FLock: Federated Machine Learning on Blockchain

Democratising AI through Decentralisation of Data, Computation, and Models

FLock Team

***Abstract*—The rapid pace of AI development has highlighted significant challenges in its creation and deployment, primarily due to the centralised control maintained by a few large corporations. Such an approach exacerbates biases within AI models due to a lack of effective governance and oversight. Furthermore, it diminishes public engagement and raises serious data protection concerns. The resulting monopolistic control over data and model outputs also poses a threat to innovation and equitable data usage, as users unknowingly contribute to data sets that serve the interests of these corporations.**

FLock democratises AI development and alignment through on-chain incentive mechanisms. By promoting open source development and data ownership, FLock facilitates an open and collaborative environment where participants can contribute models, data, and computing resources with rewards determined by on-chain consensus. This approach improves transparency and collaboration at scale without introducing biases from centralised entities. Ultimately, FLock enables diverse communities to develop purpose-built AI models, offering bespoke solutions tailored to their specific needs, revolutionising the landscape of AI development and deployment.

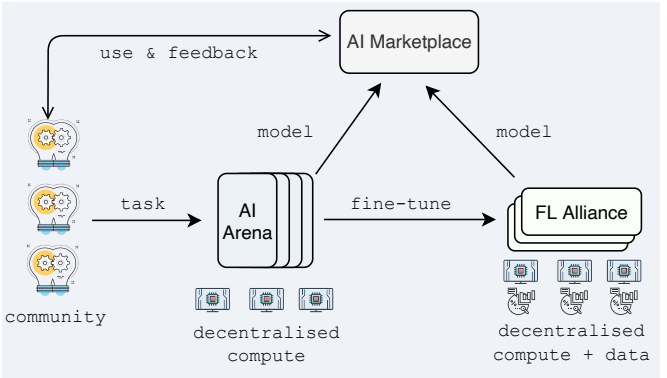


Fig. 1: FLock System Logic. Upon a task creation, the model is first trained and validated in AI Arena, a blockchain-based decentralised training platform, and then optionally further fine-tuned in FL Alliance using participants' local data. Finally, the model is deployed by applications in the AI Marketplace, where feedback will be used to further improve the model.

I. Introduction

Spanning all fields, collaboration has historically catalysed innovation. This is manifest in the case of the scientific and the digital. By pooling collective expertise, we have forged disruptive solutions at speed. At present, this ideal faces barriers when applied to AI development and deployment: notably, diminished public engagement, pervasive concerns regarding concentrated control, and data protection exerted by a handful of corporations. Meanwhile, blockchain technology [1], [2] has demonstrated its efficacy in multiple areas needing distributed corporations, such as decentralised finance [3], voting and governance. Research into and deployment of blockchain to transform AI development is now underway.

FLock, predicated on community involvement and a staunch commitment to data protection, is poised to spearhead the democratisation of AI ecosystem by using blockchain.

A. The Problems with Centralised Control over AI Creation

In the present day, the primary obstacle to innovation in the realm of AI is its centralised control. This centralised structure mandates that all AI training, decision-making processes, and data storage are controlled within a single entity or location [4]. This results in the following pitfalls:

Single Point of Failure: Vulnerability to disruptions from technical issues and cyberattacks.

Value Plurality: Lack of value plurality means biases of single entities are reflected in AI. With centralised institutions exerting absolute control over models [5], [6], the values of the output models are also centralised [7]. For instance, the world reimagined by Google's generative AI tool, Gemini, is widely criticised [8].

Data Protection: Providers of closed-source Large Language Models (LLMs) [9], such as OpenAI, have the capability to monitor all user interactions with their models, thereby raising significant data protection concerns. In addition, under this centralised framework, every user who interacts with a LLM becomes an unwitting contributor of data to these vast corporations that maintain ownership of the models. There is a pressing need to enhance the fairness of contribution incentives and to more accurately assess the value of user-contributed data.

Governance: Recent research [10], [11], [12] has highlighted a concerning trend in which the lack of governance has led to a pronounced exacerbation of biases and inaccuracies within the models.

Scalability: As the volume of data and complexity of tasks increase, limited processing power acts as a bottleneck.

Innovation: Progress is stifled in an environment where a limited number of entities have the means to experiment.

B. FLock's Solution

FLock [13], [14] is a blockchain-based platform for decentralised AI. As shown in Figure 1 and 2, FLock eliminates obstacles that prevent active participation in AI systems, em-

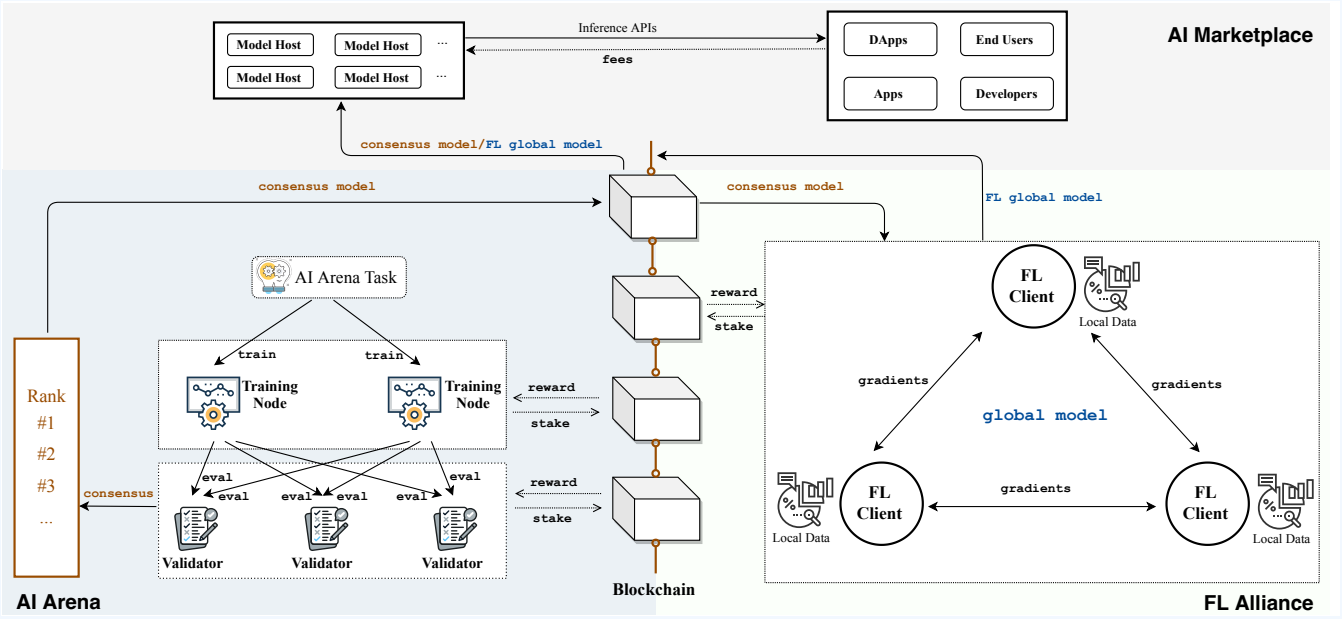


Fig. 2: FLock System Overview. When a task is created in AI Arena, it is first trained by training nodes. These nodes then submit their models to validators, who evaluate and propose scores for each submission. The validators reach a consensus on these scores to determine the ranking of the submitted models. The consensus model can then be assigned to FL clients, who fine-tune and improve it using their local data, resulting in the FL global model. The AI Arena consensus model or the FL global model can be deployed and hosted in the AI Marketplace, providing interfaces to various applications. AI Arena train nodes, validators and FL clients need to stake to participate the system, and will be rewarded based on their performance.

powering communities to contribute models, data, or computing resources in a modular and decentralised way. AI models can be trained and validated in AI Arena and further refined in Federated Learning (FL) Alliance. Harnessing blockchain technology, FLock introduces incentive mechanisms for participants, fostering a collaborative environment. This results in the development of a wide range of purpose-built models, created by, with, and for the communities, offering tailored solutions to meet specific needs.

II. FLock System Overview

The FLock system consists of the blockchain layer, AI layer, and various participants. Each component plays an essential role in ensuring the system's functionality and security.

A. Blockchain Layer

FLock's tokenomics incorporates a blockchain-based reward mechanism designed to enhance resilience against malicious user attacks. This robust security feature is underpinned by a carefully designed incentive mechanism. The blockchain layer acts as the foundation for both stakeholder participation and the distribution of rewards. This layer employs smart contracts to ensure that participants can securely lock in their stakes, fostering an environment of trust and transparency. The process is designed to incentivise participation by allocating rewards based on contributions, thus encouraging a more engaged and active community.

