Equity and Equality in Fair Federated Learning

Hamid Mozaffari [1] **Amir Houmansadr** [1]

Abstract

Federated Learning (FL) enables data owners to train a shared global model without sharing their private data. Unfortunately, FL is susceptible to an intrinsic fairness issue: due to heterogeneity in clients' data distributions, the final trained model can give disproportionate advantages across the participating clients. In this work, we present Equal and Equitable Federated Learning (E2FL) to produce fair federated learning models by preserving two main fairness properties, equity and equality, *concurrently*. We validate the efficiency and fairness of E2FL in different real-world FL applications, and show that E2FL outperforms existing baselines in terms of the resulting efficiency, fairness of different groups, and fairness among all individual clients.

1. Introduction

Federated Learning (FL) is an emerging AI technology where *clients* collaborate to train a shared model, called the *global model*, without explicitly sharing their local training data. FL training involves a *server* which collects model updates from selected FL clients in each round of training, and uses them to update the global model. In FL, the performance of the global model varies across the clients due to heterogeneity in the data that each client owns. This concern is called *representation disparity* (Hashimoto et al., 2018) and results in unfair performance gaps for the participating clients. That is, although the accuracy may be high on average, some tail user whose data distribution differs from the majority of the clients is likely to receive a much lower performance compared to the average.

In this work, we look at FL fairness with two different lenses: **a)** *Equality*: whose goal is providing similar performances for all individual clients; **b)** *Equity*: whose goal is providing similar performances across all groups of clients (i.e., groups of majority and minority), where a group is defined as a set of clients with similar data distributions. The key question we try to answer is: *Can we design an efficient federated learning algorithm that achieves both equality and equity concurrently?*

Due to the heterogeneity in clients' data distributions, one single model cannot represent all the clients equally. There is a *trade-off* between training one global model and multiple global models; if we train one global model all the clients can utilize each other's knowledge, however it will be biased towards whom that have the majority of the population. On the other hand, if we train multiple models (e.g., as in IFCA (Ghosh et al., 2020), HypCluster (Mansour et al., 2020) and MOCHA (Smith et al., 2017)), we improve fairness, but each global model will lose the knowledge from excluded clients. To get the best of both worlds, we present **E**qual and **E**quitable **F**ederated **L**earning (E2FL), a novel FL algorithm to achieve both equality and equity. In E2FL, we train multiple global models, but in each round we combine all of the models into one global model to take advantage of the knowledge of all client groups.

2. Fairness Using Two Lenses: Equity and Equality

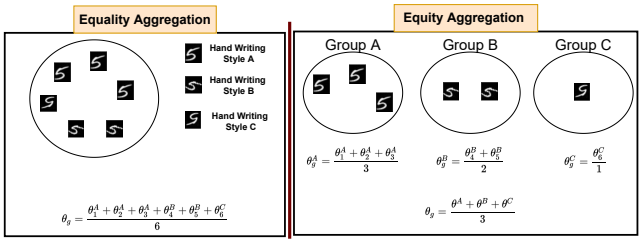


Figure 1. An example showing two different FL systems with two goals: equality (on left) and equity (on right).

Figure 1 shows an example of two FL systems where six clients want to learn a global model for prediction of handwritten digits. These clients have three handwriting styles: **(A)** normal handwriting style, **(B)** a little bit rotated handwriting, and **(C)** 180 degree rotated handwriting (upside-down). We consider each model update (θ_u^q for client u in group q) has the same effect on updating the global model, so each client update is like a vote. In this example, group A has the majority of the voters, and group B and C are in minorities. The left part of figure shows an FL in which

[1]Manning College of Information and Computer Sciences, University of Massachusetts Amherst, USA. Correspondence to: Hamid Mozaffari <hamid@cs.umass.edu>.

