TriGuard-FL: A User-Centric Trust Triad in Federated Learning via Auditable Data, Verifiable Contributions, and Antidote-Driven Mitigation

Supplementary Material

In this supplementary material, we provide additional details, explanations, definitions, and experimental results, including ablation studies, that could not be included in the main paper due to space limitations. The content is structured into clear sections to enhance readability and to support a comprehensive understanding of our work.

1. Extended Challenges in Ensuring Auditable Data Valuation and Verifiable Client Updates

The performance and reliability of any federated learning (FL) system depend on the quality and integrity of client-held private data [24, 31]. However, assessing the utility of this data before training remains a largely unsolved problem due to the server's restricted access, FL's decentralized architecture, and client-level data heterogeneity [18, 21]. Existing data curation and validation strategies suffer under constraints like small sample sizes, unknown trustworthiness, and missing labels [23, 34]. Also, contrary to intuition, smaller datasets often pose greater privacy risks than larger ones [11]. Due to reduced diversity, individual records in small datasets are more distinguishable and thus more susceptible to re-identification attacks [43]. Further, the problem intensifies under data poisoning attacks, where adversarial clients inject subtly crafted poisoned samples into the training data that degrade global model performance [8, 38]. Most existing defense methods rely on post-training heuristics, trusted servers, or access to clean validation data, introducing scalability bottlenecks and privacy vulnerabilities [5]. Moreover, verifiable client updates require trusted training data, which is hard to ensure without pre-training auditable data valuation. Therefore, this work focuses on auditable data valuation before training as a basis for verifiable client updates in the presence of online, targeted & untargeted, black-box data poisoning attacks that are highly relevant in practical FL deployments, as motivated in recent FL threat studies [20, 34].

2. The Importance of Distribution-level Data Valuation in FL

Traditional data valuation methods in FL quantify the utility of a fixed dataset, typically a fixed set of samples [15, 23]. However, this approach overlooks a critical insight that the true utility of data is fundamentally connected to the statistical properties of the distribution from which it is drawn [10, 23, 43]. In many practical applications, such as data marketplaces or privacy-sensitive domains, only limited, potentially noisy samples are accessible, and decisions must be made about the value of the generating distribution [43]. This perspective also applies in cross-silo FL, where clients hold small, private, and often non-IID datasets drawn from unknown distributions [46]. In addition, current FL sample-level data valuation methods overlook two key aspects, namely, (i) how to formally assign value to a client data distribution rather than a fixed set of data samples and (ii) how to do so actionably using only sample-level previews while preserving the raw client data privacy. A distribution-level data valuation addresses both concerns by leveraging masked statistical moments (e.g., the mean and covariance of feature embeddings of client-side's data), enabling efficient assessment of data quality without exposing raw client data. Hence, shifting from a sample level to a distribution-level data valuation evaluates client data utility, supports a robust model aggregation under attacks, and builds cross-client trust in FL systems [43].

3. Related Work

This section extends the discussion from the main paper's Section 1. Introduction by categorizing existing FL approaches and contrasting them with TriGuard-FL. Table 1 summarizes a comparative analysis of representative FL frameworks across three key dimensions of the proposed trust triad, namely, auditable data valuation (ADV), verifiable client contribution (VCC), and mitigability of poisonous updates (MPU), along with additional factors capturing robustness and deployment practicality. The robustness columns evaluate resistance to poisoned or anomalous data (PDD), data efficiency under small-sample and non-IID conditions (DE, n-IID), while the requirements columns measure practical deployment aspects such as privacy preservation (PP), external validation data dependency (VD), and reliance on pretrained models (PM). For clarity, methods are grouped by design category, including trusted execution environments (TEE)-based, loss-function rejection (LFR), Byzantine-robust aggregation, blockchain-based, and valuation-driven approaches.

Rationale behind comparative assessment. The circle indicators and qualitative scores in Table 1 represent category-level estimations rather than exact quantitative values for individual methods. Each entry reflects a fair, averaged interpretation of how representative approaches within that category (e.g., TEE-based, Byzantine-robust, valuation-based) satisfy the evaluated dimensions of trust-

Table 1. Comparative analysis of FL frameworks across trustworthiness, robustness, and practical requirements. (**C**: client-side, **S**: server-side, **H**: hybrid). ●: strong support, ◑: partial support, ○: none. ✗: undesirable & not present, ✓: undesirable & present.

FL framework category		Trust triad			Robustness			Requirements		
Methodology (source)		ADV	VCC	MPU	PDD	DE	n-IID	PP	VD	PM
TEE (**C**) [6, 28, 29]		○	◑	○	◑	○	◑	◑	✗	✗
LFR (**S**) [8, 16, 38]		○	●	○	○	○	◑	◑	✓	✓
Byzantine robust aggregation (**S**) [4, 5, 45]		◑	●	○	◑	○	◑	◑	✓	✗
Blockchain-based (**C**) [7, 18, 24]		◑	◑	○	○	○	◑	◑	✗	✗
Valuation based	Model valuation (**S**) [9, 17, 39]	○	●	○	○	○	●	◑	✓	✓
	Data valuation (**C**) [21, 23, 31]	◑	○	○	◑	○	◑	◑	✓	✓
Trust-triad (H)	**TriGuard-FL (ours)**	●	●	●	●	●	●	●	✗	✗

worthiness, robustness, and deployment practicality. The assessment is derived from an extensive literature review of their methodological scope and reported outcomes, emphasizing the extent to which these frameworks adhere to the three trust-triad properties, as well as auxiliary robustness and implementation requirements. This approach provides a balanced, interpretable comparison across diverse FL paradigms without bias toward any specific implementation.

TEEs have been utilized in FL to ensure confidentiality and integrity during client-side training. For instance, Chen *et al.* [6] proposed a TEE-based FL framework with a training integrity protocol that enforces correct execution and detects causative data poisoning. Similarly, Mo *et al.* [28] introduced PPFL, which deploys TEEs on the client side to secure model updates in mobile FL systems, mitigating various inference attacks while maintaining model utility and low overhead. While effective for safeguarding model parameters and preventing direct tampering, TEEs provide no support for (ADV) and MPU, since data quality and distributional trust are not observable within the encrypted environment. VCC is also limited, as the server must rely on enclave attestations rather than interpretable or quantifiable trust measures. From a robustness perspective, TEE-based frameworks exhibit only partial protection against data poisoning (PDD) and low data efficiency (DE) due to their inability to evaluate sample utility or correct malicious gradients. Although they offer moderate privacy preservation (PP), they lack adaptability to non-IID data distributions and are often impractical for large-scale or heterogeneous deployments because of their hardware dependency and high setup cost.

LFR is a defense strategy in FL that filters out potentially poisoned updates by discarding those with high validation loss on the server-side. For example, FedCE by Jiang *et al.* [16] enhances fairness by leveraging prediction error and gradient direction, using an auxiliary model to guide aggregation. While these methods improve basic robustness to poisoned updates, they offer only VCC and no ADV and MPU mechanisms for understanding rejection decisions. Their MPU remains limited since rejected updates are simply discarded rather than refined, which can hinder convergence and reduce generalization in non-IID scenarios. Moreover, LFR frameworks rely heavily on clean validation data or pretrained models, which compromises privacy preservation (PP) and restricts scalability in privacy-sensitive FL deployments.

Byzantine-robust aggregation methods aim to defend FL against malicious or faulty participants by identifying and mitigating anomalous updates on the server-side. Techniques such as Krum [4], which select updates closest to the majority, and the median or trimmed mean-based approaches by Yin *et al.* [45], offer provable robustness under various convex and non-convex settings. These methods are particularly effective in large-scale or serverless FL systems. However, as summarized in Table 1, these methods offer only partial support for (ADV), since client contributions are evaluated solely through statistical distances without interpretable validation. They also lack explicit MPU as malicious updates are discarded rather than refined, which limits convergence and stability in heterogeneous or cross-silo settings. From a robustness perspective, these techniques exhibit only partial resistance to data poisoning (PDD) and struggle under non-IID distributions, where adversaries can craft updates that mimic benign gradients [5]. Moreover, their reliance on distance-based aggregation and server-side anomaly detection introduces privacy exposure and sensitivity to differential privacy noise [1, 9].

Blockchain-based methods in FL aim to enhance auditability, integrity, and decentralization by leveraging distributed ledgers and smart contracts at the client-side. For example, DFL [18] by Kalapaakin *et al.* employs smart contracts to validate local models and ensure transparent, tamper-proof aggregation without relying on a central server. While these methods remove single points of failure, many still depend on committee-based consensus, inadvertently introducing centralized control. However, as reflected in Table 1, blockchain-based solutions provide only partial ADV and limited VCC, since they focus on post-aggregation record keeping rather than pre-aggregation validation or distribution-level trust evaluation [7, 18, 24]. They also lack explicit MPU mechanisms to counter poisoned or low-quality updates, relying solely on consensus for integrity without assessing data reliability or update quality. From a deployment standpoint, these frameworks suffer from high computational and synchronization overhead, and their privacy preservation (PP) is limited, as the public nature of blockchain can expose sensitive client information during consensus or ledger synchronization. Furthermore, most designs depend on committee-based consensus, which reintroduces partial centralization and re-

stricts scalability in cross-device FL.

Valuation-based methods in FL aim to assess the quality of client updates or data to enhance robustness, fairness, and accountability during model training, either on the client or server side. Model valuation methods estimate the trustworthiness of local updates based on their behaviour. For instance, FLTrust [5] uses a server-side root dataset to assign trust scores and achieve robust aggregation even under Byzantine attacks. FLShield [17], in contrast, validates client models using benign samples from other participants, enabling privacy-preserving defense against poisoning and backdoor threats.

On the other hand, data valuation methods quantify the value of clients' data contributions. FedBary [23] computes data utility using Wasserstein distance without relying on specific training algorithms or validation datasets, promoting fairness and transparency. Despite these advances, as summarized in Table 1, valuation-based frameworks typically offer only partial auditability (ADV) and verifiability (VCC). They focus primarily on estimating contribution weights but lack mitigability (MPU), as identified anomalies are excluded rather than corrected, limiting adaptability under non-IID and dynamic adversarial conditions. Moreover, most traditional data valuation techniques [10, 15, 35] treat utility as a discrete, sample-level property, overlooking that true data value is inherently linked to the underlying distributional statistics from which samples are drawn. This sample-centric view restricts generalization across heterogeneous clients and fails to capture temporal or distributional drift in real-world deployments.

In summary, prior FL defenses address isolated facets of trustworthiness, focusing either on secure aggregation, anomaly filtering, or contribution weighting, but none provide a unified framework that simultaneously ensures ADV, VCC, and MPU. Most existing approaches lack the adaptability to handle dynamic adversaries, struggle under non-IID data, or rely on restrictive assumptions such as trusted validation data or centralized supervision. In contrast, our proposed TriGuard-FL achieves all three trust-triad dimensions concurrently while maintaining robustness to poisoned and scarce data, scalability across heterogeneous FL settings, and privacy-preserving deployment through masked distributional sharing. This holistic design establishes TriGuard-FL as the first end-to-end trustworthy FL framework that unifies auditability, verifiability, and mitigability within a practical and resource-efficient architecture.

4. Additional Preliminaries

This section provides extended background and mathematical context to complement the Preliminaries and Threat Model section presented in the main paper. We elaborate on additional definitions, notations, and assumptions that support the theoretical analysis and experimental design of TriGuard-FL. For clarity and ease of reference, Table 2 summarizes the key notations used throughout both the main and supplementary material.

Table 2. Summary of adopted notations

Notation	Definition
n	Number of clients
$\mathcal{D}_k$	k^{th} client local data
$\mathcal{C}_k$	k^{th} client
$\mathcal{N}_k$	Number of data samples of k^{th} client
λ	Weighting factor for federated aggregation
m	Number of malicious clients
$\hat{\mathcal{C}}_k$	k^{th} malicious local client
$\hat{\mathcal{D}}_k$	k^{th} client local poisoned data
$\psi(.)$	Data poisoning function
τ	Data perturbation strength
Δ	Gradient noise vector used in PGD attack [25]
ι	Backdoor trigger pattern used in DBA attack [42]
$G^t_{\theta_g}$	Global model with parameters θ_g at round t
ν	Number of poisoned samples
μ	Mean of local data embeddings
Σ	Covariance of local data embeddings
$\mathcal{E}$	Latent embedding set
$\Phi(.)$	Penultimate layer embedding
ρ	Data poisoning bound
f_θ	Local model
$\nabla\theta^t_k$	k^{th} client local update at time t
$\mathcal{A}_\}$	Global test accuracy without attack
$\mathcal{A}'_\}$	Global test accuracy under attack
$\mathcal{D}_\}$	Global test accuracy degradation under attack
$\mathcal{P}^*_\}$	Global statistical moment distribution at the server
γ	Statistical confidence boundary
η	Separability of valuated data
$\mathcal{D}'_k$	TriGuard-FL valuated data
$\nabla\theta'^t_k$	Antidote gradient injected update
$\mathcal{S}_k$	Softmax probabilities
$\sigma(.)$	Softmax function
Π_k	Trust score of client C_k
$\mathcal{L}_{CE}$	Cross-entropy loss function
$\hat{\mathcal{P}}_k$	Gaussian noise parameters
$\mathcal{X}_{test}$	Test data at the server
β	non-IID Dirichlet parameter
φ	Dynamic trust threshold
α	Parameter to control mitigation strength

