AUTONOMOUS FEDERATED LEARNING: INCENTIVE-ALIGNED, ENERGY-AWARE, AND BYZANTINE-RESILIENT COORDINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) assumes clients are cooperative, honest, and energy-agnostic. These assumptions rarely hold in real deployments where participants own private data, operate under tight energy budgets, and face misaligned incentives. We address this gap by introducing Autonomous Federated Learning (AFL), a framework in which each client runs an agent that solves a local constrained decision problem to jointly optimize participation, update effort, and verification actions under explicit energy limits. To align individual incentives with collective progress, we propose Proof-of-Useful Learning (PoUL): a reward mechanism that compensates clients in proportion to verifiable utility gain per unit energy, where utility is assessed via committee-based validation on a shared reference set and energy is estimated through hardware-aware computation models. AFL maintains a persistent trust ledger encoding each client's contribution history and efficiency profile, ensuring that only reliable, energy-efficient participants accumulate long-term influence. Across CIFAR-10, FMNIST, and MNIST under heterogeneous and adversarial settings with 30% Byzantine clients, AFL achieves a worst-case accuracy degradation of only 3.93% on MNIST and 20.17% on CIFAR-10, compared to 13.52% and 46.48% for the strongest aggregation baseline (Fed-NGA), while methods such as CClip, MCA, and Geometric Median collapse to near-chance accuracy under structured attacks. AFL's trust ledger reliably isolates adversarial clients, assigning them confidence scores of less than 0.15 versus a benign-client mean of 0.75, effectively removing them from future participation. AFL improves utility-per-Joule by 15–35% over random client selection, results demonstrate a practical path toward FL that is simultaneously incentive compatible and robust without requiring computationally wasteful consensus proofs

1 INTRODUCTION

Most large-scale AI training today still happens inside a few organizations that hold both the data and the compute NVIDIA (2022); Lucinity (2024). However, this set-up is a poor fit when data sits in different institutions; hospitals, banks, city sensors that cannot pool raw records and do not fully trust one another. Without clear rewards, verifiable contribution, and reasonable privacy, many of those institutions simply refuse to join. Edge devices and software agents make the coordination problem more complex Singh et al. (2024). Decentralized learning raises several issues at once: privacy, verifiable compute, incentives, multi-agent orchestration, and scale. This paper concentrates on finding the solution to three of them: **incentive alignment**, **energy-aware participation**, and **Byzantine robustness**. While differential privacy is a valuable privacy safeguard, the noise it introduces increases gradient variance and interferes with the geometry-based checks underlying our MAD detector. Since a joint DP–robustness design involves non-trivial trade-offs in utility, convergence speed, and communication cost, we leave its principled integration to future work.

Federated Learning (FL) trains a shared model without moving raw data Konečný et al. (2016); Li et al. (2020); McMahan et al. (2017). A coordinator aggregates clients and updates over rounds. Standard FL, however, still assumes cooperative clients, fixed objectives, and similar participation costs. Open edge settings break those assumptions through free-riding Yoon et al. (2025), device and thermal volatility Afifi & Mokhtar (2025), and high communication or energy cost Wu et al.

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

(2024a). We therefore treat FL as repeated strategic interaction rather than altruistic optimization. *Autonomous Federated Learning* (AFL) pairs edge agents that decide how much to train under battery and thermal constraints. There is a coordinator that audits, updates and keeps a reputation ledger. Energy and reliability are first-class objectives; the system does not rely on unverified self-reports or proof-of-work style consensus. Our contributions are:

- We propose an autonomous framework in which a collection of decision-making agents orchestrate the federated learning process by dynamically selecting optimal client squads.
- A Byzantine filter that scores model utility, update geometry, and temporal consistency on coordinator-held data, then writes the outcome to a trust ledger. With MAD-based detection that detects free-riding, gradient poisoning, and on–off attacks.
- **Proof-of-Useful-Learning (PoUL)**: awards for verified accuracy gain per measured Joule, with a thermal penalty on inefficient hot runs, using a relative (tournament) efficiency term.
- A containerized deployment sketch: (Kubernetes/Docker, Kafka, Prometheus/Grafana) for heterogeneous agents and live energy/security monitoring.

2 Literature Review

2.1 Federated Learning: Security, Privacy, and Adversarial Threats

Federated Learning (FL) was introduced as a distributed learning paradigm in which models are trained collaboratively using decentralized data residing on client devices, without requiring the transfer of raw dataLee et al. (2022). Federated Averaging works in iterative rounds where selected clients perform local training and communicate model updates to a central coordinator for aggregation Chakraborty (2022). FL offers several advantages over centralized training, improving data privacy, mitigating the threat of data leakage and adhering to privacy regulations such as GDPR Chakraborty (2022); Takyar (2024). In regulated domains, FL allows organizations to collaborate without violating legal mandatesRodriguez (2025); Lucinity (2024). Additionally, FL facilitates learning from heterogeneous and distributed data sources, enabling models to benefit from diverse data distributions AI Sweden. Federated Learning struggles in open, large-scale decentralized settings due to several challenges, such as high communication costs for large models, non-IID data between clients, and differences in hardware, connectivity, and availability Splunk (2023). Moreover, classical FL typically assumes cooperative participants and static optimization objectives, limiting the application of FL in environments with autonomous and self-interested entities. It also introduces security vulnerabilities arising from decentralized training and limited visibility. FL systems are vulnerable to model poisoning and backdoor attacks, where malicious participants manipulate local updates to influence the global modelSplunk (2023) Desk (2023)Gu et al. (2019). Byzantine attacks in federated learning take several forms. Gaussian attacks inject random noise, while same-value attacks replace updates with constant vectors, discarding useful gradient information. Sign-flip attacks reverse update directions to disrupt optimization. More subtle attacks include Little Is Enough (LIE), which introduces statistically plausible deviations that gradually shift the global model, and Fall of Empires (FoE), where colluding clients coordinate malicious updates to influence aggregation.Several robust aggregation methods have been proposed to mitigate these threats, including **Fed-NGA** Zuo et al. (2024), **Median** Xie et al. (2018), **Krum** Blanchard et al. (2017), **Geometric Median (GM)** Pillutla et al. (2022), **MCA** Luan et al. (2024), and **CClip** Karimireddy et al. (2021). These methods reduce the influence of anomalous updates, but often involve trade-offs in utility, convergence speed, or communication cost. Moreover, many existing frameworks assume honest-but-curious participants and may not fully address strategic or adaptive adversaries.

2.2 Agentic Systems

Multi-Agent Systems (MAS) offer a natural foundation for distributed AI, where autonomous entities coordinate, cooperate, or compete toward individual or shared goals Jennings (1996). Agents vary in sophistication: reactive agents respond to environmental stimuli without internal state Mnih et al. (2015), deliberative agents reason over goals, beliefs, and action plans Panait & Luke (2005), and learning agents adapt their policies over time through reinforcement learning or online optimization Cucker & Smale (2007). In practice, composite agents blend these capabilities to support

execution, reasoning, and adaptation simultaneously. Despite this flexibility, existing agent-based approaches are rarely integrated with large-scale federated learning, and typically lack mechanisms for verifiable contribution assessment, energy awareness, and secure coordination under adversarial conditions.

2.3 Blockchain & Incentive Mechanisms

Blockchain provides a decentralized, append-only ledger for verifiable record-keeping and coordination among mutually untrusted participants Kang et al. (2017); Nakamoto (2008). This has led to its adoption in distributed systems, including in-vehicle networks Singh & Kim (2017); Dorri et al. (2017), industrial Internet of Things platforms Nanda et al. (2023); Taherdoost (2023), and medical data sharing infrastructures Liu et al. (2020b). In distributed machine learning, these properties have motivated **Blockchain-based Federated Learning (BFL)** as a way to reduce trust assumptions in federated learning. Sun et al. Zhang et al. (2021a); Sun et al. (2022) proposed a blockchain-based reputation framework that uses historical participation to improve aggregation robustness, while Wang et al. Wang et al. (2023) explored blockchain-assisted mechanisms for auditable and secure coordination. Although these approaches improve transparency and fault tolerance, they can introduce additional communication and computational costs and often rely on static participation and energy-agnostic incentives. Incentive design remains a core challenge in federated learning (FL), especially in open environments where participants are autonomous and self-interested Kairouz et al. (2021); Yang et al. (2019). The Shapley value has emerged as a principled approach to quantifying individual contributions, allocating rewards based on marginal impact on model performance Jia et al. (2019); Song et al. (2019). However, exact computation is prohibitively expensive at scale, motivating approximate and blockchain-assisted alternatives. Several blockchain-based incentive mechanisms have been proposed to address this. Liu et al.'s *FedCoin* enables peer-to-peer payments where consensus nodes compute approximate Shapley values via a Proof-of-Shapley (PoSap) protocol Liu et al. (2020a). Zhao et al. combine Multi-KRUM aggregation with reputation-based rewards to simultaneously defend against poisoning attacks and compensate honest participants Zhao et al. (2021). Lim et al. take a more formal approach, proposing a hierarchical framework grounded in contract theory and coalition game theory, solved through backward induction and merge-and-split algorithms Lim et al. (2020). Other approaches rely on system-level heuristics. BlockFL Kim et al. (2018) rewards participants in proportion to local data volume, while Zhang et al. Zhang et al. (2021b) tie rewards to centroid distance metrics for failure detection in IoT settings. Though practical, such proxies are susceptible to manipulation and fail to capture true contribution utility. Song et al. address this gap with a Shapley-inspired contribution index tailored specifically for FL Song et al. (2019). Nevertheless, a persistent blind spot across these mechanisms is the neglect of energy cost and computational efficiency as meaningful dimensions of participant contribution.