ICML 2022 Workshop on Responsible Decision Making in Dynamic Environments, Baltimore, Maryland, USA, 2022.

the goal is providing equality, so we give same chance (one vote) to each client to change the final model by an aggregation such as averaging (e.g., what we have in FedAvg). In this setting, the majority group with higher population (group A) has more influence on the final vote. On the other hand, the right part shows an FL in which the goal is providing equity. In this setting, first we aggregates the votes inside each group to find the group votes ($\theta_g^A, \theta_g^B, \theta_g^C$), and then aggregate the groups votes to produce the final model. In this setting, each client has the same chance (one vote) to influence its own group vote, and finally each group of voters have the same chance (one vote) to influence the final vote. We define two aspects of fairness in FL as follows:

Definition 1 (Equality: User-level Fairness): Trained global model θ is more *equalized* when its performance is more uniform across the individual clients participating in FL, i.e., when STD$\{F_u(\theta)\}_{u\in[N]}$ is smaller where STD$\{.\}$ is the standard deviation, and $F_u(.)$ denotes the local objective function of client u from N clients. Existing fair federated learning literature (Li et al., 2020; 2021; Smith et al., 2017; Hashimoto et al., 2018; Zhang et al., 2021; Mohri et al., 2019; Yu et al., 2020) use this definition in their designs.

Definition 2 (Equity: Group-level Fairness): Trained global model θ is more *equitable* when its performance is more uniform across the groups, i.e., when STD$\{$Avg$\{F_u(\theta)\}_{u\in[q]}\}_{q\in[Q]}$ is smaller where AVG$\{\}_{u\in[q]}$ denotes the average of performances for all the individual clients in the qth group, and there are Q total groups.

3. E2FL: Design

The key insight used in E2FL is converting the problem of model weight optimization (in standard FL) to the problem of ranking model edges (a technique recently proposed in (Mozaffari et al., 2021)). Therefore, in each round of E2FL training, the clients and the server exchange rankings for the edges of a randomly initialized neural network (called *supernetwork*), as opposed to exchanging parameter gradients. More specifically, each E2FL client computes the importance of the edges within a randomly initialized neural network on their local data, represented by a ranking vector. Next, E2FL server uses a majority voting mechanism to aggregate the collected local rankings into multiple global rankings based on the index of group they belong to. Finally, the E2FL server aggregates all the group rankings into one global ranking for next round of training. Applying the majority vote on the group rankings instead of all the local rankings helps E2FL enforce equity because each group has an equal vote to influence the global model. To provide equality in E2FL, if a client wants to use the model in a downstream task, they use their own group global ranking, instead of the global ranking, which is a better representation model for the client and its groupmates.

Algorithm 1 Equal and Equitable Federated Learning (E2FL) Algorithm.

Input: number of FL rounds T, number of local epochs E, number of selected users in each round n, number of groups Q, seed SEED, learning rate η, subnetwork size $k\%$
$\theta^s, \theta^w \leftarrow$ Initialize random scores and weights of global model θ using SEED
$R_g^1 \leftarrow$ ARGSORT(θ^s) {Sort the initial scores and obtain initial global rankings}
for $t \in [1, T]$ **do**
  $U \leftarrow$ set of n randomly selected clients out of N total clients
  for u in U **do**
    $\theta^s, \theta^w \leftarrow$ Initialize scores and weights using SEED
    $q =$ IDENTITY$(\theta^w, M^t_{g,q\in[Q]}, Q, D_u^{tr})$ {Identity estimation using binary masks of different groups}
    $\theta^s[R_g^t] \leftarrow$ SORT(θ^s) {Reorder the scores based on the global ranking}
    $S \leftarrow$ Edge-PopUp$(E, D_u^{tr}, \theta^w, \theta^s, k, \eta)$ {Train local scores on the local training data}
    $R^t_{u,q} \leftarrow$ ARGSORT(S) {Ranking of the client u with estimated group ID: q}
    return $R^t_{u,q}$
  end for
  $R^{t+1}_{g,q\in[Q]} \leftarrow$ VOTE$(R^t_{u\in U, q\in[Q]})$ {Majority vote aggregation inside each group}
  $R_g^{t+1} \leftarrow$ VOTE$(R^{t+1}_{g,q\in[Q]})$ {Majority vote aggregation among all the groups}
end for
function Vote $(R_{\{u\in U\}})$
  $V \leftarrow$ ARGSORT$(R_{\{u\in U\}})$ {Reputation of each edge in each local ranking}
  $A \leftarrow$ SUM(V) {Sum the reputations}
  return ARGSORT(A) {Order of the reputations}
end function

Our ranking-based FL training enables attractive fairness properties, as shown through our experiments, which is intuitively due to the following reason: In rank-based federated learning, each client computes a local ranking (i.e., a permutation of integers $\in [1, d]$ where d is the layer size), so each local ranking has a fixed norm (i.e., $\sqrt{1^2 + 2^2 + ... + d^2}$). This fixed norm of local updates makes the rank aggregation more fair as each local ranking has the same impact on the aggregated global ranking. On the other hand, in standard FL, when the server aggregates the local model updates into the global model, each local update has a different impact on the global model (because of their different l_2 norms). For example in FedAvg, the server averages the parameter updates for the d dimensions, therefore a large parameter update has more influence on the final average compared to a small parameter update.

