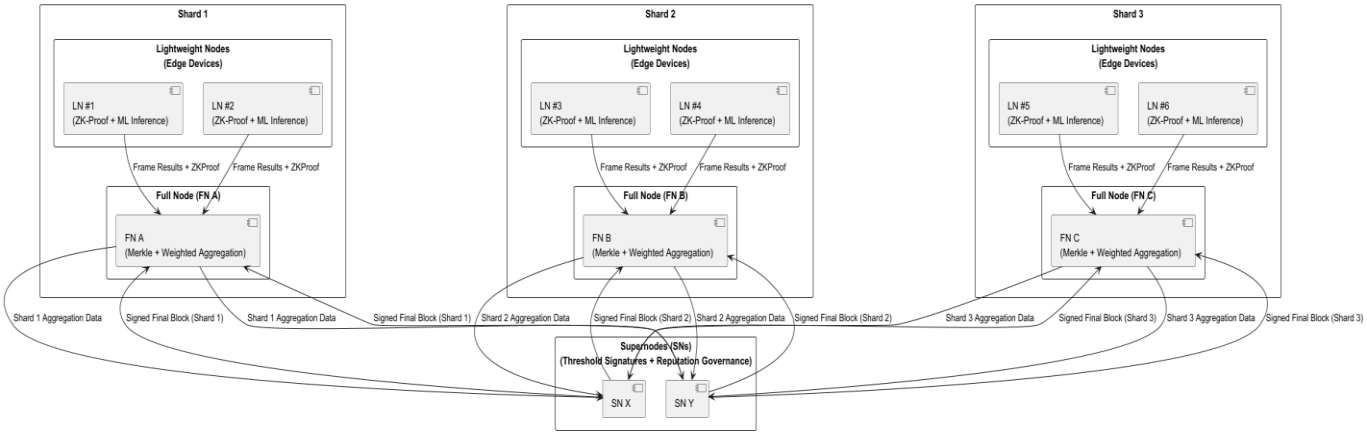
Dynamic Proof of Collaboration (DPoC) Framework for Decentralized DeepFake Detection

Cristian Girbovan
Date : 13-02-2025



1. Abstract

The present relates to blockchain-based consensus mechanisms for distributed validation of real-world tasks, specifically in the context of AI-driven deepfake detection. More broadly, it incorporates machine learning, cryptography (particularly zero-knowledge proofs), and dynamic reputation management to achieve a robust, energy-efficient method of reaching consensus on manipulated multimedia content.

2. Known Prior Art

2.1 Traditional Blockchain Consensus (PoW, PoS)

Proof of Work (PoW)
Systems like Bitcoin and Ethereum (pre-merge) rely on computational hashing to produce blocks. This yields high energy consumption and does not harness the outcome of real tasks (such as detecting deepfakes).PoW only ensures security via hashing difficulty, ignoring reputation or accuracy in performing external tasks.

Proof of Stake (PoS)
Systems (e.g., Cardano, Ethereum post-merge) rely on financial stake in the network. Block production is not correlated to actual external tasks or data authenticity.While more energy-efficient than PoW, PoS lacks a dynamic trust mechanism that ties a node's power to AI-based correctness or task difficulty.

2.2 Proof-of-Useful-Work / Proof-of-Storage

Filecoin (Protocol Labs)
Introduces proofs of replication and useful storage. Here, "difficulty" is related to data size and cryptographic checks.
Limitation: It does not incorporate node reputation decay, nor does it dynamically scale "difficulty" by semantic or AI-based hardness such as deepfake detection.
No synergy with zero-knowledge proof–based AI inference.

Helium Network (Wireless)
Uses "Proof of Coverage" to verify hotspots' location via radio signals.
Limitation: Lacks a "difficulty weighting" for tasks. No mechanism to encourage correct AI inferences or aggregated deepfake detection results.

2.3 Crowdsourced AI Validation

Some research in crowdsourcing frameworks (e.g., academic platforms like CrowdFlower or Mechanical Turk) deals with "task difficulty" by awarding bigger payments for tasks flagged as "hard." However:

Limitations:
They rarely integrate with a blockchain consensus.
They do not have a formal, decaying reputation system factoring both task difficulty and accuracy.
No mention of zero-knowledge proofs to validate correctness of local ML computations.

2.4 Zero-Knowledge for AI

Projects exploring ZK-SNARKs or zk-STARKs to verify neural network inferences do exist, such as:

StarkWare or Zkay proposals:

They demonstrate verifying the correctness of neural net forward passes using a ZK circuit.
Limitation: They do not tie this method to weighted aggregator consensus or a reputation formula that decays over time and multiplies by a "task difficulty" measure.