2.4 Positioning Relative to Prior Work

Energy, incentives, trust, Byzantine defenses, and agent framing have mostly been studied separately. Below we place AFL next to the closest systems on five axes: **(A)** energy, **(B)** incentives, **(C)** trust ledger, **(D)** Byzantine detection, **(E)** autonomous agents.TAIM Xu et al. (2025) proposes a multi-dimensional trust model evaluating participation frequency, gradient consistency, and contribution effectiveness, coupled with a Stackelberg game-based incentive design and confidence-based soft filtering for Byzantine mitigation covering incentives, trust, and partial Byzantine robustness. Kang et al. (2019) introduced a reputation-based incentive mechanism combining multi-weight subjective logic with contract theory, stored on a blockchain ledger. Worker selection based on reliability scores provides partial Byzantine resilience. Although energy is acknowledged at the IoT edge level, no formal thermal-penalty efficiency metric is defined. Cao et al. (2021) introduced FLTrust: the server scores clients by cosine alignment to a clean root update. We keep that idea of a private reference set, but score several signals together validation accuracy, distance, alignment, temporal variance, and energy efficiency and fold them into a multi-round trust ledger rather than a one-shot weight. Lai et al. (2021) introduced Oort for systems-aware client selection in FL, prioritizing clients based on both statistical utility and system-level metrics, offering partial energy awareness through system-aware selection. Liu et al. (2020a) proposed FedCoin, a peer-to-peer payment system using a Proof-of-Shapley protocol to compute approximate Shapley values for contribution-based reward allocation via blockchain covering incentive mechanisms and trust management. Some related systems add formal Differential Privacy. ZTFed-MAS2S Li et al. (2025) combines verifiable DP with

Table 1: Coverage of prior work across AFL's five design dimensions. ✓ = fully addressed (1.0), ∼ = partially addressed (0.5), × = not addressed (0).

Prior Work	Energy	Incentive	Trust	Byzantine	Autonomous	Total
TAIM Xu et al. (2025)	×	✓	✓	∼	×	2.5
VBFL/Kang Kang et al. (2019)	∼	✓	✓	∼	×	3.0
FLTrust Cao et al. (2021)	×	×	✓	✓	×	2.0
Oort Lai et al. (2021)	∼	×	×	×	×	0.5
FedCoin Liu et al. (2020a)	×	✓	✓	×	×	2.0
ZTFed-MAS Li et al. (2025)	×	×	✓	✓	∼	2.5
IAFL Wu et al. (2024b)	×	✓	×	×	×	1.0
BALANCE Fang et al. (2024)	×	×	×	✓	×	1.0
Fed-DPRoC Xia et al. (2025)	×	×	×	✓	×	1.0
Green FL Strubell et al. (2019)	✓	×	×	×	×	1.0
AFL (ours)	✓	✓	✓	✓	✓	**5.0**

trust-aware aggregation; Fed-DPRoC Xia et al. (2025) targets DP, robustness, and compression together. Wu et al. (2024b) proposed Incentive-Aware Federated Learning, which provides differentiated training-time model rewards to self-interested clients. IAFL addresses incentive compatibility (component B) with a novel stochastic reference-model recovery strategy to prevent error propagation. Fang et al. (2024) introduced BALANCE for Byzantine-robust decentralized FL, where each client uses local model similarity as a reference for filtering poisoned updates from peers. BALANCE provides strong Byzantine robustness guarantees in fully decentralized settings. Existing Green FL approaches Strubell et al. (2019); Qiu et al. (2023) focus on reducing energy consumption through adaptive client selection, update compression, and precision control, addressing only component A. Table 1 scores each system on the five dimensions; none cover all five except AFL.

3 System Architecture & Methodology

We propose **Autonomous Federated Learning (AFL)**, a framework for robust federated learning under adversarial and resource-constrained conditions. AFL uses a hierarchical coordinator–client architecture with autonomous agents for adaptive participation, security-aware filtering, and robust aggregation. Clients bid and train within local resource budgets, while the coordinator maintains the trust ledger, applies MAD-based detection, and assigns PoUL rewards before aggregating only accepted updates. We consider N heterogeneous edge clients $\mathcal{C} = \{1, 2, \ldots, N\}$ over communication rounds $t = 1, \ldots, T$. Each client $i \in \mathcal{C}$ holds a private dataset $\mathcal{D}_i$ of size D_i (in MB), while the coordinator maintains the global model $w^{(t)}$ and a private reference validation set $\mathcal{D}_{\text{val}}$ that is never exposed to clients. To jointly handle client selection, resource constraints, and temporal dependencies, AFL organizes each communication round into three phases:

1. Pre-Training Squad Selection Phase: Before executing local training, client agents express their availability and bidding intent $b_i^{(t)} \in \{0, 1\}$. The coordinator computes a selection priority confidence score $C_i^{(t)}$ based on historical trust and verified performance from the preceding round:

$$C_i^{(t)} = \tau_i^{(t-1)} \cdot \sigma\big(\kappa_{\text{acc}} \Delta\text{acc}_i^{(t-1)}\big) \cdot \left(\frac{\eta_i^{(t-1)}}{\max_{j \in \mathcal{C}} \eta_j^{(t-1)}}\right) \cdot \exp\big(-\gamma \mathbb{I}_i^{(t-1)}\big), \tag{1}$$

where $\Delta\text{acc}_i^{(t-1)} = a_i^{(t-1)} - \text{acc}_{\text{global}}^{(t-1)}$ is the historical marginal accuracy gain, $\sigma(z) = (1+e^{-z})^{-1}$ is the logistic sigmoid with scaling factor κ_{acc}, $\eta_i^{(t-1)}$ is historical compute efficiency, $\mathbb{I}_i^{(t-1)} \in \{0, 1\}$ is the previous Byzantine flag, and $\gamma > 0$ is a security penalty exponent. The coordinator selects active squad:

$$\mathcal{S}^{(t)} = \text{Top-}K \left\{ C_i^{(t)} \;\middle|\; b_i^{(t)} = 1,\ C_i^{(t)} \geq \delta_{\text{sel}} \right\}, \tag{2}$$

where $\delta_{\text{sel}} \in (0, 1)$ is the minimum admission threshold.

2. Local Training & Auditing Phase: Only selected clients $i \in \mathcal{S}^{(t)}$ train on local data and submit a hardware-telemetry payload:

$$\mathcal{U}_i^{(t)} = \big(\theta_i^{(t)},\ E_i^{(t)},\ \overline{T}_i^{(t)}\big), \tag{3}$$

comprising updated model weights $\theta_i^{(t)} = w_i^{(t)}$, measured electrical energy expenditure $E_i^{(t)}$ (in Joules), and average core temperature $\overline{T}_i^{(t)}$ (°C). The Client's accuracy is not self-reported; the coordinator independently audits the candidate weights on $\mathcal{D}_{\text{val}}$ to compute the audited accuracy $a_i^{(t)} = \text{Acc}(w_i^{(t)}; \mathcal{D}_{\text{val}})$ and the geometry of the direction update.

