Privacy-Preserving Collaborative Learning With Linear Communication Complexity

Xingyu Lu, Hasin Us Sami, *Graduate Student Member, IEEE*, and Başak Güler, *Member, IEEE*

***Abstract*— Collaborative machine learning enables privacy-preserving training of machine learning models without collecting sensitive client data. Despite recent breakthroughs, communication bottleneck is still a major challenge against its scalability to larger networks. To address this challenge, in this work we propose PICO, the first collaborative learning framework with linear communication complexity, significantly improving over the quadratic state-of-the-art, under formal information-theoretic privacy guarantees. Theoretical analysis demonstrates that PICO slashes the communication cost while achieving equal computational complexity, adversary resilience, robustness to client dropouts, and model accuracy to the state-of-the-art. Extensive experiments demonstrate up to 91× reduction in the communication overhead, and up to 8× speed-up in the wall-clock training time compared to the state-of-the-art. As such, PICO addresses a key technical challenge in multi-party collaborative learning, paving the way for future large-scale privacy-preserving learning frameworks.**

***Index Terms*— Coded computing, distributed training, collaborative machine learning, information-theoretic privacy.**

I. Introduction

PRIVACY-PRESERVING collaborative machine learning (PPML) allows multiple data owners to collaborate to train ML models without sharing their data. PPML can greatly improve ML performance by increasing the volume and diversity of data, without compromising privacy [2], [3]. It can even foster novel applications in which data is rare and collaboration has traditionally been limited due to privacy concerns, such as the treatment of rare diseases [4], [5].

Recently, coding-theoretic approaches have shown promising performance gains in the design of PPML [6], [7], [8]. This approach, known as *Lagrange Coded Computing (LCC)*, *encodes* the local datasets using a Lagrange interpolation polynomial, prior to training. The encoding operation injects randomness and (computational) redundancy within the local computations, to provide strong information-theoretic privacy guarantees and resilience to client dropouts, while also reducing the training load per client. Training is then performed on the encoded data, *as if they were performed on the clear data*. After multiple training rounds, the final model is *decoded* using polynomial interpolation, by collecting the computations (performed over encoded data) from individual clients. By doing so, an order-of-magnitude speed-up can be achieved in the training time compared to state-of-the-art cryptographic baselines, where for the latter the training load per client is as large as centralized training (over the collection of all client datasets) [7].

The major challenge against the scalability of information-theoretic PPML is the *communication complexity*, which is quadratic in the number of clients. This is caused by the multiplication operations associated with gradient computations. Specifically, interpolating a polynomial f of degree $\deg(f)$ requires collecting at least $\deg(f)+1$ interpolation points. As such, decoding the final model from the local computations requires computations to be collected from at least $N \geq \deg(f)+1$ clients. On the other hand, the multiplication operations during gradient computations lead to an exponential growth in the polynomial degree, leading to a *degree explosion* after a few training rounds. This necessitates an expensive *degree reduction step* with a quadratic communication overhead (after each round), preventing scalability to large networks.

To address this challenge, in this work we propose PICO,[1] the first information-theoretic PPML framework with linear communication complexity. Our focus is on logistic regression, a widely used machine learning mechanism due to its practicality and interpretability [9]. Although logistic regression has a long history in PPML dating back to [10], [11], and [12], enabling communication-efficient and scalable mechanisms for large-scale networks is still an open problem. The key intuition behind PICO is an online-offline communication trade-off combined with an efficient offline randomness generation mechanism. In particular, we first trade-off expensive online (data-dependent) communications with offline (data-agnostic) communications. The online phase trades-off the quadratic point-to-point communication overhead with a broadcast mechanism with linear overhead. Our key contribution is a coded efficient randomness generation mechanism for the offline phase. In particular, we then develop

Manuscript received 6 March 2023; revised 23 September 2023; accepted 13 December 2023. Date of publication 19 December 2023; date of current version 16 July 2024. This work was supported in part by OUSD (R&E)/RT&L under Grant W911NF-20-2-0267, in part by NSF CAREER Award CCF-2144927, and in part by the UCR Opportunity to Advance Sustainability Innovation and Social Inclusion (OASIS) Funding Award. An earlier version of this paper was presented in part at the 26th International Conference on Artificial Intelligence and Statistics (AISTATS'23) [1]. *(Corresponding author: Başak Güler.)*

The authors are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: xlu065@ucr.edu; hsami003@ucr.edu; bguler@ece.ucr.edu).

Communicated by W. Bajwa, Associate Editor for Machine Learning and Statistics and Signal Processing and Source Coding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3345270.

Digital Object Identifier 10.1109/TIT.2023.3345270

[1] PICO stands for privacy-preserving collaborative learning.

a coded layered randomness generation mechanism for the offline phase, that builds on MDS (Maximum Distance Separable) matrices (also related to hyperinvertible matrices [13]) and Lagrange codes, and reduces the quadratic offline communication overhead to linear, by reducing the *volume* of variables communicated by each client; communicating each variable has a quadratic cost, but the *total number of variables* scales inversely with the number of clients, leading to a linear amortized overhead. As such, in a network of N clients, PICO incurs an $O(N)$ communication complexity both offline and online, as opposed to the $O(N^2)$ online communication complexity of the state-of-the-art. A major contribution of our work is ensuring equal adversary-tolerance, dropout-resilience to the state-of-the-art, and computational complexity, while reducing the communication overhead.

Our theoretical analysis provides formal guarantees for information-theoretic privacy, correctness, and key performance trade-offs in terms of the communication and computation complexity, adversary resilience, client dropouts, and training time. We perform extensive experiments to evaluate the performance of PICO, by implementing a distributed multi-client network for various image classification tasks. We then demonstrate the communication/computation volume and the wall-clock training time of PICO with respect to state-of-the-art benchmarks, identify the impact of key system parameters and trade-offs, and present the model convergence and accuracy.

Our contributions can be summarized as follows:

- We introduce PICO, the first privacy-preserving collaborative learning framework with linear communication complexity (both online and offline), under strong end-to-end information-theoretic privacy guarantees.
- We demonstrate a novel offline (data-agnostic) coded randomness generation mechanism for privacy-preserving logistic regression, which can reduce the amortized communication complexity to linear in the number of users.
- Our theoretical analysis presents formal information-theoretic privacy guarantees (for end-to-end training), and shows that PICO cuts the communication overhead while achieving the same computation complexity, adversary resilience, robustness to client dropouts, and model accuracy of the state-of-the-art.
- Our experiments demonstrate up to 91× reduction in the communication overhead, and up to 8× speed-up in the wall-clock training time compared to the state-of-the-art, while achieving the same adversary and dropout resilience, and model accuracy.

II. Related Work

In addition to coded computing, there are several other techniques that are commonly employed for PPML. A popular approach is Secure Multi-Party Computing (MPC) [13], [14], [15], [16], which allows parties to compute a function over their inputs without revealing their inputs in the clear [10], [17], [18], [19], [20]. Secure MPC protocols often rely on a cryptographic primitive known as *secret sharing*, where clients locally add local randomness to their datasets prior sharing them with others [21]. Then, training is carried out using the secret shared datasets (as opposed to the true datasets). The injected randomness is reversible, i.e., parties can decode the computations performed on the secret shared data to recover the true computation results, preserving model accuracy. Secure MPC can provide strong information-theoretic privacy guarantees, such that no information about the datasets is revealed beyond the final model (even if adversaries have unbounded computational power) [2]. The major challenge is the extensive communication required to perform secure computations between the parties, which limits scalability in larger networks.

In addition to the secret sharing-based mechanisms, there are notable MPC mechanisms that are not based on secret sharing, including the well-known Yao's garbled circuits [22] and its modern variants [23], [24], [25], [26], [27], [28], [29], [30]. Recent works also consider computationally secure MPC mechanisms by utilizing homomorphic encryption principles [11], [31], [32], [33], [34], [35], [36]. Combining secure MPC with homomorphic encryption can further trade-off the communication and computation complexity of MPC protocols, as communication is a major bottleneck in large-scale applications [37]. For a comparative study of modern MPC frameworks, including the benefits and trade-offs of hybrid and mixed-protocol mechanisms, we refer to [38]. Recently, MPC mechanisms have also been used for aggregating the local user updates (e.g., local models or gradients) in distributed and federated learning, which is known as *secure aggregation*, where parties learn the sum of client models/gradients after each (global) training round, but without observing the individual models/gradients [39], [40], [41], [42]. In contrast, our focus in this work is on *end-to-end* PPML, where parties can learn only the *final model* (after multiple training rounds), and no intermediate model/gradient should be revealed during training.

Homomorphic encryption (HE) mechanisms enable the execution of computations on encrypted data in scenarios where adversaries possess limited computational capabilities [12], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. Such mechanisms can withstand a larger number of adversaries, surpassing what secure MPC protocols can handle. However, the level of privacy hinges on the size of the encrypted data; stronger guarantees require larger encrypted data sizes (in contrast to MPC, where the size of the secret shared data remains consistent), consequently increasing the computational overhead for the clients. As a result, HE finds more common use in the inference stage of machine learning tasks, as opposed to the more computationally intensive training phase.

Finally, differential privacy (DP) mechanisms protect the privacy of local datasets by injecting noise to local computations during training. By doing so, DP prevents information leakage from the final released model also, as opposed to secure MPC and HE protocols where the final model is released as is [53], [54], [55], [56], [57], [58], [59], and [60]. The privacy guarantees are controlled by the level of noise introduced during training, leading to an accuracy-privacy trade-off. The main challenge in distributed settings is the accumulation of noise as the number of users grow, which degrades models accuracy. To address this, DP mechanisms are recently combined with secure MPC protocols, which can improve model accuracy by reducing the amount of noise introduced by each client [61], [62], [63]. While beyond our current scope, we note that our methods can also be integrated with DP as an interesting future direction.

Notation. In the following, x is a scalar, $\mathbf{x}$ is a vector, and $\mathbf{X}$ is a matrix. A set is represented by $\mathcal{X}$ with cardinality $|\mathcal{X}|$. $\operatorname{tr}(\mathbf{X})$ denotes the trace of matrix $\mathbf{X}$, whereas $\mathbf{X}^{\mathrm{T}}$ is the matrix transpose, and $\otimes$ denotes the Kronecker product. $[N]$ represents the set $\{1,\dots,N\}$, and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. Finally, $[x]_i$ denotes a share of a secret x at client $i \in [N]$. All secret shares are generated by using Shamir's T-out-of-N Secret Sharing (SSS), which embeds the secret in a degree T random polynomial, such that the secret can be reconstructed from any set of $T+1$ shares, but any set of at most T shares reveals no information about the secret. For the details, we refer to [21]. The remainder of the paper is organized as follows. Section III provides the system model, whereas Section IV presents the potential approaches, limitations, and main results. Section V introduces our framework PICO, whereas Section VI provides a motivating example. Section VII presents the theoretical results, and Section VIII demonstrates the experiments. Section IX concludes the paper.

III. Problem Formulation

In this work, our focus is on collaborative logistic regression with N clients. Client i holds a local dataset $\mathbf{X}_i \in \mathbb{R}^{m_i \times d}$ consisting of m_i data points (where each data point has d features), along with the corresponding labels $\mathbf{y}_i \in \{0,1\}^{m_i}$. The collection of all local datasets is represented by a matrix $\mathbf{X} = \begin{bmatrix}\mathbf{X}_1^{\mathrm{T}} & \dots & \mathbf{X}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{\overline{m} \times d}$ consisting of $\overline{m} \triangleq \sum_{i=1}^{N} m_i$ data points, along with the corresponding labels $\mathbf{y} = \begin{bmatrix}\mathbf{y}_1^{\mathrm{T}} & \dots & \mathbf{y}_N^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \{0,1\}^{\overline{m} \times 1}$. The goal is to train a logistic regression model $\mathbf{w}$ jointly over the collective dataset $\mathbf{X}$, by minimizing a binary cross entropy loss function:

$$\mathcal{L}(\mathbf{w})=\frac{1}{\overline{m}}\sum_{i=1}^{\overline{m}}\Big(-y_i \log g(\mathbf{x}_i \times \mathbf{w}) - (1-y_i)\log(1-g(\mathbf{x}_i \times \mathbf{w}))\Big) \tag{1}$$

where $g(\mathbf{x}_i \times \mathbf{w}) \triangleq 1/(1+e^{-\mathbf{x}_i \times \mathbf{w}}) \in (0,1)$ denotes the sigmoid function, which quantifies the probability of label i being equal to 1, and $\mathbf{x}_i \in \mathbb{R}^{1 \times d}$ denotes the i^{th} row of $\mathbf{X}$ (features of data point i). The model is then trained via gradient descent,

$$\begin{aligned}\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} - \frac{\eta}{\overline{m}}\mathbf{X}^{T}(g(\mathbf{X} \times \mathbf{w}^{(t)}) - \mathbf{y}) \\ &= \mathbf{w}^{(t)} - \frac{\eta}{\overline{m}}\sum_{i=1}^{\overline{m}} \mathbf{x}_i^{\mathrm{T}}(g(\mathbf{x}_i \times \mathbf{w}^{(t)}) - y_i) \qquad (2)\end{aligned}$$

where $\mathbf{w}^{(t)}$ is the estimated model parameters at training round t, η is the learning rate, and function $g(\cdot)$ is applied element-wise. We consider a decentralized communication topology, where clients can communicate through point-to-point unicast or (one-to-many) broadcast links. At each training round, up to D clients may drop out from the system due to various reasons such as poor connectivity or device unavailability. We do not assume the existence of a trusted third party or a central coordinator. Our system model is presented in Fig. 1.

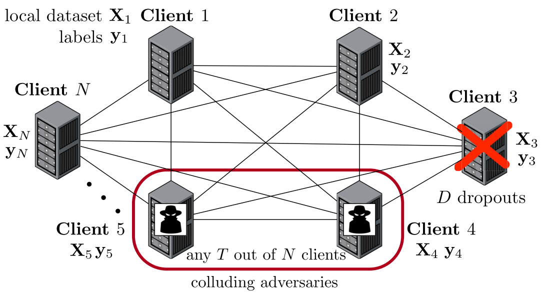


Fig. 1. **System model.** The multi-client learning setup of PICO. Client $i \in [N]$ holds a dataset $\mathbf{X}_i$ with labels $\mathbf{y}_i$. Any set of up to T out of N clients may be adversarial. Adversaries may collude with each other.

Remark 1: The binary cross entropy loss (also known as the logistic loss), which fits the model parameters $\mathbf{w}$ through a maximum likelihood principle, where minimizing the loss function $\mathcal{L}(\mathbf{w})$ corresponds to maximizing the conditional likelihood of the labels given the features [9, Section 4.4.1], is a widely used loss function in practice [64]. For the binary classification task (to predict one of two classes 0 or 1), this can be viewed as a convex surrogate of the $0-1$ loss (to minimize the number of misclassifications) [65], [66], which is NP-hard to optimize directly [67], [68]. Depending on the problem characteristics, alternative loss functions can also be considered for different tasks, which is an interesting future direction [69], [70].

Threat Model. The most common adversary model in PPML is honest-but-curious adversaries, which is also the focus of this work [2]. In this setup, adversaries follow the protocol truthfully (i.e., do not poison the datasets/messages), but may attempt to reveal sensitive local datasets of honest clients using the messages exchanged. Out of N clients, any set of up to T clients can be adversarial, who may collude with each other. The adversaries are unknown to the honest clients. The set of adversarial and honest clients are denoted by $\mathcal{T}$ and $\mathcal{H} = [N]\backslash\{T\}$, respectively.

Information-Theoretic Privacy. Our focus is on information-theoretic privacy, where the goal is to ensure that the adversaries learn no information about the local datasets of honest clients, beyond the final model [2]. Similar to former works, our framework is bound to finite field operations, and in the following we assume that all datasets and labels are represented in a finite field $\mathbb{F}_q$ of integers modulo a large prime q. For the details of this finite field transformation (which is handled via a quantization mechanism), we refer to [2], [7], [8], [39], and [40]. In the following, we let $\overline{\mathbf{X}}_i \in \mathbb{F}_q^{m_i \times d}$ and $\overline{\mathbf{y}}_i \in \mathbb{F}_q^{m_i \times 1}$ denote the finite field representation of $\mathbf{X}_i \in \mathbb{R}^{m_i \times d}$ and $\mathbf{y}_i \in \mathbb{R}^{m_i \times 1}$, respectively. Similarly, $\overline{\mathbf{X}} \in \mathbb{F}_q^{\overline{m} \times d}$ and $\overline{\mathbf{y}} \in \mathbb{F}_q^{\overline{m} \times 1}$ denotes the finite field representation of $\mathbf{X} \in \mathbb{R}^{\overline{m} \times d}$ and $\mathbf{y} \in \mathbb{R}^{\overline{m} \times 1}$. All training computations are then carried out within $\mathbb{F}_q$. The model parameters are updated in the finite field throughout the training, and are converted to the real domain only at the end of training. We let $\overline{\mathbf{w}}^{(t)}$ denote the finite field model parameters at round t. At the end of training (after J rounds), the final model $\overline{\mathbf{w}}^{(J)}$ is decoded in the finite field, and then converted to the real domain $\mathbf{w}^{(J)}$. Accordingly, the Markov relation,

$$\{\mathbf{X}_i, \mathbf{y}_i\}_{i\in[N]} - \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]} - \overline{\mathbf{w}}^{(J)} - \mathbf{w}^{(J)}$$

holds between the finite field and real domain representations, hence from the data processing inequality (DPI) [71], $\mathbf{w}^{(J)}$ does not carry any further information about the local datasets than $\overline{\mathbf{w}}^{(J)}$. Then, the information-theoretic privacy condition can be formally stated as,

$$I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]\setminus\mathcal{T}}; \mathcal{M}_{\mathcal{T}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) = 0 \tag{3}$$

for all $\mathcal{T}$ such that $|\mathcal{T}| \leq T$, where $\mathcal{M}_{\mathcal{T}}$ is the collection of all messages received or generated by the adversaries, and J is the total the number of training rounds.

Main Problem. In this work, our goal is to solve (1) with the information-theoretic guarantees from (3), We then ask the following question:

- *How can we develop a scalable PPML framework to solve (1) with linear total communication complexity, under the formal information-theoretic guarantees from (3)?*

We next review the potential approaches and challenges to address this challenge, and introduce our main results.

IV. Potential Approaches, Challenges, and Main Results

A. COPML (Coded Private Machine Learning)

To solve (1) with the end-to-end information-theoretic guarantees from (3), the state-of-the-art is the COPML framework from [7], which leverages Shamir's Secret Sharing (SSS) [21] to encode the datasets and model. For dataset encoding, each client $i \in [N]$ secret shares its local dataset $\overline{\mathbf{X}}_i$ using SSS, and sends a secret share $[\overline{\mathbf{X}}_i]_j$ to client $j \in [N]$. Client j concatenates the received shares and partitions them into K equal-sized shards $\left[[\overline{\mathbf{X}}_1]_j^{\mathrm{T}} \;\; \dots \;\; [\overline{\mathbf{X}}_N]_j^{\mathrm{T}}\right]^{\mathrm{T}} = \left[[\overline{\mathbf{X}}'_1]_j^{\mathrm{T}} \;\; \dots \;\; [\overline{\mathbf{X}}'_K]_j^{\mathrm{T}}\right]^{\mathrm{T}}$, then sends an encoded matrix,

$$\begin{aligned}[f(\alpha_i)]_j = &\sum_{k\in[K]} [\overline{\mathbf{X}}'_k]_j \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l} \\ &+ \sum_{k=K+1}^{K+T} [\mathbf{R}_k]_j \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l}\end{aligned} \tag{4}$$

to client $i \in [N]$, where $\{[\mathbf{R}_k]_j\}_{k\in\{K+1,\dots,K+T\}} \in \mathbb{F}_q^{\overline{m}/K}$ are uniformly random matrices secret shared by a crypto-service provider. After receiving $\{[f(\alpha_i)]_j\}_{j\in[N]}$, client i recovers the encoded dataset $\widetilde{\mathbf{X}}_i = f(\alpha_i)$ using polynomial interpolation. For model encoding, at each training round t, client $j \in [N]$, who holds a secret share $[\overline{\mathbf{w}}^{(t)}]_j$ of the model $\overline{\mathbf{w}}^{(t)}$ (without learning its true value), sends an encoded matrix,

$$\begin{aligned}[h(\alpha_i)]_j = &\sum_{k\in[K]} [\overline{\mathbf{w}}^{(t)}]_j \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l} \\ &+ \sum_{k=K+1}^{K+T} [\mathbf{v}_k^{(t)}]_i \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l}\end{aligned} \tag{5}$$

to client $i \in [N]$, where $\{[\mathbf{v}_k^{(t)}]_i\}_{k\in\{K+1,\dots,K+T\}} \in \mathbb{F}_q^{d\times 1}$ are uniformly random matrices secret shared by a crypto-service provider. After receiving $\{[h(\alpha_i)]_j\}_{j\in[N]}$, client i recovers the encoded model $\widetilde{\mathbf{w}}_i^{(t)} = h(\alpha_i)$ using polynomial interpolation. Training is then performed using the encoded datasets and model. The total online communication overhead is quadratic $O(N^2 d)$ across the N clients. Importantly, the polynomial degree $\deg h$ grows after each multiplication operation. To prevent a degree explosion, a *degree reduction step* has to be carried out after each training round, also with a quadratic overhead, limiting scalability to larger networks.

B. Naive Offline-Online Communication Offloading

To address the communication overhead, a potential approach is to offload the communication-intensive tasks (e.g., model encoding) to a data-independent offline phase [72], [73]. To do so, prior to training (offline), each client $i \in [N]$ can locally generate a uniformly random mask $\mathbf{r}_i^{(t)} \in \mathbb{F}_q^d$ and send to client $j \in [N]$: 1) a secret share $[\mathbf{r}_i^{(t)}]_j \in \mathbb{F}_q^d$ (e.g., using SSS), 2) an encoded mask,

$$\begin{aligned}\widetilde{\mathbf{r}}_{ij}^{(t)} = &\sum_{k\in[K]} \mathbf{r}_i^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \\ &+ \sum_{k=K+1}^{K+T} \mathbf{v}_{ik}^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l},\end{aligned} \tag{6}$$

where $\{\mathbf{v}_{ik}^{(t)}\}_{k\in\{K+1,\dots,K+T\}} \in \mathbb{F}_q^d$ are generated uniformly at random, using which client j can obtain: 1) a secret share $[\mathbf{r}^{(t)}]_j = \sum_{i\in[N]} [\mathbf{r}_i^{(t)}]_j$, and 2) an encoded mask,

$$\begin{aligned}\widetilde{\mathbf{r}}_j^{(t)} &= \sum_{i\in[N]} \widetilde{\mathbf{r}}_{ij}^{(t)} \\ &= \sum_{k\in[K]} \mathbf{r}^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \\ &\quad + \sum_{k=K+1}^{K+T} \Big(\sum_{i\in[N]} \mathbf{v}_{ik}^{(t)}\Big) \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l},\end{aligned} \tag{7}$$

of a common random mask $\mathbf{r}^{(t)} = \sum_{i\in[N]} \mathbf{r}_i^{(t)}$ shared across all users (in encoded form), without learning its true value. The common randomness $\mathbf{r}^{(t)}$ encoded by the T random vectors $\{\mathbf{v}_{ik}^{(t)}\}_{k\in\{K+1,\dots,K+T\}}$ allows clients to use broadcasting in the online phase, to reduce the communication overhead of model encoding from point-to-point quadratic to linear broadcast. To do so, client $j \in [N]$ can broadcast a secret share $[\overline{\mathbf{w}}^{(t)}]_j - [\mathbf{r}^{(t)}]_j = [\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}]_j$ of the masked model $\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}$, where the true model $\overline{\mathbf{w}}^{(t)}$ is hidden by the random mask $\mathbf{r}^{(t)}$. Using the received shares, each client $i \in [N]$ can decode $\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}$ using polynomial interpolation, and locally generate an encoded model $\widetilde{\mathbf{w}}_i^{(t)} = \widetilde{\mathbf{r}}_i^{(t)} + (\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}) \sum_{k\in[K]} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l}$. This reduces the online communication overhead from quadratic $O(N^2 d)$ point-to-point unicast, to linear $O(Nd)$ one-to-many broadcast. On the other hand, the offline communication overhead is still quadratic $O(N^2 d)$ point-to-point.

