Enhancing Fairness in Decentralized Learning with Clustering-based Personalization of Models

by Thibaud Trinca

Master Thesis

Prof. Anne-Marie Kermarrec
Thesis Advisor

Prof. Erwan Le Merrer
External Expert

Sayan Biswas, Martijn de Vos & Rishi Sharma
Thesis Supervisors

SACS - EPFL IC
BC 347 (BC Building)
Station 14
CH-1015 Lausanne

June 28, 2024

Acknowledgments

I would like to express my sincere thanks to my supervisors, Sayan Biswas, Martijn de Vos, and Rishi Sharma. Their availability and enthusiasm for engaging in a multitude of discussions and brainstorming sessions significantly enriched my research experience. I truly appreciated their thorough review of my thesis, which greatly enhanced its quality. Additionally, I would like to thank all the lab members for creating an excellent atmosphere. We had many enjoyable moments, creating a positive and supportive environment that is not commonly found in every lab.

I would also like to thank Professor Anne-Marie Kermarrec for her valuable insights during our discussions. Her expertise and feedback greatly enhanced my understanding of the subject matter. Additionally, she was very supportive throughout the project, making it a pleasure to work with her.

I'm also grateful to my family and friends for their support and encouragement throughout my master's program. Your belief in me has been a constant source of motivation.

Finally, I would like to thank Mathias Payer for providing the template for this report.

Lausanne, June 28, 2024 Thibaud Trinca

Abstract

Decentralized learning (DL) is an emerging approach in which nodes, each possessing unique data, collaboratively train models without coordination by a central server. This approach is relevant in diverse application domains, where datasets across nodes may follow similar or distinct distributions, necessitating personalized models to address the varying needs of these communities. These varying data distributions can lead to the formation of *clusters* of nodes with similar data patterns within the network. Existing DL approaches have a very limited focus on enhancing the fairness of the different *clusters* that emerge in the population. The models often display discrepancies in performance between clusters, harming the minority groups with fewer nodes. To address this, we introduce FAir Clustered And Decentralized lEarning (FACADE), a personalized learning algorithm specifically designed for fair model training with clustered data. We consider a setting where clusters of nodes in the network have similar learning objectives, but individuals a-priori do not know the identity of the cluster they belong to. FACADE (1) assigns nodes to their appropriate cluster over time, and (2) have nodes train a specialized model for each cluster in a decentralized manner. Unlike other DL approaches, each node in FACADE maintains one core model and several personalized model heads, forming multiple distinct models. In each round, a node trains one of these models and shares it with randomly selected neighbors. In the end, each learned head becomes specialized at treating data from the distribution of a specific cluster. In this work, we implement FACADE in a realistic environment and compare it against the three state-of-the-art baselines. We also introduce a new metric, which balances achieving high accuracy with minimizing disparities between groups. Our experimental results highlight our approach's superiority in achieving model accuracy and fairness, ensuring that every node has a model tailored for the data distribution of its cluster.

Contents

Acknowledgments 2

Abstract 3

1 Introduction 6

2 Preliminaries and related works 10

2.1 Problem Formulation 10

2.2 Decentralized learning 11

2.3 Fairness 11

2.4 Personalized learning 12

3 The FACADE Algorithm 14

3.1 Design 14

3.2 FACADE and fairness 16

3.3 Fair accuracy 17

4 Evaluation 19

4.1 Experimental setup 19

4.1.1 Datasets and model: 19

4.1.2 Baselines: 20

4.1.3 Experiment settings: 21

4.2 Results 22

4.2.1 Accuracy: 22

4.2.2 Fairness: 24

4.2.3 Communication cost: 26

4.3 Sensitivity to hyperparameters 27

4.4 Settlement analysis 27

5 Discussion 30

5.1 IDCA, the first version of FACADE 30

5.1.1 MNIST dataset: 30

5.1.2 Settlement plots: 31

5.1.3 Privacy: 31

5.2 Enhancing FACADE with various methods 31
5.2.1 Model leaking and exploration: 31
5.2.2 Custom losses: 32
5.2.3 Final algorithm: 32

6 Conclusion **34**

Bibliography **35**

A Appendix: Supplementary material **40**
A.1 Supplementary plots for Cifar-10 40
A.2 Imagenette dataset 40
A.3 Flickr-Mammals dataset 40

Chapter 1

Introduction

Decentralized learning (DL) is a collaborative learning approach that allows nodes to train a global machine learning (ML) model without sharing their private datasets with other entities [31]. According to a given communication topology, the nodes in DL directly communicate their models with other nodes, called *neighbors* (i.e., the other users involved). In each round, nodes locally train their model with their private datasets. Updated local models are exchanged with neighbors in the communication graph and aggregated on each node. The aggregated model is then used as the starting point for the next round, and the process repeats until model convergence is reached. Popular DL algorithms include Decentralized parallel stochastic gradient descent (D-PSGD) [31], Gossip learning (GL) [42], and Epidemic learning (EL) [9]. Furthermore, DL has been employed in various application domains such as healthcare [24, 34, 50] and Internet-of-Things (IoT) [14, 32].

A particular challenge in DL is that different nodes are likely to own data with differing characteristics [4]. It has been shown that this *data heterogeneity* can significantly slow down model convergence and attainable accuracy of DL [19, 22, 36, 48, 57].

This non-uniformity can be present at different levels [27, 36]: nodes might have different quantities of data, as well as different labels or features distributions. Feature heterogeneity is particularly present in healthcare, as different institutions have different data available, depending on acquisition differences, type of the medical device, or local demographics [41, 45]. In this work, we will only consider scenarios with nodes having heterogeneous feature distributions.

When there are similarities between nodes, a clustered distribution emerges, where data characteristics of different nodes form distinct groups. This means that the network can be clustered into subgroups containing nodes with similar data distributions [39, 48]. This setup commonly occurs in real-world applications, such as recommendation systems [46] and medical datasets [20].
Standard DL approaches, such as D-PSGD or EL, optimize the model towards the average data distribution. This negatively affects the *fairness* of the trained model in heterogeneous settings,

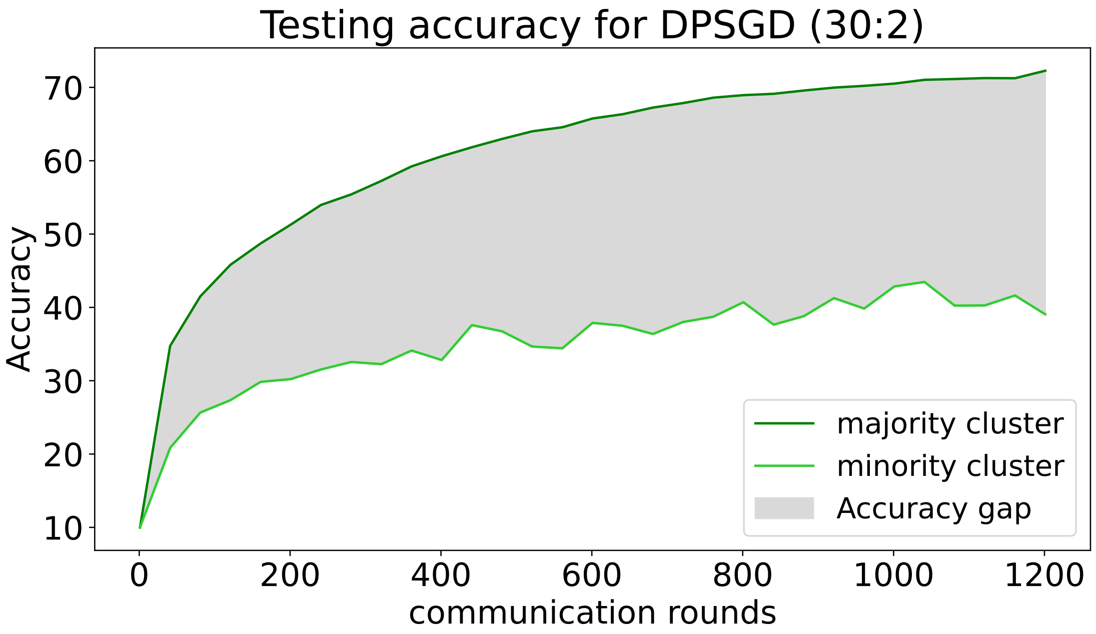


Figure 1.1: Model trained with D-PSGD results in lower accuracy for the 2 nodes in the minority cluster.

especially when dealing with minority groups in the data. Furthermore, as decision systems gain more trust, it has been shown that many of them have biases based on gender or race that affect minority groups [2], highlighting the importance of measuring fairness and addressing this problem.

For instance, consider a network of hospitals aiming to develop a powerful model for tumor detection through decentralized learning. If there are two brands of scanners in the market, each hospital would possess and use either brand A or B. Assuming slight differences in the acquisition process, the resulting feature distributions of the scans will differ between hospitals, creating data with clustered feature distribution skew. For example, features such as the contrast between the tumor and surrounding tissue or the level of image sharpness might vary significantly between the brands of scanners. Consider a scenario where scanners from brand A are much more widespread than those from brand B. In this context, a consensus-based model like D-PSGD would exhibit a strong bias towards the scan distribution of brand A. Ensuring fairness in such situations is essential, as having data with different a feature distribution should not negatively impact the prediction quality for nodes having equally contributed to the learning process.

To illustrate this example further, we conducted a small experiment where we trained a simple image classifier model (LeNet [26]) using D-PSGD in a 32-node network. We created two clusters: one with 30 nodes (the majority cluster) and one with 2 nodes (the minority cluster). We used the CIFAR10 dataset [25] for both clusters, with the only difference being that the images were turned upside down for the two minority nodes, creating a feature distribution shift. Figure 1.1 shows the averaged test accuracy for nodes in the minority and majority clusters. We observe a significant accuracy gap of more than 30% between the two clusters, demonstrating that the model produced by D-PSGD is less suitable for nodes in the minority cluster. Moreover, this gap is often hidden in the results when the global average is reported, as this metric is biased towards the majority due to their larger representation.

This example highlights the need for personalized models to ensure fairness for minority groups. In the context of DL, personalization techniques adjust the trained model to the unique characteristics of each node's dataset, promoting fair and equitable treatment across different communities. Existing approaches include learnable communication graph weights [28, 55], personalized model masks [8], or variations in sharing procedures [53]. However, many of these methods focus on individual nodes and do not leverage data similarity when handling networks with clustered non independent and identically distributed (non-IID) data.

