# Robust and Privacy-Preserving Decentralized Online Federated Learning for Streaming Data with Outliers

Yushi Wang, Yang Lu*, and Qiang Ni

***Abstract*—This paper addresses the challenging problem of online federated learning (FL) over streaming data in a decentralized communication network. To enable rapid adaptation to new observations, we develop novel non-parametric model-based local training, not deep neural network-based approaches as adopted in most previous studies. In particular, we integrate Gaussian process regression with a Student-t likelihood to improve robustness against data outliers. For global model aggregation, we propose a consensus-based Product of Experts (PoE) algorithm that enables peer-to-peer fusion of non-parametric local models and preserves robustness to outliers. To ensure privacy, we develop a secure aggregation scheme that combines Shamir's secret sharing (SSS) with public-key encryption. Compared to existing methods, the proposed approach enhances privacy guarantees for learners with limited connectivity in sparse graphs. Theoretical analyses establish robustness, correctness, and privacy properties. Extensive numerical experiments validate the effectiveness of the proposed algorithm.**

***Index Terms*—Decentralized Federated Learning; Online Learning; Robustness to Outliers; Privacy Preservation**

## I. Introduction

In many real-world applications, data is continuously generated in the form of streams across geographically distributed edge devices. Autonomous driving networks must make real-time decisions based on continuous sensory data streams [1]; mobile edge networks rely on immediate predictions of dynamic traffic flows [2]; smart manufacturing [3] and healthcare systems [4] require continuous analysis of time-varying production or physiological data. Unlike traditional static datasets, data in these scenarios exhibits distribution shifts over time. Due to bandwidth constraints, low-latency requirements, and privacy concerns regarding raw data, centralized training is often infeasible. There is thus a need for decentralized online learning paradigms that allow dispersed devices to learn locally from streaming data while collaborating to update global models.

Federated Learning (FL) has emerged as a privacy-preserving collaborative framework where learners train models locally and share only model updates rather than raw data. Recent studies and surveys have shown that FL continues to evolve rapidly, with growing attention to data heterogeneity, communication efficiency, robustness, and privacy protection in distributed edge environments [5], [6]. However, mainstream FL approaches [7], [8] face fundamental limitations in streaming scenarios. Most FL research relies on Deep Neural Networks (DNNs) as local models, focusing on offline learning with sufficient static data. Although recent studies [9], [10] have attempted to extend DNN-based FL to online settings by updating models via weighted combinations of old and new parameters, DNNs remain data-hungry. In streaming windows where only limited new samples are available, DNNs often fail to adapt rapidly to drastic shifts in data distributions. To address this limitation, non-parametric models such as Gaussian Process Regression (GPR) have been introduced to FL [11], [12]. Unlike parameterized networks, non-parametric approaches update function distributions directly through probabilistic inference, enabling faster adaptation to streaming data.

However, implementing robust online learning with non-parametric models in decentralized networks still faces several challenges. In streaming scenarios, the impact of outliers is significantly amplified compared to offline training. Unlike static analysis where anomalies can be identified and removed from the full dataset, online learners update parameters sequentially based on small real-time mini-batches. Consequently, a single corrupted sample can disproportionately skew the local model statistics. Standard GPR models rely on Gaussian likelihoods with thin tails [13], making them highly sensitive to such deviations. In decentralized networks, this vulnerability is significant as corrupted local updates propagate through aggregation, affecting multiple learners [14], [15]. Unlike centralized FL, where a server can detect and discard anomalous updates using robust aggregation rules such as Krum [16] or Trimmed Mean [17], decentralized settings lack a coordinating authority to enforce such global mitigation. These rules mainly act on client-level model updates, while our setting requires outlier suppression within local non-parametric posterior inference before peer-to-peer aggregation.

A second challenge arises in aggregating non-parametric local models in serverless federated learning. Recent decentralized FL studies have further investigated serverless learning from the perspectives of communication efficiency, robustness, personalization, and privacy protection in peer-to-peer networks [18], [19]. However, in GP-based FL, local learners maintain posterior distributions rather than shared parameter vectors, so global models cannot be obtained through direct summation or parameter averaging. Existing approaches therefore employ Product of Experts (PoE) fusion to aggregate local posteriors [11], but such schemes are mainly developed for centralized settings with a coordinating server. In serverless

This work was partially supported by the Fundamental Research Funds for the Central Universities under Grant 14380002.

Yushi Wang and Qiang Ni are with the School of Computing and Communications, Lancaster University, Lancaster, LA1 4YW, UK (e-mail: y.wang216@lancaster.ac.uk; q.ni@lancaster.ac.uk)

Yang Lu (Corresponding author) is with the School of Robotics and Automation, Nanjing University, Suzhou 215163, China (e-mail: luyang@nju.edu.cn)

FL, the absence of a central aggregator prevents the direct use of centralized PoE and requires the aggregation of non-parametric posterior models to be realized through peer-to-peer communication.

Privacy preservation compounds these challenges. Recent studies on security and privacy in decentralized FL highlight that peer-to-peer model exchange removes server-side risks but introduces topology-dependent privacy threats, especially when adversaries can observe or collude over local communication links [20]. Aggregating posterior distributions requires exchanging means and variances, which is far richer statistical information than the gradients transmitted in parametric FL. This information provides a larger attack surface for model inversion attacks [21], [22], where adversaries can infer private training data from observed model updates. Various privacy-preserving consensus methods have been proposed. Random perturbation methods [23], [24] degrade model utility through added noise. Decaying or correlated noise injection [25], [26] may still allow information inference through entropy-based analysis. Homomorphic encryption [27], [28] typically relies on a central aggregator, making it incompatible with fully decentralized architectures. Shamir's secret sharing (SSS) has been extended to decentralized FL to provide perfect secrecy without sacrificing model utility [29], but it relies on the assumption that each learner has at least one non-colluding neighbor. In sparse communication graphs—common in rural IoT deployments, mobile ad-hoc networks, or infrastructure-limited environments—learners often have only a small number of neighbors, which substantially increases the likelihood that all neighbors of a given learner are adversarial. As a result, learners with limited connectivity face a significantly higher risk of privacy leakage under existing SSS-based decentralized schemes.

### A. Contribution Statement

To address these challenges, we propose a framework for robust and privacy-preserving decentralized online federated learning that resolves the structural conflicts between outlier resistance, decentralized consensus, and privacy preservation. The main contributions of our work are summarized as follows:

1) Robust local training: We propose the first non-parametric model-based algorithm that integrates GPR with a Student-t likelihood, enabling both rapid adaptation to new observations and robustness to data outliers in online learning.
2) Decentralized global model aggregation: We design a consensus-based PoE algorithm that, for the first time, enables peer-to-peer aggregation of non-parametric local models in a decentralized setting while maintaining robustness to outliers.
3) Privacy-preserving aggregation: We develop a novel algorithm that combines SSS with public-key encryption to ensure secure and accurate model aggregation. Compared to state-of-the-art approaches, our method enhances privacy guarantees for individual learners, particularly those with few neighbors in sparse communication graphs.
4) Rigorous theoretical analysis: We provide formal analysis proving the robustness, correctness, and privacy guarantees of the proposed algorithm, offering solid theoretical foundations for its deployment in decentralized and adversarial environments.
5) Comprehensive experimental validation: We conduct extensive numerical experiments to evaluate the effectiveness of the proposed algorithm under various online FL scenarios, demonstrating superior performance in terms of accuracy, robustness to outliers, and privacy preservation.

### B. Organization

The rest of this paper is organized as follows: Section II presents motivating examples, while Section III formalizes the problem. Section IV outlines the framework and key challenges. Section V details the algorithm design, including robust local training and secure aggregation. Theoretical analyses of robustness, correctness, and privacy are provided in Section VI. Experimental results are discussed in Section VII, and Section VIII concludes the paper with future directions.

## II. Motivation Examples

In many real-world applications, learning must be performed continuously from streaming data in a distributed setting, often without access to a central server. This raises four core challenges: (i) handling evolving streaming data through online learning, (ii) resisting the influence of data outliers, (iii) enabling decentralized operation without relying on a central coordinator, and (iv) preserving the privacy of sensitive information. We illustrate the significance of these challenges with two representative examples.

*1) Autonomous Vehicle Networks:* Autonomous vehicles rely on continuous streams of sensory data (e.g., LiDAR, radar, and cameras), which exhibit high variability, making online learning essential for adapting to distribution shifts. This data is often contaminated with outliers from sources like sensor noise or hardware malfunctions. Moreover, these systems frequently operate where centralized coordination is infeasible, requiring vehicles to learn cooperatively through decentralized communication. This exchange, however, may expose sensitive information like location traces or behavioral patterns, making privacy preservation essential.

*2) Healthcare Monitoring Systems:* Wearable devices in healthcare systems continuously generate physiological data (e.g., heart rate, blood pressure, and activity levels), requiring online learning to adapt to a patient's changing health and the resulting distribution shifts. Sensor errors or patient movement can introduce outliers, distorting predictions. These systems are typically deployed across a decentralized network of devices where centralized aggregation is impractical. Moreover, as healthcare data is intrinsically sensitive, it requires careful privacy protection for compliance with regulations (e.g., HIPAA, GDPR) and to maintain user trust.

## III. Problem Formulation

This section defines the communication topology, observation model, privacy considerations, and objectives.

### A. Communication Graph

Consider a distributed network with $N$ learners, indexed by the set $\mathcal{V} = \{1, \ldots, N\}$. The communication topology is represented by a fixed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges representing the communication links between learners. For each learner $i \in \mathcal{V}$, we denote its set of one-hop neighbours as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. The set of two-hop neighbours for learner $i$ is denoted by $\mathcal{N}_i^*$, $\mathcal{N}_i^* = \{l \in \mathcal{V} : \exists j \in \mathcal{N}_i \text{ s.t. } (j, l) \in \mathcal{E}, l \neq i, \text{ and } l \notin \mathcal{N}_i\}$. We impose the following assumption on the communication graph $\mathcal{G}$.

*Assumption 1:* $\mathcal{G}$ is undirected, connected and time invariant.

*Remark 1:* Assumption 1 is standard for distributed learning, optimization and control [30], [31]. The condition of time invariance can be relaxed, for instance, by using dynamic weighting schemes as discussed in Remark 2 of Section V.

### B. Observation Model for Streaming Data

Each learner $i$ observes a stream of data $\{(x_i^{(t)}, y_i^{(t)})\}$, where $x_i^{(t)} \in \mathcal{X} \subseteq \mathbb{R}^n$, $y_i^{(t)} \in \mathcal{Y} \subseteq \mathbb{R}$. These data arrive sequentially, reflecting a streaming environment in which new samples continuously appear over time. At each time instant $t$, the relationship between the input $x_i^{(t)}$ and the output $y_i^{(t)}$ is governed by the observation model: $y_i^{(t)} = f(x_i^{(t)}) + \epsilon_i^{(t)}$, where $f : \mathcal{X} \to \mathbb{R}$ is an unknown shared function. The noise term $\epsilon_i^{(t)}$ is independent across the learners and time steps, and follows a Gaussian distribution $\epsilon_i^{(t)} \sim \mathcal{N}(0, \sigma_e^2)$.

The collective objective for the learners is to collaboratively learn the shared function $f$ by leveraging the communication graph $\mathcal{G}$ defined in Section III-A.

### C. Robustness and Privacy Issues

The learning problem introduced above presents two key challenges:

*1) Robustness to Label Outliers:* The observed labels $y_i^{(t)}$ may contain outliers defined as significant deviations from the latent function value: $|y_i^{(t)} - f(x_i^{(t)})| > \kappa\sigma_e \quad (\kappa \geq 3)$, where $\sigma_e$ is the noise standard deviation and $\kappa \geq 3$ follows 3-sigma rule [32], [33]. These outliers may arise due to various factors, such as sensor failures, adversarial attacks, or environmental changes. In a distributed network, the presence of outliers is particularly damaging, as a single corrupted label can propagate errors and bias the models of multiple learners during collaborative updates.

*2) Privacy Preservation:* The protocol for distributed learning requires continuous data exchange among learners, which inherently creates attack surfaces for privacy breaches. We consider the semi-honest attack model, i.e., attackers obey all the rules of the protocol and do not attempt to disrupt the execution mechanism, but they collect and analyze all available data to infer legitimate entities' private information [34]. In addition, we allow collaborations between attackers. Assume all communications between the learners are attack-free.

### D. Objectives

This paper aims to develop a novel FL algorithm for online, collaborative learning of the function $f$ in a fully decentralized manner. Particularly, the algorithm will simultaneously: (i) achieve online learning over streaming data and resist outliers over distributed learners; (ii) enable peer-to-peer global model aggregation and ensure its correctness; (iii) guarantee privacy of legitimate learners' local data.

TABLE I
SUMMARY OF MAIN NOTATIONS.

| Notation | Description |
|---|---|
| $V$, $N$ | Learner set and number of learners. |
| $G = (V, E)$ | Communication graph. |
| $N_i$, $N_i^*$ | One-hop and two-hop neighbors of learner $i$. |
| $D_i^{(t)}$ | Local data of learner $i$ up to round $t$. |
| $f(\cdot)$ | Unknown function to be learned. |
| $Q$ | Number of nearest neighbors for local GPR. |
| $\hat{\mu}_i^{(t)}$, $(\hat{\sigma}_i^{(t)})^2$ | Local predictive mean and variance. |
| $\bar{\mu}^{(t)}$, $(\bar{\sigma}^{(t)})^2$ | Ideal PoE global mean and variance. |
| $\tilde{\mu}_i^{(t)}$, $(\tilde{\sigma}_i^{(t)})^2$ | Recovered global mean and variance. |
| $\hat{\theta}_i^{(t)}$ | Encoded local posterior state. |
| $\tilde{\theta}_i^{(t)}$ | Recovered global posterior state. |
| $A = [a_{ij}]$ | Consensus weight matrix. |
| $M$ | Number of consensus iterations. |
| $s_i^{(t)}(m)$ | Consensus state at iteration $m$. |
| $p$ | Prime modulus. |
| $\delta$ | Real-to-integer scaling factor. |
| $n_{\text{nei}}$ | Low-degree threshold. |

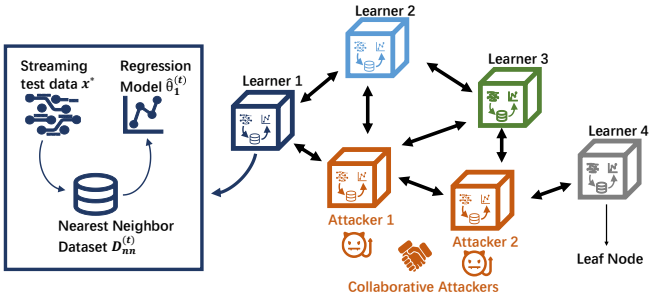



Fig. 1. Overview of a unified framework for robust and privacy-preserving online learning in decentralized networks.

