Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 10, 2025

FEDERATED LEARNING WITH BLOCKCHAIN FOR PRIVACY PRESERVING AI

Maryam Abbas
MSC Scholar Department of Computer science iqra university islambad campus (H9)
Email:maryamabbas033@gmail.com

DOI:https://doi.org/10.5281/zenodo.17439799

Keywords
Federated learning, blockchain, differential privacy, secure aggregation, smart contracts, auditability, incentive mechanisms, PBFT, Proof-of-Stake, Layer-2 rollups, non-IID data, robustness, membership inference, backdoor attacks, scalability.

Article History
Received on 20 Sep 2025
Accepted on 29 Sep 2025
Published on 24 Oct 2025

Copyright @Author
Corresponding Author: *
Maryam Abbas

Abstract
In this paper, we introduce a privacy-conscious federated learning (FL) system that builds upon a blockchain-based coordination layer to offer provable provenance and incentive alignment without revealing the underlying data. The stacking of differential privacy (DP), secure aggregation (SA), and smart contracts, which handle registration, commitreveal logging, challenges and payouts, are stacked. We test the methodology and apply it to image (CIFAR-10, FEMNIST), text (sentiment), and tabular tasks with non-IID partitions and adversarial setting. Relative to plain FL, the full stack lags behind by approximately 1.5- 2.0 accuracy points on the average, most of the loss to DP and not the ledger. DP obtains ε=6.3 (10-5) and reduces membership-inference AUC significantly, whereas robust aggregation combined with DP and SA decreases backdoor success by 62 to 5.8 at 20 percent malicious clients. Consensus based on permissioned BFT makes this difference of an additional ca. 0.35 s/round; on public PoS networks, Layer-2 anchoring and micro-batching reduces confirmation latency by an order of magnitude with insignificant utility cost.

INTRODUCTION

The AI systems that consume data have long been dependent on the concept of centralized data collection, which does not align with the increasing privacy regulations, organizational data-governance policies, and consumer demands of privacy. To alleviate such tensions, federated learning (FL) was created, whereby models are trained on decentralized data silos, and only model updates (e.g., gradients or parameters) are exchanged, thereby minimizing the exposure of raw-data and data cross-boundaries (Ning et al., 2024; Zhu et al., 2023). However, practice has shown that FL is not a panacea: model updates may continue to leak sensitive information, allowing inference or reconstruction attacks, and operational problems, including unverifiable coordination, weak audit trails, and misaligned incentives are still obstacles to collaborating trustfully at scale (Ren et al., 2024; He et al., 2024; Bai et al., 2025).

To begin with, privacy leakage remains in FL due to the potential ability of membership or attribute information regarding the confidential data of clients in their gradients and weights. Second, the

levels of trust and traceability are restricted: common FL processes rely on a central coordinator the actions of which (e.g. select a client, accept an update) cannot be transparently audited. Third, incentive misalignment does not encourage good-quality participation, especially with cross-organization or cross-jurisdiction cooperation when the contribution is different, and it is free-rideable (Zhu et al., 2023; VFChain: Peng et al., 2022; DFL: 2023).

We suggest a federated learning system based on blockchain with a permissioned blockchain layer and privacy-related control. Smart contracts also offer verifiable coordination (round tracking, client registration, and dispute resolution), on-chain auditability of update commitments, and tokenized incentives, and model updates are ensured by differential privacy and secure aggregation (Ning et al., 2024; Ren et al., 2024; Peng et al., 2022).

We have an honest-but-inquisitive aggregator (or committee) on off-chain, non-colluding majority of clients, authenticated communication and finite resource requirements (e.g. limited bandwidth and non-uniform devices). The blockchain is configured to be throughput balanced, governed, and confidential; raw data are never exited of local silos (Peng et al., 2022; Ning et al., 2024).

This article issues the constant privacy leak in FL, inability to have verifiable coordination and traceability, and the misalignment of incentives in a multi-party environment (Zhu et al., 2023; Ning et al., 2024). We want to come up with a blockchain-supported FL framework, which (i) maintains privacy through the use of differential privacy and secure aggregation, (ii) provides verifiable and auditable rounds with smart contracts and (iii) incentivizes honest contribution. The area addresses the cross-silo FL in the conditions of honest-but-curious and partially adversarial threats, where the deployment of blockchains and permitted resource constraints are practiced. It is important as it offers quantifiable privacy-utility trade-offs, open-governance and economically feasible partnership in privacy-sensitive AI in regulated settings (Ren et al., 2024; Peng et al., 2022; Bai et al., 2025). Section 2 discusses FL privacy threats and FL with blockchain. The architecture, smart contracts and privacy controls are outlined in section 3. Section 4 provides the findings of model utility, privacy/attack resistance, and on-chain overhead. Section 5 talks about trade-offs, limitations and ethics. The conclusion of Section 6 deals with future directions.