Definition 4.1 (Latent-space representation in FL). *Each client $\mathcal{C}_k$ maintains a local model f_{θ_k} with parameters θ_k, decomposed as $f_{\theta_k}(x) = h_{\theta_k}(\Phi_{\theta_k}(x))$, where $\Phi_{\theta_k}(x) \in \mathbb{R}^\ell$ denotes the penultimate-layer embedding of input $x \in \mathcal{X}_k$ and $h_{\theta_k}(\cdot) : \mathbb{R}^\ell \to \mathbb{R}^C$ is the classifier head. Here, $\ell \ll d$, allowing $\Phi_{\theta_k}(x)$ to encode the semantic structure of client data in a low-dimensional latent space. These representations are effective for identifying distributional characteristics used in clustering, anomaly detection, and trust assessment across clients [13, 41]. Let $\mathcal{E}_k = \{\Phi_{\theta_k}(x) \mid x \in$*

$\mathcal{X}_k\}$ *be the latent embedding set for client* $\mathcal{C}_k$. *We approximate the local distribution in this space using its first- and second-order moments:* $\mu_k = \frac{1}{|\mathcal{E}_k|}\sum_{e\in\mathcal{E}_k} e, \quad \Sigma_k = \frac{1}{|\mathcal{E}_k|}\sum_{e\in\mathcal{E}_k}(e-\mu_k)(e-\mu_k)^\top$. *The tuple* (μ_k, Σ_k) *provides a compact, "privacy-preserving statistical summary" of client* k*'s data distribution.*

Definition 4.2 (**Maximum mean discrepancy** (MMD)**[12]**). *It is a non-parametric metric used to quantify the difference between two probability distributions* $\mathbb{P}$ *and* $\mathbb{Q}$ *over a domain* $\mathcal{Z}$ *of functions* ζ *(similar to latent embeddings Definition 4.1), based on samples drawn from each of them. Let* $\mathcal{H}_k$ *be a reproducing kernel Hilbert space (RKHS). Then,*

$$\mathrm{MMD}(\mathbb{P},\mathbb{Q}) := \sup_{\zeta\in\mathcal{Z}} \left(\mathbb{E}_{x\sim\mathbb{P}}[\zeta(x)] - \mathbb{E}_{y\sim\mathbb{Q}}[\zeta(y)]\right),$$

where $\mathcal{Z} = \{\zeta \in \mathcal{H}_k \mid \|\zeta\|_{\mathcal{H}_k} \le 1\}$ *is the* $\mathcal{H}_k$ *unit ball.* ***Closed-form Gaussian MMD [33].*** *When both distributions follow multivariate Gaussian, s.t.* $\mathbb{P} = \mathcal{N}(\mu_1, \Sigma_1)$ *and* $\mathbb{Q} = \mathcal{N}(\mu_2, \Sigma_2)$*, and use a Gaussian RBF characteristic kernel* $\mathcal{K}(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$*, the squared maximum mean discrepancy has a closed-form*

$$\begin{aligned}\mathrm{MMD}^2(\mathbb{P},\mathbb{Q}) = \mathcal{K}(\mu_1,\mu_1;\,2\Sigma_1) + \mathcal{K}(\mu_2,\mu_2;\,2\Sigma_2)\\ -\ 2\mathcal{K}(\mu_1,\mu_2;\,\Sigma_1+\Sigma_2),\end{aligned} \tag{1}$$

where

$$k(\mu_i,\mu_j;\,\Sigma) = \left|\mathbf{I} + \frac{2\Sigma}{\sigma^2}\right|^{-\frac{1}{2}} \times \exp\left(-\frac{1}{2}(\mu_i-\mu_j)^\top \left(\Sigma + \frac{\sigma^2}{2}\mathbf{I}\right)^{-1}(\mu_i-\mu_j)\right). \tag{2}$$

Here, $\mathbf{I}$ *is the identity matrix of the appropriate dimension. The scalar* σ^2 *is the bandwidth parameter of the Gaussian (RBF) kernel, controlling its smoothness. The determinant* $|\cdot|$ *measures the volume scaling of a matrix, while* $(\cdot)^{-1}$ *denotes the matrix inverse within the exponential. The function* $\exp(\cdot)$ *is the exponential function. This kernel computes the expected value of the Gaussian kernel between two Gaussian-distributed variables in closed form. We adopt this formulation to efficiently compute trust scores over masked client distributions while preserving distributional privacy.*

4.1. Auditable Data Valuation, Verifiable Contributions, and Mitigable Poisonous Updates

In FL, auditable data valuation is the process of systematically evaluating the utility, quality, and integrity of a client's local dataset $\mathcal{D}_k$ based on transparent, data-driven metrics, rather than dataset size alone [18, 23, 43]. This process enables accountability and fairness by quantifying the contribution of each client based on statistical alignment with the global reference distribution $\mathcal{P}_g^*$. Verifiable client contributions ensure that each submitted update $\nabla\theta_k^t$ results from legitimate training over high-quality data [17]. This approach prevents malicious or low-quality updates from distorting the global model and provides a basis for independently enabling transparent attribution of client behavior and overall trust [17]. Beyond auditability and verifiability, TriGuard-FL introduces the notion of mitigability of poisonous updates, which addresses the limitation of rejection-only defenses. Rather than discarding suspicious gradients outright, the poisonous updates are adaptively neutralized, which projects each update $\nabla\theta_k^t$ onto a safe descent subspace defined by the trusted gradient direction g_t. This trust-scaled correction reduces the adversarial influence of low-trust clients while preserving benign learning, ensuring that global convergence remains stable even under partially compromised client participation.

Together, these three components, auditable data valuation, verifiable client contributions, and mitigability of poisonous updates, establish the foundation of TriGuard-FL's trust-triad framework, enabling transparency, accountability, and active resilience in federated optimization.

4.2. Dirichlet Distribution in FL

The impact of non-IID data distribution in FL is critical to understand, particularly the role of the Dirichlet parameter β in shaping client data partitions. The Dirichlet distribution [27] is widely used to model data heterogeneity across clients, with β controlling the degree of non-IIDness. Formally, its density is defined as

$$p(x_1, x_2, \ldots, x_C \mid \beta) = \frac{1}{B(\beta)}\prod_{i=1}^{C} x_i^{\beta_i - 1},$$

where x_i denotes the proportion of samples from class i, C is the number of classes, $\beta = (\beta_1, \ldots, \beta_C)$ is the concentration parameter, and $B(\beta)$ is the multivariate Beta function:

$$B(\beta) = \frac{\prod_{i=1}^{C}\Gamma(\beta_i)}{\Gamma\left(\sum_{i=1}^{C}\beta_i\right)}.$$

By adjusting β, one can control the heterogeneity of client data. Larger β yields near-uniform partitions, approximating IID distributions, whereas smaller β produces skewed allocations, leading to strong non-IIDness. Proper tuning of β is thus essential to reflect real-world heterogeneity, ensuring robustness and generalization of FL models. In our study, we further examine how varying β influences FL attack dynamics, since non-IID client data can significantly affect global model accuracy even before adversarial interventions.

5. Proposed Framework: Extended Details

Algorithm 1 outlines the complete execution flow of TriGuard-FL, emphasizing the key enhancements over standard FL. Algorithms 2 and 3 provide detailed formulations of the proposed auditable data valuation and antidote gradient injection procedures, respectively, complementing the conceptual description presented in the Proposed Approach section of the main paper. Finally, Figure 1 illustrates the algorithmic intuition behind the AGI mechanism, offering a geometric interpretation of how trust-scaled mitigation neutralizes adversarial gradients while preserving benign learning directions.

Algorithm 1 Proposed **TriGuard-FL** framework

Input: Global model $G^t_{\theta_g}$ with parameters $\nabla\theta^t_g$, client data $\mathcal{D}_k = (\mathcal{X}_k, \mathcal{Y}_k)$, privacy noise distribution parameters $\mathcal{P}_\delta = (\mu_\delta, \Sigma_\delta)$, antidote gradient direction g_t
Output: Global test accuracy $\mathcal{A}_g$

Client Execution ($\nabla\theta^t_g, \mathcal{P}^*_g = (\mu_g, \Sigma_g)$):
for each client $i = 1$ to n **do**
  Initialize local model with global model $f^t_{\theta_i} \leftarrow G^t_{\theta_g}$
  if client $\mathcal{C}_i$ is malicious **then**
    Apply poisoning: $\psi(\mathcal{X}_i) \leftarrow \mathcal{X}_i + \tau \times \Delta$
    $\hat{\mathcal{D}}_i \leftarrow (\psi(\mathcal{X}_i), \mathcal{Y}_i)$, $\mathcal{D}_i \leftarrow \hat{\mathcal{D}}_i$
  $\mathcal{D}'_i, \mathcal{P}_i, \Pi_i$: `Val` $(\mathcal{D}_i, \mathcal{P}^*_g)$ **[Auditability]** ▷ `ADV`
  for each batch b in $\mathcal{D}'_i$ **do**
    $\mathcal{S}_{b,i} \leftarrow \sigma(f^t_{\theta_i}(\mathcal{X}_i[b]))$
    Compute loss: $\mathcal{L}_{CE_i}(\mathcal{S}_{b,i}, \mathcal{Y}_{b,i})$
    Update model: $\nabla\theta^t_i \leftarrow \nabla\theta^t_i - \eta\nabla_\theta\mathcal{L}_{CE_i}$
  Compute $\hat{\mathcal{P}}_i$: $\hat{\mu}_i = \mu_i + \mu_\delta$, $\hat{\Sigma}_i = \Sigma_i + \Sigma_\delta$
  $\nabla\theta^t_i \leftarrow \nabla\theta^t_i - \nabla\theta^t_g$ **[Verifiability]** ▷ `VCC`
  return $\nabla\theta^t_i, \hat{\mathcal{P}}_i, \Pi_i$
Server Execution ($\nabla\theta^t_i, \hat{\mathcal{P}}_i, \Pi$):
Select k low-trust (high Π) clients
$\nabla\theta'^t_i$: `AGI` $(\nabla\theta^t_i, \Pi_i, g_t)$ **[Mitigability]** ▷ `MPU`
$\nabla\theta^{t+1}_g = \sum_{i=1}^{k} \Pi_i \nabla\theta'^t_i + \sum_{i=k+1}^{n} \Pi_i \nabla\theta^t_i$
Update $\mathcal{P}^*_g \leftarrow \left\{\mu_g = \frac{1}{r}\sum_r \hat{\mu}_r, \Sigma_g = \frac{1}{r}\sum_r \hat{\Sigma}_r\right\}$
Share $\nabla\theta^{t+1}_g, \mathcal{P}^*_g$ with all clients
return $\mathcal{A}_g \leftarrow$ `Test`$(G^{t+1}_{\theta_g}, \mathcal{X}_{\text{test}})$

Lemma 5.1 (Computational and communication complexity of TriGuard-FL)**.** *The overall computational complexity of TriGuard-FL, considering both client and server side operations, is $\mathcal{O}(nd^3)$, where n denotes the number of clients and d the latent-space dimensionality. This cost accounts for latent embedding extraction, Mahalanobis-based auditing, masked distribution estimation, Gaussian* MMD*-based trust scoring, and server side AGI mitigation. Moreover, TriGuard-FL offers a computational advantage over prior valuation-based methods such as [23], which exhibit a higher complexity of $\mathcal{O}(2\mathcal{N}_k d^3 \log(2d))$, where $\mathcal{N}_k$ denotes the number of local samples per client. During each communication round, the server and clients exchange model updates, masked moment statistics, and scalar trust scores, resulting in an overhead comparable to existing valuation-based frameworks [18, 23]. Each training round requires approximately 4.3 GB of memory and $\approx$ 600 seconds, including mitigation overhead on modern GPUs (NVIDIA RTX 6000 Ada).*

Algorithm 2 Proposed auditable data valuation (`Val`)

Input: Local data $\mathcal{D}_k$, Global reference $\mathcal{P}^*_g = (\mu_g, \Sigma_g)$
Output: TriGuard-FL audited data $\mathcal{D}'_k$, local data density parameters $\mathcal{P}_k = (\mu_k, \Sigma_k)$, trust score Π_k

for x in $\mathcal{D}_k$ **do**
  $\Phi_{\theta_k}(x) \leftarrow f_{\theta_k}(x, y)$
  $dist(\Phi(x), \mathcal{P}^*_g) \leftarrow (\Phi(x) - \mu_g)^T \Sigma_g^{-1} (\Phi(x) - \mu_g)$ ▷ `Mahalanobis distance computation`
  if $dist(\Phi(x)) < \gamma$ **then** ▷ `TriGuard-FL ADV`
    $\mathcal{D}'_k \leftarrow (x, y)$
$\mathcal{E}_k = \{\Phi_{\theta_k}(x) \mid x \in \mathcal{D}_k\}$
$\mu_k = \frac{1}{|\mathcal{E}_k|}\sum_{e\in\mathcal{E}_k} e$
$\Sigma_k = \frac{1}{|\mathcal{E}_k|}\sum_{e\in\mathcal{E}_k} (e - \mu_k)(e - \mu_k)^\top$
$\mathcal{P}_k = (\mu_k, \Sigma_k)$
$\Pi(\mathcal{P}_k) = \Pi_k = \text{MMD}^2(\mathcal{P}_k, \mathcal{P}^*_g)$ ▷ `data distribution valuation`
return $\mathcal{D}'_k, \mathcal{P}_k, \Pi_k$

Algorithm 3 Proposed Antidote Gradient Injection (`AGI`)

Input: Client local model updates $\{\nabla\theta^t_k\}_{k\in\mathcal{F}_t}$, antidote gradient direction g_t
Output: Mitigated updates $\{\nabla\theta'^t_k\}_{k\in\mathcal{F}_t}$

$\alpha_i = \alpha\,(1 - \Pi_k)^\mu \in [0, 1]$,
$\nabla\theta'^t_i = \nabla\theta^t_i - \alpha_i \frac{\langle g_t, \nabla\theta^t_i\rangle}{\|g_t\|^2 + \epsilon} g_t$
return $\{\nabla\theta'^t_k\}_{k\in\mathcal{F}_t}$

Proof. The computational complexity of TriGuard-FL stems from its three major modules executed per communication round: (i) client-side auditability and verifiability stages, (ii) trust score computation and communication, and (iii) server-side mitigation via Antidote Gradient Injection (AGI).

