

---

# Fluid Democracy in Federated Data Aggregation

---

Aditya Vema Reddy Kesari<sup>1</sup> Krishna Reddy Kesari<sup>2</sup>

## Abstract

Federated learning (FL) mechanisms typically require each client to transfer their weights to a central server, irrespective of how useful they are. In order to avoid wasteful data transfer costs from clients to the central server, we propose the use of consensus based protocols to identify a subset of clients with most useful model weights at each data transfer step. First, we explore the application of existing fluid democracy protocols to FL from a performance standpoint, comparing them with traditional one-person-one-vote (also known as *1p1v* or FedAvg). We propose a new fluid democracy protocol named *viscous-retained democracy* that always does better than *1p1v* under the same assumptions as existing fluid democracy protocols while also not allowing for influence accumulation. Secondly, we identify weaknesses of fluid democracy protocols from an adversarial lens in terms of their dependence on topology and/ or number of adversaries required to negatively impact the global model weights. To this effect, we propose an algorithm (FedVRD) that dynamically limits the effect of adversaries while minimizing cost by leveraging the delegation topology.

## 1. Introduction

Federated Learning (FL) offers a promising paradigm for training machine learning models by keeping sensitive data localized on client devices and only sharing model updates with a central server for aggregation. This approach prioritizes data privacy, as it allows for the collaborative improvement of a global model without direct access to raw client data. Typically, the central server averages all client weights to arrive at the global weights (McMahan et al., 2017), which is equivalent to one-person-one-vote (*1p1v*).

---

<sup>1</sup>IIT Bombay, India <sup>2</sup>Amazon, US [This work does not relate to my position at Amazon]. Correspondence to: Aditya <22b3985@iitb.ac.in>.

Accepted at the ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (CFAgentic@ICML'25), Vancouver, Canada. July 19, 2025. Copyright 2025 by the author(s).

This requires model weight transfer from all clients to central server at every training step resulting in a high cost overhead. In this paper, our first contribution is the application of fluid democracies to FL so as to leverage their inherent property of reducing number of clients that transfer data to the central server, thereby reducing data transfer cost.

Fluid democracies (Boldi et al., 2011; Kahng et al., 2018; Armstrong et al., 2024) enable a voter to delegate their vote to a more competent neighbor, who then casts a vote on both their behalf. Liquid democracy reduces the number of voters and achieves improved performance over *1p1v* voting under mild conditions. However, it is plagued by the disadvantage of few voters accumulating a high number of votes, which can affect performance if they are misaligned. This makes liquid democracy vulnerable to malicious agents. Viscous democracy was proposed to reduce influence accumulation by decaying the value of a vote with every delegation hop. Our second contribution is to show that viscous democracy achieves sub-*1p1v* performance more often than liquid democracy. Subsequently, we identify that this sub-*1p1v* performance arises due to a topological dependence. Thus, although viscous democracy addresses the issue of misaligned voters to some extent, the topological dependence introduced can be exploited by adversaries.

As our third contribution, to address these limitations, we propose *viscous-retained democracy* that a) achieves higher performance than *1p1v* under same conditions as liquid democracy, b) addresses the above weakness of viscous democracy by reducing adversaries' ability to exploit topological dependencies and c) maintains the beneficial characteristic of viscous democracy by limiting influence aggregation. However, it incurs a higher cost than both liquid and viscous democracies, that stems from a larger number of voters casting their vote. To this effect, we propose an algorithm that leverages viscous-retained democracy coupled with a threshold to solve the dual-optimization problem of cost and robustness by effectively leveraging the delegation topology.

Section 2 provides an overview of related work. Section 3 introduces the formal theoretical setup, outlining our social choice framework, delegation function and delegation graph. Section 4 presents our theoretical results on voting processes.

Section 5 details the experimental analysis conducted to validate the above theoretical results following which we propose an algorithm to effectively apply our theoretical and experimental findings in FL. Finally, Section 6 discusses our next steps and potential extensions.

