Masa: The Decentralized Protocol for Fair AI

Brendan Playford

May 2023

Abstract

Masa is a peer-to-peer protocol designed to create a global, decentralized, and incentivized network for Fair AI. It facilitates the fair, open, and permissionless contribution of AI training data and compute resources. In this ecosystem, actors collaborate and are rewarded based on the value of their contributions to the commons [1, 2]. This collaborative environment promotes innovation, transparency, and the equitable distribution of rewards, ensuring that each participant is compensated according to the value they add to the shared pool of resources [3, 4]. The Masa network comprises three key actors: Validators who maintain network integrity, Worker Nodes who contribute data and provide compute resources by processing AI requests using Open Source Large Language Models (OSLLMs), and Oracle Nodes who consume AI services. Participation is incentivized through a dual-token economic model using MASA and TAO, which actors stake to access the network. In this paper, we present Masa's formal approach to democratizing access to fair AI, with each actor incentivized based on their contributions to the commons. We detail the network architecture, staking mechanism, and reward structure that enable a global, decentralized, and incentivized self-improving AI network. Masa unlocks trillions of gigabytes of data and disrupts the current centralized paradigm, giving rise to a new wave of innovative and impactful AI applications worth trillions of dollars in economic value.

1 Multi-Agent Modeling

Before delving into the mathematical modeling of the Masa network, it is essential to define the terms actors' and agents' in the context of this whitepaper. In the Masa network, we define actors' as the primary participants - Validators, Worker Nodes, and Oracle Nodes - who actively contribute to the network's functionality and benefit from its economic incentives. Agents' refer to the mathematical abstraction of these actors when modeling their strategic behaviors and interactions using game theory and multi-agent systems. We model the Masa network as a multi-agent system composed of Validators (V), Worker Nodes (W), and Oracle Nodes (O). These agents interact through a shared global state defined by the Masa ledger (L). Each agent aims to maximize their own utility (U_i) by providing services to the network in exchange for token rewards. Let $s_{i,t}$ denote the stake of agent i at time t, and $r_{i,t}$ denote the token rewards earned by agent i at time t. The utility of agent i is then defined as:

$$U_i = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,t}\right] \tag{1}$$

where $\gamma \in (0,1)$ is a discount factor that models the time-value of rewards, and the expectation is taken over the stochastic processes governing the network dynamics.

1.1 Stochastic Network Dynamics

The Masa network evolves according to a continuous-time Markov process $\{X_t\}_{t\geq 0}$ with state space $\mathcal{S} = \mathbb{R}_+^{|V|+|W|+|O|}$. The state vector $X_t = (s_{1,t}, \ldots, s_{|V|+|W|+|O|,t})$ represents the stakes of all agents at time t.

The transition kernel of the Markov process is governed by the staking, reward, and slashing mechanisms described in the following sections. Let $Q : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ denote the transition rate matrix, where $Q(x,y)$ represents the rate of transitioning from state x to state y. The evolution of the network can be described by the following system of stochastic differential equations:

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

Here, $b : \mathcal{S} \to \mathbb{R}^{|V|+|W|+|O|}$ is the drift vector representing the expected change in stakes, $\sigma : \mathcal{S} \to \mathbb{R}^{(|V|+|W|+|O|)\times k}$ is the diffusion matrix representing the volatility of stakes, and W_t is a k-dimensional standard Brownian motion.

2 Staking Mechanism

Validator Node Staking: To be a Validator and participate in consensus and service provision, agents must stake Masa tokens as collateral. The amount of tokens staked determines an agent's eligibility and influence within the network. Let S denote the total token supply. We define the staking requirement for a validator v_i as:

$$s_{v,i} \geq \frac{S}{|V|} \times \sigma_v(U_V - U_{v,i})$$

Here, $U_V = \frac{1}{|V|}\sum_{i\in V} U_{v,i}$ is the average validator utility, and $\sigma_v : \mathbb{R} \to \mathbb{R}_+$ is a monotonic function that maps validator performance to a staking coefficient. In the simplest case, σ_v could be defined as:

$$\sigma_v(x) = \exp\left(\frac{x}{\lambda}\right)$$

where $\lambda > 0$ is a network parameter that controls the performance sensitivity of the staking requirement. Intuitively, validators with above-average performance can stake fewer tokens to earn the same reward, while below-average validators must stake more. This creates an incentive to provide high-quality service.