### E. Notation

For readability, Table I summarizes the main notations used throughout the paper. Unless otherwise specified, the superscript $(t)$ denotes the online learning round, and $\hat{(\cdot)}$, $\bar{(\cdot)}$, and $\tilde{(\cdot)}$ denote local, ideal aggregated, and recovered decentralized quantities, respectively.

## IV. Framework Overview and Key Challenges

### A. Framework Overview

Our approach unifies decentralized FL with non-parametric local training (Fig. 1). This combination leverages the strengths of both paradigms:

(1) Decentralized FL: Enables peer-to-peer learning without central server and enhances privacy by keeping raw data local while allowing collaborative model updates.

(2) Non-parametric model-based local training: Unlike DNNs, which require costly retraining to adapt to evolving data distributions, non-parametric model-based local training provides a more flexible and adaptive solution for streaming dynamic data.

### B. Key Challenges and High-Level Solutions

While this framework is promising, simply combining these paradigms is insufficient. We address three fundamental challenges with high-level solutions, with detailed design and analysis presented in Sections V and VI.

1) Robustness against Outliers: State-of-the-art non-parametric FL relies on GP models, whose thin-tailed likelihoods are highly sensitive to outliers. This vulnerability is amplified in decentralized settings because (i) Numerous learners increase the likelihood of outlier contamination; (ii) The absence of a central server prevents coordinated outlier correction; (iii) The aggregation process can propagate erroneous updates, amplifying their impact. Our solution replaces the standard GP with a heavy-tailed Student-t distribution to down-weight anomalous observations.

2) Decentralized Aggregation: Unlike parametric models that simply average weight updates, non-parametric models maintain complex posterior distributions (including mean and variance), which complicates peer-to-peer aggregation. We design a consensus-based PoE protocol and will rigorously establish its robustness, ensuring outlier resistance is maintained in the aggregated model across the network.

3) Privacy Preservation: The state-of-the-art approach for privacy-preserving decentralized FL uses SSS, which is vulnerable as it requires each learner to have at least one benign neighbor. This creates a significant risk in sparse networks, especially for a leaf learner with a single connection, who is compromised if their few neighbors are adversarial. The issue is exacerbated by non-parametric models, whose detailed statistical distributions are more susceptible to privacy breaches than traditional parametric models. We propose a hybrid mechanism that integrates SSS with asymmetric encryption. The correctness and privacy properties of the proposed method are rigorously established.

## V. Algorithm Design

This section presents the details of the proposed algorithm. First, we introduce a robust local training mechanism that employs a Student-t likelihood to handle data outliers. Second, we present a secure peer-to-peer aggregation protocol for achieving consensus on non-parametric models while protecting privacy. Finally, we provide the overall algorithm.

### A. Robust Online Local Training

For robust local learning with outliers, we depart from standard GPR. We instead incorporate a heavy-tailed Student-t likelihood [35], which is formally defined by the probability density function $\mathcal{T}(y|f(x),\nu,\sigma_s) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi\sigma_s^2}}\left(1+\frac{(y-f(x))^2}{\nu\sigma_s^2}\right)^{-\frac{\nu+1}{2}}$. In our model, for each learner $i \in \mathcal{V}$, we assume the observation $y_i^{(t)}$ given the latent function value $f(x_i^{(t)})$ follows this distribution, i.e., $y_i^{(t)}|f(x_i^{(t)}) \sim \mathcal{T}(y_i^{(t)}|f(x_i^{(t)}),\nu_i,\sigma_{s,i})$, where $\nu_i$ is the degree of freedom and $\sigma_{s,i}$ is the scale parameter. Student-t distributions are not heavily influenced by outliers.

However, this robustness creates a computational challenge, as the non-conjugate Student-t likelihood and Gaussian prior result in an analytically intractable posterior. To overcome this, we employ the Laplace approximation method, a reliable and numerically stable technique for estimating the posterior [36], [37]. The complete local training process is detailed in Algorithm 1.

**Algorithm 1:** Local Learning with GPR using Student-t Likelihood and Laplace Approximation

**Syntax:** $\hat{\mu}_i(x^*)$ , $(\hat{\sigma}_i(x^*))^2 = \text{Alg}_1(X^*, D_i^{(t)}, \nu_i, \sigma_{s,i})$

Set Newton-Raphson stopping threshold $\varepsilon$ for $\|\nabla f\|$ and maximum steps $R$

**foreach** $x^* \in X^*$ **do**

  Select local neighbors: $D_{i,x^*}^{(t)} \leftarrow \text{kNN}(x^*, D_i^{(t)}, Q)$

  Compute kernel matrix: $K_{i,x^*}^{(t)} \leftarrow k\left(X_{i,x^*}^{(t)}, X_{i,x^*}^{(t)}\right)$

  Initialize latent function:

  $\hat{f}_{i,x^*}^{(t)} \leftarrow K_{i,x^*}^{(t)}\left(K_{i,x^*}^{(t)} + \sigma_{s,i}^2 I\right)^{-1} y_{i,x^*}^{(t)}$

  `/* Laplace approximation: refine latent posterior via Newton steps */`

  **for** $r = 1$ **to** $R$ **do**

    `/* stop early if` $\|\nabla f\| < \varepsilon$ `*/`

    Compute gradient: $\nabla f = \frac{(\nu_i+1)(y_{i,x^*}^{(t)} - \hat{f}_{i,x^*}^{(t)})}{(y_{i,x^*}^{(t)} - \hat{f}_{i,x^*}^{(t)})^2 + \nu_i\sigma_{s,i}^2} - \left(K_{i,x^*}^{(t)}\right)^{-1}\hat{f}_{i,x^*}^{(t)}$

    Compute Hessian matrix: $H_{i,x^*}^{(t)} = \left(K_{i,x^*}^{(t)}\right)^{-1} - \text{diag}\left(\frac{(\nu_i+1)(y_{i,x^*}^{(t)} - \hat{f}_{i,x^*}^{(t)})^2 - \nu_i\sigma_{s,i}^2}{((y_{i,x^*}^{(t)} - \hat{f}_{i,x^*}^{(t)})^2 + \nu_i\sigma_{s,i}^2)^2}\right)$

    Newton-Raphson update:

    $\hat{f}_{i,x^*}^{(t)} \leftarrow \hat{f}_{i,x^*}^{(t)} - H^{-1}\left(\hat{f}_{i,x^*}^{(t)}\right)\nabla f\left(\hat{f}_{i,x^*}^{(t)}\right)$

  `/* Robust prediction: Student-t mean and variance */`

  Compute predictive mean:

  $\hat{\mu}_i(x^*|D_{i,x^*}^{(t)}) = k(x^*, X_{i,x^*}^{(t)})\left(K_{i,x^*}^{(t)}\right)^{-1}\hat{f}_{i,x^*}^{(t)}$

  Compute predictive variance: $(\hat{\sigma}_i(x^*|D_{i,x^*}^{(t)}))^2 = k(x^*,x^*) - k(x^*, X_{i,x^*}^{(t)})\left(K_{i,x^*}^{(t)}\right)^{-1}k(X_{i,x^*}^{(t)}, x^*)$

  Apply Laplace correction:

  $(\hat{\sigma}_i(x^*|D_{i,x^*}^{(t)}))^2 \leftarrow (\hat{\sigma}_i(x^*|D_{i,x^*}^{(t)}))^2 + k(x^*, X_{i,x^*}^{(t)})(H_{i,x^*}^{(t)})^{-1}k(X_{i,x^*}^{(t)}, x^*)$

Algorithm 1 uses efficient nearest-neighbor selection for streaming data. At each round $t$, learner $i$ possesses a history of training data, $D_i^{(t)}$, formally defined as $D_i^{(t)} = \left\{\left(x_i^{(1)}, y_i^{(1)}\right), \ldots, \left(x_i^{(t)}, y_i^{(t)}\right)\right\}$. To manage com-

plexity when making a prediction for a test input $x^*$, a local working subset $D_{i,x^*}^{(t)}$ is formed. This selection process is defined as $D_{i,x^*}^{(t)} = \text{kNN}\left(x^*, D_i^{(t)}, Q\right)$, which denotes the selection of the $Q$ nearest neighbors of $x^*$ from $D_i^{(t)}$ (line 3). This step focuses the regression on the most relevant data points. We further define the input features and corresponding target values of this subset as $X_{i,x^*}^{(t)} = \{x_{i,j}\}_{j=1}^{Q}$ and $y_{i,x^*}^{(t)} = \{y_{i,j}\}_{j=1}^{Q}$, where $(x_{i,j}, y_{i,j}) \in D_{i,x^*}^{(t)}$.

The core of the Laplace approximation is to build a Gaussian approximation centered at the mode of the true posterior. Lines 6-11 are dedicated to finding this mode—the maximum a posteriori (MAP) estimate $\hat{f}_{i,x^*}^{(t)}$. This is achieved via a Newton-Raphson optimization loop [38], which begins with an initial estimate based on a standard GP posterior mean (line 6). Here, $K_{i,x^*}^{(t)}$ denotes the kernel matrix evaluated on $X_{i,x^*}^{(t)}$ using kernel function $k(\cdot,\cdot)$, i.e., $K_{i,x^*}^{(t)} = k(X_{i,x^*}^{(t)}, X_{i,x^*}^{(t)})$. In each iteration, the latent function values $\hat{f}_{i,x^*}^{(t)}$ are updated using the gradient (line 8) and the Hessian matrix (line 9) of the log-posterior. The inverse of the final Hessian, ${H_{i,x^*}^{(t)}}^{-1}$, provides the covariance for this Gaussian approximation, quantifying the local curvature around the MAP estimate.

Once this Gaussian approximation of the posterior is established, the predictive distribution for the test point $x^*$ can be computed. Line 13 calculates the predictive mean in a manner analogous to standard GPR, conditioning on the converged MAP solution. The predictive variance calculation begins with the standard GPR formula (line 14) but is then refined by a crucial correction term (line 15). This term incorporates the Hessian from the Laplace approximation, ensuring that the final variance reflects the uncertainty captured by the more robust Student-t model.

### B. Secure Decentralized Non-parametric Model Aggregation

This section introduces our secure, decentralized algorithm for aggregating the non-parametric models learned by each learner. Our approach achieves the aggregation objective of the PoE algorithm by merging a consensus protocol for decentralization with a novel dual-mode security scheme for privacy preservation, shown in Algorithm 2.

*1) Aggregation Goal and Challenges:* The PoE algorithm [39] provides the ideal mathematical foundation for this task, combining probabilistic models by weighting them according to their reliability. Following this approach, also employed by Zhang et al. [11], the aggregated global posterior's mean $\bar{\mu}^{(t)}$ and variance $(\bar{\sigma}^{(t)})^2$ are given by:

$$\bar{\mu}^{(t)} = \frac{\sum_{i=1}^{N} (\hat{\sigma}_i^{(t)})^{-2} \hat{\mu}_i^{(t)}}{\sum_{i=1}^{N} (\hat{\sigma}_i^{(t)})^{-2}}, \qquad (\bar{\sigma}^{(t)})^2 = \left(\sum_{i=1}^{N} (\hat{\sigma}_i^{(t)})^{-2}\right)^{-1},$$

where $\left(\hat{\mu}_i^{(t)}, (\hat{\sigma}_i^{(t)})^2\right)$ are the local predictions of learner $i$. As shown, the joint computation reduces to evaluating two global sums over the network. To facilitate this, each learner $i$ defines a two-element local state $\hat{\theta}_i^{(t)}(0)$ and initializes it with its local model in line 1, i.e., $\hat{\theta}_i^{(t)}(0) = \left((\hat{\sigma}_i^{(t)})^{-2}\hat{\mu}_i^{(t)}, (\hat{\sigma}_i^{(t)})^{-2}\right)$. Here, we use the notation $v[\ell]$ to refer to the $\ell$-th component of a vector $v$. For example, the first component of the local state is $\hat{\theta}_i^{(t)}(0)[0] = (\hat{\sigma}_i^{(t)})^{-2}\hat{\mu}_i^{(t)}$ and the second is $\hat{\theta}_i^{(t)}(0)[1] = (\hat{\sigma}_i^{(t)})^{-2}$.

While straightforward in centralized settings, applying PoE in a decentralized environment presents two challenges: the global sums must be computed without a central server, and the peer-to-peer exchange of raw state vectors $\hat{\theta}_i^{(t)}$ would leak sensitive information. To overcome these, our algorithm integrates the PoE formulation with a consensus protocol and SSS, enabling fully decentralized and privacy-preserving model computation.

*2) Decentralized Consensus Foundation:* To enable usage of the consensus algorithm, in the communication graph $\mathcal{G}$, we introduce weighted adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{N\times N}$, where $a_{ij}$ represents the weight of the edge between nodes $i$ and $j$. The convergence of this process is guaranteed under standard assumptions on the network's weighted adjacency matrix, as detailed below.