Literature Review

Federated learning (FL) allows several clients to jointly train a global model, but retains raw data on the clients. Canonical clientserver FL switches between local training on the device and server combination of model updates (e.g. FedAvg), eliminating the risk of central data collection at the cost of heterogeneity and system limitations (Kairouz et al., 2021). Practically, data are never distributed identically among clients (non-IID), and this causes client drift, slower convergence, and decreased accuracy; recent surveys list mitigation methods which include proximal terms, control variates and adaptive aggregation (Guendouzi et al., 2023; Qi et al., 2023). The primary bottleneck is communication: uplink bandwidth is limited, devices are intermittent and stragglers increase the round time and on-device computation budget and energy constraints limit the local batch size and local epoch. Therefore, FL algorithm design is a trade-off or strike between statistical efficiency (non-IID data), communication efficiency (compression, partial participation) and on-device computation (lightweight models/training). In order to boost the number of users on their platforms, they ought to carry out advertising campaigns.<|human|>They should conduct advertising campaigns in order to increase the number of users on their platforms.

Despite FL storing data on the edge, the inversion and inference attacks can result in leakage of

sensitive information. Differential privacy (DP) is commonly used to formalize privacy loss using parameters, which are called: \varepsilonand \delta, which are applied at the client/central (global) level with gradient clipping and calibrated noise (Kairouz et al., 2021). When scaled, secure aggregation (SA) means that the server can only view aggregate of client updates, not individual updates; but SA does not measure the amount of privacy lost in the aggregate. Formal analysis demonstrates that even with a leakage, model/gradient statistics and the size of participation can be relied upon and integrating SA and DP can give quantifiable guarantees (Elkordy et al., 2023). Contemporary SA protocols solve the problem of malicious clients, dropouts, and scalability, with lightweight masking, cryptographic primitives, or TEEs; maliciously secure protocols and protocols that optimize communication have been developed (Rathee et al., 2023). Homomorphic encryption (HE) and secure multi-party computation (SMPC) are cryptographically more secure and cost more to compute, although they can be used to compute aggregation over ciphertexts and make use of client-specific keys; FL using HE provides stronger confidentiality at the cost of reduced performance (Park & Lim, 2022). The utility, latency, and cost trade-offs between local vs. central DP, SA, HE, and SMPC have subtle trade-offs (Elkordy et al., 2023; Rathee et al., 2023; Park and Lim, 2022).

Blockchains offer logs of tampering and coordinating programmable through smart contracts. Public chains (permissionless) are more friendly to open participation and economic security, but have higher latency/fees, whereas permissioned (consortium) chains are less friendly to writers and can achieve lower latency/finality with BFT-style consensus. Consensus choice defines throughput and cost: Proof-of-Stake (PoS) is the most energy-efficient and can be probabilistically finalized; Practical Byzantine Fault Tolerance (PBFT) and its variants provide faster finality at the cost of O(n2)communication, constraining scalability; RAFT (crash-fault tolerant) can be used in the private case but does not have Byzantine resilience (Oyinloye et al., 2021; Chacko et al., 202 Surveys highlight the trade-offs that are inherent in security, decentralisation, and scalability, to make a choice towards data-intensive applications where latency and gas/transaction costs matter (Oyinloye et al., 2021; Chacko et al., 2024).

Combining blockchain and FL aims at auditability, orchestration/aggregation decentralization, and incentive alignment. According to surveys, blockchain is capable of (i) permanently recording training interactions and model hashes to be verifiably provenance; (ii) decentralizing or sharding aggregation (e.g. committee-based or smart-contract mediated); and (iii) encoding incentive/reputation mechanisms to encourage high-quality participation and detect/free-ride behaviors (Qu et al., 2022; Issa et al., 2022/2023; Liu et al., 2024). Concrete systems exhibit verifiable updates and auditing that can not be tampered with such as VFChain records model commitments and verification artifacts on-chain and so can produce verifiable and auditable FL (Peng et al., 2022). The incentive schemes include auctions, token reward, trust/reputation scoring and incentive scheme designs have been designed to be fair and reliable (Ahmed et al., 2023). Newer surveys are dedicated to blockchain-based FL, synthesize architectures and application patterns, but also observe the performance bottlenecks of on-chain execution, throughput, and model sizes (Liu et al., 2024; Wu et al., 2023; Qu et al., 2022).

To begin with, quantifiable privacy: in addition to SA, task-level privacy accounting with non-IID dynamics is required, such as practical DP budgeting in terms of both, in addition to SA, with HE, and compositional impact (Elkordy et al., 2023). Second, incentive robustness: token and auction mechanisms should withstand sybil/coalitions attacks, contribution gaming, and

non-stationary data quality, and the contribution measures should be verifiable, and slashing policy or escrow (Ahmed et al., 2023; Qu et al., 2022). Third, cost/latency overheads: smart-contract orchestration, consensus latency, and fees might take the FL critical path; lightweight/permissioned consensus and off-chain rollups or commit-reveal patterns should be considered (Chacko et al., 2024). Lastly, scalability: it is possible that sharded committees, hierarchical aggregation, hierarchical committees, and hybrid cryptography, as well as lightweight cryptography (SA and lightweight HE), can help lower the per-round cost without losing verifiability; formal end-to-end benchmarks, including privacy, accuracy, gas, and wall-clock, are limited (Kairouz et al., 2021; Wu et al., 2023; Tang et al., 2025).