In this paper, we present FACADE[1] (FAir Clustered and Decentralized lEarning), a personalized learning algorithm designed specifically for *clustered* non-IID data. Our approach ensures fair predictions across all nodes after model training, effectively managing features from heterogeneous data sources.

The main idea behind FACADE is to integrate a single model core with multiple specialized model heads at each node in the network. The shared core is trained collaboratively among all nodes using a method similar to D-PSGD, ensuring a strong and generalized representation of the overall data. Each node also maintains several heads, which can be independently trained and shared to address specific data characteristics. We implemented a mechanism that encourages nodes to train and share the head most suitable for their data distribution. As the training progresses, nodes with similar data distributions will converge towards the same head, allowing it to specialize in their particular type of data. This process ensures that each head becomes highly proficient in handling a specific distribution, resulting in more accurate and fair predictions across the network. Moreover, the clustering is implicit, as each node simply chooses the model that suits it best and does not need to know the cluster identity of the other nodes.

In this work, we also introduce a metric designed to quantify the fairness of algorithms by considering their overall performance, an aspect often neglected in current fairness measurements.

In summary, the contributions of our work are as follows:

1. We introduce a novel DL algorithm, named FACADE, designed to address the clustered non-IID data issue in a personalized and decentralized manner (Chapter 3). It specifically ensures

[1] We chose for the name FACADE because it represents the algorithm's ability to present a fair learning process on the outside, while managing the diverse data distributions internally, ensuring fair treatment for all nodes.

fairness by training a personalized model for each cluster.

2. We propose a new metric for measuring fairness, the *Fair Accuracy* (Chapter 3). Unlike traditional metrics, Fair Accuracy evaluates models based on their performance and fairness, ensuring that models achieve good results while minimizing disparities between groups.

3. Finally, we implement FACADE and provide a comprehensive analysis of its accuracy and fairness, comparing it against various baselines (Chapter 4). Our results demonstrate that our approach results in high model utility for all clusters and excels at maintaining fairness, even for highly imbalanced scenarios where one group significantly outweighs the other.

Chapter 2

Preliminaries and related works

2.1 Problem Formulation

We consider a set $\mathcal{N}$ of n nodes that can communicate to participate in the collaborative training of a personalized model. In particular, for $i = 1, \dots, n$, let $N_i \in \mathcal{N}$ denote the i-th node. Correspondingly, let Z_i denote the local dataset of N_i for $i \in [n]$.

We assume that non independent and identically distributed (non-IID) data locally held by the nodes in $\mathcal{N}$ are partitioned into k different distributions, $\mathcal{D}_1, ..., \mathcal{D}_k$ and let us denote the set of nodes whose data follow distribution $\mathcal{D}_j$ to be in set S_j for $j = 1, \dots, k$. Thus, $\mathcal{N}$ is partitioned into k disjoint clusters denoted by $S_1, ..., S_k$ such that $S_j = \{N_i \in \mathcal{N} : \ Z_i \sim \mathcal{D}_j\}$ for every $i \in [n]$ and $j \in [k]$. In this work, we consider a setting with two clusters, where one cluster typically outnumbers the other, forming groups that represent the *majority* and the *minority* of the population. While our study focuses on this two-cluster setup, the algorithm we propose is versatile and can be effectively applied to configurations involving more than two clusters, as shown in Section 4.3. We experiment with configurations where the majority-to-minority ratio varies from 1 to 15.

At the beginning of the training, each node is unaware of its cluster identity. For each node, we assume $|Z_i|$ to be finite, with each sample in Z_i drawn independently from the data distribution of its respective cluster, *i.e.*, $z \sim \mathcal{D}_j, \forall z \in Z_i$ if $i \in S_j$. Each sample z consists of a label y and feature x. Let $\mathcal{L} \colon \Theta \mapsto \mathbb{R}_{\geq 0}$ be the loss function associated with a sample z where $\Theta \subseteq \mathbb{R}^P$ represents the parameter space of the model we wish to train. Hence, the objective of the training process is to minimize each cluster's population error function:

$$F^j(\theta) = \mathbb{E}_{z \sim \mathcal{D}_j}[\mathcal{L}(\theta, z)] \quad \forall j \in [k] \tag{2.1}$$

In particular, we aim to find the optimal model for the j-th cluster $\theta_j^* \in \Theta$ given by:

$$\theta_j^* = \text{argmin}_{\theta \in \Theta} F^j(\theta) \quad \forall j \in [k] \tag{2.2}$$

In practice, having finite datasets for each node, we write $\mathscr{L}(\theta_j, \hat{Z}_i) = \frac{1}{|\hat{Z}_i|} \sum_{z \in \hat{Z}_i} \mathscr{L}(\theta_j, z)$ to denote the empirical loss evaluated by node N_i using $\hat{Z}_i \subseteq Z_i$ for the model θ_j, where $\hat{Z}_i$ is a subset of the data points held by N_i (*e.g.*, the training dataset used by N_i) for every $i \in [n]$.

2.2 Decentralized learning

In environments where reliance on a trustworthy orchestrator is impractical, decentralized learning (DL) [31] emerges as a viable alternative. Unlike traditional federated learning (FL) frameworks [40], DL operates within a distributed architecture, allowing each node to collaborate with its neighbors autonomously. This decentralized approach offers resilience against single points of failure and features greater scalability.
One notable aspect of DL is its ability to converge toward a consensus model through iterative peer-to-peer communication. In D-PSGD [31], a standard DL algorithm , each node N_i updates its model by performing τ Stochastic gradient descent (SGD) updates, sampling batches from its local dataset Z_i. The model weights θ are then shared with its neighbors, as defined by the static communication graph. Finally, each node aggregates the received models using a weighted average. This procedure is repeated for T communication rounds until the model converges, indicating that no further learning is occurring.

EL [9] follows a similar procedure, with the key difference being that the communication graph dynamically evolves. New neighbors are sampled at each round, which improves performance and convergence rates. We introduce EL to lay the groundwork for our approach, which also utilize random communication.

2.3 Fairness

Group fairness was introduced to quantify inequality between two groups, the *privileged* group and the *unprivileged* group [13]. As we want to quantify our algorithms' effectiveness on a setup with clusters of different sizes, the *majority* and *minority*, we will use two standard group fairness metrics: demographic parity (DP) and equalized odds (EO).

Demographic parity (DP) ensures that the two groups have the same likelihood of receiving a positive treatment, regardless of their demographic characteristics. In the following definitions, Y is the ground truth label, $\hat{Y}$ the prediction, C the set of existing labels, and S the cluster membership. $S = 0$ means that the node is part of the *privileged* group. In our case, it is the cluster with the majority of nodes. DP is formally defined as follows:

$$\mathbb{P}(\hat{Y} = y|S = 1) = \mathbb{P}(\hat{Y} = y|S = 0) \quad \forall y \in C \tag{2.3}$$

DP was initially utilized as a strict constraint. However, it is usually easier to measure fairness by considering the absolute difference between the two groups,

$$|\mathbb{P}(\hat{Y} = y|S = 1) - \mathbb{P}(\hat{Y} = y|S = 0)| \quad \forall y \in C \tag{2.4}$$

To assess the global DP over all classes y, the average is taken across all $y \in C$.

Relaxation of this metric, equalized odds (EO), was later introduced [16]. Unlike DP, it allows $\hat{Y}$ to depend on S but only through the target variable Y. This encourages the use of features directly related to Y and not to S. We directly formulate EO using the absolute difference.

$$|\mathbb{P}(\hat{Y} = y|Y = y, S = 1) - \mathbb{P}(\hat{Y} = y|Y = y, S = 0)| \quad \forall y \in C \tag{2.5}$$

A third common fairness metric, equal opportunity [16], measures a similar concept but focuses solely on the positive class ($y = 1$). This weaker formulation is essentially meaningful only for binary classification cases and, therefore, will not be used in our study.

2.4 Personalized learning

Personalizing a model involves modifying it to better fit the data. This becomes necessary when dealing with non-IID data, where personalization can maintain fairness and accuracy across various subsets [27, 36, 49].
Personalization has initially been explored in FL, with early approaches maintaining one central model and trying to find the best compromise between the nodes' heterogeneous data distributions [23, 30, 37, 52].

With these approaches, the trade-off between global consensus and local personalization is inevitable. Keeping this in mind, other techniques rely on personalization of the model at the node level, with each node maintaining a personal model, in addition to training the central one [21, 29, 39, 56]. An interesting technique features nodes sharing only the *core* of their model, while the personalized *head* is kept and learns to fit the local data. [5, 6, 15]. When the nodes' data naturally form clusters, certain FL techniques group the nodes accordingly before learning a separate model for each of them [11, 35, 47]. Alternatively, some approaches prefer an iterative method, alternating between clustering steps and training phases [15, 39].

In DL, early work on personalized learning required the communication network to reflect a notion of similarity between nodes [1, 51]. However, these approaches rely on having prior knowledge of the network's similarity matrix, which is typically not easily accessible. In DL, where privacy is a big concern, the data needed to determine the similarity between nodes can't typically be shared. Latter methods propose to adopt a dynamic network to enhance personalization, with learnable communication graphs [28, 55]. To reduce the communication cost, a method proposes to sparsify the model weights [8], learning a personalised mask for each node. Building on FedRep [6], DePRL features a shareable model core and a personalized local head for each node [53]. This approach gives the best accuracy and is the current state of the art for personalized decentralized learning for non-IID data.

However, many of these methods focus on personalization at the node level and fail to leverage similarities when dealing with clustered data. Additionally, these algorithms often lack a fairness analysis in the presence of minority groups within the data. To address these issues, we present a new decentralized personalized algorithm.

Chapter 3

The FACADE Algorithm

In this section, we first introduce our algorithm by providing an overview of its design. We then formally describe each step in detail, and conclude by discussing the importance of analyzing fairness in the presence of data with skewed feature distribution.

3.1 Design

We now present the FACADE algorithm and formalize the actions that each node takes. We aim to maintain the random communication procedure while achieving personalization without increasing communication costs. Unlike most personalized decentralized methods [8, 28, 53, 55], our algorithm maintains k models per node, one for each cluster. Specifically, each model is split into a core, common to all models, and a personalized head. This means that each node stores k versions of the same head with different parameters, as opposed to other DL methods that only keep one model in memory.