*Assumption 2:* $A$ satisfies the following properties:

1) $A$ is doubly stochastic: $\sum_{j=1}^{N} a_{ij} = 1, \quad \sum_{i=1}^{N} a_{ij} = 1, \quad \forall i, j \in \mathcal{V}$.
2) $a_{ij} > 0$ if and only if $(i,j) \in \mathcal{E}$ or $i = j$; otherwise, $a_{ij} = 0$.
3) $A$ satisfies the spectral radius condition for consensus: $\rho\left(A - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T\right) < 1$, where $\rho(\cdot)$ denotes the spectral radius of a square matrix, and $\mathbf{1}_N$ is the $N$-dimensional column vector with all entries equal to 1.

This is a standard assumption widely used in consensus-based distributed optimization systems to ensure convergence and stability of the consensus process [30], [40], [41].

*Remark 2:* By adopting the Metropolis-Hastings (MH) method of [29], one way to achieve Assumption 2 is shown as: $a_{ij}^{(t)} = \begin{cases} \frac{1}{\max\{|\mathcal{N}_i|,|\mathcal{N}_j|\}+1}, & \text{if } j \in \mathcal{N}_i \\ 1 - \sum_{j\in\mathcal{N}_i} \frac{1}{\max\{|\mathcal{N}_i|,|\mathcal{N}_j|\}+1}, & \text{if } j = i \end{cases}$ where $|\cdot|$ denotes the cardinality of a set, and $\mathcal{N}_i$ represents the set of neighbors of node $i$ in the communication graph. This rule also ensures that $A$ is doubly stochastic. The MH method can be used to handle time-varying communication graphs by dynamically updating the weights according to the current graph structure. However, since handling time-varying graphs is not the contribution of the current paper, and to simplify the notation, this paper focuses on the time-invariant case.

To achieve decentralized model aggregation, a well-known approach is the average consensus algorithm. In this algorithm, each learner $i$ starts with an initial state $\hat{\theta}_i^{(t)}(0)$ and iteratively updates its state by exchanging model with its one-hop neighbors according to the update rule:

$$\hat{\theta}_i^{(t)}(m+1) = a_{ii}\hat{\theta}_i^{(t)}(m) + \sum_{j\in\mathcal{N}_i} a_{ij}\hat{\theta}_j^{(t)}(m). \tag{1}$$

The convergence of this iterative process is established by the following lemma.

*Lemma 1:* [40] Consider the iterative process defined by (1). For any initial joint state $\{\hat{\theta}_j^{(t)}(0)\}_{j\in\mathcal{V}}$, the states of all learners achieve average consensus, i.e.,

$$\lim_{m\to\infty} \hat{\theta}_i^{(t)}(m) = \frac{1}{N}\sum_{j\in\mathcal{V}} \hat{\theta}_j^{(t)}(0), \quad \forall i \in \mathcal{V}, \tag{2}$$

**Algorithm 2:** Secure Decentralized Non-parametric Model Aggregation

**Syntax:** $(\tilde{\mu}_i^{(t)}, (\tilde{\sigma}_i^{(t)})^2) = \text{Alg}_2\big(i, \hat{\mu}_i^{(t)}, (\hat{\sigma}_i^{(t)})^2, A, M, k_{pub|i}^{(t)}, k_{priv|i}^{(t)}, p\big)$

/* Encode local posterior */

1 Learner $i$ initializes $\hat{\theta}_i^{(t)}(0)[0] = (\hat{\sigma}_i^{(t)})^{-2}\hat{\mu}_i^{(t)}, \quad \hat{\theta}_i^{(t)}(0)[1] = (\hat{\sigma}_i^{(t)})^{-2}$

/* Shamir share generation & forwarding */

2 **foreach** $i \in \mathcal{V}$ **do**

3 **if** $|\mathcal{N}_i| <= n_{nei}$ **then**

4 **foreach** $j \in \mathcal{N}_i$ **do**

5 Learner $i$ picks $\tau = |\mathcal{N}_j|+1$ scalars $c_1, \ldots, c_\tau \in \mathbb{Z}_p$ and generates $\mathcal{H}_i^{l(t)}(\eta) = 10^\delta \hat{\theta}_i^{(t)} + c_1\eta + \cdots + c_\tau\eta^\tau$

6 **foreach** $l \in \mathcal{N}_j$ **do**

7 Learner $i$ constructs $\hat{S}_i^{l(t)}$ by (4)

8 Learner $i$ encrypts $\hat{S}_i^{l(t)}$ as $C_i^{l(t)}$ with $k_{pub|l}^{(t)}$ and sends it to $l$ through $j$

9 Learner $l$ decrypts $\hat{S}_i^{l(t)}$ via $k_{priv|l}^{(t)}$

10 Learner $l$ picks $\tau' = |\mathcal{N}_l|$ scalars $d_1, \ldots, d_{\tau'} \in \mathbb{Z}_p$ and generates $\mathcal{H}_l^{j(t)}(\eta) = 10^\delta \hat{\theta}_l^{(t)} + d_1\eta + \cdots + d_{\tau'}\eta^{\tau'}$

11 Learner $l$ computes $S_l^{j(t)} = \mathcal{H}_l^{j(t)}(j) \bmod p$

12 Learner $l$ computes $\hat{S}_l^{j(t)} = S_l^{j(t)} + \hat{S}_i^{l(t)}$ and sends it to $j$

13 **else**

14 Learner $i$ picks $\tau = |\mathcal{N}_i|$ scalars $c_1, \ldots, c_\tau \in \mathbb{Z}_p$ and generates $\mathcal{H}_i^{j(t)}(\eta) = 10^\delta \hat{\theta}_i^{(t)} + c_1\eta + \cdots + c_\tau\eta^\tau$

15 **foreach** $j \in \mathcal{N}_i$ **do**

16 Learner $i$ constructs $S_i^{j(t)}$ by (4) and sends it to $j$

17 **foreach** $i \in \mathcal{V}$ **do**

18 Learner $i$ receives $\{S_j^{i(t)}\}_{j\in\mathcal{N}_i}$ and constructs $s_i^{(t)}(0)$ by (6)

/* Consensus-based Aggregation */

19 **for** $m = 0$ **to** $M-1$ **do**

20 Learner $i$ sends $s_i^{(t)}(m)$ to every $j \in \mathcal{N}_i$ and receives $s_j^{(t)}(m)$

21 Learner $i$ updates $s_i^{(t)}(m+1) = a_{ii}s_i^{(t)}(m) + \sum_{j\in\mathcal{N}_i} a_{ij}s_j^{(t)}(m) \pmod p$

/* Recover global posterior */

22 Learner $i$ constructs $z_i^{(t)} = \lfloor N\, s_i^{(t)}(M) \rceil \bmod p$

23 Learner $i$ constructs $\tilde{\theta}_i^{(t)} = \begin{cases} z_i^{(t)}\, 10^{-\delta}, & 0 \le z_i^{(t)} \le \dfrac{p-1}{2}, \\ (z_i^{(t)} - p)\, 10^{-\delta}, & \dfrac{p+1}{2} \le z_i^{(t)} < p, \end{cases}$

24 Learner $i$ forms $\tilde{\mu}_i^{(t)} = \tilde{\theta}_i^{(t)}[0]/\tilde{\theta}_i^{(t)}[1], \quad (\tilde{\sigma}_i^{(t)})^2 = \big(\tilde{\theta}_i^{(t)}[1]\big)^{-1}$

25 **return** $\big(\tilde{\mu}_i^{(t)}, (\tilde{\sigma}_i^{(t)})^2\big)$

if and only if the weighted adjacency matrix $A = [a_{ij}]$ satisfies the conditions in Assumption 2.

*3) Privacy-Preserving Share Generation:* While the consensus protocol addresses decentralization, the direct exchange of state vectors introduces a critical privacy vulnerability. To secure this process, we integrate Shamir's Secret Sharing (SSS) with public-key encryption, a design that provides privacy preservation even in sparsely connected networks.

For the scheme to work, all cryptographic operations are performed over a large prime field $\mathbb{Z}_p$, where the prime $p$ is chosen large enough for all computations. Lines 3-21 implement the secure share generation and exchange process.

At the beginning of each round $t$, every learner publishes a public key $k_{pub|i}^{(t)}$ to its neighbours while keeping the corresponding private key $k_{priv|i}^{(t)}$ secret, which are used to encrypt and decrypt the exchanged shares. When a learner $i$ generates and distributes shares, its behavior is determined by one of two modes, depending on whether its number of neighbors exceeds a critical threshold $n_{nei}$.

Lines 3-12 realise the extended mode: when learner $i$ has fewer than $n_{\text{nei}}$ neighbours, it creates $|\mathcal{N}_j| + 1$ Shamir shares for every neighbour $j$ in line 5 so that the reconstruction threshold is not lowered by the small degree.

Learner $i$ constructs share $\hat{S}_i^{\ell(t)}$ for learner $\ell \in \mathcal{N}_j$ in line 7, and encrypts with learner $\ell$'s public key and relays through $j$ in line 8. After learner $\ell$ decrypts this share in line 9, it combines it with its own share $S_\ell^{j(t)}$ in lines 10-11. The combined sum denoted as $\hat{S}_\ell^{j(t)}$, is then transferred to learner $j$ in line 12.

Lines 13-16 handle the normal mode, when $|\mathcal{N}_i| > n_{\text{nei}}$. Line 14 selects the $\tau = |\mathcal{N}_i|$ random coefficients $c_1, \ldots, c_\tau \in$

$\mathbb{Z}_p$ with $c_\tau \neq 0$, and constructs the polynomial

$$\mathcal{H}_i^{j(t)}(\eta) = 10^\delta \hat{\theta}_i^{(t)} + c_1\eta + \cdots + c_\tau \eta^\tau. \tag{3}$$

The factor $10^\delta$ maps the real–valued secret $\hat{\theta}_i^{(t)}$ to an integer, and the prime $p$ is chosen large enough for all computations in $\mathbb{Z}_p$. This process generates $|\mathcal{N}_i|$ shares, and in this normal mode, all $|\mathcal{N}_i|$ shares are necessary for reconstruction.

Learner $i$ computes and transmits the value $S_i^{j(t)}$ to each neighbor $j$, computed as:

$$S_i^{j(t)} = \mathcal{H}_i^{j(t)} \cdot l_{ij} \bmod p. \tag{4}$$

$l_{ij}$ is the Lagrange basis polynomial defined as:

$$l_{ij} = \prod_{\substack{k\in\mathcal{N}_i \\ k\neq j}} \frac{-\eta_k}{\eta_j - \eta_k} \bmod p, \tag{5}$$

where we evaluate the polynomial at integer points $\eta_k = k$ (node indices).

*4) Secure Aggregation and Model Recovery:* Line 18 shows how every learner $i$ gathers the shares $\{S_j^{i(t)}\}_{j\in\mathcal{N}_i}$ and forms the masked integer state

$$s_i^{(t)}(0) = \sum_{j\in\mathcal{N}_i} S_j^{i(t)} \bmod p. \tag{6}$$

The consensus routine implemented in lines 19-21 is run for a fixed $M$ iterations. For $m = 0, 1, \ldots, M-1$, each learner applies the weighted update

$$s_i^{(t)}(m+1) = a_{ii} s_i^{(t)}(m) + \sum_{j\in\mathcal{N}_i} a_{ij} s_j^{(t)}(m), \tag{7}$$

broadcasting the new value to its neighbours after each step.

Under Assumption 2 the sequence converges geometrically; choosing $M$ so that $\rho(A - \frac{1}{N}\mathbf{1}\mathbf{1}^\mathsf{T})^M < \varepsilon$ guarantees

$$\|s_i^{(t)}(M) - \tfrac{1}{N}\sum_j s_j^{(t)}(0)\| < \varepsilon. \tag{8}$$

After the $M$ iterations each learner rescales and rounds in line 22, then maps the result back to a signed real vector in line 23, where the factor $(10^{-\delta})$ reverses the earlier scaling. This conversion ensures that $\tilde{\theta}_i^{(t)}$ both restores the correct signed real value and remains within the valid range.

Finally, each learner converts this two-element vector into the global mean and variance (line 24) and returns $\big((\tilde{\mu}_i^{(t)}, (\tilde{\sigma}_i^{(t)})^2)\big)$. With a sufficiently large $M$, the result provably converges to that of a centralized PoE aggregator, but without compromising decentralization or privacy.

### C. Overall Algorithm

Algorithm 3 shows the proposed algorithm. Each learning round $t$ comprises the following two phases:

Local Learning (lines 3-5): Each learner applies Algorithm 1 to train a robust model on its local data stream.

Secure Aggregation (lines 6-7): All learners execute Algorithm 2 to securely aggregate their local models into a new global model, a process that does not require a central server.

By iterating this process for $T$ rounds, the global model continuously adapts to the decentralized data streams.

**Algorithm 3:** Overall Algorithm

1 The learners agree on a positive prime number $p$ and positive integers $\nu_i$, $\sigma_s$, $T$, $M$, weighted matrix $A = [a_{ij}] \in \mathbb{R}^{N\times N}$, key pairs $k_{pub|i}^{(t)}$ and $k_{priv|i}^{(t)}$ for $i \in \mathcal{V}$, $t \in \{1, \ldots, T\}$
2 **for** $t = 1$ **to** $T$ **do**
  `/* Local Learning Phase */`
3  **foreach** $i \in \mathcal{V}$ **do**
4  Collect local data: $D_i^{(t)} = (X_i^{(t)}, y_i^{(t)})$
5  Compute local posterior: $(\hat{\mu}_i^{(t)}, (\hat{\sigma}_i^{(t)})^2) = \text{Alg}_1(X^*, D_i^{(t)}, \nu_i, \sigma_s)$
  `/* Secure Aggregation Phase */`
6  $\{(\tilde{\mu}_i^{(t)}, (\tilde{\sigma}_i^{(t)})^2)\}_{i\in\mathcal{V}} =$
7  $\text{Alg}_2(\{\hat{\mu}_i^{(t)}, (\hat{\sigma}_i^{(t)})^2\}_{i\in\mathcal{V}}, A, M, \{k_{\text{pub}|i}^{(t)}, k_{\text{priv}|i}^{(t)}\}_{i\in\mathcal{V}}, p)$
8 **return** $\{(\tilde{\mu}_i^{(T)}, (\tilde{\sigma}_i^{(T)})^2)\}_{i\in\mathcal{V}}$

## VI. Theoretical Analysis

This section analyzes the algorithm's robustness to outliers, aggregation correctness, privacy guarantees, and computational overhead.