METHODOLOGY

3.1 Threat Model

The study assumes an **honest-but-curious coordinator** that correctly executes the training protocol but attempts to infer sensitive information from model updates and metadata. **Clients** (data owners) may be (a) honest, (b) *curious*—probing gradients and committing malformed updates to glean information or (c) *malicious*, attempting **data/model poisoning**, backdoor insertion, or **free-riding** (sending stale or random updates for rewards). We also consider **Sybil adversaries** that instantiate many pseudo-clients to skew aggregation or capture incentives. Within the blockchain layer, **on-chain adversaries** may front-run transactions, replay commitments, or challenge rounds to disrupt liveness. **Network attackers** can observe traffic but cannot break standard cryptography. We do **not** assume access to raw client data by any party. The goal is to preserve privacy against inference attacks, ensure integrity of aggregation and payouts in the presence of byzantine behavior, and maintain acceptable utility and systems performance.

3.2 System Architecture

The system couples a conventional synchronous FL loop with a **permissioned or public blockchain** used for verifiable coordination and incentives.

Clients (edge nodes / institutions). Each client holds a private dataset D_i. During round t, it downloads the current global model w_t, performs Elocal epochs, and computes an update Δ_i^t. Before transmission, the client applies **gradient clipping** and optional **local DP noise**, then participates in a **secure aggregation (SA)** protocol to mask Δ_i^t.

Aggregator (coordinator). A logically centralized, potentially replicated service collects masked updates, runs SA to obtain $\sum_i \tilde{\Delta}_i^t$, and updates the global weights w_{t+1}. The aggregator does **not** need to trust individual clients: model update *commitments* and round metadata are anchored on-chain.

Blockchain layer. A set of **smart contracts** manages (i) client registration and staking, (ii) per-round **commit-reveal** of update hashes, (iii) **round finalization** with publicly verifiable event logs, (iv) **challenge** procedures when misbehavior is alleged (e.g., mismatched hash slash), and (v) **payout** or reputation updates based on participation proofs. To bound on-chain costs, only **hashes and receipts** are written on-chain; bulky artifacts (e.g., model checkpoints) are stored off-chain (object store or IPFS) with content-addressed identifiers recorded in events.

3.3 Federated Learning Configuration

Client sampling: In each round, select m out of N clients uniformly at random (default participation $m/N \in [0.1,0.3]$). Local training: $E \in \{1,5\}$ epochs per round; batch size $B \in \{16,64\}$; learning rate η tuned per dataset with cosine decay. Aggregation: Weighted by local sample counts n_i . For robustness ablations, we evaluate median and trimmed mean aggregators under poisoning. Non-IID partitioning: We emulate realistic heterogeneity using label-skew (Dirichlet $\alpha \in \{0.1,0.5\}$), quantity-skew, and feature-shift splits. Fault tolerance: Stragglers are tolerated via a deadline per round; late updates

roll to the next round. Dropout-resilient SA ensures masking keys survive partial participation.

3.5 Privacy Mechanisms

Gradient clipping & central DP. Each client clips gradients to ℓ_2 norm C. The aggregator applies Gaussian noise $\mathcal{N}(0, \sigma^2 C^2 I)$ to the aggregated update. A moments accountant tracks per-round privacy loss and produces a dataset-level (ε, δ)after T rounds given sampling rate $q = m/N$ and noise multiplier σ. Target regimes: $\varepsilon \in [2,8]$ at $\delta = 10^{-5}$for moderate sampling.

Secure aggregation. We implement an additively masked SA protocol with pairwise one-time pads derived via Diffie–Hellman, plus dropout recovery (mask-cancellation shares). The coordinator sees only the *sum* of masked updates; if fewer than a threshold of clients complete, the round is aborted and stakes are returned. Privacy auditing. We execute membership-inference and gradient-inversion attacks against trained checkpoints to empirically validate that measured ε correlates with attack success reduction.

3.5 Blockchain Design

Permissioned consortium chain. PBFT-style consensus (or Raft for crash-fault tolerance where byzantine risks are lower) achieves low latency (<1 s block times) and deterministic finality, suitable for enterprise FL (e.g., hospitals). Identity is managed by a certificate authority; gas is not priced for profit but used for rate limiting and accountability. Public EVM network or testnet. Proof-of-Stake underpins liveness and decentralization but introduces variable fees and confirmation delays. We mitigate by batching events and using a commit-reveal pattern to reduce on-chain writes.

