{2}$$



(a) iDASH dataset (b) edin dataset (c) lbw dataset (d) nhanes3 dataset

(e) pcs dataset (f) uis dataset (g) MNIST dataset (h) financial dataset

Figure 1: Training results of NAG and Enhanced NAG in the plaintext domain across eight datasets.

where $\mathbf{V}_{t+1}$ is an intermediate variable and $\gamma_t \in (0, 1)$ is a smoothing parameter for the moving average, enabling gradient evaluation at an approximate future position (Kim et al., 2018a). The *Enhanced NAG* algorithm modifies the original update rule equation 1 by substituting the vanilla gradient with the quadratic gradient:

$$\mathbf{V}_{t+1} = \boldsymbol{\beta}_t + N_t \mathbf{B}^{-1} \nabla J(\boldsymbol{\beta}_t) = \boldsymbol{\beta}_t + N_t \mathbf{G}_t,$$

where N_t is the adaptive learning rate for the enhanced variant. In this work, we typically set $N_t = 1 + \eta_t$, where η_t follows a specific decay schedule.

2. **AdaGrad:** AdaGrad is particularly well-suited for sparse data by adapting the learning rate for each parameter based on historical gradients. The update rules for standard AdaGrad and its *Enhanced Quadratic* version for each parameter $\boldsymbol{\beta}_{\{(i)\}}$ at iteration t are given by:

$$\text{(Standard)} \quad \boldsymbol{\beta}_{t+1}^{(i)} = \boldsymbol{\beta}_t^{(i)} - \frac{\eta_t}{\epsilon + \sqrt{\sum_{k=1}^{t} (\boldsymbol{g}_k^{(i)})^2}} \cdot \boldsymbol{g}_t^{(i)},$$

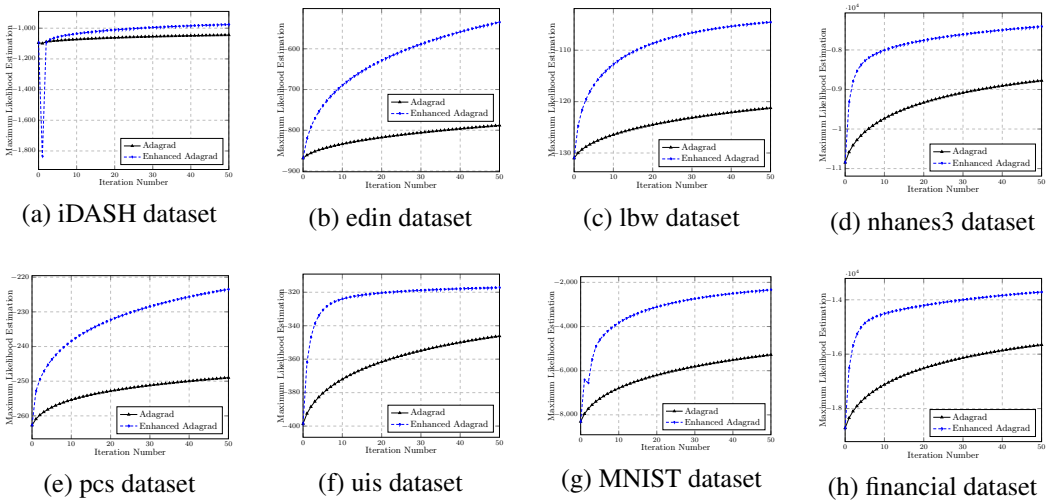
$$\text{(Enhanced)} \quad \boldsymbol{\beta}_{t+1}^{(i)} = \boldsymbol{\beta}_t^{(i)} - \frac{N_t}{\epsilon + \sqrt{\sum_{k=1}^{t} (\mathbf{G}_k^{(i)})^2}} \cdot \mathbf{G}_t^{(i)}.$$

3. **Adam:** Adam integrates the benefits of AdaGrad's element-wise adaptation with the stability of momentum, enabling faster convergence in the presence of sparse or noisy gradients. Following the same logic as the AdaGrad enhancement, the *Enhanced Adam* algorithm replaces the vanilla gradient with its quadratic form and employs the specialized learning rate within its bias-corrected first and second moment estimates.