The blockchain layer's inherent security features safeguard against fraudulent activities, ensuring uncompromised integrity of staking and reward distribution. It is an critical component to support the model safety and improve resilience against malicious user attacks. By leveraging smart contracts, the system automates an efficient and fair rewards process. Automation reduces human error and ensures that rewards are distributed in a timely and fair manner.

B. AI Layer

The AI layer offers infrastructure for decentralised training, extracting and monetising knowledge from data. It encourages compute and data contributions from the community, using blockchain for rewards based on thier contributions.

- **AI Arena.** AI layer supports a conventional machine learning (ML) model training paradigm, optimising models directly on users' devices with their own or public data. To maximise the generalisation ability and performance of the final trained models, this layer is designed to encourage community members to contribute various public or local data, harnessing the broader community's power. By leveraging blockchain, it ensures contributors are continually engaged and rewarded based on the quantifiable impact of their data on improving the models.
- **FL Alliance.** Utilising the FL [15] approach, the AI layer enables thousands of participants to collaboratively train a global model, where data sovereignty is preserved by ensuring that no local data are transmitted at any stage of the training process. Within the AI layer, a model aggregation component allows participants to upload weights

from models trained on their unique local data. These weights are then aggregated to build an optimal global model, enhancing its generalisation capabilities and performance. The integration of training task automation and deployment orchestration components simplifies the process for users to join tasks and contribute valuable knowledge extracted from their data.

In FLock, AI Arena tasks will engage participants from the Web2 AI community, who possess the necessary computational resources to train and validate models using publicly available datasets. These trained models can be further refined through FL Alliance tasks, which draw in participants capable of contributing their own local data.

C. AI Marketplace

Once models are trained and fine-tuned through AI Arena and FL Alliance, they can be hosted on our platform. This platform serves as a comprehensive environment for deploying ML models, making them accessible within blockchain networks of virtual machines (VMs). By integrating with these networks, the platform facilitates the seamless execution and inference of complex ML models, providing real-time, scalable, and secure solutions.

The infrastructure for model management includes version control, model monitoring, and automated updates, ensuring that the models remain accurate and efficient over time. It can provide inference APIs or SDKs that developers can use to integrate these models into their applications.

Model hosts are compensated based on the quality and frequency of their contributions. They play a crucial role in generating inferences and maintaining the infrastructure.

D. Participants

There are various categories of participants in FLock.

1) **Task Creators:** Task creators will define the training tasks. Any participant who is willing to stake sufficient assets into the system or has already contributed to the system can potentially selected as a task creator. This broadens the range of stakeholders, confering a sense of ownership and active involvement.
2) **Training Nodes:** Training nodes compete in AI task training and are required stake tokens to be eligible. This requirement ensures a commitment to the network's integrity and facilitates a distributed, trust-based mechanism for task assignment. This stake acts both as a gatekeeper to maintain a high standard and as a foundational element in the network's security protocol, ensuring that nodes have a vested interest in proper execution and the overall health of the ecosystem.
3) **Validators:** Validators are responsible for evaluating work done by training nodes, submitting validation scores that influence reward distribution. They participate by staking tokens, which grants them the opportunity to validate tasks assigned to them, ensuring hardware compatibility and fair task distribution proportional to their stake. Upon completion of a task, they can withdraw their stake and claim rewards, which are calculated based on their performance and adherence to the expected outcomes. The design ensures that validators are incentivised to provide accurate and honest validations, thereby maintaining the quality and reliability of the network's computational tasks.
4) **Delegators:** Delegators contribute to the FLock system by supporting other participants' staking process, enhancing the network's validation capacity without directly participating in the task training or validation process. They provide stake tokens to other participants, thereby increasing the delegatees' potential to be selected for task assignments and influencing the overall reward distribution mechanism. Delegators share in the rewards earned by their associated delegatees, based on predefined algorithms that account for their staked contribution. Note that training nodes and validators who choose to accept delegation are free to choose a reward share ratio. The higher the ratio, the bigger the reward share their delegators will receive. The role of delegators allows individuals to participate in the network's training, validation and economic activities, leveraging their tokens to support delegatees, without needing the technical capabilities to train or validate tasks themselves.
5) **FL Clients:** With a FL framework, FL clients will contribute their local data to enhance the model trained for the AI Arena task. In each FL task, participants will be randomly designated as either proposers or voters. Proposers will be tasked with training the model within a FL framework, while voters will assess the training outcomes produced by the proposers. Both proposers and voters will receive rewards or face penalties based on their respective performances. FLock ensures that all participants are motivated to contribute effectively to the overall model improvement.
6) **Model Hosts:** The role of a model host in AI Marketplace involves deploying and managing trained models, providing infrastructure for secure and scalable execution, and enabling access through APIs and SDKs. The host ensures the models are kept up-to-date, monitors their performance, and facilitates integration into applications. Additionally, they will be compensated for their contributions to generating inferences and maintaining the system's integrity.

III. FLock Tokenomics

FLock aims to build a fair and incentive-compatible ecosystem, designed to foster collaboration and ensure long-term alignment within its community. This vision is realised through a strategically designed reward allocation system, an effective slashing mechanism for accountability, and the cultivation of active token demand.

A. Token Supply

1) Emission: FLock's ecosystem will feature FML[1] tokens, set to be distributed to various stakeholders through an initial token emission and a strategically designed reward allocation system over time. Participants will receive rewards in FML tokens based on their contributions to the system. Participants in the FLock system, such as training nodes and validators, are required to contribute computing or storage resources to complete model training and validation in order to receive rewards. This means that the value of the FML token will, at a minimum, correspond to the value of the resources consumed during these processes.

2) Burn: To participate in the FLock system, developers need to pay a registration fee. Similarly, users will also need to pay a fee to access and utilize the FLock-trained models. A portion of the fees collected from both developers and users will be burned, effectively reducing the token supply. This mechanism not only helps maintain a controlled token economy but also serves as a measure to counter inflation, ensuring long-term sustainability of the system's value.

3) Slash: FLock robust mechanisms ensure the integrity and reliability of the system by penalising participants that engage in malicious activities. If a participant is identified as acting against the system's rules or attempting to undermine the system through malicious actions, they are subjected to "slashing". Slashed tokens will be rewarded to the honest participants or burned. Slashing protects the system from immediate threats by disincentivising malicious actors and reinforces a culture of trust and cooperation among participants.

B. Token Demand

Active token demand is encouraged through multifaceted approachs as follows, showing the value of circulating tokens within the ecosystem.

1) Utility: Participants are required to stake FML to play a role. This reflects their vested interest in the integrity and success of operations. For task creators facing urgent needs to gather top-notch trainers for their model training or operating under tight deadlines, they may opt to pay additional FML as bounties. These bounties will then be distributed as payments to participants involved in those specific tasks to prioritize the training processes. Participants can be also supported by delegators through FML token delegation. By doing so, the system boosts the participants' stake within the FLock system and incentivises a symbiotic relationship. Delegators, in turn, earn a share of the rewards earned by their participants, fostering a competitive environment where participants are motivated to offer attractive terms to potential delegators.

2) Payment: Community members are able to access and utilise winning models which are trained and fine-tuned in AI Arena and FL Alliance, and hosted on AI Marketplace. End users enjoy rate limit in their access to such models based on their stake amount, beyond which they will be charged in FML as payment. On the other hand, model hosts need to stake FML in order to host winning models. They are able to customise whether and how to charge end users of these models. At inception phase, model hosts will receive part of the daily emission in order to incentivise their participation. Yet such incentives are expected to diminish over time. Overall, such design creates a sustainable and competitive environment in which demand and supply for cutting-edge models are dynamically balanced, fostering innovation and ensuring that the latest advancements continue to meet the evolving needs of the market. The payment mechanism also creates a non-negligible financial barrier in access to our models, thus helps mitigate potential DoS attacks from malicious participants. What is also note-worthy is that part of such payments will be burnt. This deflationary mechanism reduces the total token supply, potentially increasing the value of remaining tokens while ensuring that only serious participants engage in the network.

3) Governance Participation: Holding FML tokens grants members the power to influence the network's future through participation in the Decentralised Autonomous Organisation (DAO) governance. This not only decentralises decision-making but also adds a layer of utility and value to the tokens, as they become a key to shaping the ecosystem's development.

IV. FLock Incentive and Security

A. Incentive

FLock leverages well-designed incentive mechanisms to reward participants. The distribution of newly emitted tokens is carefully orchestrated across AI Arena tasks and FL Alliance tasks, reflecting a strategic allocation that hinges on the staking dynamics within each task category.

In our system, verified tasks are granted a share of daily rewards, serving as an incentive to foster the growth of the task creation ecosystem. This reward distribution is intentionally restricted to tasks approved by the DAO to safeguard the protocol from being exploited by low-quality or malicious tasks that could otherwise drain emissions without contributing meaningful value.