The Masa network limits the number of validator slots to 64. Each epoch, the lowest-performing validator is pruned from the network. The performance of a validator v_i is evaluated based on their contributions to the consensus activities, data verification, and LLM output evaluation. We define the performance score $P_{v,i}$ of validator v_i as a weighted sum of these metrics:

$$P_{v,i} = \alpha C_{v,i} + \beta D_{v,i} + \gamma L_{v,i}$$

where:

- $C_{v,i}$ represents the contributions to consensus activities.
- $D_{v,i}$ represents the effectiveness in data verification.
- $L_{v,i}$ represents the quality of LLM output evaluations.
- α, β, γ are weighting factors that control the relative importance of each metric.

Each metric can be further defined as follows:

- $C_{v,i}$ is the number of successfully validated blocks or consensus participations.
- $D_{v,i}$ is a score based on the accuracy and relevance of the data verified by the validator.
- $L_{v,i}$ is a score based on the relevance, coherence, factual accuracy, and response time of LLM outputs evaluated by the validator.

The lowest-performing validator is the one with the minimum performance score $P_{v,i}$ for that epoch:

$$v_{\min} = \arg\min_{v_i \in V} P_{v,i}$$

This validator is pruned from the network, ensuring that only high-performing validators remain and that there is a continuous improvement in service quality.

Oracle Node Staking: For Oracle Nodes, the staking requirement is tied to the amount of network resources consumed and the value derived from the network. Let $C_{o,i}$ denote the consumption of network resources by Oracle Node o_i. The staking requirement for an Oracle Node is defined as:

$$s_{o,i} \geq \frac{S}{|O|} \times \sigma_o(U_O - U_{o,i})$$

Here, $U_O = \frac{1}{|O|}\sum_{i\in O} U_{o,i}$ is the average utility of Oracle Nodes, and $\sigma_o : \mathbb{R} \to \mathbb{R}_+$ is a monotonic function mapping Oracle Node performance to a staking coefficient. In the simplest case, σ_o could be defined as:

$$\sigma_o(x) = \exp\left(\frac{x}{\lambda_o}\right)$$

where $\lambda_o > 0$ is a network parameter controlling the performance sensitivity of the staking requirement. This ensures that Oracle Nodes consuming more resources or deriving higher value from the network stake more tokens, promoting responsible and efficient resource use.

In addition to staking, Oracle Nodes are required to pay fees for the network resources they consume. The fee structure can be defined as follows:

$$F_{o,i} = f(C_{o,i})$$

where $F_{o,i}$ is the fee paid by Oracle Node o_i, and $f : \mathbb{R} \to \mathbb{R}_+$ is a function mapping resource consumption to fees. This fee mechanism ensures that Oracle Nodes contribute financially to the network's sustainability while being incentivized to minimize resource usage.

Worker Node Staking: For Worker Nodes, the staking requirement is based on the node's computational capacity and the quality of the data they contribute. Let $P_{w,i}$ denote the computational power of Worker Node w_i, and $Q_{w,i}$ represent the quality of the data provided by w_i. The staking requirement for a Worker Node is defined as:

$$s_{w,i} \geq \frac{S}{|W|} \times \sigma_w(U_W - U_{w,i})$$

Here, $U_W = \frac{1}{|W|}\sum_{i\in W} U_{w,i}$ is the average utility of Worker Nodes, and $\sigma_w : \mathbb{R} \to \mathbb{R}_+$ is a monotonic function mapping Worker Node performance to a staking coefficient. The performance of a Worker Node is determined by a combination of its computational power and data quality:

$$U_{w,i} = \alpha P_{w,i} + \beta Q_{w,i}$$

where $\alpha, \beta > 0$ are network parameters that control the relative importance of computational power and data quality. The staking coefficient σ_w can be defined similarly to σ_v:

$$\sigma_w(x) = \exp\left(\frac{x}{\lambda_w}\right)$$

where $\lambda_w > 0$ controls the performance sensitivity of the staking requirement for Worker Nodes. This ensures that Worker Nodes providing higher computational power and better quality data earn higher rewards, while those with lower performance must stake more, thereby incentivizing high-quality service.

Worker Nodes also receive rewards based on their contributions to the network, and their staking requirements can be adjusted dynamically based on their ongoing performance and resource provision. This dynamic adjustment helps maintain a balance between network stability and resource availability.