NAG, AdaGrad, and Adam are representative first-order optimization methods that differ primarily in their strategies for learning-rate adaptation and momentum accumulation. The proposed quadratic gradient framework is orthogonal to these mechanisms, as it modifies the gradient representation itself rather than the optimization strategy. Consequently, the same principle can potentially be extended to a broader class of first-order optimization algorithms by replacing the conventional gradient with its quadratic counterpart while preserving the original update structure.

3.3.1 Evaluation

Setup and Datasets We evaluate the performance of our proposed algorithms in a non-encrypted context using Python on a workstation equipped with an Intel Core G640 CPU (1.60 GHz) and 7.3 GB of RAM. As our primary objective is to accelerate training convergence, we utilize the log-likelihood function $l(\boldsymbol{\beta})$ under Maximum Likelihood Estimation as the primary performance metric.



(a) iDASH dataset (b) edin dataset (c) lbw dataset (d) nhanes3 dataset

(e) pcs dataset (f) uis dataset (g) MNIST dataset (h) financial dataset

Figure 2: Training performance of Adagrad vs. Enhanced Adagrad in the plaintext domain.

A comprehensive evaluation is conducted on six optimization algorithms: NAG, AdaGrad, Adam, and their respective quadratic-gradient enhanced variants (denoted as Enhanced NAG, Enhanced Adagrad, and Enhanced Adam). We test these on the benchmark datasets utilized by (Kim et al., 2018a), which include the iDASH genomic dataset, the Myocardial Infarction dataset (edin), Low Birth Weight Study (lbw), NHANES III (nhanes3), Prostate Cancer Study (pcs), and the Umaru Impact Study (uis). The iDASH dataset, sourced from the 2017 iDASH competition (Task 3), comprises 1,579 records with 103 binary genotypes and a binary phenotype indicating cancer status.

To assess scalability, we further evaluate these algorithms on two large-scale datasets from (Han et al., 2019): a real-world financial dataset containing 422,108 samples and 200 features, and a restructured MNIST dataset featuring 11,982 training samples with 196 features. Throughout our experiments, we consistently employ the fixed Hessian approximation $\bar{\mathbf{H}} = -\frac{1}{4}\mathbf{X}^\top\mathbf{X}$ to construct the diagonal surrogate $\tilde{\mathbf{B}}$.

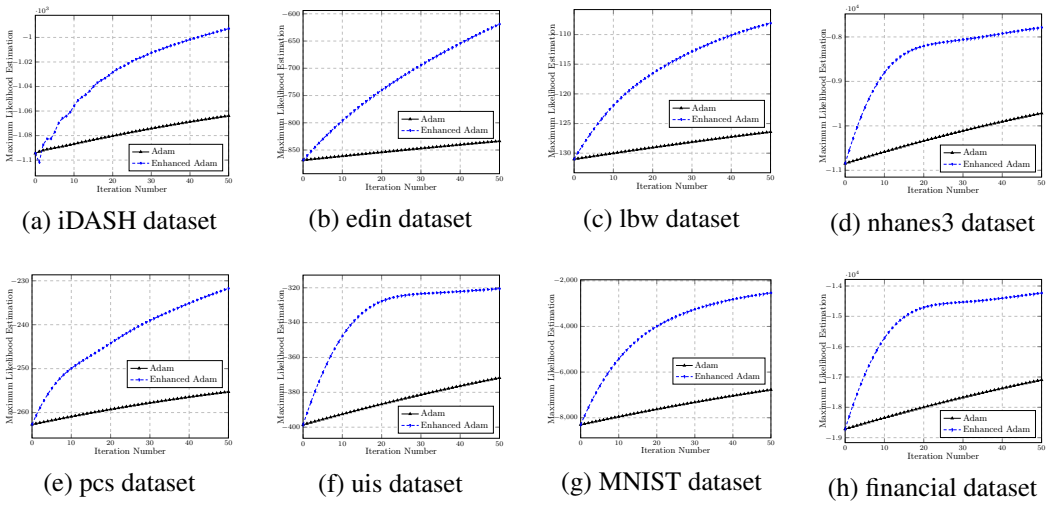
Hyperparameter Configuration To ensure a fair comparison with the baseline NAG (Kim et al., 2018a), Enhanced NAG adopts a learning rate of $1 + \frac{10}{1+t}$, mirroring the decay structure of $\frac{10}{1+t}$ used in the original work. For Enhanced AdaGrad, we employ a modified learning rate of $N_t = 0.06 + 0.01$, compared to the default 0.01 used in standard AdaGrad. Similarly, Enhanced Adam adopts an adjusted configuration with a new augmented learning rate of $\alpha = 0.011$ (calculated as the sum of a base rate 0.01 and the original 0.001), while maintaining the standard momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In contrast, the baseline Adam algorithm follows its default setting of $\alpha = 0.001$ (Kingma & Ba, 2014).