C. This Work

In this work, we introduce PICO to solve (1) with the end-to-end information-theoretic guarantees from (3). In contrast to naive offline-online communication offloading, PICO achieves *linear* communication overhead both *offline* and *online*. This is achieved by a coded randomness generation mechanism using MDS codes to reduce the total number of variables communicated in the offline phase. Specifically, in the offline phase, each client $i \in [N]$ first generates a lower-dimensional random mask $\mathbf{r}_i^{(t)} \in \mathbb{F}_q^{\frac{d}{N-T}}$ uniformly at random, where the local mask size is reduced to $\frac{d}{N-T}$ from d. Then, client i sends to each client $j \in [N]$: 1) a secret share $[\mathbf{r}_i^{(t)}]_j \in \mathbb{F}_q^{\frac{d}{N-T}}$, 2) an encoded mask $\widetilde{\mathbf{r}}_{ij}^{(t)} \in \mathbb{F}_q^{\frac{d}{N-T}}$ as described in (7), however, all coded masks communicated with the other clients are now of dimension $\frac{d}{N-T}$ as opposed to d. Using the lower-dimensional coded random masks $\{\widetilde{\mathbf{r}}_{ij}^{(t)}, [\mathbf{r}_i]_j^{(t)}\}_{i\in[N]} \in \mathbb{F}_q^{\frac{d}{N-T}}$, client j then locally generates a large-dimensional encoded mask $\widetilde{\mathbf{r}}_j^{(t)} \in \mathbb{F}_q^d$ of size d,

$$\begin{aligned}\widetilde{\mathbf{r}}_j^{(t)} &\triangleq (\mathbf{M}\otimes\mathbf{I}) \times \begin{bmatrix}\widetilde{\mathbf{r}}_{1j}^{(t)}\\ \vdots \\ \widetilde{\mathbf{r}}_{Nj}^{(t)}\end{bmatrix} \\ &= \sum_{k\in[K]} \mathbf{r}^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l} \\ &\quad + \sum_{k=K+1}^{K+T} (\mathbf{M}\otimes\mathbf{I}) \times \begin{bmatrix}\mathbf{v}_1^{(t)}\\ \vdots\\ \mathbf{v}_N^{(t)}\end{bmatrix} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l}, \quad (8)\end{aligned}$$

and a secret share $[\mathbf{r}]_j^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \Big[[\mathbf{r}_1^{(t)}]_j^{\mathrm{T}} \;\; \cdots \;\; [\mathbf{r}_N^{(t)}]_j^{\mathrm{T}}\Big]^{\mathrm{T}} \in \mathbb{F}_q^d$, corresponding to a common random mask $\mathbf{r}^{(t)} \triangleq (\mathbf{M} \otimes \mathbf{I}) \times \Big[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \;\; \cdots \;\; (\mathbf{r}_N^{(t)})^{\mathrm{T}}\Big]^{\mathrm{T}} \in \mathbb{F}_q^d$ of size d, whose true value is unknown by the clients, $\mathbf{I}$ is a $\frac{d}{(N-T)K} \times \frac{d}{(N-T)K}$ identity matrix, and $\mathbf{M}$ is an $(N-T) \times N$ MDS matrix, as will be detailed later. The key intuition is that, while the communication overhead for each variable is quadratic $O(N^2)$ point-to-point (unicast), the *total number of coded variables* to be communicated is reduced to $O(\frac{d}{N-T})$, which is *inversely proportional to the number of clients*. Hence, the overall amortized communication overhead is $O(\frac{dN^2}{N-T})$ point-to-point, which is linear $O(dN)$ for any $T = O(N)$. In the online phase, the offline encoded masks $[\mathbf{r}^{(t)}]_j, \widetilde{\mathbf{r}}_j^{(t)}$ allows client j to broadcast the secret share $[\overline{\mathbf{w}}^{(t)}]_j - [\mathbf{r}^{(t)}]_j = [\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}]_j$ of the masked model $\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}$, using which clients can decode the masked model through polynomial interpolation, and client j can obtain the encoded model $\widetilde{\mathbf{w}}_j^{(t)} = \widetilde{\mathbf{r}}_j^{(t)} + (\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}) \sum_{k\in[K]} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l}$. As a result, communication complexity of the online phase is reduced from $O(N^2)$ point-to-point unicast to $O(N)$ one-to-many broadcast. While reducing the communication overhead, PICO achieves *equal dropout-resilience, adversary-tolerance, and computation complexity to COPML*. In doing so, a reliable broadcasting mechanism is considered [74], which can be achieved through various approaches in practice, such as using an inherently broadcast medium such as cellular networks or satellite links, or through leveraging broadcasting mechanisms at the hardware level, e.g., IP multicast for local area networks.

V. The PICO Framework

We next describe the details of our framework, which consists of five main components:

1) *Dataset encoding:* Clients $i \in [N]$ encode their local datasets $\{\overline{\mathbf{X}}_i\}_{i\in[N]}$ to preserve their privacy while distributing the computation load across the clients. At the end, client $i \in [N]$ learns an *encoded dataset* $\widetilde{\mathbf{X}}_i$, whose size is $(1/K)^{th}$ of the original dataset $\overline{\mathbf{X}}$.
2) *Label encoding:* To preserve the privacy of labels, clients also encode their local labels using locally generated random masks. At the end, each client learns an encoded label.
3) *Model initialization:* To prevent information leakage from intermediate training computations, the model $\overline{\mathbf{w}}^{(0)}$ at round $t = 0$ is initialized uniformly random within $\mathbb{F}_q$, but without revealing its true value to any client (and any collusions between up to T clients).
4) *Model encoding:* To prevent information leakage from intermediate model parameters, the model at each round should be kept private from the clients. To that end, at each training round t, client $i \in [N]$ holds a *secret share* $[\overline{\mathbf{w}}^{(t)}]_i$ (as opposed to the true model) of the current state of the model $\overline{\mathbf{w}}^{(t)}$, using which the clients encode the model, to enable training computations to be performed on the encoded datasets. At the end, client $i \in [N]$ obtains an encoded model $\widetilde{\mathbf{w}}_i^{(t)}$, without learning any information about the true model $\overline{\mathbf{w}}^{(t)}$.
5) *Gradient computing and model update:* Using the encoded datasets and model, clients compute the gradient and update the model for the next training round, but without learning the true value of the gradient or the updated model. In doing so, the key ingredient is a novel degree reduction mechanism with linear communication cost, which reduces the degree of the polynomial corresponding to the gradients computed on the encoded datasets and model, to prevent an exponential growth as the number of training rounds increase.

Table I presents the communication overhead of each component of PICO and COPML [7]. The individual components of PICO comprise of online and offline phases as demonstrated in Fig. 2. We now describe the details of each component. For ease of presentation, we describe the offline and online phases sequentially, to show how the variables generated in the former are utilized in the latter. We note that each offline phase is independent from past online/offline phases, hence all offline phases can be executed in parallel.

A. Dataset Encoding

Initially, clients encode their datasets using locally generated randomness. The goal of the encoding process is two-fold. First, it hides the dataset contents against adversaries. Second, it reduces the size of the data each client should process during training. The encoding process consists of the following offline and online phases.

TABLE I

Comparison of the Total Communication Overhead (Across All Clients) for PICO (Including Both Online and Offline Phases), and COPML (online), Where $K = \Theta(N)$, $T = O(N)$, and $m_i = m$ for $i \in [N]$

	COPML (online)	**PICO (offline+online)**
1. Dataset encoding	$O(N^2dm)$	$O(Ndm)$
2. Label encoding	$O(N^2m + N^2d)$	$O(Nd)$
3. Model initialization	$O(N^2d)$	$O(Nd)$
4. Model encoding	$O(N^2dJ)$	$O(NdJ)$
5. Gradient computing and model update	$O(N^2dJ)$	$O(NdJ)$

1) Offline: Clients first agree on $N + K + T$ distinct public parameters $\{\alpha_j\}_{j\in[N]}$ and $\{\beta_j\}_{j\in[K+T]}$ from $\mathbb{F}_q$. Each client $i \in [N]$ then sends an encoded matrix,

$$\widetilde{\mathbf{R}}_{ij} \triangleq \sum_{k\in[K]} \mathbf{R}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} + \sum_{k=K+1}^{K+T} \mathbf{V}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \tag{9}$$

to client $j \in [N]$, where $\{\mathbf{R}_{ik}\}_{k\in[K]}$, $\{\mathbf{V}_{ik}\}_{k\in\{K+1,\ldots,K+T\}}$ are uniformly random matrices of size $\frac{m_i}{K} \times d$, generated locally by client i.

2) Online: In the online phase, client $i \in [N]$ locally partitions its dataset $\overline{\mathbf{X}}_i$ into K equal-sized shards $\overline{\mathbf{X}}_i = \begin{bmatrix}\overline{\mathbf{X}}_{i1}^{\mathrm{T}} & \cdots & \overline{\mathbf{X}}_{iK}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, where $\overline{\mathbf{X}}_{ik} \in \mathbb{F}_q^{\frac{m_i}{K}\times d}$ for all $k \in [K]$, and broadcasts,

$$\widehat{\mathbf{X}}_{ik} = \overline{\mathbf{X}}_{ik} - \mathbf{R}_{ik} \quad \forall k \in [K]. \tag{10}$$

After receiving $\{\widehat{\mathbf{X}}_{jk}\}_{j\in[N],k\in[K]}$, each client $i \in [N]$ generates an encoded dataset:

$$\widetilde{\mathbf{X}}_i \triangleq \sum_{k\in[K]} \left[\widehat{\mathbf{X}}_{1k}^{\mathrm{T}} \quad \cdots \quad \widehat{\mathbf{X}}_{Nk}^{\mathrm{T}}\right]^{\mathrm{T}} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l} + \left[\widetilde{\mathbf{R}}_{1i}^{\mathrm{T}} \quad \cdots \quad \widetilde{\mathbf{R}}_{Ni}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{11}$$

Intuitively, the encoding operation from (11) simultaneously cancels the additive randomness due to $\{\mathbf{R}_{jk}\}_{k\in[K],j\in[N]}$, and embeds the dataset $\overline{\mathbf{X}}$ in a degree $K + T - 1$ Lagrange polynomial,

$$f(\alpha) \triangleq \sum_{k\in[K]} \begin{bmatrix}\overline{\mathbf{X}}_{1k}^{\mathrm{T}} & \cdots & \overline{\mathbf{X}}_{Nk}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha - \beta_l}{\beta_k - \beta_l} + \sum_{k=K+1}^{K+T} \left[\mathbf{V}_{1k}^{\mathrm{T}} \quad \cdots \quad \mathbf{V}_{Nk}^{\mathrm{T}}\right]^{\mathrm{T}} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha - \beta_l}{\beta_k - \beta_l} \tag{12}$$

where $f(\beta_k) = \begin{bmatrix}\overline{\mathbf{X}}_{1k}^{\mathrm{T}} & \cdots & \overline{\mathbf{X}}_{Nk}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ for all $k \in [K]$, and client $i \in [N]$ obtains the encoded dataset $\widetilde{\mathbf{X}}_i = f(\alpha_i)$. The T random matrices $\left\{ \left[\mathbf{V}_{1k}^{\mathrm{T}} \quad \cdots \quad \mathbf{V}_{Nk}^{\mathrm{T}}\right]^{\mathrm{T}} \right\}_{k\in\{K+1,\ldots,K+T\}}$ along with the random masks $\{\mathbf{R}_{ik}\}_{k\in[K]}$ allow clients to use (one-to-many) broadcast while encoding the datasets as opposed to (point-to-point) unicast in the online phase, while hiding the true values of the local datasets against up to T adversaries. As will be described later, client i then computes the gradient on the encoded dataset $\widetilde{\mathbf{X}}_i$, whose size is $(1/K)^{th}$ of the original dataset $\overline{\mathbf{X}}$. As the network size N increases, one can select a larger K, reducing the training load per client (called the *parallelization gain*) to speed up training.

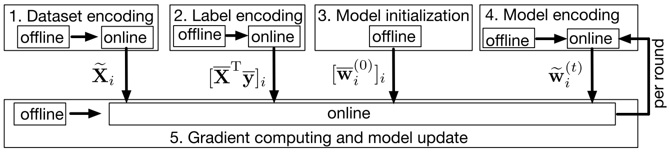

1. Dataset encoding
offline
online
2. Label encoding
offline
online
3. Model initialization
offline
4. Model encoding
offline
online
$\widetilde{\mathbf{X}}_i$
$[\overline{\mathbf{X}}^{\mathrm{T}}\overline{\mathbf{y}}]_i$
$[\overline{\mathbf{w}}_i^{(0)}]_i$
$\widetilde{\mathbf{w}}_i^{(t)}$
per round
offline
online
5. Gradient computing and model update

Fig. 2. PICO consists of five main components.

Remark 2: In practice, if m_i/K is not an integer, client i can zero-pad their local dataset [75] with synthetic data samples $\mathbf{x}_i = \mathbf{0}$, by setting all features to 0. As the gradients of such samples are zero, the pre-processing will not change the final model. Another approach is for each client to locally create additional training samples using common data augmentation mechanisms, such as label-preserving transformations (e.g., rotations, horizontal/vertical flips, and random cropping), which can further improve test accuracy [76], [77].

B. Label Encoding

Clients also encode their *labels* through the following offline and online phases.

1) Offline: Client $j \in [N]$ generates K uniformly random vectors $\mathbf{a}_{jk} \in \mathbb{F}_q^{\frac{d}{(N-T)K}\times 1}$ for $k \in [K]$, and sends to each client $i \in [N]$: 1) a secret share $[\mathbf{a}_{jk}]_i$ of $\mathbf{a}_{jk}$ using SSS, 2) an encoded vector,

$$\widetilde{\mathbf{a}}_{ji} = \sum_{k\in[K]} \mathbf{a}_{jk} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l} + \sum_{k=K+1}^{K+T} \mathbf{b}_{jk} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i - \beta_l}{\beta_k - \beta_l} \tag{13}$$

where $\mathbf{b}_{jk} \in \mathbb{F}_q^{\frac{d}{N-T}}$ are uniformly random vectors for $k \in \{K+1, \ldots, K+T\}$. By combining $\{\widetilde{\mathbf{a}}_{ji}, [\mathbf{a}_{jk}]_i\}_{j\in[N],k\in[K]}$, client i then forms a (large-dimensional) encoded vector,

$$\widetilde{\mathbf{a}}_i \triangleq (\mathbf{M} \otimes \mathbf{I}) \times \left[\widetilde{\mathbf{a}}_{1i}^{\mathrm{T}} \quad \cdots \quad \widetilde{\mathbf{a}}_{Ni}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{14}$$

and a secret share,

$$[\mathbf{a}_k]_i \triangleq (\mathbf{M} \otimes \mathbf{I}) \times \left[[\mathbf{a}_{1k}]_i^{\mathrm{T}} \quad \cdots \quad [\mathbf{a}_{Nk}]_i^{\mathrm{T}}\right]^{\mathrm{T}} \quad \forall k \in [K], \tag{15}$$

where $\mathbf{a}_k \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \left[\mathbf{a}_{1k}^{\mathrm{T}} \quad \cdots \quad \mathbf{a}_{Nk}^{\mathrm{T}}\right]^{\mathrm{T}}$, $\mathbf{I}$ is a $\frac{d}{(N-T)K} \times \frac{d}{(N-T)K}$ identity matrix, and

$$\mathbf{M} = \begin{bmatrix} 1 & \lambda_1 & \cdots & \lambda_1^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_{N-T} & \cdots & \lambda_{N-T}^{N-1} \end{bmatrix} \tag{16}$$

is a $(N-T)\times N$ MDS matrix, where $\lambda_1, \ldots, \lambda_{N-T}$ are distinct public parameters from $\mathbb{F}_q$. The key intuition is that, to generate an encoded vector of size $\frac{d}{K}$, each client only sends $\frac{d}{(N-T)K}$ parameters to every other client.[2] The final encoded vector is then generated by combining the *lower-dimensional* encoded vectors received from all N clients, using the MDS matrix $\mathbf{M}$.

2) Online: In the online phase, client $i \in [N]$ partitions $\overline{\mathbf{X}}_i^{\mathrm{T}}\overline{\mathbf{y}}_i$ into K equal-sized shards $\overline{\mathbf{X}}_i^{\mathrm{T}}\overline{\mathbf{y}}_i = \left[\overline{\mathbf{y}}_{i1}^{\mathrm{T}} \quad \cdots \quad \overline{\mathbf{y}}_{iK}^{\mathrm{T}}\right]^{\mathrm{T}}$, and sends an encoded vector,

$$\begin{aligned}\widetilde{\mathbf{y}}_{ij} \triangleq &\sum_{k\in[K]} \overline{\mathbf{y}}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l} \\ &+ \sum_{k=K+1}^{K+T} \mathbf{r}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\end{aligned} \tag{17}$$

to each client $j \in [N]$, where $\mathbf{r}_{ik} \in \mathbb{F}_q^{\frac{d}{K}}$ are generated uniformly at random. After receiving $\{\widetilde{\mathbf{y}}_{ij}\}_{i\in[N]}$, client $j \in [N]$ broadcasts,

$$\hat{\mathbf{a}}_j \triangleq \sum_{i\in[N]} \widetilde{\mathbf{y}}_{ij} - \widetilde{\mathbf{a}}_j \tag{18}$$

which can be viewed as an evaluation point of a Lagrange polynomial of degree $K+T-1$. Upon receiving $\hat{\mathbf{a}}_j$ from any set of at least $K+T$ clients, client $i \in [N]$ decodes $\sum_{j\in[N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k$ for all $k \in [K]$ via polynomial interpolation, and computes a secret share of $\overline{\mathbf{X}}^{\mathrm{T}}\overline{\mathbf{y}} = \sum_{j\in[N]} \overline{\mathbf{X}}_j^{\mathrm{T}}\overline{\mathbf{y}}_j$,

$$[\overline{\mathbf{X}}^{\mathrm{T}}\overline{\mathbf{y}}]_i \triangleq \begin{bmatrix} \left(\sum_{j\in[N]} \overline{\mathbf{y}}_{j1} - \mathbf{a}_1 + [\mathbf{a}_1]_i\right) \\ \vdots \\ \left(\sum_{j\in[N]} \overline{\mathbf{y}}_{jK} - \mathbf{a}_K + [\mathbf{a}_K]_i\right) \end{bmatrix} \tag{19}$$

C. Model Initialization

Model $\overline{\mathbf{w}}^{(0)}$ at time $t=0$ is initialized uniformly random (offline), without revealing its true value to any client. To do so, client i generates a random vector $\overline{\mathbf{w}}_i^{(0)}$ of size $\frac{d}{N-T}$, and sends a secret share $[\overline{\mathbf{w}}_i^{(0)}]_j$ of $\overline{\mathbf{w}}_i^{(0)}$ to client $j \in [N]$ using SSS. After receiving $[\overline{\mathbf{w}}_j^{(0)}]_i$ for $j \in [N]$, each client $i \in [N]$ constructs a new (larger) secret share,

$$[\overline{\mathbf{w}}^{(0)}]_i \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \left[([\overline{\mathbf{w}}_1^{(0)}]_i)^{\mathrm{T}} \quad \cdots \quad ([\overline{\mathbf{w}}_N^{(0)}]_i)^{\mathrm{T}}\right]^{\mathrm{T}} \tag{20}$$

which corresponds to a secret share of the initialized model,

$$\overline{\mathbf{w}}^{(0)} = (\mathbf{M}\otimes\mathbf{I}) \times \left[(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} \quad \cdots \quad (\overline{\mathbf{w}}_N^{(0)})^{\mathrm{T}}\right]^{\mathrm{T}} \tag{21}$$

where $\mathbf{I}$ is a $\frac{d}{N-T} \times \frac{d}{N-T}$ identity matrix.

D. Model Encoding

At the beginning of each round, client i holds a secret share $[\overline{\mathbf{w}}^{(t)}]_i$ of the current state of the model $\overline{\mathbf{w}}^{(t)}$. Initially at $t=0$, $[\overline{\mathbf{w}}^{(0)}]_i$ is generated during model initialization as described in (20). For all other training rounds (i.e., $t>0$), $[\overline{\mathbf{w}}^{(t)}]_i$ is obtained after the model updating stage, which will be described in (40). At each round, clients then *encode* the model $\overline{\mathbf{w}}^{(t)}$ using the secret shares $[\overline{\mathbf{w}}^{(t)}]_i$, to enable gradient computations to be performed on the encoded datasets. At the end of this stage, each client $i \in [N]$ learns an encoded model $\widetilde{\mathbf{w}}_i^{(t)}$. Model encoding consists of the following offline and online phases.

1) Offline: Client $i \in [N]$ generates a uniformly random vector $\mathbf{r}_i^{(t)} \in \mathbb{F}_q^{\frac{d}{N-T}}$, and sends to each client $j \in [N]$: 1) a secret share $[\mathbf{r}_i^{(t)}]_j$ of $\mathbf{r}_i^{(t)}$ using SSS, and 2) an encoded vector,

$$\begin{aligned}\widetilde{\mathbf{r}}_{ij}^{(t)} \triangleq &\sum_{k\in[K]} \mathbf{r}_i^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l} \\ &+ \sum_{k=K+1}^{K+T} \mathbf{v}_{ik}^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\end{aligned} \tag{22}$$

where $\mathbf{v}_{ik}^{(t)} \in \mathbb{F}_q^{\frac{d}{N-T}}$ for $k \in \{K+1, \ldots, K+T\}$ are generated uniformly at random. By combining $\{\widetilde{\mathbf{r}}_{ji}^{(t)}, [\mathbf{r}_j]_i^{(t)}\}_{j\in[N]}$, client i then generates a (large-dimensional) encoded vector,

$$\widetilde{\mathbf{r}}_i^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \left[(\widetilde{\mathbf{r}}_{1i}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\widetilde{\mathbf{r}}_{Ni}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{23}$$

and a (large-dimensional) secret share,

$$[\mathbf{r}]_i^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \left[[\mathbf{r}_1^{(t)}]_i^{\mathrm{T}} \quad \cdots \quad [\mathbf{r}_N^{(t)}]_i^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{24}$$

where $\mathbf{r}^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_N^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}$ is a random mask that will later be utilized to hide the true model in the online phase. In doing so, the key intuition is to generate secret shares $[\mathbf{r}^{(t)}]_i$ of a random mask $\mathbf{r}^{(t)}$ that will later be utilized to decode a masked model in the online phase (where the true model will be hidden by the mask $\mathbf{r}^{(t)}$), after which the encoded masks $\widetilde{\mathbf{r}}_i^{(t)}$ will be utilized to encode the model for training.

2) Online: In the online phase, client i initially broadcasts,

$$[\widehat{\mathbf{w}}^{(t)}]_i \triangleq [\overline{\mathbf{w}}^{(t)}]_i - [\mathbf{r}^{(t)}]_i = [\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}]_i \tag{25}$$

which corresponds to a secret share of the masked model $\overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}$. After receiving $\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}$, each client can decode a masked model,

$$\widehat{\mathbf{w}}^{(t)} = \overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)} \tag{26}$$

via polynomial interpolation, where the true value of the model $\overline{\mathbf{w}}^{(t)}$ is hidden by the random mask $\mathbf{r}^{(t)}$. Using (26), client i then constructs an encoded model,

$$\widetilde{\mathbf{w}}_i^{(t)} \triangleq \sum_{k\in[K]} \widehat{\mathbf{w}}^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_i-\beta_l}{\beta_k-\beta_l} + \widetilde{\mathbf{r}}_i^{(t)} \tag{27}$$

[2] Typically $d \gg N$ in real-world tasks [78].

Intuitively, the encoding operation in (27) embeds the model $\overline{\mathbf{w}}^{(t)}$ in a Lagrange polynomial,

$$h(\alpha) \triangleq \sum_{k\in[K]} \overline{\mathbf{w}}^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha-\beta_l}{\beta_k-\beta_l} + \sum_{k=K+1}^{K+T} \mathbf{v}_k^{(t)} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha-\beta_l}{\beta_k-\beta_l} \tag{28}$$

such that $\mathbf{v}_k^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I})\times\left[(\mathbf{v}_{1k}^{(t)})^{\mathrm{T}} \;\cdots\; (\mathbf{v}_{Nk}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}$, where $h(\beta_k) = \overline{\mathbf{w}}^{(t)}$ for $k\in[K]$, and client i obtains an encoded model $\widetilde{\mathbf{w}}_i^{(t)} = h(\alpha_i)$. The random vectors $\{\mathbf{v}_k^{(t)}\}_{k\in\{K+1,\dots,K+T\}}$ hide the true value of $\overline{\mathbf{w}}^{(t)}$ against up to T adversaries.

E. Gradient Computing and Model Update

The last component of PICO is gradient computation and model update, using the encoded datasets and model. At the end, client i learns a secret share $[\overline{\mathbf{w}}^{(t+1)}]_i$ of the model $\overline{\mathbf{w}}^{(t+1)}$ for the next training round.

(Gradient Computing): Initially, clients compute the gradient using the encoded dataset and model. The offline and online phases of this stage proceed as follows.

