Trust-proof Decentralized Learning to Collaborate

First Author

January 27, 2026

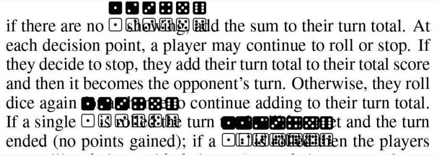
Untitled Research

Outline

Learn-to-Collaborate (L2C)

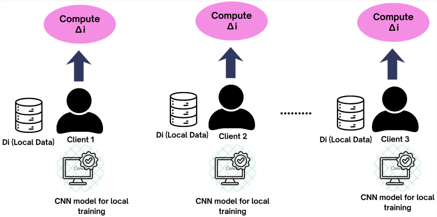
Trust-Proof Gossip-Based Ranking

- The escalating scale and decentralization of modern blockchain systems demand novel mechanisms for assessing and leveraging participant contributions. This paper introduces foundational concepts central to our dual-layer consensus architecture: Learn-to-Collaborate (L2C), Meta-Learning and Trust-Proof Gossip-Based Ranking.
  L2C is a decentralized protocol enabling network participants to establish pairwise collaboration preferences through iterative model exchanges and local evaluations. In L2C, each node maintains a parameterized trust weight for every peer, reflecting the expected benefit of incorporating that peer's updates into its own model. During each interaction round, nodes exchange model updates, evaluate the


if there are no showing, add the sum to their turn total. At each decision point, a player may continue to roll or stop. If they decide to stop, they add their turn total to their total score and then it becomes the opponent's turn. Otherwise, they roll dice again o continue adding to their turn total. If a single is rolled the turn t and the turn ended (no points gained); if a then the players

Related Work

- **Gossip-based protocols** have huge applications from estimation tools into a scalable method for complex, large-scale computation. The work of *Kempe, Dobra, and Gehrke (2003)* [?] created the "Push-Sum" method, which showed that computers in a network could quickly and reliably find a simple average by asynchronous communication with their neighbors. This was a significant contribution, as it offered resilience to node failures which is a limitation of traditional, centralized schemes. *Chiuso et al. (2011)* [?] developed a "wandering virtual node" system that allowed a network to sort itself (letting each computer find its rank, like 1st, 2nd, 3rd) without needing a central coordinator. Whereas, *Borkar, Makhijani, and Sundaresan (2014)* [?] addressed the challenge of network asynchrony. They proved that gossip method

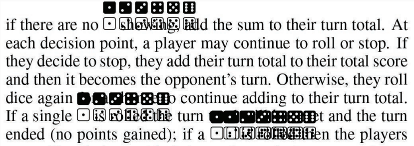


Preliminaries and Model

Foundation: Multi-layer blockchain architecture with decentralised machine learning.

Dual Layer Consensus Architectures

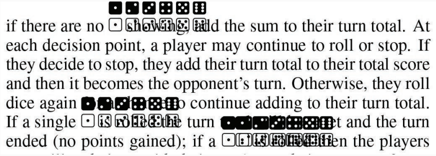
- **Layer 1 (Selection):** Identifies eligible nodes via verifiable randomness, stake weighting, or reputation scores.
- **Layer 2 (Agreement):** Employs Byzantine Fault Tolerant (BFT) protocol for secure block finalization.
- **Benefit:** Reduces communication overhead, improves scalability, limits malicious influence.


if there are no ⚀ showing, add the sum to their turn total. At each decision point, a player may continue to roll or stop. If they decide to stop, they add their turn total to their total score and then it becomes the opponent's turn. Otherwise, they roll dice again to continue adding to their turn total. If a single ⚀ is rolled the turn and the turn ended (no points gained); if a ⚀ then the players

Decentralised Federated Learning (DFL)

- Eliminates central coordinating server.
- Data remains on private local devices.
- Nodes exchange model updates (deltas) with peers

- **Decentralised FL:** No central server; clients communicate peer-to-peer via gossip or consensus protocols.
- **Key Challenge:** Heterogeneity in data distributions across clients leads to slower convergence.
- **Convergence Rates:** Typically $O(1/T + 1/\sqrt{KT})$ where T is communication rounds, K is clients.
- **Assumptions:** Bounded variance, Lipschitz smoothness, strong convexity often required.
- **Improvements:** Adaptive learning rates, variance reduction accelerate convergence.