3. Trust Ledger Update & Aggregation Phase: The coordinator updates the persistent trust score $\tau_i^{(t)} \in [0, 1]$ stored on an append-only ledger via exponential moving average:

$$\tau_i^{(t)} = (1 - \mu_{\text{trust}})\, \tau_i^{(t-1)} + \mu_{\text{trust}} \left[\gamma_1 \sigma\big(\kappa_{\text{acc}} \Delta\text{acc}_i^{(t)}\big) + \gamma_2 \left(\frac{\eta_i^{(t)}}{\max_{j \in \mathcal{S}^{(t)}} \eta_j^{(t)}} \right) - \gamma_3 \mathbb{I}_i^{(t)} \right], \quad (4)$$

clipped to $[0, 1]$, where $\mu_{\text{trust}} \in (0, 1)$ controls historical memory decay, and $\gamma_1, \gamma_2, \gamma_3 > 0$ balance utility, efficiency, and security penalties.

Figure 1 illustrates the coordination workflow.

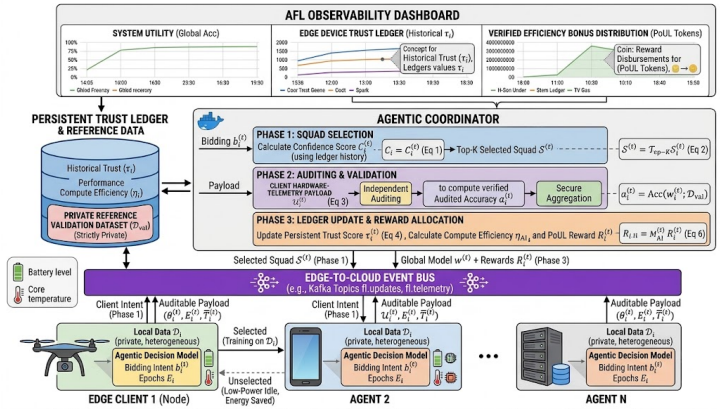

Figure 1: High-level architecture of the AFL framework.

Local Agent Decision Model. At the beginning of round t, local agent i observes private state $s_i^{(t)} = \big(\tau_i^{(t-1)}, R_i^{(t-1)}, E_i^{\text{budget}}, \overline{T}_i^{(t-1)}, D_i, \eta_i^{(t-1)}\big)$. Agent i optimizes its bidding decision $b_i^{(t)} \in \{0, 1\}$ and training epochs E_i to maximize expected cumulative reward subject to remaining battery E_i^{budget} and thermal safety limits. We define client compute efficiency η_{AI} as useful data processed per unit electrical energy, penalizing CMOS thermal leakage overhead Hennessy & Patterson (2020):

$$\eta_{\text{AI}} = \frac{D_i}{W_{\text{elec},i}} \cdot \exp\Big[-\beta_{\text{therm}} \big(\overline{T}_i - T_{\text{ref}}\big) \Big], \quad \text{where} \quad W_{\text{elec},i} = \int_0^{\Delta t_i} P_i(u)\, du, \quad (5)$$

where D_i is data processed (MB), $P_i(u)$ is instantaneous power draw over duration Δt_i, T_{ref} is a reference operating temperature, and $\beta_{\text{therm}} > 0$ is a CMOS leakage scaling coefficient. On the coordinator, the PoUL reward $R_i^{(t)}$ allocated to client i couples verified marginal learning gain with relative hardware efficiency:

$$R_i^{(t)} = w_{\text{acc}} \cdot \Delta\text{acc}_i^{(t)} + w_{\text{eng}} \cdot \left(\frac{\eta_i^{(t)}}{\max_{j \in \mathcal{S}^{(t)}} \eta_j^{(t)}} \right) \cdot \mathbb{I}\big(\Delta\text{acc}_i^{(t)} > 0\big), \quad (6)$$

where $w_{\text{acc}}, w_{\text{eng}} \geq 0$ balance accuracy improvement and green compute, and $\mathbb{I}(\cdot)$ ensures efficiency bonuses are disbursed only when the client delivers strictly positive marginal utility.

3.0.1 Proof-of-Useful-Learning (PoUL)

Unlike blockchain Proof-of-Useful-Work (PoUW), which substitutes mining hashes with model training to secure a ledger, PoUL only prices contributions *inside* one federation: how much verified accuracy a client bought per Joule of measured energy.

(i) Relative efficiency contest: Clients are scored against the best efficiency in the current squad, $\eta_i / \max_j \eta_j$, so absolute energy reports matter less than relative standing Lazear & Rosen (1981). Private training cost remains private; the contest still pushes agents to spend effort that shows up in audited utility.

Proposition 1 (**Incentive alignment under PoUL**). *Assume $\mathcal{D}_{val}$ stays private to the coordinator. Under the PoUL reward (Eq. 6) and trust update (Eq. 4), free-riding with a stale model, injecting noise, or overfitting to local data yields $\mathbb{E}[R_i] \leq 0$ and drives $\tau_i \to 0$. Truthful training with positive marginal gain is then the only strategy that keeps a client eligible for future rewards.*

Appendix B sketches the argument.

(ii) Checks against gaming: To prevent clients from artificially inflating their scores, AFL evaluates updates on the private validation set $\mathcal{D}_{\text{val}}$, which is not shared with clients. The framework mitigates free-riding by identifying updates that remain too close to the global model, i.e., $\|w_i^{(t)} - w^{(t)}\|_2 < \epsilon_{\text{dist}}$, and only grants efficiency bonuses when the client provides a positive accuracy improvement, $\Delta\text{acc}_i^{(t)} > 0$.

(iii) Late-training taper: As the global model improves, $\Delta\text{acc}_i^{(t)}$ shrinks, so rewards fall even for honest clients payoffs track where the federation still needs work.

Coordinator-Side Byzantine Detection: At round t, the coordinator audits candidate updates $\{w_i^{(t)}\}_{i \in \mathcal{S}^{(t)}}$ across model utility, update geometry, and temporal consistency against private dataset $\mathcal{D}_{\text{val}}$. To remain resilient against coordinated collusion and Little Is Enough (LIE) attacks, we com-

Algorithm 1 Coordinator-Side Byzantine Detection

Require: Candidate models $\{w_i^{(t)}\}_{i \in \mathcal{S}^{(t)}}$, global model $w^{(t)}$, validation set $\mathcal{D}_{\text{val}}$, sensitivity κ
Ensure: Flagged Byzantine set $\mathcal{B}^{(t)}$ and indicator vector $\mathbb{I}^{(t)}$

Compute server reference update: $\Delta w_{\text{ref}}^{(t)} \leftarrow -\nabla F(w^{(t)}; \mathcal{D}_{\text{val}})$
for each client $i \in \mathcal{S}^{(t)}$ **do**
  Audited accuracy: $a_i^{(t)} \leftarrow \text{Acc}(w_i^{(t)}; \mathcal{D}_{\text{val}})$
  Update Euclidean distance: $d_i^{(t)} \leftarrow \|w_i^{(t)} - w^{(t)}\|_2$
  Directional alignment: $c_i^{(t)} \leftarrow \frac{\langle w_i^{(t)} - w^{(t)},\, \Delta w_{\text{ref}}^{(t)} \rangle}{\|w_i^{(t)} - w^{(t)}\|_2 \, \|\Delta w_{\text{ref}}^{(t)}\|_2}$
  Temporal consistency: $\sigma_i^{(t)} \leftarrow \text{Std}(a_i^{(t-2)}, a_i^{(t-1)}, a_i^{(t)})$
end for
Compute robust thresholds via Median $\pm$ MAD:
$\Theta_{\text{acc}} \leftarrow \widetilde{\mathbf{a}} - \kappa \cdot \text{MAD}(\mathbf{a}), \quad \Theta_{\text{dist}} \leftarrow \widetilde{\mathbf{d}} + \kappa \cdot \text{MAD}(\mathbf{d})$
$\Theta_{\text{cos}} \leftarrow \widetilde{\mathbf{c}} - \kappa \cdot \text{MAD}(\mathbf{c}), \quad \Theta_{\text{dist,min}} \leftarrow \epsilon_{\text{dist}}$
Initialize $\mathcal{B}^{(t)} \leftarrow \emptyset$
for each client $i \in \mathcal{S}^{(t)}$ **do**
  if $a_i^{(t)} < \Theta_{\text{acc}} \vee d_i^{(t)} > \Theta_{\text{dist}} \vee d_i^{(t)} < \Theta_{\text{dist,min}} \vee c_i^{(t)} < \Theta_{\text{cos}} \vee \sigma_i^{(t)} > \delta_{\text{var}}$ **then**
    $\mathcal{B}^{(t)} \leftarrow \mathcal{B}^{(t)} \cup \{i\}, \quad \mathbb{I}_i^{(t)} \leftarrow 1$
  else
    $\mathbb{I}_i^{(t)} \leftarrow 0$
  end if
end for
return $\mathcal{B}^{(t)}, \mathbb{I}^{(t)}$

pute distribution medians and Median Absolute Deviations (MAD). For metric vector $\mathbf{x} = \{x_i\}$:

$$\widetilde{x} = \text{Median}(\mathbf{x}), \quad \text{MAD}(\mathbf{x}) = \text{Median}\big(|x_i - \widetilde{x}|\big) + \epsilon_{\text{stab}}, \tag{7}$$

where $\epsilon_{\text{stab}} > 0$ prevents division by zero. Cutoffs are established as $\Theta_{\text{acc}} = \widetilde{a} - \kappa\text{MAD}(\mathbf{a})$, $\Theta_{\text{dist}} = \widetilde{d} + \kappa\text{MAD}(\mathbf{d})$, and $\Theta_{\text{cos}} = \widetilde{c} - \kappa\text{MAD}(\mathbf{c})$, where $\kappa = 2.5$ sets detection sensitivity.