To the best of our knowledge, IFCA [15] pioneered the concept of maintaining multiple models per node to design a personalized federated learning framework. In contrast, we extend this notion to the decentralized context and reduce the memory cost by only duplicating the heads. This design choice was driven by the fact that, while the global task remains the same for all nodes, each cluster presents slight differences. Therefore, learning a common core with all available data is ideal for achieving the best data representation. The heads can then effectively capture the unique variations within each cluster. Moreover, this design allows for varying the head size for a given model, with a general rule that the greater the differences between clusters, the larger the head size should be (hence, the smaller the common core).

As with many other clustering algorithms [38, 44], the number of clusters (k) is a hyperparameter which should be estimated by the system designer beforehand. This value heavily depends on the application domain and characteristics of individual datasets. We experimentally show in Section 4.3

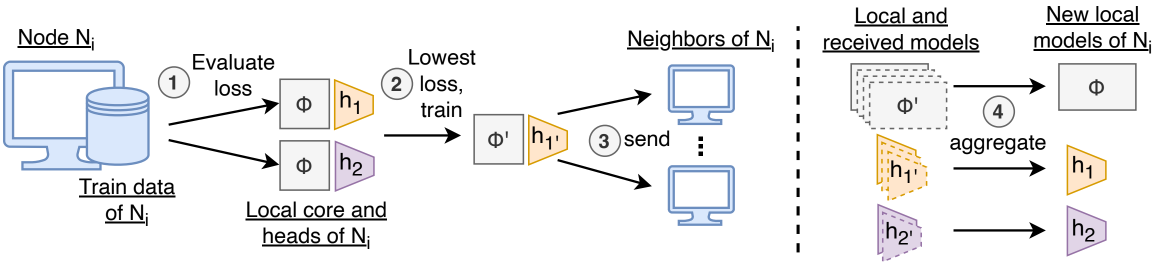


Figure 3.1: A training round in FACADE, from the perspective of node N_i. In FACADE, each node maintains a single common core and $k = 2$ different heads, indicated with different colors. N_i first evaluates the loss of each local model (step 1) and trains the model with the lowest loss (step 2). N_i then sends this updated model to its neighbors (step 3). Finally, N_i aggregates its local and received models, using different schemes for the core and the heads (step 4).

that our algorithm performs well even if this number is poorly estimated.

The full algorithm is detailed in algorithm 1 and illustrated in Figure 3.1. FACADE starts by initializing k models for each node, where each model consists of a unique head h_j and a common core ϕ. For instance, the common core might consist of three convolutional layers, and each head could be a different feed forward layer. We denote the j-th model of node N_i at the t-th communication round as $\theta_{i,j}^{(t)}$. Formally, this model is the composition of the head and the core: $\theta_{i,j}^{(t)} = h_{i,j}^{(t)} \circ \phi_i^{(t)}$. Each node then selects one of the models, trains it, and shares it with its neighbors. The choice is made by evaluating all models on the trainset and picking the one giving the smallest loss (step 1 on figure 3.1, line 3 in algorithm 1). In practice, we discovered that evaluating each model on a subset of the dataset, rather than the entire dataset, was enough to make the *right* decision, speeding up the process. This chosen model is then trained with SGD for a few steps (step 2, line 4) before being broadcast to d selected neighbors (step 3, line 6). A node also receives trained models from its neighbors and aggregates them (step 4, lines 7-11). The cores are all aggregated together, while the heads are aggregated among models with the same index, i.e. for the node N_i, $h_{i,j}^{(t+1)} = \frac{1}{C_j+1}(h_{i,j}^{(t)} + \sum_{N_p \in \mathcal{V}_i^{(t)}} h_p^{(t)} \delta_{p,j})$ for $j \in [k]$, with $\delta_{p,j}$ the Kronecker delta function that equals 1 if the node N_p sent the model $\theta_{p,j}^{(t)}$ this round, 0 otherwise. C_j represents the count of heads of index j received by all neighbors and $h_p^{(t)}$ the head sent by node N_p. We use $\mathcal{V}_i^{(t)}$ to refer to the set of the d neighbors of node N_i, that have been randomly sampled at round t.

We adopt a dynamic topology approach, similar to that used in EL. First, it has been demonstrated that altering random communication topologies leads to faster model convergence compared to traditional DL approaches that maintain a static, fixed topology throughout training [9]. Second, in the context of FACADE, dynamic topologies also prevent nodes in a cluster from becoming isolated due to initial neighbors from other clusters. By sampling random neighbors each round, an isolated node will eventually exchange models with nodes that have similar data distributions with a high probability.

Algorithm 1: Fair, Clustered and Decentralized learning (FACADE)

```
Data: $T; \tau; k; \mathcal{N}; \mathcal{L}$; learning rate $\eta$; initialize $\theta_{i,j}^{(0)} = \phi_i^{(t)} \circ h_{i,j}^{(t)}$ $\forall j \in [k]$ and $i \in [n]$; datasets $\hat{Z}_i$ $\forall i \in [n]$
      ($z \sim \mathcal{D}_j, \forall z \in \hat{Z}_i$ if $i \in S_j$); neighbors sets $\mathcal{V}_i^{(t)}$ $\forall i \in [n], \forall t \in [T]$
for $t = 0, 1, ..., T-1$ do
    for node $N_i \in \mathcal{N}$ in parallel do
        record cluster identity estimation $j_i^{(t)} \leftarrow \text{argmin}_{j \in [k]} \mathcal{L}(\theta_{i,j}^{(t)}, \hat{Z}_i)$
        $\theta_{i,j_i^{(t)}}^{(t+1/2)} \leftarrow$ LOCAL_TRAINING$(\theta_{i,j_i^{(t)}}^{(t)}, \hat{Z}_i)$
        // keep the other heads $h_{i,j}^{(t+1/2)} \leftarrow h_{i,j}^{(t)}$ $\forall j \neq j_i^{(t)}$
        for neighbor $N_p \in \mathcal{V}_i^{(t)}$ do
            Send to $N_p$ the cluster identity estimation $j_i^{(t)}$ and the trained model $\theta_{i,j_i^{(t)}}^{(t+1/2)}$
            Receive from $N_p$ its cluster identity estimation $j_p^{(t)}$ and trained model $\theta_p^{(t+1/2)} = h_p^{(t+1/2)} \circ \phi_p^{(t+1/2)}$
        update core $\phi_i^{(t+1)} \leftarrow \frac{1}{|\mathcal{V}_i^{(t)}|+1}(\phi_i^{(t+1/2)} + \sum_{N_p \in \mathcal{V}_i^{(t)}} \phi_p^{(t+1/2)})$
        for head $j = 1, ..., k$ do
            count of head $j$ received $C_j \leftarrow \sum_{N_p \in \mathcal{V}_i^{(t)}} \delta_{j, j_p^{(t)}}$
            update head $h_{i,j}^{(t+1)} \leftarrow \frac{1}{C_j+1}(h_{i,j}^{(t+1/2)} + \sum_{N_p \in \mathcal{V}_i^{(t)}} h_p^{(t+1/2)} \delta_{j, j_p^{(t)}})$
LOCAL_TRAINING$(\theta, \hat{Z}_i)$:
for $e = 0, ..., \tau - 1$ do
    (mini-batched)-stochastic gradient descent $\theta^+ \leftarrow \theta - \eta \hat{\nabla} \mathcal{L}(\theta, \hat{Z}_i)$
return $\theta^+$
```

The essence of FACADE is that no explicit clustering is required: nodes are not assigned to specific clusters. Instead, nodes that select the same model can be viewed as implicitly clustered. This allows the clustering to be dynamically evolving and nodes to detect similarities with others as the models get more accurate. What happens is that nodes simply exchange models and aggregate them according to their respective indices. If these models share the same index, the nodes likely belong to the same community, making the aggregation beneficial. If the models do not share the same index, they are not aggregated together, ensuring that the best model of the node remains unaffected by the incoming model.

3.2 FACADE and fairness

It is worth mentioning that FACADE was specifically designed to ensure fairness in networks where nodes hold non-IID data clustered across different distributions. After training, each head specializes in treating features from a specific cluster, ensuring that each node benefits from a model tailored to its unique data characteristics.

However, when dealing with other non-IID data scenarios, such as label-skewed distributions [27, 36], the concept of fairness as introduced in section 2.3 does not apply. To illustrate this, consider a label-skewed distribution with data labeled as A or B. In a cluster where most data is labeled A and only a few instances are labeled B, the model *should* perform better on the majority label A. Conversely, in another cluster where label B is more common, the model *should* focus on label B. Enforcing fairness in such scenarios would require models to perform equally well on each label *in both clusters*, which is undesirable as it would decrease the accuracy for both groups and ultimately harm the performance for every node.

FACADE is a personalized learning technique that leverages cluster differences to enhance the overall performance. It promotes fairness when diversity lies in features by identifying these differences and training personalized models. When feature distributions vary but labels are consistent, the fairness metrics from formulas 2.4 and 2.5 can be effectively applied.

3.3 Fair accuracy

While analyzing the fairness of our algorithm, we observed that the fairness metrics introduced in section 2.3 were *only* measuring if a quantity was the same across privileged and unprivileged groups. This lack of consideration for performance is problematic, as a model that performs equally poorly on both groups would still receive a favorable fairness measurement. To illustrate, a random model achieving 10% accuracy on a ten-class dataset would exhibit the highest possible fairness under these metrics, simply because there would be no performance disparity between the groups. Consequently, these fairness metrics alone are inadequate for evaluating the overall quality and effectiveness of a model.

Building on the idea that a fairness metric should also reflect the overall algorithm performance, we introduce the *fair accuracy*. This metric balances the goal of achieving high overall performance while minimizing the performance difference between groups.

$$\textsc{Acc}_{\textsc{fair}} = \frac{\alpha}{|C|}\sum_{c\in C}\textsc{Acc}_{\mathrm{c}} + (1-\alpha)(1-(\max_{c\in C}\textsc{Acc}_{\mathrm{c}} - \min_{c\in C}\textsc{Acc}_{\mathrm{c}})) \tag{3.1}$$

where C is the set of all data clusters in the network.
The goal of this metric is to ensure that the model performs well across all clusters, like standard accuracy, while penalizing disparities between the most and least accurate groups. Fair accuracy reaches its maximum value of 1 when the accuracy for both the majority and minority groups is perfect.