Performance Discussion As demonstrated in Figures 1–3, the proposed enhanced algorithms demonstrate both guaranteed convergence and a superior convergence rate compared to their first-order baselines.

Notably, the learning rates for Enhanced NAG were initially selected to maintain parity with baseline methods rather than for peak optimization. However, the framework demonstrates even more pronounced gains under alternative schedules. For instance, by adopting an exponentially decaying policy ($N_t = 1 + 6 \times 0.9^t$) on the restructured MNIST dataset, Enhanced NAG exhibits a more clearly accelerated convergence trajectory relative to standard NAG.

Overall, the experimental results confirm that the integration of the enhanced gradient framework leads to a systematic improvement in optimization efficiency:

- **Rapid Convergence:** The enhanced variants reach convergence thresholds significantly faster than their original counterparts—an essential attribute for computationally expensive environments like Homomorphic Encryption.



(a) iDASH dataset (b) edin dataset (c) lbw dataset (d) nhanes3 dataset

(e) pcs dataset (f) uis dataset (g) MNIST dataset (h) financial dataset

Figure 3: Plaintext training results of Adam and Enhanced Adam across eight datasets.

- **Consistent Superiority:** Performance gains are consistent across diverse datasets and high-dimensional architectures, confirming the generalizability of the quadratic gradient.
- **Iteration Efficiency:** Near-optimal results are typically achieved within 4 to 5 iterations, providing a critical advantage in resource-constrained or privacy-preserving settings.

Table 1: Performance comparison on the iDASH dataset using 10-fold cross-validation.

Dataset	Sample Num	Feature Num	Method	deg g	Iter Num	Storage (GB)	Learn Time (min)	Accuracy (%)	AUC
iDASH	1579	18	Ours	5	4	0.08	4.43	61.46	0.696
			Kim et al.	5	7	0.04	6.07	62.87	0.689

Table 2: Experimental results on five benchmark datasets under 5-fold cross-validation.

Dataset	Sample Num	Feature Num	Method	deg g	Iter Num	Storage (GB)	Learn Time (min)	Accuracy (%)	AUC
edin	1253	9	Ours	5	4	0.04	0.6	89.52	0.943
			Kim et al.	5	7	0.02	3.6	91.04	0.958
lbw	189	9	Ours	5	4	0.04	0.6	71.35	0.667
			Kim et al.	5	7	0.02	3.3	69.19	0.689
nhanes3	15649	15	Ours	5	4	0.31	4.5	79.23	0.637
			Kim et al.	5	7	0.16	7.3	79.22	0.717
pcs	379	9	Ours	5	4	0.04	0.6	63.20	0.733
			Kim et al.	5	7	0.02	3.5	68.27	0.740
uis	575	8	Ours	5	4	0.04	0.6	74.43	0.597
			Kim et al.	5	7	0.02	3.5	74.44	0.603

4 EXPERIMENTS

For privacy-preserving logistic regression training, we deploy the *Enhanced NAG* algorithm on homomorphically encrypted data via the HEAAN library. The homomorphic evaluations are executed on a cloud-based instance provisioned with 32 vCPUs and 64 GB of RAM.