Table 2: Auditing signals evaluated by the coordinator in AFL.

Metric	Formulation	Operational Detection Purpose
Audited Accuracy	$a_i^{(t)} = \mathrm{Acc}(w_i^{(t)}; \mathcal{D}_{\mathrm{val}})$	Prevents accuracy inflation and verifies genuine generalization on held-out data.
Update Distance	$d_i^{(t)} = \|w_i^{(t)} - w^{(t)}\|_2$	Binds perturbation scale ($d_i^{(t)} > \Theta_{\mathrm{dist}}$) and catches zero-effort free-riding ($d_i^{(t)} < \epsilon_{\mathrm{dist}}$).
Directional Alignment	$c_i^{(t)} = \cos\Big(w_i^{(t)} - w^{(t)},\ \Delta w_{\mathrm{ref}}^{(t)}\Big)$	Validates cosine alignment against the clean server reference gradient $\Delta w_{\mathrm{ref}}^{(t)}$, detecting sign-flip and orthogonal drift.
Temporal Consistency	$\sigma_i^{(t)} = \mathrm{Std}\Big(a_i^{(t-2)}, a_i^{(t-1)}, a_i^{(t)}\Big)$	Flags alternating or adaptive on–off adversaries exhibiting suspicious performance volatility.

3.1 Theoretical Convergence Analysis

We establish the convergence properties of AFL's trust-weighted aggregation under non-IID data distributions and Byzantine contamination.

Assumption 1 (Standard Optimization Conditions). *The global loss function $F(w)$ is L-smooth: $\|\nabla F(v) - \nabla F(w)\| \leq L\|v - w\|$. For benign clients, stochastic gradients have bounded variance $\mathbb{E}\|\nabla F_i(w, \xi) - \nabla F_i(w)\|^2 \leq \sigma^2$ and bounded non-IID gradient dissimilarity $\|\nabla F_i(w) - \nabla F(w)\|^2 \leq G^2$.*

Theorem 1 (Ergodic Convergence of AFL Aggregation). *Let Assumption 1 hold and learning rate satisfy $\eta = \frac{1}{L\sqrt{T}}$. Under AFL multi-signal MAD filtering and trust-weighted aggregation with active squad size K, the average squared gradient norm over T communication rounds satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^{(t)})\|^2 \leq \frac{2\big(F(w^{(0)}) - F^*\big)}{\eta T} + \frac{L\eta\sigma^2}{K} + \mathcal{O}(G^2) + \mathcal{O}\left(\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{B}^{(t)}}\tau_i^{(t)}\|\Delta w_{adv,i}^{(t)}\|\right). \tag{8}$$

Asymptotic intuition. Flagged Byzantines get $\mathbb{I}_i^{(t)} = 1$, so Eq. 4 contracts their trust: $\tau_i^{(t)} \leq (1 - \mu_{\text{trust}})^t \tau_i^{(0)} \to 0$. The adversarial leakage term in Eq. 8 then vanishes, and the rate matches standard clean FL, $\mathcal{O}(1/\sqrt{T})$. The proof is in Appendix A.

4 Experimental Setup

Experiments are conducted on randomly partitioned, non-IID and imbalanced splits of CIFAR-10, FMNIST, and MNIST using a Dirichlet distribution with $\alpha_{\mathrm{Dir}} = 0.6$. We use a lightweight **SimpleCNN** with two 3×3 convolutional layers (32 and 64 channels), ReLU activations, 2×2 max pooling, and fully connected layers of sizes 512 and 10 with dropout ($p = 0.5$). Global updates are aggregated using Federated Averaging. Unless stated otherwise, experiments use $N = 10$ clients, $T = 100$ communication rounds, and $E = 3$–5 local epochs. Experiments run on a workstation equipped with an *Intel Core i9-12900K*, *NVIDIA RTX 4070*, and 32 GB RAM. To emulate heterogeneous edge environments, clients are deployed as containerized Docker/Kubernetes processes with isolated CPU cores, restricted memory, and hardware classes $h_i \in \{\text{CPU-only}, \text{GPU-desktop}, \text{GPU-server}, \text{edge}\}$. Client resource usage is monitored through a hardware telemetry daemon based on **LibreHardwareMonitor**, which exposes a local REST API and samples CPU power and temperature, along with GPU power and temperature, at 1 Hz. Since energy is reported by edge daemons, the coordinator implements a 3-stage verification pipeline to detect fabricated or implausible telemetry:

Stage 1: Hardware-Class Plausibility Envelopes. Upon registration, each client declares its hardware class h_i. The coordinator maintains per-class energy envelopes $[E_{\min}^{(h_i)}, E_{\max}^{(h_i)}]$ calibrated from empirical benchmarks. Reports outside this envelope are flagged:

$$\text{flag}_{E,\text{env}}(i,t) = \mathbf{1}\left[E_i^{(t)} < E_{\min}^{(h_i)} \ \vee\ E_i^{(t)} > E_{\max}^{(h_i)}\right]. \tag{9}$$

Table 3: Test accuracy (%) under five Byzantine attack scenarios for **MNIST** ($\alpha = 0.6$, $\beta = 0.4$, $f/N = 0.3$) and **CIFAR-10** ($\alpha = 0.6$, $\beta = 0.3$, $f/N = 0.3$). Δ_{worst} denotes the relative accuracy drop from the *No Attack* baseline to the worst-case attack.

MNIST							
Method	**No Attack**	**Gaussian**	**Same-val.**	**Sign-flip**	**LIE**	**FoE**	$\boldsymbol{\Delta}_{\text{worst}}$ (%)↓
AFL (ours)	**96.80**	**95.50**	**93.00**	**95.70**	**95.10**	**95.30**	**3.93**
Fed-NGA	96.72	94.98	83.66	94.71	94.92	94.71	13.52
Median	92.61	92.57	61.60	92.33	92.59	68.09	33.48
Krum	92.60	92.73	60.63	92.28	92.71	67.50	34.53
GM	94.67	94.46	70.11	94.31	94.33	11.12	88.25
MCA	95.10	94.82	11.35	11.35	94.84	11.35	88.07
CClip	94.99	94.87	10.09	9.80	94.93	9.80	89.68

CIFAR-10							
Method	**No Attack**	**Gaussian**	**Same-val.**	**Sign-flip**	**LIE**	**FoE**	$\boldsymbol{\Delta}_{\text{worst}}$ (%)↓
AFL (ours)	**63.64**	**61.27**	**50.80**	**60.95**	**61.01**	**61.25**	**20.17**
Fed-NGA	54.48	52.07	29.16	51.16	51.82	51.16	46.48
Median	39.98	39.64	10.70	38.87	39.75	14.19	73.24
Krum	40.76	39.55	10.11	39.67	39.96	14.02	75.20
GM	48.41	48.08	28.81	47.89	47.90	10.47	78.37
MCA	48.67	48.53	10.00	10.00	48.28	10.00	79.45
CClip	48.59	43.42	10.00	10.00	47.79	10.00	79.42