1) Offline: Client $i\in[N]$ generates $C \triangleq (2r+1)(K+T-1)+1$ random vectors $\mathbf{u}_{ik}$ of size $\frac{d}{N-T}$, and constructs a Lagrange polynomial of degree $C-1$,

$$\phi_i(\alpha) \triangleq \sum_{k\in[C]} \mathbf{u}_{ik}^{(t)} \prod_{l\in[C]\setminus\{k\}} \frac{\alpha-\beta_l}{\beta_k-\beta_l} \tag{29}$$

where $\{\beta_k\}_{k\in\{K+1,\dots,C\}}$ are distinct public parameters from $\mathbb{F}_q$, and $\phi_i(\beta_k) = \mathbf{u}_{ik}^{(t)}$ for $k\in[C]$. Client i then sends an encoded vector,

$$\widetilde{\mathbf{u}}_{ij}^{(t)} \triangleq \phi_i(\alpha_j) \tag{30}$$

to each client $j\in[N]$. After receiving $\{\widetilde{\mathbf{u}}_{ji}^{(t)}\}_{j\in[N]}$, client i constructs a new (large-dimensional) encoded vector,

$$\widetilde{\mathbf{u}}_i^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I})\times\left[(\widetilde{\mathbf{u}}_{1i}^{(t)})^{\mathrm{T}} \;\cdots\; (\widetilde{\mathbf{u}}_{Ni}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}} \tag{31}$$

which can be viewed as an evaluation of a degree $C-1$ Lagrange polynomial,

$$\phi(\alpha) \triangleq \sum_{k\in[C]} \mathbf{u}_k^{(t)} \prod_{l\in[C]\setminus\{k\}} \frac{\alpha-\beta_l}{\beta_k-\beta_l} \tag{32}$$

such that $\mathbf{u}_k^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I})\times\left[(\mathbf{u}_{1k}^{(t)})^{\mathrm{T}} \;\cdots\; (\mathbf{u}_{Nk}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\phi(\beta_k) = \mathbf{u}_k^{(t)}$ for all $k\in[C]$, and client i obtains an encoded vector $\widetilde{\mathbf{u}}_i^{(t)} = \phi(\alpha_i)$. Client i then secret shares the sum $\sum_{k\in[K]}\mathbf{u}_{ik}^{(t)}$, by sending each client $j\in[N]$ a secret share,

$$\Big[\sum_{k\in[K]} \mathbf{u}_{ik}^{(t)}\Big]_j \triangleq \sum_{k\in[K]} \mathbf{u}_{ik}^{(t)} + \sum_{l\in[T]} \gamma_j^l \mathbf{z}_{il}^{(t)} \tag{33}$$

where $\mathbf{z}_{il}^{(t)}$ are uniformly random vectors, and $\{\gamma_j\}_{j\in[N]}$ are distinct public parameters. After receiving $[\sum_{k\in[K]}\mathbf{u}_{jk}^{(t)}]_i$ for $j\in[N]$, client i generates a secret share of $\sum_{k\in[K]}\mathbf{u}_k^{(t)}$,

$$\Big[\sum_{k\in[K]} \mathbf{u}_k^{(t)}\Big]_i \triangleq (\mathbf{M}\otimes\mathbf{I}) \times \begin{bmatrix} [\sum_{k\in[K]}\mathbf{u}_{1k}^{(t)}]_i \\ \vdots \\ [\sum_{k\in[K]}\mathbf{u}_{Nk}^{(t)}]_i \end{bmatrix} \tag{34}$$

2) Online: PPML frameworks that build on polynomial embeddings, as in our framework, are bound to finite field polynomial operations. The sigmoid function in (1) is not a polynomial, hence is often approximated with a polynomial $\hat{g}(x) = \sum_{i=0}^{r}\theta_i x^i$ [38] where $\{\theta_i\}_{i\in[r]}$ are public coefficients fitted via least squares (prior to training), and degree r quantifies the accuracy of approximation [79]. Then, client i computes a local gradient,

$$\varphi(\alpha_i) \triangleq \widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)}) \tag{35}$$

using the encoded dataset $\widetilde{\mathbf{X}}_i$ and model $\widetilde{\mathbf{w}}_i^{(t)}$, where we define a degree $C-1$ polynomial $\varphi(\alpha) = f(\alpha)^{\mathrm{T}}\hat{g}(f(\alpha)\times h(\alpha))$ using (12) and (28), such that client i computes the encoded gradient $\varphi(\alpha_i)$, whereas the true gradient is given by,

$$\overline{\mathbf{X}}^{\mathrm{T}}\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)}) = \sum_{k\in[K]}\varphi(\beta_k) = \sum_{k\in[K]}(\overline{\mathbf{X}}'_k)^{\mathrm{T}}\hat{g}(\overline{\mathbf{X}}'_k\times\overline{\mathbf{w}}^{(t)}), \tag{36}$$

where $\overline{\mathbf{X}}'_k \triangleq f(\beta_k) = \left[\overline{\mathbf{X}}_{1k}^{\mathrm{T}} \;\cdots\; \overline{\mathbf{X}}_{Nk}^{\mathrm{T}}\right]^{\mathrm{T}}$ from (12). Then, client i broadcasts a masked gradient,

$$\hat{\mathbf{u}}_i^{(t)} \triangleq \widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)}) - \widetilde{\mathbf{u}}_i^{(t)} = \varphi(\alpha_i) - \phi(\alpha_i), \tag{37}$$

which is an evaluation of the degree $C-1$ polynomial $\psi(\alpha) \triangleq \varphi(\alpha) - \phi(\alpha)$. Upon receiving $\hat{\mathbf{u}}_j^{(t)}$ from any set $j\in\mathcal{S}$ of at least $\deg(\psi)+1 = C$ clients, client i can recover $\psi(\alpha)$ via polynomial interpolation, and compute a secret share of the *true gradient* $\overline{\mathbf{X}}^{\mathrm{T}}\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)})$ using (34),

$$\begin{aligned}\Big[\overline{\mathbf{X}}^{\mathrm{T}}\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)})\Big]_i &\triangleq \sum_{k\in[K]}\psi(\beta_k) + \Big[\sum_{k\in[K]}\mathbf{u}_k^{(t)}\Big]_i && (38)\\ &= \overline{\mathbf{X}}^{\mathrm{T}}\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)}) + \sum_{l\in[T]}\gamma_i^l\mathbf{z}_l^{(t)}, && (39)\end{aligned}$$

where $\mathbf{z}_l^{(t)} \triangleq (\mathbf{M}\otimes\mathbf{I})\times\left[(\mathbf{z}_{1l}^{(t)})^{\mathrm{T}} \;\cdots\; (\mathbf{z}_{Nl}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}$ for $l\in[T]$ are random masks that hide the true gradient against up to T adversaries. The model update at client i can then be written as,

$$[\overline{\mathbf{w}}^{(t+1)}]_i = [\overline{\mathbf{w}}^{(t)}]_i - \frac{\eta}{m}([\overline{\mathbf{X}}^{T}\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)})]_i - [\overline{\mathbf{X}}^{\mathrm{T}}\overline{\mathbf{y}}]_i), \tag{40}$$

where, on the other hand, $\frac{\eta}{m} \ll 1$ in (40). To handle this operation in the finite field, one can either convert (40) to a computation on integers [2] by assuming a sufficiently large field size, as will be detailed in Appendix D, or can utilize a secure multi-party truncation (quantization) protocol [80] to reduce the required field size (albeit with weaker privacy) as will be detailed in Section VIII. In our theoretical analysis in Section VII, we assume a sufficiently large field size and consider the former, whereas we utilize the latter in our experiments from Section VIII.

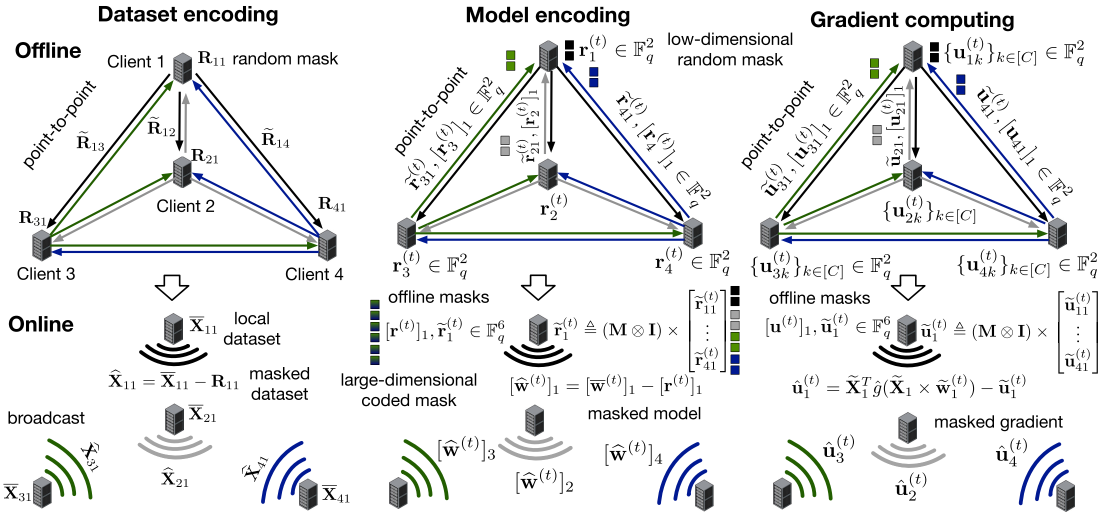

Dataset encoding
Offline
Client 1
$\mathbf{R}_{11}$ random mask
point-to-point
$\widetilde{\mathbf{R}}_{13}$
$\widetilde{\mathbf{R}}_{12}$
$\widetilde{\mathbf{R}}_{14}$
$\mathbf{R}_{21}$
Client 2
$\mathbf{R}_{31}$
$\mathbf{R}_{41}$
Client 3
Client 4
Online
$\overline{\mathbf{X}}_{11}$
local dataset
$\widehat{\mathbf{X}}_{11} = \overline{\mathbf{X}}_{11} - \mathbf{R}_{11}$
masked dataset
$\overline{\mathbf{X}}_{21}$
broadcast
$\widehat{\mathbf{X}}_{31}$
$\overline{\mathbf{X}}_{31}$
$\widehat{\mathbf{X}}_{21}$
$\widehat{\mathbf{X}}_{41}$
$\overline{\mathbf{X}}_{41}$
Model encoding
$\mathbf{r}_1^{(t)} \in \mathbb{F}_q^2$
low-dimensional random mask
point-to-point
$\widetilde{\mathbf{r}}_{31}^{(t)}, [\mathbf{r}_3^{(t)}]_1 \in \mathbb{F}_q^2$
$\widetilde{\mathbf{r}}_{21}^{(t)}, [\mathbf{r}_2^{(t)}]_1$
$\widetilde{\mathbf{r}}_{41}^{(t)}, [\mathbf{r}_4^{(t)}]_1 \in \mathbb{F}_q^2$
$\mathbf{r}_2^{(t)}$
$\mathbf{r}_3^{(t)} \in \mathbb{F}_q^2$
$\mathbf{r}_4^{(t)} \in \mathbb{F}_q^2$
offline masks
$[\mathbf{r}^{(t)}]_1, \widetilde{\mathbf{r}}_1^{(t)} \in \mathbb{F}_q^6$
$\widetilde{\mathbf{r}}_1^{(t)} \triangleq (\mathbf{M} \otimes \mathbf{I}) \times \begin{bmatrix} \widetilde{\mathbf{r}}_{11}^{(t)} \\ \vdots \\ \widetilde{\mathbf{r}}_{41}^{(t)} \end{bmatrix}$
large-dimensional coded mask
$[\widehat{\mathbf{w}}^{(t)}]_1 = [\overline{\mathbf{w}}^{(t)}]_1 - [\mathbf{r}^{(t)}]_1$
masked model
$[\widehat{\mathbf{w}}^{(t)}]_3$
$[\widehat{\mathbf{w}}^{(t)}]_2$
$[\widehat{\mathbf{w}}^{(t)}]_4$
Gradient computing
$\{\mathbf{u}_{1k}^{(t)}\}_{k\in[C]} \in \mathbb{F}_q^2$
point-to-point
$\widetilde{\mathbf{u}}_{31}^{(t)}, [\mathbf{u}_{31}^{(t)}]_1 \in \mathbb{F}_q^2$
$\widetilde{\mathbf{u}}_{21}, [\mathbf{u}_{21}^{(t)}]_1$
$\widetilde{\mathbf{u}}_{41}^{(t)}, [\mathbf{u}_{41}]_1 \in \mathbb{F}_q^2$
$\{\mathbf{u}_{2k}^{(t)}\}_{k\in[C]}$
$\{\mathbf{u}_{3k}^{(t)}\}_{k\in[C]} \in \mathbb{F}_q^2$
$\{\mathbf{u}_{4k}^{(t)}\}_{k\in[C]} \in \mathbb{F}_q^2$
offline masks
$[\mathbf{u}^{(t)}]_1, \widetilde{\mathbf{u}}_1^{(t)} \in \mathbb{F}_q^6$
$\widetilde{\mathbf{u}}_1^{(t)} \triangleq (\mathbf{M} \otimes \mathbf{I}) \times \begin{bmatrix} \widetilde{\mathbf{u}}_{11}^{(t)} \\ \vdots \\ \widetilde{\mathbf{u}}_{41}^{(t)} \end{bmatrix}$
$\hat{\mathbf{u}}_1^{(t)} = \widetilde{\mathbf{X}}_1^T \hat{g}(\widetilde{\mathbf{X}}_1 \times \widetilde{\mathbf{w}}_1^{(t)}) - \widetilde{\mathbf{u}}_1^{(t)}$
masked gradient
$\hat{\mathbf{u}}_3^{(t)}$
$\hat{\mathbf{u}}_2^{(t)}$
$\hat{\mathbf{u}}_4^{(t)}$

Fig. 3. Motivating example. (Offline) Locally generated lower dimensional random vectors are combined to construct large dimensional shared randomness. (Online) The randomness generated offline is utilized to encode the datasets and model.

3) Final Model Recovery : After J training rounds, clients can collect the secret shares $\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{I}}$ from any set $\mathcal{I}$ of at least $|\mathcal{I}| \geq T+1$ clients, and decode the final model $\overline{\mathbf{w}}^{(J)}$.

Our overall algorithm is given in Appendix A.

VI. Motivating Example

We next present a motivating example for $N = 4$ clients, with $d = 6$ and $K = T = 1$ as illustrated in Fig 3. Initially, clients encode their local datasets. The main intuition is to generate and encode random masks offline, where each client $i \in [4]$ generates a random mask $\mathbf{R}_{i1}$, and sends an encoded mask $\widetilde{\mathbf{R}}_{ij}$ to client $j \in [4]$. The offline random masks are later used in the online phase to hide the local datasets $\overline{\mathbf{X}}_{i1}$ where client i broadcasts a masked dataset $\widehat{\mathbf{X}}_{i1} = \overline{\mathbf{X}}_{i1} - \mathbf{R}_{i1}$, using which, along with the offline encoded masks $\widetilde{\mathbf{R}}_{ij}$, clients encode the datasets. At the end, each client i learns an encoded dataset $\widetilde{\mathbf{X}}_i$. In addition to dataset encoding, clients also encode their labels and initialize the model as described in Sections V-B and V-C, respectively.

At each training round t, clients also encode the model $\overline{\mathbf{w}}^{(t)}$. To prevent information leakage from intermediate model parameters, no client can learn the true model during encoding. The key intuition is to use locally generated *lower-dimensional* coded masks to generate high dimensional shared coded randomness. To do so, client i locally generates a random mask $\mathbf{r}_i^{(t)}$ of size $\frac{d}{N-T} = 2$ offline, and then sends to each client $j \in [4]$: 1) an encoded mask $\widetilde{\mathbf{r}}_{ij}^{(t)} \in \mathbb{F}_q^2$, 2) a secret share $[\mathbf{r}_i^{(t)}]_j \in \mathbb{F}_q^2$. After receiving $\{\widetilde{\mathbf{r}}_{ji}^{(t)}, [\mathbf{r}_j^{(t)}]_i\}_{j\in[4]}$, client i generates two *large-dimensional* random vectors (each of size $d = 6$): 1) encoded mask $\widetilde{\mathbf{r}}_i^{(t)} \in \mathbb{F}_q^6$, and 2) secret shared mask $[\mathbf{r}^{(t)}]_i \in \mathbb{F}_q^6$. The offline random masks are then used to mask and encode the true model in the online phase, where each client decodes the masked model $\widehat{\mathbf{w}}^{(t)} = \overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)} \in \mathbb{F}_q^6$, and obtains an encoded model $\widetilde{\mathbf{w}}_i \in \mathbb{F}_q^6$, but without learning the true model $\overline{\mathbf{w}}^{(t)} \in \mathbb{F}_q^6$, which is hidden by the random mask $\mathbf{r}^{(t)} \in \mathbb{F}_q^6$ throughout the encoding.

Using the encoded dataset $\widetilde{\mathbf{X}}_i$ and encoded model $\widetilde{\mathbf{w}}_i^{(t)}$, clients then compute the gradient and update the model. In doing so, no client should learn the true gradient $\overline{\mathbf{X}}^T \hat{g}(\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)})$ or the updated model $\overline{\mathbf{w}}^{(t+1)}$, as gradients may carry sensitive information about the true datasets. The intuition is again to use lower-dimensional local randomness to generate large-dimensional encoded shared randomness. To do so, offline, client i generates C random masks $\{\mathbf{u}_{ik}^{(t)}\}_{k\in[C]}$ of size $\frac{d}{N-T} = 2$, and sends to every other client $j \in [4]$ an encoded mask $\widetilde{\mathbf{u}}_{ij}^{(t)} \in \mathbb{F}_q^2$, and a secret share $[\mathbf{u}_{i1}^{(t)}]_j \in \mathbb{F}_q^2$. After receiving $\{\widetilde{\mathbf{u}}_{ji}^{(t)}, [\mathbf{u}_{j1}^{(t)}]_i\}_{j\in[4]}$, each client i generates a large-dimensional encoded mask $\widetilde{\mathbf{u}}_i^{(t)} \in \mathbb{F}_q^6$ and secret share $[\mathbf{u}^{(t)}]_i \in \mathbb{F}_q^6$, each of size $d = 6$. Online, each client i computes a local gradient $\widetilde{\mathbf{X}}_i^T \hat{g}(\widetilde{\mathbf{X}}_i \times \widetilde{\mathbf{w}}_i^{(t)}) \in \mathbb{F}_q^6$ and broadcasts $\hat{\mathbf{u}}_i^{(t)} = \widetilde{\mathbf{X}}_i^T \hat{g}(\widetilde{\mathbf{X}}_i \times \widetilde{\mathbf{w}}_i^{(t)}) - \widetilde{\mathbf{u}}_i^{(t)} \in \mathbb{F}_q^6$, using which each client can decode a masked gradient $\overline{\mathbf{X}}^T \hat{g}(\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)}) - \mathbf{u}^{(t)} \in \mathbb{F}_q^6$ where the true gradient $\overline{\mathbf{X}}^T \hat{g}(\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)}) \in \mathbb{F}_q^6$ is hidden by the offline mask $\mathbf{u}^{(t)} \in \mathbb{F}_q^6$, to generate a secret share $[\overline{\mathbf{X}}^T \hat{g}(\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)})]_i \in \mathbb{F}_q^6$ and update the model $[\overline{\mathbf{w}}^{(t+1)}]_i \in \mathbb{F}_q^6$ for the next training round.

VII. Theoretical Analysis

In this section, we provide the theoretical performance guarantees of PICO. We first present the total communication complexity (across all clients). To explicitly demonstrate the complexity with respect to the number of clients, in the following we let $m_i = m$ for $i \in [N]$.

Theorem 1 (Communication Complexity): For training a logistic regression model of size d in a network of N clients, where up to T clients are adversarial, and each client has m data samples partitioned into K shards, the total communication complexity of PICO after J training rounds is given by $O(Ndm + \frac{N^2}{K}d + NdJ)$ in the online phase, and $O(\frac{N^2}{K}dm + \frac{N^2}{N-T}dJ)$ in the offline phase. With $K = \Theta(N)$ and $T = O(N)$, the total communication complexity (offline+online) is linear in the number of clients, which is $O(Ndm + NdJ)$.

Proof: The proof is provided in Appendix B. □

As can be observed from Theorem 1, PICO achieves a linear communication complexity both offline and online, significantly improving over the quadratic (online) communication complexity of the state-of-the-art. We next demonstrate the per-client computation complexity for PICO.

Theorem 2: (Computation complexity) For training a logistic regression model of size d in a network of N clients, where up to T clients are adversarial, and each client has m data samples partitioned into K shards, after J training rounds, PICO incurs a per-client computation overhead $O(Nmd + N\frac{d}{K}\log^2(K+T)\log\log(K+T) + J\frac{Nm}{K}(d+r) + Jdr(K+T)\log^2 r(K+T)\log\log r(K+T))$ in the online phase, and $O(Nd\frac{m}{K}\log^2(K+T)\log\log(K+T) + JN\frac{d}{N-T}\log^2 r(K+T)\log\log r(K+T)+JNd)$ in the offline phase.

Proof: The proof is provided in Appendix C. □

In Appendix C, we also compare the computational complexity of PICO with COPML, and show that PICO reduces the communication complexity without any additional overhead on the computation complexity.

The *recovery threshold* is defined as the minimum number of clients needed for correct recovery of the final model. We next present the recovery threshold of PICO.

Theorem 3 (Recovery Threshold): In a network of N clients, where up to T clients are adversarial, and up to D clients may drop out (or are unavailable) in each training round, the recovery threshold of PICO is $N \geq D + (2r+1)(K+T-1)+1$, where r is the degree of polynomial approximation of the sigmoid function.

Proof: The minimum number of clients is determined by the number of local computations required for polynomial interpolation, which, from Section V is given by $N - D \geq (2r+1)(K+T-1)+1$. □

From [7], the recovery threshold of COPML is given by $N \geq D+(2r+1)(K+T-1)+1$, where $r \geq 1$. Hence, PICO achieves equal adversary-robustness (T), dropout-resilience (D), and parallelization (K) guarantees, while also slashing the communication overhead.

Remark 3: PICO can also be applied to the simpler linear regression problem, with the same algorithm steps.

We next present the formal information-theoretic privacy guarantees from (3).

Theorem 4 (Information-Theoretic Privacy): In a network of N clients, where $\mathcal{T}$ and $\mathcal{H} = [N]\backslash\mathcal{T}$ denote the set of adversarial and honest clients, respectively, PICO guarantees information theoretic-privacy for training a logistic regression model $\mathbf{w}^{(J)}$ after J training rounds,

$$I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) = 0 \tag{41}$$

where $\mathcal{M}_{\mathcal{T}}$ denotes the collection of all messages received or generated by the adversaries throughout the training.

Proof: The proof is provided in Appendix D. □

Finally, we show that the training operations correctly recover the target model given in (40).

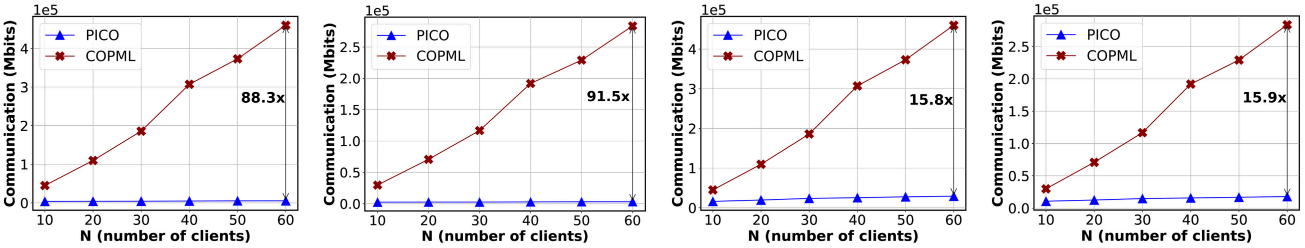
Theorem 5 (Correctness): PICO correctly recovers the target model from (40), given a sufficiently large field $\mathbb{F}_q$.

Proof: The proof is given in Appendix E. □

VIII. Experiments

To evaluate the performance of PICO, we implement a distributed logistic regression task for binary classification on the CIFAR-10 (on classes *plane* and *car*) [81], and MNIST (on digits 0 and 1) [82] datasets, with dataset sizes $(\overline{m}, d) = (9019, 3073)$ and $(11432, 785)$, respectively. The datasets are distributed evenly across the clients. In all experiments, the inter-client communication is implemented using the `MPI4Py` Message Passing Interface (MPI) for Python [83]. The *broadcast* functionality of the MPI protocol communicates messages through a tree topology, as opposed to an ideal broadcast. As such, the communication overhead of PICO scales with respect to $O(N\log N)$ in the experiments, slightly higher than $O(N)$. This suggests PICO could in principle achieve even higher gains in an ideal broadcasting setting, such as a cellular network among devices within the same coverage area. The other hyperparameters are $J = 50$ and $\eta = 1.4 \times 10^{-7}$, respectively. For CIFAR-10, 9019 samples are used in the training set, and 1000 samples in the test set. Then, each local training set is complemented with simple random crop augmentation (to avoid having too few samples per client as the number of clients increase), leading to a total number of 18038 training samples. Similarly, for MNIST, 11432 samples are used for training, and 2115 samples for testing. Then, each local training set is complemented with random crop augmentation, leading to 22864 training samples. Model accuracy is evaluated on the test set, using the model trained jointly across the N clients.

We evaluate the performance with respect to both COPML [7] and conventional logistic regression. For PICO and COPML, we leverage the secure truncation protocol from [80] to carry out the multiplication with $\frac{\eta}{m}$ during the model update in (40), to ensure that the range of the updated model stays within the range of the finite field as suggested by [7]. This protocol takes as input the secret shares $\{[x]_i\}_{i\in[N]}$ of a variable x (where client i holds a share $[x]_i$), along with two public integer parameters κ_1 and κ_2 such that $0 < \kappa_1 < \kappa_2$, and $x \in \mathbb{F}_{2^{\kappa_2}}$. Then, the protocol returns the secret shares $\{[z]_i\}_{i\in[N]}$ of a variable z such that $z = \lfloor \frac{x}{2^{\kappa_1}} \rfloor + b$ where b is a Bernoulli random variable (random bit) with probability $P[b=1] = (x \mod 2^{\kappa_1})/2^{\kappa_1}$. As such, the secret x is quantized by rounding $x/(2^{\kappa_1})$ to the nearest integer with probability $1-\rho$, where ρ is the distance between the two. The quantization is unbiased, ensuring the convergence of the trained model. In the experiments, $(\kappa_1, \kappa_2) = (22, 24)$ is used for both datasets and benchmarks. We further optimize (speed up) COPML by leveraging the grouping strategy suggested in [7], which partitions clients into groups of size $T+1$, and communicates the secret shares only between clients within the same group. To ensure correct recovery of the final model, the number of clients (for both PICO and COPML) must satisfy the recovery threshold from Thm. 3. We then compare the performance under the same system configurations from [7] to ensure a fair comparison, by letting $r = 1$, and considering the scenario where the degree of privacy (T) and parallelization (K) are (almost) equal, such that $N = 3(K+T-1)+1$ with $T = \lfloor \frac{N-3}{6} \rfloor$ and $K = \lfloor \frac{N+2}{3} \rfloor - T$. The bandwidth and finite field size are set as 40Mbps and $q = 2^{26} - 5$, respectively.



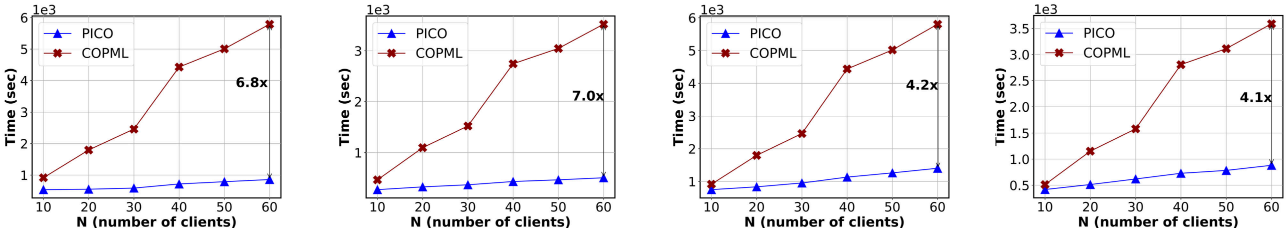
(a) Online (CIFAR-10). (b) Online (MNIST). (c) Online+offline (CIFAR-10). (d) Online+offline (MNIST).