To ensure a fair comparison with the baseline established by (Kim et al., 2018a), we adopt an identical 10-fold cross-validation (CV) procedure on the iDASH dataset (1,579 samples, 18 features) and 5-fold CV for the remaining five datasets. We report the average accuracy and Area Under the Curve (AUC) as the primary evaluation metrics. The experimental results, including average runtime and storage consumption (comprising the encrypted dataset for the baseline and the dataset plus $\mathsf{ct}_{\bar{\mathbf{B}}}$ for our method), are summarized in Tables 1 and 2. To circumvent the inability of homomorphic encryption to directly evaluate non-linear activations, we approximate the Sigmoid function using the 5th-degree polynomial $g(x)$ over $[-8, 8]$ (Kim et al., 2018a):

$$g(x) = 0.5 + 0.19131 \cdot x - 0.0045963 \cdot x^3 + 0.0000412332 \cdot x^5.$$

To avoid the overhead of inverting $\tilde{\mathbf{B}}$ during training, we offload the pre-calculation of $\bar{\mathbf{B}}$ to the data owner. This preprocessing is computationally efficient and introduces no additional privacy leakage beyond the standard data normalization process. Kim et al. (2018a) packed the training features and label information into a single ciphertext $\mathsf{ct}_{\mathbf{Z}}$. To support SIMD operations, the initial zero-initialized weight vector $\boldsymbol{\beta}^{(0)}$ and the diagonal entries of $\bar{\mathbf{B}}$ are replicated n times into matrix forms. Following this design, we adopt the same packing strategy to ensure structural consistency in ciphertext storage. This results in comparable ciphertext sizes, with the only additional overhead being the ciphertexts required to encode the diagonal approximation matrix $\bar{\mathbf{B}}$. Throughout the training procedures, we employ a dynamic learning rate of $N_t = 1 + 0.9^t$. For the detailed homomorphic gradient calculation procedure, we refer the reader to the methodologies in (Kim et al., 2018a).

The HEAAN parameters are configured to align closely with the security levels in (Kim et al., 2018a): $\log N = 16$, $\log Q = 1200$, $\log p = 30$, and the number of slots set to $32,768$, providing a security level of $\lambda \approx 80$. Notably, we utilize a larger scaling factor, $\log p = 40$, to encrypt the matrix $\bar{\mathbf{B}}$ to preserve the precision of the second-order information.

Due to the increased multiplicative depth—specifically, one additional ciphertext multiplication per iteration compared to the baseline—our scheme consumes the modulus more rapidly. Consequently, without bootstrapping, the evaluation is limited to four iterations of the Enhanced NAG algorithm. Nevertheless, as shown in the results, the proposed method achieves accuracy and AUC levels comparable to the baseline even within this restricted iteration budget.

Limitations and Trade-offs In a privacy-preserving context, the Enhanced NAG method introduces two primary overheads compared to standard NAG: (1) one additional ciphertext-ciphertext multiplication per iteration to compute the quadratic gradient, and (2) the initial transmission of $\mathsf{ct}_{\bar{\mathbf{B}}}$ by the data owner. However, these costs are effectively compensated by the significantly reduced number of total iterations required to reach convergence, leading to a net reduction in the overall homomorphic execution time.

5 CONCLUSION

In this paper, we proposed a novel gradient variant, termed the *quadratic gradient*, and developed an *Enhanced NAG* framework specifically tailored for privacy-preserving logistic regression training in the encrypted domain.