Stage 2: Cross-Round Temporal Smoothness. Physical energy draw fluctuates smoothly across rounds for continuous local workloads. A client whose temporal standard deviation $\sigma_{E,i}^{(t)} = \mathrm{Std}(E_i^{(t-2)}, E_i^{(t-1)}, E_i^{(t)})$ exceeds a MAD-based threshold is flagged:

$$\text{flag}_{E,\text{smooth}}(i,t) = \mathbf{1}\left[\sigma_{E,i}^{(t)} > \widetilde{\sigma}_E + \kappa \cdot \mathrm{MAD}(\boldsymbol{\sigma}_E)\right]. \tag{10}$$

Stage 3: Compute–energy consistency. A client can still game Stage 1–2 by reporting a flat energy value near $E_{\min}^{(h)}$. We therefore check whether reported Joules match observed work and wall time:

$$\chi_i^{(t)} = \frac{D_i \cdot E_{\text{loc},i}}{E_i^{(t)} \cdot \Delta t_{\text{train},i}}, \tag{11}$$

where $E_{\text{loc},i}$ is the number of local epochs, $E_i^{(t)}$ is reported energy (J), and $\Delta t_{\text{train},i}$ is measured training time. Implausibly high $\chi_i^{(t)}$ (lots of data, little energy or time) sets $\text{flag}_{E,\text{Pareto}}(i,t) = 1$. The energy indicator $\mathbb{I}_{E,i}^{(t)} = \text{flag}_{E,\text{env}} \vee \text{flag}_{E,\text{smooth}} \vee \text{flag}_{E,\text{Pareto}}$ enters the trust update as an additional security penalty (same role as $\mathbb{I}_i^{(t)}$ in Eq. 4); hard rejection into $\mathcal{B}^{(t)}$ remains driven by Algorithm 1. In production, software-reported metrics could be replaced with TPM/RAPL-based attestation (Appendix D). AFL tracks a compact set of signals covering model performance, training progress, client energy use, anomalous behavior, and update contribution.

5 Experimental Evaluation

We evaluate AFL on Byzantine robustness, energy-aware squad selection, and per-client trust scores. Attacks cover Gaussian noise, sign-flip, LIE Baruch et al. (2019), same-value, and Fall-of-Empires (FoE) Xie et al. (2020), each with $f = 3$ of $N = 10$ clients (30%). Baselines are Fed-NGA Zuo et al. (2024), coordinate-wise Median Xie et al. (2018), Krum Blanchard et al. (2017), Geometric Median Pillutla et al. (2022), MCA Luan et al. (2024), and Centered Clipping Karimireddy et al. (2021), all under the same splits, model, and local training setup. Table 3 list final test accuracy on MNIST and CIFAR-10. On MNIST, AFL is best in every column, including the clean run. Under same-value attacks, Krum and Median fall to about 61%, Fed-NGA to 83.66%, and AFL stays at 93.00% ($+9.34$ pp over the next best). CClip and MCA drop near chance ($\sim 10\%$) on same-value and FoE, so their Δ_{worst} exceeds 88%; AFL's is 3.93%. On CIFAR-10 the pattern repeats.

Table 4: Per-client metrics on **FMNIST** at convergence ($T = 100$): marginal accuracy contribution Δacc_i, compute efficiency η_i, Byzantine flag $\mathbb{I}_i$, and composite confidence score C_i. Clients 7 and 9 were injected as sign-flip adversaries in an illustrative 10-client checkpoint trace.

Client	$\boldsymbol{\Delta\text{acc}_i}$ (%)	$\boldsymbol{\eta_i}$ (MB/J)	**Risk** $\boldsymbol{\mathbb{I}_i}$	**Confidence** $\boldsymbol{C_i}$
1	+3.0	0.024	0	0.82
2	+2.0	0.019	0	0.70
3	+4.0	0.026	0	0.85
4	+2.5	0.022	0	0.77
5	+3.5	0.024	0	0.83
6	+0.5	0.016	0	0.55
7	-2.0	0.027	1	0.10
8	+2.0	0.020	0	0.68
9	-1.0	0.017	1	0.15
10	+2.5	0.022	0	0.77
Mean (benign clients only): $\overline{\Delta\text{acc}} = +2.50\%$, $\bar{\eta} = 0.022$ MB/J, $\bar{C} = 0.75$				

AFL stays above 50.80% on every attack; every baseline dips below 30% on at least one. MCA, CClip, Median, and Krum hit 10% under same-value, sign-flip, or FoE. Those rules each protect one geometric property (coordinate median, pairwise distance, or norm). When the attack respects that property, they miss it. AFL checks accuracy on $\mathcal{D}_{\text{val}}$, distance, alignment to the server reference update, and short-term accuracy variance together (Algorithm 1), so an attack has to clear all four. Table 4 is a per-client snapshot on FMNIST at $T = 100$. Confidence $C_i = \tau_i^{(t-1)} \cdot \sigma(\kappa_{\text{acc}} \Delta\text{acc}_i) \cdot (\eta_i / \max_j \eta_j) \cdot \exp(-\gamma\, \mathbb{I}_i)$ is what the selector uses. Adversaries 7 and 9 sit at $C_i \leq 0.15$, below $\delta_{\text{sel}} = 0.3$, so they are dropped from later rounds. Once flagged, trust decays quickly (3–5 rounds in our runs), which limits both persistent poisoning and on–off reinjection. Among benign clients, C_i tracks utility-per-Joule (Spearman $\rho \approx 0.94$). Client 6 is honest but slow and low-gain ($+0.5\%$, 0.016 MB/J), so it gets $C_6 = 0.55$ the ledger down-weights weak contributors even without malice. Against random, round-robin, and efficiency-only selection over 100 FMNIST rounds, Top-K by C_i gains 12.8%, 18.4%, and 8.6% final accuracy respectively, and uses 22.3% less cumulative energy than random. Utility-per-Joule gain versus random is

$$\text{Gain}_{\text{util/Joule}} = \frac{(\text{Acc}/E_{\text{total}})_{\text{AFL}} - (\text{Acc}/E_{\text{total}})_{\text{Random}}}{(\text{Acc}/E_{\text{total}})_{\text{Random}}} \times 100\%. \tag{12}$$

AFL achieves efficiency gains of $+35.2\%$ on MNIST (96.80% at 214 kJ vs. 94.10% at 289 kJ), $+22.3\%$ on FMNIST (88.40% at 312 kJ vs. 82.10% at 401 kJ), and $+15.4\%$ on CIFAR-10 (63.64% at 580 kJ vs. 56.20% at 672 kJ). The trend reflects task complexity: simpler datasets offer more opportunity to prune inefficient clients, while CIFAR-10 demands broader participation, moderating the gain. Baseline failures follow predictable patterns tied to each method's assumptions, Krum is fooled when FoE keeps pairwise distances small, Median breaks under coordinate-wise constant shifts, and CClip fails when sign-flips remain within the clipping radius. AFL instead requires an update to simultaneously appear plausible across accuracy, alignment, distance bounds, and temporal variance (OR of flags in Algorithm 1), making evasion substantially harder. For $\kappa \in [2.0, 3.0]$, this yields $> 96\%$ Byzantine recall with $< 2.4\%$ false positives on benign non-IID clients ($\alpha_{\text{Dir}} = 0.6$).

6 Conclusion and Limitations

We presented AFL, a decentralized federated learning framework integrating energy efficiency, incentive alignment, trust management, Byzantine robustness, and autonomous coordination. Our results demonstrate reliable and energy-efficient learning even under adversarial conditions. Several promising directions remain open: the logically centralized coordinator could be replaced by a rotating high-trust committee using lightweight BFT voting over validation-set shards (Appendix C); the threshold-based participation policy could be made adaptive via multi-agent reinforcement learning; and further extensions include differential privacy integration with the MAD detector, carbon-aware scheduling, and on-device LLM agents operating under Proof-of-Useful-Learning (PoUL) constraints.

AI USE STATEMENT

In this work, we used generative AI tools for grammar correction, sentence-structure improvements, and proofreading of portions of the manuscript. AI tools were not used to generate the core research contributions, experimental methodology, results, or conclusions. We also used generative AI tools to review and test parts of the codebase, primarily to identify potential issues and improve code clarity and readability for users. All AI-assisted text and code suggestions were manually reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To support reproducibility, the complete codebase used in our experiments is provided as supplementary material. The repository includes a README with the required environment setup, dependencies, configuration, and instructions for running the experiments. The implementation and experimental details described in the main paper and appendix are provided alongside the code to facilitate independent reproduction and verification of our results.

REFERENCES

G. Afifi and B. Mokhtar. A federated meta-transfer learning framework for agentic vehicular edge-of-things computing-based resource allocation and task offloading. *Ad Hoc Netw.*, 2025.