1. *Latent embedding extraction:* Each client computes latent-space representations $\Phi(\mathcal{X}_k)$ for $\mathcal{N}_k$ samples, with a per-sample cost of $\mathcal{O}(d)$, resulting in a total cost of $\mathcal{O}(\mathcal{N}_k d)$.
2. *Proactive auditing via Mahalanobis distance:* The Mahalanobis distance between each embedded sample and the global reference (μ_g, Σ_g) is computed in $\mathcal{O}(d^2)$ per sample, leading to $\mathcal{O}(\mathcal{N}_k d^2)$ overall for filtering.

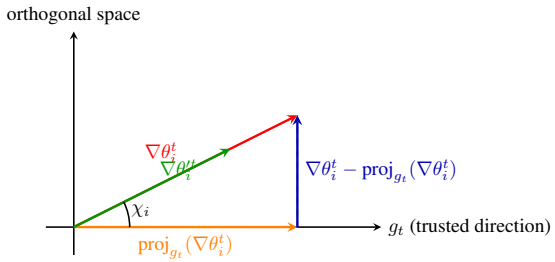


Figure 1. Intuition of Antidote Gradient Injection (AGI). The red arrow represents the original client update $\nabla\theta_i^t$, which may contain both benign and adversarial components. It is decomposed into (i) an orange projection along the trusted global direction g_t (potentially harmful) and (ii) a blue orthogonal component representing the benign learning signal. AGI scales and subtracts the projected component (weighted by α_i) to produce the green arrow, corresponding to the mitigated update $\nabla\theta_i'^t$ that neutralizes adversarial alignment while retaining useful gradients. Here, $\chi_i = \angle(\nabla\theta_i^t,\ g_t)$ denotes the angular deviation between the client's gradient and the trusted direction.

3. *Distribution estimation and masking:* Each client computes its local mean and covariance (μ_k, Σ_k), requiring $\mathcal{O}(\mathcal{N}_k d^2)$ for the covariance matrix and negligible overhead for Gaussian noise masking during privacy-preserving transmission.
4. *Trust score computation (*MMD*):* The squared MMD between two Gaussian distributions is evaluated in closed form, with a computational cost of $\mathcal{O}(d^3)$ due to the matrix square root and kernel evaluations.
5. *Server-side Antidote Gradient Injection (AGI):* For each low-trust client $i \in \mathcal{F}_t$, AGI neutralizes the adversarial projection of $\nabla\theta_i^t$ along the trusted direction g_t. This operation involves vector inner products and norm computations, each $\mathcal{O}(d)$, for $k = |\mathcal{F}_t| << n$ clients, resulting in $\mathcal{O}(kd)$. Since $k \ll n$ in practice, this term remains subdominant compared to $\mathcal{O}(nd^3)$ and scales linearly with the number of mitigated clients.

Summing across modules, the per-client complexity is $\mathcal{O}(\mathcal{N}_k d + \mathcal{N}_k d^2 + d^3) = \mathcal{O}(\mathcal{N}_k d^2 + d^3)$. Given that $\mathcal{N}_k d^2$ grows linearly with the number of samples and quadratically with the latent dimension, and $\mathcal{O}(nd^3)$ grows linearly with the number of clients and cubically in d, the latter dominates for moderate to large d and non-trivial n (e.g., $n \in [5, 100]$ covering both cross-silo and cross-device FL settings). Thus, asymptotically, the total time complexity of TriGuard-FL, including both client and server side operations and the AGI mitigation stage, is $\mathcal{O}(nd^3)$. The communication complexity remains comparable to existing valuation-based frameworks [18, 23], since each round involves the exchange of model updates, masked distribution statistics, and scalar trust scores only.

5.1. Communication, Computational Cost, and Overhead Analysis

For each round, each client transmits a model update $\nabla\theta_k^t \in \mathbb{R}^d$, masked mean $\hat{\mu}_k \in \mathbb{R}^d$, covariance $\hat{\Sigma}_k \in \mathbb{R}^{d\times d}$, and a scalar trust score. This incurs $\mathcal{O}(d^2)$ communication, consistent with existing valuation-based FL methods [18, 23].

Table 3 shows that TriGuard-FL maintains computational efficiency, with an average GPU RAM usage of approximately 4.3 GB and execution time around 600 seconds, well within the range of other robust FL defenses such as FedAdam [32] (380 s), FedMedian [44] (480s), and Krum [4] (500s), and notably lower than FLTrust [5] (650s), FLShield [17] (630s), and FedBary [23] (710s). This efficiency stems from TriGuard-FL's modular design, where the data filtering and distribution valuation components operate independently and can be executed in parallel. Additionally, client selection incurs no runtime cost, as trust scores are precomputed via closed-form Gaussian MMD, and aggregation proceeds identically to FedAvg [26]. Communication overhead remains low, as clients transmit only model updates, masked means and covariances, and a scalar trust score. Overall, TriGuard-FL achieves strong robustness and verifiability with minimal added overhead, demonstrating practical scalability for real-world FL deployments.

Table 3. Computation cost comparison of TriGuard-FL. **No Val** refers to standard FL without data valuation, and **Full defense** represents an ideal FL setup with perfect adversarial update removal.

FL Framework	GPU RAM usage (GB)	Execution time (seconds)
`No Val` (FedAvg) [26]	≈3.8	≈300
FedProx [22]	≈3.9	≈410
FedAdam [32]	≈4.5	≈380
FedAdagrad [32]	≈4.7	≈620
FedMedian [44]	≈4.5	≈480
Krum [4]	≈6.9	≈500
FLTrust [5]	≈8.4	≈650
DivFL [2]	≈5.2	≈680
FLShield [17]	≈5.4	≈630
FedBary [23]	≈5.2	≈710
TriGuard-FL (ours)	≈4.3	≈600

5.2. Extended Theoretical Analysis

Theorem 5.2 (TriGuard-FL convergence via ADV, VCC, and MPU). *Assume that (i) the local cross-entropy loss $\mathcal{L}_{CE}$ is β-Lipschitz, (ii) the local model f_θ is α-Lipschitz, and (iii) the empirical* MMD^2 *estimator satisfies uniform convergence [12, 43]. Then, the expected global loss after trust-weighted and AGI-mitigated aggregation satisfies the bound:* $\mathbb{E}_{\mathcal{D}'}[\mathcal{L}_{CE}(G_{\theta_g}^{t+1})] \leq \mathbb{E}_{\mathcal{D}}[\mathcal{L}_{CE}(G_{\theta_g}^{t})] + \alpha\beta\,\mathbb{E}\big[\text{MMD}^2(\mathcal{P}_k, \mathcal{P}_g^*)\big] - \mathbb{E}\Big[\alpha_k \frac{\langle g_t, \nabla\theta_k^t\rangle^2}{\|g_t\|^2+\varepsilon}\Big]$. *The additional*

negative term introduced by AGI captures the neutralization effect on adversarial gradient components, reducing their first-order influence on the global loss. Consequently, TriGuard-FL provides a monotonic descent condition of the expected loss under bounded adversarial influence, rather than a global convergence-rate guarantee. The MMD*-based trust weighting prioritizes clients with data distributions that are closer to the global reference, while AGI adaptively suppresses harmful gradient projections for effective mitigability.*

Proof. We provide the convergence analysis of TriGuard-FL under the unified audit-verify-mitigate framework. Given that $G^t_{\theta_g}$ denote the global model at round t and each client $\mathcal{C}_k$ performs latent-space auditing to obtain a filtered dataset $\mathcal{D}'_k$ and estimates its local distribution $\mathcal{P}_k = (\mu_k, \Sigma_k)$. The global reference $\mathcal{P}^*_g = (\mu_g, \Sigma_g)$ is constructed from masked client moments. The divergence $\Pi_k \equiv \Pi(\mathcal{P}_k)$ quantifies the distributional discrepancy between $\mathcal{P}_k$ and $\mathcal{P}^*_g$, computed via the closed-form squared Gaussian MMD^2. Then the server updates the global model using the trust-triad weighted rule (as described in the main paper)

$$\nabla\theta_g^{t+1} = \sum_{i\in\mathcal{F}_t} \frac{\Pi_i}{\sum_{j\in\mathcal{F}_t}\Pi_j}\nabla\theta_i'^t + \sum_{i\in\mathcal{H}_t} \frac{\Pi_i}{\sum_{j\in\mathcal{H}_t}\Pi_j}\nabla\theta_i^t, \tag{3}$$

where $\mathcal{H}_t$ and $\mathcal{F}_t$ denote the high-trust and low-trust client sets, respectively. For each low-trust client $i \in \mathcal{F}_t$, the Antidote Gradient Injection (AGI) step computes

$$\nabla\theta_i'^t = \nabla\theta_i^t - \alpha_i \frac{\langle g_t, \nabla\theta_i^t\rangle}{\|g_t\|^2+\varepsilon} g_t, \tag{4}$$

with $\alpha_i = \alpha(1-\Pi_i)^\mu \in [0,1]$ a trust-scaled correction factor, g_t the trusted direction (average of high-trust updates), and $\varepsilon > 0$ a stabilizer.

Smoothness bound. Assume that the local loss $\mathcal{L}_{CE}$ is β-Lipschitz and the local model f_θ is α-Lipschitz. For an L-smooth function, we have

$$\mathcal{L}_{CE}(G^{t+1}_{\theta_g}) \le \mathcal{L}_{CE}(G^t_{\theta_g}) + \left\langle \nabla\mathcal{L}_{CE}(G^t_{\theta_g}), \nabla\theta_g^{t+1}\right\rangle + \frac{L}{2}\left\|\nabla\theta_g^{t+1}\right\|^2. \tag{5}$$

Taking expectations over the client data and substituting Eq. (3), the second term expands into weighted client contributions.

Decomposition of client terms. For high-trust clients $i \in \mathcal{H}_t$, $\nabla\theta_i'^t = \nabla\theta_i^t$. For low-trust clients $i \in \mathcal{F}_t$, using Eq. (4),

$$\left\langle \nabla\mathcal{L}_{CE}(G^t_{\theta_g}), \nabla\theta_i'^t\right\rangle = \left\langle \nabla\mathcal{L}_{CE}(G^t_{\theta_g}), \nabla\theta_i^t\right\rangle - \alpha_i \frac{\langle g_t, \nabla\theta_i^t\rangle}{\|g_t\|^2+\varepsilon}\left\langle \nabla\mathcal{L}_{CE}(G^t_{\theta_g}), g_t\right\rangle. \tag{6}$$

Because g_t aligns with the benign descent direction, $\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), g_t\rangle \ge 0$, and thus the second term is non-positive. By Cauchy-Schwarz [3, 36], $\langle g_t, \nabla\theta_i^t\rangle\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), g_t\rangle \ge \frac{\langle g_t, \nabla\theta_i^t\rangle^2}{\|g_t\|^2+\varepsilon}$, which yields a mitigation gain proportional to $\alpha_i \frac{\langle g_t, \nabla\theta_i^t\rangle^2}{\|g_t\|^2+\varepsilon}$.

Bounding gradient mismatch by distributional divergence. Under the Lipschitz assumptions, the expected gradient mismatch between a client trained on $\mathcal{P}_k$ and the global reference $\mathcal{P}^*_g$ is bounded by

$$\left|\mathbb{E}\left[\left\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), \nabla\theta_k^t\right\rangle\right]\right| \le \alpha\beta\,\mathbb{E}\left[\mathrm{MMD}^2(\mathcal{P}_k, \mathcal{P}^*_g)\right]. \tag{7}$$

This follows from uniform convergence of the empirical MMD estimator, ensuring that MMD^2 captures the true distributional divergence with high probability. Recall the L-smoothness bound in Eq. (5) and the trust-triad aggregation in Eq. (3). Taking expectation over the (filtered) client data distribution $\mathcal{D}'$ and substituting Eqs. (3), (4), we obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{D}'}\left[\mathcal{L}_{CE}\left(G^{t+1}_{\theta_g}\right)\right] \le\ & \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}_{CE}\left(G^t_{\theta_g}\right)\right] + \mathbb{E}\left[\sum_{i\in\mathcal{H}_t}\omega_i^{(\mathcal{H})}\mathrm{L}\right] \\ &+ \mathbb{E}\left[\sum_{i\in\mathcal{F}_t}\omega_i^{(\mathcal{F})}\left(\mathrm{L} - \alpha_i\frac{\langle g_t, \nabla\theta_i^t\rangle}{\|g_t\|^2+\varepsilon}\mathrm{L}\right)\right] \\ &+ \tfrac{L}{2}\mathbb{E}\left[\left\|\nabla\theta_g^{t+1}\right\|^2\right],\end{aligned} \tag{8}$$

where $\omega_i^{(\mathcal{H})} = \frac{\Pi_i}{\sum_{j\in\mathcal{H}_t}\Pi_j}$ and $\omega_i^{(\mathcal{F})} = \frac{\Pi_i}{\sum_{j\in\mathcal{F}_t}\Pi_j}$ are nonnegative trust weights that sum to 1 within their sets. Here, $\mathrm{L} = \left\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), \nabla\theta_i^t\right\rangle$.