### A. Outlier Proneness

To validate the robustness of our algorithm, we analyze its robustness to outliers. Specifically, we prove that the desirable outlier-rejection property of the Student-t likelihood is maintained even after local models are aggregated via the PoE method. This analysis extends the single-model result of [13] to a multi-model aggregation setting.

*Definition 1 ( [13, p. 361]):* Let $p(f \mid y_1, \ldots, y_n)$ be a posterior distribution function for latent variable $f$ given observations $y_1, \ldots, y_n$. We say $p$ is outlier-prone of order $n$ if $p\big(f \mid y_1, \ldots, y_n, y_{n+1}\big) \longrightarrow p\big(f \mid y_1, \ldots, y_n\big)$ as $y_{n+1} \to \infty$, where $y_{n+1} \to \infty$ in the usual real number sense, and the convergence is in total variation.

*Theorem 1:* The distribution of $N$ Student-t distributions after PoE is outlier-prone of order 1.

*Remark 3:* Theorem 1 is established under the assumption of exact Student-t distributions. However, in our implementation, Student-t regression is approximated using Algorithm 1, which provides a PoE-based approximation for the aggregation of $N$ Student-t distributions. This approximation allows for efficient inference while maintaining the key characteristics of the underlying distributions.

*Proof 1:*

*Lemma 2 ( [13, p. 362]):* A distribution function $g(\cdot)$ is outlier-prone of order 1 if it satisfies the following conditions:

(i) $g(\cdot)$ is symmetric.
(ii) Given any $\varepsilon > 0$, $h > 0$, there exists $A$ such that if $y > A$ then $|g(y') - g(y)| < \varepsilon g(y)$ whenever $|y' - y| < h$.
(iii) The function $g(y)$ satisfies all of the following:
  (a) Continuity and positivity: $g(y)$ is continuous and positive for all $y \in \mathbb{R}$.
  (b) There exists a $B$ such that, for all $y > B$,

$g(y)$ is decreasing in $y$,
$b(y) = d\log g(y)/dy$ exists and is increasing in $y$.

(c) There exists a $C \leq B$ (as defined in condition (b)) such that, for all $y < C, g(y)$ is increasing in $y$.

*Lemma 3 ( [13, p. 362]):* The Student-t distribution satisfies all conditions in Lemma 2. Hence, a Student-t distribution-based observation model is outlier-prone of order 1.

In our implementation of the PoE algorithm, each individual regression model takes a multi-dimensional input $x \in \mathbb{R}^n$ and produces a posterior distribution $p_i(y \mid x)$. The combination is performed directly on the means and variances of these posterior distributions to obtain the final aggregated model. Equivalently, this can be viewed as computing a new probability density function by multiplying the individual posterior distributions and normalizing the result $p(y \mid x) \propto \prod_{i=1}^{N} p_i(y \mid x)$ [39]. Thus the Student-t probability density function combination with PoE shows as $\mathcal{T}_{\text{combined}}(y \mid x) = Z \prod_{i=1}^{N} \mathcal{T}_i(y \mid x)$, where $Z = \left( \int \prod_{i=1}^{N} \mathcal{T}_i(y \mid x)\, \mathrm{d}y \right)^{-1}$ is a positive normalization constant. In PoE algorithm, each model $i$ has its own predicted function value $f_i(x)$ at input $x$, and the corresponding Student-t likelihood is centered at this prediction $\mathcal{T}_i(y \mid x) \propto \{\nu\sigma_s^2 + (y - f_i(x))^2\}^{-\frac{\nu+1}{2}}$.

For the outlier-proneness analysis, we consider the residual for each model $i$ as $r_i = y - f_i(x)$. The true combined probability distribution is the product of the individual likelihoods, each depending on its own residual $\mathcal{T}_i(y \mid x) \propto \{\nu\sigma_s^2 + (y - f_i(x))^2\}^{-\frac{\nu+1}{2}}$. To build upon the theory for single distributions, we align our multi-model problem with the original analytical setting of O'Hagan. We achieve this by analyzing the aggregated system's behavior in terms of a common residual, $r = y - f(x)$, where $f(x)$ is the consensus prediction. With a slight abuse of notation, we express our combined distribution in terms of this common residual $r$ by substituting $r$ for each $r_i$:

$$\mathcal{T}_{\text{combined}}(r) = Z \prod_{i=1}^{N} \mathcal{T}_i(r). \tag{9}$$

Proof of (i): By Lemma 2 and 3, each Student-t distribution $\mathcal{T}_i(r)$, $i = 1, \ldots, N$ is symmetric about 0 for the residual, meaning $\mathcal{T}_i(r) = \mathcal{T}_i(-r)$ for all $i$. By (9), replacing $r$ with $-r$ gives $\mathcal{T}_{\text{combined}}(-r) = Z \prod_{i=1}^{N} \mathcal{T}_i(-r)$. Since each $\mathcal{T}_i(r)$ satisfies $\mathcal{T}_i(-r) = \mathcal{T}_i(r)$, it follows that $Z \prod_{i=1}^{N} \mathcal{T}_i(-r) = Z \prod_{i=1}^{N} \mathcal{T}_i(r)$.Thus, $\mathcal{T}_{\text{combined}}(-r) = \mathcal{T}_{\text{combined}}(r)$.

Proof of (ii): By Lemma 2 and 3, the Student-t distribution satisfies condition Lemma 2 (ii), that is for any given $\tilde{\varepsilon} > 0$ and $h > 0$ there exists a constant $A_i > 0$ such that for all $r > A_i$ and $|r' - r| < h$ holds $|\mathcal{T}_i(r') - \mathcal{T}_i(r)| < \tilde{\varepsilon}\, \mathcal{T}_i(r)$. To ensure all $\mathcal{T}_i$ satisfy the same bound, set $A_{max} = \max\{A_1, A_2, \ldots, A_N\}$. Then for $r > A_{max}$ and $|r' - r| < h$, $\left|\mathcal{T}_{\text{combined}}(r') - \mathcal{T}_{\text{combined}}(r)\right| = Z \left| \prod_{i=1}^{N} \mathcal{T}_i(r') - \prod_{i=1}^{N} \mathcal{T}_i(r) \right|$.

We now apply a telescoping expansion:

$$\begin{aligned} &\prod_{i=1}^{N} \mathcal{T}_i(r') - \prod_{i=1}^{N} \mathcal{T}_i(r) \\ &= \sum_{j=1}^{N} \left( \prod_{k=1}^{j-1} \mathcal{T}_k(r') \right) \left[ \mathcal{T}_j(r') - \mathcal{T}_j(r) \right] \left( \prod_{k=j+1}^{N} \mathcal{T}_k(r) \right). \end{aligned} \tag{10}$$

Taking absolute values and by the triangle inequality:

$$\begin{aligned} &\left| \prod_{i=1}^{N} \mathcal{T}_i(r') - \prod_{i=1}^{N} \mathcal{T}_i(r) \right| \\ &\leq \sum_{j=1}^{N} \left( \prod_{k=1}^{j-1} \mathcal{T}_k(r') \right) \tilde{\varepsilon}\, \mathcal{T}_j(r) \left( \prod_{k=j+1}^{N} \mathcal{T}_k(r) \right). \end{aligned} \tag{11}$$

Since $\mathcal{T}_k(r) > 0$ and $\mathcal{T}_k(r') \leq (1 + \tilde{\varepsilon})\, \mathcal{T}_k(r)$ for each $k$, we have:

$$\prod_{k=1}^{j-1} \mathcal{T}_k(r') \leq (1 + \tilde{\varepsilon})^{\,j-1} \prod_{k=1}^{j-1} \mathcal{T}_k(r). \tag{12}$$

By (11) and (12),

$$\begin{aligned} \left| \prod_{i=1}^{N} \mathcal{T}_i(r') - \prod_{i=1}^{N} \mathcal{T}_i(r) \right| &\leq \tilde{\varepsilon} \sum_{j=1}^{N} (1 + \tilde{\varepsilon})^{\,j-1} \prod_{i=1}^{N} \mathcal{T}_i(r) \\ &= \tilde{\varepsilon} \sum_{j=0}^{N-1} (1 + \tilde{\varepsilon})^{j} \prod_{i=1}^{N} \mathcal{T}_i(r) \\ &= \left[ (1 + \tilde{\varepsilon})^N - 1 \right] \prod_{i=1}^{N} \mathcal{T}_i(r). \end{aligned} \tag{13}$$

Multiplying by the positive constant $Z$, we have:

$$\begin{aligned} \left| \mathcal{T}_{\text{combined}}(r') - \mathcal{T}_{\text{combined}}(r) \right| &\leq Z \left[ (1 + \tilde{\varepsilon})^N - 1 \right] \prod_{i=1}^{N} \mathcal{T}_i(r) \\ &= \left[ (1 + \tilde{\varepsilon})^N - 1 \right] \mathcal{T}_{\text{combined}}(r). \end{aligned} \tag{14}$$

Since $(1 + \tilde{\varepsilon})^N - 1$ is a continuous and strictly increasing function of $\tilde{\varepsilon}$, for any given $\varepsilon > 0$ one can choose $\tilde{\varepsilon} > 0$ such that $(1 + \tilde{\varepsilon})^N - 1 \leq \varepsilon$. This implies $\left| \mathcal{T}_{\text{combined}}(r') - \mathcal{T}_{\text{combined}}(r) \right| < \varepsilon\, \mathcal{T}_{\text{combined}}(r)$.

Proof of (iii): (iii) a) Since each participating distribution function $\mathcal{T}_i(r)$ is continuous and positive for $r \in \mathbb{R}$, and $Z > 0$, it is easy to obtain that the combined distribution $\mathcal{T}_{\text{combined}}(r)$ is continuous and positive for all $r \in \mathbb{R}$.

(iii) b) By equation (9), the combined logarithmic derivative with respect to the residual is shown as:

$$b(r) = \frac{d \log(\mathcal{T}_{\text{combined}}(r))}{dr} = \frac{d}{dr} \left( \log Z + \sum_{i=1}^{N} \log(\mathcal{T}_i(r)) \right). \tag{15}$$

The observation model for each learner follows the Student-t distribution in terms of residuals:

$$\mathcal{T}(r) \propto \{\nu\sigma_s^2 + r^2\}^{-\frac{\nu+1}{2}} = W\{\nu\sigma_s^2 + r^2\}^{-\frac{\nu+1}{2}}, \tag{16}$$

where $W > 0$ is some positive constant. By (16), we have:

$$\log(\mathcal{T}_i(r)) = \log W_i - \frac{1}{2}(\nu+1)\log(\nu\sigma_s^2 + r^2). \tag{17}$$

Combining (15) and (17), we have:

$$b(r) = -N\frac{(\nu+1)r}{\nu\sigma_s^2 + r^2}. \tag{18}$$

From (18), since $\frac{N(\nu+1)r}{\nu\sigma_s^2+r^2}$ has the same sign as $r$ and tends to 0 as $|r| \to \infty$, it follows that $b(r)$ remains negative for large positive residuals and positive for large negative residuals. Hence, $\log \mathcal{T}_{\text{combined}}(r)$ has a negative slope for large positive residuals, so $\mathcal{T}_{\text{combined}}(r)$ is eventually decreasing in the positive direction.

Next, to see that $b(r)$ is increasing beyond a certain point, we compute its derivative: $\frac{d\,b(r)}{dr} = -N(\nu+1)\frac{\nu\sigma_s^2 - r^2}{\left(\nu\sigma_s^2+r^2\right)^2}$. When $|r| > \sqrt{\nu\sigma_s^2}$, we have $\nu\sigma_s^2 - r^2 < 0$, such that the product is positive for positive $r$. Thus, $\frac{d\,b(r)}{dr} > 0$ for $r > B$, where we define $B = \sqrt{\nu\sigma_s^2}$. This confirms that $b(r)$ is strictly increasing in that region, satisfying both requirements in b).

(iii) c) By Lemma 2, each $\mathcal{T}_i(r)$ has a critical point $C_i$ at which it transitions from increasing to decreasing. Since all distributions share the same residual $r$, let $C_{\min} = \min\{C_1, \ldots, C_N\}$. For $r < C_{\min}$, Lemma 2 implies $\frac{d}{dr}\log \mathcal{T}_i(r) > 0$ for each $i$, such that $\sum_{i=1}^{N} \frac{d}{dr}\log\big(\mathcal{T}_i(r)\big) > 0$. Hence, $\mathcal{T}_{\text{combined}}(r)$ is strictly increasing on $\big(-\infty, C_{\min}\big)$. This completes the proof.

### B. Correctness

This section proves the correctness of Algorithm 3, showing that each learner $i \in \mathcal{V}$ converges to the global aggregate model. Our approach extends the work of [29] by demonstrating that convergence is maintained even with the addition of public-key encryption and a two-hop secret sharing mechanism. The proof structure is adapted from [29].

*Theorem 2:* Assuming the weighted adjacency $A$ satisfies Assumption 2. With sufficiently large $p$ and $M$ such that

$$p > \max\left\{N, 1 + 2\times 10^{\delta} N \max_{t,i}\left|\hat{\theta}_i^{(t)}\right|\right\} \tag{19}$$

$$\max_t 2p\sqrt{N}\|N(A^M - \frac{1}{N}1_N 1_N^T)\| < 1 \tag{20}$$

where $\|\cdot\|$ the $l_2$ norm of a matrix, then for $\forall i \in \mathcal{V}, \forall t \in \{1,\ldots,T\}$, $\tilde{\theta}_i^{(t)} = \sum_{j\in\mathcal{V}} \hat{\theta}_j^{(t)}(0)$.