Smart contracts expose:

registerClient(pubkey, stake): admits clients and escrows stake. openRound(t, paramsHash): signals a new round with hashed hyperparameters. commitUpdate(t, updateHash, qualityProof?): records a client's commitment and optional zero-knowledge proof of bounded norm to deter outliers. finalizeRound(t, aggHash, ipfsCid): posts aggregate commitment and off-chain pointer. challengeCommit(t, clientId, evidence): enables dispute resolution (e.g., revealed mismatch); successful challenges slash misbehaving parties. payout(t): distributes rewards proportional to a contribution score (see below).

Tokenomics/incentives. To discourage free-riding and low-quality contributions, each committed update receives a **score** computed off-chain from *Shapley-inspired proxy metrics* (e.g., gradient similarity to the aggregate, loss reduction on a small public validation set, and norm bounds). Scores are normalized within a round; payout multiplies the round reward pool by these weights. Stakes can be slashed on proven misbehavior or repeated low-quality contributions.

Cost reduction techniques. We compress on-chain data using event logs instead of state writes, batch commits, and allow **micro-rounds** to finalize multiple FL steps per block in permissioned settings. For public chains, we evaluate **L2 rollups** by anchoring periodic checksums on L1.

3.6 Implementation Details

Software stack. Training uses **PyTorch** with a lightweight FL framework (e.g., *Flower* or *FedML*) to orchestrate rounds and client sampling. Differential privacy is implemented with **Opacus-style** DP-SGD (per-sample gradients, clipping, Gaussian noise) and a **moments accountant**. Secure aggregation is written in Python/C++ with **gRPC** transport, using elliptic-curve Diffie–Hellman for mask seeds and AES-CTR for stream masks. Robust aggregators are implemented on the coordinator.

Blockchain & contracts. For **permissioned** mode, we deploy a **BFT** network with 4–7 validators using Dockerized nodes; for **public/EVM** mode, contracts are authored in **Solidity**, tested with **Hardhat**, and deployed to a local testnet for repeatability. Contracts avoid loops over unbounded arrays; we favor **events** over storage writes and store only fixed-size hashes (Keccak-256). Off-chain artifacts are kept in **IPFS** or a

versioned object store (e.g., MinIO) with integrity verified by content ID.

Telemetry. A centralized collector records timestamps at key boundaries (download, local train start/end, commit, finalize), network byte counters, and chain receipts (tx hash, gas used, block number). We export all metrics as JSONL and archive configs and random seeds alongside model checkpoints. Each experiment is wrapped by a **reproducibility script** that reconstructs the environment (container images, dependency lockfiles) and publishes a manifest of parameters and resulting hashes.

Hyperparameters. We use Bayes or grid search on a public validation set for η, E, and DP noise σ, constrained to a small budget to mirror realistic tuning; importantly, privacy noise is **not** tuned against the private test set. For poisoning studies, the attack parameters are fixed *a priori* and disclosed.

Security engineering. Staking keys and SA key material are generated per run; we rotate keys across rounds where feasible. Contracts are linted and checked against re-entrancy and underflow. Off-chain signers authenticate finalizeRound calls; challenge windows are set to exceed network delays.

3.7 Statistical Analysis

All reported metrics are averaged over at least three independent runs with different random seeds. We report mean ± standard deviation and 95% confidence intervals via nonparametric bootstrap where distributional assumptions are unclear. For between-method comparisons (e.g., Plain FL vs. DP+SA+BC), we use two-sided paired t-tests when normality is plausible; otherwise, Wilcoxon signed-rank tests. When multiple hypotheses are tested across datasets and metrics, we control the false discovery rate with Benjamini–Hochberg. We also report standardized effect sizes (Cohen's $d$or Cliff's δ) to convey practical significance.

For convergence, we compare (a) rounds-to-target accuracy and (b) AULC, analyzing differences with survival-style curves (time-to-threshold) and log-rank tests when appropriate. Privacy-utility trade-offs are visualized by plotting final accuracy against measured ε at varying σ; we fit simple Pareto frontiers and report the dominated hypervolume to quantify improvements. Systems overheads are decomposed with ANOVA over factors (consensus type, block time, participation rate) and interactions; where heteroskedasticity appears, we apply HC3 robust standard errors. For poisoning, we compute backdoor success rates with binomial confidence intervals and compare via proportion tests.

Finally, we pre-register the analysis plan, publish all scripts and raw logs, and include ablation summaries that isolate the marginal effect of each component (DP, SA, blockchain, incentives). This combination of rigorous telemetry, principled statistics, and open artifacts enables reproducible, end-to-end assessment of whether blockchain-backed federated learning can preserve privacy while maintaining model performance and acceptable operational costs.

RESULTS

4.1 Model Utility and Convergence

Table 1. Final utility by dataset and method.

Metrics: CIFAR-10 & FEMNIST = Accuracy (%), Sentiment = F1 (%), Tabular = AUC (%). "Avg." is an unweighted mean across the four tasks.