In E2FL, different FL users gather together to learn a global model, but each one of them belong to a different group (which could be considered as known or unknown). In this section we assume the clients know their group IDs, and we defer our identity inference methods (using features

of rankings) to the full version (Mozaffari & Houmansadr, 2022), when the groups are unknown. Algorithm 1 describes E2FL training. In E2FL, the server trains multiple global rankings, each one belong to a different group. These global group rankings are showing different orders of importance of same supernetwork for different groups from least to most important edges. Each client participates in training of their group model by sending the local ranking they have. For aggregation, the server performs a majority vote among the local rankings (local votes) in each group, and then performs another majority vote among global group rankings (group votes) to find the global model for next round (i.e., global ranking that clients will start their training for next E2FL round.)

We detail a round of E2FL training and depict it in Figure 2, where we use a supernetwork with six edges $e_{i\in[0,5]}$ to demonstrate a single E2FL round and consider six clients $C_{j\in[1,6]}$ from three groups (handwriting style A, B, C) who aim to find a subnetwork of size k=50% of the original supernetwork.

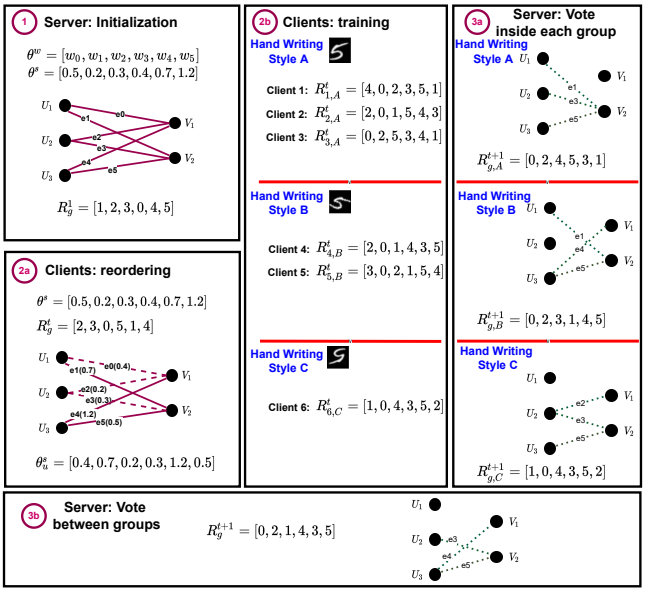


Figure 2. A single E2FL round with six clients from three groups and network of 6 edges. Please note that, all the operations in E2FL training are performed in a layer-wise manner.

Server: Initialization Phase (Only for round $t = 1$) : In the first round, the E2FL server chooses a random seed SEED to generate initial random weights θ^w and scores θ^s for the global supernetwork θ; note that, θ^w, θ^s, and SEED remain constant during the entire E2FL training. Next, the E2FL server shares SEED with E2FL clients, who can then locally reconstruct the initial weights θ^w and scores θ^s using SEED. Figure 2-① depicts this step. Recall that, the goal of E2FL training is to find the most important edges in θ^w without changing the weights. At the beginning, the E2FL server finds the global rankings of the initial random scores , i.e., $R_g^1 = \text{ARGSORT}(\theta^s)$. We define *rankings of a vector* as the indices of elements of vector when the vector is sorted from low to high, which is computed using ARGSORT function.

Clients: Calculating the ranks (For each round t) : In the t^{th} round, E2FL server shares the global rankings R_g^t with the clients. Each of the clients locally reconstructs the weights θ^w's and scores θ^s's using SEED. Then, each E2FL client reorders the random scores based on the global rankings, R_g^t. We depict this in Figure 2-(2a). For instance, the initial global rankings for this round are $R_g^t = [2, 3, 0, 5, 1, 4]$, meaning that edge e_4 should get the highest score ($s_4 = 1.2$), and edge e_2 should get the lowest score ($s_2 = 0.2$).