AI Sweden. Federated learning in banking. [Online]. Available: `https://www.ai.se/en/project/federated-learning-banking`, 2023.

G. Baruch, M. Baruch, and Y. Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Inf. Process. Syst. (NeurIPS)*, volume 32, 2019.

P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, pp. 118–128. Curran Associates Inc., 2017.

X. Cao, M. Fang, J. Liu, and N. Z. Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2021. doi: 10.14722/ndss.2021.24434.

D. Chakraborty. Understanding federated learning in AI/ML development. Online, 2022.

F. Cucker and S. Smale. On the mathematics of emergence. *Jpn. J. Math.*, 2(1):197–227, March 2007.

I. Desk. Unveiling federated learning in the AI landscape. Online, September 2023.

A. Dorri, S. S. Kanhere, and R. Jurdak. Towards an optimized BlockChain for IoT. In *Proc. IEEE/ACM 2nd Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, pp. 173–178, 2017.

M. Fang, Z. Zhang, F. Hairi, P. Khanduri, J. Liu, S. Lu, Y. Liu, and N. Gong. Byzantine-robust decentralized federated learning. In *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, pp. 2874–2888, 2024. doi: 10.1145/3658644.3670307. Introduces BALANCE (Byzantine-robust averaging through local similarity in decentralization).

T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. doi: 10.1109/ACCESS.2019.2909068.

J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 6th edition, 2020.

N. R. Jennings. Coordination techniques for distributed artificial intelligence. In *Foundations of Distributed Artificial Intelligence*, pp. 187–210. John Wiley & Sons, New York, NY, USA, 1996.

R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos. Towards efficient data valuation based on the Shapley value. In *Proc. 22nd Int. Conf. Artif. Intell. Statist. (AISTATS)*, volume 89 of *Proc. Mach. Learn. Res.*, pp. 1167–1176. PMLR, 2019.

P. Kairouz et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1–2):1–210, 2021.

J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Trans. Ind. Informat.*, 13(6):3154–3164, 2017. doi: 10.1109/TII.2017.2709784.

J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet Things J.*, 6(6):10700–10714, 2019. doi: 10.1109/JIOT.2019.2940820.

S. P. Karimireddy, L. He, and M. Jaggi. Learning from history for Byzantine robust optimization. In *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, volume 139 of *Proc. Mach. Learn. Res.*, pp. 5311–5319. PMLR, 2021.

H. Kim, J. Park, M. Bennis, and S.-L. Kim. On-device federated learning via blockchain and its latency analysis. *arXiv preprint*, 2018. arXiv:1808.03949.

J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint*, 2016. arXiv:1610.05492.

F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury. Oort: Efficient federated learning via guided participant selection. In *Proc. 15th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, pp. 19–35, 2021.

E. P. Lazear and S. Rosen. Rank-order tournaments as optimum labor contracts. *J. Political Economy*, 89(5):841–864, 1981.

S. Lee, A. K. Sahu, C. He, and S. Avestimehr. Partial model averaging in federated learning: Performance guarantees and benefits. Amazon Science, 2022.

T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.*, 37(3):50–60, 2020.

Y. Li, H. Wang, Y. Li, J. Li, and Z. Dong. ZTFed-MAS2S: A zero-trust federated learning framework with verifiable privacy and trust-aware aggregation for wind power data imputation. *IEEE Trans. Ind. Informat.*, 2025. arXiv:2508.18318.

W. Y. B. Lim, Z. Xiong, C. Miao, D. Niyato, Q. Yang, C. Leung, and H. V. Poor. Hierarchical incentive mechanism design for federated machine learning in mobile networks. *IEEE Internet Things J.*, 7(10):9575–9588, 2020. doi: 10.1109/JIOT.2020.2985694.

Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu. FedCoin: A peer-to-peer payment system for federated learning. In *Federated Learning: Privacy and Incentive*, pp. 125–138. Springer, 2020a. doi: 10.1007/978-3-030-63076-8_9.

Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. Abd El-Latif. A secure federated learning framework for 5G networks. *IEEE Wireless Commun.*, 27(4):24–31, 2020b. doi: 10.1109/MWC.01.1900525.

Z. Luan, W. Li, M. Liu, and B. Chen. Robust federated learning: Maximum correntropy aggregation against Byzantine attacks. *IEEE Trans. Neural Netw. Learn. Syst.*, 36:62–75, 2024.

Lucinity. Federated learning for secure data sharing in FinCrime. Online, November 2024.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, pp. 1273–1282, 2017.

V. Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. [Online]. Available: https://bitcoin.org/bitcoin.pdf, 2008.

S. K. Nanda, S. K. Panda, and M. Dash. Medical supply chain integrated with blockchain and IoT to track the logistics of medical products. *Multimedia Tools Appl.*, pp. 1–23, March 2023.

NVIDIA. Using federated learning to bridge data silos in financial services. Online, August 2022.

L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Auton. Agents Multi-Agent Syst.*, 11(3):387–434, November 2005.

K. Pillutla, S. M. Kakade, and Z. Harchaoui. Robust aggregation for federated learning. *IEEE Trans. Signal Process.*, 70:1142–1154, 2022. doi: 10.1109/TSP.2022.3153135.

X. Qiu, T. Parcollet, D. Raber, and N. D. Lane. A first look at the carbon footprint of federated learning. *J. Mach. Learn. Res.*, 24(129):1–23, 2023.

T. Rodriguez. How can Substra ensure privacy while enabling AI collaboration? [Online]. Available: https://jelly.aidialoguehub.com/owkin-substra.html, 2025.

A. Singh, P. Chari, C. Lu, G. Gupta, A. Chopra, J. Blanc, T. Klinghoffer, K. Tiwary, and R. Raskar. A perspective on decentralizing AI. In *MIT Media Lab*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:278308967.

M. Singh and S. Kim. Blockchain based intelligent vehicle data sharing framework. *arXiv preprint*, 2017. arXiv:1708.09721.

T. Song, Y. Tong, and S. Wei. Profit allocation for federated learning. In *Proc. IEEE Int. Conf. Big Data*, pp. 2577–2586, 2019. doi: 10.1109/BigData47090.2019.9006327.

Splunk. Federated learning in AI: How it works, benefits and challenges. [Online]. Available: https://www.splunk.com/en_us/blog/learn/federated-ai.html, 2023.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In *Proc. Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2019.

Z. Sun, J. Wan, L. Yin, Z. Cao, T. Luo, and B. Wang. A blockchain-based audit approach for encrypted data in federated learning. *Digit. Commun. Netw.*, 8(5):614–624, 2022. doi: 10.1016/j.dcan.2022.05.006.

H. Taherdoost. The role of blockchain in medical data sharing. *Cryptography*, 7(3):36, 2023. doi: 10.3390/cryptography7030036.

A. Takyar. Federated learning: Unlocking the potential of secure distributed AI. Online, 2024.

Y. Wang, J. Zhou, G. Feng, X. Niu, and S. Qin. Blockchain assisted federated learning for enabling network edge intelligence. *IEEE Netw.*, 37(1):96–102, 2023. doi: 10.1109/MNET.115.2200014.

P. Wu, K. Li, J. Nan, and F. Wang. Federated in-context LLM agent learning. *arXiv preprint*, 2024a. arXiv:2412.08054v1.

Z. Wu, M. M. Amiri, R. Raskar, and B. K. H. Low. Incentive-aware federated learning with training-time model rewards. In *Proc. 12th Int. Conf. Learn. Representations (ICLR)*, 2024b. [Online]. Available: https://openreview.net/forum?id=uN6J6aL7H6.

Y. Xia, T. Jahani-Nezhad, and R. Bitar. Fed-DPRoC: Communication-efficient differentially private and robust federated learning. In *Proc. 3rd Int. Conf. Federated Learn. Technol. Appl. (FLTA)*, 2025. Extended version: arXiv:2508.12978.

C. Xie, S. Koyejo, and I. Gupta. Generalized Byzantine-tolerant SGD. *arXiv preprint*, 2018. arXiv:1802.10116.

C. Xie, O. Koyejo, and I. Gupta. Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation. In *Proc. 36th Conf. Uncertainty Artif. Intell. (UAI)*, volume 115, pp. 261–270. PMLR, 2020.

J. Xu, C. Zhang, J. Liu, and C. Su. A trust-aware incentive mechanism for federated learning with heterogeneous clients in edge computing. *J. Cybersecur. Priv.*, 5(3):37, 2025. doi: 10.3390/jcp5030037.

Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning. *ACM Trans. Intell. Syst. Technol.*, 10(2):1–19, March 2019.

T. Yoon, S. Choudhury, and N. Loizou. Multiplayer federated learning: Reaching equilibrium with less communication. *arXiv preprint*, 2025. arXiv:2501.08263v2.

Q. Zhang, Q. Ding, J. Zhu, and D. Li. Blockchain empowered reliable federated learning by worker selection: A trustworthy reputation evaluation method. In *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, pp. 1–6, 2021a. doi: 10.1109/WCNCW49093.2021.9420026.

W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, S. Chen, X. Xu, and L. Zhu. Blockchain-based federated learning for device failure detection in industrial IoT. *IEEE Internet Things J.*, 8(7):5926–5937, 2021b. doi: 10.1109/JIOT.2020.3032544.

Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu. Privacy-preserving blockchain-based federated learning for IoT devices. *IEEE Internet Things J.*, 8(3):1817–1829, 2021. doi: 10.1109/JIOT.2020.3017377.

S. Zuo, X. Yan, R. Fan, L. Shen, P. Zhao, J. Xu, and H. Hu. Efficient federated learning against Byzantine attacks and data heterogeneity via aggregating normalized gradients. *arXiv preprint*, 2024. arXiv:2408.09539.

A Proof of Convergence for AFL Aggregation (Theorem 1)

In this section, we provide the complete proof of Theorem 1 establishing the ergodic convergence of AFL's trust-weighted aggregation under non-IID data distributions and Byzantine contamination.

Preliminaries & Aggregation Rule. Let $\mathcal{S}^{(t)}$ denote the set of clients selected at round t with $|\mathcal{S}^{(t)}| = K$. Let $\mathcal{B}^{(t)} \subset \mathcal{S}^{(t)}$ denote the set of Byzantine adversaries admitted into the aggregation pool, and let $\mathcal{H}^{(t)} = \mathcal{S}^{(t)} \setminus \mathcal{B}^{(t)}$ denote honest benign clients. At round t, the coordinator computes normalized trust weights $\widetilde{\tau}_i^{(t)} = \frac{\tau_i^{(t)}}{\sum_{j \in \mathcal{S}^{(t)}} \tau_j^{(t)}}$. The global model update is:

$$w^{(t+1)} = w^{(t)} + \Delta w^{(t)}, \quad \text{where} \quad \Delta w^{(t)} = \sum_{i \in \mathcal{S}^{(t)}} \widetilde{\tau}_i^{(t)} \Delta w_i^{(t)}. \tag{13}$$

For benign clients $i \in \mathcal{H}^{(t)}$, the local update corresponds to stochastic gradient descent: $\Delta w_i^{(t)} = -\eta g_i(w^{(t)})$, where $\mathbb{E}[g_i(w^{(t)})] = \nabla F_i(w^{(t)})$.

Step 1: One-Step Descent under L-Smoothness. By Assumption 1, $F(w)$ is L-smooth. Applying the descent lemma:

$$F(w^{(t+1)}) \le F(w^{(t)}) + \langle \nabla F(w^{(t)}),\ \Delta w^{(t)} \rangle + \frac{L}{2} \|\Delta w^{(t)}\|_2^2. \tag{14}$$

Taking conditional expectations on both sides given $w^{(t)}$:

$$\mathbb{E}[F(w^{(t+1)}) \mid w^{(t)}] \le F(w^{(t)}) + \underbrace{\langle \nabla F(w^{(t)}),\ \mathbb{E}[\Delta w^{(t)}] \rangle}_{A_1} + \frac{L}{2} \underbrace{\mathbb{E}\|\Delta w^{(t)}\|_2^2}_{A_2}. \tag{15}$$

Step 2: Decomposing the Update Step. We decompose the expected update $\Delta w^{(t)}$ into honest and Byzantine components:

$$\Delta w^{(t)} = \sum_{i \in \mathcal{H}^{(t)}} \widetilde{\tau}_i^{(t)} \big(-\eta g_i(w^{(t)}) \big) + \sum_{i \in \mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \Delta w_{adv,i}^{(t)}. \tag{16}$$

Let $\omega_H^{(t)} = \sum_{i \in \mathcal{H}^{(t)}} \widetilde{\tau}_i^{(t)}$ and $\omega_B^{(t)} = \sum_{i \in \mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)}$, such that $\omega_H^{(t)} + \omega_B^{(t)} = 1$. Taking the expectation of A_1:

$$\begin{aligned} A_1 &= \left\langle \nabla F(w^{(t)}),\ -\eta \sum_{i \in \mathcal{H}^{(t)}} \widetilde{\tau}_i^{(t)} \nabla F_i(w^{(t)}) + \sum_{i \in \mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \Delta w_{adv,i}^{(t)} \right\rangle \\ &= -\eta \omega_H^{(t)} \|\nabla F(w^{(t)})\|_2^2 - \eta \sum_{i \in \mathcal{H}^{(t)}} \widetilde{\tau}_i^{(t)} \langle \nabla F(w^{(t)}),\ \nabla F_i(w^{(t)}) - \nabla F(w^{(t)}) \rangle \\ &\quad + \sum_{i \in \mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \langle \nabla F(w^{(t)}),\ \Delta w_{adv,i}^{(t)} \rangle. \end{aligned} \tag{17}$$

Applying Young's inequality $\langle a, b \rangle \le \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$ and the Cauchy-Schwarz inequality:

$$\left| \sum_{i \in \mathcal{H}^{(t)}} \widetilde{\tau}_i^{(t)} \langle \nabla F(w^{(t)}), \nabla F_i(w^{(t)}) - \nabla F(w^{(t)}) \rangle \right| \le \frac{1}{4} \|\nabla F(w^{(t)})\|_2^2 + G^2, \tag{18}$$

where G^2 is the bounded non-IID gradient dissimilarity from Assumption 1. For the Byzantine term:

$$\left| \sum_{i \in \mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \langle \nabla F(w^{(t)}), \Delta w_{adv,i}^{(t)} \rangle \right| \le \frac{\eta}{4} \|\nabla F(w^{(t)})\|_2^2 + \frac{1}{\eta} \sum_{i \in \mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \|\Delta w_{adv,i}^{(t)}\|_2^2. \tag{19}$$

Step 3: Bounding Variance (A_2). Using $\mathbb{E}\|X\|^2 = \|\mathbb{E}[X]\|^2 + \mathbb{E}\|X - \mathbb{E}[X]\|^2$ and bounded variance σ^2:

$$A_2 \le 2\eta^2 \|\nabla F(w^{(t)})\|_2^2 + \frac{2\eta^2 \sigma^2}{K} + 2 \sum_{i \in \mathcal{B}^{(t)}} (\widetilde{\tau}_i^{(t)})^2 \|\Delta w_{adv,i}^{(t)}\|_2^2. \tag{20}$$

Step 4: Telescoping Sum and Convergence Bound. Substituting A_1 and A_2 into Eq. 15 with step size $\eta \leq \frac{1}{4L}$:

$$\frac{\eta}{2}\mathbb{E}\|\nabla F(w^{(t)})\|_2^2 \leq F(w^{(t)}) - \mathbb{E}[F(w^{(t+1)})] + \eta^2 L\sigma^2 + \eta G^2 + \mathcal{O}\left(\sum_{i\in\mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \|\Delta w_{adv,i}^{(t)}\|_2\right). \quad (21)$$

Summing over rounds $t = 0, 1, \ldots, T-1$, dividing by $\frac{\eta T}{2}$, and utilizing $F(w^{(T)}) \geq F^*$:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla F(w^{(t)})\|_2^2 \leq \frac{2\big(F(w^{(0)}) - F^*\big)}{\eta T} + 2\eta L\sigma^2 + 2G^2 + \frac{C_{\text{adv}}}{T}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \|\Delta w_{adv,i}^{(t)}\|_2. \quad (22)$$

By setting $\eta = \frac{1}{L\sqrt{T}}$, the first two terms attain $\mathcal{O}(1/\sqrt{T})$.