In this paper, we focus on an environment where two data distributions prevail in the network. Specifically, we consider the case where the nodes form a majority and a minority group concerning their data distributions. In this two-group context, the Fair Accuracy metric simplifies to:

$$\textsc{Acc}_{\textsc{fair}} = \alpha\frac{\textsc{Acc}_{\mathrm{maj}} + \textsc{Acc}_{\mathrm{min}}}{2} + (1-\alpha)(1-|\textsc{Acc}_{\mathrm{maj}} - \textsc{Acc}_{\mathrm{min}}|) \tag{3.2}$$

where ACC_{maj} and ACC_{min} represent the accuracy in the majority and minority groups, respectively. In our experiments, we used $\alpha = 2/3$, as it slightly favors well-performing models while still giving significant weight to penalizing large discrepancies. We chose not to use a more intuitive value like $\alpha = 1/2$ because we wanted to more strongly penalize models that do not leverage the cluster structure to gain insights about the data.

Chapter 4

Evaluation

We implemented FACADE in Python and we now conduct an extensive experimental evaluation of it to demonstrate its efficiency against state-of-the-art personalized DL algorithms. Our results show that FACADE not only outperforms other approaches in terms of fairness but also achieves better accuracy.
Our experiments focus on supervised image classification as a universal task setting. However, FACADE is model-agnostic, allowing it to adapt to any model choice. We implemented our algorithm and the baselines using the DecentralizePy framework [12], which supports realistic, real-time threads for each node.

To encourage reproducibility, our code is available on our GitHub repository [1].

4.1 Experimental setup

4.1.1 Datasets and model:

We conduct our experiments on Cifar-10 [25], Flickr-Mammals [19] and Imagenette [18], a subset of ten easy-to-classify classes from Imagenet [10].

To create an environment with clustered non-IID data, we chose to partition those real-world datasets into several smaller subsets following standard practice [27, 40]. The initial split of the dataset will determine the size of each cluster. As explained in section 3.2, the heterogeneity must be reflected in the feature composition of each cluster. Importantly, this split must ensure that the label distribution remains consistent across clusters, avoiding strategies like Dirichlet distribution split or k-shards method, which might alter this uniformity.

[1] `https://github.com/TicaGit/decentralizepy`

We then randomly apply different rotations to the images of each cluster [15, 33], ensuring that no two clusters share the same rotation. This approach maintains the same label distribution across clusters while introducing recognizable differences in features. In this scenario, referred to as *feature distribution skew*, the distribution of image features, $\mathbb{P}(x_i)$, varies among clusters, whereas the distribution of labels given the features, $\mathbb{P}(y_i \mid x_i)$, remains consistent across them [36], [27].
To further partition the data within the clsuter, we use a standard split, randomly allocating the same amount of data to each node. Alternative partitioning strategies can be employed if one wishes to introduce additional layers of heterogeneity at the node level. Additionally, all nodes within the same cluster share a common test set with the same rotation as their training set, ensuring consistency in evaluation.

The model we use to experiment on Cifar-10 is a slightly modified version of LeNet [26]. It has about 120k parameters, consisting of three convolution layers followed by one feed-forward layer. When training models with FACADE, we designate the last fully connected layer as the head and retain the rest of the model as the common core. We emphasize that our goal here is not to obtain the best accuracy but to investigate the effects of our algorithm on a model with the right capacity given the available data.
For Imagenette, the model remains almost the same as for Cifar-10, with minor adjustments made to accommodate the image sizes. The resulting number of parameters of the model is about 250k. Finally, as the Flickr-Mammals dataset has much more data from 41 classes, we modified ResNet8 [17] to achieve a size of about 310k parameters. For this more challenging dataset, when training with FACADE, we enlarge the head size of ResNet8 and include the last two Basic Blocks in the head, along with the final fully connected layer.

Table 4.1: Summary of datasets used to evaluate FACADE and DL baselines.

DATASET	NODES	MODEL	MODEL PARAMS.	LEARNING RATES			
				EL	DAC	DePRL	FACADE
Cifar-10 [25]	32	CNN (LeNet [26])	120k	$\eta = 0.05$	$\eta = 0.005$	$\eta = 0.01$	$\eta = 0.01$
Imagenette [18]	24	CNN (LeNet [26])	250k	$\eta = 0.001$	$\eta = 0.001$	$\eta = 0.0005$	$\eta = 0.0003$
Flickr-Mammals [19]	16	ResNet8 [17]	310k	$\eta = 0.1$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.3$

4.1.2 Baselines:

We compared against a variety of personalized decentralized learning techniques: EL [9], DePRL [53], and DAC [55]. EL, similar to D-PSGD, is one of the most simple baselines to compare against, and we chose it since FACADE also relies on communication with random nodes.
To our knowledge, DePRL is the state-of-the-art for personalized decentralized learning. It addresses the challenge of decentralized learning by allowing each node to optimize its model head locally while ensuring the overall network performance by sharing the core model through periodic com-

munication and aggregation steps. However, the setting researchers used to test this algorithm featured nodes, all having data coming from different distributions. In contrast, our approach assumes that some nodes come from the same distribution, hence forming clusters in the network. We investigated whether our method could benefit from this specific data structure.
We also chose to compare against DAC, which, as far as we know, is the most recent and best-performing approach for personalized decentralized learning on clustered non-IID data. DAC utilizes a dynamic communication topology and has been tested in environments with clusters similar to our configuration. This approach adapts the communication weights between nodes based on their data distributions, enhancing learning efficiency and performance in clustered settings.

4.1.3 Experiment settings:

To assess the fairness of our algorithm, we designed experiments with two clusters having varying proportions. Specifically, we keep the total number of nodes constant while adjusting the proportions of nodes in each cluster. This ensures that each node has the same amount of data, making the learning comparable between experiments. We demonstrate that, even when the *unprivileged* group is heavily outnumbered by the *privileged* group, FACADE maintains a high accuracy for both groups, indicating a good fairness measurement.

For Cifar-10, we consider three different configurations, each with $k = 2$ clusters and 32 nodes, with majority-to-minority ratios of 16:16, 24:8, and 30:2. For instance, in the experiment with a ratio of 24:8, 24 nodes have 3/4 of the dataset Cifar-10, while the remaining 8 nodes share the rest of the dataset, but rotated 180°. In all experiments, cluster 0 is the *privileged* group (the cluster with the most nodes). To produce each results, we run a total of $T = 1200$ communication rounds, averaged over four seeds. Each local training round features $\tau = 10$ local steps of batch size $B = 8$. For each baseline, we use an SGD optimizer, and the learning rate was independently fine-tuned with a grid search. Table 4.1 summarizes the parameters used for each dataset and baselines. Additionally, to enhance performance whenever feasible, we implemented a final all-reduce step [43], where all nodes share their models and perform a final aggregation.
To evaluate the algorithms, we measure the accuracy on the test set every 80 rounds. We also record the final performances of the algorithm on the entire test set and compute the fairness of the models with equations 2.4 and 2.5.

For Imagenette, we utilize a similar setup with rotations, featuring 24 nodes and ratios of 12:12, 18:6, and 20:4. Again, we perform a grid search for each algorithm to tune the learning rate and run 800 communications steps. As the Flickr-Mammals dataset is much bigger, we decrease the number of nodes to 16, and only consider two setups, 8:8 and 14:2. Resnet8 being a more complex model, we increase the local steps to $\tau = 40$ and let the training run for 1200 communication rounds.

4.2 Results

We now discuss the results of our experiments on Cifar-10. We tested all the baselines in the two-cluster configurations detailed above. Our goal is to analyze how FACADE behaves in an environment where the minority group is increasingly outnumbered. Complementary plots for Cifar-10 and all results for Imagenette and Flickr-Mammals are provided in the Appendix. We also detailed any differences FACADE encountered with these other datasets.

4.2.1 Accuracy:

In Figure 4.1, the accuracy of the model is reported for all configurations and is detailed for each cluster, the majority on the left and the minority on the right. We can see that FACADE outperforms other decentralized methods. When considering the minority cluster (right), we note that FACADE is much better than other methods at giving the *unprivileged* groups a good treatment.
When the number of nodes in each cluster is balanced (top row), FACADE shows a slight performance advantage thanks to its multiple heads, which provide greater capacity to adapt to variations within each cluster (ACC≈70%). However, other methods also perform reasonably well, with accuracies around 63%, as there is enough data to find a good compromise model that suits both clusters. In the scenario in which one cluster massively outnumbers the other (bottom row), we observe that the performance of most methods is similar for the majority cluster(ACC≈72%). This is due to consensus-based methods like EL, being influenced by the data distribution of the dominant group. However, our algorithm excels when examining the minority (down-right). Specifically, FACADE outperforms DePRL by approximately 7 percentage points and EL by 20 percentage points, demonstrating greater fairness to the minority group. This improvement is due to a head being exclusively used by the minority group. This allows the head to remain unaffected by the majority's data distribution and to adapt specifically to the data distribution of the unprivileged group. The difference in accuracy of FACADE between the majority and the minority is solely due to the majority group having more nodes, leading to more training data and better generalization.

Similar plots for Imagenette and Flickr-Mammals are provided in the Appendix. Overall, we observe similar trends, with FACADE outperforming baselines, and shining especially for the minority.