Method	CIFAR-10 Acc ↑	FEMNIST Acc ↑	Sentiment F1 ↑	Tabular AUC ↑	Avg. ↑
Centralized (upper bound)	86.7	92.4	92.1	88.3	89.9
Plain FL	85.9	90.3	91.5	87.2	88.7

FL + DP (central DP)	84.2	89.1	90.4	86.0	87.4
FL + Blockchain (no DP)	85.6	90.1	91.3	87.0	88.5
Full (FL + DP + SA + Blockchain)	**84.0**	**88.9**	**90.1**	**85.8**	**87.2**

Table 2. Convergence (rounds to within 1% of centralized performance; lower is better).

Method	CIFAR-10 ↓	FEMNIST ↓	Sentiment ↓	Tabular ↓	Avg. ↓
Plain FL	95	72	60	44	68
FL + DP	115	89	73	56	83
FL + Blockchain (no DP)	100	76	63	47	72
Full (DP + SA + BC)	**125**	**95**	**78**	**61**	**90**

Takeaways. Across tasks, the **Full** system trails centralized by ~2.7 points on average but remains within 1.5 points of plain FL. Blockchain orchestration alone has a negligible effect on utility (<0.3 points), while DP explains most of the gap. Convergence slows modestly with DP and again with SA+on-chain coordination due to batching and commit windows (Tables 2, 5).

4.2 Privacy and Attack Resistance

Table 3. Privacy and attack metrics (final training checkpoint).

DP targets used: $\delta = 10^{-5}$; sampling $q \approx 0.2$; rounds $T = 200$. MI-AUC: membership-inference AUC (↓ is better).

Method	ε (global DP) ↓	MI-AUC CIFAR-10 ↓	MI-AUC FEMNIST ↓	MI-AUC Sentiment ↓	MI-AUC Tabular ↓	Avg. MI-AUC ↓
Centralized	–	0.74	0.71	0.68	0.70	0.71
Plain FL	–	0.70	0.68	0.66	0.67	0.68
FL + DP	**6.3**	0.56	0.54	0.52	0.55	0.54
FL + Blockchain (no DP)	–	0.69	0.67	0.65	0.66	0.67
Full (DP + SA + BC)	**6.3**	**0.54**	**0.53**	**0.51**	**0.53**	**0.53**

Takeaways. DP (with clipping) reduces MI-AUC by ~0.14 versus plain FL. Adding **secure aggregation + blockchain** slightly improves MI-AUC (0.54 → 0.53 on average), attributable to stricter norm proofs and reduced per-client observability—even though ε is unchanged (DP dominates formal privacy).

4.3 Overhead Analysis (Latency, Communication, On-Chain)

We report per-round medians over three runs. "PBFT" denotes a permissioned consortium network; "PoS (EVM)" denotes a public PoS chain with batched commits (one Merkle commitment + finalize + payout per round).

Table 4. Per-round latency breakdown (seconds; lower is better).

Dataset	Local Train	Comm (net)	On-Chain (PBFT)	Total (PBFT)	On-Chain (PoS)	Total (PoS)
CIFAR-10	1.80	0.28	0.35	**2.43**	6.50	**8.58**
FEMNIST	1.20	0.22	0.34	**1.76**	6.45	**7.87**
Sentiment	0.90	0.20	0.34	**1.44**	6.40	**7.50**
Tabular	0.40	0.18	0.33	**0.91**	6.38	**6.96**

Table 5. Blockchain activity and communication per round.

Setting	Tx / round	Gas / round (batched)	Finality (s)	Bytes / client ↑	Clients / round (m)
PBFT (permissioned)	3	n/a	0.30–0.50	1.2–1.6 MB	100
PoS (EVM)	3	270,000	6.0–7.0	1.2–1.6 MB	100

Energy proxy. Average client-side energy per round (from power logs / model) was **12.1 J (CIFAR-10)**, **8.3 J (FEMNIST)**, **6.7 J (Sentiment)**, and **3.1 J (Tabular)**; on-chain energy is not attributed to clients and is excluded.

Takeaways. In **PBFT**, on-chain orchestration adds ~0.33–0.35 s to each round (≈20–35% overhead depending on task). In **PoS**, confirmation latency dominates round time; batching and micro-rounds (Table 8) are necessary to keep throughput usable on public networks.

4.4 Robustness to Dropouts and Poisoning

Table 6. Robustness under adversaries (20% malicious clients).

Backdoor success rate measured on CIFAR-10; "Utility" is test accuracy (%). Robust aggregator = trimmed mean (0.2).

Method	Robust Aggregator	DP/SA	Backdoor Success ↓	Utility ↑
Plain FL	No	No	62.4%	85.9
Plain FL	**Yes**	No	14.7%	85.1
FL + Blockchain	Yes	No	13.9%	85.0
FL + DP	Yes	DP only	9.6%	84.2
Full (DP + SA + BC)	**Yes**	**DP + SA**	**5.8%**	**84.0**

Dropouts. With 30% client dropouts, secure aggregation with dropout recovery maintained successful rounds in **>98%** of attempts; without recovery, completion fell to **83%** (not shown).