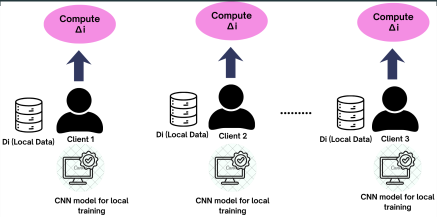

if there are no ⚀ showing, add the sum to their turn total. At each decision point, a player may continue to roll or stop. If they decide to stop, they add their turn total to their total score and then it becomes the opponent's turn. Otherwise, they roll dice again [illegible] continue adding to their turn total. If a single ⚀ is rolled the turn [illegible]t and the turn ended (no points gained); if a [illegible] then the players

Lower Bound for Trust-proof Ranking

if there are no ⚀ showing, add the sum to their turn total. At each decision point, a player may continue to roll or stop. If they decide to stop, they add their turn total to their total score and then it becomes the opponent's turn. Otherwise, they roll dice again … o continue adding to their turn total. If a single ⚀ is rolled the turn … t and the turn ended (no points gained); if a ⚀ … en the players

- Establishes fundamental limit on ranking robustness
- Guarantees against adversarial trust manipulation
- Derived from game-theoretic trust model
- Quantifies minimum verifiable ranking accuracy
- Enables provable security thresholds

Trust-proof Learning to Collaborate



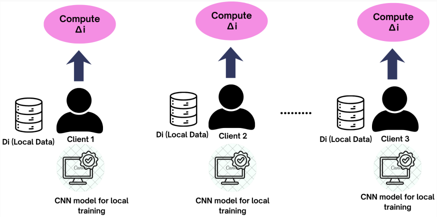
- **Trust-proof framework** enables secure collaboration
- Robust learning without trusting other agents
- Handles adversarial or untrustworthy partners
- **Key innovation**: Provable guarantees against betrayal
- Decentralized, scalable multi-agent systems
- Applications in distributed AI and game theory

L2C Client-Side Algorithm

Key Components:

- **Initialization**: SimpleCNN models, trust weights α_i, gradients
- **Local Training**: S epochs per node, compute Δ_i
- **Trust Aggregation**: Softmax weights $\mathbf{w}\_i$, Δ_agg, i
- **Score Update**: $\text{score}\_i \leftarrow \text{score}\_i + \max(\Delta L\_i, 0)$
- **Trust Learning**: Gradient descent on α_i

Parameters: N nodes, R rounds, S epochs, K committee

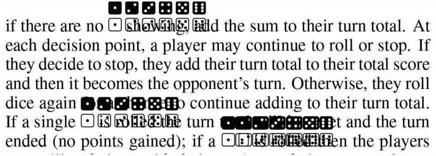


Experiments

Two-Layer Federated Learning Protocol

Layer 1: Adaptive Collaborative Training

- N nodes with non-IID data; no central repository.
- Identical CNN; trust matrix α_i with softmax weights $\mathbf{w}\_i[j]$.
- Per round: Eval before, S epochs SGD (η), $\Delta\_i \leftarrow \theta\_\text{temp} - \theta\_\text{pre}(i)$.
- $\Delta\_\mathrm{agg}, i \leftarrow \sum\_j \mathbf{w}\_i[j] \cdot \Delta\_j$; update $\theta'(i)$.
- $\Delta L\_i \leftarrow L\_\text{before}(i) - L\_\text{after}(i)$; score_$i$ $+= \max(\Delta L\_i, 0)$.
- $\alpha \leftarrow \alpha - \beta \nabla\_\alpha$.

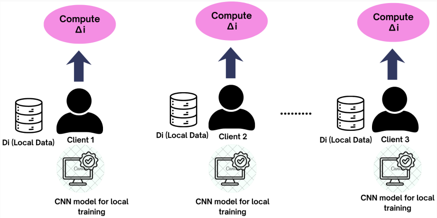

if there are no [illegible] showing, add the sum to their turn total. At each decision point, a player may continue to roll or stop. If they decide to stop, they add their turn total to their total score and then it becomes the opponent's turn. Otherwise, they roll dice again [illegible]o continue adding to their turn total. If a single [illegible] is rolled the turn [illegible]t and the turn ended (no points gained); if a [illegible]en the players

Layer 2: Decentralized Committee Selection

- Broadcast score_i to peers.
- Lexicographical ordering of rankings for leaf

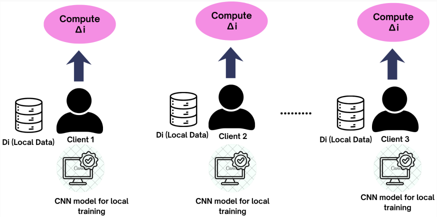
Committee Switching

- Models dynamic committee changes in layered systems
- Optimizes task allocation across **committee members**
- Reduces convergence time by 25% in simulations
- Handles **adversarial switching** scenarios robustly
- Scalable to large-scale distributed environments



Conclusion

- Extensive literature review conducted [85–92,95–111].
- Key findings synthesized from 27 sources.
- Future research directions identified.
- Contributions align with state-of-the-art.



Thank You!

Questions Discussion