## 2. Related Work

As this work lies at the intersection of adversarial defense and election mechanisms, we detail related work on both aspects in the context of FL.

### 2.1. Adversarial Defense in FL

Several existing solutions protect against vulnerability of FL to malicious client updates at the central server. (Blanchard et al., 2017) introduced robust aggregation strategies like Krum and Trimmed Mean, which selectively filter or disregard outlier updates. Building on this, (Yin et al., 2018) further explored Byzantine-tolerant gradient descent, providing theoretical guarantees for robustness. Recognizing the inherent resilience of median-based approaches, subsequent research gravitated towards geometric median variants. (Pillutla et al., 2022) proposed Robust Federated Aggregation (RFA), leveraging the geometric median of local model updates to effectively counter poisoning attacks. Beyond statistical robustness, (Cao et al., 2020) introduced FLTrust, a mechanism that evaluates the trustworthiness of client updates using root of trust to filter suspicious contributions. Zeno (Xie et al., 2020) prioritized updates that demonstrably improve the global model’s performance on a validation set. (Li et al., 2023) provides empirical comparisons of various schemes under diverse attack scenarios and data heterogeneity.

Contrary to the above methods, the central server in our proposed framework a) dynamically adapts to provide robustness leveraging information of the delegation topology adopted by the clients and b) accepts data transfer from a subset of clients in the delegation graph elected by the clients themselves, both occurring at every data transfer step.

### 2.2. Client Election Mechanisms in FL

FedVote (Yue et al., 2022) and FedVoting (Liu et al., 2021) utilize plurality voting, treating validation results as votes to determine the optimal global model. Similarly, (Sohn et al., 2020) introduce election coding to identify majority opinions during aggregation. DETOX (Rajput et al., 2019) employs a hierarchical aggregation strategy based on majority votes within groups of updates. Furthermore, election mechanisms are crucial for robustness against attacks during aggregation, as demonstrated by DRACO (Chen et al., 2018) and ByzShield (Konstantinidis & Ramamoor-

thy, 2021), both of which use majority voting to defend against malicious contributions.

More recent fluid democracy voting techniques such as liquid democracy (Kahng et al., 2018; Götz et al., 2021; Bersetche, 2024) offer avenues for voters to delegate their vote to a more informed neighbor so as to increase overall performance. Viscous democracy (Boldi et al., 2011; Armstrong et al., 2024) builds on it to reduce the extent of influence aggregation in addition to better overall performance in certain cases. In this paper, we present the applicability of fluid democracy protocols as client-side aggregation methods in FL to enable efficient weight transfer from clients to the central server at each training step, leading to cost benefits. In addition, we propose viscous-retained democracy and explore adversarial robustness of viscous and viscous-retained democracy in FL.

## 3. Model

In this section, we formally define our social choice framework, delegation process, delegation graph and specific graph topologies. (Berinsky et al., 2024; Armstrong et al., 2024).

### 3.1. Social Choice Framework for Federated Learning

We consider a fundamental epistemic social choice framework involving  $n$  voters, denoted as  $V = \{v_1, \dots, v_n\}$ , and two possible alternatives,  $A = \{a^+, a^-\}$ . We posit that  $a^+$  represents the objectively correct outcome, which all voters collectively endeavor to select. Each voter  $v_i \in V$  possesses a competency level  $q_i \in Q$  within the interval  $[0, 1]$ , signifying the probability that  $v_i$  will vote “correctly” (i.e., cast a vote for  $a^+$ ).

Mapping the above terminology to the FL setting, voters refer to clients,  $q_i$  is relevance of client  $v_i$ ’s data to the global task at hand,  $a^+$  is the set of universal data points across clients relevant to the global task while  $a^-$  is the set of universal data points across clients detrimental to the global task. Voting correctly refers to  $v_i$  providing model weight updates from data which is a subset of  $a^+$ .

### 3.2. Delegation Process

The voters are interconnected through an underlying graph  $G = (V, E)$ , where  $E$  is a set of undirected edges illustrating the relationships between voters through which delegations can occur.