The mapping between performance and staking requirements for Validators, Worker Nodes, and Oracle Nodes is a key network design choice that can be tuned to optimize different objectives, such as maximizing quality, efficiency, and decentralization.

2.1 Staking Dynamics

The staking dynamics of the network can be modeled as a mean-field game (MFG), where each agent optimizes their staking strategy in response to the aggregate behavior of other agents. Let $\pi_i : \mathcal{S} \to \mathbb{R}_+$ denote the staking policy of agent i, mapping the current network state to a staking amount.

The optimal staking policy π_i^* for agent i is the solution to the following stochastic control problem:

$$\pi_i^* = \arg\max_{\pi_i} U_i(\pi_i, \pi_{-i})$$

subject to the staking requirements and network dynamics. Here, π_{-i} represents the staking policies of all agents except i, and $U_i(\pi_i, \pi_{-i})$ is the expected utility of agent i under the joint policy profile (π_i, π_{-i}).

The staking requirements for each type of node are defined as follows:

- **Validators**:
  $$s_{v,i} \geq \frac{S}{|V|} \times \sigma_v(U_V - U_{v,i})$$
  where U_V is the average validator utility and σ_v is a function mapping validator performance to a staking coefficient.

- **Oracle Nodes**:
  $$s_{o,i} \geq \frac{S}{|O|} \times \sigma_o(U_O - U_{o,i})$$
  where U_O is the average utility of Oracle Nodes and σ_o is a function mapping Oracle Node performance to a staking coefficient.

- **Worker Nodes**:
  $$s_{w,i} \geq \frac{S}{|W|} \times \sigma_w(U_W - U_{w,i})$$
  where U_W is the average utility of Worker Nodes and σ_w is a function mapping Worker Node performance to a staking coefficient. The performance of a Worker Node is determined by its computational power ($P_{w,i}$) and data quality ($Q_{w,i}$):
  $$U_{w,i} = \alpha P_{w,i} + \beta Q_{w,i}$$

The existence and uniqueness of a Nash equilibrium for the staking MFG can be established under certain regularity conditions on the utility functions and network dynamics. The equilibrium staking policies $\{\pi_i^*\}_{i \in V \cup W \cup O}$ induce a flow of tokens between agents that balances the incentives for service provision with the costs of staking.

The performance-sensitive staking requirements ensure that agents with higher performance metrics can stake fewer tokens, while those with lower performance must stake more. This dynamic adjustment promotes high-quality service provision across the network.

To further analyze the equilibrium, consider the agent's utility function as dependent on their staking amount and the network state:

$$U_i(\pi_i, \pi_{-i}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_{i,t}(\pi_i, \pi_{-i}) - c_{i,t}(\pi_i, \pi_{-i})\right)\right]$$

where $r_{i,t}$ represents the rewards earned by agent i and $c_{i,t}$ represents the costs associated with staking and participation.

The optimal staking policy π_i^* is then derived by solving the following Hamilton-Jacobi-Bellman (HJB) equation:

$$0 = \max_{\pi_i} \{r_{i,t}(\pi_i, \pi_{-i}) - c_{i,t}(\pi_i, \pi_{-i}) + \gamma \mathbb{E}[V(X_{t+1}) - V(X_t)]\}$$

where $V(X_t)$ is the value function representing the expected utility given the current network state X_t.

By aligning individual staking strategies with the network's aggregate behavior, the MFG framework ensures that the overall system remains robust, decentralized, and fair, promoting sustainable and high-quality AI service provision within the Masa network.

3 Worker Nodes: Data Scraping and LLMs

Worker nodes in the Masa network are responsible for processing AI requests using Large Language Models (LLMs).

3.1 Data Scraping

The data scraping process can be modeled as a cooperative game among worker nodes. Each worker node $w_i \in W$ contributes a dataset $D_{w,i}$ to the global training data pool $D = \bigcup_{i \in W} D_{w,i}$. The quality of the scraped data directly impacts the performance of the LLMs and, consequently, the rewards earned by the worker nodes. We can define a characteristic function $v : 2^W \to \mathbb{R}$ that assigns a value to each subset of worker nodes based on the quality and diversity of their collected data:

$$v(C) = Q\left(\bigcup_{i \in C} D_{w,i}\right) \quad \forall C \subseteq W$$

where $Q : 2^D \to \mathbb{R}$ is a quality measure that assesses the value of a dataset for LLM training. The Shapley value $\phi_i(v)$ can be used to determine a fair allocation of rewards among worker nodes based on their individual contributions:

$$\phi_i(v) = \sum_{C \subseteq W \setminus \{i\}} \frac{|C|!(|W| - |C| - 1)!}{|W|!}(v(C \cup \{i\}) - v(C))$$

This reward scheme encourages worker nodes to contribute high-quality, diverse data to improve the overall performance of LLM responses.