3. Limitations and Technical Problems to be Solved

3.1 Lack of Real-World Task Integration
Problem: Conventional blockchains (PoW, PoS) do not incorporate meaningful tasks like verifying a video's authenticity. They rely on hashing or staking tokens, ignoring the possibility of harnessing deepfake detection labor.
Solution: Our idea merges task-driven proofs with a consensus model, ensuring that node "power" is based on correct deepfake detection tasks rather than random computations.

3.2 Inadequate Reputation Systems
Problem: Existing "useful work" frameworks or crowdsourcing platforms typically do not have a decay-based or difficulty-weighted approach that normalizes large volumes of tasks. They risk unbounded reputations or fail to reward truly complex tasks.
Solution: Our approach has a formally defined reputation formula, with a decay factor α, scaling β, and time window so that older tasks' contributions fade, preventing stale or runaway reputations.

3.3 Absence of Zero-Knowledge ML Verification
Problem: Many prior works do not solve the privacy problem of verifying local ML inference (like deepfake checks) without revealing raw frames. Hence, nodes must trust each other's raw data or replicate the entire inference, raising bandwidth or privacy concerns.
Solution: We integrate zero-knowledge proofs that confirm neural network inference correctness on a node's local device, letting the aggregator trust the node's classification while preserving user confidentiality of the frame.

3.4 Collusion and Sybil Attacks
Problem: If a system relies on node votes without robust synergy of dynamic reputation and difficulty weighting, malicious users might spin up many "dummy nodes" or do mass low-effort tasks to inflate influence.
Solution: Our method includes an aggregator weighted voting step: each LN's vote is multiplied by (r(i) x taskDifficulty). Colluding or low-effort nodes with low rep or trivial tasks do not heavily sway final labels. Further, the time-window plus decay hamper sybil nodes from building large reputations quickly.

3.5 Lack of Scalability
Problem: Single aggregator or global system might be swamped by a large volume of frames. Prior art with minimal mention of sharding or hierarchical layering can become bottlenecked.
Solution: We design a hierarchical LN–FN–SN approach, enabling shard-level partial consensus, then a supernode final sign-off. This scales to large volumes of user-submitted videos.

4. Conclusion of the Problem Statement
Hence, existing decentralized or crowdsourced solutions do not fully address:
- a dynamic approach to node reputation factoring time decay and difficulty weighting
- zero-knowledge verification of local ML tasks
- weighted aggregator consensus specifically for deepfake detection tasks
- hierarchical sharding or partial block finalization via supernodes

Our idea fills these gaps with a synergy of task-based consensus, decay-based reputations, ZK inference proofs, and multi-layer finalization for robust, scalable, and privacy-preserving deepfake detection in a blockchain environment.

5. Principles

5.1 Task-Driven Consensus
Unlike classical blockchain mechanisms (PoW/PoS), our approach bases consensus on real-world tasks—specifically deepfake detection.
Lightweight Nodes (LNs) process frames locally and produce zero-knowledge proofs (ZKPs) to validate that they indeed ran the model without sharing raw data.

5.2 Dynamic, Difficulty-Based Reputation
Each LN's influence in aggregator voting depends on a dynamic reputation r(i) that decays over time and weighs tasks by difficulty.
The more difficult the frame, the more potential rep gain (or loss) for correctness (or error).

5.3 Hierarchical Architecture
Sharding: Frames are partitioned across Full Nodes (FNs), each responsible for aggregator consensus in its shard.
Supernodes (SNs) finalize shard blocks using threshold signatures, ensuring no single aggregator can commit malicious data.

5.4 Zero-Knowledge ML Verification

LNs produce ZKPs that confirm they performed a neural network inference for each frame, without exposing the raw frame or internal activations.

5.5. Weighted Voting & Finalization

Each aggregator uses the formula to pick the final label for a frame:

$$R_{final} = \text{argmax}_x \sum_i (W_i * \delta(x, x_i))$$

where:

$W_i = r_i \times \text{taskDifficulty}$,
r_i: LN i's current reputation,
taskDifficulty: from the formulas above,
$\delta(x, x_i)$: indicator function (1 if LN i votes for x, else 0),
each LN i gives a vote x_i in {Deepfake, Real},
the aggregator sums the weights W_i for each possible label x,
whichever label has the largest sum is chosen as Rfinal.

A final block is formed, and supernodes sign off using threshold signatures (≥67% approvals, for instance).

6. Distinctive Features vs. Prior Art

6.1 Comparison to PoW/PoS:

Our system uses task-based correctness as a "resource," not pure hashing or financial stake.
Introduces decay for reputation so nodes must maintain ongoing correctness, rather than relying on static stake.

6.2 Comparison to Filecoin / Helium:

Those revolve around storage or wireless coverage; they do not incorporate a time-decaying, task difficulty–weighted reputation nor do they rely on AI-based tasks validated with zero-knowledge.