*Proof 2:* At round $t$, each learner $i$ applies SSS to its scaled local model $10^{\delta}\hat{\theta}_i^{(t)}(0)$, generating shares using polynomial $\mathcal{H}_i^{j(t)}(\eta) = 10^{\delta}\hat{\theta}_i^{(t)}(0) + c_1\eta + \cdots + c_\tau\eta^\tau$ such that

$$\sum_{j\in\mathcal{N}_i} S_i^{j(t)} \equiv 10^{\delta}\hat{\theta}_i^{(t)}(0) \pmod p, \tag{21}$$

where the recipients are all nodes receiving shares from learner $i$. The algorithm operates in two modes, but both preserve this fundamental property. In normal mode, learner $i$ distributes shares directly to its one-hop neighbors $\mathcal{N}_i$, and each learner $k$ forms its initial consensus state as $s_k^{(t)}(0) = \sum_{i\in\mathcal{N}_k} S_i^{k(t)} \bmod p$. In extended mode, the algorithm employs encrypted two-hop forwarding to protect sparsely-connected learners. Consider a share $\hat{S}_i^{l(t)}$ intended for two-hop neighbor $l$, relayed through intermediary $j$. The encryption and decryption operations are algebraically transparent in $\mathbb{Z}_p$:

$$\begin{aligned}\hat{S}_i^{l(t)} &= \text{Decrypt}(\text{Encrypt}(S_i^{l(t)}, k_{pub|l}^{(t)}), k_{priv|l}^{(t)}) \\ &\equiv S_i^{l(t)} \pmod p.\end{aligned} \tag{22}$$

When learner $l$ combines this with its own share for $j$, the composite share becomes $\hat{S}_l^{j(t)} = S_l^{j(t)} + \hat{S}_i^{l(t)} \bmod p$, which preserves the total contribution in $j$'s initial state $s_j^{(t)}(0)$.

Regardless of the operational mode, the sum of all generated shares equals the sum of all initial consensus states. By exchanging the order of summation across the network:

$$\begin{aligned}\sum_{k\in\mathcal{V}} s_k^{(t)}(0) &= \sum_{k\in\mathcal{V}}\sum_{i\in\mathcal{N}_k} S_i^{k(t)} \\ &= \sum_{i\in\mathcal{V}} \begin{cases} \sum_{k\in\mathcal{N}_i} S_i^{k(t)}, & \text{normal mode} \\ \sum_{k\in\mathcal{N}_i\cup(\mathcal{N}_i^*\cap\mathcal{B})} S_i^{k(t)}, & \text{extended mode} \end{cases} \\ &\equiv \sum_{i\in\mathcal{V}} 10^{\delta}\hat{\theta}_i^{(t)}(0) \pmod p.\end{aligned} \tag{23}$$

This summation invariant ensures that the distributed initial states collectively encode the exact global sum.

The consensus iterations proceed according to the update rule $s^{(t)}(m+1) = As^{(t)}(m)$, where all operations are performed modulo $p$. After $M$ iterations, we have $s^{(t)}(M) = A^M s^{(t)}(0) \bmod p$. Under the spectral properties of consensus matrix $A$ specified in the algorithm assumptions, the process converges to the average: $\lim_{M\to\infty} A^M = \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$. For finite $M$, the effect of rounding errors must be bounded to guarantee correct recovery. Let $(A^M)_i$ denote the $i$-th row of $A^M$. The absolute error between the scaled consensus state and the true integer sum is bounded as:

$$\begin{aligned}|Ns_i^{(t)}(M) - \mathbf{1}_N^T s^{(t)}(0)| &= |N(A^M)_i s^{(t)}(0) - \mathbf{1}_N^T s^{(t)}(0)| \\ &\le \|N(A^M)_i - \mathbf{1}_N^T\| \cdot \|s^{(t)}(0)\| \\ &\le \|N(A^M - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\| \cdot \|s^{(t)}(0)\|.\end{aligned} \tag{24}$$

Since each component $s_i^{(t)}(0) \in [0, p)$, we have $\|s^{(t)}(0)\| < p\sqrt{N}$. Combined with the algorithm's parameter condition:

$$\max_t 2p\sqrt{N}\|N(A^M - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T)\| < 1, \tag{25}$$

this yields $|Ns_i^{(t)}(M) - \mathbf{1}_N^T s^{(t)}(0)| < 0.5$. Since $\mathbf{1}_N^T s^{(t)}(0)$ is an integer and the error is strictly less than 0.5, the rounding operation achieves perfect recovery: $\lfloor Ns_i^{(t)}(M)\rceil = \mathbf{1}_N^T s^{(t)}(0) = \sum_{j\in\mathcal{V}} s_j^{(t)}(0)$, where $\lfloor\cdot\rceil$ stands for rounding to the nearest integer.

In reconstruction steps, each learner computes $z_i^{(t)} = \lfloor Ns_i^{(t)}(M)\rceil \bmod p$, which by analysis satisfies:

$$z_i^{(t)} \equiv \sum_{j\in\mathcal{V}} 10^{\delta}\hat{\theta}_j^{(t)}(0) \pmod p. \tag{26}$$

The modular operation potentially obscures the sign of the original sum. However, the algorithm's parameter condition

ensures $\left|10^{\delta}\sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}(0)\right| < \frac{p-1}{2}$. This constraint enables unambiguous sign recovery. Let $10^{\delta}\sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}(0)$ be the true scaled sum. The algorithm's reconstruction correctly handles both cases:

$$z_i^{(t)} = \begin{cases} 10^{\delta}\sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}, & \text{if } 0 \le 10^{\delta}\sum_j \hat{\theta}_j^{(t)} < \frac{p-1}{2}, \\ p + 10^{\delta}\sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}, & \text{if } -\frac{p-1}{2} < 10^{\delta}\sum_j \hat{\theta}_j^{(t)} < 0. \end{cases} \tag{27}$$

The recovery operation in the algorithm checks whether $z_i^{(t)}$ lies in the lower half $[0, \frac{p-1}{2}]$ or upper half $[\frac{p+1}{2}, p)$ of the modular range. For the lower half, the algorithm applies $\tilde{\theta}_i^{(t)} = z_i^{(t)} \cdot 10^{-\delta}$; for the upper half, it applies $\tilde{\theta}_i^{(t)} = (z_i^{(t)} - p) \cdot 10^{-\delta}$. In both cases, this correctly recovers:

$$\tilde{\theta}_i^{(t)} = \frac{10^{\delta}\sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}(0)}{10^{\delta}} = \sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}(0). \tag{28}$$

Therefore, every learner $i \in \mathcal{V}$ successfully computes the identical and correct global aggregate model $\tilde{\theta}_i^{(t)} = \sum_{j\in\mathcal{V}}\hat{\theta}_j^{(t)}(0)$ for all rounds $t \in \{1, \ldots, T\}$. This completes the proof.

### C. Privacy Preservation

This section analyzes the privacy guarantees of Algorithm 2, focusing on how the secure aggregation mechanism protects each learner's model during the consensus process.

Let $\mathcal{V}$ denote the set of learners, partitioned into disjoint subsets: benign learners $\mathcal{B}$ and adversarial learners $\mathcal{A}$, such that $\mathcal{V} = \mathcal{B} \cup \mathcal{A}$ and $\mathcal{B} \cap \mathcal{A} = \varnothing$. We consider a semi-honest adversarial model where adversaries follow the protocol but attempt to infer private information from observed messages. At round $t$, each benign learner $i \in \mathcal{B}$ possesses a secret local model $\hat{\theta}_i^{(t)}(0) \in \mathbb{Z}_p$.

In the context of our algorithm, a learner's local model $\hat{\theta}_i^{(t)}(0)$ is considered private if adversaries cannot distinguish it from a random value, despite observing all protocol messages they are entitled to receive.

**Algorithm Analysis.** Algorithm 2 employs a two-phase privacy protection strategy. In the share generation phase (lines 3-18), each learner applies SSS to encode its local model into polynomial shares. In the consensus phase (lines 19-21), learners perform consensus on these encoded shares while preserving the underlying secrets.

The algorithm operates in two modes based on connectivity:

- *Normal mode* ($|\mathcal{N}_i| > n_{nei}$): Direct share distribution to one-hop neighbors
- *Extended mode* ($|\mathcal{N}_i| \le n_{nei}$): Encrypted share forwarding through two-hop paths

The privacy properties of each mode are analyzed below.

**Normal Mode Analysis.** In normal mode, each learner $i$ applies SSS to protect its local model. Specifically, learner $i$ constructs a degree-$|\mathcal{N}_i|$ polynomial $\mathcal{H}_i^{j(t)}(\eta) = 10^{\delta}\hat{\theta}_i^{(t)} + d_1\eta + \cdots + d_{\tau'}\eta^{\tau'}$, where coefficients $c_1, \ldots, c_{|\mathcal{N}_i|}$ are chosen uniformly at random from $\mathbb{F}_p$, and distributes shares $S_{ji}^{(t)} = \mathcal{H}_i^{j(t)}$ to neighbors $j \in \mathcal{N}_i$. Our analysis relies on the fundamental security property of SSS.

*Lemma 4 (Information-theoretic security of SSS [42]):* For any secret sharing scheme based on a degree-$d$ polynomial over $\mathbb{F}_p$, an adversary requires exactly $d+1$ shares to reconstruct the secret. Possession of $d$ or fewer shares provides no information about the secret.

*Proof 3:* This follows directly from the information-theoretic security of SSS scheme. With fewer than $d+1$ shares, polynomial interpolation is underdetermined, and all possible secret values remain equally probable from the adversary's perspective [42].

Applying this property to the normal mode yields the following privacy guarantees.

*Lemma 5 (Adversarial view in normal mode):* Assume every benign learner has at least one benign neighbor, i.e., $\mathcal{N}_i \cap \mathcal{B} \neq \varnothing$ for all $i \in \mathcal{B}$. In round $t$, the adversary $\mathcal{A}$ can recover only those linear combinations $\sum_{i\in\mathcal{C}}\hat{\theta}_i^{(t)}(0)$, where $\mathcal{C} \subseteq \mathcal{B}$ is a maximal connected component of benign learners such that every communication path from $\mathcal{C}$ to any benign node outside $\mathcal{C}$ is intercepted by at least one adversarial node. No other linear combination of the benign secrets is information-theoretically recoverable.

*Proof 4:* In normal mode, learner $i$ generates a degree-$|\mathcal{N}_i|$ polynomial and distributes $|\mathcal{N}_i|$ shares to its neighbors. Let $\mathbf{s}^{(t)}(0) = (s_i^{(t)}(0))_{i\in\mathcal{V}}$ denote the initial consensus state defined in Algorithm 2, where $s_i^{(t)}(0) = \sum_{j\in\mathcal{N}_i} S_{ji}^{(t)} \bmod p$.

For the benign subset $\mathcal{C}$ satisfying the isolation condition, the sum of secrets within $\mathcal{C}$ equals the total information flowing out of this component: $\sum_{i\in\mathcal{C}}\hat{\theta}_i^{(t)}(0) \equiv \sum_{k\in\mathcal{A}}\mathbf{1}^{\top}\mathbf{s}_{\mathcal{C}\to k}^{(t)} \bmod p$, where $\mathbf{s}_{\mathcal{C}\to k}^{(t)}$ represents shares transmitted from component $\mathcal{C}$ to adversarial node $k$. Since all boundary communications are visible to $\mathcal{A}$, this linear combination is recoverable.

Conversely, consider any proper subset $\mathcal{D} \subset \mathcal{C}$. Since each learner in $\mathcal{D}$ has at least one benign neighbor, at least one share generated by nodes in $\mathcal{D}$ remains within the benign region. By Lemma 4, adversaries possess fewer than the required $|\mathcal{N}_i|+1$ reconstruction threshold for any individual secret in $\mathcal{D}$. Therefore, no additional information about individual secrets in $\mathcal{D}$ can be extracted beyond the boundary sum.

**Extended Mode Analysis.** Extended mode addresses the privacy vulnerability of sparsely connected learners through encrypted two-hop forwarding. When $|\mathcal{N}_i| \le n_{nei}$, learner $i$ constructs a degree-$(|\mathcal{N}_j|+1)$ polynomial for each neighbor $j$ and employs public-key encryption to securely forward shares to benign two-hop neighbors. This hybrid approach combines the information-theoretic security of SSS with the computational security of encryption. The adversarial view under this enhanced protection is characterized as follows.

*Lemma 6 (Information leakage in extended mode):* Let $i \in \mathcal{B}$ satisfy $|\mathcal{N}_i| \le n_{nei}$ and assume $\mathcal{N}_i^* \cap \mathcal{B} \neq \varnothing$. In round $t$, the maximum information leaked to adversaries is one linear combination $\hat{\theta}_i^{(t)}(0) + \hat{\theta}_j^{(t)}(0) \bmod p, \quad j \in \mathcal{N}_i^* \cap \mathcal{B}$, while individual secrets remain information-theoretically indistinguishable.

*Proof 5:* Let $q = |\mathcal{N}_i \cap \mathcal{A}|$ denote the number of adversarial one-hop neighbors of learner $i$. In extended mode, learner $i$ creates $|\mathcal{N}_i|+1$ independent Shamir shares with reconstruction threshold $q+1$, specifically designed to exceed the number of adversarial neighbors.

The share distribution follows a two-path strategy: (1) $q$

shares are distributed directly to adversarial one-hop neighbors, and (2) one additional share is forwarded along the encrypted two-hop path $i \to k \to j$ where $k \in \mathcal{A}$ and $j \in \mathcal{N}_i^* \cap \mathcal{B}$.

Consequently, adversaries observe exactly $q+1$ shares during the transmission phase: the $q$ direct shares plus the encrypted forwarded share. However, these $q+1$ shares correspond to the polynomial evaluation $H_i^{(t)}(\eta) + H_j^{(t)}(\eta)$, which encodes the linear combination $\hat{\theta}_i^{(t)}(0) + \hat{\theta}_j^{(t)}(0)$ as its constant term. Crucially, the final share remains exclusively with the benign recipient $j$, ensuring that adversaries cannot separate the individual secrets by Lemma 4.