Each voter  $v_i$  can undertake the following action: they can delegate  $\alpha$  of their vote to a voter in their set of neighbors,  $N_G(i) = \{j \in V | (i, j) \in E\}$  and they keep  $\beta(1 - \alpha)$  of their vote them supporting  $a^+$  with probability  $q_i$ . The setting of liquid democracy is described by  $\alpha = 1$  and viscous

democracy by  $\beta = 0$  with  $\alpha \in [0, 1]$ . In our proposed viscous-retained democracy, we have  $\beta = 1$  and  $\alpha \in [0, 1]$ . If a voter chooses to vote directly without delegation, we categorize them as a *guru*.

The delegation process is formalized by a function  $d : V \rightarrow V$ , where  $d(v_i) = v_j$  signifies a delegation from voter  $v_i$  to voter  $v_j$ . This delegation can travel transitively across multiple hops until it reaches a guru. The notation  $d^*(v_i)$  represents the repeated application of  $d(v_i)$  until a guru is reached, effectively identifying  $v_i$ 's guru. The set of all gurus is denoted as  $G(V)$ . In our paper, we employ the Upward Delegation process wherein each delegating voter only delegates to a neighbor more competent than themselves, hence no cycles are formed. We chose this delegation process because it is widely used in relevant literature (Berinsky et al., 2024; Armstrong et al., 2024) and intuitively aligns with the idea that a voter would like to delegate to a more competent neighbor.

### 3.3. Delegation Graph

The application of a delegation function creates a directed subgraph of  $G$ , which we refer to as the delegation graph,  $D = (V, \{(i, j) \in E \mid d(i) = j\})$ . This graph selectively includes directed versions of edges from  $E$  only when a delegation relationship exists between the connected nodes.

### 3.4. Accuracy

To ascertain the winner between the two alternatives, we employ weighted plurality voting, a mechanism consistent with May's theorem (May, 1952). The process involves each guru, leveraging their inherent competence, casting their entire accumulated weight in favor of a single alternative. The outcome of the election is decided by the alternative that garners the highest cumulative weight. The likelihood of  $a^+$  being chosen as the winning alternative is referred to as accuracy. This accuracy is directly influenced by the established delegations and the individual voter competencies while also exhibiting a dependence on the parameter  $\alpha$ .

### 3.5. Star and Chain Graphs

We define star and chain graphs by parameters  $(s, n_s, c, n_c)$  which establish a delegation graph comprising  $s$  *star* components with  $n_s$  nodes and  $c$  *chain* components with  $n_c$  nodes. Each star component features one guru at its center and  $n_s - 1$  delegators, while each chain component contains  $n_c - 1$  delegators, with a guru positioned at one end.

## 4. Vote Propagation

In this section, we analyze liquid and viscous democracy with respect to the *do no harm* property and propose viscous-retained democracy to address the shortfall. The three key

theoretical results are

- Liquid democracy satisfies *do no harm* property under mild assumptions.
- There exist cases where viscous democracy does not satisfy *do no harm* property when liquid democracy does
- Viscous-retained democracy satisfies *do no harm* property under the same assumptions as liquid democracy.

### 4.1. Do No Harm Property

We state the *do no harm* (DNH) property (Kahng et al., 2018) as the following :

For a mechanism  $M$  and a graph  $G$ , we define the gain over direct democracy  $D$  as

$$\text{gain}(M, G) = P_M(G) - P_D(G).$$

The objective of the *do no harm* property to never significantly underperform direct voting formally defined as -

- A mechanism  $M$  satisfies the *do no harm* property if for all  $\varepsilon > 0$ , there exists  $n_1 \in \mathbb{N}$  such that for all social networks  $G_n$  on  $n \geq n_1$  vertices,  $\text{gain}(M, G_n) \geq -\varepsilon$ .

The DNH property stipulates that any potential loss of delegative voting relative to direct voting must asymptotically approach zero as the graph size tends to infinity. This implies that while small instances might show delegative voting performing worse than direct voting, this disadvantage should vanish in large-scale settings.