3.2 LLM Inference

The LLM inference process can be modeled as a repeated game between worker nodes and oracle nodes. At each round t, an oracle node submits a request x_t, and a worker node w_i generates a response y_t using its LLM. The quality of the response determines the reward $R_{\text{req}}(x_t, y_t)$ earned by the worker node and the satisfaction of the oracle node. We can define the stage game as follows:

- Players: Worker node w_i and oracle node o_j.
- Strategies: Worker node's strategy space is $\Sigma_w = \{y : x \to Y\}$, where Y is the set of possible responses. Oracle node's strategy space is $\Sigma_o = \{0, 1\}$, where 0 represents rejection and 1 represents acceptance of the response.
- Payoffs: If the oracle node accepts the response, the worker node receives a reward $R_{\text{req}}(x_t, y_t)$, and the oracle node receives a utility $U_o(x_t, y_t)$. If the oracle node rejects the response, both players receive a payoff of 0.

The repeated interaction between worker nodes and oracle nodes creates an incentive for worker nodes to continuously improve their LLMs and provide high-quality responses. The long-term payoff of a worker node w_i can be expressed as:

$$U_{w,i} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{w,i,t}\right]$$

where γ is the discount factor, and $R_{w,i,t}$ is the reward earned by worker node w_i at round t.

4 Reward Structure

Worker nodes earn rewards for contributing data to the protocol and processing LLM requests. The rewards are designed to align individual incentives with the overall performance and security of the network.

4.1 Request Processing Rewards

When an oracle node submits a request to the Masa network, worker nodes compete to process the request using their LLMs. The selected worker node w_i earns a reward $R_{\text{req}}(x, y)$ that depends on the quality of the generated response y to the prompt x:

$$R_{\text{req}}(x, y) = \begin{cases} R_{\text{base}} \cdot Q(x, y) & \text{if } Q(x, y) \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

Here, R_{base} is a base reward amount, $Q(x, y)$ is a quality score assigned to the response based on factors such as relevance, coherence, factual accuracy, and response time, and τ is a minimum quality threshold.

4.1.1 Quality Score Definition

The quality score $Q(x, y)$ for a response y to a prompt x is defined as:

$$Q(x, y) = \alpha R(x, y) + \beta C(x, y) + \gamma F(x, y) + \delta T(y)$$

where:

- $R(x, y)$ is the relevance score of the response y to the prompt x,
- $C(x, y)$ is the coherence score of the response y,
- $F(x, y)$ is the factual accuracy score of the response y,
- $T(y)$ is the response time score of the response y,
- $\alpha, \beta, \gamma, \delta$ are the weights assigned to each factor.

4.2 Slashing

To discourage malicious behavior and ensure the integrity of the network, Masa imposes slashing penalties on worker nodes that submit low-quality or fraudulent responses. If a worker node's response fails to meet the minimum quality threshold τ or is flagged as fraudulent by validators, a portion of its staked tokens is slashed:

$$s'_{w,i} = \max\{s_{w,i} - \kappa R_{\text{req}}, 0\}$$

where $s'_{w,i}$ is the updated stake of worker node w_i after slashing, R_{req} is the potential reward for the request, and $\kappa \in (0, 1)$ is a slashing coefficient.

Slashing acts as a disincentive for worker nodes to submit low-quality or malicious responses, as they risk losing a significant portion of their staked tokens. The slashing coefficient can be adjusted through governance to balance the trade-off between security and leniency.

5 Governance

Masa employs a liquid democracy governance model, where staked tokens can be delegated to validators or worker nodes to vote on proposals. This model combines the efficiency of representative democracy with the direct influence of a participatory system, ensuring that all participants can have their say in the network's evolution.

The voting power of an agent is the sum of its own stake and the stakes delegated to it by other agents:

$$\mathrm{VP}_i = s_i + \sum_{j \in V \cup W \cup O} d_{j \to i}$$

where $d_{j \to i}$ denotes the tokens delegated from agent j to agent i.