Each newly created AI Arena and FL Alliance task has the option to undergo a verification process conducted by the community-led DAO. This process is designed to ensure that tasks meet the necessary standards of quality and alignment with the ecosystem's goals. Once a task successfully passes verification, it becomes eligible for FML's daily emissions, providing the task creator with additional resources to incentivise participation and collaboration.

On the other hand, if a task is created permissionlessly without the FLock DAO's verification, the responsibility falls on the task creator to self-fund the task. This involves using their own FML to cover the costs associated with reward allocations for various participants. While this route allows for greater flexibility and decentralisation in task creation, it also places the financial burden of supporting the task's ecosystem on the creator. This mechanism is designed to balance innovation with quality control, ensuring that only well-constructed

[1] FML stands for "FLock My Life".

tasks benefit from community-supported rewards while still allowing for creative freedom in the ecosystem.

In the long run, this dual approach aims to encourage high-quality task creation, foster a vibrant and trustworthy ecosystem, and maintain the integrity of the FML reward system by aligning incentives with the community's standards and goals.

Once tasks are created, the distribution of rewards between DAO-verified AI Arena and FL Alliance tasks is dependent on the their relative stake amount of active tasks. As such, the rewards of FML allocated to all active AI Arena tasks will be:

$$R^{AI} = C_0 \cdot \frac{S^{AI}}{S^{AI} + S^{FL}}$$

and for all active FL Alliance tasks:

$$R^{FL} = C_0 \cdot \frac{S^{FL}}{S^{AI} + S^{FL}}$$

in which C_0 is the daily emission of FML, S^{AI} refers to the total stake amount of all active AI Arena tasks and S^{FL} refers to the total stake amount of all active FL Alliance tasks.

Note that at the initial phase, to incentivise participation, task creators will also receive a slice of the reward pool. This reward, however, is expected to be phased out over time.

In AI Arena, this allocation is meticulously calculated based on the aggregate staking contributions from task creators, training nodes, validators, and delegators for each task.

1) Rewards among AI Arena Tasks: Within the span of a single day, consider the situation where there are M AI Arena tasks with the total staking amounts of $(S_1, \ldots, S_M)$. The total staking amount, S_i, includes the stakes from all participants involved in this task. This means that the stakes from any type of user will influence the reward distribution among tasks. p is a system parameter that can be adjusted via DAO decision.

Assume the amount of daily emitted FML token is C_{AI}. For an AI Arena task with the total staking amount of S_i, its daily total rewards is:

$$R_i^{AI} = C_{AI} \cdot \frac{S_i^p}{\sum_{k=1}^{M} S_k^p}$$

For each AI Arena task, rewards are allocated among task creators, training nodes, validators, and delegators. In the initial version of FLock, if a validator has delegators, $d_1\%$ of their rewards are designated for these delegators. It is important to note that this distribution parameter is flexible and subject to adjustments through the FLock DAO governance.

2) Rewards among FL Alliance Tasks: A FL Alliance task should be is derived from a finished AI Arena task to be further fine-tuned. The initiation of an FL task automatically triggers the creation of a new FL contract. For each active FL Alliance task within the ecosystem, daily rewards are transferred to the respective FL smart contract, provided the task is still in progress and has not surpassed its maximum allotted lifecycle. This preliminary step ensures that the rewards are earmarked and protected for participants actively engaged in the task. Subsequently, upon meeting the predefined conditions, the FL smart contract autonomously distributes the rewards to the participants, according to their contributions.

B. Security

As shown in Table I, the FLock system's security is designed to be resilient against attacks.

Sybil attacks are mitigated by a requirement to stake a minimum amount of assets, making it costly to control multiple identities. Validators are kept unaware of the model origins, reducing the risk of collusion. Only the top-performing training nodes and validators receive rewards, discouraging poor performance and manipulation. To mitigate DoS attacks, the system implements rate limiting, preventing any single participant from monopolising resources. Free-rider attacks are addressed by rewarding only the top contributors, ensuring that participants who do not genuinely contribute cannot benefit. The use of dual datasets (Dataset A and B) in evaluations prevents lookup attacks, as optimising for one dataset does not guarantee success in the other. For FL model poisoning attacks, a majority voting system and slashing mechanism protect the model's integrity, punishing malicious actors and discouraging future attempts. These measures collectively fortify FLock against a range of threats, promoting a secure and reliable decentralised training environment for participants.

V. FLock Consensus in AI Arena

Figure 3 shows the overview of the workflow of a FLock AI Arena task.

A. Task Creators

Task creation is the primary stage of the training cycle. Creators define the desired models and submit tasks to the platform. Anyone who satisfies the criteria is eligible to be a task creator, making the system inherently democratic and accessible to a wide range of stakeholders. This inclusivity fosters a sense of ownership and active involvement within the FLock community.

To qualify as a task creator, users must meet one or more of the following criteria:

- Stake a sufficient amount of FML.
- Have successfully trained or validated a task previously, as evidenced by on-chain records.
- Possess a reputation in the ML space or be recognised as a domain expert in relevant fields, as verified by the FLock community.

If the task creator and the created task are verified by the FLock DAO, the task will be eligible for daily FML emissions. However, if the task creator chooses not to undergo verification by the community-led DAO, they must self-fund the task using FML to cover the costs associated with reward allocations for the participants.

In addition to gaining access to the desired training model, task creators may also earn rewards for their contributions. However, these rewards are expected to be gradually phased out over time.

Attacks	Description	FLock Mitigation
Sybil Attacks	An attacker might gain disproportionate influence in the FLock system by creating and controlling multiple fake identities of participants.	► Staking assets increases the difficulty of controlling many training nodes or validators. ► Blind validation mechanism prevents collusion between training nodes and validators. ► In each task, only the top k_1 training nodes and the top k_2 validators will rewarded, ensuring that participants with poor performance do not receive rewards.
DoS Attacks	An attacker might exhaust the FLock system resource and make it unavailable to honest participants.	► Rate limiting is implemented to restrict the frequency and volume of actions within a certain time frame, ensuring that no single participant can overwhelm the system.
Free-rider Attacks	Free riders benefit from a system without contributing fairly. In the FLock system, a free rider training node may randomly submit models without actuall training. Similarly, free rider validators give random scores instead of honestly evaluating models.	► In each task, only the top k_1 training nodes and the top k_2 validators will rewarded, ensuring that participants with poor performance do not receive compensation. ► FLock AI Arena consensus guarantees that honest participants who contribute diligently are appropriately recognised and rewarded, deterring free riders from exploiting the process.
Lookup Attacks	Training nodes could cheat by learning to predict past validation score calculations.	► Two datasets, i.e., Datasets A and B, are used as validation sets to evaluate the models. Consequently, even if a training node manages to optimise its performance for Dataset A, it could still underperform on Dataset B. By carefully calibrating the rewards between Dataset A and B, FLock effectively motivates training nodes towards developing genuinely high-quality models.
FL Model Poisoning Attacks	In FL Alliance, an attacker may use biased or corrupted data during the training process to degrade the model's performance.	► By aggregating contributions, majority voting minimises the impact of single malicious participants. ► The slashing mechanism penalises malicious clients, deterring model poisoning by reducing their rewards and discouraging future attacks.

TABLE I: Summary of how FLock system mitigation against potential attacks.

B. Training Node and Validator Selection

In this setup, each participant first stakes in the system to be eligible to perform task training or validation.

In practice, rate limiting is adopted to determine the number of times participants can access validation for a given task. As illustrated in Figure 4, the likelihood of a participant being selected to validate a task submission increases with their stake. However, the rate at which validation frequency increases relative to the staking amount tends to diminish as the staking amount grows.

C. Training in AI Arena

We consider the dataset held by the training node, $\mathcal{D}_{\text{local}}$, which contains locally sourced data samples, comprising feature set X and label set Y, with each sample $x_i \in X$ corresponding to a label $y_i \in Y$. We define a predictive model f, aiming to learn patterns within $\mathcal{D}$ such that $f(x_i) \approx y_i$.

To quantify the prediction metric, accuracy as an example, the task trainer will introduce a loss function $L(f(x_i), y_i)$, assessing the discrepancy between predictions $f(x_i)$ and actual labels y_i. A generic expression for this function is:

$$L = \frac{1}{N} \sum_{i=1}^{N} l(f(x_i), y_i)$$

where N denotes the total sample count, and l signifies a problem-specific loss function, e.g., mean squared error or cross-entropy loss.

The optimisation goal is to adjust the model parameters θ to minimise L, typically through algorithms such as gradient descent:

$$\theta_{\text{new}} = \theta_{\text{old}} - \eta \nabla_\theta L$$

where η represents the learning rate, and $\nabla_\theta L$ the gradient of L with respect to θ. Utilising the aggregated dataset $\mathcal{D}$, parameter θ is iteratively updated to reduce L, consequently improving the model's predictive accuracy. This optimisation process is conducted over a predefined number of epochs E, each epoch consisting of a complete pass through the entire dataset $\mathcal{D}$.