We state that DNH is of utmost importance to us as the gold standard for FL aggregation of weights is FedAvg which is equivalent to direct democracy. Hence, always performing better than direct democracy would imply performing better than FedAvg in the FL setting.

We begin by stating the following lemma (Berinsky et al., 2024)

**Lemma 4.1.** *If  $M$  is a mechanism, there is an  $\beta \in (0, 1)$  and  $C : \mathbb{N} \rightarrow \mathbb{N}$  with  $C(n) \in o(n)$  such that*

$$\max\text{-weight}(G_n) \leq C(n) \quad (1)$$

$$\sum_{i=1}^n \text{weight}_i(G_n) \cdot p_i - \sum_{i=1}^n p_i \geq 2\beta n \quad (2)$$

*then  $M$  satisfies do no harm property.*

**Assumption 4.2.** In all the proofs in this section, we assume that the number of edges  $|E|$  in the delegation graph  $D$  having  $n$  vertices  $V$  satisfies  $|E| \geq C(n)$  for some  $C(n) \in \mathcal{O}(n)$ . We also assume that no two voters have equal competencies.

## 4.2. Liquid Democracy

In liquid democracy using the upward delegation process, a voter may either delegate their entire vote to a more competent neighbor or vote independently if no such neighbor exists (Kahng et al., 2018). The voting weight of a guru is determined by the total number of direct and indirect delegations they receive.

**Lemma 4.3.** *Liquid democracy satisfies do no harm property in upward delegation processes, provided that no guru has  $\mathcal{O}(n)$  delegators (direct or indirect), and Assumption 4.2 holds.*

*Proof.* From the first condition, Equation 1 holds directly.

To establish Equation (2), since delegation happens only to more competent neighbors, the minimum improvement in competence per delegation is  $c_{\min} > 0$ . By Assumption 4.2, no two voters have equal competence values. Additionally, since the number of delegations is at least  $\mathcal{C}(n)$ , the cumulative effect of competence differences across all delegations is some constant times  $n$ , satisfying Equation 2.  $\square$

## 4.3. Viscous Democracy

In viscous democracy (Boldi et al., 2011), each hop a delegation traverses by reduces its weight by a constant dampening factor,  $\alpha \in [0, 1]$ , referred to as the *viscosity*. Consequently, the weight of each voter  $v_i$  is defined as:

$$w_i = \begin{cases} 0 & \text{if } v_i \notin G(V) \\ \sum_{p \in \text{Path}(-, i)} \alpha^{|p|} & \text{otherwise} \end{cases}$$

where  $\text{Path}(-, i)$  denotes all delegation paths that terminate at  $v_i$  and  $G(V)$  is the set of all gurus. Key implications are:

- When  $\alpha = 1$ , viscous democracy simplifies to the standard liquid democracy model, where no weight is lost during delegation.
- For smaller values of  $\alpha$ , guru weights are directly influenced by the structure of the delegation graph. Specifically, the weight contributions from voters located further away from their guru diminish more significantly compared to those closer to their guru.

Although viscous democracy has shown advantages over liquid democracy in some upward delegation settings (Armstrong et al., 2024), it falls short of meeting a key criterion for a well-functioning voting system: the *do no harm* property, even under the assumptions used for liquid democracy. We present the following theorem to show the same.

**Theorem 1.** *Under the upward delegation process, viscous democracy does not satisfy do no harm property under identical assumptions used for liquid democracy.*

*Proof.* Consider  $n$  voters are divided into two equally sized families of disjoint subgraphs. The first  $n/2$  voters form chains of length 10, each voter in these chains has competence in the interval  $(b, c)$ ; the guru in each chain accrues at most

$$\sum_{k=0}^9 \alpha^k \leq \frac{1}{1-\alpha}$$

votes. The remaining  $n/2$  voters form stars of size 10, each leaf and each central guru having competence  $a$  and all other nodes having competence in the range  $(q, a)$ , so that the total weight of each guru is  $1 + 9\alpha$ . Before delegation, the mean competence of the group is at least  $\frac{q+b}{2}$ . After viscous delegation, the weighted mean competence is at most

$$\frac{\left(\frac{1}{1-\alpha}\right) c + (1 + 9\alpha) a}{\frac{1}{1-\alpha} + (1 + 9\alpha)}.$$