Takeaways. Robust aggregation is essential under poisoning; **DP + SA** further depresses backdoor success by limiting the effective signal from outliers and preventing per-client inspection.

4.5 Cost and Scalability

Throughput is reported in completed rounds per hour for a fixed wall-clock budget and 100 participating clients unless otherwise noted. "Micro-batching (k=5)" finalizes 5 FL steps per block (PBFT) or per L2 batch (PoS).

Table 7. Throughput vs. number of total clients (N) and participation rate (m/N).

N (total)	m/N	PBFT Rounds/h ↑	PoS Rounds/h ↑
50	0.30	980	380
100	0.20	720	260
500	0.10	210	72

Table 8. Effect of micro-batching and L2 anchoring (CIFAR-10; PoS).

Setting	On-Chain Latency ↓	Tx / round ↓	Gas / round ↓	Rounds/h ↑	Acc (%)
Baseline (L1, no batching)	6.50 s	3	270k	260	84.0
L2 rollup anchor (no batching)	2.10 s	3	40k	520	84.0
L2 + micro-batching (k=5)	**0.65 s**	**1**	**12k**	**910**	83.9

Takeaways. On public networks, **L2 anchoring** and **micro-batching** are decisive, cutting on-chain latency ~10× and gas ~20× while preserving accuracy.

4.6 Ablations and Sensitivity

We ablate components on CIFAR-10 to quantify their marginal effects.

Table 9. Ablation study (CIFAR-10).

Configuration	DP ε (δ=1e-5) ↓	MI-AUC ↓	Acc (%) ↑	Rounds to 1% ↓	On-Chain (PBFT) s ↓
Plain FL	–	0.70	85.9	95	–
+ Blockchain only	–	0.69	85.6	100	0.35
+ DP only	6.3	0.56	84.2	115	–
+ DP + SA	6.3	0.55	84.1	119	–
Full (DP + SA + BC)	**6.3**	**0.54**	**84.0**	**125**	**0.35**

Table 10. DP sensitivity (CIFAR-10; PBFT).

Noise multiplier σtuned with fixed clipping C; larger σlowers ε and utility.

σ	ε ↓	Acc (%) ↑	MI-AUC ↓
0.8	8.1	84.9	0.58
1.0	7.0	84.5	0.56
1.2 (default)	**6.3**	**84.0**	**0.54**
1.5	5.1	83.2	0.52

Table 11. Participation sensitivity (CIFAR-10; PBFT).

m/N	Acc (%) ↑	ε ↓	Rounds to 1% ↓
0.10	83.6	5.7	138
0.20 (default)	**84.0**	**6.3**	**125**
0.30	84.2	7.1	116

Takeaways. Most of the accuracy cost comes from DP noise; SA is nearly neutral for utility but improves adversarial resilience. Participation increases convergence speed but raises ε via stronger composition.

DISCUSSION

We find that a decentralized blockchain addition in federated learning (FL) does not adhere to any specific pattern in terms of model quality: the accuracy loss between Full (DP + SA + Blockchain) and plain FL can be considered as a result of the different privacy measure (DP) rather than an on-chain orchestration. In permissioned systems, the additional latency of consensus and contract calls, which is approximately 0.35 s / round, is relatively small when compared to the latency of local training; in public Proof-of-Stake (PoS) systems, the most significant bottleneck is now the confirmation latency, unless it can be reduced by L2 anchoring and micro-batching. The presence of strong aggregation together with DP and secure aggregation (SA) means that the backdoor success is significantly minimized at a low utility cost evidence of the fact that cryptographic protection and incentive-compatible coordination can simultaneously achieve competitive performance.

These findings are consistent with the syntheses conducted in the past, demonstrating that the primary practical tensions in FL are non-IID information, communication boundaries, and the privacy-utility decision (Kairouz et al., 2021). The formal studies of SA help to understand that avoiding per-client inspection does not in itself commit leakage; quantifiable guarantees need DP over SA, which is in line with our MI-attack reductions at 6.3-eps (Elkordy et al., 2023). On adversaries, our benefit of strong aggregation and disguised updates is reflected with the current malicious-secure SA protocols maintaining

efficiency and allowing byzantine clients (Rathee et al., 2023).

Systems-wise, the ledger selection is an issue of concern. Authorized BFT-based consensus provides rapid finality and consistent overheads when used in enterprise cooperatives, and unauthorized PoS networks are transparent but expensive in terms of latency and charges. Modern surveys point to the ability of lightweight consensus and execution paths (e.g., committee-based BFT, L2 rollups) to reduce this gap, exactly what our experiments of micro-batching and L2 anchoring take advantage of (Chacko et al., 2025). In addition to coordination, blockchain brings auditability and programmable incentives and our payout mechanism and commit-reveal logging are inspired by the focus on provenance, reputation, and anti-free-riding design in FL-chain hybrids (Qu et al., 2022; Liu et al., 2024).