Fig. 4. Online (a)-(b) and online+offline (c)-(d) communication overhead.

TABLE II

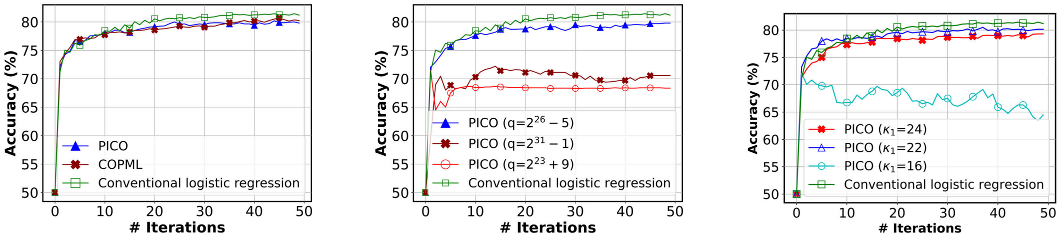
COMMUNICATION OVERHEAD (IN MBITS) ACROSS ALL CLIENTS FOR $N = 60$

Stage	CIFAR-10				MNIST			
	Online		Online + Offline		Online		Online + Offline	
	COPML	PICO	COPML	PICO	COPML	PICO	COPML	PICO
1. Dataset enc.	4.1×10^5	3.8×10^3	4.1×10^5	2.4×10^4	2.6×10^5	2.4×10^3	2.6×10^5	1.5×10^4
2. Label enc.	8.3×10^2	0.7×10^2	8.3×10^2	8.6×10^1	4.7×10^2	3.52×10^1	4.7×10^2	4.34×10^1
3. Model init.	-	-	7.6×10^2	1.4×10^1	-	-	3.8×10^2	0.74×10^1
4. Model enc.	2.9×10^3	3.2×10^2	2.9×10^3	1.1×10^3	5.8×10^3	6.5×10^2	5.8×10^3	2.1×10^3
5. Gradient	3.6×10^4	6.5×10^2	3.6×10^4	2.1×10^3	1.8×10^4	3.2×10^2	1.8×10^4	1.1×10^3
Total	4.6×10^5	5.2×10^3	4.6×10^5	2.9×10^4	2.8×10^5	3.1×10^3	2.8×10^5	1.7×10^4



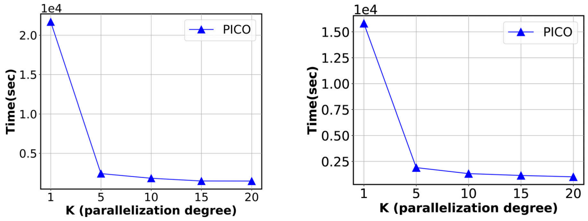
(a) CIFAR-10. (b) MNIST. (c) CIFAR-10. (d) MNIST.

Fig. 5. Online (a)-(b) and online+offline (c)-(d) wall-clock training time.



(a) Model test accuracy. (b) Varying finite field size (q). (c) Varying truncation level (κ_1).

Fig. 6. Model convergence (a), impact of finite field size (b), and secure truncation (quantization) level (c) on CIFAR-10.

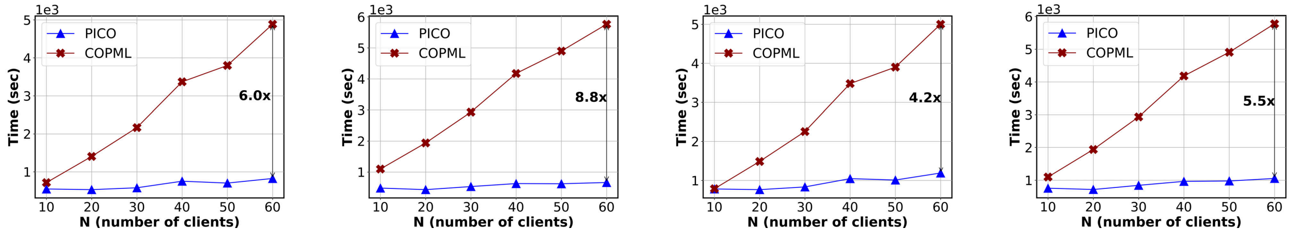


(a) CIFAR-10. (b) MNIST.

Fig. 7. Online+offline wall-clock training time.

We first compare the online communication overhead (in Mbits) in Fig. 4 (a)-(b), including all communication during the online phases throughout training. We observe that PICO significantly decreases the communication overhead, by up to 88.3× and 91.5× on CIFAR-10 and MNIST, respectively. Note that some one-time communications (i.e., secret sharing the dataset/labels) were omitted in [7], which we also include as they are data-dependent. In Fig. 4 (c)-(d), we compare the overall (online+offline) communication overhead, and observe a reduction by up to 15.8× on CIFAR-10 and 15.9× on MNIST. In Table II we provide the details of the online and overall (online+offline) communication overhead from Fig. 4 for $N = 60$ clients, where we illustrate the cost breakdown for each protocol component. Fig. 5 (a)-(b) compares the wall-clock training time of PICO and COPML, including all (online) communication and computations. We observe that PICO speeds-up the training time by up to 6.8× and 7× on CIFAR-10 and MNIST, respectively. In Fig. 5 (c)-(d), we present the overall wall-clock time by including both online and offline operations, and observe a reduction by up to 4.2× on CIFAR-10 and 4.1× on MNIST.

In Fig. 6(a), we compare the test accuracy of PICO for $N = 60$ and CIFAR-10 with respect to both COPML and



(a) Online (CIFAR-10). (b) Online (MNIST). (c) Online+offline (CIFAR-10). (d) Online+offline (MNIST).

Fig. 8. Online (a)-(b) and online+offline (c)-(d) wall-clock training time (maximum parallelization gain).

conventional logistic regression (representing our target accuracy), where for the latter training is done in the domain of real numbers, without any privacy constraints, in a centralized setting with all data located at a single party. We observe that PICO achieves comparable accuracy to both COPML and conventional logistic regression. In Fig. 6(b), we further evaluate the impact of the finite field size q, and in Fig. 6(c) we demonstrate the impact of the secure truncation parameter κ_1 on accuracy. We observe that accuracy degrades for very small κ_1, which increases the accuracy of quantization but also the overflow errors, hence there exists a trade-off between quantization and overflow errors. In practice, these hyperparameters can be tuned through a local validation set, where each client can locally identify a feasible range prior to training, after which clients can agree on the final parameters.

In Fig. 7, we demonstrate the role of parameter K on the overall (offline+online) wall-clock training time of PICO (including all communication and computations), by letting $N = 60$ and varying K. As K increases, training time decreases, as the size of the encoded dataset processed by each client is proportional to $1/K$ (reducing the training load per client). Fig. 7 also illustrates a trade-off between parallelization (accordingly, training time) and adversary resilience, as increasing K decreases the maximum number of adversaries T that can be tolerated, as shown in Thm. 3. Finally, we consider the scenario with the maximum parallelization gain (i.e., highest K), by setting $T = 1$ and selecting K to be the highest value that is allowed by the recovery threshold from Thm. 3. We then present the online and overall (offline+online) wall-clock training time in Fig. 8 for the two datasets. We observe that PICO significantly speeds up training by cutting the online wall-clock training time by up to $8.8\times$ and the overall (offline+online) wall-clock training time by up to $5.5\times$, respectively.

IX. Conclusion and Future Directions

This work presents PICO, the first collaborative learning framework with linear communication complexity, under strong information-theoretic privacy guarantees. PICO builds on an online-offline trade-off where the communication intensive operations are offloaded to a data-agnostic offline phase. Then, the amortized communication complexity for the latter is further reduced to linear via an efficient shared randomness generation mechanism. In doing so, PICO achieves an order of magnitude reduction in the communication overhead, while providing the same accuracy, dropout-resilience and privacy guarantees as the state-of-the-art. Future directions include expanding our mechanisms to different machine learning tasks and loss functions. Extending our work to more complex machine learning tasks, such as neural networks, necessitates addressing several key challenges, including the increase in the polynomial degree of coded computations as the number of layers increases, due to consecutive multiplication operations during forward and backpropagation, as well as handling the impact of consecutive polynomial approximations for the activation functions (e.g., ReLu activations), which can accumulate error as the number of layers increases. Addressing these challenges with efficient neural network architectures and training mechanisms is an interesting future direction. Another future direction is developing novel secure quantization mechanisms for multi-party machine learning, to enhance model accuracy under resource limitations.

Appendix A
Algorithm

The offline and online steps of PICO are presented in Algorithms 1 and 2, respectively. The offline phase consists of randomness generation across the N clients, which will later be used for masking the datasets, models, and computations in the online phase.

Appendix B
Communication Complexity

In the following, we analyze the per-client communication complexity of PICO.

A. Online

The online communication per-client consists of the following components: 1) $O(dm)$ for dataset encoding (Stage 1), 2) $O(\frac{Nd}{K})$ for label encoding (Stage 2), 3) $O(d)$ for model encoding (Stage 4) per training round, 4) $O(d)$ for gradient computing and model update (Stage 5) per training round.

B. Offline

The offline communication per-client consists of the following components: 1) $O(Nd\frac{m}{K})$ for dataset encoding (Stage 1), 2) $O(\frac{Nd}{(N-T)})$ for label encoding (Stage 2), 3) $O(\frac{Nd}{N-T})$ for model initialization (Stage 3), 4) $O(\frac{Nd}{N-T})$ for model encoding (Stage 4) per training round, 5) $O(\frac{Nd}{N-T})$ for gradient computing and model update (Stage 5) per training round.

Hence, the communication overhead per-client is $O(dm + \frac{N}{K}d + dJ)$ in the online phase, and $O(\frac{N}{K}dm + \frac{N}{N-T}dJ)$ in the

Algorithm 1 PICO - Offline Phas

```
Input: Number of clients $N$, polynomial coefficients $(\alpha_1, \dots, \alpha_N), (\beta_1, \dots, \beta_K)$.
Output: Random masks $\{\widetilde{\mathbf{R}}_{ij}\}_{i,j \in [N]}$, $\{[\mathbf{a}_i]_j\}_{i,j \in [N]}$, $\{\widetilde{\mathbf{r}}_i^{(t)}, [\mathbf{r}^{(t)}]_i, [\mathbf{u}_i^{(t)}]_j\}_{i \in [N], t \in \{0,\dots,J-1\}}$, random initial model $\{[\overline{\mathbf{w}}^{(0)}]_i\}_{i \in [N]}$.
// 1. Dataset Encoding
for client $i = 1, \dots, N$ do
    Encode the random matrices $\{\mathbf{R}_{ik}\}_{k \in [K]}$, $\{\mathbf{V}_{ik}\}_{k \in \{K+1,\dots,K+T\}}$ from (9).
    for $j = 1, \dots, N$ do
        Send the encoded matrix $\widetilde{\mathbf{R}}_{ij}$ to client $j$.
// 2. Label Encoding
for client $i = 1, \dots, N$ do
    Encode the random vectors $\{\mathbf{a}_{ik}\}_{k \in [K]}$, $\{\mathbf{b}_{ik}\}_{k \in \{K+1,\dots,K+T\}}$ from (13).
    for $j = 1, \dots, N$ do
        Send the encoded vector $\widetilde{\mathbf{a}}_{ij}$ and secret share $[\mathbf{a}_i]_j$ to client $j$ to client $j$.
for $i = 1, \dots, N$ do
    Construct the encoded vector $\tilde{\mathbf{a}}_i \triangleq (\mathbf{M} \otimes \mathbf{I}) \times (\widetilde{\mathbf{a}}_{1i}^T, \dots, \widetilde{\mathbf{a}}_{Ni}^T)^T$ from (14).
    Construct the secret share $[\mathbf{a}_k]_i \triangleq \mathbf{M} \times ([\mathbf{a}_{1k}]_i^T, \dots, [\mathbf{a}_{Nk}]_i^T)^T$ for all $k \in [K]$.
// 3. Model Initialization
for client $i = 1, \dots, N$ do
    Generate a random vector $\overline{\mathbf{w}}_i^{(0)}$ from $\mathbb{F}_q$.
    for $j = 1, \dots, N$ do
        Send a secret share $[\overline{\mathbf{w}}_i^{(0)}]_j$ to client $j$ using Shamir's secret sharing.
for client $i = 1, \dots, N$ do
    Initialize the model $[\overline{\mathbf{w}}^{(0)}]_i$ using $\{[\overline{\mathbf{w}}_j^{(0)}]_i\}_{j \in [N]}$ as given in (20).
for iteration $t = 0, \dots, J-1$ do
    // 4. Model Encoding
    for client $i = 1, \dots, N$ do
        Encode the random vectors $\mathbf{r}_i^{(t)}$, $\{\mathbf{v}_{ik}^{(t)}\}_{k \in \{K+1,\dots,K+T\}}$ as in (22).
        for $j = 1, \dots, N$ do
            Send the encoded vector $\widetilde{\mathbf{r}}_{ij}^{(t)}$ and secret share $[\mathbf{r}_i^{(t)}]_j$ to client $j$.
    for client $i = 1, \dots, N$ do
        Compute the coded vector, $\widetilde{\mathbf{r}}_i^{(t)}$ as given in (23).
        Compute the secret share $[\mathbf{r}^{(t)}]_i$ after receiving $\{[\mathbf{r}_j^{(t)}]_i\}_{j \in [N]}$ as given in (24).
    // 5. Gradient Computing and Model Update
    for client $i = 1, \dots, N$ do
        Encode $\{\mathbf{u}_{ik}^{(t)}\}_{k \in (2r+1)(K+T-1)+1}$ as given in (29).
        for $j = 1, \dots, N$ do
            Send the encoded vector $\widetilde{\mathbf{u}}_{ij}^{(t)}$ to client $j$.
            Send a secret share $[\sum_{k \in [K]} \mathbf{u}_{ik}^{(t)}]_j$ to client $j$ using Shamir's secret sharing.
    for client $i = 1, \dots, N$ do
        Compute the coded vector, $\widetilde{\mathbf{u}}_i^{(t)}$ after receiving $\{\widetilde{\mathbf{u}}_{ji}^{(t)}\}_{j \in [N]}$ as given in (31).
        Compute the secret share, $[\sum_{k \in [K]} \mathbf{u}_k^{(t)}]_i$ after receiving $\{[\sum_{k \in [K]} \mathbf{u}_{jk}^{(t)}]_i\}_{j \in [N]}$ from (33).
```

offline phase. The total communication complexity across all N clients is $O(Ndm + \frac{N^2}{K}d + NdJ)$ in the online phase, and $O(\frac{N^2}{K}dm + \frac{N^2}{N-T}dJ)$ in the offline phase.

C. Communication Complexity of PICO vs COPML

In Table III, we present the total communication complexity (across all N clients) of PICO versus COPML [7] for each stage. We observe that PICO incurs a linear communication overhead both in the online and offline phases. As such, PICO not only reduces the online communication overhead from quadratic point-to-point to linear broadcast (by offloading the communication-intensive operations to the offline phase), but also reduces the offline amortized communication overhead to linear, as opposed to the naive offloading strategy discussed in Section IV, where the quadratic communication overhead is offloaded to the offline phase, but the resulting offline overhead is still quadratic.

Appendix C
Computation Complexity

In the following we analyze the per-client computational overhead of each stage of PICO, for both the offline and online phases, respectively.

A. Offline Phase

The offline phase consists of encoding the local randomness generated by the clients, and random initialization of the model as follows.

Stage 1: Generation of $\{\widetilde{\mathbf{R}}_{ij}\}_{j \in [N]}$ requires evaluating a Lagrange polynomial of degree $K + T - 1$ at N

Algorithm 2 PICO - Online Phas

```
Input: Dataset $(\overline{\mathbf{X}}, \overline{\mathbf{y}}) = ((\overline{\mathbf{X}}_1, \overline{\mathbf{y}}_1), \ldots, (\overline{\mathbf{X}}_N, \overline{\mathbf{y}}_N))$ distributed over $N$ clients.
Output: Model parameters $\overline{\mathbf{w}}^{(J)}$ after $J$ training rounds.
// 1. Dataset Encoding
for client $i = 1, \ldots, N$ do
    Partition the dataset $\overline{\mathbf{X}}_i$ into $K$ equal-sized shards $(\overline{\mathbf{X}}_{i1}, \ldots, \overline{\mathbf{X}}_{iK})$.
    Broadcast the masked dataset $\widehat{\mathbf{X}}_{ik} = \overline{\mathbf{X}}_{ik} - \mathbf{R}_{ik}$ for $k \in [K]$.
for client $i = 1, \ldots, N$ do
    Generate the coded dataset $\widetilde{\mathbf{X}}_i$ from (11).
// 2. Label Encoding
for client $i = 1, \ldots, N$ do
    Partition $\overline{\mathbf{X}}_i^{\mathrm{T}} \overline{\mathbf{y}}_i$ into K equal-sized shards $(\overline{\mathbf{y}}_{i1}, \ldots, \overline{\mathbf{y}}_{iK})$.
    for client $j = 1, \ldots, N$ do
        Encode $\{\overline{\mathbf{y}}_{ik}\}_{k \in [K]}$ as described in (17), and send the encoded vector $\widetilde{\mathbf{y}}_{ij}$ to client $j$.
for client $i = 1, \ldots, N$ do
    Broadcast $\hat{\mathbf{a}}_i = \sum_{j \in [N]} \widetilde{\mathbf{y}}_{ji} - \widetilde{\mathbf{a}}_i$ from (18).
for client $i = 1, \ldots, N$ do
    Reconstruct $\sum_{j \in [N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k$ for all $k \in [K]$ using polynomial interpolation.
    Compute a secret share $[\overline{\mathbf{X}}^T \overline{\mathbf{y}}]_i$ of $\overline{\mathbf{X}}^T \overline{\mathbf{y}}$ as given in (19).
for iteration $t = 0, \ldots, J-1$ do
    // 4. Model Encoding
    for $i = 1, \ldots, N$ do
        Broadcast $[\widehat{\mathbf{w}}^{(t)}]_i$ from (25).
    for $i = 1, \ldots, N$ do
        Decode $\widehat{\mathbf{w}}^{(t)} \triangleq \overline{\mathbf{w}}^{(t)} - \mathbf{r}^{(t)}$ using polynomial interpolation.
        Compute the coded model $\widetilde{\mathbf{w}}_i^{(t)}$ in (27).
    // 5. Gradient Computing and Model Update
    for client $i = 1, \ldots, N$ do
        Compute the gradient $\widetilde{\mathbf{X}}_i^T \hat{g}(\widetilde{\mathbf{X}}_i \times \widetilde{\mathbf{w}}_i^{(t)})$.
        Broadcast $\hat{\mathbf{u}}_i^{(t)} = \widetilde{\mathbf{X}}_i^T \hat{g}(\widetilde{\mathbf{X}}_i \times \widetilde{\mathbf{w}}_i^{(t)}) - \widetilde{\mathbf{u}}_i^{(t)}$.
    for client $i = 1, \ldots, N$ do
        Decode $\psi(\beta_k) = \varphi(\beta_k) - \phi(\beta_k) = \overline{\mathbf{X}}_k^T \hat{g}(\overline{\mathbf{X}}_k \times \overline{\mathbf{w}}^{(t)}) - \mathbf{u}_k^{(t)}$ for $k \in [K]$ via polynomial interpolation.
        Compute a secret share $[\overline{\mathbf{X}}^T \hat{g}(\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)})]_i$ of the gradient $\overline{\mathbf{X}}^T \hat{g}(\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)})$ as given in (38).
        Update the model with $[\overline{\mathbf{w}}^{(t+1)}]_i$ from (40).
// Final Model Recovery
Collect the secret shares $[\overline{\mathbf{w}}^{(J)}]_i$ from any $T+1$ clients.
Decode the final model $\overline{\mathbf{w}}^{(J)}$ via polynomial interpolation.
```

TABLE III

COMPARISON OF THE TOTAL COMMUNICATION OVERHEAD (ACROSS ALL N CLIENTS) FOR PICO AND COPML WHERE $m_i = m$ FOR $i \in [N]$, $K = \Theta(N)$, AND $T = O(N)$

	COPML	PICO	
1. Dataset encoding	$O(N^2dm + Ndm)$	(online)	$O(Ndm)$
		(offline)	$O(Ndm)$
2. Label encoding	$O(N^2m + N^2d)$	(online)	$O(Nd)$
		(offline)	$O(Nd)$
3. Model initialization	$O(N^2d)$	(online)	–
		(offline)	$O(Nd)$
4. Model encoding	$O(N^2dJ)$	(online)	$O(NdJ)$
		(offline)	$O(NdJ)$
5. Gradient computing and model update	$O(N^2dJ)$	(online)	$O(NdJ)$
		(offline)	$O(NdJ)$

points. It is known that by leveraging efficient algebraic structures, interpolating a polynomial of degree κ (and evaluating it at κ points) has a computational complexity of $O(\kappa \log^2 \kappa \log\log \kappa)$ [6], [84]. As such, this stage has a complexity of $O(Nd\frac{m}{K} \log^2(K+T) \log\log(K+T))$ per client.

Stage 2: Computing $\{\widetilde{\mathbf{a}}_{ij}\}_{j \in [N]}$ requires evaluating a polynomial of degree $K + T - 1$ at N points, which has a computational complexity of $O(N\frac{d}{(N-T)K} \log^2(K + T) \log\log(K + T))$ per client. Computing $\widetilde{\mathbf{a}}_i$ in (14) has a complexity of $O(\frac{Nd}{K})$ per client (since only the non-zero terms should be multiplied due to the identity matrix). Computing the secret shares $\{[\mathbf{a}_{ik}]_j\}_{j \in [N]}$ for all $k \in [K]$ requires evaluating each of the K polynomials of degree T at N points, which has complexity $O(N\frac{d}{N-T} \log^2 T \log\log T)$ for each client. Evaluating the secret shares $\{[\mathbf{a}_k]_i\}_{k \in [K]}$ has an overhead of $O(Nd)$ per client.

Stage 3: Computing the secret share $\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{j \in [N]}$ requires evaluating a polynomial of degree T at N points, which has complexity $O(N\frac{d}{N-T}\log^2 T\log\log T)$ for each client. Finally, computation of the final secret share, $\overline{\mathbf{w}}^{(0)}$ from (20) has complexity $O(Nd)$ per client.

Stage 4: Computation of $\widetilde{\mathbf{r}}_{ij}^{(t)}$ requires evaluating a Lagrange polynomial of degree $K+T-1$ at N points, which has a complexity of $O(N\frac{d}{N-T}\log^2(K+T)\log\log(K+T))$ per client. Given $\{\widetilde{\mathbf{r}}_{ji}^{(t)}\}_{j\in[N]}$, the computation of $\widetilde{\mathbf{r}}_i^{(t)}$ from (23) has an overhead of $O(Nd)$ per client. Constructing the secret share $[\mathbf{r}_i^{(t)}]_j$ requires evaluating a polynomial of degree T at N points, which has complexity $O(N\frac{d}{N-T}\log^2 T\log\log T)$ for each client. Afterwards, creating the secret share $[\mathbf{r}^{(t)}]_i$ has a complexity of $O(Nd)$ per client. Overall, this stage has a per client computational overhead of $O(N\frac{d}{N-T}\log^2(K+T)\log\log(K+T)+Nd)$ per training round. For J rounds, this leads to an overhead of $O(JN\frac{d}{N-T}\log^2(K+T)\log\log(K+T)+JNd)$ per client.

Stage 5: Computing $\{\widetilde{\mathbf{u}}_{ij}^{(t)}\}_{j\in[N]}$ requires evaluating a Lagrange polynomial of degree $(2r+1)(K+T-1)$ at N points, which has a complexity of $O(N\frac{d}{N-T}\log^2 r(K+T)\log\log r(K+T))$ per client per training round. Given $\{\widetilde{\mathbf{u}}_{ji}^{(t)}\}_{j\in[N]}$, computation of $\widetilde{\mathbf{u}}_i^{(t)}$ in (31) has complexity of $O(Nd)$ per client per training round. Next, computing $\sum_{k\in[K]}\mathbf{u}_{ik}^{(t)}$ has a computational overhead of $O(K\frac{d}{N-T})$ per client per training round. Computing the secret shares $\{[\sum_{k\in[K]}\mathbf{u}_{ik}^{(t)}]_j\}_{j\in[N]}$ requires evaluating a polynomial of degree T at N points, which incurs a complexity of $O(N\frac{d}{N-T}\log^2 T\log\log T)$ per client per training round. Finally, given $\{[\sum_{k\in[K]}\mathbf{u}_{jk}^{(t)}]_i\}_{j\in[N]}$, the computation of $[\sum_{k\in[K]}\mathbf{u}_k^{(t)}]_i$ has complexity of $O(Nd)$ per client per training round. For J iterations, the computational complexity is $O(JN\frac{d}{N-T}\log^2 r(K+T)\log\log r(K+T)+JNd)$ per client.

Overall, the computation complexity of the offline phase is $O(Nd\frac{m}{K}\log^2(K+T)\log\log(K+T)+JN\frac{d}{N-T}\log^2 r(K+T)\log\log r(K+T)+JNd)$ per client.

B. Online Phase

The online phase consists of encoding the dataset and the model, gradient computations, and model update.

Stage 1: Computing $\{\widehat{\mathbf{X}}_{ik}\}_{k\in[K]}$ has an overhead of $O(md)$ per client, as each client holds a local dataset of size m locally. Computing $\widetilde{\mathbf{X}}_i$ has an overhead of $O(Nmd)$ per client.