Concretely, with  $\alpha = 0.5$ ,  $q = 0.39$ ,  $a = 0.4$ ,  $b = 0.65$ , and  $c = 0.7$ , each chain root accumulates at most  $1/(1-\alpha) = 2$  votes while each star root accumulates  $1 + 9\alpha = 5.5$  votes. Under these weights, the mean competence after delegation falls from above 0.5 (in direct voting) to at most 0.48 (in viscous democracy). By the Condorcet Jury Theorem, as  $n$  increases, direct voting thus converges to the correct majority decision with probability tending to 1, whereas viscous delegation converges to the wrong decision with probability tending to 1. Hence, viscous democracy can strictly worsen collective accuracy and fails the 'do no harm' requirement. This completes the proof.  $\square$

We observe that the primary reason such cases arise is the dynamic nature of the total number of votes. As a result, there are scenarios where a larger fraction of high-competence votes is lost during delegation compared to lower-competence votes, leading to a decrease in the overall mean competence. Voters with relatively high competence located at the end of chain-like topologies tend to lose a large fraction of their voting weight, whereas voters with low competence in star-like topologies retain a significant portion of their voting weight. Consequently, each delegation does not necessarily result in a strict increase in competence, as there different amount of loss of votes. Motivated by this observation, we propose a new delegation mechanism: *viscous-retained democracy*.

## 4.4. Viscous-retained democracy

In viscous-retained democracy, similar to viscous democracy above, each hop a delegation traverses reduces its weight by a constant dampening factor,  $\alpha \in [0, 1]$ . In addition, in viscous-retained democracy, the delegator retains  $1 - \alpha$  fraction of their vote post delegation. Consequently,

the weight of each voter  $v_i$  is defined as:

$$w_i = \begin{cases} \sum_{p \in \text{Path}(-, i)} (1 - \alpha) * \alpha^{|p|} & \text{if } v_i \notin G(V) \\ \sum_{p \in \text{Path}(-, i)} \alpha^{|p|} & \text{otherwise} \end{cases}$$

where  $\text{Path}(-, i)$  denotes all delegation paths that terminate at  $v_i$  and  $G(V)$  is the set of all gurus. Key implications are:

- When  $\alpha = 1$ , viscous-retained democracy also simplifies to the standard liquid democracy model, where no weight is lost during delegation.
- The number of votes remain constant irrespective of the value of  $\alpha$ . The weight of gurus remain same as in the case of viscous democracy, thus retaining its ability to reduce accumulation of votes in the hands of misaligned voters.

We demonstrate that, in contrast to viscous democracy, viscous-retained democracy preserves the *do no harm* property requiring identical conditions as liquid democracy.

**Theorem 2.** *Viscous-retained democracy satisfies the Do No Harm property in upward delegation processes, provided that no guru has  $\mathcal{O}(n)$  delegators (direct or indirect), and Assumption 4.2 holds.*

*Proof.* From the first condition, Equation 1 holds directly.

To establish Equation 2, since delegation happens only to more competent neighbors, the minimum competence difference between any delegator and their chosen guru is  $c_{\min} > 0$ . Let  $m$  be the length of the longest path in the delegation graph  $D$ . Since no guru has  $\mathcal{O}(n)$  delegators,  $m$  is a constant. Thus,  $\alpha^m > 0$ . Hence, the minimum competence increase per delegation is  $c_{\min} \cdot \alpha^m$ . By Assumption 4.2, no two voters have equal competence values. Additionally, since the number of delegations is at least  $\mathcal{C}(n)$ , the cumulative effect of competence differences across all delegations is some constant times  $n$ , satisfying Equation 2.  $\square$

We motivate the use of viscous-retained democracy by highlighting its improved robustness against malicious agents compared to standard viscous democracy. In viscous-retained democracy, the most influential voters retain the same voting power as viscous democracy, while the overall vote mass in the system is increased. This enhancement not only increases robustness but also ensures that *do no harm* property is satisfied whenever it holds for liquid democracy.