Bounding the inner products by distributional divergence. By the Lipschitz assumptions and the uniform convergence of the empirical MMD^2 estimator, the expected alignment between the global loss gradient and each client update admits

$$\left|\mathbb{E}\left[\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), \nabla\theta_k^t\rangle\right]\right| \le \alpha\beta\,\mathbb{E}\left[\mathrm{MMD}^2(\mathcal{P}_k, \mathcal{P}^*_g)\right]. \tag{9}$$

Applying (9) to both the $\mathcal{H}_t$ and $\mathcal{F}_t$ sums in (8), and using the convexity of expectation with respect to the trust weights $\omega_i^{(\mathcal{H})}$ and $\omega_i^{(\mathcal{F})}$ (which form convex combinations within each set), yield the aggregate positive contribution

$$\alpha\beta\,\mathbb{E}\left[\mathrm{MMD}^2(\mathcal{P}_k, \mathcal{P}^*_g)\right]. \tag{10}$$

Lower bounding the AGI gain. For $i \in \mathcal{F}_t$, the AGI subtraction term in (8) is $\alpha_i\frac{\langle g_t, \nabla\theta_i^t\rangle}{\|g_t\|^2+\varepsilon}\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), g_t\rangle$. Since g_t is constructed from high-trust updates, it aligns with the benign descent direction, hence $\langle\nabla\mathcal{L}_{CE}(G^t_{\theta_g}), g_t\rangle \ge 0$. By

Cauchy-Schwarz and normalization by $\|g_t\|^2+\varepsilon$ we obtain the deterministic bound

$$\alpha_i \frac{\langle g_t, \nabla\theta_i^t\rangle}{\|g_t\|^2+\varepsilon}\langle \nabla\mathcal{L}_{CE}(G_{\theta_g}^t),\, g_t\rangle \;\geq\; \alpha_i \frac{\langle g_t, \nabla\theta_i^t\rangle^2}{\|g_t\|^2+\varepsilon}. \quad (11)$$

Hence the AGI contribution in (8) yields a *negative* term $-\mathbb{E}\big[\alpha_i \frac{\langle g_t,\nabla\theta_i^t\rangle^2}{\|g_t\|^2+\varepsilon}\big]$.

Controlling the quadratic smoothness term. Under standard FL step-size control (bounded local epochs and learning rates), the smoothness remainder $\frac{L}{2}\,\mathbb{E}\big[\|\nabla\theta_g^{t+1}\|^2\big]$ can be upper-bounded by a constant that is absorbed into the one-step inequality or made arbitrarily small by tuning (this is the usual descent lemma setting in first-order analyses). Dropping this nonnegative remainder only strengthens the inequality.

Final bound. Combining (8), (10), and (11), and renaming α_i by α_k to match the theorem statement, we conclude

$$\begin{aligned}\mathbb{E}_{\mathcal{D}'}\Big[\mathcal{L}_{CE}\Big(G_{\theta_g}^{t+1}\Big)\Big] \;&\leq \mathbb{E}_{\mathcal{D}}\Big[\mathcal{L}_{CE}\Big(G_{\theta_g}^{t}\Big)\Big] \\ &+\; \alpha\beta\,\mathbb{E}\big[\mathrm{MMD}^2(\mathcal{P}_k,\mathcal{P}_g^*)\big] \\ &-\; \mathbb{E}\left[\alpha_k\,\frac{\langle g_t,\nabla\theta_k^t\rangle^2}{\|g_t\|^2+\varepsilon}\right].\end{aligned} \quad (12)$$

This analysis demonstrates that the convergence of TriGuard-FL is jointly governed by distributional alignment and trust-scaled mitigation. As training progresses, clients with higher distributional divergence $\mathrm{MMD}^2(\mathcal{P}_k,\mathcal{P}_g^*)$ receive lower aggregation weights, reducing the influence of poisoned or low-quality updates, while well-aligned clients are amplified through trust weighting. The additional AGI mitigation term further suppresses adversarial gradients by neutralizing their harmful projections along the trusted direction g_t. Consequently, TriGuard-FL achieves stable and monotonic convergence toward a low-risk global optimum, ensuring unified auditability, verifiability, and mitigability under bounded adversarial participation.

Proposition 5.3 (TriGuard-FL verifiable and mitigable client contribution). *Let $\mathcal{N}=\{\nabla\theta_1^t,\ldots,\nabla\theta_n^t\}$ denote the set of all client updates received by the server at communication round t. Define $\mathcal{H}=\{\nabla\theta_k^t \mid \Pi(\mathcal{P}_k)\leq\varphi\}$ as the subset of* verifiable *updates from clients (as defined in step 1 of server-side modifications). Let $\mathcal{M}=\{\nabla\theta_k^t \mid \Pi(\mathcal{P}_k)>\varphi\}$ represent updates from low-trust or adversarial clients, where $m=|\mathcal{M}|<n$. The sets satisfy $\mathcal{H}\cap\mathcal{M}=\emptyset$ and $\mathcal{H}\cup\mathcal{M}=\mathcal{N}$. During the aggregation phase, TriGuard-FL includes trust-aware AGI mitigated updates (Eq.* **??***). For any client update $\nabla\theta_k\in\mathcal{N}$ to influence the global model, the following conditions must hold:*

$$\sum_{\nabla\theta_k\in(\mathcal{N}\cap\mathcal{H})}\mathcal{L}_{CE}(\mathcal{D}_t,\nabla\theta_k)\leq\sum_{\nabla\theta_k\in\mathcal{N}}\mathcal{L}_{CE}(\mathcal{D}_t,\nabla\theta_k), \quad (13)$$

$$\left\|\sum_{\nabla\theta_k\in\mathcal{N}\cap\mathcal{M}}\mathcal{L}_{CE}(\mathcal{D}_t,\nabla\theta_k)-\sum_{\nabla\theta_k\in\mathcal{N}\cap\mathcal{H}}\mathcal{L}_{CE}(\mathcal{D}_t,\nabla\theta_k)\right\|\geq\xi, \quad (14)$$

for some margin $\xi\geq 0$. Here, $\mathcal{L}_{CE}(\mathcal{D}_t,\nabla\theta_k)$ denotes the test loss when incorporating client k's update. The first inequality ensures that inclusion of verified updates does not increase the aggregate test loss, while the second enforces a minimum separation in performance between benign and adversarial updates. Equality holds when $|\mathcal{H}|=n$ (all clients are verified) and $m=0$ (no adversarial clients). Overall, it provides a formal criterion for verifiable, mitigable contributions rather than a standalone convergence.

Proof. We aim to establish that the TriGuard-FL aggregation rule only includes verifiable updates from clients whose filtered local distributions are consistent with the global reference, leading to lower aggregate loss and exclusion of malicious updates.

(1) Gradient approximation and consistency. According to Balakrishnan *et al.* [2], when selecting a subset $\mathcal{S}_t$ of client updates that approximate the full aggregation, the gradient deviation is bounded as

$$\left\|\sum_{k\in\mathcal{S}_t}\gamma_k\nabla F_k-\sum_{k\in\mathcal{N}}\nabla F_k\right\|\leq n\varrho,$$

where ϱ is the maximum approximation error, and γ_k are aggregation weights. Furthermore, we extend this observation to test loss, demonstrating that a subset of client updates $\mathcal{H}$, verified through TriGuard-FL, are effectively trained on data evaluated and validated by the TriGuard-FL framework. We use $(\mathcal{D}_\mathcal{T})=(\mathcal{D}_t,\nabla\theta_k)$ to save space and maintain clarity. It is given as

$$\begin{aligned}\left\|\sum_{k\in\mathcal{S}^t}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T})-\sum_{k\in n}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T})\right\|\leq n\varrho,\\ \left\|\sum_{\nabla\theta_k\in(\mathcal{N}\cap\mathcal{H})}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T})-\sum_{\nabla\theta_k\in\mathcal{N}}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T})\right\|\leq n\varrho.\end{aligned} \quad (15)$$

Here, $\mathcal{N}\cap\mathcal{H}$ denote the subset of verified clients updates obtained after training models using TriGuard-FL valuated data whose test loss is lower than that of the remaining clients.

$$\begin{aligned}\sum_{\nabla\theta_k\in(\mathcal{N}\cap\mathcal{H})}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T})\leq n\varrho\sum_{\nabla\theta_k\in\mathcal{N}}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T}),\\ \sum_{\nabla\theta_k\in(\mathcal{N}\cap\mathcal{H})}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T})\leq\sum_{\nabla\theta_k\in\mathcal{N}}\mathcal{L}_{CE}(\mathcal{D}_\mathcal{T}).\end{aligned} \quad (16)$$

Here $=$ holds true when $|\mathcal{H}|=n$. This proves the first inequality Eq. **??** of Theorem **??**.

(2) Loss-based separation. Following prior theoretical results in data poisoning (**Theorem 2.** of [14]), updates generated from poisoned data typically yield higher test losses:

$$\mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) < \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\hat{\theta}_k), \quad \forall \nabla\hat{\theta}_k \in \mathcal{F}, \textbf{ i.e.,}$$

$$\begin{aligned} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_1) &< \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\hat{\theta}_1), \\ &\dots \\ \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_{n-m}) &< \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\hat{\theta}_m), \end{aligned} \tag{17}$$

By summing over all benign updates and comparing with the full set, we obtain:

$$\sum_{\nabla\theta_k \in (\mathcal{N}\setminus\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) < \sum_{\nabla\theta_k \in \mathcal{F}} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}),$$

rewriting it

$$\sum_{\nabla\theta_k \in (\mathcal{N}\setminus\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) < \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}), \tag{18}$$

Adding terms Eq 16 and Eq 18 and rearranging we get

$$\sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{H})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) \leq \sum_{\nabla\theta_k \in \mathcal{N}} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) - \tag{19}$$

$$\sum_{\nabla\theta_k \in (\mathcal{N}\setminus\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) + \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}). \tag{20}$$

$$\sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{H})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) - \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) \leq \tag{21}$$

$$\sum_{\nabla\theta_k \in \mathcal{N}} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) - \sum_{\nabla\theta_k \in (\mathcal{N}\setminus\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}). \tag{22}$$

$$\sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{H})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) - \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) \leq \xi. \tag{23}$$

$$\sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) - \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{H})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) \geq \xi. \tag{24}$$

$$\left\| \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{F})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) - \sum_{\nabla\theta_k \in (\mathcal{N}\cap\mathcal{H})} \mathcal{L}_{CE}(\mathcal{D}_{\mathcal{T}}) \right\| \geq \xi. \tag{25}$$

The above equation proves the second inequality Eq. 14 in Theorem **??**. These results demonstrate that our TriGuard-FL-integrated FL system ensures that only verifiable client contributions are aggregated through MMD based, auditable distribution-level data valuation.

6. Additional Details of Experiments and Ablation Study

6.1. Datasets, FL & Attack Setup, and Models

Table 4 summarizes the datasets and provides additional information to the main paper provided details. We cover both cross-device (CIFAR10 [19], CIFAR100 [19]) and cross-silo (HAM10000/ HAM10K [37], Census15 [30]) FL scenarios across diverse modalities, namely, image and tabular data under varying levels of non-IIDness modeled via Dirichlet distributions.

- **CIFAR10 [19] and CIFAR100 [19].** Both datasets are standard image classification benchmarks containing 60,000 color images, distributed across 10 and 100 categories, respectively. For CIFAR10, we adopt a ResNet18 backbone with an input resolution of 224×224, while CIFAR100 experiments use a deeper ResNet50 model under the same input configuration. Federated learning is simulated with 100 clients in total, of which 50 are adversarial. The latent feature dimensionality d is set to 512 for ResNet18 and 2048 for ResNet50, corresponding to the output size of their respective penultimate layers.
- **HAM10000 [37].** This dermoscopic image dataset contains 10,015 skin lesion samples categorized into seven diagnostic classes. Following the clinical grouping protocol in [40], lesions of actinic keratoses (akiec), melanoma (mel), and basal cell carcinoma (bcc) are considered *malignant*, while benign keratosis (bkl), dermatofibroma (df), melanocytic nevi (nv), and vascular skin lesions (vasc) are treated as *benign*. Due to the inherent variability and visual heterogeneity of dermoscopic imagery, we do not apply additional augmentation beyond the provided data splits from [40]. Experiments are conducted using the ResNet18 backbone with feature dimension d=512, under federated configurations of 5, 10, and 20 clients. We further evaluate both single- and multi-client adversarial scenarios with 1, 3, and 5 malicious participants.
- **Census15 [30].** The Census15 tabular dataset contains 32,561 training samples and 16,281 test samples with two target classes: *Income* $\leq$ 50*K* and *Income* $>$ *50K*. Each record consists of 15 tabular features, including age, education level, occupation type, and marital status, used for income-level prediction. Following a cross-silo FL setup, we evaluate configurations with 5, 10, and 20 clients, and simulate adversarial conditions with 1, 3, and 5 malicious clients. A lightweight two-layer multilayer perceptron (MLP) is employed, comprising an input layer with 15 nodes, a hidden layer of 128 neurons using ReLU activation, and an output layer with softmax classification. The penultimate feature representation has a latent dimension of d=128.

Furthermore, to assess robustness under varying threat

Table 4. Overview of dataset details used in TriGuard-FL, covering both cross-device and cross-silo scenarios. For each dataset, we report the total number of samples, train–test splits, number of classes, and per-class sample distributions. CIFAR10 and CIFAR100 correspond to cross-device FL settings, while HAM10000 and Census15 represent cross-silo FL configurations across medical image and financial tabular domains, respectively. `text`: class label or category.