Proposals are submitted to the network in the form of executable code snippets or parameter updates. Each proposal p is associated with a minimum quorum q_p and an approval threshold τ_p. For a proposal to pass, it must receive a minimum number of votes equal to the quorum, and the weighted percentage of approving votes must exceed the threshold:

$$\frac{\sum_{i \in V_p^+} \mathrm{VP}_i}{\sum_{i \in V_p} \mathrm{VP}_i} \geq \tau_p$$

where V_p is the set of agents who voted on proposal p, and V_p^+ is the subset who voted in favor.

The liquid democracy model enables agents to delegate their voting power to trusted experts while still retaining the flexibility to vote directly on proposals they feel strongly about. This strikes a balance between efficiency and individual autonomy in the governance process.

5.1 Governance Mechanisms

To ensure robust and fair decision-making, the following governance mechanisms are implemented:

Proposal Submission: Proposals can be submitted by any agent with a minimum stake, ensuring that all participants can contribute to the network's evolution. Proposals must include detailed descriptions and rationale to facilitate informed voting.

Voting Delegation: Agents can delegate their voting power to other agents, typically those with expertise or a proven track record in specific areas. Delegation can be retracted at any time, allowing agents to regain direct control over their votes.

Quorum and Approval Thresholds: The quorum and approval thresholds are designed to ensure that only well-supported proposals are enacted. These thresholds can be dynamically adjusted through governance to reflect changes in network size and participation levels.

Transparency and Accountability: All governance actions, including proposal submissions, voting, and delegation, are recorded on the blockchain. This ensures transparency and allows the community to hold delegates and decision-makers accountable.

Periodic Reviews: Governance parameters, such as quorum and approval thresholds, are subject to periodic reviews and can be adjusted through new proposals. This flexibility ensures that the governance model remains effective as the network evolves.

5.2 Governance Incentives

To encourage active participation in governance, the following incentives are provided:

Staking Rewards: Agents who participate in governance by voting or submitting proposals are eligible for additional staking rewards. These rewards are proportional to their participation and the impact of their contributions.

Reputation System: A reputation system tracks the performance of agents in governance activities. Agents with high reputation scores can gain more influence and trust within the network, further incentivizing responsible and active participation.

Penalties for Inactivity: Agents who fail to participate in governance activities for extended periods may face reduced rewards or other penalties. This ensures that the governance process remains dynamic and engages the entire community.

By incorporating these governance mechanisms and incentives, the Masa network ensures a fair, transparent, and effective decision-making process that aligns with the principles of decentralization and community-driven development.

6 Security Analysis

The security of the Masa network relies on the incentive compatibility of the staking, slashing, and reward mechanisms. This section analyzes the economic security of the network under a rational adversary model.

Consider an adversary who controls a fraction α of the total stake in the network. The adversary's objective is to maximize its expected rewards while minimizing the risk of getting slashed. Let R_{adv} denote the adversary's expected rewards, and let p be the probability of getting slashed for a malicious response.

The adversary's expected utility can be modeled as:

$$U_{\text{adv}} = R_{\text{adv}} - \alpha S \cdot \kappa \cdot p$$

where S is the total token supply, and κ is the slashing coefficient.

For the network to be secure against rational adversaries, the expected utility of honest behavior $U_{\text{adv}}^{\text{honest}}$ must exceed the expected utility of malicious behavior

$U_{\text{adv}}^{\text{malicious}}$ for all adversaries controlling less than a certain stake threshold $\alpha < \alpha^*$:

$$U_{\text{adv}}^{\text{honest}}(\alpha) > U_{\text{adv}}^{\text{malicious}}(\alpha) \quad \forall \alpha < \alpha^*$$

This condition can be satisfied by setting the slashing coefficient κ and the reward parameters appropriately, based on the expected frequency and magnitude of malicious behavior.

To quantify the security threshold α^*, we model the adversary's decision as a binary classification problem. Let $Y \in \{0, 1\}$ be a random variable representing the adversary's behavior, where $Y = 1$ indicates honest behavior and $Y = 0$ indicates malicious behavior. The adversary's stake α acts as a feature that influences the behavior.

Using logistic regression, we can model the probability of honest behavior as:

$$P(Y = 1|\alpha) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \alpha)}}$$

where β_0 and β_1 are model parameters that can be estimated from historical data on adversarial behavior in similar networks.