## 5. Experimental Analysis

In networks composed entirely of star and chain topologies, viscous democracy can be viewed as a special case of

viscous-retained democracy, specifically the case in which only voters with weight greater than 1 are permitted to vote. In FL scenarios, where communication costs are significant, minimizing the number of active voters is essential. However, reducing the number of voters increases the system’s vulnerability to adversarial attacks. To address this trade-off, we propose a hybrid model that blends viscous and viscous-retained democracy. This hybrid model introduces a tunable threshold parameter  $\tau \in (0, 1)$ : lowering  $\tau$  increases the number of voters and thereby improves robustness, while higher values reduce communication overhead by limiting participation.

We consider a network  $n$  with 40 nodes such that  $n_c$  is 10 and  $c$  is 4, wherein the guru is positioned at one end of each and  $\alpha = 0.5$ . In order to ascertain the clients that will send their model weights to the central server, we pick a threshold  $\tau$ . The clients who have vote share higher than  $\tau$  after all clients vote end up transmitting their model weights to the central server.

Each transfer of model weights to the central server incurs a cost  $c$ . The total data transfer cost  $c_{total}$  is the product of cost  $c$  and the number of clients that are elected to send model weights to the central server.

We now consider an adversary with a fixed budget  $c_{adv}$ . The number of adversarial clients that can actively transfer data, denoted  $n_{adv}$ , is determined by the integer division of  $c_{adv}$  by the cost per client  $c$ . The adversary strategically arranges its  $n_{adv}$  clients in a topology that maximizes their influence. We note that while the adversary deploys a total of  $n_{adv}$  agents, only the  $n_{adv}$  clients are capable of casting votes i.e., they are the only ones to accumulate votes above the specified threshold.

We present our adversarial robustness analysis based on the optimal network topology an adversary might construct to attack the voting network of size  $n$ . For example, in the case of  $\tau = 1$ , which corresponds to standard viscous democracy, wherein only four agents cast votes, each accumulating slightly less than two votes. Therefore, the adversary would need to secure at least eight votes to gain a majority.

As an example, assuming that the cost per vote lies between 0.04 and 0.05, the adversary can afford to activate four core agents within their subnetwork. To collectively accumulate the required eight votes, each of these core agents must receive additional delegations, which would require connecting two more agents to each core as  $\alpha$  is equal to 0.5. As a result, the adversarial network must comprise at least 12 agents in total in this case. In Figure 1, we plot the minimum number of adversaries required to successfully gain a majority of the given voting network at a certain total cost of data transfer for various values of  $\tau$ .

From Figure 1, we find that lowering the threshold  $\tau$  in-

**Algorithm 1** FEDVRD

---

```

1: Input:  $w^0 \leftarrow$  random initialization
2: Server:
3: for Iteration  $t \leftarrow 1$  to  $T$  do
4:   Broadcast  $w^{t-1}$  to all clients
5:   Receive delegations, compute votes  $v_i^t$  and optimal
    $\tau$ 
6:   Receive updates from clients with  $v_i^t > \tau$ 
7:    $w_n^t \leftarrow \frac{\sum_i v_i^t w_{i,n}^t}{\sum_i v_i^t}$ 
8: end for
9: Client:
10: for Client  $i$  do
11:   Receive  $w^{t-1}$  and do local training  $w_i^t \leftarrow w^{t-1} -$ 
    $r_{t-1} \nabla \ell_i(w^{t-1}; \mathcal{D}_i)$ 
12:   Compute similarity  $s_i^t \leftarrow \frac{\langle w_i^t, w^{t-1} \rangle}{\|w_i^t\| \|w^{t-1}\|}$ 
13:   Delegate/vote and send decision to server
14:   if server requests weights then
15:     send  $w_i^t$ 
16:   end if
17: end for