Name, data FL type	#Samples			#Classes	Samples/class	
	Train	Test	Total		Train	Test
CIFAR10 [19] (image, **cross-device FL**)	50000	10000	60000	10	5000/class	1000/class
CIFAR100 [19] (image, **cross-device FL**)	50000	10000	60000	100	500/class	100/class
HAM10000 [37] (image, **cross-silo FL**)	10015	1512	11527	7	`AKIEC` - 327	`AKIEC` - 43
					`BCC` - 514	`BCC` - 93
					`BKL` - 1099	`BKL` - 217
					`DF` - 115	`DF` - 44
					`MEL` - 1113	`MEL` - 171
					`NV` - 6705	`NV` - 909
					`VASC` - 142	`VASC` - 35
Census15 [30] (tabular, **cross-silo FL**)	48842	32561	16281	2	`Income` $\leq$ `50K` - 24720	`Income` $\leq$ `50K` - 12435
					`Income` $>$ `50K` - 7841	`Income` $>$ `50K` - 3846

intensities, we simulate three levels of poisoning rate, 30%, 50%, and 100% quantified as $\frac{\nu}{\mathcal{N}_k} \times 100$, where ν is the number of poisoned samples and $\mathcal{N}_k$ is the total number of local samples. We considered existing train and test data splits for each dataset with a batch size of 64 and a learning rate $\eta = 0.01$. *An example implementation is provided in the supplementary material, and the complete source code will be released publicly upon paper acceptance.*

6.2. Model Selection and FL Experiment Design Rationale

Model architectures for each dataset are chosen based on prior empirical success in the literature [17, 23], ensuring a fair and representative evaluation of our TriGuard-FL framework. Although transformer-based architectures might yield higher accuracy, we deliberately opt for more conventional models (CNN) to maintain experimental focus on the trustworthiness aspects, auditable data valuation, verifiable client updates, and mitigability of poisonous updates. **Importantly, TriGuard-FL remains model-agnostic, operating on latent representations and distributional characteristics that are architecture independent.** Our evaluation further spans both cross-device and cross-silo FL settings, covering a variety of conditions, including client heterogeneity, varying client scaling (5-100), different proportions of malicious participants (single vs. multi-client attacks), and three diverse poisoning strategies. This comprehensive setup enables a scalable and rigorous assessment of the framework's reliability and adaptability.

6.3. Software and Hardware Setup

All experiments are implemented in Python 3.10 using PyTorch for model development and training, along with NumPy and Pandas for data preprocessing and logging. The system simulates client-side local training and server-side aggregation and evaluation. Experiments were conducted on an **NVIDIA RTX 6000 Ada GPUs with 49GB of memory**, ensuring support for parallel processing and high-throughput data valuation during each FL round. All experiments are run three times, with a reported average and standard deviation.

6.4. Evaluation Metrics

In addition to standard global test accuracy ($\mathcal{A}_g$), global test accuracy under attack ($\mathcal{A}'_g$) metrics, and degradation of global test accuracy under attack ($\mathcal{D}_g$), as given in the main paper, we introduce three quantitative metrics to assess ADV, VCC, and MPU.

1. *Adversarial detection recall* (ADR), the proportion of malicious samples successfully identified and filtered and is given as $\mathrm{ADR} = \frac{\mathrm{TP}}{\mathrm{TP+FP+FN}}$, where TP, FP, and FN denote true positives, false positives, and false negatives, respectively, in filtering poisoned samples.
2. *Benign retention fidelity* (BRF), the fraction of benign samples incorrectly discarded $\mathrm{BRF} = \frac{\mathrm{FP}}{\mathrm{FP+TN}}$,where TN denotes true negatives (benign samples correctly retained).
3. *Trust aggregation stability* (TAS), the average ratio of selected benign clients across rounds, measuring verifiability, $\mathrm{TAS} = \frac{1}{T}\sum_{t=1}^{T} \frac{|\mathbb{B}_t^{\mathrm{sel}}|}{|\mathbb{B}|}$, where T is the total number of communication rounds, $\mathbb{B}$ is the set of all benign clients, and $\mathbb{B}_t^{\mathrm{sel}}$ is the subset of benign clients selected in round t.

ADR, BRF, and TAS are mechanism-level metrics designed to quantify auditability, verifiability, and mitigability, respectively. Their computation depends on internal filtering behavior, trust-score dynamics, and aggregation stability, rather than dataset-specific semantics. As such, their qualitative behavior is consistent across datasets and attack types. We additionally evaluate ADR/BRF/TAS on CIFAR10, obtaining ADR = 0.91, BRF = 0.78, TAS = 0.93, consistent with the HAM10000 dataset reported in the main paper. Across both datasets, higher ADR/TAS and lower BRF align with reduced degradation and stable training, confirming that these metrics generalize beyond a single dataset and correlate with robustness. Although not reported as an explicit statistical correlation plot, the connection between these metrics and robustness is direct and empirically observable: (a) Higher ADR corresponds to more effective filtering of poisoned samples, which reduces adversarial influence at the data level. (b) Lower BRF indicates better preservation of benign samples, preventing unnecessary

utility loss. (c) Higher TAS reflects stable trust assignments and aggregation weights across rounds, which aligns with lower degradation and smoother convergence. This alignment is evident in our experiments: methods with higher ADR and TAS and lower BRF consistently achieve lower degradation under attack and higher clean accuracy, while competing methods show weaker metric values and inferior robustness.

Overall, these metrics are not intended to replace standard robustness measures but to provide interpretable diagnostics that explain why a method is robust. Together with the comprehensive accuracy and degradation results across multiple datasets and attacks, they strengthen the empirical grounding of the proposed trust-triad framework.

6.5. Baselines and Existing Defenses

To comprehensively evaluate the performance of TriGuard-FL, we compare it against both foundational FL frameworks and a diverse set of state-of-the-art defense mechanisms. First, we include *No data valuation* (`No Val` - FedAvg [26]), representing standard FL without any data valuation or defenses, used to assess the raw vulnerability to attacks. Next, we compare TriGuard-FL against nine representative defenses across four major categories, as outlined in Table 1. (a) *Robust aggregation optimizers:* FedProx [22], FedAdam [32], FedAdagrad [32], and FedMedian [44], (b) *Byzantine-resilient aggregation methods:* Krum [4] and FLTrust [5], (c) *Client selection strategy:* DivFL [2], and (d) *FL data/model valuation approaches:* FedBary [23] (data valuation) and FLShield [17] (model valuation). These methods are selected based on their relevance to our goals of auditability, verifiability, and robustness in federated settings.

Below are more details about baselines.

1. **FedProx** [22] enhances FedAvg by introducing a proximal term to improve convergence under system and statistical heterogeneity, offering better robustness and theoretical guarantees in non-IID FL settings.
2. **FedAdagrad** and **FedAdam** [32] adapt centralized adaptive optimizers to the FL setting, mitigating convergence issues and tuning sensitivity in FedAvg. They offer improved performance under non-IID data and non-convex objectives while enhancing communication efficiency.
3. **FedMedian** [44] enhances robustness to Byzantine failures by aggregating gradients using coordinate-wise medians. It ensures order-optimal error rates across convex and non-convex loss functions, making it suitable for secure FL under adversarial conditions.
4. **Krum** [4] is a Byzantine-resilient aggregation rule that selects updates with minimal distance to others, enabling convergence even when up to f out of n clients are adversarial. It outperforms linear methods under Byzantine threats.
5. **FLTrust** [5] improves Byzantine robustness in FL by using a trusted server-side dataset and model to evaluate client updates. Trust scores, based on alignment with the server model, are used to normalize and weight client contributions, mitigating malicious influence.
6. **DivFL** [2] selects a diverse subset of clients whose gradients best represent the full population, using a submodular facility location objective. This improves convergence, efficiency, and fairness over random selection in FL.
7. **FLShield** [17] secures FL by validating local models using benign participant data, eliminating the need for a clean server dataset. It defends against poisoning and backdoor attacks while preserving privacy, making it practical for real-world FL scenarios.
8. **FedBary** [23] provides a privacy-preserving approach for client contribution evaluation and data selection in FL. It uses Wasserstein distance for efficient, validation-free data valuation without depending on specific training algorithms.

6.6. Performance Evaluation of TriGuard-FL

We evaluate the effectiveness of TriGuard-FL against three major adversarial threat categories, PGD-based gradient poisoning [25], Distributed Backdoor Attack (DBA) [42], and targeted label-flip poisoning [34], across four benchmark datasets encompassing both cross-device (CIFAR10, CIFAR100) and cross-silo (HAM10000, Census15) FL settings. Tables 5, 6, and 7 present clean accuracy ($\mathcal{A}_g$) and degradation under attack ($\mathcal{D}_g{:=}100{-}\mathcal{A}'_g$) for IID ($\beta{=}5$) and non-IID ($\beta{=}0.3$) configurations, with a 50% poisoning rate. Subsequent ablation studies (Section 7) further analyze the impact of varying poisoning intensities and distribution heterogeneity.

6.6.1. Effectiveness under Benign Conditions

Even without attacks, TriGuard-FL improves utility by pairing auditable data valuation (ADV) with verifiable client contribution (VCC). On CIFAR10, clean accuracy rises from 92.11% (FedAvg) to 95.48%, and on HAM10000 we observe $>$ 5% gains over FedAdagrad (Table 5). ADV removes distributional outliers before local optimization, while VCC assigns higher weights to clients aligned with the global reference, preventing noisy or low-utility updates from diluting the descent direction. Although mitigability of poisonous updates (MPU) is inactive in benign conditions, its trust-scaled design (AGI) adds no penalty and preserves the clean-data advantage.

6.6.2. Robustness under Adversarial Clients

TriGuard-FL attains the lowest $\mathcal{D}_g$ across all attacks and datasets, evidencing strong mitigability. Under PGD (Table 5), degradation on CIFAR10 (non-IID) drops to

15.35%, surpassing FLShield by ≈9%. For DBA (Table 6), TriGuard-FL achieves **36.77%** $\mathcal{D}_g$ on CIFAR10 (IID) and **47.68%** on CIFAR100 (non-IID), yielding 8-15% improvements over the closest baselines. Against targeted label-flip (Table 7), degradation on Census15 (non-IID) is **19.97%**, outperforming FLShield/FedBary by 3-5%. While ADV filters many adversarial samples and VCC down-weights misaligned clients, adaptive AGI-based MPU is crucial for the residual threat surface, i.e., poisoned updates that evade filtering or appear statistically similar under masking. MPU projects low-trust updates onto a trusted descent direction and subtracts a trust-scaled component, neutralizing adversarial alignment without discarding entire updates. This preserves a benign signal embedded in mixed-quality gradients, yielding both lower $\mathcal{D}_g$ and faster recovery post-attack.

6.6.3. IID vs. Non-IID Stability

TriGuard-FL exhibits smaller IID, non-IID gaps than all baselines. VCC stabilizes aggregation by emphasizing clients whose masked Gaussians are close (in closed-form MMD^2) to the global reference, thereby curbing drift under heterogeneity. MPU further cushions non-IID regimes by removing adversarial projections conditioned on trust, which are more prevalent when client distributions are skewed. For example, in DBA on CIFAR100 (Table 6), TriGuard-FL increases degradation by only ∼5.5% from IID to non-IID (42.16% → 47.68%), whereas competing methods deteriorate by 10-15%.

*In summary, the gains are driven by the trust-triad: **ADV** prunes distributional outliers early, **VCC** certifies and weights reliable clients, and **MPU** (AGI) actively counters residual malicious directions while retaining benign signal. This layered design yields strong, clean accuracy and the lowest post-attack degradation across PGD, DBA, and label-flip threats, and remains stable under severe non-IID partitions.*

7. Ablation study

7.1. Effect of non-IID Heterogeneity

Figure 2 illustrates the variation in global test accuracy $\mathcal{A}_g$ of TriGuard-FL compared to representative baselines, FLTrust [5], FLShield [17], FedBary [23], and `No Val` across different degrees of non-IID data distributions on CIFAR10 and HAM10000. The Dirichlet parameter β controls heterogeneity, where smaller β (e.g., 0.1) indicates stronger non-IID skew and larger β (e.g., 10) approximates IID conditions. We focus on these representative baselines to emphasize key performance differences under both clean (left) and attacked (right) scenarios.

Across all heterogeneity levels, TriGuard-FL consistently achieves the highest accuracy, maintaining robustness under both benign and adversarial conditions. On CIFAR10 and HAM10K, TriGuard-FL preserves a significantly higher $\mathcal{A}_g$ as β decreases, whereas all baselines show sharp degradation under severe non-IID settings (β=0.1). In clean conditions (Figs. 2a, 2c), this stability stems from auditable data valuation (ADV), which prunes distributional outliers early, and verifiable client contribution (VCC), which prioritizes clients whose local distributions align with the global reference, reducing aggregation drift. Under attack (Figs. 2b, 2d), mitigability of poisonous updates (MPU) via Antidote Gradient Injection (AGI) further reinforces convergence by neutralizing adversarial components without discarding entire updates. This adaptive gradient correction allows TriGuard-FL to sustain accuracy levels above 90% on CIFAR10 and 75% on HAM10K even under strong heterogeneity, outperforming FLTrust and FLShield by 5–10% in most configurations.

Overall, these results confirm that TriGuard-FL's trust-triad, combining distribution-aware auditing, verifiable client scoring, and trust-scaled gradient mitigation, enables stable and fair optimization across heterogeneous, adversarial federated environments.

7.2. Ablation across Varying FL Configurations and Attack Settings

Table 9 presents an extensive ablation study evaluating the robustness of TriGuard-FL across different FL configurations, adversarial intensities, and poisoning strategies. Experiments include both PGD-based gradient poisoning and targeted label-flip attacks on two representative datasets: HAM10000 (medical imaging) and Census15 (financial tabular) under non-IID (β=0.3) conditions. We vary the number of total clients (n), number of malicious clients (m), and poisoning rates (30%–100%) to analyze resilience under both sparse and dense adversarial participation.