Step 5: Trust Contraction of Byzantine Adversaries. Under Algorithm 1, any Byzantine update admitted into the ledger has its perturbation strictly bounded by the MAD acceptance sphere: $\|\Delta w_{adv,i}^{(t)}\|_2 \leq \Theta_{\text{dist}}$. Whenever a malicious client injects significant bias, it is flagged with $\mathbb{I}_i^{(t)} = 1$. By Eq. 4, its trust score contracts according to:

$$\tau_i^{(t)} \leq (1 - \mu_{\text{trust}})\,\tau_i^{(t-1)} \implies \tau_i^{(t)} \leq (1 - \mu_{\text{trust}})^t \tau_i^{(0)}. \quad (23)$$

Because $(1 - \mu_{\text{trust}}) \in (0, 1)$, $\tau_i^{(t)} \to 0$ exponentially fast. Consequently, the cumulative adversarial contamination term satisfies:

$$\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i\in\mathcal{B}^{(t)}} \widetilde{\tau}_i^{(t)} \|\Delta w_{adv,i}^{(t)}\|_2 \leq \frac{\Theta_{\text{dist}}}{T}\sum_{t=0}^{\infty}(1 - \mu_{\text{trust}})^t = \frac{\Theta_{\text{dist}}}{\mu_{\text{trust}}\, T} = \mathcal{O}\left(\frac{1}{T}\right). \quad (24)$$

Hence, as $T \to \infty$, the Byzantine bias term decays at rate $\mathcal{O}(1/T)$, asymptotically vanishing and recovering the optimal clean federated convergence rate $\mathcal{O}(1/\sqrt{T})$. ■

B Game-Theoretic Incentive Compatibility of PoUL (Proposition 1)

Here we formalize the microeconomic properties of the Proof-of-Useful-Learning (PoUL) mechanism grounded in rank-order tournament theory Lazear & Rosen (1981).

Game Formulation. Consider K client agents selected in squad $\mathcal{S}^{(t)}$. Each agent i chooses an effort level $e_i \geq 0$, representing local compute epochs and batch sizes. Effort incurs a private physical energy cost $\mathcal{C}_i(e_i) = P_i \Delta t_i(e_i) \geq 0$, strictly increasing and convex in e_i. The resulting model accuracy improvement on the true data distribution is $\Delta\text{acc}_i = \mu(e_i) + \xi_i$, where $\mu(\cdot)$ is concave and strictly increasing in effort, and $\xi_i \sim \mathcal{N}(0, \sigma_\xi^2)$ is local stochastic noise.

The net expected payoff of agent i is:

$$U_i(e_i; e_{-i}) = \mathbb{E}\left[R_i(e_i)\right] - \mathcal{C}_i(e_i), \quad (25)$$

where $R_i = w_{\text{acc}}\Delta\text{acc}_i + w_{\text{eng}}\frac{\eta_i}{\max_j \eta_j}\mathbb{I}(\Delta\text{acc}_i > 0)$ (Eq. 6).

Analysis of Strategic Deviations.

1. **Free-Riding Strategy ($e_i = 0$):** A free-rider submits $w_i^{(t)} = w^{(t)}$ to incur $\mathcal{C}_i(0) = 0$. In this case, $\|w_i^{(t)} - w^{(t)}\|_2 = 0 < \Theta_{\text{dist,min}} = \epsilon_{\text{dist}}$. Algorithm 1 triggers flag $\mathbb{I}_i^{(t)} = 1$. By Eq. 4, trust contracts ($\tau_i \to 0$), deterministically excluding the agent from future squads. Furthermore, $\Delta\text{acc}_i \leq 0$, which zeros both the accuracy reward and efficiency bonus ($R_i = 0$). Hence, $U_i(0) = 0$, while truthful effort yields $U_i(e^*) > 0$, eliminating free-riding as a viable strategy.

2. **Strategic Overfitting Strategy:** An adversary trains locally to minimize loss on $\mathcal{D}_i$ without generalization. Because the coordinator audits candidate models strictly on a private, held-out validation set $\mathcal{D}_{\text{val}}$, by standard generalization bounds:

$$\mathbb{E}\left[\text{Acc}(w_i; \mathcal{D}_{\text{val}})\right] \leq \text{Acc}(w_i; \mathcal{D}_i) - \mathcal{O}\left(\sqrt{\frac{d_{\text{VC}}}{|\mathcal{D}_i|}}\right). \tag{26}$$

Absent knowledge of $\mathcal{D}_{\text{val}}$, overfitting to $\mathcal{D}_i$ yields negative marginal utility on $\mathcal{D}_{\text{val}}$ ($\Delta\text{acc}_i < 0$), triggering $\mathbb{I}_i^{(t)} = 1$ and zeroing rewards.

3. **Contest Tournament Equilibrium:** Under relative efficiency $\frac{\eta_i}{\max_j \eta_j}$, each agent solves $\max_{e_i} w_{\text{eng}} \mathbb{P}(\eta_i > \max_{j \neq i} \eta_j) + w_{\text{acc}} \mu(e_i) - \mathcal{C}_i(e_i)$. By Theorem 1 of Lazear & Rosen Lazear & Rosen (1981), rank-order tournaments over verifiable outputs produce a unique symmetric Nash equilibrium where each agent expends socially optimal effort $e_i^* > 0$, maximizing system-wide utility-per-Joule. ■

C Decentralized Committee Coordinator Protocol

To address the single point of failure in centralized coordination, Algorithm 2 specifies the decentralized committee workflow.

Algorithm 2 Decentralized Committee Coordinator Rotation Protocol

Require: Client set $\mathcal{C}$, trust ledger $\{\tau_i^{(t-1)}\}$, validation dataset $\mathcal{D}_{\text{val}}$, committee threshold θ_{comm}
Ensure: Consensus aggregated global model $w^{(t+1)}$

Committee Election:
Form committee $\mathcal{M}^{(t)} \leftarrow \text{Top-}M \left\{ \tau_i^{(t-1)} \;\middle|\; \tau_i^{(t-1)} > \theta_{\text{comm}} \right\}$
Select active training squad $\mathcal{S}^{(t)} \leftarrow \text{Top-}K\{C_i^{(t)}\}$ from remaining clients $\mathcal{C} \setminus \mathcal{M}^{(t)}$
Secret-Shared Auditing:
Partition $\mathcal{D}_{\text{val}}$ into M disjoint subsets $\{\mathcal{D}_{\text{val},m}\}_{m=1}^{M}$ distributed to committee members
for each training client $i \in \mathcal{S}^{(t)}$ concurrently **do**
    Broadcast candidate update $w_i^{(t)}$ to committee $\mathcal{M}^{(t)}$
end for
for each committee member $m \in \mathcal{M}^{(t)}$ concurrently **do**
    Evaluate candidate weights on partition $\mathcal{D}_{\text{val},m}$
    Compute local MAD thresholds $\Theta_{\text{acc},m}, \Theta_{\text{dist},m}, \Theta_{\text{cos},m}$
    Generate vote vector $\mathbf{v}_m \in \{0,1\}^K$, where $v_{m,i} = 0$ if client i is flagged as Byzantine
    Sign and broadcast vote $\text{Sign}_m(\mathbf{v}_m)$ to committee peers
end for
BFT Consensus & Aggregation:
Each committee member aggregates votes: $V_i = \sum_{m \in \mathcal{M}^{(t)}} v_{m,i}$
Valid client set: $\mathcal{S}_{\text{clean}}^{(t)} \leftarrow \left\{ i \in \mathcal{S}^{(t)} \;\middle|\; V_i \geq \lfloor \frac{2M}{3} \rfloor + 1 \right\}$
Compute consensus weights: $w^{(t+1)} \leftarrow w^{(t)} + \sum_{i \in \mathcal{S}_{\text{clean}}^{(t)}} \widetilde{\tau}_i^{(t)} (w_i^{(t)} - w^{(t)})$
Commit state update to persistent distributed ledger via threshold multi-signature
return $w^{(t+1)}$

D Physical Energy Bounds and Hardware Attestation

CMOS power model. Dynamic and leakage power for digital logic are commonly written

$$P = \alpha_{\text{gate}} C_{\text{load}} V_{\text{dd}}^2 f_{\text{clk}} + I_{\text{leak}} V_{\text{dd}}. \tag{27}$$

For a local epoch with M_{FLOPs} operations we use a coarse lower bound $E_{\min} \gtrsim M_{\text{FLOPs}} \cdot \text{EPR}$, where EPR is an empirical energy-per-operation for the declared hardware class (not a Landauer limit). Reports with $E_i^{(t)} \ll E_{\min}$ are treated as Stage 3 anomalies.

Hardware roots of trust. Where available, energy should come from processor counters rather than a user-space guess:

- Intel RAPL / AMD package energy MSRs updated by on-die power management;
- TPM-backed attestation of those readings so the coordinator can reject unsigned telemetry.