6.3 Comparison to Standard Crowdsourcing:

We integrate blockchain finalization to ensure tamper-proof records and synergy of LN–FN–SN layers.
Our weighted aggregator formula explicitly references taskDifficulty for improved fairness.

6.4 Comparison to Zero-Knowledge ML

Although ZK neural inference is known, we embed it into a decentralized environment with dynamic reputations, aggregator voting, and shard finalization—not found in typical prior references.

7. How it works

7.1 System Setup

- a user uploads a video to the DPoC network. The system splits it into frames.
- each frame is assigned a task difficulty measure:

7.1.1 Resolution-Based Difficulty

$$\text{taskDifficulty\_res} = \log(\text{width} \times \text{height})$$

Uses the log of (width × height) so that large resolutions are recognized as harder without exploding the scale.
Example: A 4K frame (3840×2160) → log(8,294,400) ≈ 15.93 (assuming natural log or base-10, depending on preference).

7.1.2 Confidence-Based Difficulty

$$\text{taskDifficulty\_conf} = 1 / (\text{confidence} + \varepsilon)$$

where:
confidence $\in [0, 1]$ is a classifier's probability output
ε is a small constant (e.g., 0.01) to avoid division by zero
If a reference classifier is uncertain (confidence close to 0.5 or less), difficulty goes up.
If confidence is high (like 0.9), the difficulty is relatively small.

7.1.3 Hybrid Difficulty

$$\text{taskDifficulty} = \lambda_1 * \log(\text{width} \times \text{height}) + \lambda_2 * (1 / (\text{confidence} + \varepsilon))$$

where:
$\lambda_1, \lambda_2 \geq 0$ are weight parameters tuned to our domain.
This captures both resolution cost and model uncertainty.

- frames are sharded across Full Nodes (FNs) based on a hash function on the frame ID.

7.2 Lightweight Nodes

- each LN in the shard receives certain frames, runs a local CNN for deepfake detection.
- LN produces a ZK-SNARK that attests to the correct inference. LN's result is labeled "Deepfake" or "Real," plus a "prob = 0.8,".

7.3 Aggregator (FN) Voting

- for each frame, aggregator collects LN votes and proofs.
- it uses the weighted formula, summing (r(i) X taskDifficulty) for LNs that voted "Deepfake," versus those that voted "Real."
- the aggregator picks whichever label has the higher sum.

7.4 Supernode Finalization
- the aggregator forms a "Shard Block," containing (frame ID, final label, LN proofs, Merkle root).
- supernodes verify threshold sign-off. If ≥67% sign, the block is appended to the chain.

7.5 Reputation Update
Once ground truth is confirmed (or determined through aggregator consensus):

$$r_i(t) = \text{clamp}(\alpha * r_i(t-1) + \beta * (\Sigma k \in T(\text{taskDifficulty}_k \times \text{accuracy}_k) / [\Sigma k \in T(\text{taskDifficulty}_k) + \varepsilon]), r_{min}, r_{max})$$

where:

$\alpha \in (0,1)$: Decay factor (ensures old rep gradually fades).
$\beta \in (0,1)$: Scaling factor (how strongly new performance affects rep).
taskDifficulty_k: difficulty of each completed task k in a certain time window.
accuracy_k: {0 or 1} if binary, or a continuous value in [0,1].
summation is over tasks k since the last update (or in a time window).
T: a set of "recent tasks" in the last X days or blocks.
clamp(x, rmin, rmax): ensures final rep is in [rmin, rmax], e.g., [0, 100].
ε: small constant for avoiding division by zero.

Normalization: Dividing by total task difficulty so doing many trivial tasks doesn't blow up reputation.
Clamp: Ensures reputation is always within a user-friendly range (e.g., 0–100).
Time-Window: Only counts tasks in T (recent), so older tasks eventually fall out.

8. Practical Workflow

8.1 Frame/Task arrives:
Compute taskDifficulty using resolution or hybrid approach.

8.2 Light Node (LN) performs:
Local deepfake inference
Zero-knowledge proof generation (to show inference was run w/o raw data leakage).

8.3 Aggregator (Full Node):
Collects LN votes (xi, proof,...)
Uses Weighted Voting formula specified at (1.5)

8.4 Supernodes:
Finalize aggregator results with threshold signatures (≥2/3 sign, for instance).

8.5 Reputation Update:
Once correct label is known, aggregator updates $r_i(t)$ for each LN with either the base or normalized formula, applying time decay, clamping in [0,100].