Stage 2: Computation of $\overline{\mathbf{X}}_i^{\mathrm{T}}\overline{\mathbf{y}}_i$ has complexity of $O(md)$ per client. Computation of $\{\widetilde{\mathbf{y}}_{ij}\}_{j\in[N]}$ requires evaluation of a Lagrange polynomial of degree $K+T-1$ at N points, which has a complexity of $O(N\frac{d}{K}\log^2(K+T)\log\log(K+T))$ per client. Given $\{\widetilde{\mathbf{y}}_{ji}\}_{j\in[N]}$ and $\widetilde{\mathbf{a}}_i$ (from offline computation), computation of $\hat{\mathbf{a}}_i$ incurs a complexity of $O(N\frac{d}{K})$ per client. Next, upon receiving $\{\hat{\mathbf{a}}_j\}_{j\in[N]}$ from at least $K+T$ clients, client i recovers $\sum_{j\in[N]}\overline{\mathbf{y}}_{jk}-\mathbf{a}_k$ for all $k\in[K]$, which has a complexity of $O(\frac{d}{K}(K+T)\log^2(K+T)\log\log(K+T))$. Next, computation of $[\mathbf{X}^T\mathbf{y}]_i$ from (19) has a complexity of $O(d)$ per client.

Stage 4: Computing $\widehat{\mathbf{w}}^{(t)}$ requires interpolating a polynomial of degree T, which has a complexity of $O(Td\log^2 T\log\log T)$ per client per training round. Computing the encoded model $\widetilde{\mathbf{w}}_i^{(t)}$ has a computation overhead of $O(Kd)$ per client. As the above computation steps should be repeated at every training round, for a total number of J training iterations, the computational overhead is $O(KdJ+TdJ\log^2 T\log\log T)$ per client.

Stage 5: Computation of the gradient $\widetilde{\mathbf{X}}_i^T\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})$ has an overhead of $O(\frac{Nm}{K}(d+r))$ per client, at each training round. The computation of $\hat{\mathbf{u}}_i$ has an overhead of $O(d)$ per client. Then, each client recovers the polynomial $\psi(\alpha)$, which requires interpolating a polynomial of degree $(2r+1)(K+T-1)$, which has complexity $O(dr(K+T)\log^2 r(K+T)\log\log r(K+T))$ per client. Finally, the summation to obtain $[\overline{\mathbf{X}}^T\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)})]_i$ has a computational cost $O(Kd)$ per client. The computation overhead of model update is $O(d)$. The above computation steps are repeated over J training rounds. For J rounds, the computation complexity is $O(J\frac{Nm}{K}(d+r)+Jdr(K+T)\log^2 r(K+T)\log\log r(K+T))$ per client.

Overall, computation complexity of the online phase is $O(Nmd+N\frac{d}{K}\log^2(K+T)\log\log(K+T)+J\frac{Nm}{K}(d+r)+Jdr(K+T)\log^2 r(K+T)\log\log r(K+T))$ per client.

C. Computation Complexity of PICO vs COPML

In Table IV, we present the per-client computational complexity of PICO versus COPML [7] for each stage. For a fair comparison, we also consider the utilization of fast polynomial interpolation mechanisms [84] for COPML (hence the complexity we report is even lower than the one originally reported in [7]). In Table V, we present the per-client computational complexity for PICO (offline+online) and COPML, with $T=O(N)$ and $K=\Theta(N)$. We observe that the overall per-client complexity (across all algorithm steps) is $O(Ndm+dm\log^2 N\log\log N+JNd\log^2 N\log\log N+Jm(d+r))$ for PICO and $O(Ndm\log^2 N\log\log N+JNd\log^2 N\log\log N+Jm(d+r))$ for COPML, respectively. Hence, PICO achieves the same computation complexity as COPML. This is due to the fact that PICO reduces the overall number of variables encoded, hence the additional operations due to the matrix transformations with MDS matrices do not increase the overall computation complexity.

APPENDIX D
INFORMATION-THEORETIC PRIVACY

Proof: For tractability of theoretical analysis, in this section we consider a sufficiently large field size q, and treat all training operations as integer operations [2]. This can be achieved by considering a learning rate η such that $M\triangleq\overline{m}/\eta$ is an integer and redefining the gradient computation at client i from (35) as follows,

$$\varphi(\alpha_i)=\sum_{j=0}^{r}\theta_j M^{(r-j)a_t}\widetilde{\mathbf{X}}_i^{\mathrm{T}}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})^j \tag{42}$$

where we define the polynomial $\varphi(\alpha)=\sum_{j=0}^{r}\theta_j M^{(r-j)a_t}f(\alpha)^{\mathrm{T}}(f(\alpha)\times h(\alpha))^j$ such that client i computes $\varphi(\alpha_i)$, the exponent $(\cdot)^j$ is applied element-wise, and coefficient a_t is defined as,

$$a_t\triangleq\begin{cases}0 & \text{for } t=0\\ ra_{t-1}+1 & \text{for } t\geq 1\end{cases} \tag{43}$$

TABLE IV

COMPARISON OF THE COMPUTATION OVERHEAD (PER CLIENT) FOR PICO AND COPML WITH $m_i = m$ FOR $i \in [N]$

	COPML	PICO	
1. Dataset encoding	$O(N^2 \frac{m}{K} d \log^2(K+T) \log\log(K+T))$	(online)	$O(Ndm)$
		(offline)	$O(N \frac{m}{K} d \log^2(K+T) \log\log(K+T))$
2. Label encoding	$O(N(m+d) \log^2 T \log\log T)$	(online)	$O(N \frac{d}{K} \log^2(K+T) \log\log(K+T))$
		(offline)	$O(N \frac{d}{(N-T)K} \log^2(K+T) \log\log(K+T)$ $+ \frac{Nd}{K} + N \frac{d}{N-T} \log^2 T \log\log T)$
3. Model initialization	$O(Nd \log^2(K+T) \log\log(K+T))$	(online)	–
		(offline)	$O(N \frac{d}{N-T} \log^2 T \log\log T + Nd)$
4. Model encoding	$O(JNd \log^2(K+T) \log\log(K+T))$	(online)	$O(KdJ + TdJ \log^2 T \log\log T)$
		(offline)	$O(JN \frac{d}{N-T} \log^2(K+T) \log\log(K+T)$ $+JNd)$
5. Gradient comp./ model update	$O(J \frac{Nm}{K}(d+r) + Jdr(K+T)$ $\times \log^2 r(K+T) \log\log r(K+T))$	(online)	$O(J \frac{Nm}{K}(d+r) + Jdr(K+T)$ $\times \log^2 r(K+T) \log\log r(K+T))$
		(offline)	$O(JN \frac{d}{N-T} \log^2 r(K+T)$ $\times \log\log r(K+T) + JNd)$

TABLE V

COMPARISON OF THE COMPUTATION OVERHEAD (PER CLIENT) FOR PICO AND COPML WITH $m_i = m$ FOR $i \in [N]$, $K = \Theta(N)$, AND $T = O(N)$

	COPML	PICO (online+offline)
1. Dataset encoding	$O(Ndm \log^2 N \log\log N)$	$O(Ndm + dm \log^2 N \log\log N)$
2. Label encoding	$O(N(m+d) \log^2 N \log\log N)$	$O(d \log^2 N \log\log N)$
3. Model initialization	$O(Nd \log^2 N \log\log N)$	$O(d \log^2 N \log\log N + Nd)$
4. Model encoding	$O(JNd \log^2 N \log\log N)$	$O(JNd \log^2 N \log\log N)$
5. Gradient comp./ model update	$O(Jm(d+r) + JNd \log^2 N \log\log N)$	$O(Jm(d+r) + JNd \log^2 N \log\log N))$

whereas the true gradient is given by,

$$\sum_{k\in[K]} \varphi(\beta_k) = \sum_{j=0}^{r} \theta_j M^{(r-j)a_t} (\overline{\mathbf{X}}'_k)^{\mathrm{T}} (\overline{\mathbf{X}}'_k \times \overline{\mathbf{w}}^{(t)})^j$$
$$= \sum_{j=0}^{r} \theta_j M^{(r-j)a_t} \overline{\mathbf{X}}^{\mathrm{T}} (\overline{\mathbf{X}} \times \overline{\mathbf{w}}^{(t)})^j \quad (44)$$

such that $\overline{\mathbf{X}}'_k \triangleq f(\beta_k) = \begin{bmatrix} \overline{\mathbf{X}}_{1k}^{\mathrm{T}} & \cdots & \overline{\mathbf{X}}_{Nk}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ from (12), replacing (35) and (36), respectively. After collecting $\hat{\mathbf{u}}_i^{(t)} = \psi(\alpha_i)$ from any set of at least $C+1$ clients, client i can recover $\psi(\alpha)$ via polynomial interpolation, compute a secret share of the gradient $\sum_{k\in[K]} \varphi(\beta_k)$,

$$\Big[\sum_{k\in[K]} \varphi(\beta_k)\Big]_i \triangleq \sum_{k\in[K]} \psi(\beta_k) + \Big[\sum_{k\in[K]} \mathbf{u}_k^{(t)}\Big]_i \quad (45)$$
$$= \sum_{k\in[K]} \varphi(\beta_k) + \sum_{l\in[T]} \gamma_i^l \mathbf{z}_l^{(t)} \quad (46)$$

and update the model as,

$$[\overline{\mathbf{w}}^{(t+1)}]_i = M^{(r-1)a_t+1} [\overline{\mathbf{w}}^{(t)}]_i - \bigg(\Big[\sum_{k\in[K]} \varphi(\beta_k)\Big]_i - M^{ra_t} [\overline{\mathbf{X}}^{\mathrm{T}} \overline{\mathbf{y}}]_i\bigg). \quad (47)$$

replacing the model update operation from (40). After J training rounds, clients collect the secret shares $\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in[N]}$ to decode $\overline{\mathbf{w}}^{(J)}$, and compute the final model as $\mathbf{w}^{(J)} \leftarrow \overline{\mathbf{w}}^{(J)}/M^{a_J}$. The correctness of the model update operations from (47) are provided in Appendix E.

We next present the information-theoretic privacy analysis for PICO. Consider an arbitrary set of adversaries $\mathcal{T} \subseteq N$. For ease of exposition, we focus on the worst case scenario by setting $|\mathcal{T}| = T$, while noting that the same analysis holds for all $|\mathcal{T}| < T$. Let $\mathcal{M}_{\mathcal{T}}^1$ and $\mathcal{M}_{\mathcal{T}}^2$, denote the collection of all messages received by the adversaries during the dataset encoding (Stage 1), and label encoding (Stage 2) stages, respectively. Let $\mathcal{M}_{\mathcal{T}}^3$ denote the collection of all messages received by the adversaries during model initialization stage (Stage 3). Similarly, let $\mathcal{M}_{\mathcal{T}}^{4,t}$ denote the collection of all messages received by the adversaries in model encoding stage (Stage 4) at training round $t \in \{0, \ldots, J-1\}$. Let $\mathcal{M}_{\mathcal{T}}^{5,t}$ denote the collection of all messages received by the adversaries during the gradient computing and model update stage (Stage 5) at training round $t \in \{0, \ldots, J-1\}$. Finally, let $\mathcal{M}_{\mathcal{T}}^6$ denote the collection of all messages received by the adversaries during the reconstruction of the final model $\overline{\mathbf{w}}^{(J)}$ after J training rounds. Then, from the chain rule of mutual information [71], one can rewrite (41) as follows:

$$I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$
$$= I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{t\in[J]} \mathcal{M}_{\mathcal{T}}^{4,t}, \cup_{t\in[J]} \mathcal{M}_{\mathcal{T}}^{5,t}, \mathcal{M}_{\mathcal{T}}^6 | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \quad (48)$$
$$= I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^1 | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$
$$+ I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^2 | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$
$$+ I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^3 | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$
$$+ \sum_{t=0}^{J-1} I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^{4,t} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1} \mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1} \mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$
$$+ \sum_{t=0}^{J-1} I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^{5,t} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t} \mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1} \mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$

$$+ I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^6 | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{J-1} \mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{J-1} \mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \tag{49}$$

We next investigate each term in the summation (49).

Stage 1: Dataset Encoding: First, we start with the first term in (49), which corresponds to Stage 1 of PICO, i.e., encoding the datasets. For this stage, the first term in the right hand side of (49) can be written as:

$$I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^1 | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$

$$= I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \tag{50}$$

$$= H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$
$$- H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}) \tag{51}$$

We next bound the first term in (51) as follows:

$$H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})$$

$$= H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in\mathcal{H}\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \tag{52}$$

$$\leq H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in\mathcal{H}\\ k\in[K]}}) \tag{53}$$

$$\leq \log\left(q^{(\sum_{i\in\mathcal{H}} \frac{Tdm_i}{K}) + (\sum_{i\in\mathcal{T}} dm_i) + (\sum_{i\in\mathcal{T}} \frac{Tdm_i}{K}) + (\sum_{i\in\mathcal{H}} dm_i)}\right) \tag{54}$$

$$= d\Big(\frac{T}{K} + 1\Big)\Big(\sum_{i\in[N]} m_i\Big) \log q \tag{55}$$

where (53) holds since conditioning cannot increase entropy. Equation (54) follows from the fact that uniform distribution maximizes entropy, and that the entropy of a uniform random variable distributed over an alphabet $\mathcal{A}$ is equal to $\log|\mathcal{A}|$. For the second term in (51), we find that,

$$H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})$$

$$= H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}} | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}) \tag{56}$$

$$= H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}) \tag{57}$$

$$= H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}} | \{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}})$$
$$+ H(\{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}} | \{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}) + H(\{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}) \tag{58}$$

$$= H(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T}\\ i\in\mathcal{H}}} | \{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}})$$
$$+ H(\{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}) + H(\{\mathbf{R}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}) \tag{59}$$

$$= H\Big(\Big\{ \sum_{k=K+1}^{K+T} \mathbf{V}_{ik} \prod_{l\in[K+T]\backslash\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \Big\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}\Big)$$
$$+ \log(q^{Td\sum_{i\in\mathcal{T}} \frac{m_i}{K}}) + \log(q^{Kd\sum_{i\in[N]} \frac{m_i}{K}}) \tag{60}$$

$$= \sum_{i\in\mathcal{H}} H\Big(\Big\{ \sum_{k=K+1}^{K+T} \mathbf{V}_{ik} \prod_{l\in[K+T]\backslash\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \Big\}_{j\in\mathcal{T}}\Big)$$
$$+ \frac{Td}{K}(\sum_{i\in\mathcal{T}} m_i) \log q + d(\sum_{i\in[N]} m_i) \log q \tag{61}$$

$$= \sum_{i\in\mathcal{H}} H(\{\mathbf{Z}_{ij}\}_{j\in\mathcal{T}}) + \frac{Td}{K}(\sum_{i\in\mathcal{T}} m_i) \log q + d(\sum_{i\in[N]} m_i) \log q \tag{62}$$

where (56) holds since given $\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}$, there is no uncertainty remaining in $\{X_{ik}\}_{i\in[N], k\in[K]}$, (57) holds since the generated randomness is independent from the local datasets, (58) follows from the chain rule of entropy, (59) holds since the random matrices are generated independently where each element is distributed uniformly at random (and independent from other elements) from the finite field $\mathbb{F}_q$. In (62), we define:

$$\mathbf{Z}_{ij} \triangleq \sum_{k=K+1}^{K+T} \mathbf{V}_{ik} \prod_{l\in[K+T]\backslash\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \tag{63}$$

for all $i \in \mathcal{H}$ and $j \in \mathcal{T}$. In the following, without loss of generality we let the first $N - T$ clients be honest (the last T clients are adversarial), i.e., $\mathcal{H} = [N - T]$ and $\mathcal{T} = \{N - T + 1, \ldots, N\}$. The assumption is for notational simplicity, and the same analysis holds for any set of adversarial clients $\mathcal{T}$ of size T. We also represent the Lagrange polynomial coefficients as:

$$\rho_{jk} \triangleq \prod_{l\in[K+T]\backslash\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \tag{64}$$

for all $j \in [N]$ and $k \in [K + T]$. Then, from (63), one can write:

$$\begin{bmatrix} \mathbf{Z}_{i,N-T+1} & \cdots & \mathbf{Z}_{i,N} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_{i,K+1} & \cdots & \mathbf{V}_{i,K+T} \end{bmatrix} \mathbf{\Gamma} \tag{65}$$

where

$$\mathbf{\Gamma} \triangleq \begin{bmatrix} \rho_{N-T+1,K+1} & \cdots & \rho_{N,K+1} \\ \vdots & \ddots & \vdots \\ \rho_{N-T+1,K+T} & \cdots & \rho_{N,K+T} \end{bmatrix} \tag{66}$$

is a $T \times T$ MDS matrix (hence is invertible), which follows from the MDS property of Lagrange coding as shown in [6]. An MDS matrix guarantees that (65) is a bijective mapping, hence,

$$H(\{\mathbf{Z}_{ij}\}_{j\in\mathcal{T}}) = H(\mathbf{Z}_{i,N-T+1}, \ldots, \mathbf{Z}_{i,N}) \tag{67}$$
$$= H(\mathbf{V}_{i,K+1}, \ldots, \mathbf{V}_{i,K+T}) \tag{68}$$

$$= \frac{Tdm_i}{K}\log q \tag{69}$$

where (68) is from (65) and that $\mathbf{\Gamma}$ is an MDS matrix, and (69) holds since each element of $\mathbf{V}_{ik}$ is distributed uniformly at random over $\mathbb{F}_q$. By combining (69) with (62), we have:

$$\begin{aligned}
&H\Big(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T},\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\dots,K+T\}}},\\
&\quad \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}\Big|\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\Big)\\
&= \Big(\sum_{i\in\mathcal{H}}\frac{Tdm_i}{K}\log q\Big) + \frac{Td}{K}\Big(\sum_{i\in\mathcal{T}} m_i\Big)\log q + d\Big(\sum_{i\in\mathcal{T}} m_i\Big)\log q \qquad (70)\\
&= d\Big(\frac{T}{K}+1\Big)\Big(\sum_{i\in[N]} m_i\Big)\log q \qquad (71)
\end{aligned}$$

Finally, by combining (54) and (71) with (51), we have:

$$\begin{aligned}
&0 \le I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^1 | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \qquad (72)\\
&= H\Big(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T},\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\dots,K+T\}}},\\
&\quad \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}\Big|\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}\Big)\\
&\quad - H\Big(\{\widetilde{\mathbf{R}}_{ij}\}_{\substack{j\in\mathcal{T},\\ i\in\mathcal{H}}}, \{\mathbf{R}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in[K]}}, \{\mathbf{V}_{ik}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\dots,K+T\}}},\\
&\quad \{\widehat{\mathbf{X}}_{ik}\}_{\substack{i\in[N]\\ k\in[K]}}\Big|\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\Big) \qquad (73)\\
&\le d\Big(\frac{T}{K}+1\Big)\Big(\sum_{i\in[N]} m_i\Big)\log q - d\Big(\frac{T}{K}+1\Big)\Big(\sum_{i\in[N]} m_i\Big)\log q \qquad (74)\\
&= 0 \qquad (75)
\end{aligned}$$

where the first inequality follows from the non-negativity of mutual information. Therefore, the first term in (49) satisfies the following:

$$I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^1 | \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) = 0 \tag{76}$$

Stage 2: Label Encoding: We next consider the second term in (49), which corresponds to the secret sharing of the labels. Without loss of generality, we represent the secret share of $\mathbf{a}_{ik}$ from client i to client j as follows:

$$[\mathbf{a}_{ik}]_j \triangleq \mathbf{a}_{ik} + \sum_{l\in[T]} \gamma_j^l \mathbf{e}_{ikl} \tag{77}$$

where $\mathbf{e}_{ikl}$ are random vectors of size $\frac{d}{K}$, where each element is distributed independently and uniformly at random from $\mathbb{F}_q$. Coefficients $\{\gamma_i\}_{i\in[N]}$ are distinct public parameters agreed in advance between all N clients, where $\gamma_i \in \mathbb{F}_q$ for all $i \in [N]$ such that $\{\gamma_i\}_{i\in[N]} \cap \{\beta_k\}_{k\in[K+T]} \cap \{\alpha_j\}_{j\in[N]} = \emptyset$. Using (77), we can rewrite the second term in (49) as follows:

$$\begin{aligned}
&I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^2 | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\
&= I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]}, \{\hat{\mathbf{a}}_i\}_{i\in[N]},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \qquad (78)\\
&= I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]},\\
&\quad \{\sum_{j\in[N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k\}_{k\in[K]}, \{\sum_{j\in[N]} \mathbf{r}_{jk} - \mathbf{b}_k\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \qquad (79)\\
&= H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]},\\
&\quad \{\sum_{j\in[N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k\}_{k\in[K]}, \{\sum_{j\in[N]} \mathbf{r}_{jk} - \mathbf{b}_k\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\
&\quad - H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]}, \{\sum_{j\in[N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k\}_{k\in[K]},\\
&\quad \{\sum_{j\in[N]} \mathbf{r}_{jk} - \mathbf{b}_k\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}) \qquad (80)
\end{aligned}$$

where (79) follows from the fact that any polynomial of degree $K+T-1$ can be determined from at least $K+T$ evaluation points, therefore there is a bijective mapping from any feasible set $\{\hat{\mathbf{a}}_i\}_{i\in[N]}$ to a set of $K+T$ coefficients $\{\sum_{j\in[N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k\}_{k\in[K]}, \{\sum_{j\in[N]} \mathbf{r}_{jk} - \mathbf{b}_k\}_{k\in\{K+1,\dots,K+T\}}$. For the second term in (80), we find that,

$$\begin{aligned}
&H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]},\\
&\quad \{\sum_{j\in[N]} \overline{\mathbf{y}}_{jk} - \mathbf{a}_k\}_{k\in[K]}, \{\sum_{j\in[N]} \mathbf{r}_{jk} - \mathbf{b}_k\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})\\
&= H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]}, \{\mathbf{a}_k\}_{k\in[K]},\\
&\quad \{\sum_{j\in[N]} \mathbf{r}_{jk} - \mathbf{b}_k\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]},\\
&\quad \overline{\mathbf{w}}^{(J)}) \qquad (81)\\
&= H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]},\\
&\quad \Big\{(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}\mathbf{a}_{1k}^{\mathrm{T}} & \cdots & \mathbf{a}_{Nk}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[K]},\\
&\quad \Big\{\sum_{j\in[N]} \mathbf{r}_{jk} - (\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}\mathbf{b}_{1k}^{\mathrm{T}} & \cdots & \mathbf{b}_{Nk}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]},\\
&\quad \overline{\mathbf{w}}^{(J)}) \qquad (82)\\
&= H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]},\\
&\quad \Big\{(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{a}_{1k}^{\mathrm{T}} & \cdots & \mathbf{a}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[K]}, \Big\{\sum_{j\in[N-T]} \mathbf{r}_{jk} -\\
&\quad (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{b}_{1k}^{\mathrm{T}} & \cdots & \mathbf{b}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in\{K+1,\dots,K+T\}},\\
&\quad \{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T}, k'\in[K], l\in[T]\\ k\in\{K+1,\dots,K+T\}}} | \mathcal{M}_{\mathcal{T}}^1, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]},\\
&\quad \overline{\mathbf{w}}^{(J)}) \qquad (83)\\
&= H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H}, j\in\mathcal{T}, k\in[K]},
\end{aligned}$$

$$\begin{aligned}&\Big\{ \left[\mathbf{a}_{1k}^{\mathrm{T}} \quad \cdots \quad \mathbf{a}_{(N-T)k}^{\mathrm{T}}\right]^{\mathrm{T}} \Big\}_{k\in[K]}, \Big\{ \sum_{j\in[N-T]} \mathbf{r}_{jk} \\ &- (\overline{\mathbf{M}} \otimes \mathbf{I}) \left[\mathbf{b}_{1k}^{\mathrm{T}} \quad \cdots \quad \mathbf{b}_{(N-T)k}^{\mathrm{T}}\right]^{\mathrm{T}} \Big\}_{k\in\{K+1,\ldots,K+T\}}) \\ &+ H(\{\mathbf{r}_{ik}, \mathbf{b}_{ik}, \mathbf{a}_{ik'}, \mathbf{e}_{ik'l}\}_{i\in\mathcal{T},k'\in[K],l\in[T],k\in\{K+1,\ldots,K+T\}}) \end{aligned} \tag{84}$$

$$\begin{aligned}= &H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]}, \{\mathbf{a}_{jk}\}_{j\in[N-T],k\in[K]}, \\ &\{ \sum_{j\in[N-T]} \mathbf{r}_{jk} - (\overline{\mathbf{M}} \otimes \mathbf{I})\, \overline{\mathbf{b}}_k\}_{k\in\{K+1,\ldots,K+T\}}) \\ &+ Td\Big(\frac{T}{K} + \frac{T}{K(N-T)} + \frac{1}{N-T} + \frac{T}{N-T}\Big) \log q \end{aligned} \tag{85}$$

where (81) follows from the fact that given $\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}$, there is no uncertainty in $\sum_{j\in[N]} \overline{\mathbf{y}}_{jk}$ for all $k \in [K]$. In (83), we define the following square submatrix of $\mathbf{M}$ from (16),

$$\overline{\mathbf{M}} \triangleq \begin{bmatrix} 1 & \lambda_1 & \ldots & \lambda_1^{N-T-1} \\ 1 & \lambda_2 & \ldots & \lambda_2^{N-T-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_{N-T} & \ldots & \lambda_{N-T}^{N-T-1} \end{bmatrix} \tag{86}$$

which is an $(N-T) \times (N-T)$ MDS matrix (hence is invertible), from which (84) follows. Equation (85) follows from the entropy of uniform random variables, and,

$$\overline{\mathbf{b}}_k \triangleq \begin{bmatrix} \mathbf{b}_{1k} \\ \vdots \\ \mathbf{b}_{(N-T)k} \end{bmatrix} \tag{87}$$