The security threshold α^* is then the value of α at which the probability of honest behavior drops below a certain level ρ:

$$\alpha^* = \inf \{\alpha \in [0, 1] : P(Y = 1|\alpha) < \rho\}$$

By setting the slashing coefficient κ and reward parameters to ensure $U_{\text{adv}}^{\text{honest}}(\alpha) > U_{\text{adv}}^{\text{malicious}}(\alpha)$ for all $\alpha < \alpha^*$, the Masa network can maintain a high level of security against rational adversaries.

6.1 Incentive Compatibility

The incentive structures within the Masa network ensure that honest behavior is the rational choice for participants. This is achieved through a combination of rewards for honest contributions and penalties for malicious actions. Key components include:

- **Staking Requirements:** Higher staking requirements for lower-performing agents incentivize improvements in service quality.
- **Reward Mechanisms:** Performance-based rewards align individual incentives with network goals.
- **Slashing Mechanisms:** Penalties for malicious behavior discourage attempts to undermine the network.

6.2 Economic Analysis

The economic model of the Masa network ensures that the costs of malicious behavior outweigh the potential benefits. By adjusting parameters such as the slashing coefficient and reward distribution, the network maintains a balance that promotes honest participation.

6.3 Resilience to Attacks

The stochastic nature of the staking dynamics and the diverse set of participants enhance the network's resilience to various attack vectors. The mean-field game framework ensures that no single participant can unilaterally influence the network's stability.

6.4 Future Enhancements

Continuous monitoring and periodic reviews of network parameters are essential to adapt to evolving threats. Implementing advanced machine learning models to predict and mitigate potential attacks can further enhance the network's security.

6.5 Challenges and Limitations

Relying on historical data for parameter estimation may have limitations. Further research is needed to address these challenges and enhance the security model.

7 Consensus

The consensus mechanism in the Masa network is a critical component that ensures the integrity and consistency of the ledger maintained by Validators. This section describes the consensus protocol employed by the Masa network and how Validators play a role in verifying data contributions and LLM outputs, in addition to maintaining the blockchain.

7.1 Consensus Protocol

The Masa network employs a Proof-of-Stake (PoS) based consensus protocol, where Validators are responsible for validating transactions, maintaining the ledger, and verifying the quality of data and LLM outputs. Validators are selected to propose and validate blocks based on the amount of MASA tokens they have staked.

7.2 Validator Selection

Validator selection in the Masa network is determined by a combination of factors, including:

- **Stake Size:** Validators with higher stakes have a higher probability of being selected to propose a block.
- **Performance:** Validators with a history of reliable performance and low latency are favored in the selection process.

- **Randomness:** A degree of randomness is introduced to ensure fairness and prevent predictability.

7.3 Block Proposal and Validation

The consensus process involves the following steps:

- **Block Proposal:** A selected Validator proposes a new block containing a set of transactions and results of data and LLM output verification.
- **Block Validation:** Other Validators verify the validity of the proposed block, including the transactions, data contributions, and LLM outputs. This involves checking the validity of transactions, ensuring data quality, and evaluating LLM responses based on predefined criteria.
- **Consensus Achievement:** Once a sufficient number of Validators have validated the block, it is added to the blockchain.

7.4 Verification of Data and LLM Outputs

Validators play a crucial role in maintaining the quality of data and LLM outputs by:

- **Data Verification:** Validators assess the quality and relevance of the data contributed by Worker Nodes. This includes checking for data integrity, completeness, and adherence to the specified standards.
- **LLM Output Verification:** Validators evaluate the responses generated by LLMs based on factors such as relevance, coherence, factual accuracy, and response time. The quality score $Q(x, y)$ for a response y to a prompt x is defined as:

  $$Q(x, y) = \alpha R(x, y) + \beta C(x, y) + \gamma F(x, y) + \delta T(y)$$

  where:

  - $R(x, y)$ is the relevance score of the response y to the prompt x,
  - $C(x, y)$ is the coherence score of the response y,
  - $F(x, y)$ is the factual accuracy score of the response y,
  - $T(y)$ is the response time score of the response y,
  - $\alpha, \beta, \gamma, \delta$ are the weights assigned to each factor.