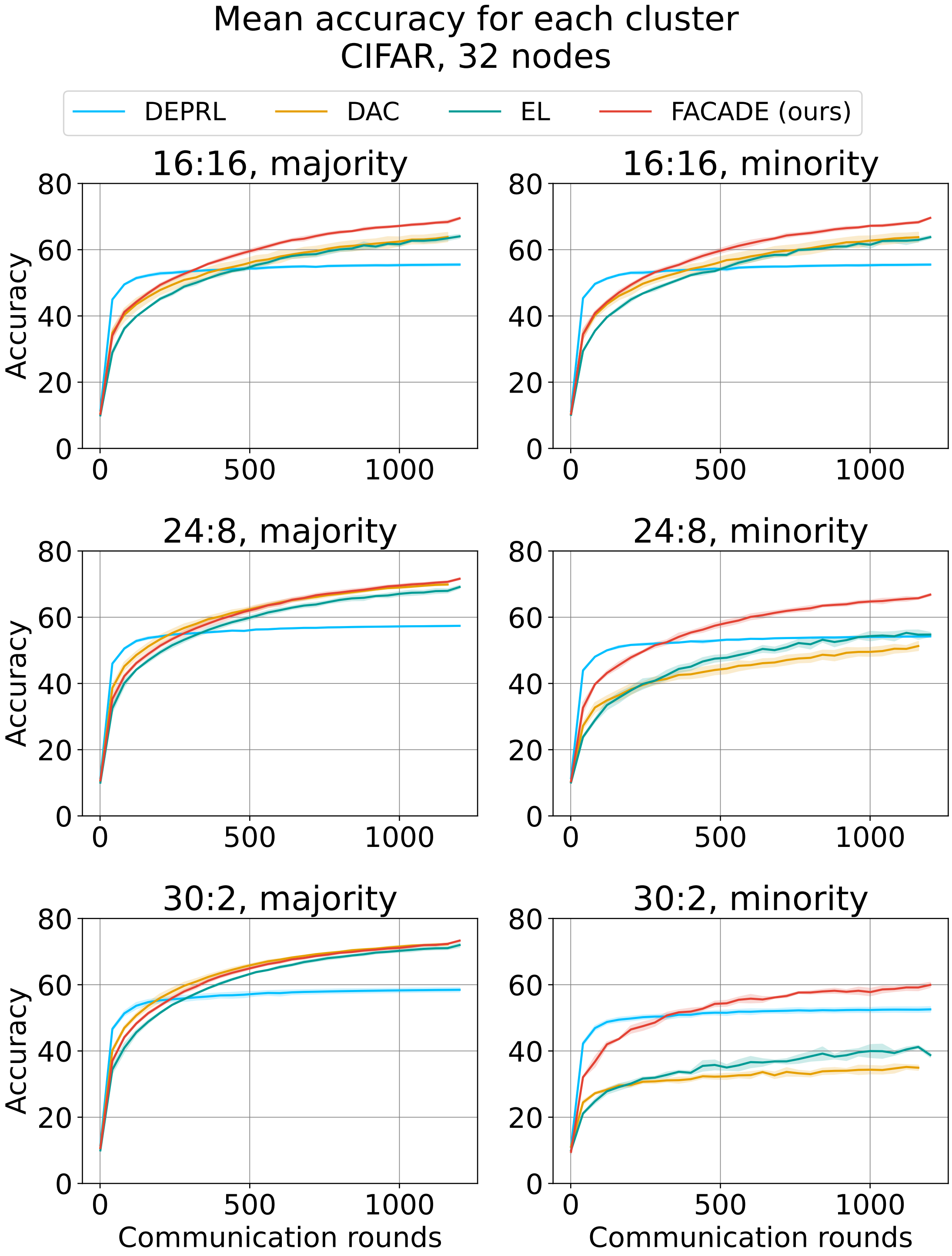


Figure 4.1: Average accuracy (↑ is better) for the nodes in the majority cluster (left) and those in the minority (right) obtained on CIFAR-10. Each row represents the results of one experiment.

Table 4.2: Performance comparison of all algorithm on Cifar-10. The metrics evaluated are, in this order: the averaged accuracy of all nodes in the majority group, the minority group, and across the entire network; followed by demographic parity, equalized odds, and finally, fair accuracy.

CONFIG	ALGORITHM	ACC$_{\text{MAJ}}$ ↑	ACC$_{\text{MIN}}$ ↑	ACC$_{\text{ALL}}$ ↑	DEMO. PAR. ↓	EQU. ODDS ↓	ACC$_{\text{FAIR}}$ ↑
16:16	EL	64.03±0.54	63.80±0.42	63.91±0.43	0.0032±0.0007	0.0204±0.0038	75.87
	DAC	63.82±1.34	63.79±1.43	63.81±0.31	0.0065±0.0008	0.0402±0.0057	75.86
	DePRL	55.51±0.34	55.50±0.29	55.50±0.26	**0.0020±0.0004**	**0.0099±0.0026**	70.33
	FACADE	**69.50±0.32**	**69.61±0.20**	**69.55±0.25**	0.0060±0.0015	0.0267±0.0079	**79.67**
24:8	EL	69.13±0.45	54.76±0.59	65.53±0.31	0.0103±0.0012	0.1596±0.0097	69.84
	DAC	69.88±0.28	51.30±1.31	65.24±0.19	0.0131±0.0015	0.2067±0.0175	67.53
	DePRL	57.40±0.13	54.21±0.24	56.60±0.15	**0.0030±0.0003**	**0.0361±0.0013**	69.48
	FACADE	**71.61±0.27**	**66.81±0.34**	**70.41±0.29**	0.0079±0.0025	0.0582±0.0033	**77.87**
30:2	EL	71.99±0.70	38.77±0.62	69.91±0.67	0.0306±0.0014	0.3693±0.0081	59.18
	DAC	72.21±0.24	34.94±0.72	69.88±0.25	0.0340±0.0027	0.4143±0.0075	56.63
	DePRL	58.47±0.59	52.56±0.78	58.10±0.57	**0.0047±0.0013**	**0.0684±0.0072**	68.37
	FACADE	**73.32±0.15**	**59.96±0.72**	**72.48±0.19**	0.0086±0.0026	0.1491±0.0059	**73.31**

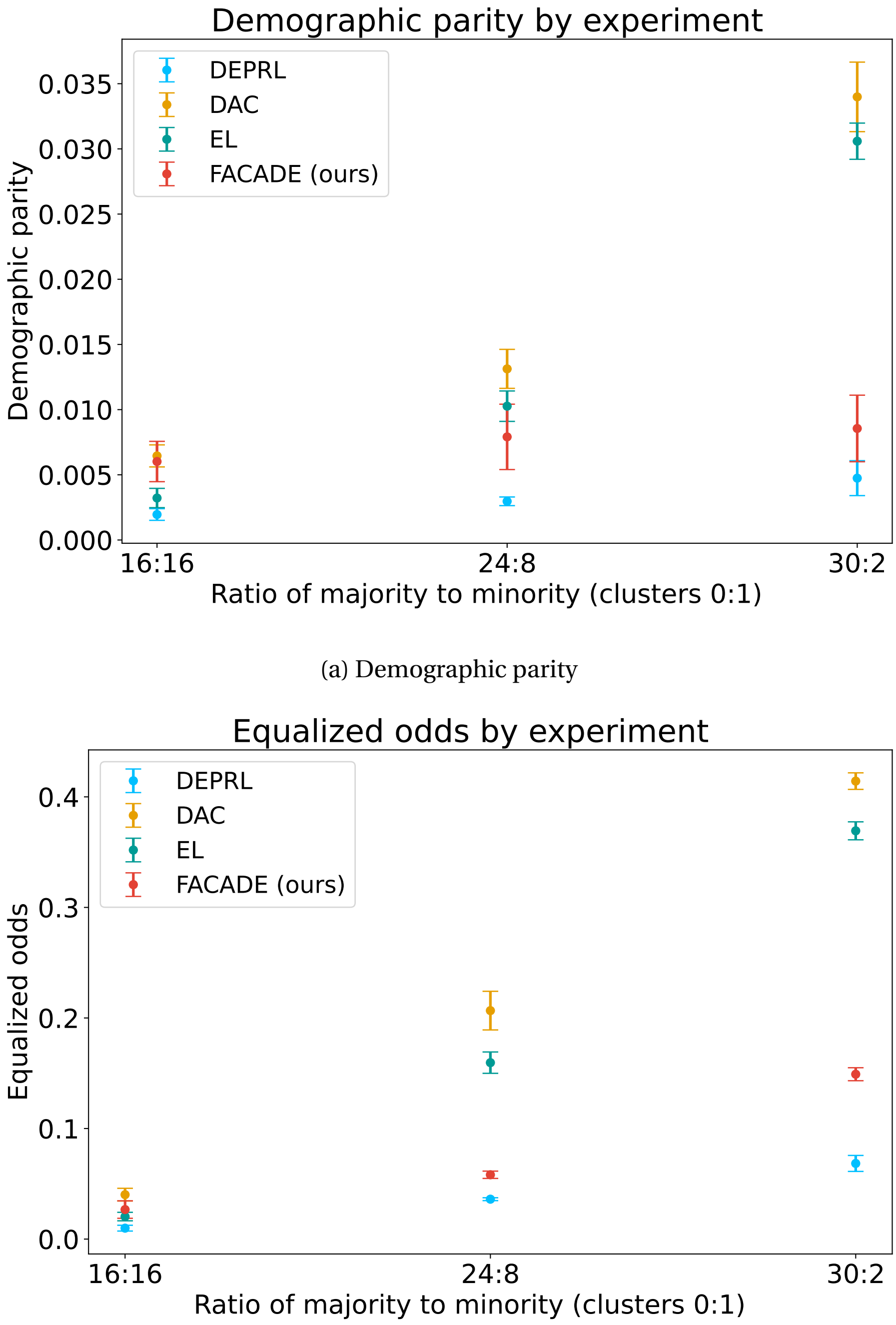
4.2.2 Fairness:

The fairness of FACADE becomes even clearer when considering the analysis presented in figure 4.2. We measure the demographic parity 2.4 and the equalized odds 2.5 on the final model of each experiment. Our algorithm significantly outperforms others in ensuring fairness between the two groups.

The only baseline that seems to outperform FACADE is DePRL, which exhibits lower demographic parity and equalized odds. However, the accuracy for all nodes, regardless of cluster membership, is quite poor (Figure 4.1), which is misleadingly defined as having good fairness. We noticed that the head of the model in DePRL overfits significantly because it is never shared with other nodes. Consequently, the algorithm cannot leverage the similar data distribution of other nodes and struggles to generalize on the test set. This results in DePRL having similar accuracy across all nodes, regardless of whether they belong to the minority or majority group. As this uniformity aligns with what the two fairness metrics measure, DePRL appears to have the best results in Figure 4.2 and is supposedly the fairest algorithm.

However, this uniformity is not truly beneficial. As seen in the accuracy plot, FACADE outperforms DePRL for both the majority and minority groups. The behavior of DePRL is undesirable because achieving low global accuracy without differences between groups is not advantageous.

This example perfectly justifies the introduction of the *Fair Accuracy*, detailed in section 3.3. Under this new metric, both model performance and fairness are considered, penalizing models that disproportionately benefit the privileged group. The table 4.2 summarizes all results obtained for CIFAR-10 and evaluates the fair accuracy of all algorithms in the last column. These results demonstrate that our method achieves the highest Fair Accuracy, indicating superior performance and fairness across both majority and minority groups.



(a) Demographic parity

(b) Equalized odds

Figure 4.2: Demographic parity (↓ is better) and equalized odds (↓ is better) obtained for on Cifar-10. FACADE is fairer than all the baselines except DePRL.