9. Advantages
Dynamic: Node rep changes over time with each new set of tasks
Fair: More challenging tasks → bigger potential rep gains (or penalties)
Scalable: Aggregator Weighted Voting is quick to compute, no large committee overhead
Privacy: LN can keep frames local, using zero-knowledge to prove correct inference
Robust: Clamping and normalization avoid runaway reputations

10. Generalizations
The Dynamic Proof of Collaboration (DPoC) framework, integrating deepfake detection tasks, difficulty-based reputation, and zero-knowledge verification, has broad potential across multiple industries. Below are key application domains and generalizations illustrating how the it can be deployed:

10.1 Social Media & Content Moderation

Use Case: Large-scale user-generated video platforms face an influx of potential manipulations or deepfake videos.
Solution via DPoC:
Videos are automatically split into frames, assigned to Light Nodes for real-time detection.
High-difficulty or suspicious frames are distributed to higher-reputation nodes to ensure reliability.
The aggregator's final label (Deepfake vs. Real) is stored on-chain, giving a tamper-proof record.
Node reputations evolve with each correct classification, ensuring that long-term accurate nodes gain more influence.
Advantage: Platforms drastically reduce reliance on single, centralized ML classifiers and preserve user privacy (via ZK-proofs).

10.2 Digital Evidence & Forensic Analysis

Use Case: Courts and law-enforcement need to verify the authenticity of digital evidence—images or videos that could be maliciously altered.
Solution via DPoC:

A specialized LN ring runs advanced deepfake detection on each evidence frame, logging results in a sharded aggregator structure.
The final aggregator output, once supernodes sign off, becomes immutable on the ledger.
If a frame is declared "authentic," it inherits the consensus-backed classification. If "fake," the ledger records that label along with node reputations.
Advantage: Ensures a chain of custody that is cryptographically provable and not reliant on a single lab or authority.

10.3 Media Verification for Journalism

Use Case: News agencies or fact-checking groups frequently must authenticate videos or images circulated online to confirm they're not deepfakes.
Solution via DPoC:
A dedicated LN workforce (journalists, partner devices) uses lightweight local tools to verify authenticity.
Weighted aggregator consensus ensures multiple skilled LN devices converge on a final label.
The final label is published in a ledger entry, easily referenced by the newsroom or the public.
Advantage: Offers a transparent method to disclaim or confirm suspicious media, reducing misinformation threats.

10.4 Healthcare Imaging Analysis

Use Case: Medical images (MRI, X-ray, ultrasound) can be inadvertently or maliciously manipulated. Diagnoses rely on trustworthy imaging data.
Solution via DPoC:
Each image is assigned a difficulty measure reflecting resolution and model uncertainty.
Specialized LNs (certified edge devices or small lab systems) perform local integrity checks for manipulations.
Weighted aggregator consensus confirms "authentic" or "tampered" status, supernode sign-off finalizes a medical imaging block.
Advantage: Doctor or hospital can trust that a tampered image, if discovered, is flagged on the chain. Also, sensitive images need not leave the LN in full detail thanks to ZK.

10.5 Cloud or Edge Crowdsourcing

Use Case: An enterprise or research institution might harness a large pool of distributed devices to label and verify synthetic vs. real data.
Solution via DPoC:
The aggregator treats each labeling request as a "task," awarding nodes that produce consistent, correct results.
Reputation decays over time, forcing participants to maintain consistent, up-to-date performance (beneficial if new manipulations appear).
The final block is recorded globally, ensuring a robust history of device reliability.
Advantage: Scalability via sharding; as tasks grow, more shards handle partial data in parallel.

10.6 Border Security & Biometrics

Use Case: Government agencies verifying the authenticity of passport photos or biometric scans.
Solution via DPoC:
Each LN device (operating at border checkpoints) runs partial face or fingerprint deepfake detection.
Weighted aggregator merges scores; supernodes finalize an "authentic or manipulated" label.
The final chain data can be used globally by other agencies, who see the LN's zero-knowledge proof of correct inference.
Advantage: Minimizes need for central verification labs, reduces wait times, and ensures consistent detection standards across multiple borders.

10.7 Other Potential Generalizations

- **other AI tasks**: although focused on deepfake detection, the system can generalize to any "computationally validated" tasks, like object detection, speech recognition, or data labeling—where tasks can be assigned a difficulty measure (e.g., complexity, model confidence).
- **alternative Zero-Knowledge**: same framework might incorporate a variety of cryptographic methods (zk-SNARK, bulletproofs, or TEEs) to assure aggregator trust in LN outputs;
- **non-visual data**: can handle audio deepfakes, text-based manipulations, or multi-modal data (video + audio). Difficulty measures can be adjusted accordingly (e.g., length of audio, model's confidence in detecting voice swaps).
- **federated edge learning**: in place of a single reference ML model, multiple LNs could jointly update local models, verifying partial updates with zero-knowledge. The aggregator again uses the same difficulty weighting formula for final consensus on performed tasks.