*Lemma 7 (Semantic security of encrypted forwarding [43]):* Under the IND-CPA assumption, the encrypted shares in extended mode are semantically secure. Any probabilistic polynomial-time adversary's advantage in extracting plaintext information from encrypted shares is bounded by $\mathrm{negl}(\lambda)$ where $\lambda$ is the security parameter and $\mathrm{negl}(\cdot)$ is a negligible function. This follows directly from the semantic security of IND-CPA secure encryption schemes.

Combining the analysis of both modes yields our main privacy preservation result.

*Theorem 3 (Privacy preservation guarantee):* Algorithm 2 ensures privacy preservation for every benign learner $i$ in every round $t$:

1) **Perfect secrecy:**[1] If $|\mathcal{N}_i| > n_{nei}$ and $\mathcal{N}_i \cap \mathcal{B} \neq \varnothing$, then $\hat{\theta}_i^{(t)}(0)$ achieves perfect secrecy against unbounded adversaries.
2) **Computational secrecy:**[2] If $|\mathcal{N}_i| \leq n_{nei}$ and $\mathcal{N}_i^* \cap \mathcal{B} \neq \varnothing$, then $\hat{\theta}_i^{(t)}(0)$ achieves computational secrecy with adversarial advantage bounded by $\mathrm{negl}(\lambda)$.

*Proof 6:* **Case 1 (Normal mode):** By Lemma 5, adversaries can only recover linear combinations corresponding to isolated benign components. For any individual learner $i$ with at least one benign neighbor, no information about $\hat{\theta}_i^{(t)}(0)$ is leaked, ensuring perfect secrecy.

**Case 2 (Extended mode):** By Lemma 6, adversaries observe at most one linear combination during transmission. By Lemma 7, the probability of successfully decrypting the forwarded share is negligible. Therefore, adversaries effectively obtain fewer than the $q+1$ shares required for reconstruction by Lemma 4, ensuring computational security with advantage bounded by $\mathrm{negl}(\lambda)$.

Thus, Algorithm 2's dual-mode approach provides provable privacy guarantees across varying network topologies, with the security level gracefully degrading from perfect to computational as connectivity decreases.

### D. Overhead Analysis

This subsection analyzes the overhead of our algorithm, focusing on two key aspects: local model training and the privacy-preserving aggregation protocol.

*1) Local Training Overhead:* Training the GPR model with Student-t likelihood involves computing and inverting the kernel matrix, leading to a complexity of $O(n^3)$ per iteration, where $n$ is the number of training samples at each node. For each learner $i \in \mathcal{V}$, the computational complexity of nearest-neighbor search scales as $\mathcal{O}(n \cdot n_{\text{test}})$ for a linear search. By training on the nearest-neighbor subset, we reduce the kernel computation complexity to $\mathcal{O}(Q^3)$, where $Q \ll n$ is the number of neighbors used in the local model. This approach significantly lowers the computational cost compared to traditional FL methods requiring full model training with complexity $\mathcal{O}(n^3)$.

*2) Privacy and Communication Overhead:* For privacy computation, generating SSS shares has a complexity of $\mathcal{O}(|\mathcal{N}_i|)$ per learner, while adversarial reconstruction has a complexity of $\mathcal{O}(|\mathcal{N}_i|^2)$. The extended mode adds IND-CPA secure encryption with complexity $\mathcal{O}(\lambda^c)$, where $c$ depends on the scheme (e.g., $c=1$ for AES [45], $c=3$ for RSA [46]). Decryption has a similar complexity.

For communication, the overhead in normal mode is $\mathcal{O}(|\mathcal{N}_i| \log p)$ bits per learner. The extended mode requires additional encrypted communication of size $\mathcal{O}(\lambda)$ bits along each two-hop path.

This dual-mode approach balances security and performance. Compared to fully homomorphic encryption with complexity $\mathcal{O}(n^k)$ for large $k$, our algorithm provides strong security with significant efficiency gains.

## VII. Experiment

This section evaluates the algorithm's robustness, convergence, privacy and overhead. On the hardware side, the simulation is performed on a Lenovo ThinkPad laptop with Intel® Core™ i7-1360P CPU at 2200 MHz. On the software side, the simulation is performed on MATLAB R2021b.

### A. Comparative Regression With Outliers

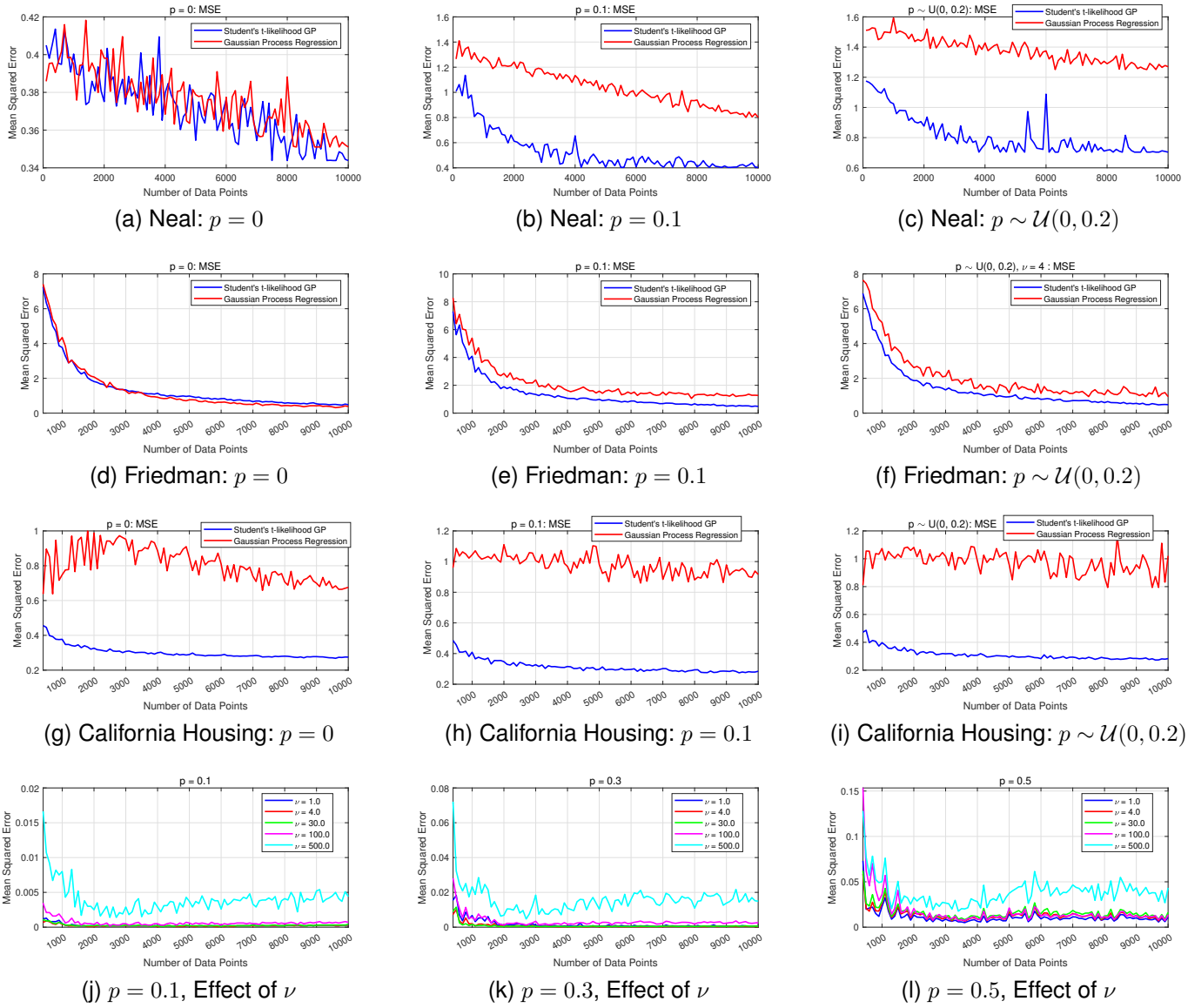
This section benchmarks our proposed Student-t likelihood GPR against the standard GPR on several datasets with varying outlier contamination, as shown in Fig. 2.

*1) Data Generation and Benchmarks:* We evaluate on two synthetic functions and one real-world dataset. First, the Neal data [47] is a one-dimensional function with complex nonlinearities, defined as: $f_{\text{Neal}}(x) = 0.3 + 0.4x + 0.5\sin(2.7x) + \frac{1.1}{1+x^2}$. Inputs are sampled from $x \sim \mathcal{U}[-3,3]$ with noise $\epsilon \sim \mathcal{N}(0, 0.1^2)$. Second, the Friedman function [48] is a 10-dimensional multivariate benchmark designed to test performance with irrelevant features: $f_{\text{Friedman}}(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$. Inputs $x_d \sim \mathcal{U}[0,1]$ with noise $\epsilon \sim \mathcal{N}(0, 1^2)$. While only $x_1, ..., x_5$ are relevant to the function's output, $x_6, ..., x_{10}$ are irrelevant features. Finally, the California Housing Dataset [49] serves as the real-world benchmark, comprising 20,640 samples with 8 features. It contains inherent noise and natural data irregularities.

To simulate data corruption, outliers were artificially injected by perturbing the target values of randomly selected data points (with probability $p$) by a large offset, $\delta_{\text{train}} \in \{-3, +3\}$.

[1]Perfect secrecy ensures that observed messages provide zero information about the secret, making it indistinguishable from a random value even to an adversary with unbounded computational power [44].

[2]Computational secrecy guarantees that no computationally bounded (polynomial-time) adversary can distinguish the secret from a random value with more than a negligible advantage [44].




(a) Neal: $p = 0$ (b) Neal: $p = 0.1$ (c) Neal: $p \sim \mathcal{U}(0, 0.2)$

(d) Friedman: $p = 0$ (e) Friedman: $p = 0.1$ (f) Friedman: $p \sim \mathcal{U}(0, 0.2)$

(g) California Housing: $p = 0$ (h) California Housing: $p = 0.1$ (i) California Housing: $p \sim \mathcal{U}(0, 0.2)$

(j) $p = 0.1$, Effect of $\nu$ (k) $p = 0.3$, Effect of $\nu$ (l) $p = 0.5$, Effect of $\nu$

Fig. 2. Performance comparison between Student-t GPR and standard GPR under different outlier contamination scenarios.

*2) Performance Comparison on Different Datasets:* Each node trains two models: the proposed Student-t GPR (RBF kernel, $\nu = 4$) and a standard GPR with Gaussian likelihood. For every incoming mini-batch we keep a $Q$=100 nearest-neighbour cache to speed up prediction, fuse the local PoE, and report the MSE.

On the Neal function (Fig. 2a,2b,2c) with pristine data ($p = 0$), both models show comparable performance, with our proposed method showing a small edge of **1.2%** in MSE. With 10% outliers ($p = 0.1$), the Student-t GPR reduces the MSE by **41.2%** compared to the standard GPR. This advantage grows further to **49.4%** as the contamination level increases to 20%. On Friedman dataset (Fig. 2d,2e,2f), when $p = 0$, the standard GPR holds a **12.1%** advantage. However, in contaminated scenarios ($p \in \{0.1, 0.2\}$), the heavy-tailed Student-t model shows better performance, reducing the MSE by an average of **30-36%**. On the California Housing dataset (Fig. 2g,2h,2i), the model's robustness is particularly evident. Even on the original, unmodified data ($p = 0$), our model achieves **61.4%** MSE reduction. This suggests the dataset contains inherent irregularities or heavy-tailed noise that the Student-t likelihood is better suited to handle. This performance gap is maintained as artificial outliers are added, with the MSE reduction reaching **67.0%** and **67.1%** for 10% and 20% contamination, respectively.

In summary, these results validate that the Student-t GPR consistently achieves a **40-70%** reduction in prediction error on datasets with outliers. Under ideal, clean conditions, its performance remains consistent with GPR.

*3) Ablation Study on Hyperparameter $\nu$:* To investigate the hypothesis that the degrees of freedom parameter, $\nu$, can be tuned to optimize robustness, we used the Neal function and evaluated five distinct $\nu$ values ($\{1.0, 4.0, 30.0, 100, 500.0\}$) across three contamination levels ($p \in \{0.1, 0.3, 0.5\}$).

The study (Fig. 2j, 2k, 2l) shows that $\nu$ governs model robustness, with lower values yielding greater robustness. For

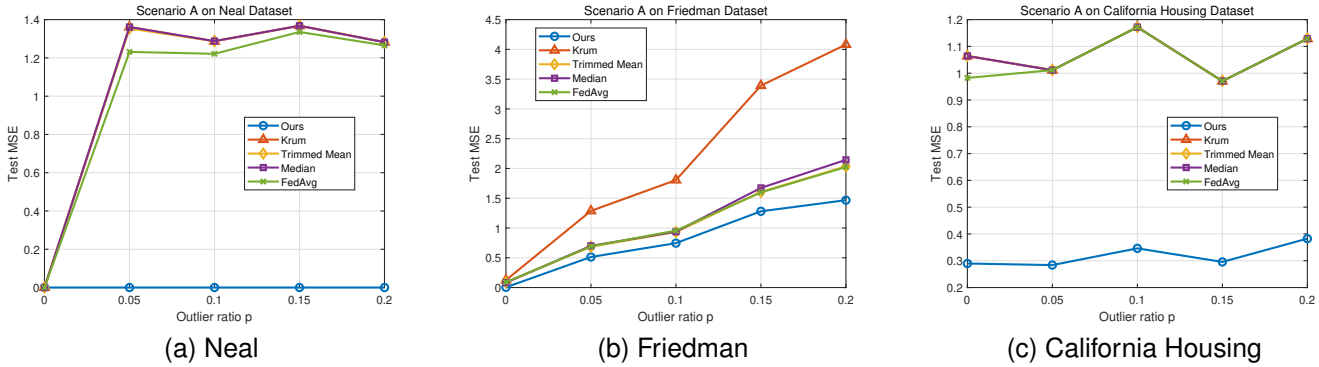



Fig. 3. Robustness comparison under Scenario A, where all nodes contain label outliers with different outlier ratios.

instance, under high contamination ($p = 0.5$), a choice of $\nu = 1.0$ was optimal. As $\nu$ grows ($\nu \geq 500.0$), the model's performance converges to that of the standard GPR, with MSE increasing by up to **145.7%** in high-contamination settings.