D. Validation in AI Arena

Consider a selected group of validators, denoted as $V_j \in V$, each equipped with the evaluation dataset $\mathcal{D}_{\text{eval}}$ from the

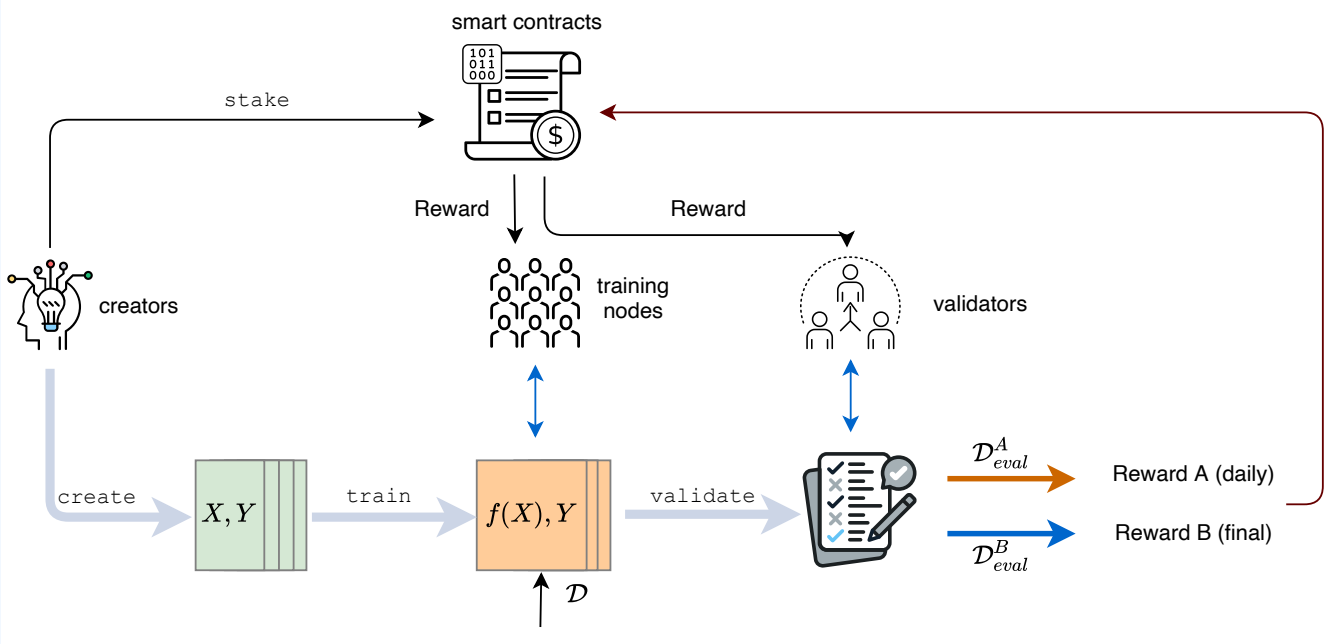


Fig. 3: Overview of the workflow of a FLock AI Arena task. Validators earn rewards based on their consensus scores. Two types of rewards are used to incentivise trainig nodes in order to mitigate their lookup/overfitting attacks.

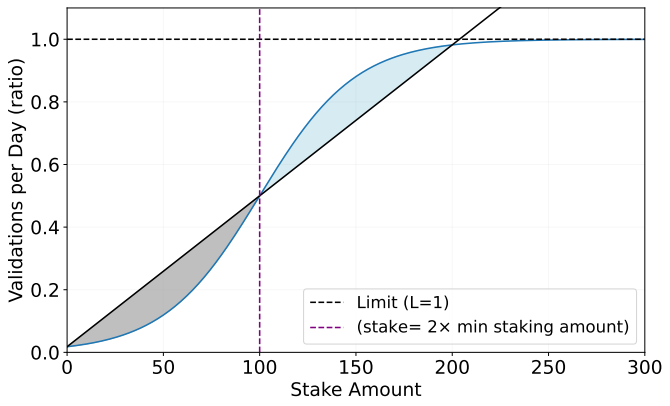


Fig. 4: Example: rate limiting for validation frequency.

task creator. This dataset consists of pairs (x_i, y_i), where x_i represents the features of the i-th sample, and y_i is the corresponding true label.

The model, trained by designated training nodes, is denoted as θ_p^{task}. The primary objective of θ_p^{task} is to predict the label $\hat{y}_i$ for each feature vector x_i contained within $\mathcal{D}_{\text{eval}}$.

To assess the performance of θ_p^{task} on $\mathcal{D}_{\text{eval}}$, we use an general evaluation metric denoted by $eval$. Here, we exemplify with accuracy, which is calculated as follows:

$$eval(\theta_p^{task}, \mathcal{D}_{\text{eval}}) = \frac{1}{|\mathcal{D}_{\text{eval}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{eval}}} \mathbf{1}(\hat{y}_i = y_i)$$

Here, $\mathbf{1}$ represents the indicator function that returns 1 if the predicted label $\hat{y}_i$ matches the true label y_i, and 0 otherwise.

The function $|\mathcal{D}_{\text{eval}}|$ denotes the total number of samples within the evaluation dataset.

Each predicted label $\hat{y}_i$ from the model θ_p^{task} is compared against its corresponding true label y_i within the dataset $\mathcal{D}_{\text{eval}}$. The calculated metric result (accuracy here) serves as a quantifiable measure of θ_p^{task}'s effectiveness at label prediction across the evaluation dataset.

E. Reward for Training Nodes in AI Arena

We assume there are n submissions $(O_1, \ldots, O_n)$ from n training nodes, and m validators $(V_1, \ldots, V_m)$, each with stakes $(s_1, \ldots, s_m)$. The stakes represent the validators' commitment and trust in the process, influencing the weight of their evaluations in the aggregated score.

- Each validator $V_j (1 \leq j \leq m)$ evaluates the n models submitted by the training nodes, producing a score vector $\vec{r}_j = (r_{j1}, \ldots, r_{jn})$. These scores reflect the perceived accuracy, reliability, or performance of each model according to predefined criteria. The outlier scores proposed by malicious validators will be ignored by honest validators before taking into amount in the following steps.
- The final score for each model from the training nodes is determined through a weighted aggregation:

$$\vec{r} = \left(\frac{\sum_j r_{j1} \cdot s_j}{\sum_j s_j}, \ldots, \frac{\sum_j r_{jn} \cdot s_j}{\sum_j s_j} \right)$$

  This means that the evaluations of validators with higher stakes have a larger impact on the final outcome.
- We then compute the following geometry series:

$$g_k = \frac{1-q}{1-q^m} \cdot q^{k-1}$$

in which k denotes a given training node's rank amongst its peers in the same task, whereas q represents the common ratio of the geometric series and m is the number of training nodes in a given task.

Participants in the FLock system, such as training nodes and validators, are required to contribute computing or storage resources to complete model training and validation in order to receive rewards. This means that the value of the FML token will, at a minimum, correspond to the value of the resources consumed during these processes.

- We finally compute the total rewards allocated for the training nodes as well as their delegators, which is based on the quality of their submission and their total amount of stake:

$$f_i(g_i, t_i) = \frac{g_i \cdot t_i^{\alpha_t}}{\sum_{k=1}^{n} g_k \cdot t_k^{\alpha_t}}$$

where α_t is a system parameter, and t_i the total stake amount of from the training node i as well as its respective delegators.

- Consider σ as the reward ratio set by training node i itself which determines the ratio of rewards shared between training node i and its respective delegators. Consider also that a training node i' stake in the task is t_n and stakes delegated to training node i is t_d, i.e. $t_i = t_n + t_d$, then the actual reward for training node i is:

$$f_i \cdot \left(\sigma + (1-\sigma) \cdot \frac{t_n}{t_n + t_d}\right)$$

F. Reward for Validators in AI Arena

For each validator V_j, we compute the distances between their score and the final aggregated score:

$$\begin{aligned}\vec{\Delta_j} &= (\Delta_{j1}, ..., \Delta_{jn}) \\ &= \left(\left|\frac{\sum_j r_{j1} \cdot s_j}{\sum_j s_j} - r_{j1}\right|, ..., \left|\frac{\sum_j r_{jn} \cdot s_j}{\sum_j s_j} - r_{jn}\right|\right)\end{aligned}$$

We define a distribution function f_i, which satisfies:

$$\begin{cases} f_i(\Delta_{1i}, s_1) + \ldots + f_i(\Delta_{mi}, s_m) = 1, \\ f_i \text{ decreases over the distance } \Delta_{ji}, \\ f_i \text{ increases over the stake amount } s_j. \end{cases}$$

To fulfill the three criteria, we can employ a modified version of the Softmax Function:

$$f_i(\Delta_{ji}, s_j) = \frac{e^{-\lambda_v \Delta_{ji}} \cdot s_j^{\alpha_v}}{\sum_{k=1}^{m} e^{-\lambda_v \Delta_{ki}} \cdot s_k^{\alpha_v}}$$

The parameters λ and α play crucial roles:

- λ_v
  - Purpose: Controls the sensitivity of the function to the distance Δ_{ji}. This distance measures the discrepancy between a validator's score and the aggregated score.
  - Effect: A higher λ_v increases the function's sensitivity to score accuracy, emphasising the importance of precise evaluations.
  - Selection Criteria: The choice of λ_v balances the need to penalise inaccuracies against the goal of rewarding nearly accurate evaluations.
- α_v
  - Purpose: Determines the influence of the stake amount s_j on the reward distribution, thereby adjusting the weight given to validators' financial contributions.
  - Effect: Allows for balancing between the importance of validators' financial commitment and their performance accuracy. A higher α_v gives more weight to the stake amount in the reward calculation.
  - Selection Criteria: Reflects the system's philosophy regarding the stake's importance relative to score accuracy. An α_v of 0 means stake amounts are ignored, while a higher value increases their impact.