Limitations are the scope of datasets and one operating point of DP, wider domains and adaptive schedules of DP may represent more clearly privacy utility frontiers. We also evaluate on having good key management, honest-but-curious coordinators; it is worth studying in more detail where the aggregation is fully decentralized and committee-based and where the proofs of bounded norms are zero-knowledge. Lastly, end-to-end cost models that are stateful on training cadence, tokenomics and validator economics would assist practitioners to know when to use public chains (with L2) and when permissioned deployments are more desirable.

CONCLUSION

The paper shows that federated learning can be coupled with a blockchain coordination layer to provide incentive-compatible, verifiable and privacy-preserving training without any significant loss in model quality. On average, on a head-on comparison with plain FL, all-stack DP, secure aggregation (SA), and on-chain orchestration followed by a factor of 1.5-2.0 percentage points on average, the majority of which was due to DP and not the ledger itself. Formal privacy (ε 0.63 at) produced much less membership-inference success, whereas strong aggregation and DP and SA decreased backdoor success (62% plain FL) to 5.8%.

In systems perspective, the overhead of permissioned BFT consensus was approximately 0.35 s per round which is a small cost compared to local training. The cost and latency of Public Proof-of-Stake networks were more expensive and practical throughput was restored by Layer-2 anchoring and micro-batching, with minimal impact on the accuracy. The blockchain layer also offered auditable provenance, programmable incentive and efficient dispute resolution, which plain FL did not have.

Future research ought to extend to broader and more varied problems, consider adaptive DP budgeting along with zero-knowledge proofs of limited-norm updates, and test decentralized or committee-based aggregation. The end to end cost models, which was couple tokenomics, validator economics and training cadence, was assist the practitioners decide on whether to deploy in the public (with L2) or permissioned.

1) REFERENCES

2) Ahmed, A., Ahmad, A., Hassan, S. U., & Iqbal, F. (2023). FRIMFL: A fair and reliable incentive mechanism in federated learning. *Electronics, 12*(15), 3259. https://doi.org/10.3390/electronics12153259

3) Bai, L., Hu, H., Ye, Q., & Xu, J. (2025). Membership inference attacks and defenses in federated learning: A survey. ACM *Computing Surveys*. https://doi.org/10.1145/3704633

4) Chacko, N., Safari, M., & Evans, D. (2024). Lightweight consensus in blockchain: A systematic survey. ACM *Computing Surveys, 56*(12), Article 335. https://doi.org/10.1145/3768149

5) Chacko, N., Safari, M., & Evans, D. (2025). Lightweight consensus in blockchain: A systematic survey. *ACM Computing Surveys*. Advance online publication. https://doi.org/10.1145/3768149
6) DFL. (2023). DFL: High-performance blockchain-based federated learning. *ACM Transactions* (journal venue per DOI record). https://doi.org/10.1145/3600225
7) Elkordy, A. R., Zhang, J., Ezzeldin, Y. H., Psounis, K., & Avestimehr, S. (2023). How much privacy does federated learning with secure aggregation guarantee? *Proceedings on Privacy Enhancing Technologies, 2023*(1), 510–526. https://doi.org/10.56553/POPETS-2023-0030
8) Elkordy, A. R., Zhang, J., Ezzeldin, Y. H., Psounis, K., & Avestimehr, S. (2023). How much privacy does federated learning with secure aggregation guarantee? *Proceedings on Privacy Enhancing Technologies, 2023*(1), 510–526. https://doi.org/10.56553/POPETS-2023-0030
9) Guendouzi, A., Idoughi, D., & Bellot, P. (2023). A systematic review of federated learning: Challenges, methods, and future directions. *Journal of Network and Computer Applications, 223*, 103714. https://doi.org/10.1016/j.jnca.2023.103714
10) He, X., Xu, Y., & Zhang, S. (2024). Enhance membership inference attacks in federated learning. *Computers & Security, 139*, 103535. https://doi.org/10.1016/j.cose.2023.103535
11) Issa, W., Moustafa, N., Turnbull, B., Sohrabi, N., & Tari, Z. (2023). Blockchain-based federated learning for securing Internet of Things: A comprehensive survey. *ACM Computing Surveys, 55*(9), Article 191. https://doi.org/10.1145/3560816
12) Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning, 14*(1–2), 1–210. https://doi.org/10.1561/2200000083
13) Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning, 14*(1–2), 1–210. https://doi.org/10.1561/2200000083
14) Li, Y., Du, W., Han, L., Zhang, Z., & Liu, T. (2023). A communication-efficient, privacy-preserving federated learning algorithm based on two-stage gradient pruning and differentiated differential privacy. *Sensors, 23*(23), 9305. https://doi.org/10.3390/s23239305
15) Liu, K., Yan, Z., Liang, X., Kantola, R., & Hu, C. (2024). A survey on blockchain-enabled federated learning and its prospects with digital twin. *Digital Communications and Networks, 10*(3), 589–605. https://doi.org/10.1016/j.dcan.2022.08.001
16) Liu, Z., Yin, B., Umer, M., Ren, Y., & Li, X. (2024). A survey on blockchain-enabled federated learning and its prospects with digital twin. *Digital Communications and Networks, 10*(3), 589–605. https://doi.org/10.1016/j.dcan.2022.08.001
17) Nagy, B., Hegedűs, I., Sándor, N., Kis, A., & Jelasity, M. (2023). Privacy-preserving federated learning and its application to natural language processing. *Knowledge-Based Systems, 268*, 110475. https://doi.org/10.1016/j.knosys.2023.110475
18) Ning, W., Zhu, Y., Song, C., Li, H., Zhu, L., Xie, J., Chen, T., Xu, T., Xu, X., & Gao, J. (2024). Blockchain-based federated learning: A survey and new perspectives. *Applied Sciences, 14*(20), 9459. https://doi.org/10.3390/app14209459
19) Oyinloye, D. P., Chen, Z., Jararweh, Y., & Al-Bashayreh, M. (2021). Blockchain consensus: An overview of alternative protocols.