These findings confirm that $\nu$ is a key parameter for adapting the model. While very small values ($\nu = 1.0$) are best for highly contaminated data, our default choice of $\nu = 4.0$ provides a robust and effective setting for general applications, allowing the model's robustness to be tailored to the anticipated noise level.

### B. Comparison with Robust FL Aggregation Baselines

We additionally compare our method with four robust FL aggregation baselines: Krum [16], Trimmed Mean [17], coordinate-wise Median [17], and FedAvg [50]. These methods are originally designed for aggregating parametric model updates. For comparison, we apply them to the same non-parametric posterior states used in our aggregation step. Scenario A adds label outliers to all learners, while Scenario B corrupts only a subset of learners.

Fig. 3 shows the results of Scenario A. Since every learner contains label outliers, node-level filtering is less effective because there are no fully clean learners to select from. Our method gives much lower and more stable test MSE, as Student-$t$ GPR reduces the effect of abnormal labels during local posterior inference before aggregation.

Fig. 4 shows the results of Scenario B. This setting is closer to the usual assumption of robust aggregation methods, where only some learners produce corrupted updates. The proposed method remains stable as the corrupted-node ratio increases. These results indicate that classical baselines mainly handle abnormal node-level updates, whereas our method suppresses abnormal observations in the local posterior.

### C. Converge State with Extended Privacy Preserving Mode

We now analyze Algorithm 3's convergence across various network topologies and evaluate the privacy mechanism's impact on aggregation. For the simulation, key parameters were set to $\delta = 2$ and $p = 1020431$ to ensure correctness. The number of consensus iterations, $M$, was adapted for each topology to achieve convergence. The experiments were conducted on a network of 100 learners across eight representative topologies: Complete, Star, Ring, Small-World [51], IEEE-37 [52], and three random graphs with an average of 10, 20, and 40 neighbors per learner. The Small-World graph is generated with the Watts–Strogatz model ($n$=21, $k$=4, rewiring probability 0.3). IEEE-37 feeder is a 37-bus radial distribution network in power systems.

Fig. 5 shows the convergence trajectories for all 100 learners on each of the eight network topologies. The y-axis plots a combined error metric, $|\tilde{\mu}_i^{(1)}(m) - \mu^*| + |\tilde{\sigma}_i^{(1)}(m) - \sigma^*|$, representing the deviation of each learner's local estimate from the true global model parameters over the consensus iterations ($m$). In every subplot, despite different initial states and transient dynamics, all learners' models successfully converge to 0, demonstrating that consensus is achieved. This result also indicates that the proposed secure aggregation mechanism does not compromise the correctness of the final model.

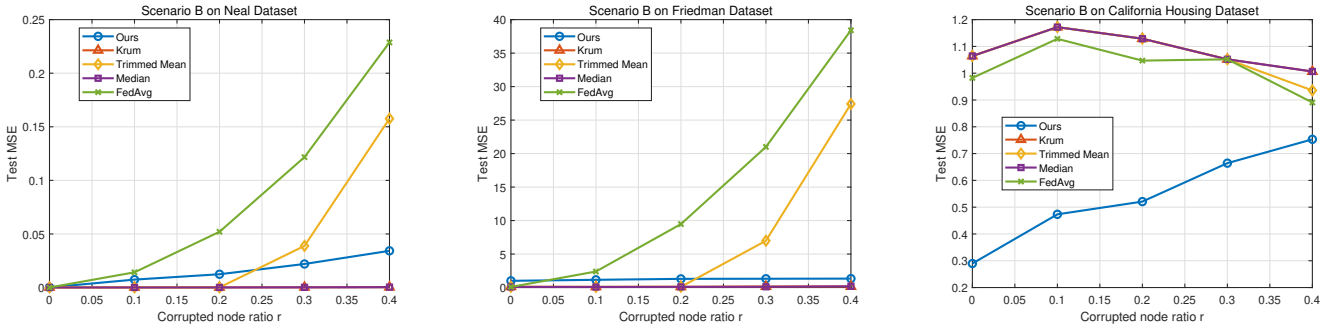
The algorithm's rate of convergence is highly dependent on network connectivity, with denser graphs converging faster. As analyzed in Fig. 6, which measures performance by plotting the decaying norm of the consensus error matrix $\left\| N\left(A^M - \frac{1_N 1_N^T}{N}\right)\right\|$, the results show this strong correlation. Dense topologies such as Complete, Neighbor40, and Small-World converge almost instantly, while sparse topologies like Ring and Star require significantly more iterations to reach consensus. These findings demonstrate that the algorithm's efficiency is directly tied to the network structure.

### D. Sensitivity Analysis

We study the effect of three parameters: the local subset size $Q$, the consensus iteration number $M$, and the low-degree threshold $n_{\text{nei}}$. The first parameter affects local GPR prediction, the second affects consensus accuracy, and the third controls the use of the extended privacy mode.

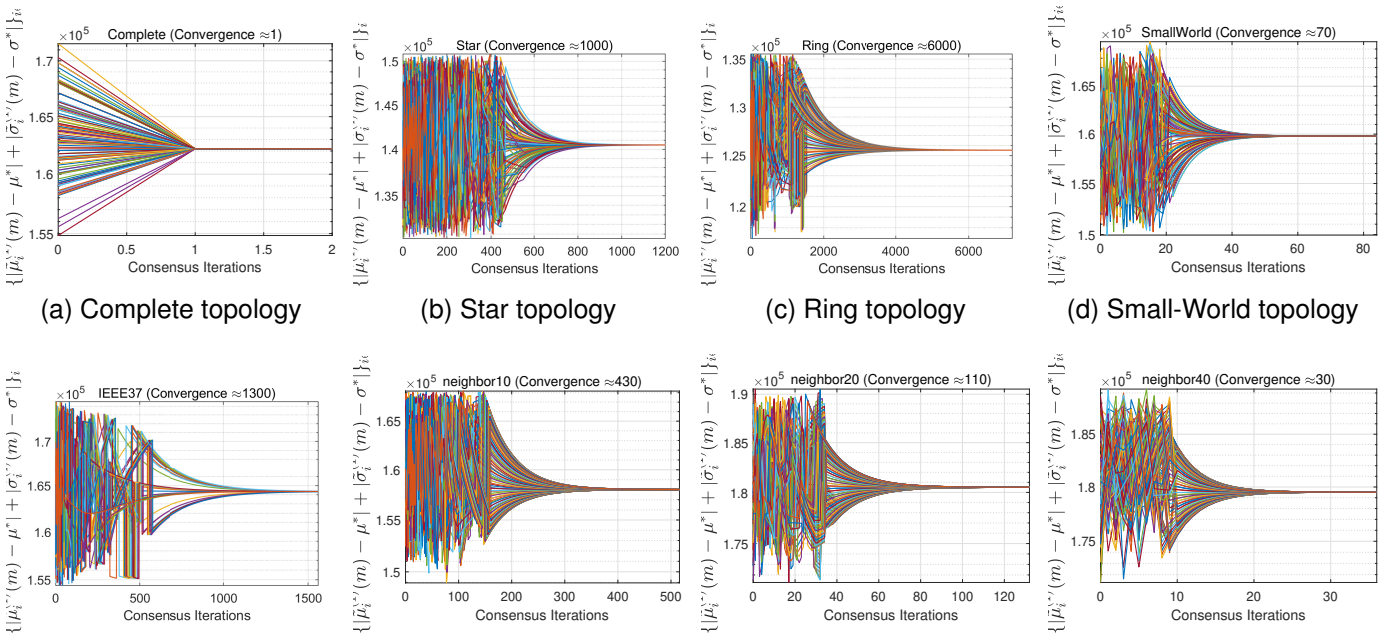
Fig. 7 shows the effect of $Q$. A larger $Q$ uses more nearby samples for local Student-$t$ GPR, but also increases the kernel inversion cost. The MSE becomes stable when $Q$ is moderately large, while the computation time keeps increasing. Thus, we use $Q = 100$ in the experiments as a balance between accuracy and cost.

The choice of $M$ depends on the convergence speed of the consensus matrix. A denser graph usually has faster decay and




(a) Neal (b) Friedman (c) California Housing

Fig. 4. Robustness comparison under Scenario B, where only a minority of nodes are corrupted with different corrupted-node ratios.




(a) Complete topology (b) Star topology (c) Ring topology (d) Small-World topology

(e) IEEE-37 topology (f) Neighbor10 topology (g) Neighbor20 topology (h) Neighbor40 topology

Fig. 5. Convergence behavior under different network topologies: Trajectories of $|\tilde{\mu}_i^{(1)}(m) - \mu^*| + |\tilde{\sigma}_i^{(1)}(m) - \sigma^*|$ for $i \in \mathcal{V}$

needs fewer iterations, while sparse graphs such as ring or star topologies need larger $M$. This trend is also observed in Fig. 5 and Fig. 6, where dense topologies converge faster and sparse topologies show slower error decay. In practice, $M$ can be increased until the aggregated prediction becomes stable.

Fig. 8 shows the effect of $n_{\text{nei}}$. Increasing this threshold protects more low-degree learners through encrypted forwarding, which reduces leakage in sparse topologies. The cost is higher privacy overhead. In dense topologies, the leakage is already low under the normal SSS mode, so a large threshold brings limited benefit. Therefore, $n_{\text{nei}}$ should be chosen according to graph sparsity, privacy risk, and available communication resources.

### *E. Privacy Leakage*

*1) Theoretical Analysis:* We evaluate privacy leakage under three typical communication topologies, from Fig. 9.

Following the standard terminology in secure multiparty computation and secret sharing [44], perfect secrecy means that the adversary's view is independent of the protected local value $\xi$, i.e.,

$$I(\xi; \text{View}_{\mathcal{A}}) = 0, \tag{29}$$

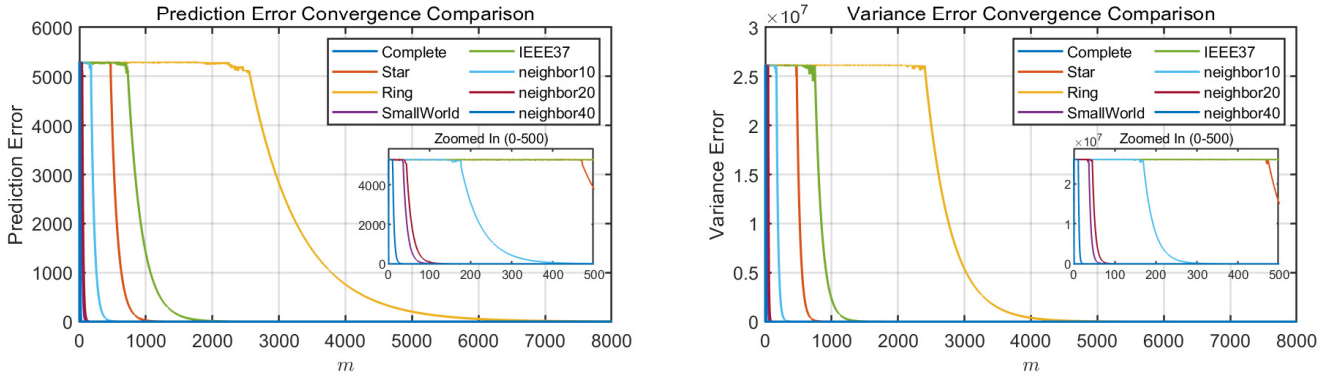
where $I(\cdot;\cdot)$ denotes mutual information and $\text{View}_{\mathcal{A}}$ denotes all messages observed by the adversarial learners during the protocol.

Computational secrecy means that any polynomial-time adversary can distinguish the protected value from a random one only with negligible advantage, i.e.,

$$|\Pr[\mathcal{D}(\text{View}_{\mathcal{A}}(\xi)) = 1] - \Pr[\mathcal{D}(\text{View}_{\mathcal{A}}(\xi')) = 1]| \leq \text{negl}(\lambda), \tag{30}$$

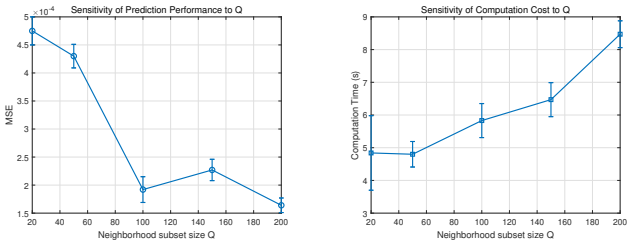
where $\lambda$ is the security parameter.

The work [29] applied SSS in distributed FL aggregation models, achieving perfect secrecy when each benign learner has at least one benign neighbor. For example, in Fig. 9 i),



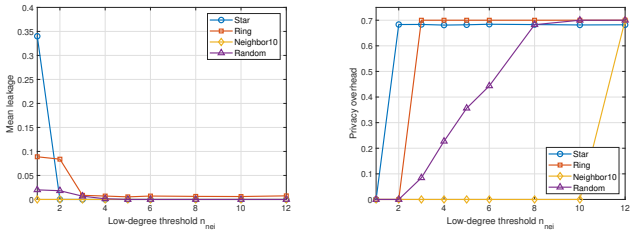

(a) Prediction error convergence trajectories (b) Variance error convergence trajectories

Fig. 6. Convergence performance under different network topologies through trajectories $\left\| N(A^M - \frac{1_N 1_N^T}{N}) \right\|$




(a) MSE versus $Q$ (b) Computation cost versus $Q$

Fig. 7. Sensitivity analysis of the neighborhood subset size $Q$.