If the validator finishes multiple (i.e., N) validation tasks, then it reward ratio is:

$$\sum_{i=1}^{N} f_i(\Delta_{ji}, s_j)$$

If a validator's stake in the task is s_v, and s_j is its accumulative stake by considering the total delegation amount S_d on this validator, i.e., $s_j = s_v + S_d$, the actual reward ratio for this valdiator is:

$$\left(\sum_i f_i(\Delta_{ji}, s_j)\right) \cdot \left(\sigma + (1-\sigma) \cdot \frac{s_v}{s_v + S_d}\right)$$

where σ is a system parameter.

G. Delegate Staking

Delegators may entrust their tokens to participants of their choosing to receive a passive investment income stream. The receivers can thus amplify their stake, influence, voting power, and rewards. These rewards are shared with the delegators, furthering cooperation. This extends participation to users who have tokens but lack the technical expertise to perform AI model training or validation.

Specifically, reward for the delegator depends on:

- The quality of the training nodes or validators selected for delegation.
- The amount of stake delegator has delegated.

Formally, reward for delegator who delegates to a training node can be calculated as:

$$f_i \cdot \left(\sigma \cdot \frac{t_d}{t_n + t_d}\right)$$

whereas f_i refers to the total reward distributed to the training node i and delegator based on the quality of the training node's submission, t_d is the stake amount from this given delegator, t_n is the stake amount from training node i and σ is the reward share ratio pre-determined by training node i .

	Total	Rewards for Training Nodes						Rewards for Validators	
		Reward A			Reward B			Reward	Reward
		Node 1	Node 2	Node 3	Node 1	Node 2	Node 3	Validator 1	Validator 2
Day 1	200	$20 \times 0.5 \times 0.39 = 3.9$	$20 \times 0.5 \times 0.33 = 3.3$	$20 \times 0.5 \times 0.28 = 2.8$	$180 \times 0.5 \times 0.39 = 35.1$	$180 \times 0.5 \times 0.33 = 29.7$	$180 \times 0.5 \times 0.28 = 25.2$	$100 \times 0.4 = 40$	$100 \times 0.6 = 60$
Day 2	200	$20 \times 0.5 \times 0.39 = 3.9$	$20 \times 0.5 \times 0.33 = 3.3$	$20 \times 0.5 \times 0.28 = 2.8$	$180 \times 0.5 \times 0.39 = 35.1$	$180 \times 0.5 \times 0.33 = 29.7$	$180 \times 0.5 \times 0.28 = 25.2$	$100 \times 0.4 = 40$	$100 \times 0.6 = 60$
Day 3	200	$20 \times 0.5 \times 0.39 = 3.9$	$20 \times 0.5 \times 0.33 = 3.3$	$20 \times 0.5 \times 0.28 = 2.8$	$180 \times 0.5 \times 0.39 = 35.1$	$180 \times 0.5 \times 0.33 = 29.7$	$180 \times 0.5 \times 0.28 = 25.2$	$100 \times 0.45 = 45$	$100 \times 0.55 = 55$
Day 4	200	$20 \times 0.5 \times 0.39 = 3.9$	$20 \times 0.5 \times 0.33 = 3.3$	$20 \times 0.5 \times 0.28 = 2.8$	$180 \times 0.5 \times 0.39 = 35.1$	$180 \times 0.5 \times 0.33 = 29.7$	$180 \times 0.5 \times 0.28 = 25.2$	$100 \times 0.45 = 45$	$100 \times 0.55 = 55$
Day 5	200	$20 \times 0.5 \times 0.39 = 3.9$	$20 \times 0.5 \times 0.33 = 3.3$	$20 \times 0.5 \times 0.28 = 2.8$	$180 \times 0.5 \times 0.39 = 35.1$	$180 \times 0.5 \times 0.33 = 29.7$	$180 \times 0.5 \times 0.28 = 25.2$	$100 \times 0.5 = 50$	$100 \times 0.5 = 50$

TABLE II: Example: Reward distribution among the participants in one task over five days.

Similarly, reward for delegator who delegates to a validator can be calculated as:

$$\left(\sum_i f_i(\Delta_{ji}, s_j)\right) \cdot \sigma \cdot \frac{s_d}{s_v + S_d}$$

in which $\sum_i f_i(\Delta_{ji}, s_j)$ is the Softmax function for validator mentioned above, whereas s_d refers to the stake amount from a given delegator and s_v is the stake amount of the validator the delegator delegated to.

In the future, FLock delegate staking has the option to be integrated with existing restaking platforms to attract users from a border blockchain community.

H. Various Validation Sets

To mitigate the lookup attacks from malicious training nodes, FLock validators adopt diverse validation datasets. Specifically, for a AI Arena task spanning x days, the validation dataset used during the initial $x-1$ days differs from that of the final day. These distinct validation datasets are associated with two types of rewards: **Reward A** for the initial period and **Reward B** for the final day. This strategic approach enhances security by varying the data against which training nodes are validated, thereby complicating any potential malicious attempts to exploit predictable validation scenarios.

- For each AI Arena within the ecosystem, the rewards mechanism for training nodes is thoughtfully designed to comprise two distinct parts: **Reward A** and **Reward B**.
  - **Reward A** provides a daily contingent reinforcement schedule, incentivizing participant engagement with AI Arena tasks through immediate gratification. This continuous reward mechanism fosters sustained participation by providing consistent feedback and reinforcing contributions.
  - **Reward B**, implements vesting, releasing tokens upon successful task completion within a predetermined lifespan. This incentivizes participants to both engage in tasks and ensure their timely and efficient completion. Vesting also functions as a quality control mechanism, promoting focused contributions aligned with project deadlines.
  - Reward A of the AI Arena task is:

$$R_i^{AI,A} = \delta \cdot R_i^{AI}$$

  - Reward B of the AI Arena task is:

$$R_i^{AI,B} = (1-\delta) \cdot R_i^{AI}$$

  δ is a configurable system parameter.

I. Example

We consider the rewards for the participants in task 1. We assume that:

- Distribution among the participants with a task is: task creator (0%), training nodes (50%), and validators (50%);
- There are three training nodes: Node 1, Node 2 and Node 3, meaning m, the number of training nodes in a given task, is 3;
- Nodes 1, 2 and 3 rank first, second and third respectively, and assume their respective rankings remain the same every day and they have the same amount of stake;
- q, which is the order of geometry series, is 0.85;
- There are two validators: Validator 1 and Validator 2;

Thus, their reward distribution during the five days are:

- Day 1: (Node 1: 0.39, Node 2: 0.33, Node 3: 0.28), (Validator 1: 0.4, Validator 2: 0.6);
- Day 2: (Node 1: 0.39, Node 2: 0.33, Node 3: 0.28), (Validator 1: 0.4, Validator 2: 0.6);
- Day 3: (Node 1: 0.39, Node 2: 0.33, Node 3: 0.28), (Validator 1: 0.45, Validator 2: 0.55);
- Day 4: (Node 1: 0.39, Node 2: 0.33, Node 3: 0.28), (Validator 1: 0.45, Validator 2: 0.55);
- Day 5: (Node 1: 0.39, Node 2: 0.33, Node 3: 0.28), (Validator 1: 0.5, Validator 2: 0.5);

The reward distribution among the participants in task 1 during the 5 days is shown in Table II.

VI. FLock Consensus in FL Alliance

Figure 5 depicts the workflow of a FL Alliance task in FLock. As shown in our work in leveraging blockchain to defend against poisoning attacks in FL Alliance [14], FLock adopts a distributed voting and a reward-and-slash mechanism to construct secure FL Alliance systems.