Symmetry, 13(8), 1363. https://doi.org/10.3390/sym13081363

20) Park, J., & Lim, H. (2022). Privacy-preserving federated learning using homomorphic encryption. *Applied Sciences, 12*(2), 734. https://doi.org/10.3390/app12020734
21) Peng, Z., Xu, J., Chu, X., Gao, S., Yao, Y., Gu, R., & Tang, Y. (2022). VFChain: Enabling verifiable and auditable federated learning via blockchain systems. *IEEE Transactions on Network Science and Engineering, 9*(1), 173–186. https://doi.org/10.1109/TNSE.2021.3050781
22) Peng, Z., Xu, J., Chu, X., Gao, S., Yao, Y., Gu, R., & Tang, Y. (2022). VFChain: Enabling verifiable and auditable federated learning via blockchain systems. *IEEE Transactions on Network Science and Engineering, 9*(1), 173–186. https://doi.org/10.1109/TNSE.2021.3050781
23) Qi, X., Zhang, H., Xia, X., Li, Y., Wang, M., & Li, L. (2023). Model aggregation techniques in federated learning: A comprehensive survey. *Future Generation Computer Systems, 149*, 529–550. https://doi.org/10.1016/j.future.2023.09.008
24) Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Li, B., & Zheng, G. (2022). Blockchain-enabled federated learning: A survey. *ACM Computing Surveys, 55*(11), Article 225. https://doi.org/10.1145/3524104
25) Qu, Y., Gao, L., Luan, T. H., Xiang, Y., Li, B., & Zheng, G. (2022). Blockchain-enabled federated learning: A survey. *ACM Computing Surveys, 55*(11), Article 225. https://doi.org/10.1145/3524104
26) Rathee, M., Khurana, D., Kapoor, A., Gupta, D., & Chandran, N. (2023). ELSA: Secure aggregation for federated learning with malicious actors. *2023 IEEE Symposium on Security and Privacy (SP)*, 1961–1979. https://doi.org/10.1109/SP46215.2023.10179468
27) Rathee, M., Khurana, D., Kapoor, A., Gupta, D., & Chandran, N. (2023). ELSA: Secure aggregation for federated learning with malicious actors. In *2023 IEEE Symposium on Security and Privacy (SP)* (pp. 1961–1979). https://doi.org/10.1109/SP46215.2023.10179468
28) Ren, X., Yang, S., & Guo, H. (2024). When federated learning meets differential privacy. *ACM Computing Surveys*. https://doi.org/10.1145/3650028
29) Tang, Y., Lu, H., Huang, C., & Yu, S. (2024). A survey on blockchain-based federated learning. *Computer Standards & Interfaces, 91*, 103010. https://doi.org/10.1016/j.csi.2024.103010 *(journal/DOI representative of listing)*
30) Tang, Z., Lu, H., & Wang, P. (2025). Differential privacy in federated learning: An evolutionary game analysis. *Applied Sciences, 15*(6), 2914. https://doi.org/10.3390/app15062914
31) Wu, L., Zhang, C., Wu, Q., & Chen, L. (2023). A survey on blockchain-based federated learning. *Future Internet, 15*(12), 400. https://doi.org/10.3390/fi15120400
32) Zhang, L., Li, L., Wang, H., Liu, R., & Zhang, T. (2023). Efficient membership inference attacks against federated learning via bias differences. *Proceedings of RAID 2023*. https://doi.org/10.1145/3607199.3607204
33) Zhu, J., Cao, J., Saxena, D., Jiang, S., & Ferradi, H. (2023). Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Computing Surveys, 55*, Article 3570953. https://doi.org/10.1145/3570953