(a) Mean leakage versus $n_{\text{nei}}$ (b) Privacy overhead versus $n_{\text{nei}}$

Fig. 8. Sensitivity analysis of the low-degree threshold $n_{\text{nei}}$.

learner 1 (an attacker) cannot access the model information of learner 2 as long as there is at least one benign learner among learners 3-6. In Fig. 9 ii), learners 3 and 6 protect learner 2's model unless both are adversarial. However, in the star topology (Fig. 9 iii), learner 2's information is directly exposed if its only neighbor is adversarial.

Our algorithm overcomes these vulnerabilities by using extended mode in low-connectivity scenarios. For example, in Fig. 9 ii) and iii), with the hybrid key encryption mechanism, learner 2's model information remains secure in extended mode provided that there is one benign learner among learners 3-6. By Theorem 3, under the standard IND-CPA security assumption, the algorithm maintains the same level of privacy across all three topologies.

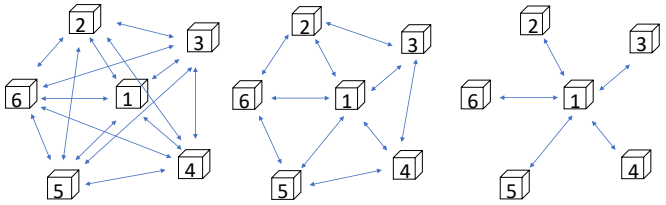


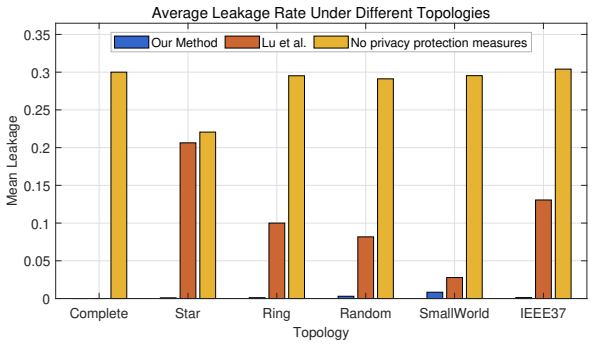
Fig. 9. Three common topologies, from left to right: i) Fully connected, ii) Non-fully connected, iii) Star

*2) Experiment on privacy leakage degree in different topological graphs:* We evaluate the proposed secret-sharing protocol on six network topologies—Complete, Star, Ring, Random, Small-World [51], and IEEE-37 [52].

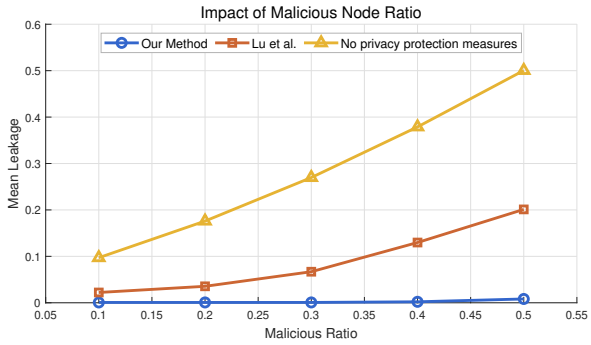
For each topology, a fraction $r \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ of nodes is randomly designated as semi-honest adversaries, and every $(\text{topology}, r)$ configuration is repeated for 15 independent trials. We use mean leakage to measure the average privacy exposure during one aggregation round. For each node, a leakage score is computed according to whether the adversarial set can obtain sufficient neighborhood information under the corresponding protection mode. The reported mean leakage is then averaged over all nodes and trials.

The proposed protocol is compared with the neighbour-threshold method of [29] and with an unprotected baseline.

Fig. 10(a) reports the average leakage across topologies. The unprotected baseline gives a leakage score higher than 0.29 in all cases. The method of [29] proves effective only on dense graphs, but on sparse structures such as Star and Ring it still shows noticeable leakage. By contrast, the proposed protocol keeps leakage below 0.01 for every topology, effectively suppressing privacy loss. Fig. 10(b) aggregates the six topologies and shows the evolution of leakage as the malicious-node ratio increases. When $r$ rises to 0.5, the baseline reaches 0.50 and [29] climbs to 0.20. The proposed algorithm therefore limits the mean leakage score to below 0.02 across all tested connectivity topologies and adversary densities, confirming




(a) Average leakage in six representative topologies




(b) Impact of malicious-node ratio across all topologies

Fig. 10. Privacy leakage evaluation

its practicality for deployment in sparse or heterogeneous federated networks.

### F. Overhead with Extended Privacy-Preserving Mode

To evaluate the efficiency of the proposed privacy algorithm, this section analyzes its computational and communication overhead across various network topologies, with a detailed breakdown presented in Table II.

The metrics in the table are defined as follows: $M$ is the total number of consensus iterations required to reach convergence in a single round; $t_{\text{gpr-t}}$ denotes the local Student-t GPR training time; $t_{\text{PoE}}$ is the total computational time for PoE aggregation over all $M$ iterations; $t_{\text{computation}}$ is the total computational overhead per round, calculated as the sum of $t_{\text{gpr-t}}$ and $t_{\text{PoE}}$; Msg(kB) represents the average communication overhead per single consensus iteration; $t_{\text{SSS}}$ is the computational overhead from SSS; $t_{\text{key}}$ is the overhead from key encryption, which is triggered in the extended mode for sparsely-connected learners; and $t_{\text{privacy}}$ is the total privacy-related computation overhead, being the sum of $t_{\text{SSS}}$ and $t_{\text{key}}$.

The results show a trade-off between computation and communication shaped by the network topology. Denser topologies like Complete and neighbor40 require very few consensus iterations ($M = 1$ and $M = 30$, respectively) but incur a high communication cost per iteration (9281.2 kB and 4687.5 kB). Conversely, sparser topologies such as Ring and Star need thousands of iterations to converge, which significantly increases the total aggregation time ($t_{\text{PoE}}$), even though their per-iteration communication cost is much lower.

Furthermore, the overhead of the privacy-preserving mechanism ($t_{\text{privacy}}$) remains efficient compared to the overall computation time, particularly in topologies that require extensive consensus steps. For instance, in the Ring topology, which needs 6000 iterations for consensus, the privacy overhead (196.5 ms) is less than 8% of the total computation time ($t_{\text{computation}}$). This shows that the computational cost of our privacy mechanism is well-managed, ensuring strong security without imposing an excessive burden on the overall process.

## VIII. Conclusion

This paper presented a robust, privacy-preserving decentralized FL algorithm for online learning with outliers. The algorithm uses a non-parametric approach, integrating GPR with a Student-t likelihood for robust local learning against outliers. A consensus-based PoE algorithm enables robust peer-to-peer aggregation, and a dual-mode mechanism combining SSS with public-key encryption secures the process and enhances privacy in sparse networks.

Future technical efforts could focus on enhancing the PoE aggregation against over-weighted models and evaluating performance on non-independent and identically distributed (non-IID) or drifting data. Further validation in dynamic scenarios like autonomous vehicle networks or real-time healthcare monitoring is needed to demonstrate practical viability.

## References

[1] A. M. Elbir, B. Soner, S. Çöleri, D. Gündüz, and M. Bennis, "Federated learning in vehicular networks," in *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pp. 72–77, IEEE, 2022.

[2] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[3] R. Cioffi, M. Travaglioni, G. Piscitelli, A. Petrillo, and F. De Felice, "Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions," *Sustainability*, vol. 12, no. 2, p. 492, 2020.

[4] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of healthcare informatics research*, vol. 5, pp. 1–19, 2021.

[5] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, vol. 597, p. 128019, 2024.

[6] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19188–19209, 2024.

[7] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-resilient secure federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2168–2181, 2020.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, 2017.

[9] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *2020 IEEE International Conference on Big Data*, pp. 15–24, IEEE, 2020.

[10] A. Mitra, H. Hassani, and G. J. Pappas, "Online federated learning," in *2021 60th IEEE Conference on Decision and Control*, pp. 4083–4090, IEEE, 2021.

[11] X. Zhang, Z. Yuan, and M. Zhu, "Byzantine-tolerant federated gaussian process regression for streaming data," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 13499–13511, 2022.

[12] Z. Yang, D. Zhang, X. Dai, F. Yu, C. Zhang, B. Huang, H. Sadeghian, and S. Haddadin, "Streaming generated gaussian process experts for online learning and control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, pp. 27719–27727, 2026.

TABLE II
PER-ROUND OVERHEAD ANALYSIS FOR THE PROPOSED ALGORITHM ACROSS VARIOUS NETWORK TOPOLOGIES.

| Topology | $M$ | Computation (ms) | | | Communication | Privacy Overhead (ms) | | |
|---|---|---|---|---|---|---|---|---|
| | | $t_{\text{gpr-t}}$ | $t_{\text{PoE}}$ | $t_{\text{computation}}$ | Msg(kB) | $t_{\text{SSS}}$ | $t_{\text{key}}$ | $t_{\text{privacy}}$ |
| Complete | 1 | 31.0 | 20.3 | 51.4 | 9281.2 | 20.0 | 2.8 | 22.8 |
| Star | 1000 | 31.0 | 513.4 | 544.5 | 278.4 | 129.4 | 110.2 | 239.6 |
| Ring | 6000 | 31.0 | 2537.7 | 2568.7 | 351.6 | 25.8 | 170.7 | 196.5 |
| Small-World | 70 | 31.0 | 97.2 | 128.2 | 633.0 | 111.5 | 0.7 | 112.2 |
| IEEE-37 | 1300 | 31.0 | 650.1 | 681.2 | 301.6 | 43.5 | 126.0 | 169.5 |
| neighbor10 | 430 | 31.0 | 396.9 | 428.0 | 1171.9 | 318.7 | 63.3 | 382.0 |
| neighbor20 | 110 | 31.0 | 326.9 | 357.9 | 2343.8 | 55.0 | 0.6 | 55.6 |
| neighbor40 | 30 | 31.0 | 292.8 | 323.8 | 4687.5 | 20.0 | 0.6 | 20.6 |


[13] A. O'Hagan, "On outlier rejection phenomena in bayes inference," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 41, no. 3, pp. 358–367, 1979.

[14] Z. Luan, W. Li, M. Liu, and B. Chen, "Robust federated learning: Maximum correntropy aggregation against byzantine attacks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 62–75, 2024.

[15] C. Feng, A. H. Celdrán, J. Von der Assen, E. T. M. Beltrán, G. Bovet, and B. Stiller, "Dart: A solution for decentralized federated learning model robustness analysis," *Array*, vol. 23, p. 100360, 2024.

[16] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[17] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International conference on machine learning*, pp. 5650–5659, Pmlr, 2018.

[18] A. Zhang, P. Zhao, W. Lu, and G. Zhang, "Decentralized federated learning towards communication efficiency, robustness, and personalization," *ACM Transactions on Sensor Networks*, vol. 21, no. 3, pp. 1–20, 2025.

[19] L. Xu, D. Xu, X. Yi, C. Deng, T. Chai, and T. Yang, "Decentralized federated learning algorithm under adversary eavesdropping," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 2, pp. 448–456, 2025.

[20] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Transactions on Big Data*, vol. 10, no. 2, pp. 194–213, 2024.

[21] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1322–1333, 2015.

[22] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 739–753, IEEE, 2019.

[23] J. He, L. Cai, and X. Guan, "Differential Private Noise Adding Mechanism and Its Application on Consensus Algorithm," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4069–4082, 2020.

[24] B. Jeon, S. M. Ferdous, M. R. Rahman, and A. Walid, "Privacy-preserving decentralized aggregation for federated learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2021.

[25] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, "Distributed Privacy-Preserving Data Aggregation Against Dishonest Nodes in Network Systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1462–1470, 2019.

[26] S. Gade and N. H. Vaidya, "Privacy-preserving distributed learning via obfuscated stochastic gradients," in *2018 IEEE Conference on Decision and Control*, pp. 184–191, 2018.

[27] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, 2013.

[28] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *Proceedings of the 2016 IEEE 55th Conference on Decision and Control*, pp. 5053–5058, 12 2016.

[29] Y. Lu, Z. Yu, and N. Suri, "Privacy-preserving decentralized federated learning over time-varying communication graph," *ACM Transactions on Privacy and Security*, vol. 26, no. 3, pp. 1–39, 2023.

[30] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[32] D. C. Montgomery, *Introduction to statistical quality control*. John Wiley & Sons, 2007.

[33] S. M. Kay, "Fundamentals of statistical signal processing: Estimation theory," 1993.

[34] C. Hazay and Y. Lindell, *Efficient secure two-party protocols: Techniques and constructions*. Springer Science & Business Media, 2010.

[35] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[36] J. Vanhatalo, P. Jylänki, and A. Vehtari, "Gaussian process regression with student-t likelihood," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 22, 2009.

[37] R. Martinez-Cantin, K. Tee, and M. McCourt, "Practical bayesian optimization in the presence of outliers," in *International conference on artificial intelligence and statistics*, pp. 1722–1731, PMLR, 2018.

[38] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[39] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[40] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[41] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pp. 63–70, IEEE, 2005.

[42] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[43] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[44] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure multiparty computation and secret sharing*. Cambridge University Press, 2015.

[45] M. J. Dworkin, E. Barker, J. R. Nechvatal, J. Foti, L. E. Bassham, E. Roback, J. F. Dray Jr, *et al.*, "Advanced encryption standard," 2001.

[46] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[47] R. M. Neal, "Monte carlo implementation of gaussian process models for bayesian regression and classification," *arXiv preprint physics/9701026*, 1997.

[48] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[49] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

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

[51] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[52] W. H. Kersting, "Radial distribution test feeders," in *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 01CH37194)*, vol. 2, pp. 908–912, IEEE, 2001.