A. Task Creators

Similar to task creation in AI Arena, an FL Alliance task creator must satisfy predefined criteria. Only FL Alliance tasks verified by the FLock DAO will be eligible for rewards from

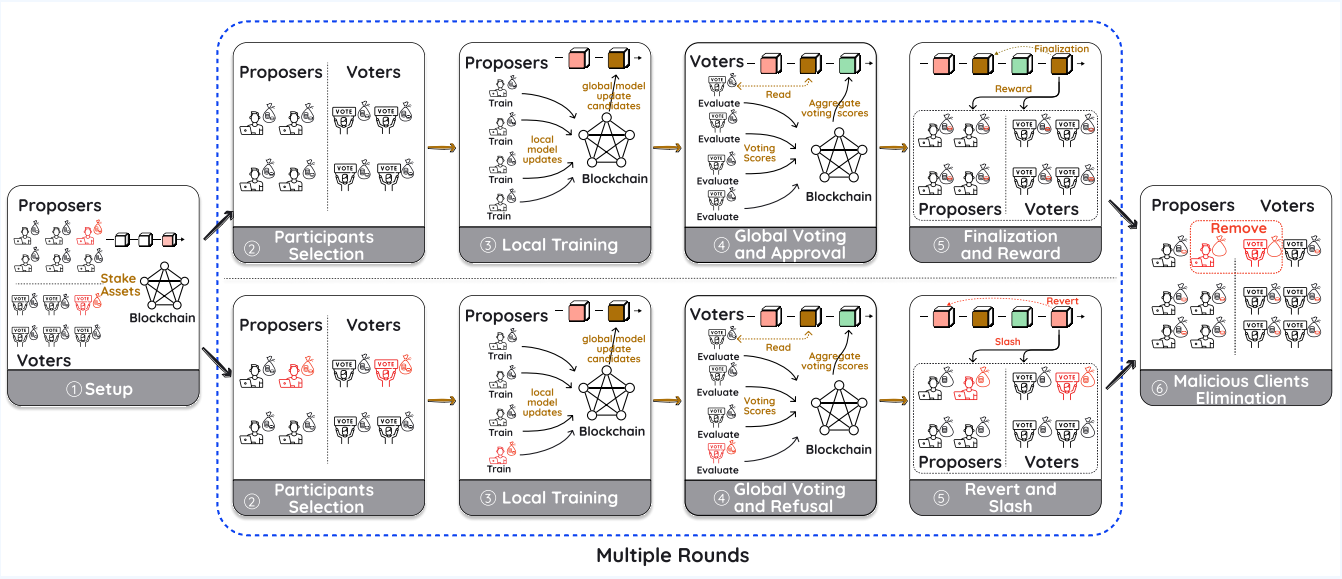


Fig. 5: Overview of the workflow of a FLock FL task, adopted from our previous work [14]. During each round, FL clients are randomly selected to act as proposers or voters. Their staked amount can be either rewarded or slashed based on the outcome of the majority voting results. Malicious clients will ultimately be removed from the FL system.

the daily emissions. Otherwise, the FL Alliance task creator must self-fund the reward pool using their own FML.

B. Random Role Selection

Consider a FL Alliance task involving P participants, denoted as $\mathcal{P} = 1, \ldots, P$. To participate in the training process, each participant needs to stake a specified quantity of coins. Upon formally joining the training task, each participant's local dataset $\mathcal{D}_p$ is randomly partitioned into a training set $\mathcal{D}_p^{train}$ and a test set $\mathcal{D}_p^{test}$, which will not be shared with other participants at any time. At the beginning of each round t in the FL Alliance task, participants are randomly assigned roles as either a proposer (P_T) or a voter (P_V) through an on-chain random function. Subsequently, a model initialised or pre-trained model downloaded from AI Arena by one of the proposers is chosen at random temporarily to serve as the pioneering global model. The selected model's weights or gradients are then distributed to all participants, ensuring a unified starting point for local models. Proposers are responsible for training their local models using their own data and subsequently sharing the updated model weights or gradients with all participants. Voters, on the other hand, aggregate these updates from proposers. They then proceed to validate the aggregated model updates, resulting in the generation of a validation score.

C. FL Alliance Training

At the start of round t, proposers initially download the global model, denoted as θ^{t-1}, which was finalised in the previous round $(t-1)$. Using the local training dataset $\mathcal{D}_p^{train}$, proposers then proceed to update the model θ^{t-1} through E epochs of local training. Then the updated model θ^t of the current round t will be uploaded to the voters for evaluation.

D. FL Alliance Aggregation

Upon the completion of task training during round t, the voter gathers the local models $\{\theta_p^t\}_{p=1}^{P_T}$ from proposers. These models are then aggregated into the latest global model using a weighted averaging approach, as described below:

$$\hat{\theta}^t = \sum_{p=1}^{P_T} \beta_p \cdot \theta_p^t$$

Here, the weight β_p is defined as $\frac{n_p}{N}$, with $n_p = |\mathcal{D}_p^{train}|$ indicates the number of local training data samples for each proposer p, and $N = \sum_{p=1}^{P_T} n_p$ is the total number of training data samples across all proposers P_T.

E. FL Alliance Validation and Voting

- After the model aggregation process is finalised, the voter proceeds to evaluate the aggregated model $\hat{\theta}^t$, utilising their own local testing datasets $\mathcal{D}_p^{test}$. This evaluation phase involves the computation of a local validation score, s_p^t, which functions as a criterion for assessing the model's performance. These individual validation scores are then submitted to a smart contract for aggregation. Following the aggregation, the aggregated score is compared with the previous round's score, s_p^{t-1}, to assess progress or decline in model performance. The smart contract then determines the next steps for the aggregated model based on these scores: advancement to the next phase for satisfactory performance improvement, or a return to the preceding validated model to begin a new cycle of training, aggregation and evaluation, if progress is deemed insufficient.

Algorithm 1 FLock Federated Training.

T: Total number of global communication rounds
E: Total number of local model update epochs
θ_g: Global model
$\mathcal{D}_p^{train}$: local training dataset
$\mathcal{D}_p^{test}$: local test dataset

procedure INIT
  Download pre-trained model from AI Arena or initialise global model θ_g^t
  Broadcast θ_g^t to all participants
end procedure
procedure UPDATE(θ_g^t)
  Initialise local model θ_p^t
  Update model parameters $\theta_p^t \leftarrow \theta_g^t$
  $\theta_p^{t+1} \leftarrow \theta_p^t - \eta \nabla L(\theta_p^t; b)$ ▷ Local model update
  return θ_p^{t+1}
end procedure
procedure EVALUATE($\theta_t + 1^P$)
  $\theta_g^{t+1} \leftarrow \sum_{p=1}^{P} \frac{n_p}{N} \theta_p^{t+1}$ ▷ Model updates aggregation
  Res $\xleftarrow{\theta_g^{t+1}}$ $\mathcal{D}_p^{test}$ ▷ Evaluate aggregated model
  return Res
end procedure
procedure MAIN
  Random select initialisation leader $\rightarrow p_{lead}$
  $\rightarrow p_{lead}$ Do procedure **Init**
  for $t = 1$ **to** T **do**
    Random assign roles for all participants
    Proposer does procedure **Update**
    Voter does procedure **Evaluate**
    $\mathsf{vote}_p^t \xleftarrow{\text{calculate}} \text{Res}_p^t$
    $\mathsf{aggVote}^t = \sum_{p=1}^{P_v} \mathsf{vote}_p^t$ ▷ Votes aggregation
    $\theta_g^{t+1} \leftarrow \text{GlobalModelSelection}(\mathsf{aggVote}^t)$
    Broadcast θ_g^{t+1}
  end for
end procedure

$$\mathsf{vote}_p^t = \begin{cases} 1, & s_p^t \geq (1-\epsilon) \cdot s_p^{t-1} \\ -1, & s_p^t < (1-\epsilon) \cdot s_p^{t-1} \end{cases}$$

Here, ϵ is a hyperparameter within the range $(0, 1)$, designated to tolerate the permissible margin of performance decline across successive rounds.