7.5 Finality and Security

The PoS consensus protocol in the Masa network ensures finality and security through:

- **Finality Threshold:** A block is considered final after a certain number of confirmations from Validators, making it irreversible.
- **Slashing Conditions:** Validators that behave maliciously or fail to perform their duties are subject to slashing penalties, deterring malicious activities and ensuring honest participation.

7.6 Integration with Staking and Rewards

The consensus mechanism is tightly integrated with the network's staking and reward system. Validators earn rewards for participating in the consensus process and for their role in verifying data and LLM outputs. Their staking requirements are adjusted based on their performance. This integration ensures that Validators are incentivized to maintain the network's integrity and security.

7.7 Mathematical Framework

The validation process by Validators ties back into the mathematical framework of the network. Let $s_{i,t}$ denote the stake of Validator i at time t, and $r_{i,t}$ denote the token rewards earned by Validator i at time t. The utility of Validator i is defined as:

$$U_i = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,t}\right]$$

where $\gamma \in (0, 1)$ is a discount factor that models the time-value of rewards.

The reward $r_{i,t}$ is a function of the Validator's performance in consensus activities, data verification, and LLM output evaluation. Let $R_{V,i}$ denote the rewards from validating blocks, $R_{D,i}$ denote the rewards from data verification, and $R_{L,i}$ denote the rewards from LLM output evaluation. The total reward for Validator i at time t can be expressed as:

$$r_{i,t} = R_{V,i} + R_{D,i} + R_{L,i}$$

The staking dynamics can be modeled as a mean-field game (MFG), where each Validator optimizes their staking strategy $\pi_i : \mathcal{S} \to \mathbb{R}_+$ in response to the aggregate behavior of other Validators. The optimal staking policy π_i^* for Validator i is the solution to the following stochastic control problem:

$$\pi_i^* = \arg\max_{\pi_i} U_i(\pi_i, \pi_{-i})$$

subject to the staking requirements and network dynamics. Here, π_{-i} represents the staking policies of all Validators except i, and $U_i(\pi_i, \pi_{-i})$ is the expected utility of Validator i under the joint policy profile (π_i, π_{-i}).

By aligning individual staking strategies with the network's aggregate behavior, the MFG framework ensures that the overall system remains robust, decentralized, and fair, promoting sustainable and high-quality AI service provision within the Masa network.

8 Conclusion

This paper introduced Masa, a decentralized protocol for creating a global, incentivized, and self-improving AI network. Masa facilitates fair, open, and permissionless contributions of AI training data and compute resources, promoting innovation, transparency, and equitable reward distribution.

Masa's network consists of Validators, Worker Nodes, and Oracle Nodes, each playing crucial roles in maintaining network integrity, contributing data, and consuming AI services. The dual-token economic model, using MASA and TAO, incentivizes participation through performance-sensitive staking mechanisms.

Our multi-agent modeling framework, based on continuous-time Markov processes and mean-field games, provides a robust mathematical foundation for the network's dynamics. Performance-sensitive staking ensures high-quality service provision, while cooperative and repeated game models for data scraping and LLM inference foster continuous improvement among Worker Nodes.

The liquid democracy governance model enables efficient decision-making, allowing participants to delegate voting power while retaining direct voting capabilities. This model ensures community-driven development and robust governance.

Security analysis using rational adversary models and logistic regression demonstrates that the Masa network maintains high security by properly setting slashing coefficients and reward parameters. The integration of staking, slashing, and reward mechanisms ensures honest behavior is incentivized, enhancing network resilience.

In conclusion, Masa aims to democratize AI development and disrupt the centralized AI paradigm by leveraging cryptoeconomic incentives, LLMs, and decentralized infrastructure. This approach promises to unlock significant economic value and empower individuals and communities globally.

References

[1] Harris, J. D., & Waggoner, B. (2019). Decentralized & Collaborative AI on Blockchain. In *2019 IEEE International Conference on Blockchain (Blockchain)* (pp. 368-375). IEEE.

[2] Dinh, T. T. A., & Thai, M. T. (2018). AI and Blockchain: A Disruptive Integration. *Computer*, 51(9), 48-53.

[3] Voshmgir, S. (2019). *Token Economy: How Blockchains and Smart Contracts Revolutionize the Economy.* BlockchainHub Berlin.

[4] Buterin, V. (2017). Introduction to cryptoeconomics. *URL: https://vitalik.ca/general/2017/10/17/cryptoeconomics.html.*