By directly incorporating second-order curvature information from the Hessian matrix, the quadratic gradient effectively bridges the gap between first-order gradient-based methods and second-order Newton-type techniques.

Future research will investigate the extension of the quadratic gradient framework to broader numerical optimization problems, including convex and non-convex settings, to further evaluate its scalability and generality.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, including improving grammar, clarity, and readability of the manuscript.

We have not used generative AI tools for generating research results, conducting experiments, generating scientific claims, or producing the core intellectual content of this work, and other required disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools for language polishing and writing assistance. We have reviewed all AI-assisted work and verified that the resulting text accurately reflects the authors' intended content and does not introduce unsupported claims. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on privacy-preserving machine learning and does not involve human subjects or the collection of personally identifiable information. The datasets used in our experiments are publicly available and are used in accordance with their respective terms of use. Although the proposed methods are designed to improve privacy in machine learning, we do not claim that they provide protection against threats beyond the security assumptions and threat model explicitly described in this work. We are not aware of any additional ethical concerns associated with the methods or experiments presented in this paper.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed methods and experimental settings in the main paper. The implementation details, including model architectures, datasets, data preprocessing procedures, hyperparameters, baseline configurations, and evaluation metrics, are provided in the main paper and supplementary materials. Anonymous source code and additional implementation resources are provided to facilitate independent reproduction of our results: `https://anonymous.4open.science/r/IDASH2017-245B`.

REFERENCES

Paul D. Allison. Convergence failures in logistic regression. 2008. doi: https://doi.org/10.1002/0471475769.ch10.

Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang. Privacy-preserving logistic regression with distributed data sources via homomorphic encryption. *IEICE TRANSACTIONS on Information and Systems*, 99(8):2079–2089, 2016a.

Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang. Scalable and secure logistic regression via homomorphic encryption. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 142–144, 2016b. doi: https://doi.org/10.1145/2857705.2857731.

Michael Artin. *Algebra*. Pearson Prentice Hall, 2011. ISBN 9780132413770. URL `https://books.google.com.ph/books?id=S6GSAgAAQBAJ`.

Flavio Bergamaschi, Shai Halevi, Tzipora T Halevi, and Hamish Hunt. Homomorphic training of 30,000 logistic regression models. In *Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17*, pp. 592–611. Springer, 2019.

Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, Kurt Rohloff, and Vinod Vaikuntanathan. Optimized homomorphic encryption solution for secure genome-wide association studies. *IACR Cryptology ePrint Archive*, 2019:223, 2019. doi: https://doi.org/10.1186/s12920-020-0719-9.

Dankmar Böhning and Bruce G Lindsay. Monotonicity of quadratic-approximation algorithms. *Annals of the Institute of Statistical Mathematics*, 40(4):641–663, 1988. doi: https://doi.org/10.1007/BF00049423.

Charlotte Bonte and Frederik Vercauteren. Privacy-preserving logistic regression training. *BMC medical genomics*, 11(4):86, 2018. doi: https://doi.org/10.1186/s12920-018-0398-y.

Junyoung Byun, Woojin Lee, and Jaewook Lee. Parameter-free he-friendly logistic regression. *Advances in Neural Information Processing Systems*, 34:8457–8468, 2021.

Sergiu Carpov, Nicolas Gama, Mariya Georgieva, and Juan Ramon Troncoso-Pastoriza. Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption. *BMC medical genomics*, 13(Suppl 7):88, 2020.

Hao Chen, Ran Gilad-Bachrach, Kyoohyung Han, Zhicong Huang, Amir Jalali, Kim Laine, and Kristin Lauter. Logistic regression over encrypted data from fully homomorphic encryption. *BMC medical genomics*, 11(4):3–12, 2018. doi: https://doi.org/10.1186/s12920-018-0397-z.

Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 409–437. Springer, 2017. doi: https://doi.org/10.1007/978-3-319-70694-8_15.