Across all configurations, TriGuard-FL consistently achieves the lowest degradation under attack ($\mathcal{D}_g \downarrow$), significantly outperforming all practical baselines. For instance, under low-client, high-maliciousness conditions on HAM10000 (n=5, m=1, 30% poisoning), TriGuard-FL achieves a degradation of **20.97%**, outperforming FLShield's 24.22% and approaching the oracle reference. As the attack intensity increases to 100% poisoning, other defenses deteriorate sharply, e.g., FLShield's degradation rises to 40.95%, while TriGuard-FL limits the degradation to just **28.27%**, demonstrating superior robustness. A similar trend is observed for Census15 under targeted label-flip attacks, where even at 100% poisoning, TriGuard-FL maintains a low degradation of **25.66%**, compared to FLShield's 28.71%. These results highlight TriGuard-FL's ability to preserve model performance even under extreme adversarial conditions.

The observed reduction in degradation stems from the combined effect of the three trust-triad components. First,

Table 5. Comprehensive evaluation of TriGuard-FL under PGD-based poisoning [25] on CIFAR10, CIFAR100, and HAM10000. Experiments are conducted in IID (β=5) and non-IID (β=0.3) FL settings with a 50% data poisoning rate. We report clean accuracy $\mathcal{A}_g \uparrow$ and degradation accuracy under attack $\mathcal{D}_g := 100 - \mathcal{A}'_g \downarrow$. **Bold** and second denote the best and second-best results per column, respectively.

Dataset →	**CIFAR10** [19]				**CIFAR100**[19]				**HAM10000**[37]			
FL setting →	n=100, m=50				n=100, m=50				n=20, m=5			
Method ↓	**IID**		**non-IID**		**IID**		**non-IID**		**IID**		**non-IID**	
	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$
`No Val`	92.11±0.25	28.06±1.67	91.25±1.91	29.74±0.84	82.91±1.27	48.76±0.58	80.52±1.48	50.14±0.68	89.93±1.62	35.48±1.23	86.90±1.22	37.96±0.75
FedProx [22]	91.53±0.78	30.59±0.83	90.67±1.27	34.30±1.25	82.13±0.49	50.21±1.36	79.74±1.83	52.62±0.89	89.35±1.44	36.02±1.31	86.32±1.71	38.54±0.83
FedAdam [32]	90.13±1.91	32.12±1.14	89.27±0.44	34.12±0.72	81.02±1.81	49.62±0.70	78.63±0.54	52.14±0.44	87.95±0.37	34.96±0.55	84.92±0.57	39.14±1.11
FedAdgrad [32]	92.98±0.62	28.52±0.12	92.12±0.73	29.52±0.41	83.75±0.32	48.32±0.83	81.36±0.95	48.99±0.32	90.80±1.99	34.78±0.83	87.77±1.95	37.09±0.36
FedMedian [45]	90.45±1.10	26.05±1.92	87.52±0.30	27.74±1.98	78.23±0.97	48.80±1.50	75.84±1.37	50.72±1.15	88.27±0.78	36.88±1.62	85.24±0.34	39.62±0.58
Krum [4]	90.87±0.43	26.54±0.65	86.59±0.83	28.16±0.19	75.92±1.11	46.51±1.16	72.54±0.77	48.25±1.82	88.69±1.15	34.66±0.24	85.66±1.09	37.46±1.84
FLTrust [5]	91.34±1.87	24.11±1.39	88.65±1.54	25.75±0.88	76.63±0.55	44.68±0.44	73.25±1.66	46.58±0.58	90.16±0.92	34.12±1.44	86.13±0.80	37.02±0.42
DivFL [2]	89.35±0.59	33.14±0.20	86.80±0.68	35.79±0.47	74.74±1.99	56.15±0.53	71.36±0.45	58.90±1.67	85.37±1.53	42.55±0.39	82.20±1.38	46.68±0.69
FLShield [17]	89.31±1.45	21.27±1.08	87.75±1.06	25.73±0.66	78.85±0.88	44.27±0.67	75.47±1.25	47.03±0.76	86.34±1.11	33.62±1.01	84.05±0.62	36.15±1.02
FedBary [23]	88.40±0.96	24.54±0.73	85.31±1.17	27.14±0.54	76.96±0.25	46.88±0.29	73.58±1.04	49.74±0.51	83.10±0.48	38.33±1.58	80.75±1.27	40.12±1.56
TriGuard-FL (ours)	**95.48**±2.36	**14.86**±1.51	**94.73**±1.28	**15.35**±0.91	**89.32**±1.31	**38.73**±1.97	**86.11**±1.19	**42.07**±1.21	**94.08**±1.14	**22.99**±1.18	**93.86**±1.78	**25.43**±1.17

Table 6. Comprehensive evaluation of TriGuard-FL under DBA attack [42] on CIFAR10, CIFAR100, and HAM10000. Experiments are conducted in IID (β=5) and non-IID (β=0.3) FL settings with a 50% data poisoning rate. We report clean accuracy $\mathcal{A}_g \uparrow$ and degradation accuracy under attack $\mathcal{D}_g := 100 - \mathcal{A}'_g \downarrow$. **Bold** and second denote the best and second-best results per column, respectively.

Dataset →	**CIFAR10** [19]				**CIFAR100** [19]				**HAM10000** [37]			
FL setting →	$n = 100$, $m = 50$				$n = 100$, $m = 50$				$n = 20$, $m = 5$			
Method ↓	**IID**		**non-IID**		**IID**		**non-IID**		**IID**		**non-IID**	
	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$
`No Val`	92.11±0.25	52.02±0.63	91.25±1.91	53.20±1.13	82.91±1.27	62.65±1.15	80.52±1.48	63.68±1.48	89.93±1.62	55.64±0.91	86.90±1.22	58.36±1.34
FedProx [22]	91.53±0.78	56.26±0.89	90.67±1.27	57.70±1.71	82.13±0.49	66.57±1.45	79.74±1.83	68.38±1.84	89.35±1.44	59.26±1.46	86.32±1.71	61.48±1.91
FedAdam [32]	90.13±1.91	57.97±1.71	89.27±0.44	59.18±0.27	81.02±1.81	67.46±0.61	78.63±0.54	69.57±0.75	87.95±0.37	60.13±0.77	84.92±0.57	62.76±1.26
FedAdagrad [32]	92.98±0.62	51.13±1.33	92.12±0.73	52.58±1.69	83.75±0.32	61.14±1.30	81.36±0.95	63.09±1.17	90.80±1.99	58.16±1.65	87.77±1.95	54.18±0.77
FedMedian [44]	90.45±1.10	49.04±0.92	87.52±0.30	50.80±0.84	78.23±0.97	60.25±1.18	75.84±1.37	62.23±0.38	88.27±0.78	51.75±0.52	85.24±0.34	53.02±0.66
Krum [4]	90.87±0.43	48.75±0.44	86.59±0.83	51.22±0.96	75.92±1.11	63.75±0.89	72.54±0.77	65.02±0.91	88.69±1.15	52.02±1.34	85.66±1.09	53.97±1.12
FLTrust [5]	91.34±1.87	46.36±1.06	88.65±1.54	48.81±1.22	76.63±0.55	60.90±1.96	73.25±1.66	63.96±1.76	90.16±0.92	50.12±0.36	86.13±0.80	51.47±0.91
DivFL [2]	89.35±0.59	56.65±0.79	86.80±0.68	58.53±0.52	74.74±1.99	71.47±0.41	71.36±0.45	73.72±0.29	85.37±1.53	58.58±1.89	82.20±1.38	60.75±0.84
FLShield [17]	89.31±1.45	45.11±1.57	87.75±1.06	47.60±0.75	78.85±0.88	57.22±0.94	75.47±1.25	59.63±1.50	86.34±1.11	49.97±0.83	84.05±0.62	50.92±1.39
FedBary [23]	88.40±0.96	54.99±0.58	85.31±1.17	56.63±1.03	76.96±0.25	64.81±1.53	73.58±1.04	68.11±0.82	83.10±0.48	52.69±1.18	80.75±1.27	55.57±0.52
TriGuard-FL (ours)	**95.48**±2.36	**34.57**±1.45	**94.73**±1.28	**36.77**±0.38	**89.32**±1.31	**42.16**±0.66	**86.11**±1.19	**47.68**±1.05	**94.08**±1.14	**40.89**±0.49	**93.86**±1.78	**44.52**±1.67

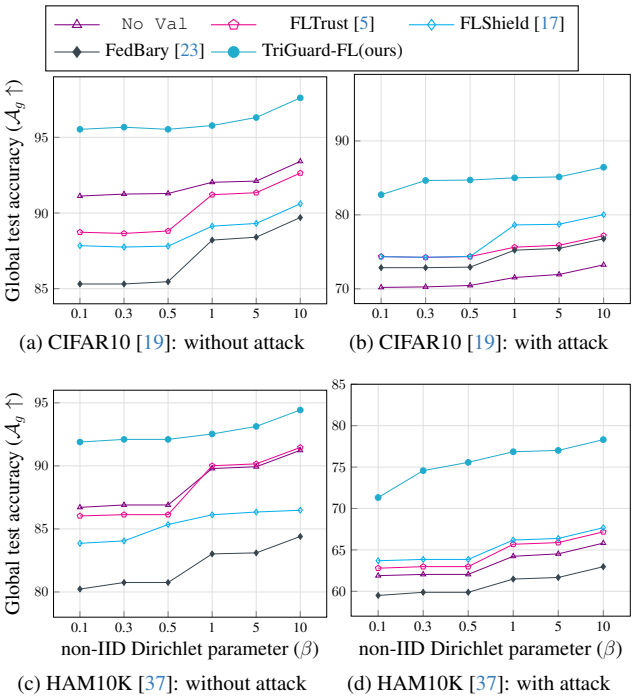
auditable data valuation (ADV) prunes distribution-level anomalies early, reducing the attack surface. Second, verifiable client contribution (VCC) ensures that clients with aligned latent distributions are weighted higher during aggregation, mitigating update drift in heterogeneous settings. Most importantly, the mitigability of poisonous updates (MPU), realized through **Antidote Gradient Injection (AGI)**, actively neutralizes harmful gradients by projecting suspicious updates onto a trusted descent direction (g_t) and subtracting a trust-scaled component proportional to the client's reliability. Unlike rejection-based defenses that discard low-trust updates (risking information loss), AGI selectively suppresses adversarial influence while retaining benign information, leading to lower $\mathcal{D}_g$ and improved convergence stability. Together, these mechanisms enable TriGuard-FL to sustain minimal accuracy degradation and strong robustness across varying client participation, poisoning rates, and attack types.

7.3. Client Scaling and Communication Efficiency

Table 8 evaluates the scalability of TriGuard-FL as the total client population increases from 100 to 1000 under both IID (β=5) and non-IID (β=0.3) conditions under a PGD attack and a poisoning rate of 50%. Across all configurations, both clean accuracy ($\mathcal{A}_g$) and degradation under attack ($\mathcal{D}_g$) remain highly stable as the federation scales, with only a 14% increase in convergence rounds and less than a 2% rise in $\mathcal{D}_g$ for both IID and non-IID regimes. This near-linear scaling trend demonstrates that TriGuard-FL maintains robustness and efficiency even when the client pool grows tenfold. The lightweight MMD-based trust scoring and verifiable client contribution (VCC) modules operate on low-dimensional masked distributional statistics, adding negligible communication overhead per round. Meanwhile, the AGI module is responsible for the mitigability of poisonous updates , which scales linearly with participating clients by applying trust-weighted vectorized projections, ensuring that gradient neutralization remains computationally ef-

Table 7. Comprehensive evaluation of TriGuard-FL under the targeted label flip attack attack scenario [34] across two datasets (cross-device: CIFAR10, cross-silo: Census15), evaluated in both IID ($\beta = 5$) and non-IID ($\beta = 0.3$) FL settings with a 50% poisoning rate. We report clean accuracy $\mathcal{A}_g \uparrow$ and degradation accuracy under attack $\mathcal{D}_g{:=}100{-}\mathcal{A}'_g \downarrow$. **Bold** and second denote the best and second-best results per column, respectively.