For the first term in (85), we find that,

$$\begin{aligned}&H(\{\widetilde{\mathbf{a}}_{ij}, [\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]}, \{\mathbf{a}_{jk}\}_{j\in[N-T],k\in[K]}, \\ &\quad\{ \sum_{j\in[N-T]} \mathbf{r}_{jk} - (\overline{\mathbf{M}} \otimes \mathbf{I})\, \overline{\mathbf{b}}_k\}_{k\in\{K+1,\ldots,K+T\}}) \\ &= H(\{\widetilde{\mathbf{a}}_{ij}\}_{i\in\mathcal{H},j\in\mathcal{T}}, \\ &\quad\{ \sum_{j\in[N-T]} \mathbf{r}_{jk} - (\overline{\mathbf{M}} \otimes \mathbf{I})\, \overline{\mathbf{b}}_k\}_{k\in\{K+1,\ldots,K+T\}} \\ &\quad|\{[\mathbf{a}_{ik}]_j\}_{\substack{i\in\mathcal{H},j\in\mathcal{T},\\ k\in[K]}}, \{\mathbf{a}_{jk}\}_{\substack{j\in[N-T],\\ k\in[K]}}) \\ &\quad+ H(\{[\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]}|\{\mathbf{a}_{jk}\}_{j\in[N-T],k\in[K]}) \\ &\quad+ H(\{\mathbf{a}_{jk}\}_{j\in[N-T],k\in[K]}) \qquad (88) \\ &= H\Bigg(\Big\{ \sum_{k=K+1}^{K+T} \mathbf{b}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \Big\}_{i\in\mathcal{H},j\in\mathcal{T}}, \\ &\quad\Big\{ \sum_{j\in[N-T]} \mathbf{r}_{jk} - (\overline{\mathbf{M}} \otimes \mathbf{I})\, \overline{\mathbf{b}}_k \Big\}_{k\in\{K+1,\ldots,K+T\}} \Bigg) \\ &\quad+ H\Bigg(\{ \sum_{l\in[T]} \gamma_j^l \mathbf{e}_{ikl}\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]} \Bigg) \\ &\quad+ H(\{\mathbf{a}_{jk}\}_{j\in[N-T],k\in[K]}) \qquad (89) \end{aligned}$$

where (88) follows from the chain rule of entropy, and (89) holds since the random vectors are generated independently. To simplify the analysis of (89), we let,

$$\begin{aligned}&\left[\textstyle\sum_{l\in[T]} \gamma_{N-T+1}^l \mathbf{e}_{ikl} \quad \cdots \quad \sum_{l\in[T]} \gamma_N^l \mathbf{e}_{ikl}\right] \\ &= \left[\mathbf{e}_{ik1} \quad \cdots \quad \mathbf{e}_{ikT}\right] \mathbf{A} \end{aligned} \tag{90}$$

where

$$\mathbf{A} \triangleq \begin{bmatrix} \gamma_{N-T+1}^1 & \cdots & \gamma_N^1 \\ \vdots & \ddots & \vdots \\ \gamma_{N-T+1}^{\mathrm{T}} & \cdots & \gamma_N^{\mathrm{T}} \end{bmatrix} \tag{91}$$

is an $T \times T$ MDS matrix (invertible). From (90), it follows for the second term in (89) that,

$$\begin{aligned}&H\Bigg(\{ \sum_{l\in[T]} \gamma_j^l \mathbf{e}_{ikl}\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]} \Bigg) \\ &= \sum_{i\in\mathcal{H}} \sum_{k\in[K]} H(\{ \sum_{l\in[T]} \gamma_j^l \mathbf{e}_{ikl}\}_{j\in\mathcal{T}}) \qquad (92) \\ &= \sum_{i\in[N-T]} \sum_{k\in[K]} H(\{ \sum_{l\in[T]} \gamma_j^l \mathbf{e}_{ikl}\}_{j\in\{N-T+1,\ldots,N\}}) \qquad (93) \\ &= \sum_{i\in[N-T]} \sum_{k\in[K]} H\Big(\left[\mathbf{e}_{ik1} \quad \cdots \quad \mathbf{e}_{ikT}\right] \mathbf{A} \Big) \qquad (94) \\ &= \sum_{i\in[N-T]} \sum_{k\in[K]} H(\mathbf{e}_{ik1}, \ldots, \mathbf{e}_{ikT}) \qquad (95) \\ &= (N-T)KT \frac{d}{(N-T)K} \log q \qquad (96) \\ &= Td \log q \qquad (97) \end{aligned}$$

where (92) is from the independence of the generated random variables, (94) follows from (90), and (95) holds since matrix $\mathbf{A}$ is invertible, hence represents a bijective mapping. Finally, (96) follows from the entropy of uniform random variables. Similarly, for the last term in (89),

$$H(\{\mathbf{a}_{jk}\}_{j\in[N-T],k\in[K]}) = (N-T)K \frac{d}{(N-T)K} \log q = d \log q \tag{98}$$

which also follows from the entropy of uniform random variables.

For the first term in (89), we rewrite $\Big\{ \sum_{k=K+1}^{K+T} \mathbf{b}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \Big\}_{j\in\mathcal{T}}$ as:

$$\begin{aligned}&\Big[\sum_{k=K+1}^{K+T} \mathbf{b}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_{N-T+1} - \beta_l}{\beta_k - \beta_l} \\ &\quad \cdots \quad \sum_{k=K+1}^{K+T} \mathbf{b}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_N - \beta_l}{\beta_k - \beta_l} \Big] \\ &\quad = \left[\mathbf{b}_{i,K+1} \quad \cdots \quad \mathbf{b}_{i,K+T}\right] \mathbf{\Gamma} \end{aligned} \tag{99}$$

where $\mathbf{\Gamma}$ is the $T \times T$ MDS matrix from (66) (hence is invertible). Using (99), one can then rewrite the first term in (89) as:

$$\begin{aligned}&H\Bigg(\Big\{ \sum_{k=K+1}^{K+T} \mathbf{b}_{ik} \prod_{l\in[K+T]\setminus\{k\}} \frac{\alpha_j - \beta_l}{\beta_k - \beta_l} \Big\}_{i\in\mathcal{H},j\in\mathcal{T}}, \\ &\quad\Big\{ \sum_{j\in[N-T]} \mathbf{r}_{jk} - (\overline{\mathbf{M}} \otimes \mathbf{I})\, \overline{\mathbf{b}}_k \Big\}_{k\in\{K+1,\ldots,K+T\}} \Bigg) \\ &= H\Bigg(\Big\{ \left[\mathbf{b}_{i,K+1} \quad \cdots \quad \mathbf{b}_{i,K+T}\right] \mathbf{\Gamma} \Big\}_{i\in\mathcal{H}}, \end{aligned}$$

$$\Big\{\sum_{j\in[N-T]}\mathbf{r}_{jk}-(\overline{\mathbf{M}}\otimes\mathbf{I})\;\overline{\mathbf{b}}_k\Big\}_{k\in\{K+1,\ldots,K+T\}}\bigg) \tag{100}$$

$$\begin{aligned}&=H\bigg(\{\mathbf{b}_{i,K+1},\ldots,\mathbf{b}_{i,K+T}\}_{i\in\mathcal{H}},\\&\quad\Big\{\sum_{j\in[N-T]}\mathbf{r}_{jk}-(\overline{\mathbf{M}}\otimes\mathbf{I})\;\overline{\mathbf{b}}_k\Big\}_{k\in\{K+1,\ldots,K+T\}}\bigg)\end{aligned} \tag{101}$$

$$\begin{aligned}&=H\bigg(\{\mathbf{b}_{i,k}\}_{i\in\mathcal{H},k\in\{K+1,\ldots,K+T\}},\\&\qquad\Big\{\sum_{j\in[N-T]}\mathbf{r}_{jk}-(\overline{\mathbf{M}}\otimes\mathbf{I})\;\overline{\mathbf{b}}_k\Big\}_{k\in\{K+1,\ldots,K+T\}}\bigg)\end{aligned} \tag{102}$$

$$\begin{aligned}&=H\bigg(\Big\{\sum_{j\in[N-T]}\mathbf{r}_{jk}-(\overline{\mathbf{M}}\otimes\mathbf{I})\;\overline{\mathbf{b}}_k\Big\}_{k\in\{K+1,\ldots,K+T\}}\\&\qquad\Big|\{\mathbf{b}_{i,k}\}_{i\in[N-T],k\in\{K+1,\ldots,K+T\}}\bigg)\\&\qquad\qquad+H(\{\mathbf{b}_{i,k}\}_{i\in[N-T],k\in\{K+1,\ldots,K+T\}})\end{aligned} \tag{103}$$

$$\begin{aligned}&=H(\{\sum_{j\in[N-T]}\mathbf{r}_{jk}\}_{k\in\{K+1,\ldots,K+T\}}\\&\quad|\{\mathbf{b}_{i,k}\}_{i\in[N-T],k\in\{K+1,\ldots,K+T\}})\\&\quad+H(\{\mathbf{b}_{i,k}\}_{i\in[N-T],k\in\{K+1,\ldots,K+T\}})\end{aligned} \tag{104}$$

$$\begin{aligned}&=H(\{\sum_{j\in[N-T]}\mathbf{r}_{jk}\}_{k\in\{K+1,\ldots,K+T\}})\\&\quad+H(\{\mathbf{b}_{i,k}\}_{i\in[N-T],k\in\{K+1,\ldots,K+T\}})\end{aligned} \tag{105}$$

$$=T\frac{d}{K}\log q+(N-T)T\frac{d}{(N-T)K}\log q \tag{106}$$

$$=\frac{2Td}{K}\log q \tag{107}$$

where (101) holds since $\mathbf{\Gamma}$ is invertible, representing a bijective mapping. Equation (103) follows from the chain rule of entropy, (104) holds since given $\{\mathbf{b}_{i,k}\}_{i\in[N-T],k\in\{K+1,\ldots,K+T\}}$, there is no uncertainty in $(\overline{\mathbf{M}}\otimes\mathbf{I})\;\overline{\mathbf{b}}_k$, (105) follows from the independence of the random vectors, and (106) follows from the entropy of uniform random variables. By combining (107), (97), and (98), with (85), we can rewrite the second term in (80) as follows,

$$\begin{aligned}&H(\{\widetilde{\mathbf{a}}_{ij},[\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]},\{\sum_{j\in[N]}\overline{\mathbf{y}}_{jk}-\mathbf{a}_k\}_{k\in[K]},\\&\{\sum_{j\in[N]}\mathbf{r}_{jk}-\mathbf{b}_k\}_{k\in\{K+1,\ldots,K+T\}},\\&\{\mathbf{r}_{ik},\mathbf{b}_{ik},\mathbf{a}_{ik'},\mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T},k'\in[K],l\in[T]\\k\in\{K+1,\ldots,K+T\}}}|\mathcal{M}^1_{\mathcal{T}},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)})\\&=\frac{2Td}{K}\log q+Td\log q+d\log q\\&\quad+Td\Big(\frac{T}{K}+\frac{T}{K(N-T)}+\frac{1}{N-T}+\frac{T}{N-T}\Big)\log q\end{aligned} \tag{108}$$

$$\begin{aligned}&=\Big(d\Big(\frac{2T}{K}+T+1\Big)\\&\quad+Td\Big(\frac{T}{K}+\frac{T}{K(N-T)}+\frac{1}{N-T}+\frac{T}{N-T}\Big)\Big)\log q\end{aligned} \tag{109}$$

Next, for the first term in (80), we observe that,

$$\begin{aligned}&H(\{\widetilde{\mathbf{a}}_{ij},[\mathbf{a}_{ik}]_j\}_{i\in\mathcal{H},j\in\mathcal{T},k\in[K]},\{\sum_{j\in[N]}\overline{\mathbf{y}}_{jk}-\mathbf{a}_k\}_{k\in[K]},\\&\quad\{\sum_{j\in[N]}\mathbf{r}_{jk}-\mathbf{b}_k\}_{k\in\{K+1,\ldots,K+T\}},\\&\qquad\{\mathbf{r}_{ik},\mathbf{b}_{ik},\mathbf{a}_{ik'},\mathbf{e}_{ik'l}\}_{\substack{i\in\mathcal{T},k'\in[K],l\in[T]\\k\in\{K+1,\ldots,K+T\}}}|\mathcal{M}^1_{\mathcal{T}},\\&\qquad\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\\&\le\Big(\frac{(N-T)Td}{(N-T)K}+\frac{(N-T)TdK}{(N-T)K}\\&\quad+\frac{Kd}{K}+\frac{Td}{K}+\frac{T^2d}{K}+\frac{T^2d}{K(N-T)}+\frac{Td}{N-T}+\frac{T^2d}{N-T}\Big)\log q\end{aligned} \tag{110}$$

$$\begin{aligned}&=\Big(d\Big(\frac{2T}{K}+T+1\Big)\\&\quad+Td\Big(\frac{T}{K}+\frac{T}{K(N-T)}+\frac{1}{N-T}+\frac{T}{N-T}\Big)\Big)\log q\end{aligned} \tag{111}$$

Finally, by combining (111) and (109) with (80), we find that,

$$0\le I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}^2_{\mathcal{T}}|\mathcal{M}^1_{\mathcal{T}},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}) \tag{112}$$

$$\begin{aligned}&\le\Big(d\Big(\frac{2T}{K}+T+1\Big)\\&\quad+Td\Big(\frac{T}{K}+\frac{T}{K(N-T)}+\frac{1}{N-T}+\frac{T}{N-T}\Big)\Big)\log q\\&\quad-\Big(d\Big(\frac{2T}{K}+T+1\Big)\\&\quad+Td\Big(\frac{T}{K}+\frac{T}{K(N-T)}+\frac{1}{N-T}+\frac{T}{N-T}\Big)\Big)\log q\end{aligned} \tag{113}$$

$$=0 \tag{114}$$

where the inequality in (112) follows from the non-negativity of mutual information. Hence,

$$I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}^2_{\mathcal{T}}|\mathcal{M}^1_{\mathcal{T}},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})=0 \tag{115}$$

for the second term in (49).

Stage 3: Model Initialization: We now consider the third term in (49), which corresponds to Stage 3 of PICO, i.e., model initialization. Without loss of generality, we represent the secret share $[\overline{\mathbf{w}}_i^{(0)}]_j$ sent from client $i\in[N]$ to client $j\in[N]$ as follows:

$$[\overline{\mathbf{w}}_i^{(0)}]_j\triangleq\overline{\mathbf{w}}_i^{(0)}+\sum_{k\in[T]}\gamma_j^k\mathbf{s}_{ik}^{(0)} \tag{116}$$

where $\{\mathbf{s}_{ik}^{(0)}\}_{k\in[T]}$ are T random vectors of size $\frac{d}{N-T}$, where each element is generated independently and uniformly at random from $\mathbb{F}_q$, and coefficients $\{\gamma_j\}_{j\in[N]}$ are as defined in (77). We can then rewrite the mutual information condition for the third term in (49) as follows:

$$\begin{aligned}&I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}^3_{\mathcal{T}}|\mathcal{M}^1_{\mathcal{T}},\mathcal{M}^2_{\mathcal{T}},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\\&=I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}},\end{aligned}$$

$$\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \tag{117}$$

$$\begin{aligned}&= I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\quad \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\ &\quad + I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}},\\ &\quad \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\end{aligned} \tag{118}$$

$$\begin{aligned}&= H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\ &\quad - H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})\\ &\quad + H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}, \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\quad \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\ &\quad - H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}, \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\quad \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})\end{aligned} \tag{119}$$

We next consider each term in (119) separately. For the first term in (119), we have that,

$$\begin{aligned}&H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\ &\le (N-T)T\frac{d}{N-T}\log q = Td\log q\end{aligned} \tag{120}$$

which holds since uniform distribution maximizes entropy. For the second term in (119), we let,

$$\begin{aligned}&\left[[\overline{\mathbf{w}}_i^{(0)}]_{N-T+1} \ \cdots \ [\overline{\mathbf{w}}_i^{(0)}]_N\right]\\ &= \overline{\mathbf{w}}_i^{(0)} \underbrace{\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}}_{\mathbf{1}} + \underbrace{\begin{bmatrix}\mathbf{s}_{i1}^{(0)} & \cdots & \mathbf{s}_{iT}^{(0)}\end{bmatrix}}_{\mathbf{s}_i^{(0)}} \mathbf{A}\end{aligned} \tag{121}$$

$$= \overline{\mathbf{w}}_i^{(0)}\mathbf{1} + \mathbf{s}_i^{(0)}\mathbf{A} \tag{122}$$

where $\mathbf{A}$ is a $T\times T$ MDS matrix as defined in (91), and $\mathbf{1}$ is a $1\times T$ vector, where each element is equal to 1. Using (122), the second term in (119) can be written as,

$$\begin{aligned}&H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})\\ &\ge H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)},\\ &\quad \{\overline{\mathbf{w}}_i^{(0)}\}_{i\in\mathcal{H}})\end{aligned} \tag{123}$$

$$\begin{aligned}&= H(\{\overline{\mathbf{w}}_i^{(0)}\mathbf{1} + \mathbf{s}_i^{(0)}\mathbf{A}\}_{i\in\mathcal{H}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)},\\ &\quad \{\overline{\mathbf{w}}_i^{(0)}\}_{i\in\mathcal{H}})\end{aligned} \tag{124}$$

$$= H(\{\mathbf{s}_i^{(0)}\mathbf{A}\}_{i\in\mathcal{H}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}, \{\overline{\mathbf{w}}_i^{(0)}\}_{i\in\mathcal{H}}) \tag{125}$$

$$= H(\{\mathbf{s}_i^{(0)}\}_{i\in\mathcal{H}} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}, \{\overline{\mathbf{w}}_i^{(0)}\}_{i\in\mathcal{H}}) \tag{126}$$

$$= H\Big(\{\mathbf{s}_i^{(0)}\}_{i\in\mathcal{H}}\Big) \tag{127}$$

$$= (N-T)T\frac{d}{N-T}\log q \tag{128}$$

$$= dT\log q \tag{129}$$

where (123) holds since conditioning cannot increase entropy, and matrix $\mathbf{A}$ in (124) is a $T\times T$ MDS matrix as defined in (91). Equation (126) holds since $\mathbf{A}$ is an MDS matrix, hence is invertible. Equation (127) follows from the independence of the generated random vectors, and (128) is from the entropy of uniform random variables. For the third term in (119), we have,

$$\begin{aligned}&H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}, \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\quad \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)})\\ &\quad \le H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]})\end{aligned} \tag{130}$$

$$\le \left(\frac{Td}{N-T} + \frac{T^2d}{N-T}\right)\log q \tag{131}$$

where (130) holds since conditioning cannot increase entropy, and (131) follows from the entropy of uniform random variables. For the last term in (119), we find that,

$$\begin{aligned}&H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}, \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\quad \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})\\ &\ge H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}, \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\quad \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}, \overline{\mathbf{w}}^{(0)})\end{aligned} \tag{132}$$

$$= H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} | \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}, \overline{\mathbf{w}}^{(0)}) \tag{133}$$

$$= H(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}) \tag{134}$$

$$= \left(\frac{Td}{N-T} + \frac{T^2d}{N-T}\right)\log q \tag{135}$$

where (132) is from the fact that conditioning cannot increase entropy, and (133) holds since:

$$\begin{aligned}&\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]} - \overline{\mathbf{w}}^{(0)}, \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} - \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2,\\ &\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\end{aligned} \tag{136}$$

forms a Markov chain, and (135) follows from the entropy of uniform random variables. For (134), we first observe,

$$\begin{aligned}&I(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}; \overline{\mathbf{w}}^{(0)}, \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}})\\ &= I(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}; \overline{\mathbf{w}}^{(0)})\\ &\quad + I(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}; \{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}} | \overline{\mathbf{w}}^{(0)}).\end{aligned} \tag{137}$$

For the first term in (137), we find that,

$$0 \le I(\{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}; \overline{\mathbf{w}}^{(0)}) \tag{138}$$

$$= H(\overline{\mathbf{w}}^{(0)}) - H(\overline{\mathbf{w}}^{(0)} | \{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}) \tag{139}$$

$$\begin{aligned}&\le d\log q - H\Big((\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_N^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \Big|\\ &\quad \{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}\Big)\end{aligned} \tag{140}$$

$$\begin{aligned}&= d\log q - H\Big((\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_{N-T}^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \Big|\\ &\quad \{\overline{\mathbf{w}}_i^{(0)}, \mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}\Big)\end{aligned} \tag{141}$$

$$= d\log q - H\Big((\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_{N-T}^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big) \tag{142}$$

$$= d\log q - H\Big(\overline{\mathbf{w}}_1^{(0)}, \ldots, \overline{\mathbf{w}}_{N-T}^{(0)}\Big) \tag{143}$$

$$= d\log q - (N-T)\frac{d}{N-T}\log q \tag{144}$$

$$= 0 \tag{145}$$

where $\mathbf{M}$ and $\overline{\mathbf{M}}$ are as defined in (16), and (86), respectively, (138) is due to the non-negativity of mutual information, (140) holds since entropy is maximized by uniform distribution,

(142) holds since the randomness generated by the honest clients is independent from the randomness generated by adversaries, (143) holds since $\overline{\mathbf{M}}$ is an $(N-T)\times(N-T)$ MDS matrix, hence is invertible, and (144) follows from the entropy of uniformly random variables. From (145),

$$I(\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]};\overline{\mathbf{w}}^{(0)})=0 \tag{146}$$

Next, for the second term in (137), we find that,

$$0\le I(\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]};\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\overline{\mathbf{w}}^{(0)}) \tag{147}$$

$$=H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\overline{\mathbf{w}}^{(0)})$$
$$-H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\overline{\mathbf{w}}^{(0)},\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}) \tag{148}$$

where

$$H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\overline{\mathbf{w}}^{(0)})\le\frac{d}{N-T}(N-T)T\log q \tag{149}$$

since uniform distribution maximizes entropy, and

$$H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\overline{\mathbf{w}}^{(0)},\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]})$$
$$=H\Big(\{\overline{\mathbf{w}}_i^{(0)}\mathbf{1}+\mathbf{s}_i^{(0)}\mathbf{A}\}_{i\in\mathcal{H}}|(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_N^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},$$
$$\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}\Big) \tag{150}$$
$$=H\Big(\{\overline{\mathbf{w}}_i^{(0)}\mathbf{1}+\mathbf{s}_i^{(0)}\mathbf{A}\}_{i\in\mathcal{H}}|$$
$$(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_{N-T}^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}\Big) \tag{151}$$
$$=H\Big(\{\overline{\mathbf{w}}_i^{(0)}\mathbf{1}+\mathbf{s}_i^{(0)}\mathbf{A}\}_{i\in\mathcal{H}}|\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_{N-T}^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big) \tag{152}$$
$$=H\Big(\{\mathbf{s}_i^{(0)}\mathbf{A}\}_{i\in\mathcal{H}}|\begin{bmatrix}(\overline{\mathbf{w}}_1^{(0)})^{\mathrm{T}} & \cdots & (\overline{\mathbf{w}}_{N-T}^{(0)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big) \tag{153}$$
$$=H(\{\mathbf{s}_i^{(0)}\}_{i\in\mathcal{H}}) \tag{154}$$
$$=\frac{d}{N-T}(N-T)T\log q \tag{155}$$

which holds since $\overline{\mathbf{M}}$ and $\mathbf{A}$ are MDS matrices (invertible) and that the random vectors are generated independently. By combining (148) with (149) and (155), we find that,

$$I(\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]};\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\overline{\mathbf{w}}^{(0)})=0. \tag{156}$$

Then, by combining (146) and (156) with (137), we have that,

$$I(\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]};\overline{\mathbf{w}}^{(0)},\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}})=0 \tag{157}$$

from which (134) follows. Finally, by combining (120), (129), (131), and (135) with (119), we find that,

$$0\le I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}^3|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}) \tag{158}$$

$$=H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})$$
$$-H(\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)})$$
$$+H(\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}|\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}},\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,$$
$$\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})-H(\{\overline{\mathbf{w}}_i^{(0)},\mathbf{s}_{ik}^{(0)}\}_{i\in\mathcal{T},k\in[T]}|$$
$$\{[\overline{\mathbf{w}}_i^{(0)}]_j\}_{i\in\mathcal{H},j\in\mathcal{T}},\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)}) \tag{159}$$

$$\le Td\log q-Td\log q+\Big(\frac{Td}{N-T}+\frac{T^2d}{N-T}\Big)\log q$$
$$-\Big(\frac{Td}{N-T}+\frac{T^2d}{N-T}\Big)\log q \tag{160}$$
$$=0 \tag{161}$$

Hence, the third term in (49) satisfies:

$$I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}^3|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})=0. \tag{162}$$

Stage 4: Model Encoding. We next consider the fourth term in (49), which corresponds to model encoding. We represent the secret share of $\mathbf{r}_i^{(t)}$ at client $j\in[N]$ as,

$$[\mathbf{r}_i^{(t)}]_j=\mathbf{r}_i^{(t)}+\sum_{k\in[T]}\gamma_j^k\mathbf{g}_{ik}^{(t)} \tag{163}$$

for $i\in[N]$, where $\mathbf{g}_{ik}^{(t)}$ is a random vector of size $\frac{d}{N-T}$ where each element is generated independently and uniformly at random from $\mathbb{F}_q$, and the coefficients γ_i for $i\in[N]$ are as defined in (77). Then, for the third term in (49), we observe that:

$$I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}^{4,t}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},$$
$$\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})$$
$$=I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\{[\mathbf{r}_i^{(t)}]_j,\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},$$
$$\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,$$
$$\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}) \tag{164}$$
$$=H(\{[\mathbf{r}_i^{(t)}]_j,\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},$$
$$\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,$$
$$\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})$$
$$-H(\{[\mathbf{r}_i^{(t)}]_j,\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},$$
$$\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,$$
$$\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)}) \tag{165}$$