- After receiving all reported voting results $\{\mathsf{vote}_1^t, ..., \mathsf{vote}_{P_V}^t\}$ from the validators, the aggregator will calculate the aggregated voting result via the following formula:

$$\mathsf{aggVote}^t = \sum_{p=1}^{P_v} \mathsf{vote}_p^t$$

For each round t, the finalised aggregated global model update is determined by the aggregated voting result:

$$\theta^t = \begin{cases} \hat{\theta}^t, & \mathsf{aggVote}^t \geq 0 \\ \theta^{t-1}, & \mathsf{aggVote}^t < 0 \end{cases}$$

F. FL Alliance Rewards for Participants

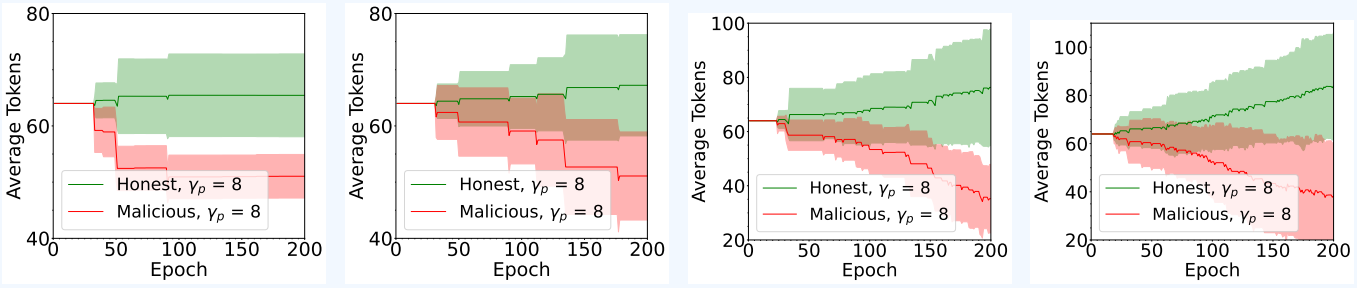
The aggregated voting result $\mathsf{aggVote}^t$ will also determine the rewards distribution for participants in a FL Alliance task.

- **Rewards and Penalties for Proposers/Training Nodes:** As shown in Algorithm 2, in any given round t, should $\mathsf{aggVote}^t$ be non-negative, all training nodes selected for that round will receive rewards. Conversely, a negative aggregate vote will result in penalties for these nodes.
- **Rewards and Penalties for Voters/Validators:** As shown in Algorithm 2, for round t, if $\mathsf{aggVote}^t \geq 0$, then validators who issued a positive vote will be rewarded,

Algorithm 2 Reward-and-slash design for FL clients.

$\mathcal{P}$: Set of participants at round t
$rewardPool$: Total reward pool
Bal_k: Stake amount of participant k
s_p: Slashed rate
$participantsDistributionRate$: Ratio of participants in one round
TotalRounds: Total numbers of communication round
$\mathsf{BalThreshold}_p$: minimum stake of participants

$pool_p \leftarrow initialRewardPoolSize$
for $t = 0; t <$ TotalRounds **do**
  Candidates $\leftarrow \{\}$
  for $k \in \mathcal{P}$ **do**
    if $\mathsf{Bal}_k \geq \mathsf{BalThreshold}_p$ **then**
      Candidates $\leftarrow$ Candidates $\cup \{k\}$
    end if
  end for
  $\mathcal{P}_p^t, \mathcal{P}_v^t \leftarrow$ Randomly select proposers and voters from Candidates based on $participantsDistributionRate$
  $\mathsf{aggVote}^t \leftarrow$ Aggregated voting results from the voters $\mathcal{P}_v^t$ at round t
  $roundRewardAmount \leftarrow \frac{pool_p}{\mathsf{TotalRounds}-t}$
  $roundTotalStakedTokensForGoodParticipants \leftarrow 0$
  if $\mathsf{aggVote}^t \geq 0$ **then** ▷ Compute the participants that should be rewarded in this round
    for $k \in \mathcal{P}_p^t$ **do**
      $roundTotalStakedTokensForGoodParticipants \mathrel{+}= \mathsf{Bal}_k$
    end for
  end if
  for $k \in \mathcal{P}_v^t$ **do**
    if $\mathsf{vote}_k^t \cdot \mathsf{aggVote}^t \geq 0$ **then**
      $roundTotalStakedTokensForGoodParticipants \mathrel{+}= \mathsf{Bal}_k$
    end if
  end for
  if $\mathsf{aggVote}^t \geq 0$ **then** ▷ Reward and Slash Proposers
    for $k \in \mathcal{P}_p^t$ **do**
      $\mathsf{Bal}_k \leftarrow \mathsf{Bal}_k + roundRewardAmount \cdot \frac{\mathsf{Bal}_k}{roundTotalStakedTokensForGoodParticipants}$
      $pool_p \leftarrow pool_p - roundRewardAmount \cdot \frac{\mathsf{Bal}_k}{roundTotalStakedTokensForGoodParticipants}$
    end for
  else
    for $k \in \mathcal{P}_p^t$ **do**
      $\mathsf{Bal}_k \leftarrow \mathsf{Bal}_k - \mathsf{Bal}_k \cdot s_p$
      $pool_p \leftarrow pool_p + \mathsf{Bal}_k \cdot s_p$
    end for
  end if
  for $k \in \mathcal{P}_v^t$ **do** ▷ Reward and Slash Voters
    if $\mathsf{vote}_k^t \cdot \mathsf{aggVote}^t \geq 0$ **then**
      $\mathsf{Bal}_k \leftarrow \mathsf{Bal}_k + roundRewardAmount \cdot \frac{\mathsf{Bal}_k}{roundTotalStakedTokensForGoodParticipants}$
      $pool_p \leftarrow pool_p - roundRewardAmount \cdot \frac{\mathsf{Bal}_k}{roundTotalStakedTokensForGoodParticipants}$
    else
      $\mathsf{Bal}_k \leftarrow \mathsf{Bal}_k - \mathsf{Bal}_k \cdot s_p$
      $pool_p \leftarrow pool_p + \mathsf{Bal}_k \cdot s_p$
    end if
  end for
  $t \leftarrow t + 1$
end for



(a) $\eta = 0.1$. (b) $\eta = 0.2$. (c) $\eta = 0.3$. (d) $\eta = 0.4$.

Fig. 6: Example: FL system with reward and slash mechanism under different values of the ratio of malicious clients (η), taken from [14]. The average balance of honest clients increases, while the average balance of malicious clients decreases over time.

while others will face penalties. Conversely, should the aggregate vote be negative, validators who aligned with this outcome are rewarded, whereas those who did not will be penalised.

G. Example

As illustrated in Figure 6(d), taken from our previous work [14], proper configuration of the slashing and reward mechanisms enables the expulsion of malicious FL participants from the system, while incentivising honest behaviour.

H. FL Alliance Improvement: ZKPs-based FL

FLock also adopts advanced techniques such as Zero-knowledge proof (ZKP) [16], [17] to construct secure decentralised AI training systems.

ZKPs for FL Alliance Aggregation: As demonstrated in our prior study, FLock incorporates ZKP to address the issues arising from the centralisation of the FL Alliance aggregator/server, as detailed in our earlier research [18]. Our FL system, which can be underpinned by both blockchain technology and ZKPs and function in the following manner:

- **Setup Phase:** Each participant, comprising N clients and an aggregator, generates their unique private/public key pairs. These pairs are directly associated with their respective blockchain addresses.
- **Client Selection Phase:** At the beginning of each epoch, a subset of n clients is selected from the total N by using Verifiable Random Functions.
- **Local Computation Phase:** The selected n clients start local model training to derive their individual model updates $w_1, w_2, \ldots, w_n$. Utilising the Pedersen commitment, each client encrypts their update as $Enc(w_i) = g^{w_i} \cdot h^{s_i}$, where g and h are predefined public parameters and s_i is a randomly generated number by the client. Following encryption, clients authenticate these updates using their private keys to produce a signature sig_i and subsequently transmit the compilation of their local model update, the generated random number, the encrypted update, and the signature $(w_i, s_i, Enc(w_i), sig_i)$ to the aggregator.
- **Aggregation and ZKP Generation Phase:** The aggregator aggregates the incoming local updates to form a unified global model update $w = \sum_{i=1}^{n} w_i$. It also calculates the collective encrypted value of this global update as $Enc(w) = \prod_{i=1}^{n} Enc(w_i)$ and signs this encrypted value to produce a signature sig. Utilising zkSnarks, the aggregator issues a proof π to validate the accuracy and authenticity of the aggregation process, based on the provided statement and witness, ensuring the integrity of both the individual updates and the aggregate model. Specifically, the aggregator then leverages zkSnark to issue a proof π for the following statement and witness:

$$\begin{cases} statement = (Enc(w_1), sig_1, Enc(w_2), sig_2, \\ \qquad\qquad ..., Enc(w_n), sig_n, Enc(w)) \\ witness = (w_1, s_1, w_2, s_2, ..., w_n, s_n, w) \end{cases}$$

  where the corresponding circuit $C(statement, witness)$ outputs 0 if and only if:

$$\begin{cases} \forall 1 \leq i \leq n, Enc(w_i) = g^{w_i} \cdot h^{s_i} \\ w = \sum_{i=1}^{n} w_i \\ sig_i \text{ is signed by the client } i \end{cases}$$

- **Global Model and Proof Dissemination Phase:** The aggregator distributes the global model update w and its encryption $Enc(w)$ back to the n clients. Concurrently, it broadcasts the validity proof π along with the encrypted global model update to the block proposers.
- **Blockchain Verification Phase:** Upon receiving the proof π and the encrypted global model update from the aggregator, block proposers verify π. If deemed valid, the hash of $H(Enc(w))$ is inscribed onto the blockchain, cementing the update's correctness.
- **Blockchain Consultation Phase:** As a new epoch initiates, the next cohort of n selected clients peruses the blockchain to verify the inclusion of $H(Enc(w))$. Upon successful validation, they proceed with their local training, guided by the insights gleaned from the aggregated global model update w.