Next, each of the clients uses reordered θ_u^s and finds a subnetwork within θ^w using edge-popup algorithm (Ramanujan et al., 2020); to find a subnetwork, they use their local data and E local epochs. Note that, each iteration of edge-popup algorithm updates the scores θ_u^s. Then client u computes their local rankings R_u^t using the final updated scores and ARGSORT(.), and sends $R_{u,q}^t$ to the server where q is the group identifier. We defer our group inference methods we propose to full version (Mozaffari & Houmansadr, 2022). Figure 2-(2b) shows, for each client, the local rankings they obtained after finding their local subnetwork. For example, rankings of client C_1 are $R_{1,A}^t = [4, 0, 2, 3, 5, 1]$, i.e., e_4 is the least important and e_1 is the most important edge for C_1. Considering desired subnetwork size to be 50%, C_1 uses edges {3,5,1} in their final subnetwork in this round.

Server: Majority Vote (For each round t) : The server receives all the local rankings of the clients, i.e., $\{R_{1,A}^t, R_{2,A}^t, R_{3,A}^t, R_{4,B}^t, R_{5,B}^t, R_{6,C}^t\}$. Then, it performs a majority vote over all the local rankings inside each group, i.e., $\{A, B, C\}$. We depict this in Figure 2-(3a). Note that, for group q, the index i in $R_{g,q}^{t+1}$ represents the importance of the edge ith for clients in group q. For instance, in Figure 2-(3a), rankings of A are $R_{g,A}^t = [0, 2, 4, 5, 3, 1]$ and rankings of B are $R_{g,B}^t = [0, 2, 3, 1, 4, 5]$, hence the edge e_1 is the most important edge for group A, while the edge e_5 is the most important edge for group B. Next, the server performs a majority vote over all the group rankings of different groups $\{R_{g,A}^{t+1}, R_{g,B}^{t+1}, R_{g,C}^{t+1}\}$ to find the global ranking R_g^{t+1}. We depict this in Figure 2-(3b).

E2FL provides both equity and equality. E2FL provides both equity and equality. In this algorithm, at the final round of the learning, instead of using the global ranking, each group uses its own group global rankings. The global rankings can provide better performances to the majority groups as they have access to more training data they can train better group global ranking. For example, a client

of handwriting style A will use $f(x, \theta^w \bigodot M^t_{g,A})$ in their downstream classification task, where $M^t_{g,A}$ is the learned binary mask for group A at FL round t, and θ^w is the random weights (initialized randomly and kept fixed), and x is the test input. Please note that in E2FL and its variants, $M^t_{g,A}$ is the supermask trained for group A where for top $k\%$ of the top rankings of group ranking $R^t_{g,A}$, we put 1's and we set other masks to 0's.

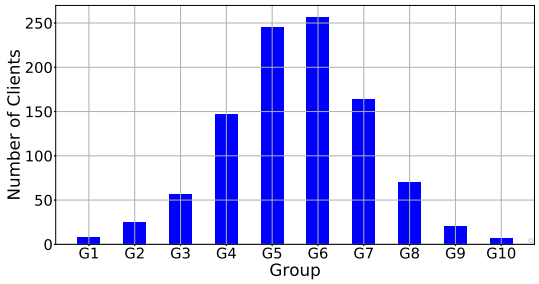
E2FL when the group IDs are unknown. In many applications, clients may be unaware of their protected attributes (i.e., the group they belong to). We propose one approach on server-side and three approaches on client-side for inferring group IDs. To infer the group IDs on the server-side, we propose to use a rank clustering approach to cluster clients into groups. We also design three approaches on client-side to infer the group IDs, where the clients can pick the right group based on their local training data. Using rankings allows us to exchange only the binary masks produced by each group ranking which lowers the communication cost compared to prior works. Each client can pick the right binary mask based on three approaches. First, each client can pick the binary mask that produces the smallest loss. Binary masks also enable the clients to find their matching group by a new novel idea from (Wortsman et al., 2020), where clients can infer the group ID using gradient based optimization to find a linear superposition of learned masks which minimizes the output entropy. We propose two variants of this approach, one based on a binary search and the other using OneShot optimization.