Without loss of generality, we denote the secret share of the model $\overline{\mathbf{w}}^{(t)}$ held at client $i\in[N]$ at time t as follows:

$$[\overline{\mathbf{w}}^{(t)}]_i=\overline{\mathbf{w}}^{(t)}+\sum_{k\in[T]}\gamma_i^k\mathbf{s}_k^{(t)}\quad\text{for all } i\in[N], \tag{166}$$

where $\mathbf{s}_k^{(t)}\in\mathbb{F}_q^d$, and coefficients γ_i for $i\in[N]$ are as defined in (77). From (25), we find that:

$$[\widehat{\mathbf{w}}^{(t)}]_i=[\overline{\mathbf{w}}^{(t)}]_i-[\mathbf{r}^{(t)}]_i \tag{167}$$
$$=[\overline{\mathbf{w}}^{(t)}]_i-(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}([\mathbf{r}_1^{(t)}]_i)^{\mathrm{T}} & \cdots & ([\mathbf{r}_N^{(t)}]_i)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \tag{168}$$
$$=\Big(\overline{\mathbf{w}}^{(t)}-(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_N^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big)$$
$$+\sum_{k\in[T]}\gamma_i^k\Big(\mathbf{s}_k^{(t)}-(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{g}_{Nk}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big) \tag{169}$$

is an evaluation point of a polynomial of degree T. Since any polynomial of degree T can be uniquely determined from at least $T+1$ evaluation points, there is a bijective mapping between the set of $T+1$ coefficients,

$$\begin{aligned}&\Big\{\overline{\mathbf{w}}^{(t)} - (\mathbf{M}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_N^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}, \mathbf{s}_1^{(t)}\\ &\quad - (\mathbf{M}\otimes\mathbf{I})\left[(\mathbf{g}_{11}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{N1}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\qquad\qquad \ldots, \mathbf{s}_T^{(t)} - (\mathbf{M}\otimes\mathbf{I})\left[(\mathbf{g}_{1T}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{NT}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\},\end{aligned}$$

and the feasible set of evaluation points $\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}$. Then, the second term in (165) can be rewritten as follows:

$$\begin{aligned}&H(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, \{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\\ &\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]} | \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3,\\ &\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)})\\ &= H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, \{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\\ &\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \overline{\mathbf{w}}^{(t)} - (\mathbf{M}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_N^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{\mathbf{s}_k^{(t)} - (\mathbf{M}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{Nk}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\\ &\Big|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\Bigg)\end{aligned}\tag{170}$$

$$\begin{aligned}&= H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, \{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\\ &\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, (\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{\mathbf{s}_k^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\\ &\Big|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\Bigg)\end{aligned}\tag{171}$$

$$\begin{aligned}&= H\Big(\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}}, \{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}\Big)\\ &+ H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, (\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{\mathbf{s}_k^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\\ &\Big|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\Bigg)\end{aligned}\tag{172}$$

$$\begin{aligned}&= \left(\frac{Td}{N-T} + \frac{T^2d}{N-T} + \frac{T^2d}{N-T}\right)\log q + H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\\ &(\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{\mathbf{s}_k^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\\ &\Big|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}\Bigg)\end{aligned}\tag{173}$$

$$\begin{aligned}&\geq \frac{Td}{N-T}(1+2T)\log q + H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\\ &(\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{\mathbf{s}_k^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\Big|\\ &\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\ &\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}, \{\mathbf{s}_k^{(t)}\}_{k\in[T]}\Bigg)\end{aligned}\tag{174}$$

$$\begin{aligned}&= \frac{Td}{N-T}(1+2T)\log q + H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\\ &(\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{(\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\Big|\\ &\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\ &\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}, \overline{\mathbf{w}}^{(J)}, \{\mathbf{s}_k^{(t)}\}_{k\in[T]}\Bigg)\end{aligned}\tag{175}$$

$$\begin{aligned}&= \frac{Td}{N-T}(1+2T)\log q + H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\ &(\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{(\overline{\mathbf{M}}\otimes\mathbf{I})\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\Bigg)\end{aligned}\tag{176}$$

$$\begin{aligned}&= \frac{Td}{N-T}(1+2T)\log q + H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\\ &\left[(\mathbf{r}_1^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{r}_{N-T}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}},\\ &\Big\{\left[(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} \quad \cdots \quad (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\right]^{\mathrm{T}}\Big\}_{k\in[T]}\Bigg)\end{aligned}\tag{177}$$

$$\begin{aligned}&= \frac{Td}{N-T}(1+2T)\log q\\ &+ H\Bigg(\{[\mathbf{r}_i^{(t)}]_j, \widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, \{\mathbf{r}_i^{(t)}\}_{i\in[N-T]}, \{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]}\Bigg)\end{aligned}\tag{178}$$

where (171) follows from $\mathcal{H} = [N-T]$, and that $\overline{\mathbf{w}}^{(t)}$ can be determined from $\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in[N]}$ and $\overline{\mathbf{w}}^{(J)}$; (172) follows from the independence of random vectors generated by honest clients; (173) follows from the entropy of uniform random variables; (174) holds since conditioning cannot increase entropy; (176) holds from the independence of generated random vectors, and (177) follows from the fact that $\overline{\mathbf{M}}$ is an $(N-T)\times(N-T)$ MDS matrix (hence is invertible) as defined in (86). Using (121), we next rewrite $\{[\mathbf{r}_i^{(t)}]_j\}_{j\in[T]}$ as follows,

$$\begin{aligned}&\begin{bmatrix}[\mathbf{r}_i^{(t)}]_{N-T+1} & \cdots & [\mathbf{r}_i^{(t)}]_N\end{bmatrix}\\&=\mathbf{r}_i^{(t)}\underbrace{\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}}_{\mathbf{1}}+\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A} \qquad (179)\\&=\mathbf{r}_i^{(t)}\mathbf{1}+\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A} \qquad (180)\end{aligned}$$

where $\mathbf{A}$ is the $T\times T$ MDS matrix defined in (91). Similarly, using (99), we can rewrite $\{\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{j\in\mathcal{T}}$ as follows,

$$\begin{aligned}&\begin{bmatrix}\widetilde{\mathbf{r}}_{i,N-T+1}^{(t)} & \cdots & \widetilde{\mathbf{r}}_{i,N}^{(t)}\end{bmatrix}\\&=\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix}\\&\quad+\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma} \qquad (181)\end{aligned}$$

where $\mathbf{\Gamma}$ is a $T\times T$ MDS matrix (hence invertible) as defined in (66). By using (180) and (181), we rewrite the second term in (178) as follows,

$$\begin{aligned}&H\Bigg(\{[\mathbf{r}_i^{(t)}]_j,\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]},\{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]}\Bigg)\\&=H\Bigg(\{\mathbf{r}_i^{(t)}\mathbf{1}+\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\}_{i\in[N-T]},\\&\Bigg\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix}\\&+\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Bigg\}_{i\in[N-T]},\\&\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]},\{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]}\Bigg) \qquad (182)\\&=H\Bigg(\{\mathbf{r}_i^{(t)}\mathbf{1}+\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\}_{i\in[N-T]},\\&\Bigg\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix}\\&+\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Bigg\}_{i\in[N-T]},\\&\{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]}\Bigg|\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]}\Bigg)+H(\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]})\\&\qquad (183)\\&=H\Big(\{\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\}_{i\in[N-T]},\\&\{\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\}_{i\in[N-T]},\\&\{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]}\Big|\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]}\Big)+H(\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]})\\&\qquad (184)\\&=H(\{\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\}_{i\in[N-T]})\\&+H(\{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]})+H(\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]}) \qquad (185)\\&=H(\{\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\}_{i\in[N-T]})\\&+H(\{\mathbf{g}_{ik}^{(t)}\}_{i\in[N-T],k\in[T]})+H(\{\mathbf{r}_i^{(t)}\}_{i\in[N-T]}) \qquad (186)\\&=(N-T)T\frac{d}{N-T}\log q+(N-T)T\frac{d}{N-T}\log q\\&+(N-T)\frac{d}{N-T}\log q \qquad (187)\\&=d(1+2T)\log q \qquad (188)\end{aligned}$$

where (183) follows from the chain rule of entropy; (185) follows from the independence of generated random vectors; (186) holds since $\mathbf{\Gamma}$ is an MDS matrix (hence invertible); (187) follows from the entropy of uniform random variables.

By combining (178) with (188), the following holds for the second term in (165),

$$\begin{aligned}&H(\{[\mathbf{r}_i^{(t)}]_j,\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\\&\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\\&\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)})\\&\geq d(2T+1)\left(\frac{T}{N-T}+1\right)\log q \qquad (189)\end{aligned}$$

We next analyze the first term in (165). By utilizing (169), (180), and (181), we find that:

$$\begin{aligned}&H(\{[\mathbf{r}_i^{(t)}]_j,\widetilde{\mathbf{r}}_{ij}^{(t)}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\\&\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\{[\widehat{\mathbf{w}}^{(t)}]_i\}_{i\in[N]}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\\&\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\\&=H\Bigg(\Big\{\mathbf{r}_i^{(t)}\mathbf{1}+\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\Big\}_{i\in\mathcal{H}},\\&\Big\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix}\\&+\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Big\}_{i\in\mathcal{H}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\\&\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\overline{\mathbf{w}}^{(t)}-(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_N^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\&\Big\{\mathbf{s}_k^{(t)}-(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{g}_{Nk}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[T]}\\&\Big|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\Bigg)\\&\qquad (190)\\&=H\Bigg(\Big\{\mathbf{r}_i^{(t)}\mathbf{1}+\begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\Big\}_{i\in\mathcal{H}},\\&\Big\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix}\\&+\begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Big\}_{i\in\mathcal{H}},\{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}},\\&\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}},\{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}},\\&\overline{\mathbf{w}}^{(t)}-(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_{(N-T)}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}},\\&\Big\{\mathbf{s}_k^{(t)}-(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[T]}\\&\Big|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\Bigg)\\&\qquad (191)\end{aligned}$$

Note that at the beginning of this stage, adversaries hold secret shares $\{[\overline{\mathbf{w}}^{(t)}]_j\}_{j\in\mathcal{T}}$ of the model $\overline{\mathbf{w}}^{(t)}$.

Accordingly, $\{[\overline{\mathbf{w}}^{(t)}]_j\}_{j\in\mathcal{T}} \in \mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}$. Then, by letting,

$$\begin{bmatrix}[\overline{\mathbf{w}}^{(t)}]_{N-T+1} & \cdots & [\overline{\mathbf{w}}^{(t)}]_N\end{bmatrix} = \overline{\mathbf{w}}^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{s}_1^{(t)} & \cdots & \mathbf{s}_T^{(t)}\end{bmatrix}\mathbf{A} \tag{192}$$

denote the secret shares of the model $\overline{\mathbf{w}}^{(t)}$ held by the adversaries $\mathcal{T} = \{N-T+1.\ldots,N\}$, one can observe that,

$$\begin{aligned}
&\Bigg(\mathbf{s}_1^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{g}_{11}^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{g}_{(N-T)1}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}, \ldots, \mathbf{s}_T^{(t)} \\
&- (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{g}_{1T}^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{g}_{(N-T)T}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Bigg) \\
&= \begin{bmatrix}\mathbf{s}_1^{(t)} & \cdots & \mathbf{s}_T^{(t)}\end{bmatrix} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{g}_{11}^{(t)} & \cdots & \mathbf{g}_{1T}^{(t)} \\ \vdots & \ddots & \vdots \\ \mathbf{g}_{(N-T)1}^{(t)} & \cdots & \mathbf{g}_{(N-T)T}^{(t)}\end{bmatrix}
\end{aligned} \tag{193}$$

$$\begin{aligned}
&= \Bigg(\overline{\mathbf{w}}^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{s}_1^{(t)} & \cdots & \mathbf{s}_T^{(t)}\end{bmatrix}\mathbf{A} \\
&- \left(\overline{\mathbf{w}}^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_{(N-T)}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\right)\mathbf{1} \\
&- (\overline{\mathbf{M}}\otimes\mathbf{I})\Bigg[\left(\mathbf{r}_1^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{g}_{11}^{(t)} & \cdots & \mathbf{g}_{1T}^{(t)}\end{bmatrix}\mathbf{A}\right)^{\mathrm{T}} \cdots \\
&\quad \left(\mathbf{r}_{(N-T)}^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{g}_{(N-T)1}^{(t)} & \cdots & \mathbf{g}_{(N-T)T}^{(t)}\end{bmatrix}\mathbf{A}\right)^{\mathrm{T}}\Bigg]^{\mathrm{T}}\Bigg)\mathbf{A}^{-1}
\end{aligned} \tag{194}$$

From (194), we then observe the following for (191):

$$\begin{aligned}
&H\Bigg(\left\{\mathbf{r}_i^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\right\}_{i\in\mathcal{H}}, \\
&\Big\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix} \\
&+ \begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Big\}_{i\in\mathcal{H}}, \{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \\
&\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}}, \{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \\
&\overline{\mathbf{w}}^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_{(N-T)}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}, \\
&\left\{\mathbf{s}_k^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{g}_{1k}^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{g}_{(N-T)k}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\right\}_{k\in[T]} \\
&\Big|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}\Bigg) \\
&= H\Bigg(\left\{\mathbf{r}_i^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}\mathbf{A}\right\}_{i\in\mathcal{H}}, \\
&\Big\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix} \\
&+ \begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Big\}_{i\in\mathcal{H}}, \{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \\
&\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}}, \{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \\
&\overline{\mathbf{w}}^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_{(N-T)}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \\
&|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}\Bigg)
\end{aligned} \tag{195}$$

$$\begin{aligned}
&\le H\Bigg(\left\{\mathbf{r}_i^{(t)}\mathbf{1} + \begin{bmatrix}\mathbf{g}_{i1}^{(t)} & \cdots & \mathbf{g}_{iT}^{(t)}\end{bmatrix}^{\mathrm{T}}\mathbf{A}\right\}_{i\in\mathcal{H}}, \\
&\Big\{\mathbf{r}_i^{(t)}\begin{bmatrix}\sum_{k\in[K]}\rho_{N-T+1,k} & \cdots & \sum_{k\in[K]}\rho_{N,k}\end{bmatrix} \\
&+ \begin{bmatrix}\mathbf{v}_{i,K+1}^{(t)} & \cdots & \mathbf{v}_{i,K+T}^{(t)}\end{bmatrix}\mathbf{\Gamma}\Big\}_{i\in\mathcal{H}}, \{\mathbf{r}_i^{(t)}\}_{i\in\mathcal{T}}, \\
&\{\mathbf{g}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[T]}}, \{\mathbf{v}_{ik}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in\{K+1,\ldots,K+T\}}}, \\
&\overline{\mathbf{w}}^{(t)} - (\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}(\mathbf{r}_1^{(t)})^{\mathrm{T}} & \cdots & (\mathbf{r}_{(N-T)}^{(t)})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Bigg)
\end{aligned} \tag{196}$$

$$\begin{aligned}
&\le \bigg((N-T)T\frac{d}{N-T} + (N-T)T\frac{d}{N-T} \\
&+ T\frac{d}{N-T} + T^2\frac{d}{N-T} + T^2\frac{d}{N-T} + d\bigg)\log q
\end{aligned} \tag{197}$$

$$= d(2T+1)\left(\frac{T}{N-T}+1\right)\log q \tag{198}$$

where (195) follows from (194), (196) holds since conditioning cannot increase entropy, and (197) holds since entropy is maximized by uniform distribution. Finally, by combining (189) and (198) with (165), we find that:

$$\begin{aligned}
0 &\le I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^{4,t}|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \\
&\quad \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \\
&\le d(2T+1)\left(\frac{T}{N-T}+1\right)\log q \\
&\quad - d(2T+1)\left(\frac{T}{N-T}+1\right)\log q \qquad (199) \\
&= 0 \qquad (200)
\end{aligned}$$

Therefore, the fourth term in (49) satisfies the following:

$$\begin{aligned}
&I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^{4,t}|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \\
&\quad \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) = 0
\end{aligned} \tag{201}$$

for all $t \in \{0,\ldots,J-1\}$.

Stage 5: Gradient Computing and Model Update: We next consider the fifth term in (49), which corresponds to Stage 5 of the proposed framework, i.e., local gradient computation and model updates. In the following, we define $C \triangleq (2r+1)(K+T-1)+1$. Then, the last term in (49) can be written as:

$$\begin{aligned}
&I(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \mathcal{M}_{\mathcal{T}}^{5,t}|\mathcal{M}_{\mathcal{T}}^1, \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \\
&\quad \cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}) \\
&= I\bigg(\{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{H}}; \{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, \{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}, \\
&\quad \{\widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i \times \widetilde{\mathbf{w}}_i^{(t)}) - \widetilde{\mathbf{u}}_i\}_{i\in[N]}, \{\mathbf{u}_{ik}^{(t)}, \mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C], l\in[T]}}|\mathcal{M}_{\mathcal{T}}^1, \\
&\quad \mathcal{M}_{\mathcal{T}}^2, \mathcal{M}_{\mathcal{T}}^3, \cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l}, \cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}, \{\overline{\mathbf{X}}_i, \overline{\mathbf{y}}_i\}_{i\in\mathcal{T}}, \overline{\mathbf{w}}^{(J)}\bigg)
\end{aligned} \tag{202}$$

Recall that the local computations $\{\widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})-\widetilde{\mathbf{u}}_i\}_{i\in[N]}$ correspond to evaluations of the polynomial $\varphi(\alpha)-\phi(\alpha)$ at $\alpha\in\{\alpha_i\}_{i\in[N]}$. Next, consider a second set of coefficients β_k for $k\in[C]$, where β_k is as defined in (29). We know that polynomial $\varphi(\alpha)-\phi(\alpha)$ has degree $(2r+1)(K+T-1)=C-1$. Any polynomial of degree $C-1$ can be uniquely determined from any set of at least C evaluation points. As $N\geq(2r+1)(K+T-1)+1=C$, there is a bijective mapping from any C evaluation points $\{\varphi(\beta_k)-\phi(\beta_k)\}_{k\in[C]}$ to a valid set of local computations $\{\widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})-\widetilde{\mathbf{u}}_i\}_{i\in[N]}$. As a result, one can rewrite (202) as follows,

$$\begin{aligned}
&I\Big(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})-\widetilde{\mathbf{u}}_i\}_{i\in[N]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\\
&|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)\\
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})-\widetilde{\mathbf{u}}_i\}_{i\in[N]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\\
&|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)\\
&-H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\widetilde{\mathbf{X}}_i^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_i\times\widetilde{\mathbf{w}}_i^{(t)})-\widetilde{\mathbf{u}}_i\}_{i\in[N]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\\
&|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)}\Big)\\
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\varphi(\beta_k)-\phi(\beta_k)\}_{k\in[C]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\\
&|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)\\
&-H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\varphi(\beta_k)-\phi(\beta_k)\}_{k\in[C]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}|\mathcal{M}_{\mathcal{T}}^1,\\
&\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)}\Big)
\end{aligned}\tag{203}$$

$$\begin{aligned}
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\varphi(\beta_k)-\mathbf{u}_k\}_{k\in[C]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\\
&|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)\\
&-H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\varphi(\beta_k)-\mathbf{u}_k\}_{k\in[C]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}|\mathcal{M}_{\mathcal{T}}^1,\\
&\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)}\Big)
\end{aligned}\tag{204}$$

where (204) holds since $\phi(\beta_k)=\mathbf{u}_k$ by definition from (32). For the second term in (204), we find that,

$$\begin{aligned}
&H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\{\varphi(\beta_k)-\mathbf{u}_k\}_{k\in[C]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\\
&|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)}\Big)\\
&\geq H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\varphi(\beta_k)-\mathbf{u}_k\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}|\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\overline{\mathbf{w}}^{(J)},\{\varphi(\beta_k)\}_{k\in[C]}\Big)
\end{aligned}\tag{205}$$

$$\begin{aligned}
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\mathbf{u}_k\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\Big)
\end{aligned}\tag{206}$$

$$\begin{aligned}
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{Nk}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[C]},\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\Big)
\end{aligned}\tag{207}$$

$$\begin{aligned}
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\Big)
\end{aligned}\tag{208}$$

$$\begin{aligned}
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\mathbf{u}_{ik}\}_{\substack{i\in\mathcal{H},\\ k\in[C]}}\Big)\\
&+H\Big(\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\Big)
\end{aligned}\tag{209}$$

$$\begin{aligned}
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\mathbf{u}_{ik}\}_{\substack{i\in\mathcal{H},\\ k\in[C]}}\Big)\\
&+\Big(\frac{d}{N-T}TC+\frac{d}{N-T}T^2\Big)\log q
\end{aligned}\tag{210}$$

where (205) holds since conditioning cannot increase entropy; (209) holds since $\overline{\mathbf{M}}$ is a $(N-T)\times(N-T)$ MDS matrix (hence invertible), and that the randomness generated by the honest clients $\mathcal{H}=[N-T]$ is independent from the adversaries. Note that $\{\widetilde{\mathbf{u}}_{ij}\}_{j\in\mathcal{T}}$ can be perfectly reconstructed from $\{\mathbf{u}_{ik}\}_{k\in[C]}$ using (29). Then, the first term in (210) can be rewritten as:

$$
\begin{aligned}
&H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\mathbf{u}_{ik}\}_{\substack{i\in\mathcal{H},\\ k\in[C]}}\Big)\\
&= H\Big(\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\mathbf{u}_{ik}\}_{\substack{i\in\mathcal{H},\\ k\in[C]}}\Big) \qquad (211)\\
&= \sum_{i\in\mathcal{H}} H\Big(\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{j\in\mathcal{T}}, \{\mathbf{u}_{ik}\}_{k\in[C]}\Big) \qquad (212)\\
&= \sum_{i\in\mathcal{H}} H\Big(\Big\{\sum_{k\in[K]}\mathbf{u}_{ik} + \sum_{l\in[T]}\gamma_j^l\mathbf{z}_{il}\Big\}_{j\in\mathcal{T}}, \{\mathbf{u}_{ik}\}_{k\in[C]}\Big) \qquad (213)\\
&= \sum_{i\in\mathcal{H}} H\Big(\Big\{\sum_{k\in[K]}\mathbf{u}_{ik} + \sum_{l\in[T]}\gamma_j^l\mathbf{z}_{il}\Big\}_{j\in\mathcal{T}} \Big| \{\mathbf{u}_{ik}\}_{k\in[C]}\Big)\\
&\quad + \sum_{i\in\mathcal{H}} H(\{\mathbf{u}_{ik}\}_{k\in[C]}) \qquad (214)\\
&= \sum_{i\in\mathcal{H}} H\Big(\Big\{\sum_{l\in[T]}\gamma_j^l\mathbf{z}_{il}\Big\}_{j\in\mathcal{T}} \Big| \{\mathbf{u}_{ik}\}_{k\in[C]}\Big) + \sum_{i\in\mathcal{H}} H(\{\mathbf{u}_{ik}\}_{k\in[C]}) \qquad (215)\\
&= \sum_{i\in\mathcal{H}} H\Big(\Big\{\sum_{l\in[T]}\gamma_j^l\mathbf{z}_{il}\Big\}_{j\in\mathcal{T}}\Big) + \sum_{i\in\mathcal{H}} H(\{\mathbf{u}_{ik}\}_{k\in[C]}) \qquad (216)\\
&= \sum_{i\in\mathcal{H}} H((\mathbf{z}_{i1},\dots,\mathbf{z}_{iT})\mathbf{A}) + \sum_{i\in\mathcal{H}} H(\{\mathbf{u}_{ik}\}_{k\in[C]}) \qquad (217)\\
&= \sum_{i\in\mathcal{H}} H(\mathbf{z}_{i1},\dots,\mathbf{z}_{iT}) + \sum_{i\in\mathcal{H}} H(\{\mathbf{u}_{ik}\}_{k\in[C]}) \qquad (218)\\
&= (N-T)T\frac{d}{N-T}\log q + (N-T)C\frac{d}{N-T}\log q \qquad (219)\\
&= (T+C)d\log q \qquad (220)
\end{aligned}
$$

where (213) follows from (33); (214) follows from the chain rule of entropy; (216) follows from the independence of random vectors generated; (217) follows from the definition of matrix $\mathbf{A}$ from (91); (218) holds since $\mathbf{A}$ is a $T\times T$ MDS matrix (hence is invertible); (219) follows from the entropy of uniform random variables. By combining (220) with (210), we have the following for the second term in (204),

$$
\begin{aligned}
&H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\varphi(\beta_k)-\mathbf{u}_k\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}} | \mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in[N]},\mathbf{w}^{(J)}\Big)\\
&\geq \Big(\frac{d}{N-T}TC + \frac{d}{N-T}T^2\Big)\log q + (T+C)d\log q \qquad (221)\\
&= (T+C)d\Big(1+\frac{T}{N-T}\Big)\log q \qquad (222)
\end{aligned}
$$

For the first term in (204), we observe that,

$$
\begin{aligned}
&H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\varphi(\beta_k)-\mathbf{u}_k\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}} | \mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\mathbf{w}^{(J)}\Big)\\
&= H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\mathbf{M}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{Nk}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}} \Big| \mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\mathbf{w}^{(J)}\Big) \qquad (223)\\
&= H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}}, \{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}} \Big| \mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\mathbf{w}^{(J)}\Big) \qquad (224)
\end{aligned}
$$