Jack LH Crawford, Craig Gentry, Shai Halevi, Daniel Platt, and Victor Shoup. Doing real work with fhe: the case of logistic regression. In *Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, pp. 1–12, 2018. doi: https://doi.org/10.1145/3267973.3267974.

Eric Crockett. A low-depth homomorphic circuit for logistic regression model training. *Cryptology ePrint Archive*, 2020.

Martine De Cock, Rafael Dowsley, Anderson Nascimento, Davis Railsback, Jianwei Shen, and Ariel Todoki. Fast secure logistic regression for high dimensional gene data. In *Privacy in machine learning (PriML2019). Workshop at NeurIPS*, pp. 1–7, 2019.

Martine De Cock, Rafael Dowsley, Anderson CA Nascimento, Davis Railsback, Jianwei Shen, and Ariel Todoki. High performance logistic regression for privacy-preserving genome analysis. *BMC Medical Genomics*, 14(1):23, 2021.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178, 2009. doi: https://doi.org/10.1145/1536414.1536440.

Kyoohyung Han, Seungwan Hong, Jung Hee Cheon, and Daejun Park. Logistic regression on homomorphic encrypted data at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9466–9471, 2019. doi: https://doi.org/10.1609/aaai.v33i01.33019466.

Kyoohyung Han, Jinhyuck Jeong, Jung Hoon Sohn, and Yongha Son. Efficient privacy preserving logistic regression inference and training. *Cryptology ePrint Archive*, 2020.

Seungwan Hong, Yoolim A Choi, Daniel S Joo, and Gamze Gürsoy. Privacy-preserving model evaluation for logistic and linear regression using homomorphically encrypted genotype data. *Journal of biomedical informatics*, 156:104678, 2024.

Angela Jäschke and Frederik Armknecht. Accelerating homomorphic computations on rational numbers. In *International Conference on Applied Cryptography and Network Security*, pp. 405–423. Springer, 2016. doi: https://doi.org/10.1007/978-3-319-39555-5_22.

Andrey Kim, Yongsoo Song, Miran Kim, Keewoo Lee, and Jung Hee Cheon. Logistic regression model training based on the approximate homomorphic encryption. *BMC medical genomics*, 11 (4):83, 2018a. doi: https://doi.org/10.1186/s12920-018-0401-7.

Miran Kim, Yongsoo Song, Shuang Wang, Yuhou Xia, and Xiaoqian Jiang. Secure logistic regression based on homomorphic encryption: Design and evaluation. *JMIR medical informatics*, 6(2): e19, 2018b. doi: https://doi.org/10.2196/medinform.8805.

Miran Kim, Junghye Lee, Lucila Ohno-Machado, and Xiaoqian Jiang. Secure and differentially private logistic regression for horizontally distributed data. *IEEE Transactions on Information Forensics and Security*, 15:695–710, 2019a.

Miran Kim, Yongsoo Song, Baiyu Li, and Daniele Micciancio. Semi-parallel logistic regression for gwas on encrypted data. *IACR Cryptology ePrint Archive*, 2019:294, 2019b. doi: https://doi.org/10.1186/s12920-020-0724-z.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv e-prints*, pp. arXiv–1412, 2014.

Junyi Li and Heng Huang. Faster secure data mining via distributed homomorphic encryption. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2706–2714, 2020.

Luis Montero, Jordan Frery, Celia Kherfallah, Roman Bredehoft, and Andrei Stoian. Machine learning training on encrypted data with tfhe. In *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*, pp. 71–76, 2024.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge, MA, 2012. URL `https://books.google.co.jp/books?id=NZP6AQAAQBAJ`.

Tabitha Ogilvie, Rachel Player, and Joe Rowell. Improved privacy-preserving training using fixed-hessian minimisation. *Cryptology ePrint Archive*, 2020.

Nigel P Smart and Frederik Vercauteren. Fully homomorphic simd operations. *Designs, codes and cryptography*, 71(1):57–81, 2014.