4. Experiments

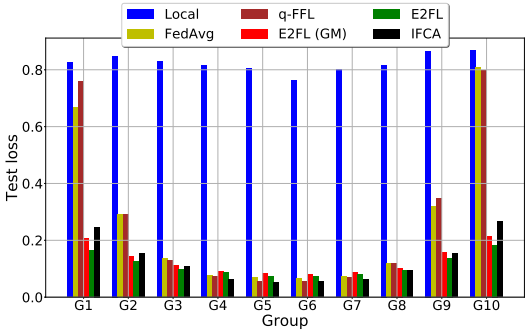
FairMNISTPerm: To measure the equity and equality, we release a new dataset for fairness experiments in FL application. We note that creating different data distributions by manipulating standard datasets such as MNIST has been widely adopted in the continual research community (Goodfellow et al., 2013; Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017), therefore, we create this dataset by rotating the images in MNIST for each group. Figure 3 (a) shows image samples in each group, and Figure 3(b) shows the number of clients in each group. In this dataset, we assign same number of data samples to 1000 clients with 10 different data distributions (with different number of users in each group). There are majority groups with large number of clients, e.g., G6 with 257 clients, and there are minority groups with small number of clients, e.g., G1 with 8 clients. In this dataset G1 and G10 are minorities, and G5 and G6 are in majorities. We defer more experimental results on FEMNIST (Caldas et al., 2018) and Adult Census Income Dataset (Kohavi et al., 1996) to the full version (Mozaffari & Houmansadr, 2022).

In Figure 3 (c), we compare the performance of different FL algorithms (Li et al., 2020; Ghosh et al., 2020; McMahan et al., 2017) by plotting the test loss of the global trained model for all the ten groups. In addition to the results of E2FL, which is the performance of the global ranking trained for each group, we report the results of the global model as E2FL (GM). Our experimental results on FairMNISTPerm show that: **(1) clients have motivation to participate in FL**. All the groups including minorities and majorities get benefit by participating in an FL framework. **(2) FedAvg gives more attention to majority groups.** Clients from majority groups can get more benefit by participating in FedAvg as they have more chance to be selected in each round, so they have more impact on the global model. FedAvg can achieve 97.61% mean test accuracy for all the individual clients (i.e., user-level fairness), but the mean of accuracies for groups is low as 93.89% which shows that this learning paradigm is focusing on user-level fairness more than on group-based fairness (equity). **(3) q-FFL improves equality while worsens equity**. q-FFL is helping the majority groups by ignoring the minorities. q-FFL is a user-level fairness framework, so it makes the results more fair compared to FedAvg in equality; however it produces more unfair results compared to equity. **(4) Training 10 different FLs (i.e., IFCA) is not the best situation for the minorities**. It is important that all the groups in FL share their knowledge. Figure 3 (c) shows that groups G1, G2,

(a) Data sample from each group



(b) Number of clients in each group



(c) Final test loss for all the groups on FairMNISTRotate

Figure 3. FairMNISTPerm: a new dataset to investigate equality and equity in FL application.

and G10 cannot get similar benefits by participating in IFCA since there is no shared knowledge, and these clients have access to limited data. **(5) E2FL is providing equality and equity**. While q-FFL reduces the variance of accuracies for all the clients by 4% while it increases the variance between groups by 81% compared to FedAvg. On the other hand our algorithm can reduce both variance of clients and groups by 93% and 95% respectively compared to FedAvg.

Acknowledgements

This work was supported by NSF grants 1553301 and 2131910.

References

Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018. URL http://arxiv.org/abs/1812.01097.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *arXiv preprint arXiv:2006.04088*, 2020.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, 2018.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Kohavi, R. et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020.

Li, T., Hu, S., Beirami, A., and Smith, V. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, 2021.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476, 2017.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics*, 2017.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Mozaffari, H. and Houmansadr, A. E2fl: Equal and equitable federated learning. *arXiv preprint arXiv:2205.10454*, 2022.

Mozaffari, H., Shejwalkar, V., and Houmansadr, A. Fsl: Federated supermask learning. *arXiv preprint arXiv:2110.04350*, 2021.

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., and Rastegari, M. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11893–11902, 2020.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. Federated multi-task learning. In *Neural Information Processing Systems (NIPS)*, 2017.

Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., and Farhadi, A. Supermasks in superposition. In *Advances in Neural Information Processing Systems 33: NeurIPS*, 2020.

Yu, T., Bagdasaryan, E., and Shmatikov, V. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Zhang, M., Sapra, K., Fidler, S., Yeung, S., and Alvarez, J. M. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.