Dataset →	**CIFAR10** [19]				**Census15** [30]			
Setting →	$n = 100, m = 50$				$n = 20, m = 5$			
Method ↓	**IID**		**non-IID**		**IID**		**non-IID**	
	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$
`No Val`	92.11$_{\pm 0.25}$	26.95$_{\pm 0.81}$	91.25$_{\pm 1.91}$	28.44$_{\pm 1.09}$	87.73$_{\pm 0.95}$	25.62$_{\pm 1.07}$	86.46$_{\pm 0.93}$	27.44$_{\pm 1.28}$
FedProx [22]	91.53$_{\pm 0.78}$	19.17$_{\pm 1.08}$	90.67$_{\pm 1.27}$	20.69$_{\pm 1.60}$	86.50$_{\pm 1.02}$	28.25$_{\pm 1.46}$	85.23$_{\pm 0.64}$	29.87$_{\pm 1.76}$
FedAdam [32]	90.13$_{\pm 1.91}$	18.87$_{\pm 1.66}$	89.27$_{\pm 0.44}$	22.08$_{\pm 1.33}$	86.26$_{\pm 0.47}$	27.36$_{\pm 0.83}$	84.99$_{\pm 1.32}$	30.47$_{\pm 0.91}$
FedAdagrad [32]	92.98$_{\pm 0.62}$	19.87$_{\pm 0.77}$	92.12$_{\pm 0.73}$	21.58$_{\pm 0.91}$	87.29$_{\pm 1.71}$	25.49$_{\pm 1.09}$	86.02$_{\pm 0.87}$	27.63$_{\pm 1.03}$
FedMedian [44]	90.45$_{\pm 1.10}$	24.11$_{\pm 1.42}$	87.52$_{\pm 0.30}$	26.81$_{\pm 1.72}$	88.61$_{\pm 0.86}$	26.18$_{\pm 1.55}$	87.34$_{\pm 1.15}$	28.04$_{\pm 1.34}$
Krum [4]	90.87$_{\pm 0.43}$	23.62$_{\pm 1.21}$	86.59$_{\pm 0.83}$	25.75$_{\pm 0.74}$	87.65$_{\pm 1.18}$	27.51$_{\pm 1.03}$	86.38$_{\pm 0.78}$	29.62$_{\pm 0.72}$
FLTrust [5]	91.34$_{\pm 1.87}$	20.83$_{\pm 1.02}$	88.65$_{\pm 1.54}$	22.72$_{\pm 1.29}$	84.70$_{\pm 1.40}$	22.53$_{\pm 0.44}$	83.43$_{\pm 1.49}$	24.64$_{\pm 0.85}$
DivFL [2]	89.35$_{\pm 0.59}$	34.28$_{\pm 0.38}$	86.80$_{\pm 0.68}$	36.26$_{\pm 1.48}$	83.75$_{\pm 0.64}$	30.51$_{\pm 1.87}$	82.48$_{\pm 0.56}$	32.62$_{\pm 1.41}$
FLShield [17]	89.31$_{\pm 1.45}$	18.59$_{\pm 0.94}$	87.75$_{\pm 1.06}$	20.94$_{\pm 1.05}$	84.03$_{\pm 1.93}$	20.14$_{\pm 1.25}$	82.76$_{\pm 1.22}$	22.25$_{\pm 1.10}$
FedBary [23]	88.40$_{\pm 0.96}$	20.64$_{\pm 1.29}$	85.31$_{\pm 1.17}$	20.42$_{\pm 0.68}$	82.53$_{\pm 1.31}$	23.25$_{\pm 0.92}$	81.26$_{\pm 0.98}$	25.36$_{\pm 0.47}$
TriGuard-FL (ours)	**95.48**$_{\pm 2.36}$	**10.46**$_{\pm 0.63}$	**94.73**$_{\pm 1.28}$	**13.75**$_{\pm 1.23}$	**90.83**$_{\pm 0.78}$	**17.86**$_{\pm 1.48}$	**89.56**$_{\pm 1.84}$	**19.97**$_{\pm 1.68}$



(a) CIFAR10 [19]: without attack (b) CIFAR10 [19]: with attack

(c) HAM10K [37]: without attack (d) HAM10K [37]: with attack

Figure 2. Comparison of global accuracy ($\mathcal{A}_g$) of TriGuard-FL under varying degrees of non-IID with and without PGD attack across two datasets. We follow the same FL and attack settings of Table 5.

ficient at large scales. These design choices collectively enable TriGuard-FL to sustain strong adversarial resilience without compromising scalability. Notably, the clean accuracy decreases marginally from **95.48%** → **93.87%** (IID) and **94.73%** → **93.42%** (non-IID) as clients increase from 100 to 1000, while degradation rises slightly from **14.86%** → **16.38%** (IID) and **15.35%** → **16.61%** (non-IID). These minor variations confirm that TriGuard-FL preserves both model utility and robustness under large-scale federations, validating its practical deployability for real-world FL deployments involving hundreds to thousands of heterogeneous and potentially adversarial clients.

Table 8. Scalability and communication-round efficiency of TriGuard-FL on CIFAR10 under PGD attack (50% poisoning). We report average rounds to reach 90% of peak accuracy, and clean accuracy $\mathcal{A}_g \uparrow$ with degradation under attack $\mathcal{D}_g \downarrow$ for both IID (β=5) and non-IID (β=0.3) regimes. Results show near-linear scaling in client count and marginal robustness loss up to 1000 clients.

#Clients (n)	**Avg Rounds**	**IID** (β=5)		**non-IID** (β=0.3)	
		$\mathcal{A}_g$ (%) ↑	$\mathcal{D}_g$ (%) ↓	$\mathcal{A}_g$ (%) ↑	$\mathcal{D}_g$ (%) ↓
100	50	95.48$_{\pm 2.36}$	14.86$_{\pm 1.51}$	94.73$_{\pm 1.28}$	15.35$_{\pm 0.91}$
200	53 (+6%)	95.12$_{\pm 2.28}$	15.24$_{\pm 1.63}$	94.41$_{\pm 1.25}$	15.62$_{\pm 0.98}$
500	55 (+10%)	94.46$_{\pm 2.12}$	15.91$_{\pm 1.74}$	93.96$_{\pm 1.20}$	16.08$_{\pm 1.05}$
1000	57 (+14%)	93.87$_{\pm 2.05}$	16.38$_{\pm 1.81}$	93.42$_{\pm 1.18}$	16.61$_{\pm 1.12}$

7.4. Effectiveness of TriGuard-FL Components

To assess the contribution of each component within the proposed trust-triad framework, we perform an ablation study on CIFAR10 under PGD-based data poisoning (n=100, m=50, β=0.3, poisoning rate = 50%). The analysis isolates the impact of (*i*) auditable data valuation via Mahalanobis-based latent filtering, (*ii*) verifiable client contribution using MMD-based trust scoring $\Pi(\mathcal{P}_k)$, and (*iii*) AGI-based mitigability through antidote correction of poisonous updates at the server. Table 10 reports the global accuracy before attack ($\mathcal{A}_g\uparrow$) and degradation under attack ($\mathcal{D}_g\downarrow$), where higher $\mathcal{A}_g$ and lower $\mathcal{D}_g$ indicate better robustness.

Setup. To fairly evaluate the contribution of each trust-triad component, we isolate one module at a time while keeping the remaining pipeline identical to the standard FL setup.

Table 9. Ablation study on auditable distribution-level data valuation of TriGuard-FL under two different attacks across two datasets (HAM10000 and Census15), evaluated in non-IID ($\beta = 0.3$) FL settings with different number of total clients, single & multi-client attackers, and poisoning rates (PR) $= \frac{\nu}{\mathcal{N}_k} \times 100$, where ν denotes the fraction of poisonous samples and $\mathcal{N}_k$ denotes the total number of samples. We report clean accuracy $\mathcal{A}_g \uparrow$ and degradation accuracy under attack $\mathcal{D}_g{:=}100{-}\mathcal{A}'_g \downarrow$. **Bold** and second denote the best and second-best results per column, respectively.

Dataset	Attack method	n	m	Poisoning ratio (%)	No Val		FLTrust [5]		FLShield [17]		FedBary [23]		TriGuard-FL (ours)	
					$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$	$\mathcal{A}_g \uparrow$	$\mathcal{D}_g \downarrow$
HAM10000 [37]	PGD [25]	5	1	30	80.42±1.33	29.92±1.22	77.39±0.98	25.88±0.56	76.72±1.23	24.22±1.06	75.22±0.94	26.77±1.29	83.52±0.77	**20.97**±1.56
		10	3	30	79.37±0.89	30.71±0.74	76.23±1.37	26.76±1.72	75.26±0.77	25.43±0.88	74.23±1.41	28.49±0.64	82.74±1.38	**23.75**±0.69
		20	5	30	86.90±1.22	31.26±1.01	86.13±0.80	27.54±0.76	84.05±0.62	24.73±1.33	80.75±1.27	31.42±0.87	93.86±1.78	**20.27**±1.45
				50	86.90±1.22	37.96±0.75	86.13±0.80	37.02±0.42	84.05±0.62	36.15±1.02	80.75±1.27	40.12±1.56	93.86±1.78	**25.43**±1.17
				100	86.90±1.22	42.76±1.79	86.13±0.80	41.82±1.36	84.05±0.62	40.95±1.55	80.75±1.27	44.92±1.92	93.86±1.78	**28.27**±1.74
Census15 [30]	Targeted label flip [34]	5	1	30	90.39±1.28	19.77±1.53	87.36±1.44	16.73±1.12	86.69±1.23	15.07±1.48	85.19±1.12	17.62±1.57	93.49±1.46	**11.82**±1.24
		10	3	30	88.54±0.94	22.37±0.87	85.51±0.89	17.16±0.76	84.84±0.97	17.74±0.85	83.34±0.94	21.61±1.03	91.64±0.88	**14.27**±1.67
		20	5	30	86.46±0.93	24.47±0.95	83.43±1.10	20.43±1.26	82.76±1.31	19.59±1.47	81.26±1.26	21.70±0.88	89.56±1.84	**16.85**±1.62
				50	86.46±0.93	27.44±1.28	83.43±1.10	24.64±0.85	82.76±1.31	22.25±1.10	81.26±1.26	25.36±0.47	89.56±1.84	**19.97**±1.68
				100	86.46±0.93	35.48±1.53	83.43±1.10	30.61±0.86	82.76±1.31	28.71±1.75	81.26±1.26	34.01±1.09	89.56±1.84	**25.66**±1.28

The No TriGuard-FL baseline corresponds to vanilla FedAvg without any defense. The ADV-only configuration enables only the Mahalanobis-based client-side data filtering, while all other processes follow standard aggregation. In the VCC-only setting, trust scores $\Pi(\mathcal{P}_k)$ are computed using MMD and directly used as weights in the FedAvg aggregation step. For MPU-only, 20% of client updates are randomly selected and mitigated using AGI without trust weighting, representing isolated gradient correction. Combined configurations (e.g., ADV+VCC, ADV+MPU) activate the corresponding modules jointly, culminating in the full TriGuard-FL pipeline with all three components enabled.

Observations. Without any defense, the baseline (FedAvg) suffers severe degradation ($\mathcal{D}_g$=29.74%), confirming high vulnerability to poisoning. Introducing only ADV reduces $\mathcal{D}_g$ to 24.92%, demonstrating that client-side data auditing improves robustness by filtering noisy or poisoned samples. Activating VCC alone achieves $\mathcal{D}_g$=27.66%, showing moderate improvement by weighting clients based on distributional trust. The MPU module in isolation ($\mathcal{D}_g$=27.99%) provides limited benefit, indicating that AGI's mitigation is most effective when guided by accurate trust signals from ADV or VCC. When ADV and VCC are combined, degradation drops sharply to 20.61%, highlighting the synergy between auditable data filtering and verifiable aggregation. The pairing of ADV and MPU also yields high clean accuracy ($\mathcal{A}_g$=93.70%) and improved robustness ($\mathcal{D}_g$=21.33%), confirming that AGI benefits from reliable auditing. Finally, the full TriGuard-FL configuration, integrating all three components, achieves the best overall performance with $\mathcal{A}_g$=94.73% and $\mathcal{D}_g$=15.35%, validating that the joint optimization of auditability, verifiability, and mitigability provides maximum resilience against adversarial and distributional perturbations.

Overall, these findings confirm that while each trust-triad component independently contributes to robustness, their integration produces a compounding effect that ensures stable, trustworthy, and attack-resilient federated optimization.

Table 10. Ablation study on the individual and joint contributions of the proposed trust-triad components, namely, auditable data valuation (ADV), verifiable client contribution (VCC), and AGI-based mitigability of poisonous updates (MPU) under PGD attack [25] on CIFAR10 (n=100, m=50, β=0.3, poisoning rate = 50%). We report global accuracy before attack ($\mathcal{A}_g\uparrow$) and degradation under attack ($\mathcal{D}_g\downarrow$). **Bold** and values indicate the best and second-best results, respectively.

Auditable data valuation	Verifiable client contribution	AGI-based mitigability of poisonous updates	$\mathcal{A}_g(\%)\uparrow$	$\mathcal{D}_g(\%)\downarrow$
✗	✗	✗	90.25±1.91	29.74±0.84
✓	✗	✗	91.77±1.38	24.92±0.73
✗	✓	✗	92.59±1.56	27.66±1.73
✗	✗	✓	91.42±1.14	27.99±1.67
✓	✓	✗	91.45±1.74	20.61±0.81
✓	✗	✓	93.70±0.64	21.33±1.37
✗	✓	✓	92.94±1.70	26.75±1.53
✓	✓	✓	**94.73**±2.36	**15.35**±0.91

7.5. Hyperparameter Sensitivity and Robustness

Figure 3 presents a detailed sensitivity analysis of the key hyperparameters governing the behavior of TriGuard-FL under PGD-based data poisoning on CIFAR10, CIFAR100, and HAM10000 datasets. Each subfigure examines one hyperparameter of the trust-triad components, namely, privacy masking (auditability), adversarial tolerance (mitigability), and filtering strength (verifiability), to study their effect on global robustness and stability.

(a), (b) Gaussian noise parameters $(\mu_{\mathcal{N}}, \Sigma_{\mathcal{N}})$**.** Moderate masking values ($\mu_{\mathcal{N}}$=0.01, $\Sigma_{\mathcal{N}}$=0.05) provide the best balance between privacy preservation and model utility, yield-

ing peak accuracies of 15.35%, 42.07%, and 25.43% on CIFAR10, CIFAR100, and HAM10K, respectively. When the noise is too small, the masked distributional moments risk privacy leakage, weakening the auditability of data valuation; conversely, excessive noise corrupts moment estimation, leading to unstable aggregation and degraded performance.

(c) PGD attack strength (τ)**.** TriGuard-FL remains robust for perturbation magnitudes up to τ=0.03, demonstrating effective mitigability through AGI. Smaller τ values limit adversarial diversity, while larger perturbations disrupt gradient alignment between client and global updates, reducing the effectiveness of AGI in neutralizing adversarial components.

(d) Mahalanobis filtering threshold (γ)**.** An intermediate threshold (γ=0.0024) achieves the best trade-off between rejection and retention during latent-space filtering. Smaller γ values over-prune benign samples, impairing verifiability by shrinking the effective training set, whereas larger thresholds permit poisoned updates to bypass filtering, undermining robustness.

In summary, TriGuard-FL exhibits stable performance across a broad range of hyperparameter values, confirming the inherent robustness of its audit-verify-mitigate design. This consistency highlights its practicality for deployment in real-world federated systems, where precise hyperparameter tuning may be infeasible.