```

---

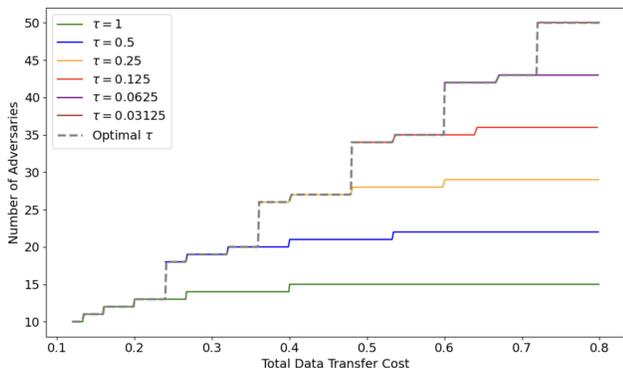


Figure 1. Plot depicting the optimal  $\tau$  for adversarial robustness at a fixed total data transfer cost

creases the cost of data transmission, since more voters are involved in the process. However, with a low threshold, the system becomes more robust, more adversaries are needed to successfully manipulate the outcome. This reveals a trade-off, as we allow for higher data transmission costs, we can adopt a lower threshold, which in turn increases the system’s resistance to manipulation. Thus, improving robustness comes at the expense of higher transmission cost. The optimal  $\tau$  for a certain cost for maximum adversarial robustness is shown in Figure 1.

In order to leverage this tradeoff effectively at each data transfer step, we propose FedVRD in Algorithm 1. At the start of time step  $t$ , the server broadcasts weights  $w_{t-1}$  to

all clients. The clients then perform local training to obtain their respective local weights  $w_i^t$ . The clients then compute the similarity  $s_i^t$  between their local weights  $w_i^t$  and  $w_{t-1}$  received from the server. The clients then broadcast  $s_i^t$  to their neighbors. Post this, all clients make the decision on whether they would like to delegate to one of their neighbors or transfer their own weights  $w_i^t$  for timestep  $t$ . This decision is relayed to the server. The server then computes an optimal  $\tau$  given the topology of the delegation graph. The server then requests for local weights  $w_i^t$  from clients with acquired votes greater than  $\tau$ . The selected clients then transmit their local weights to the server. The server applies the acquired votes to the weights to compute the global update  $w_t$  marking the end of timestep  $t$ . This process is repeated for number of timesteps required.

## 6. Future Work

Our future work includes obtaining empirical results on real world datasets and identifying optimal threshold values  $\tau$  for arbitrary graph structures. Another potential direction is to investigate if the number of instances that violate the *do no harm* property decreases strictly as  $\tau$  increases, in more general graph structures. Finally, establishing convergence guarantees for FL utilizing viscous and viscous-retained mechanisms is a promising direction.

## References

- Armstrong, B., Alouf-Heffetz, S., and Talmon, N. Optimizing viscous democracy. In Larson, K. (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 2643–2650. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/292. URL <https://doi.org/10.24963/ijcai.2024/292>. Main Track.
- Berinsky, A., Halpern, D., Halpern, J. Y., Jadbabaie, A., Mossel, E., Procaccia, A. D., and Revel, M. Tracking truth with liquid democracy, 2024. URL <https://arxiv.org/abs/2107.11868>.
- Bersetche, F. M. Generalizing liquid democracy to multi-agent delegation: A voting power measure and equilibrium analysis, 2024. URL <https://arxiv.org/abs/2209.14128>.
- Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL <https://proceedings.neurips.org>.