Note that $\widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j\times\widetilde{\mathbf{w}}_j^{(t)})$ for any $j\in\mathcal{T}$ can be perfectly reconstructed by the adversaries, since the encoded dataset and model $\widetilde{\mathbf{X}}_j,\widetilde{\mathbf{w}}_j^{(t)}\in\mathcal{M}_{\mathcal{T}}^1,\mathcal{M}_{\mathcal{T}}^2,\mathcal{M}_{\mathcal{T}}^3,\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l}$ for $j\in\mathcal{T}$ is already known from previous stages. In addition, for any $j\in\mathcal{T}$, the following holds,

$$
\begin{aligned}
&(\overline{\mathbf{M}}\otimes\mathbf{I})^{-1}\Big(\widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j\times\widetilde{\mathbf{w}}_j^{(t)}) - \sum_{k\in[C]}\Big(\varphi(\beta_k)\\
&-(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\Big)\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\Big)\\
&= (\overline{\mathbf{M}}\otimes\mathbf{I})^{-1}\Big(\widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j\times\widetilde{\mathbf{w}}_j^{(t)})\\
&-\sum_{k\in[C]}\varphi(\beta_k)\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\\
&+\sum_{k\in[C]}(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\Big) \qquad (225)\\
&= (\overline{\mathbf{M}}\otimes\mathbf{I})^{-1}\Big(\widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j\times\widetilde{\mathbf{w}}_j^{(t)}) - \widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j\times\widetilde{\mathbf{w}}_j^{(t)})\\
&+\sum_{k\in[C]}(\overline{\mathbf{M}}\otimes\mathbf{I})\begin{bmatrix}\mathbf{u}_{1k}^{\mathrm{T}} & \cdots & \mathbf{u}_{(N-T)k}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\Big) \qquad (226)\\
&= \Big[\sum_{k\in[C]}\mathbf{u}_{1k}^{\mathrm{T}}\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l} \quad \cdots\\
&\qquad \sum_{k\in[C]}\mathbf{u}_{(N-T)k}^{\mathrm{T}}\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l}\Big]^{\mathrm{T}} \qquad (227)\\
&= \begin{bmatrix}\widetilde{\mathbf{u}}_{1j}^{\mathrm{T}} & \cdots & \widetilde{\mathbf{u}}_{(N-T)j}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \qquad (228)
\end{aligned}
$$

where (226) holds since $\widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j \times \widetilde{\mathbf{w}}_j^{(t)}) = \varphi(\alpha_j) = \sum_{k\in[C]}\varphi(\beta_k)\prod_{l\in[C]\setminus\{k\}}\frac{\alpha_j-\beta_l}{\beta_k-\beta_l}$, which can be observed from polynomial interpolation, hence $\{\widetilde{\mathbf{u}}_{ij}\}_{i\in\mathcal{H},j\in\mathcal{T}}$ can be reconstructed from $\{\widetilde{\mathbf{X}}_j^{\mathrm{T}}\hat{g}(\widetilde{\mathbf{X}}_j \times \widetilde{\mathbf{w}}_j^{(t)})\}_{j\in\mathcal{T}}$ and $\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\big[\mathbf{u}_{1k}^{\mathrm{T}}\ \cdots\ \mathbf{u}_{(N-T)k}^{\mathrm{T}}\big]^{\mathrm{T}}\Big\}_{k\in[C]}$. Then, from (228), the following holds for (224),

$$
\begin{aligned}
&H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\big[\mathbf{u}_{1k}^{\mathrm{T}}\ \cdots\ \mathbf{u}_{(N-T)k}^{\mathrm{T}}\big]^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)\\
&=H\Big(\{\widetilde{\mathbf{u}}_{ij}\}_{\substack{i\in\mathcal{H}\\ j\in\mathcal{T}}}\Big|\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\big[\mathbf{u}_{1k}^{\mathrm{T}}\ \cdots\ \mathbf{u}_{(N-T)k}^{\mathrm{T}}\big]^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}},\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)\\
&+H(\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\big[\mathbf{u}_{1k}^{\mathrm{T}}\ \cdots\ \mathbf{u}_{(N-T)k}^{\mathrm{T}}\big]^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)
\end{aligned}
\tag{229}
$$

$$
\begin{aligned}
&=H\Big(\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\big[\mathbf{u}_{1k}^{\mathrm{T}}\ \cdots\ \mathbf{u}_{(N-T)k}^{\mathrm{T}}\big]^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\\
&\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)}\Big)
\end{aligned}
\tag{230}
$$

$$
\begin{aligned}
&\leq H\Big(\{\big[\sum_{k\in[K]}\mathbf{u}_{ik}\big]_j\}_{\substack{i\in\mathcal{H},\\ j\in\mathcal{T}}},\\
&\Big\{\varphi(\beta_k)-(\overline{\mathbf{M}}\otimes\mathbf{I})\big[\mathbf{u}_{1k}^{\mathrm{T}}\ \cdots\ \mathbf{u}_{(N-T)k}^{\mathrm{T}}\big]^{\mathrm{T}}\Big\}_{k\in[C]},\\
&\{\mathbf{u}_{ik}^{(t)},\mathbf{z}_{il}^{(t)}\}_{\substack{i\in\mathcal{T}\\ k\in[C],l\in[T]}}\Big)
\end{aligned}
\tag{231}
$$

$$
\leq\Big((N-T)T\frac{d}{N-T}+Cd+TC\frac{d}{N-T}+T^2\frac{d}{N-T}\Big)\log q \tag{232}
$$

$$
=(T+C)d\Big(1+\frac{T}{N-T}\Big)\log q \tag{233}
$$

where (229) follows from the chain rule of entropy; (230) follows from (228); (231) holds since conditioning cannot increase entropy, and (232) holds since entropy is maximized by uniform distribution. By combining (222) and (233), we find for the last term in (49) that:

$$
\begin{aligned}
&0\leq I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}^{5,t}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\\
&\quad\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\\
&\ \leq(T+C)d\Big(1+\frac{T}{N-T}\Big)\log q-(T+C)d\Big(1+\frac{T}{N-T}\Big)\log q
\end{aligned}
\tag{234}
$$

$$
=0 \tag{235}
$$

hence, the fifth term in (49) satisfies:

$$
\begin{aligned}
&I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}^{5,t}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{t}\mathcal{M}_{\mathcal{T}}^{4,l},\\
&\qquad\cup_{l=0}^{t-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})=0
\end{aligned}
\tag{236}
$$

for all $t\in\{0,\dots,J-1\}$.

A. Final Model Recovery

Finally, we consider the last term in (49), which corresponds to the recovery of the final model $\overline{\mathbf{w}}^{(J)}$ by collecting the secret shares $\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{I}}$ from any set $\mathcal{I}$ of size $|\mathcal{I}|\geq T+1$. From (166), the secret share of $\overline{\mathbf{w}}^{(J)}$ at client $i\in[N]$ is given by,

$$
[\overline{\mathbf{w}}^{(J)}]_i=\overline{\mathbf{w}}^{(J)}+\sum_{k\in[T]}\gamma_i^k\mathbf{s}_k^{(J)}\qquad\text{for all } i\in[N], \tag{237}
$$

which can be viewed as an evaluation point of a degree T polynomial $\sigma(\cdot)$ where $\sigma(0)=\overline{\mathbf{w}}^{(J)}$ is the true model and $[\overline{\mathbf{w}}^{(J)}]_i$ is an interpolation point held by client $i\in[N]$. Then, one can rewrite the last term in the mutual information condition from (49) as,

$$
\begin{aligned}
&I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}^{6}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{J-1}\mathcal{M}_{\mathcal{T}}^{4,l},\\
&\cup_{l=0}^{J-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})\\
&=I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{I}}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\cup_{l=0}^{J-1}\mathcal{M}_{\mathcal{T}}^{4,l},\\
&\cup_{l=0}^{J-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})
\end{aligned}
\tag{238}
$$

$$
\begin{aligned}
&=I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\overline{\mathbf{w}}^{(J)},\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{T}}|\mathcal{M}_{\mathcal{T}}^{1},\mathcal{M}_{\mathcal{T}}^{2},\mathcal{M}_{\mathcal{T}}^{3},\\
&\cup_{l=0}^{J-1}\mathcal{M}_{\mathcal{T}}^{4,l},\cup_{l=0}^{J-1}\mathcal{M}_{\mathcal{T}}^{5,l},\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})
\end{aligned}
\tag{239}
$$

$$
=0 \tag{240}
$$

where (239) holds since any polynomial of degree T can be uniquely constructed from $T+1$ interpolation points. Hence, there is a bijective mapping between $\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{I}}$ and $\overline{\mathbf{w}}^{(J)},\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{T}}$. Finally, (240) holds since $\{[\overline{\mathbf{w}}^{(J)}]_i\}_{i\in\mathcal{T}}\in\mathcal{M}_{\mathcal{T}}^{5,J-1}$.

B. Combining Stages 1-6

By combining (49) with (76), (115), (162), (201), and (236), we have,

$$
I(\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{H}};\mathcal{M}_{\mathcal{T}}|\{\overline{\mathbf{X}}_i,\overline{\mathbf{y}}_i\}_{i\in\mathcal{T}},\overline{\mathbf{w}}^{(J)})=0 \tag{241}
$$

which completes the proof. □

Appendix E
Correctness

The correctness of the encoding and decoding process follows from the decodability of the Lagrange interpolation polynomial [6], in particular, any polynomial φ of degree $\deg(\varphi)$ can be uniquely reconstructed from any set of at least $\deg(\varphi)+1$ interpolation points. As such, as long as the total number of clients N satisfy the minimum number identified by the recovery threshold, i.e., $N-D \geq (2r+1)(K+T-1)+1$, then one can correctly recover the final model $\overline{\mathbf{w}}^{(J)}$ from the gradient computations performed on the encoded datasets and models. This completes the correctness for the model update rule from (40).

We next study the model update rule from (47), and show that it correctly recovers the target model from (40). For the theoretical analysis, it is assumed that the finite field size is sufficiently large to avoid overlap errors. From (47), at the end of round t, each client holds a secret share $[\overline{\mathbf{w}}^{(t+1)}]_i$ of the updated model,

$$\begin{aligned}&\overline{\mathbf{w}}^{(t+1)}\\&= M^{(r-1)a_t+1}\overline{\mathbf{w}}^{(t)} - \left(\sum_{k\in[K]} \varphi(\beta_k) - M^{ra_t}\overline{\mathbf{X}}^T\overline{\mathbf{y}}\right) \quad (242)\\&= M^{(r-1)a_t+1}\overline{\mathbf{w}}^{(t)} - \left(\sum_{j=0}^{r} \theta_j M^{(r-j)a_t}\overline{\mathbf{X}}^{\mathrm{T}}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)})^j \right.\\&\qquad\qquad \left. - M^{ra_t}\overline{\mathbf{X}}^T\overline{\mathbf{y}}\right) \quad (243)\end{aligned}$$

We next describe a virtual variable $\overline{\mathbf{w}}_v^{(t)}$, where $\overline{\mathbf{w}}_v^{(0)} \triangleq \overline{\mathbf{w}}^{(0)}$, and

$$\overline{\mathbf{w}}_v^{(t+1)} \triangleq \overline{\mathbf{w}}_v^{(t)} - \frac{1}{M}\overline{\mathbf{X}}^T(\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}_v^{(t)}) - \overline{\mathbf{y}}) \quad \text{for } t\in\{0,\dots,J-1\}, \quad (244)$$

which denotes the target model from (40), by letting $M = \overline{m}/\eta$. Then, one can show that,

$$\frac{\overline{\mathbf{w}}^{(t)}}{M^{a_t}} = \overline{\mathbf{w}}_v^{(t)} \quad \text{for all } t \geq 1. \quad (245)$$

Then, the proof follows by induction, by considering the following steps.

1) *(Base Case):* For the base case ($t = 0$), it follows from (243) that,

$$\begin{aligned}\overline{\mathbf{w}}^{(1)} &= M\overline{\mathbf{w}}^{(0)} - (\overline{\mathbf{X}}^T\sum_{j=0}^{r}\theta_j(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(0)})^j - \overline{\mathbf{X}}^T\overline{\mathbf{y}}) \quad (246)\\&= M\overline{\mathbf{w}}^{(0)} - (\overline{\mathbf{X}}^T\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(0)}) - \overline{\mathbf{X}}^T\overline{\mathbf{y}}) \quad (247)\end{aligned}$$

hence $\frac{\overline{\mathbf{w}}^{(1)}}{M} = \overline{\mathbf{w}}_v^{(1)}$, which validates (245) for the base case.

2) *(Induction step):* Next, we assume that (245) holds for an arbitrary t, and show that it also holds for $t+1$. From (243), we have that,

$$\begin{aligned}&\overline{\mathbf{w}}^{(t+1)}\\&= M^{(r-1)a_t+1}\overline{\mathbf{w}}^{(t)} - (\sum_{j=0}^{r}\theta_j M^{(r-j)a_t}\overline{\mathbf{X}}^{\mathrm{T}}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}^{(t)})^j\\&\quad - M^{ra_t}\overline{\mathbf{X}}^T\overline{\mathbf{y}}) \quad (248)\\&= M^{(r-1)a_t+1}M^{a_t}\overline{\mathbf{w}}_v^{(t)} - (\sum_{j=0}^{r}\theta_j M^{(r-j)a_t}\overline{\mathbf{X}}^{\mathrm{T}}(\overline{\mathbf{X}}\times M^{a_t}\overline{\mathbf{w}}_v^{(t)})^j\\&\quad - M^{ra_t}\overline{\mathbf{X}}^T\overline{\mathbf{y}}) \quad (249)\\&= M^{ra_t+1}\overline{\mathbf{w}}_v^{(t)} - (\overline{\mathbf{X}}^T\sum_{j=0}^{r}M^{(r-j)a_t}\theta_j(\overline{\mathbf{X}}\times M^{a_t}\overline{\mathbf{w}}_v^{(t)})^j\\&\quad - M^{ra_t}\overline{\mathbf{X}}^T\overline{\mathbf{y}}) \quad (250)\\&= M^{ra_t+1}\overline{\mathbf{w}}_v^{(t)} - M^{ra_t}(\overline{\mathbf{X}}^T\sum_{j=0}^{r}\theta_j(\overline{\mathbf{X}}\times\overline{\mathbf{w}}_v^{(t)})^j - \overline{\mathbf{X}}^T\overline{\mathbf{y}})\\&\quad (251)\\&= M^{ra_t+1}\overline{\mathbf{w}}_v^{(t)} - M^{ra_t}(\overline{\mathbf{X}}^T\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}_v^{(t)}) - \overline{\mathbf{X}}^T\overline{\mathbf{y}}) \quad (252)\\&= M^{ra_t+1}\left(\overline{\mathbf{w}}_v^{(t)} - \frac{1}{M}(\overline{\mathbf{X}}^T\hat{g}(\overline{\mathbf{X}}\times\overline{\mathbf{w}}_v^{(t)}) - \overline{\mathbf{X}}^T\overline{\mathbf{y}})\right) \quad (253)\\&= M^{a_{t+1}}\overline{\mathbf{w}}_v^{(t+1)} \quad (254)\end{aligned}$$

where (249) follows from the fact that $\overline{\mathbf{w}}^{(t)} = M^{a_t}\overline{\mathbf{w}}_v^{(t)}$ since (245) for round t holds by assumption, (254) follows from $a_{t+1} = ra_t + 1$ by definition, along with (244). Equation (254) demonstrates that (245) also holds for $t+1$, which completes the proof.

References

[1] X. Lu, H. U. Sami, and B. Güler, "Dropout-resilient secure multi-party collaborative learning with linear communication complexity," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2023, pp. 10566–10593.

[2] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 19–38.

[3] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Secur. Privacy*, vol. 17, no. 2, pp. 49–58, Mar. 2019.

[4] R. Nosowsky and T. J. Giordano, "The health insurance portability and accountability act of 1996 (HIPAA) privacy rule: Implications for clinical research," *Annu. Rev. Med.*, vol. 57, no. 1, pp. 575–590, Feb. 2006.

[5] A. Telenti and X. Jiang, "Treating medical data as a durable asset," *Nature Genet.*, vol. 52, no. 10, pp. 1005–1010, Oct. 2020.

[6] Q. Yu et al., "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2019, pp. 1215–1225.

[7] J. So, B. Güler, and S. Avestimehr, "A scalable approach for privacy-preserving collaborative machine learning," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020, pp. 8054–8066.

[8] J. So, B. Güler, and A. S. Avestimehr, "CodedPrivateML: A fast and privacy-preserving framework for distributed machine learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 441–451, Mar. 2021.

[9] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Berlin, Germany: Springer, 2009.

[10] A. B. Slavkovic, Y. Nardi, and M. M. Tibbits, "'Secure' logistic regression of horizontally and vertically partitioned distributed databases," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Oct. 2007, pp. 723–728.

[11] Y. Aono, T. Hayashi, L. Trieu Phong, and L. Wang, "Scalable and secure logistic regression via homomorphic encryption," in *Proc. 6th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2016, pp. 142–144.

[12] S. Wu, T. Teruya, J. Kawamoto, J. Sakuma, and H. Kikuchi, "Privacy-preservation for stochastic gradient descent application to secure logistic regression," in *Proc. 27th Annu. Conf. Jpn. Soc. Artif. Intell.*, vol. 27, 2013, pp. 1–4.

[13] Z. Beerliova-TrubiniovA and M. Hirt, "Perfectly-secure MPC with linear communication complexity," in *Proc. Theory Cryptography Conf.* Cham, Switzerland: Springer, 2008, pp. 213–230.

[14] A. C. Yao, "Protocols for secure computations," in *Proc. IEEE Symp. Found. Comput. Sci.*, Mar. 1982, pp. 160–164.

[15] M. Ben-Or and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th Annu. ACM Symp. Theory Comput.*, 1988, pp. 1–10.

[16] I. Damgård and J. B. Nielsen, "Scalable and unconditionally secure multiparty computation," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 2007, pp. 572–590.

[17] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 334–348.

[18] A. Gascón et al., "Privacy-preserving distributed linear regression on high-dimensional data," *Proc. Privacy Enhancing Technol.*, vol. 2017, no. 4, pp. 345–364, Oct. 2017.

[19] P. Mohassel and P. Rindal, "ABY 3: A mixed protocol framework for machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 35–52.

[20] S. Wagh, D. Gupta, and N. Chandran, "SecureNN: Efficient and private neural network training," Int. Assoc. Cryptol. Res. (IACR), Cryptol. ePrint Arch., San Diego, CA, USA, Tech. Rep. 442, 2018.

[21] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[22] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. 27th Annu. Symp. Found. Comput. Sci. (SFCS)*, Oct. 1986, pp. 162–167.

[23] P. Mohassel and M. Franklin, "Efficiency tradeoffs for malicious two-party computation," in *Proc. 9th Int. Conf. Theory Pract. Public-Key Cryptography*, 2006, pp. 458–473.

[24] Y. Lindell and B. Pinkas, "An efficient protocol for secure two-party computation in the presence of malicious adversaries," in *Advances in Cryptology—EUROCRYPT*. Barcelona, Spain: Springer, 2007, pp. 52–78.

[25] Y. Ishai, E. Kushilevitz, R. Ostrovsky, M. Prabhakaran, and A. Sahai, "Efficient non-interactive secure computation," in *Proc. 30th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2011, pp. 406–425.

[26] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game, or a completeness theorem for protocols with honest majority," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 307–328.

[27] B. Kreuter, A. Shelat, and C.-H. Shen, "Billion-gate secure computation with malicious adversaries," in *Proc. 21st USENIX Secur. Symp.*, 2012, pp. 285–300.

[28] A. Shelat and C.-H. Shen, "Fast two-party secure computation with minimal assumptions," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 523–534.

[29] Y. Huang, J. Katz, and D. Evans, "Efficient secure two-party computation using symmetric cut-and-choose," in *Proc. Annu. Cryptol. Conf.* Cham, Switzerland: Springer, 2013, pp. 18–35.

[30] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits," in *Proc. 20th USENIX Secur. Symp.*, 2011, pp. 1–16.

[31] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Proc. Annu. Cryptol. Conf.* Cham, Switzerland: Springer, 2012, pp. 643–662.

[32] M. Keller, "MP-SPDZ: A versatile framework for multi-party computation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 1575–1590.

[33] D. Demmler, T. Schneider, and M. Zohner, "ABY—A framework for efficient mixed-protocol secure two-party computation," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–15.

[34] A. Patra, T. Schneider, A. Suresh, and H. Yalame, "ABY2.0: Improved mixed-protocol secure two-party computation," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 2165–2182.

[35] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "GAZELLE: A low latency framework for secure neural network inference," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 1651–1669.

[36] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, "Chameleon: A hybrid secure computation framework for machine learning applications," in *Proc. Asia Conf. Comput. Commun. Secur.*, May 2018, pp. 707–721.

[37] A. Choudhury, J. Loftus, E. Orsini, A. Patra, and N. P. Smart, "Between a rock and a hard place: Interpolating between MPC and FHE," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Cham, Switzerland: Springer, 2013, pp. 221–240.

[38] L. K. L. Ng and S. S. M. Chow, "SoK: Cryptographic neural-network computation," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 497–514.

[39] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.

[40] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (poly) logarithmic overhead," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 1253–1269.

[41] J. So et al., "LightSecAgg: A lightweight and versatile design for secure aggregation in federated learning," in *Proc. Mach. Learn. Syst. (MLSys)*, 2022, pp. 694–720.

[42] Y. Zhao and H. Sun, "Information theoretic secure aggregation with user dropouts," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2021, pp. 1124–1129.

[43] C. Gentry and D. Boneh, *A Fully Homomorphic Encryption Scheme*, vol. 20, no. 9. Stanford, CA, USA: Stanford Univ., 2009.

[44] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, May 2009, pp. 169–178.

[45] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 201–210.

[46] T. Graepel, K. Lauter, and M. Naehrig, "ML confidential: Machine learning on encrypted data," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Cham, Switzerland: Springer, 2012, pp. 1–21.

[47] J. Yuan and S. Yu, "Privacy preserving back-propagation neural network learning made practical with cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 212–221, Jan. 2014.

[48] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network," *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 35, Mar. 2017.

[49] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Comput.*, vol. 21, no. 1, pp. 277–286, Mar. 2018.

[50] A. Kim, Y. Song, M. Kim, K. Lee, and J. H. Cheon, "Logistic regression model training based on the approximate homomorphic encryption," *BMC Med. Genomics*, vol. 11, no. 4, p. 83, Oct. 2018.

[51] Q. Wang et al., "Privacy-preserving collaborative model learning: The case of word vector training," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2381–2393, Dec. 2018.

[52] K. Han, S. Hong, J. H. Cheon, and D. Park, "Logistic regression on homomorphic encrypted data at scale," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 9466–9471.

[53] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptography Conf.* Cham, Switzerland: Springer, 2006, pp. 265–284.

[54] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. Adv. Neural Inf. Proc. Syst.*, 2009, pp. 1–8.

[55] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 909–910.

[56] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[57] M. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *Adv. Neural Inf. Process. Syst.*, 2010, pp. 1876–1884.

[58] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.

[59] A. Rajkumar and S. Agarwal, "A differentially private stochastic gradient descent algorithm for multiparty classification," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, vol. 22, Apr. 2012, pp. 933–941.

[60] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6346–6357.

[61] W.-N. Chen, A. Ozgur, and P. Kairouz, "The Poisson binomial mechanism for unbiased federated learning with secure aggregation," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 3490–3506.

[62] W.-N. Chen, C. A. C. Choo, P. Kairouz, and A. T. Suresh, "The fundamental price of secure aggregation in differentially private federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 3056–3089.

[63] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete Gaussian mechanism for federated learning with secure aggregation," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5201–5212.

[64] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, vol. 398. Hoboken, NJ, USA: Wiley, 2013.

[65] T. Nguyen and S. Sanner, "Algorithms for direct 0–1 loss optimization in binary classification," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1085–1093.

[66] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *J. Amer. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, Mar. 2006.

[67] S. Ben-David, N. Eiron, and P. M. Long, "On the difficulty of approximately maximizing agreements," *J. Comput. Syst. Sci.*, vol. 66, no. 3, pp. 496–514, May 2003.

[68] V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu, "Agnostic learning of monomials by halfspaces is hard," *SIAM J. Comput.*, vol. 41, no. 6, pp. 1558–1590, Jan. 2012.

[69] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 1–26.

[70] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[71] M. C. Thomas and A. T. Joy, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.

[72] X. Lu, H. U. Sami, and B. Güler, "SCALR: Communication-efficient secure multi-party logistic regression," *IEEE Trans. Commun.*, early access, doi: 10.1109/TCOMM.2023.3308954.

[73] H. U. Sami and B. Güler, "Secure aggregation for clustered federated learning," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2023, pp. 186–191.

[74] C. Cachin, R. Guerraoui, and L. Rodrigues, *Introduction to Reliable and Secure Distributed Programming*. Berlin, Germany: Springer, 2011.

[75] M. M. Amiri and D. Gündüz, "Computation scheduling for distributed machine learning with straggling workers," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6270–6284, Dec. 2019.

[76] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[77] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[78] N. C. Codella et al., "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging," in *Proc. IEEE 15th Int. Symp. Biomed. Imag.*, Feb. 2018, pp. 168–172.

[79] J. Brinkhuis and V. Tikhomirov, *Optimization: Insights and Applications*. Princeton, NJ, USA: Princeton Univ. Press, 2005.

[80] O. Catrina and A. Saxena, "Secure computation with fixed-point numbers," in *Proc. Int. Conf. Financial Cryptography Data Secur.* Cham, Switzerland: Springer, 2010, pp. 35–50.

[81] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Apr. 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[82] Y. LeCun, C. Cortes, and C. Burges. (210). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist

[83] L. Dalcín, R. Paz, and M. Storti, "MPI for Python," *J. Parallel Distrib. Comput.*, vol. 65, no. 9, pp. 1108–1115, Sep. 2005.

[84] K. S. Kedlaya and C. Umans, "Fast polynomial factorization and modular composition," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1767–1802, Jan. 2011.

Xingyu Lu received the Bachelor of Engineering degree from the Computer Science and Information Technology Department, Zhejiang Gongshang University, China, in 2019, and the Master of Science degree in robotics (computer science) from the Khoury College of Computer Science and the College of Engineering, Northeastern University, Boston, USA, in 2021. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of California at Riverside. His research interests include private machine learning, distributed learning, and federated learning.

Hasin Us Sami (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California at Riverside. His research interests include federated and distributed machine learning, information theory, secure and private computing, and wireless networks.

Başak Güler (Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University (METU), Ankara, Turkey, and the Ph.D. degree from the Wireless Communications and Networking Laboratory, The Pennsylvania State University, in 2017. From 2018 to 2020, she was a Postdoctoral Scholar with the University of Southern California. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of California at Riverside. Her research interests include information theory, distributed computing, machine learning, and wireless networks. She has received the NSF CAREER Award in 2022.