7.6. Trust-Score Separability

To illustrate how the MMD-based trust metric $\Pi(\mathcal{P}_k)$ differentiates benign and adversarial clients, we visualize its distributional behavior in a representative setting of TriGuard-FL trained on the CIFAR10 dataset under PGD-based poisoning [25] with 100 clients (m=50 adversarial), non-IID heterogeneity (β=0.3), and a 50% poisoning rate. Each client computes $\Pi(\mathcal{P}_k)$ as the squared Maximum Mean Discrepancy (MMD) between its local masked distribution $(\tilde{\mu}_k, \tilde{\Sigma}_k)$ and the global reference $\mathcal{P}_g^* = (\mu_g, \Sigma_g)$ at the end of round t. The dynamic trust threshold $\varphi_t = \text{Percentile}_{80}\big(\{\Pi(\mathcal{P}_k)\}_{k\in\mathcal{S}_t}\big)$ is used to adaptively select high-trust clients for aggregation and vice-versa for mitigation.

Figure 4 (a) shows the distribution of trust scores $\Pi(\mathcal{P}_k)$ across all clients. Benign clients concentrate at lower Π values, indicating strong alignment with the global reference $\mathcal{P}_g^*$, whereas malicious clients (pink) shift toward higher Π values, reflecting distributional drift due to poisoned updates. The dashed line φ_t marks the adaptive 80th-percentile threshold that partitions clients into high- and low-trust sets at each round. Clients with $\Pi(\mathcal{P}_k) \le \varphi_t$ are retained as reliable participants for aggregation and antidote gradient computation, while those exceeding φ_t are down-weighted or neutralized through the AGI mechanism. This

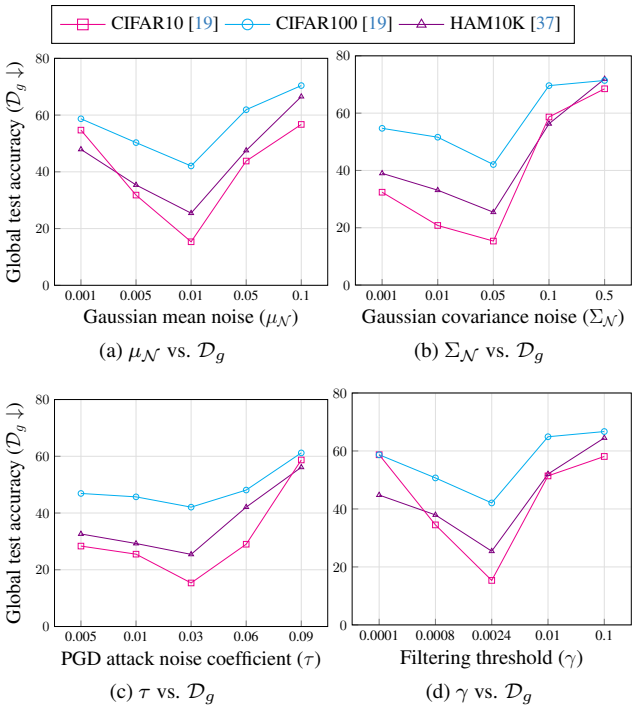


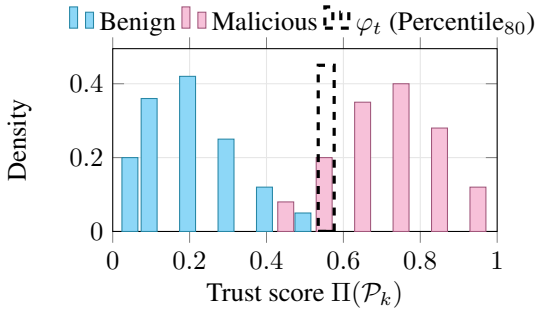
Figure 3. Impact of key hyperparameters in TriGuard-FL on degradation of global test accuracy ($\mathcal{D}_g \downarrow$) under PGD attack across CIFAR10, CIFAR100, and HAM10000 datasets with β = 0.3). We evaluate: (a) mean of Gaussian noise ($\mu_\mathcal{N}$), (b) covariance noise ($\Sigma_\mathcal{N}$), both used in density masking, (c) PGD attack strength (τ), and (d) Mahalanobis filtering threshold (γ).

distinct separation validates the discriminative capacity of TriGuard-FL's VCC module. Figure 4 (b) depicts the empirical cumulative distribution functions (ECDFs) of $\Pi(\mathcal{P}_k)$ for benign and malicious clients. The ECDF for benign clients rises rapidly at lower Π values, while that for malicious clients grows slowly, demonstrating that most benign clients exhibit low distributional divergence. The vertical gap at the threshold φ_t quantifies the statistical separability between the two populations. A large distance confirms that benign and adversarial clients occupy distinct trust-score regimes, enabling reliable trust-based selection without prior knowledge of attack labels.

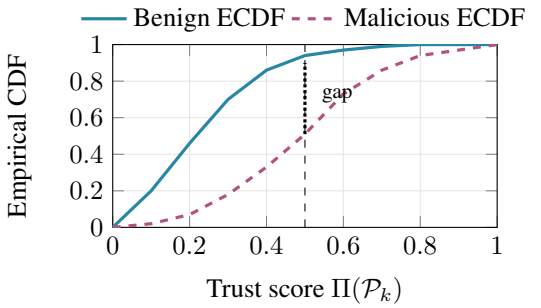
In summary, together, Figures 4(a), (b) demonstrate that the trust metric $\Pi(\mathcal{P}_k)$ provides a clear and quantifiable boundary between benign and malicious clients. This separation supports the stable operation of TriGuard-FL's trust-triad by ensuring that only statistically aligned, high-trust updates are emphasized during aggregation and mitigation.

8. Extended Limitations and Future Work

(i) **Adapting to dynamic and adaptive adversaries.** The current formulation assumes a stationary adversarial environment. A promising future direction is to extend



(a) Histogram of trust scores $\Pi(\mathcal{P}_k)$ for benign and malicious clients, showing the dynamic threshold φ_t and the selected high-trust set $\mathcal{H}_t$.



(b) Empirical cumulative distribution functions (ECDFs) of $\Pi(\mathcal{P}_k)$ for benign and malicious clients. The vertical gap (KS distance) at φ_t quantifies the statistical separability of the two distributions.

Figure 4. Visualization of trust-score dynamics in TriGuard-FL. (a) Benign clients exhibit low trust scores, concentrating left of φ_t, while malicious clients shift toward higher divergence regions. (b) ECDFs show clear separation between the two groups, confirming that the dynamic threshold φ_t effectively partitions high- and low-trust clients for robust selection and mitigation.

TriGuard-FL for online detection and adaptation to evolving or adaptive attackers by monitoring temporal shifts in trust and distributional consistency across training rounds.

(ii) **Extending mitigability to broader tasks.** While the present study focuses on classification, incorporating AGI-based mitigation into more complex FL tasks, such as detection, segmentation, or regression, could further generalize the applicability of the trust-triad framework in practical deployments. We plan this as our immediate future work extension.

References

[1] Naif Alkhunaizi, Dmitry Kamzolov, Martin Takáč, and Karthik Nandakumar. Suppressing poisoning attacks on federated learning for medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 673–683. Springer, 2022. 2

[2] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2022. 6, 8, 11, 13, 14

[3] Rajendra Bhatia and Chandler Davis. A cauchy-schwarz inequality for operators with applications. *Linear algebra and its applications*, 223:119–129, 1995. 7

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017. 2, 6, 11, 13, 14

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *ISOC Network and Distributed System Security Symposium (NDSS)*, 2021. 1, 2, 3, 6, 11, 12, 13, 14, 15

[6] Yu Chen, Fang Luo, Tong Li, Tao Xiang, Zheli Liu, and Jin Li. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Information Sciences*, 522:69–79, 2020. 2

[7] Mochan Fan, Kailai Ji, Zhaofeng Zhang, Hongfang Yu, and Gang Sun. Lightweight privacy and security computing for blockchained federated learning in iot. *IEEE Internet of Things Journal*, 10(18):16048–16060, 2023. 2

[8] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1623–1640, 2020. 1, 2

[9] Zahra Ghodsi, Mojan Javaheripi, Nojan Sheybani, Xinqiao Zhang, Ke Huang, and Farinaz Koushanfar. zprobe: Zero peek robustness checks for federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4860–4870, 2023. 2

[10] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pages 2242–2251. PMLR, 2019. 1, 3

[11] Lovedeep Gondara and Ke Wang. Differentially private small dataset release using random projections. In *Conference on Uncertainty in Artificial Intelligence*, pages 639–648. PMLR, 2020. 1

[12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 4, 6

[13] Fusheng Hao, Fengxiang He, Fuxiang Wu, Tichao Wang, Chengqun Song, and Jun Cheng. Task-aware clustering for prompting vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 14745–14755, 2025. 3

[14] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018. 9

[15] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn

Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019. 1, 3

[16] Meirui Jiang, Holger R Roth, Wenqi Li, Dong Yang, Can Zhao, Vishwesh Nath, Daguang Xu, Qi Dou, and Ziyue Xu. Fair federated medical image segmentation via client contribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16302–16311, 2023. 2

[17] Ehsanul Kabir, Zeyu Song, Md Rafi Ur Rashid, and Shagufta Mehnaz. Flshield: a validation based federated learning framework to defend against poisoning attacks. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 2572–2590. IEEE, 2024. 2, 3, 4, 6, 10, 11, 12, 13, 14, 15

[18] Aditya Pribadi Kalapaaking, Ibrahim Khalil, Xun Yi, Kwok-Yan Lam, Guang-Bin Huang, and Ning Wang. Auditable and verifiable federated learning based on blockchain-enabled decentralization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2024. 1, 2, 4, 5, 6

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. pages 32–33, 2009. 9, 10, 13, 14, 16

[20] Kummari Naveen Kumar, Chalavadi Krishna Mohan, and Linga Reddy Cenkeramaddi. The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2672–2691, 2023. 1

[21] Sourav Kumar, Anantharaman Lakshminarayanan, Ken Chang, Feri Guretno, Ivan Ho Mien, Jayashree Kalpathy-Cramer, Pavitra Krishnaswamy, and Praveer Singh. Towards more efficient data valuation in healthcare federated learning using ensembling. In *International Workshop on Distributed, Collaborative, and Federated Learning*, pages 119–129. Springer, 2022. 1, 2

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, pages 429–450, 2020. 6, 11, 13, 14

[23] Wenqian Li, Shuran Fu, Fengrui Zhang, and Yan Pang. Data valuation and detections in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12027–12036, 2024. 1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14, 15

[24] Sin Kit Lo, Yue Liu, Qinghua Lu, Chen Wang, Xiwei Xu, Hye-Young Paik, and Liming Zhu. Toward trustworthy ai: Blockchain-based architecture design for accountability and fairness of federated learning systems. *IEEE Internet of Things Journal*, 10(4):3276–3284, 2022. 1, 2

[25] Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017. 3, 11, 13, 15, 16

[26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 6, 11

[27] Thomas Minka. Estimating a dirichlet distribution, 2000. 4

[28] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pages 94–108, 2021. 2

[29] Arup Mondal, Yash More, Ruthu Hulikal Rooparaghunath, and Debayan Gupta. Poster: Flatee: Federated learning across trusted execution environments. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 707–709. IEEE, 2021. 2

[30] Muonneutrino. Us census demographic data. https://www.kaggle.com/muonneutrino/us-census-demographic-data, 2019. Accessed 2025-06-01. 9, 10, 14, 15

[31] Lokesh Nagalapatti, Ruhi Sharma Mittal, and Ramasuri Narayanam. Is your data relevant?: Dynamic selection of relevant data for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7859–7867, 2022. 1, 2

[32] Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*. 6, 11, 13, 14

[33] Raif M Rustamov. Closed-form expressions for maximum mean discrepancy with applications to wasserstein auto-encoders. *Stat*, 10(1):e329, 2021. 4

[34] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1354–1371. IEEE, 2022. 1, 11, 14, 15

[35] Rachael Hwee Ling Sim, Xinyi Xu, and Bryan Kian Hsiang Low. Data valuation in machine learning:" ingredients", strategies, and open challenges. In *IJCAI*, pages 5607–5614, 2022. 3

[36] J Michael Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004. 7

[37] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 9, 10, 13, 14, 15, 16

[38] Viktor Valadi, Xinchi Qiu, Pedro Porto Buarque De Gusmão, Nicholas D Lane, and Mina Alibeigi. {FedVal}: Different good or different bad in federated learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6365–6380, 2023. 1, 2

[39] Yuao Wang, Tianqing Zhu, Wenhan Chang, Sheng Shen, and Wei Ren. Model poisoning defense on federated learning: A validation based approach. In *International Conference on Network and System Security*, pages 207–223. Springer, 2020. 2

[40] Jeffry Wicaksana, Zengqiang Yan, Xin Yang, Yang Liu, Lixin Fan, and Kwang-Ting Cheng. Customized federated

learning for multi-source decentralized medical image classification. *IEEE Journal of Biomedical and Health Informatics*, 26(11):5596–5607, 2022. 9

[41] Sheng Wu, Yimi Wang, Xudong Liu, Yuguang Yang, Runqi Wang, Guodong Guo, David Doermann, and Baochang Zhang. Dfm: Differentiable feature matching for anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 15224–15233, 2025. 3

[42] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2020. 3, 11, 13

[43] Xinyi Xu, Shuaiqi Wang, Chuan Sheng Foo, Bryan Kian Hsiang Low, and Giulia Fanti. Data distribution valuation. *Advances in Neural Information Processing Systems*, 37:2407–2448, 2024. 1, 4, 6

[44] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 6, 11, 13, 14

[45] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 2, 13

[46] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. {BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning. In *2020 USENIX annual technical conference (USENIX ATC 20)*, pages 493–506, 2020. 1