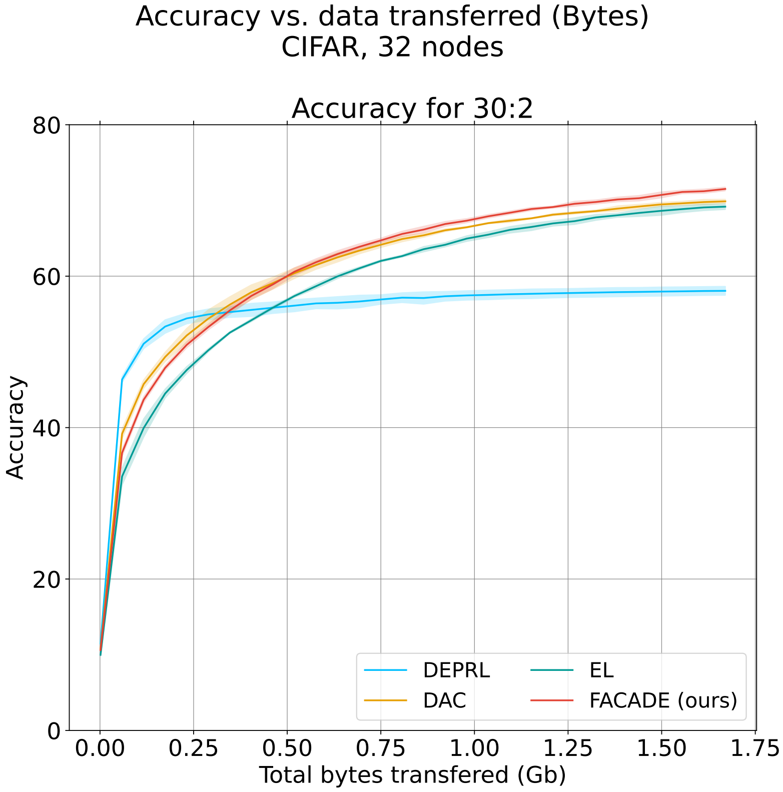


Figure 4.3: Accuracy vs. total data transfered (↑ is better) obtained on CIFAR-10.

We provide two similar tables in the Appendix. In these, we again observe that the model head of DePRL tends to overfit, resulting in good standard fairness measurements. This effect is especially pronounced for Imagenette, given its smaller training set compared to the other datasets. However, when our new metric is applied, which accounts for global performance, DePRL performs worse than all other algorithms.

4.2.3 Communication cost:

As mentioned earlier, the communication cost of FACADE is almost the same as EL or D-PSGD. Each node still sends only one model, with the addition of the model index for our approach. Figure 4.3 illustrates the global accuracy of each method relative to the total data transferred by each node to any neighbors. The accuracy is averaged across all nodes, regardless of which cluster they belong to. We took as an example the configuration 30:2, but all of them have the same communication cost. We observe that all four methods have similar communication costs, but FACADE achieves the highest accuracy per byte transferred.

Table 4.3: Performances of Facade with an incorrect number of heads as a hyperparameter. There were three clusters, with 20, 10, and 2 nodes. The average accuracies achieved by all nodes within each cluster are reported.

Models	**Acc$_{20}$ ↑**	**Acc$_{10}$ ↑**	**Acc$_{2}$ ↑**	**Acc$_{\text{FAIR}}$↑**
EL	66.13	58.50	39.54	60.95
1 head	68.02	59.41	40.75	61.62
2 heads	**69.60**	65.04	46.11	65.67
3 heads	**69.13**	**66.45**	**58.90**	**73.14**
4 heads	**69.53**	66.09	58.08	72.56
5 heads	68.48	65.52	**58.80**	**72.95**

4.3 Sensitivity to hyperparameters

To evaluate the sensitivity of Facade to variations in the number of cluster hyperparameters, we conduct a detailed experiment. Building on our previous setup with CIFAR-10, we create three clusters with 20, 10, and 2 nodes. Each cluster has images rotated 0°, 90°, and 180°, respectively. This setup also demonstrates the capacity of Facade to handle more than two clusters.
We intentionally vary the number of heads used by Facade from one to five, with the ideal number being three, which corresponds to the actual number of clusters. The results are presented in Table 4.3, where we report the accuracy of each cluster Acc$_{20}$, Acc$_{10}$ and Acc$_{2}$.

When only one head is used, Facade essentially replicated the behavior of EL, delivering similar performance. With two heads, the two smallest clusters have to *share* a head, while the largest cluster has a head specialized for its data distribution. Three heads demonstrate the best performance, as they match the number of clusters. However, even with four and five heads, the results are remarkably close to those obtained with the optimal number of heads. The dynamics reveal that multiple heads tend to specialize in the same cluster, often the largest one. In other clusters, each has at least one model head specialized in their specific data distribution. The only drawback is that nodes *waste* training rounds by selecting different heads, resulting in slightly lower accuracy compared to consistently training with the same head.

This experiment highlights the robustness of our algorithm to variations in the hyperparameter k. The system as a whole maintains a performance level close to the optimum, showcasing its resilience.

4.4 Settlement analysis

In this section, we provide insights into the concept of *settlement* in our algorithm. In Facade, a scenario can arise where one or more heads consistently outperform the others across all nodes.

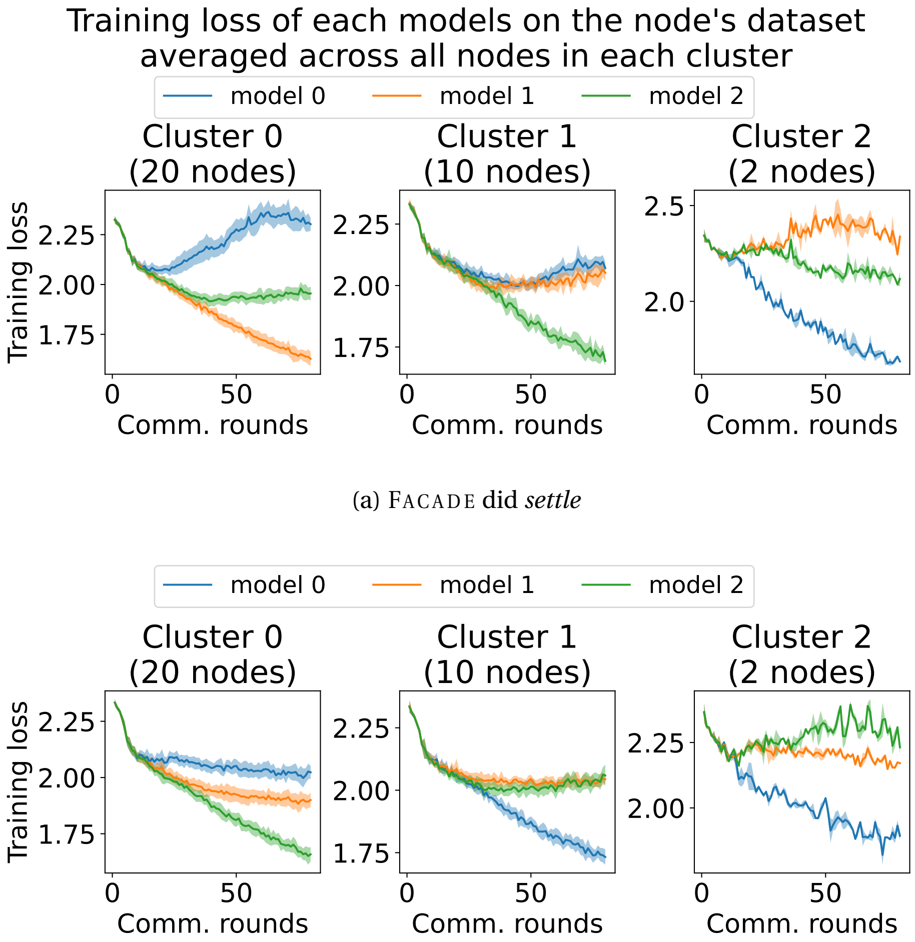
When this happens, only the superior heads are selected, causing the other heads never to be trained again. This situation is referred to as the algorithm not *settling*. Reusing the previous setup, figure 4.4 illustrates this behavior, with an example where FACADE settled and another where it did not.

The risk of not settling is higher when there is a large majority-to-minority ratio among the nodes with different data distributions. When the algorithm does not settle, it cannot fully exploit its potential. However, it is important to note that not settling is not a catastrophic issue. In such cases, performance merely drops to the level of EL, and a simple change of seeds is usually enough to achieve settlement.

To mitigate the risk of not settling, we employ several strategies. First, with careful selection of model hyperparameters, we can reduce the likelihood of this occurrence. Another effective strategy is to initiate the training with a few rounds of EL, where all heads share the same weights before transitioning to independent parameters for each head. This initial shared training phase is particularly crucial during the early stages of the algorithm when the models are still largely predicting randomly. During this phase, one head can easily capture a better data representation and quickly outperform the others. By beginning with shared training, the core and heads develop a solid data representation foundation. When the heads eventually train independently, it becomes easier for each to specialize in a specific cluster's data distribution, thus stabilizing the training process.

Although we are no longer using it, we also experimented with an exploration technique. Instead of always choosing the model with the smallest loss, we allowed each node a small probability of selecting and training a random model. This exploration phase ensured that all models had opportunities to learn and improve, reducing the chances of one model prematurely dominating the training process.

These techniques proved effective in stabilizing the training process and enhancing the overall performance of FACADE. By incorporating initial shared training, we significantly reduced the likelihood of the algorithm failing to settle, thereby maximizing its potential.



(a) FACADE did *settle*

(b) FACADE did not *settle*

Figure 4.4: Training loss evolution (↓ is better) of the three models, averaged across nodes within the same cluster (unknown to the algorithm). The top plot illustrates a case where FACADE successfully *settled*, meaning all nodes within the same cluster favor the same model, and no nodes of another cluster picked it. The bottom plot shows a case where the algorithm did not *settle*, resulting in all nodes from clusters 1 and 2 selecting and training the same model (model 0). We observe that model 1 is not selected at iteration 80, indicating it will no longer be chosen, as it will not be trained and thus will not improve on any distribution.

Chapter 5

Discussion

In this section, I will explain the work conducted during my master's thesis that was not presented in the other sections of the paper. Throughout the project, I maintained a Google Slide[1] as a progress journal to keep track of my advancements.

5.1 IDCA, the first version of FACADE

Initially, we designed FACADE without the core-head split. This first version was called IDCA, and each node maintained k fully independent models instead of only duplicating the heads. However, the rest of the algorithm, including the model selection step, sharing, and aggregation, remained as presented in Algorithm 1. The old version of the algorithm could be replicated by setting the head to be the total model and the core to be empty. Our inspiration was an algorithm called IFCA [15], which essentially implements this idea of model duplication in the centralized federated setting.