- [cc/paper\\_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf](https://arxiv.org/abs/2017.09.05).
- Boldi, P., Bonchi, F., Castillo, C., and Vigna, S. Viscous democracy for social networks. *Commun. ACM*, 54(6):129–137, June 2011. ISSN 0001-0782. doi: 10.1145/1953122.1953154. URL <https://doi.org/10.1145/1953122.1953154>.
- Cao, X., Fang, M., Liu, J., and Gong, N. Z. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *CoRR*, abs/2012.13995, 2020. URL <https://arxiv.org/abs/2012.13995>.
- Chen, L., Wang, H., Charles, Z., and Papailiopoulos, D. DRACO: Byzantine-resilient distributed training via redundant gradients. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 903–912. PMLR, 10–15 Jul 2018. URL <https://proceedings.mlr.press/v80/chen181.html>.
- Gözl, P., Kahng, A., Mackenzie, S., and Procaccia, A. D. The fluid mechanics of liquid democracy. *ACM Trans. Econ. Comput.*, 9(4), October 2021. ISSN 2167-8375. doi: 10.1145/3485012. URL <https://doi.org/10.1145/3485012>.
- Kahng, A., Mackenzie, S., and Procaccia, A. Liquid democracy: An algorithmic perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11468. URL <https://ojs.aaai.org/index.php/AAAI/article/view/11468>.
- Konstantinidis, K. and Ramamoorthy, A. Byzshield: An efficient and robust system for distributed training. In Smola, A., Dimakis, A., and Stoica, I. (eds.), *Proceedings of Machine Learning and Systems*, volume 3, pp. 812–828, 2021. URL [https://proceedings.mlsys.org/paper\\_files/paper/2021/file/d0b9a3081f811b2a307c38ad457a487c-Paper.pdf](https://proceedings.mlsys.org/paper_files/paper/2021/file/d0b9a3081f811b2a307c38ad457a487c-Paper.pdf).
- Li, S., Wu, Z., Song, W., Ma, X., Khan, N., Chen, K., and Song, X. An experimental study of byzantine-robust aggregation schemes in federated learning. *IEEE Transactions on Big Data*, 10, 2023.
- Liu, Y., Fan, Z., Song, X., and Shibasaki, R. Fedvoting: A cross-silo boosting tree construction method for privacy-preserving long-term human mobility prediction. *Sensors*, 21(24), 2021. ISSN 1424-8220. doi: 10.3390/s21248282. URL <https://www.mdpi.com/1424-8220/21/24/8282>.
- May, K. O. A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica*, 20(4):680–684, 1952. URL <https://onlinelibrary.wiley.com/doi/abs/10.2307/195210> 20:4&lt;680:ASOINA&gt;2.0.CO;2-S.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL <https://proceedings.mlr.press/v54/mcmahan17a.html>.
- Pillutla, K., Kakade, S. M., and Harchaoui, Z. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. doi: 10.1109/TSP.2022.3153135.
- Rajput, S., Wang, H., Charles, Z., and Papailiopoulos, D. Detox: A redundancy-based framework for faster and more robust gradient aggregation. In Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL [https://proceedings.neurips.cc/paper\\_files/paper/2019/file/415185ea244ea2b2bedeb0449b926802-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2019/file/415185ea244ea2b2bedeb0449b926802-Paper.pdf).
- Sohn, J.-y., Han, D.-J., Choi, B., and Moon, J. Election coding for distributed learning: Protecting signsgd against byzantine attacks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14615–14625. Curran Associates, Inc., 2020. URL [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/a7f0d2b95c60161b3f3c82f764b1d1c9-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/a7f0d2b95c60161b3f3c82f764b1d1c9-Paper.pdf).
- Xie, C., Liu, M., Gong, Z., Li, L., Li, B., and Liang, P. Zeno: Robust and communication-efficient federated learning with stragglers. In *International Conference on Machine Learning*, pp. 10617–10628, 2020.
- Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 10–15 Jul 2018. URL <https://proceedings.mlr.press/v80/yin18a.html>.

Yue, H., Li, X., Jiang, X., Zhang, X., and Yu, X. Fedvote: A voting-based federated learning aggregation method. In *Proceedings of the 2022 IEEE International Conference on Big Data (BigData)*, pp. 2432–2441. IEEE, 2022.