VII. FLock Governance

FLock token holders are entitled to engage in the system's democratised governance through a DAO. To participate in governance, token holders typically need to lock their tokens in a smart contract. Each token can represent a vote, aligning the distribution of power proportional to users' stake.

Users can propose, debate, and vote on various aspects of development and management, from technical updates and protocol modifications to treasury management and community initiatives.

- **Proposing:** The FLock community actively shapes the protocol's future through a proposal system for all token holders. Proposals can range from addressing technical issues like bug fixes and algorithm optimisation to driving wider community impact, such as allocating treasury funds for research or launching educational programs.
- **Debating:** Proposed ideas are then open for discussion and critique within the FLock community. Token holders can engage in forums, discussions, and possibly even direct communication with developers to analyze the merits and potential consequences of each proposal. This debate fosters transparency and ensures that decisions are well-informed and considered from multiple perspectives.
- **Voting:** Once a proposal has been sufficiently debated, token holders cast their votes to decide its fate. The voting system likely incorporates mechanisms like weighted voting (where larger holdings carry more weight) or quadratic voting (which incentivises thoughtful contributions and discourages manipulation) to ensure fair representation.

The statement emphasises that FLock's governance model allows for continuous adaptation as the platform and the decentralised AI landscape evolve:

- **Policy Adaptation:** As new challenges and opportunities arise, token holders can use the voting system to modify existing policies or create entirely new ones. This ensures that FLock remains relevant and responsive to the changing needs of its community and the broader AI ecosystem.
- **Feature Implementation:** Proposals for implementing new features can be put forward and voted on, allowing the FLock platform to grow and evolve based on user demand and feedback. This fosters innovation and keeps FLock at the forefront of decentralised AI development.
- **Responding to Challenges:** The ability to quickly adapt policies and implement changes allows FLock to effectively respond to unforeseen challenges like security vulnerabilities, regulatory shifts, or market fluctuations.

As FLock and decentralised AI landscape mature, token holders can adapt policies, implement new features, and respond to emerging challenges.

VIII. FLock Applications

The FLock system can be used to construct centralised AI, which have been proven to applied in the following cases.

A. Decentralised AI for LLMs

- **Pre-training of LLMs:** FLock facilitates the pre-training of LLMs by leveraging a decentralised network whereby members can contribute computational resources and diverse data sets. This unlocks proprietary data that would otherwise remain inaccessible or unused in traditional, centralised open-source development. Diverse datasets ensure LLM versatility and ensures a broader representation of linguistic and cultural nuances, as well as community-defined values for LLMs.
- **Fine-tuning of LLMs:** Fine-tuning involves adapting a pre-trained model to perform specific tasks or improve its accuracy on particular types of data. FLock supports fine-tuning in several ways:
  - *Fine-tuning for Financial Transactions:* LLMs can be fine-tuned to act as intelligent agents for cryptocurrency transactions. Capabilities include transfers, swaps, and bridging between different cryptocurrencies. FLock's collaborations with platforms such as Morpheus Network and 0xscope can facilitate hosting these AI models, ensuring that they are accessible and operational for the community. This enables secure and efficient AI-driven financial transactions.
  - Fine-tuning for AI Companions: AI models can be fine-tuned to interact with users in more personalised and engaging ways, similar to those on platforms like Character.ai. FLock can host these sophisticated AI companions, enhancing user experience through more natural and context-aware interactions.

B. Decentralised AI for Stable Diffusion Models

The FLock system can be used to fine-tune Stable Diffusion text-to-image models. One critical component of this process involves Low-Rank Adaptation (LoRA) [19], which modifies certain parameters within the model's architecture to make it more adaptable to specific tasks without extensive retraining.

- **Fine-tuning LoRA:** LoRA is designed to adapt pre-trained models by introducing trainable low-rank matrices into the architecture. This technique allows for efficient adaptation with minimal additional computational cost and a smaller number of trainable parameters. In the context of FLock and Stable Diffusion Models, applying LoRA is particularly advantageous for several reasons:
  - *Community-Driven Enhancements:* By decentralising the fine-tuning process, FLock broadens participation in contributing specific knowledge and preferences. Artists, designers, and other creatives can input unique styles or features they wish to see enhanced, improving output quality and ensuring that it serves a wider array of cultural contexts and artistic expressions.
  - *Scalability and Accessibility:* Fine-tuning with LoRA can be scaled across multiple nodes, facilitating more widespread and continuously iterative improvements.
  - *Use Case Expansion:* By fine-tuning Stable Diffusion Models with LoRA, FLock can cater to specific indus-

tries or niches. For example, the model could be fine-tuned to generate medical illustrations for educational purposes, architectural visualisations for real estate, or unique art styles for digital media.

C. Decentralised AI for Linear Regression Models

Linear regression models [20] are fundamental tools in statistical analysis and predictive modeling, widely used for their simplicity and effectiveness in understanding relationships between variables. FLock applies these principles in a decentralised setting to address specific healthcare challenges, such as diabetes management.

Diabetes management presents a critical area where linear regression can be effectively utilised to predict patient outcomes based on various inputs such as blood sugar levels, diet, exercise, and medication adherence. FL Alliance facilitates the development of these predictive models with decentralised data sources in a way that respects patient data protection.

- *Data Protection and Security:* FLock allows multiple healthcare providers to collaborate in the model training process without actually sharing the data. This method is crucial for complying with stringent health data protection regulations such as HIPAA in the U.S. Each participant (e.g., hospitals, and clinics) retains control over their data, which is used to compute model updates locally. These updates are then aggregated to improve a shared model without exposing individual patient data.
- *Enhanced Model Accuracy and Reliability:* By integrating data from a diverse range of demographics and geographical locations, FLock can help develop more accurate and generalised linear regression models for diabetes management. This diversity is especially important in healthcare, where patient populations can vary significantly, affecting the reliability of predictive models.
- *Collaborative Innovation:* Different healthcare entities contribute to a common goal, accelerating innovation and leading to the discovery of novel insights into diabetes management and treatment strategies.

IX. Conclusion

FLock provides solutions to build decentralised AI through AI Arena, FL Alliance, and AI Marketplace. FLock dismantles obstacles that hinder participation in AI systems, enabling developers to contribute models, data, or computational resources in a flexible, modular fashion. FLock fosters the creation of a diverse array of models, meticulously crafted by and expressly for the communities they serve in AI models.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[3] S. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. Knottenbelt, "Sok: Decentralized finance (defi)," in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, 2022.

[4] P. Bellavista, L. Foschini, and A. Mora, "Decentralised learning in federated deployment environments: A system-level survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–38, 2021.

[5] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, 2023.

[6] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[7] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[8] "Why google's ai tool was slammed for showing images of people of colour," available at: https://www.aljazeera.com/news/2024/3/9/why-google-gemini-wont-show-you-white-people.

[9] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[10] NIST, "There's more to ai bias than biased data, nist report highlights," available at: https://www.nist.gov/news-events/news/2022/03/theres-more-ai-bias-biased-data-nist-report-highlights.

[11] IBM, "Shedding light on ai bias with real world examples," available at: https://www.ibm.com/blog/shedding-light-on-ai-bias-with-real-world-examples/.

[12] "When ai gets it wrong: Addressing ai hallucinations and bias," available at: https://mitsloanedtech.mit.edu/ai/basics/addressing-ai-hallucinations-and-bias/.

[13] N. Dong, J. Sun, Z. Wang, S. Zhang, and S. Zheng, "Flock: Defending malicious behaviors in federated learning with blockchain," *NeurIPS 2022 Workshops on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications. Runner-up Award.*, 2022.

[14] N. Dong, Z. Wang, J. Sun, M. Kampffmeyer, W. Knottenbelt, and E. Xing, "Defending against poisoning attacks in federated learning with blockchain," *IEEE Transactions on Artificial Intelligence*, 2024, https://arxiv.org/pdf/2307.00543.pdf.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017.

[16] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *IEEE Symposium on Security and Privacy*, 2014.

[17] J. Groth, "On the size of pairing-based non-interactive arguments," in *Advances in Cryptology–EUROCRYPT*, 2016.

[18] Z. Wang, N. Dong, J. Sun, W. Knottenbelt, and Y. Guo, "zkfl: Zero-knowledge proof-based gradient aggregation for federated learning," *IEEE Transactions on Big Data*, 2024.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[20] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2021.