5.1.1 MNIST dataset:

The first experiments with this algorithm were conducted on the MNIST dataset, which is why it is implemented in the code. We quickly discovered that the task was too easy and decided to use Cifar-10 as the baseline dataset instead. Consequently, the compatibility of the code with MNIST was not maintained and is not up to date.

[1] https://docs.google.com/presentation/d/1xRalq0L-uQjW7EbqG59NDVKHcUpyaf8JuvAj8CsxSBg/edit?usp=sharing

5.1.2 Settlement plots:

One of our goals was to analyze the *settlement* of FACADE, as described in section 4.4, and we produced plots to quickly detect if the algorithm did settle or not. The plots in 5.1 use the same setup presented for the settlement analysis 4.4. These plots track the head chosen by each node in each cluster over time and assess the final head selection to determine if FACADE has settled. In the upper plot, we tracked the number of nodes within the same cluster that select each different head. After a *fuzzy* phase, we observe that all nodes in the same cluster converge to the same head. This is confirmed by the lower plot, which counts the final head of each index chosen by all the nodes within a cluster. These plots were generated using our latest version of FACADE with the core-head split. However, the initial version of our algorithm operated similarly, it just utilized full models instead of heads.

5.1.3 Privacy:

Next, we spent a few weeks focusing on privacy. We implemented membership inference attacks (MIA) [54] on FACADE and used Muffliato as a defense [7]. However, we quickly abandoned this line of work, not because it wasn't interesting, but because we wanted to concentrate on the fairness aspect of our research.

5.2 Enhancing FACADE with various methods

Having a clear direction in mind, we then focused on improving FACADE. Before implementing the head-core split, we explored several ideas to enhance the performance of our algorithm.

5.2.1 Model leaking and exploration:

Inspired by the relationship between soft k-means [3] and k-means [38], we developed a *softer* assignment of models to nodes. The idea was that if training losses at the assignment step were similar, indicating that no model was a perfect fit for the node's data, a mix of multiple models could be aggregated and used for training. We named this the *model leaking* strategy, but it did not improve the results.

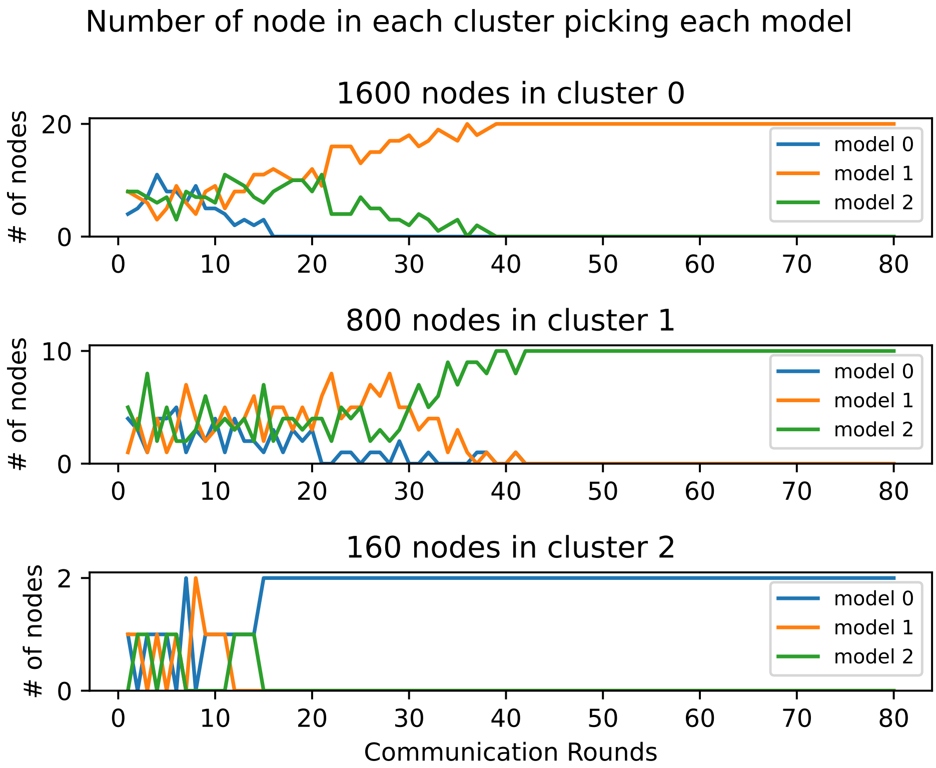
We also explored the idea of enhancing performance through exploration (Section 4.4). By allowing nodes to pick their model randomly, we hoped to expose models to data from other clusters, thus promoting better generalization. However, this method consistently worsened results because it prevented the models from specializing in a cluster's distribution.

5.2.2 Custom losses:

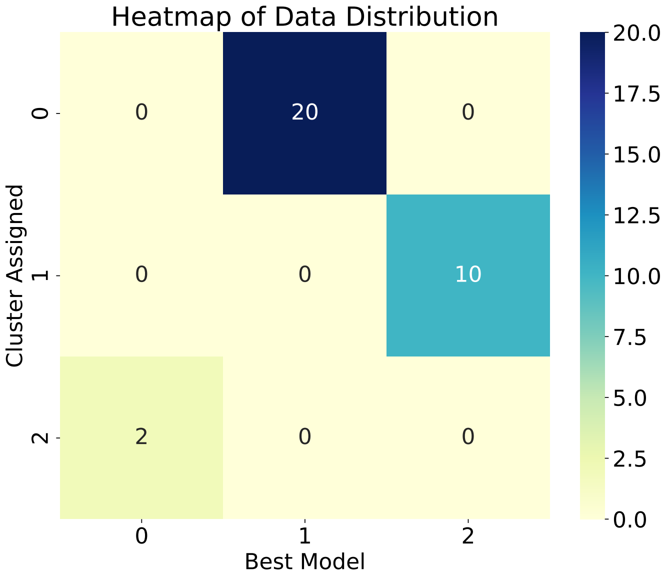
Later, still without the core-head split, we attempted to use custom loss functions to encourage fairness and enhance results. The idea was to add a regularization term that would minimize the difference between two quantities that depend on all models, thereby indirectly influenced by all data distributions. We tried this approach with demographic parity, equalized odds, accuracy, and loss computed on the training set. We also designed a loss that penalized models that were too different. Initially, we computed the difference between the parameters of the entire models, and then we focused on a subset of the weights, specifically those now part of the core. A final attempt involved penalizing the difference between the feature representations of the data after passing through all models, aiming to make them learn similar data representations.

5.2.3 Final algorithm:

After these attempts, we realized that none of these techniques were more effective than the core-head split, a technique already used in IFCA to improve the settlement. We decided to make this technique the heart of our algorithm, as it consistently improved results over simpler model duplication. The final version of the algorithm now consists of a single core and multiple heads. It is the one presented in the other sections of this paper and tested during our experiments.



(a) Model choice evolution



(b) Final model distribution

Figure 5.1: (up) Evolution of the number of models (heads) chosen per each node within the same cluster. (down) Final model (head) choice (x-axis) for each node in each cluster (y-axis). On both plots, we can assess that FACADE converge.

Chapter 6

Conclusion

In this paper, we addressed the clustered non-IID data issue by presenting a decentralized algorithm designed for secure and scalable environments. FACADE features the innovative idea of maintaining multiple model heads in each node, enabling the learning of personalized models for each cluster. This decentralized process ensures fairness, allowing even minority groups to achieve high accuracy. The design of FACADE keeps the communication cost comparable to simpler algorithms like D-PSGD or EL, effectively ensuring fairness without additional communication overhead.

Additionally, we introduced a new metric that balances fairness and performance, providing a comprehensive evaluation of our algorithm's effectiveness. We experimentally verified our approach, demonstrating its stability and superior performance compared to baseline methods in the clustered non-IID data setting.

In conclusion, FACADE offers a robust solution to the clustered non-IID data problem, combining scalability, decentralization, and fairness. Our experimental results highlight the potential of FACADE to significantly improve the performance of decentralized learning systems, making it a valuable contribution to the field.

Bibliography

[1] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. “Personalized and private peer-to-peer machine learning”. In: *International conference on artificial intelligence and statistics*. PMLR. 2018, pp. 473–481.

[2] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. “Fairness in criminal justice risk assessments: The state of the art”. In: *Sociological Methods & Research* 50.1 (2021), pp. 3–44.

[3] Christopher M Bishop. “Pattern recognition and machine learning”. In: *Springer google schola* 2 (2006), pp. 1122–1128.

[4] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. “Leaf: A benchmark for federated settings”. In: *arXiv preprint arXiv:1812.01097* (2018).

[5] Rich Caruana. “Multitask learning”. In: *Machine learning* 28 (1997), pp. 41–75.

[6] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. “Exploiting shared representations for personalized federated learning”. In: *International conference on machine learning*. PMLR. 2021, pp. 2089–2099.

[7] Edwige Cyffers, Mathieu Even, Aurélien Bellet, and Laurent Massoulié. “Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 15889–15902.

[8] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. “Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training”. In: *International conference on machine learning*. PMLR. 2022, pp. 4587–4604.

[9] Martijn De Vos, Sadegh Farhadkhani, Rachid Guerraoui, Anne-Marie Kermarrec, Rafael Pires, and Rishi Sharma. “Epidemic Learning: Boosting Decentralized Learning with Randomized Communication”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[11] Don Kurian Dennis, Tian Li, and Virginia Smith. “Heterogeneity for the win: One-shot federated clustering”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2611–2620.

[12] Akash Dhasade, Anne-Marie Kermarrec, Rafael Pires, Rishi Sharma, and Milos Vujasinovic. “Decentralized learning made easy with DecentralizePy”. In: *Proceedings of the 3rd Workshop on Machine Learning and Systems*. 2023, pp. 34–41.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. “Fairness through awareness”. In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[14] Fabian Gerz, Tolga Renan Bastürk, Julian Kirchhoff, Joachim Denker, Loui Al-Shrouf, and Mohieddine Jelali. “A comparative study and a new industrial platform for decentralized anomaly detection using machine learning algorithms”. In: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2022, pp. 1–8.

[15] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. “An efficient framework for clustered federated learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 19586–19597.

[16] Moritz Hardt, Eric Price, and Nati Srebro. “Equality of opportunity in supervised learning”. In: *Advances in neural information processing systems* 29 (2016).

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[18] Jeremy Howard. *ImageNette*. URL: `https://github.com/fastai/imagenette/`.

[19] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. “The non-iid data quagmire of decentralized machine learning”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 4387–4398.

[20] Li Huang, Andrew L Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. “Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records”. In: *Journal of biomedical informatics* 99 (2019), p. 103291.

[21] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. “Personalized cross-silo federated learning on non-iid data”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 9. 2021, pp. 7865–7873.

[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. “Advances and open problems in federated learning”. In: *Foundations and trends® in machine learning* 14.1–2 (2021), pp. 1–210.

[23] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. “Scaffold: Stochastic controlled averaging for federated learning”. In: *International conference on machine learning*. PMLR. 2020, pp. 5132–5143.

[24] Harsh Kasyap and Somanath Tripathy. “Privacy-preserving decentralized learning framework for healthcare system”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 17.2s (2021), pp. 1–24.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[27] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. “Federated learning on non-iid data silos: An experimental study”. In: *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE. 2022, pp. 965–978.

[28] Shuangtong Li, Tianyi Zhou, Xinmei Tian, and Dacheng Tao. “Learning to collaborate in decentralized learning of personalized models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9766–9775.

[29] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. “Ditto: Fair and robust federated learning through personalization”. In: *International conference on machine learning*. PMLR. 2021, pp. 6357–6368.

[30] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. “Federated optimization in heterogeneous networks”. In: *Proceedings of Machine learning and systems* 2 (2020), pp. 429–450.

[31] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. “Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent”. In: *Advances in neural information processing systems* 30 (2017).

[32] Zhuotao Lian and Chunhua Su. “Decentralized federated learning for Internet of Things anomaly detection”. In: *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 2022, pp. 1249–1251.

[33] David Lopez-Paz and Marc’Aurelio Ranzato. “Gradient episodic memory for continual learning”. In: *Advances in neural information processing systems* 30 (2017).

[34] Songtao Lu, Yawen Zhang, and Yunlong Wang. “Decentralized federated learning for electronic health records”. In: *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2020, pp. 1–5.

[35] Yibo Luo, Xuefeng Liu, and Jianwei Xiu. “Energy-efficient clustering to address data heterogeneity in federated learning”. In: *ICC 2021-IEEE International Conference on Communications*. IEEE. 2021, pp. 1–6.

[36] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. "A state-of-the-art survey on solving non-IID data in Federated Learning". In: *Future Generation Computer Systems* 135 (2022), pp. 244–258.

[37] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. "Layer-Wised Model Aggregation for Personalized Federated Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10092–10101.

[38] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[39] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. "Three approaches for personalization with applications to federated learning". In: *arXiv preprint arXiv:2002.10619* (2020).

[40] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[41] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. "Federated learning for smart healthcare: A survey". In: *ACM Computing Surveys (Csur)* 55.3 (2022), pp. 1–37.

[42] Róbert Ormándi, István Hegedűs, and Márk Jelasity. "Gossip learning with linear models on fully distributed data". In: *Concurrency and Computation: Practice and Experience* 25.4 (2013), pp. 556–571. DOI: 10.1002/cpe.2858.

[43] Pitch Patarasuk and Xin Yuan. "Bandwidth optimal all-reduce algorithms for clusters of workstations". In: *Journal of Parallel and Distributed Computing* 69.2 (2009), pp. 117–124.

[44] Douglas A Reynolds et al. "Gaussian mixture models." In: *Encyclopedia of biometrics* 741.659-663 (2009).

[45] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. "The future of digital health with federated learning". In: *NPJ digital medicine* 3.1 (2020), pp. 1–7.

[46] Badrul M Sarwar, George Karypis, Joseph Konstan, and John Riedl. "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering". In: *Proceedings of the fifth international conference on computer and information technology*. Vol. 1. 2002, pp. 291–324.

[47] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints". In: *IEEE transactions on neural networks and learning systems* 32.8 (2020), pp. 3710–3722.

[48] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. "Robust and communication-efficient federated learning from non-iid data". In: *IEEE transactions on neural networks and learning systems* 31.9 (2019), pp. 3400–3413.

[49] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. "Towards personalized federated learning". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[50] Bernardo Camajori Tedeschini, Stefano Savazzi, Roman Stoklasa, Luca Barbieri, Ioannis Stathopoulos, Monica Nicoli, and Luigi Serio. "Decentralized federated learning for healthcare networks: A case study on tumor segmentation". In: *IEEE access* 10 (2022), pp. 8693–8708.

[51] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. "Decentralized collaborative learning of personalized models over networks". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 509–517.

[52] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. "Tackling the objective inconsistency problem in heterogeneous federated optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 7611–7623.

[53] Guojun Xiong, Gang Yan, Shiqiang Wang, and Jian Li. "DePRL: Achieving Linear Convergence Speedup in Personalized Decentralized Learning with Shared Representations". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 14. 2024, pp. 16103–16111.

[54] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. "Privacy risk in machine learning: Analyzing the connection to overfitting". In: *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE. 2018, pp. 268–282.

[55] Edvin Listo Zec, Ebba Ekblom, Martin Willbo, Olof Mogren, and Sarunas Girdzijauskas. "Decentralized adaptive clustering of deep nets is beneficial for client collaboration". In: *International Workshop on Trustworthy Federated Learning*. Springer. 2022, pp. 59–71.

[56] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. "Personalized federated learning with first order model optimization". In: *arXiv preprint arXiv:2012.08565* (2020).

[57] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. "Federated learning with non-iid data". In: *arXiv preprint arXiv:1806.00582* (2018).

Appendix A

Appendix: Supplementary material

In this section, we provide an additional plot for Cifar-10 and all plots and tables for the datasets Imagenette and Flickr-Mammals

A.1 Supplementary plots for Cifar-10

We provide the plot showing the evolution of the Fair Accuracy for all the algorithms on Cifar-10 A.1. It allows to compare the convergence of FACADE and the baselines.

A.2 Imagenette dataset

We present here the results and insight obtained with the Imagenette dataset [18]. Figure A.2 presents the accuracy separated for the majority (right) and minority (left). FACADE again outperforms all other baselines. We also provide the plots for the demographic parity A.3a, equalized odds A.3b and fair accuracy A.4. The problem of DePRL overfitting is even more important, as the trainset of Imagenette is much smaller than CIFAR's. This again leads to misleadingly good fairness metrics, which supports the need for the *fair accuracy*. All results obtained for Imagenette are summarized in Table A.1

A.3 Flickr-Mammals dataset

We also reported in A.2 the results obtained on the Flickr-Mammals dataset. As the dataset contained much more images we let the training run for *only* 1200 communication rounds, with 40 local steps. After this time, all algorithms could still improve, as the training loss did not reach a plateau. However, it is clear from the results that FACADE again outperforms the baselines. We did not include the plots, as the table should be sufficient to judge the performance of the algorithms.

Table A.1: Performance comparison of all algorithm on Imagenette. The metrics evaluated are, in this order: the averaged accuracy of all nodes in the majority group, the minority group, and across the entire network; followed by demographic parity, equalized odds, and finally, fair accuracy.

Config	Algorithm	Acc$_{\text{MAJ}}$ ↑	Acc$_{\text{MIN}}$ ↑	Acc$_{\text{ALL}}$ ↑	Demo. Par. ↓	Equ. Odds ↓	Acc$_{\text{FAIR}}$ ↑
12:12	EL	66.43±0.56	66.85±0.67	66.64±0.59	0.0033±0.0008	0.0208±0.0035	77.62
	DAC	65.73±0.73	64.45±0.54	65.09±0.47	**0.0054±0.0004**	**0.0286±0.0036**	76.30
	DePRL	43.14±1.00	43.49±1.20	43.31±1.07	0.0078±0.0016	0.0319±0.0039	62.10
	FACADE	**68.18±0.35**	**68.59±0.34**	**68.39±0.27**	**0.0050±0.0010**	**0.0239±0.0035**	**78.78**
16:8	EL	**69.69±0.27**	60.21±0.44	67.32±0.29	0.0121±0.0008	0.1061±0.0050	73.48
	DAC	68.55±0.62	56.92±1.32	65.64±0.57	0.0136±0.0019	0.1299±0.0176	71.28
	DePRL	43.40±0.79	43.09±1.22	43.33±0.89	0.0096±0.0025	**0.0361±0.0060**	62.06
	FACADE	**69.61±0.37**	**66.44±0.19**	**68.82±0.30**	**0.0054±0.0011**	**0.0397±0.0014**	**77.63**
20:4	EL	**70.17±0.28**	56.06±0.56	67.81±0.33	0.0186±0.0006	0.1566±0.0026	70.71
	DAC	69.05±0.74	50.93±1.65	66.03±0.45	0.0226±0.0039	0.2009±0.0249	67.29
	DePRL	43.67±1.01	42.64±1.16	43.50±1.02	**0.0090±0.0024**	**0.0388±0.0078**	61.76
	FACADE	**69.61±0.46**	**64.15±0.39**	**68.70±0.42**	**0.0064±0.0015**	0.0630±0.0057	**76.10**

Table A.2: Performance comparison of all algorithm on Flickr-Mammals. The metrics evaluated are, in this order: the averaged accuracy of all nodes in the majority group, the minority group, and across the entire network; followed by demographic parity, equalized odds, and finally, fair accuracy.

Config	Algorithm	Acc$_{\text{MAJ}}$ ↑	Acc$_{\text{MIN}}$ ↑	Acc$_{\text{ALL}}$ ↑	Demo. Par. ↓	Equ. Odds ↓	Acc$_{\text{FAIR}}$ ↑
8:8	EL	59.97±0.23	59.92±0.22	59.94±0.19	**0.0006±0.0001**	**0.0094±0.0014**	73.28
	DAC	60.56±0.60	59.94±0.32	60.25±0.33	0.0016±0.0002	0.0203±0.0012	73.29
	DePRL	44.92±0.61	45.61±1.23	45.26±0.85	0.0047±0.0004	0.0373±0.0039	63.28
	FACADE	**65.50±0.55**	**64.92±0.41**	**65.21±0.47**	0.0035±0.0001	0.0467±0.0018	**76.62**
14:2	EL	64.92±0.21	49.71±0.20	63.02±0.17	**0.0057±0.0002**	0.1349±0.0034	66.47
	DAC	66.11±0.46	46.70±4.75	63.68±0.63	0.0067±0.0010	0.1836±0.0409	64.47
	DePRL	45.69±0.82	45.91±0.79	45.72±0.81	0.0072±0.0005	**0.0644±0.0027**	63.80
	FACADE	**67.63±0.49**	**59.55±1.06**	**66.62±0.56**	**0.0058±0.0004**	0.1077±0.0109	**73.03**

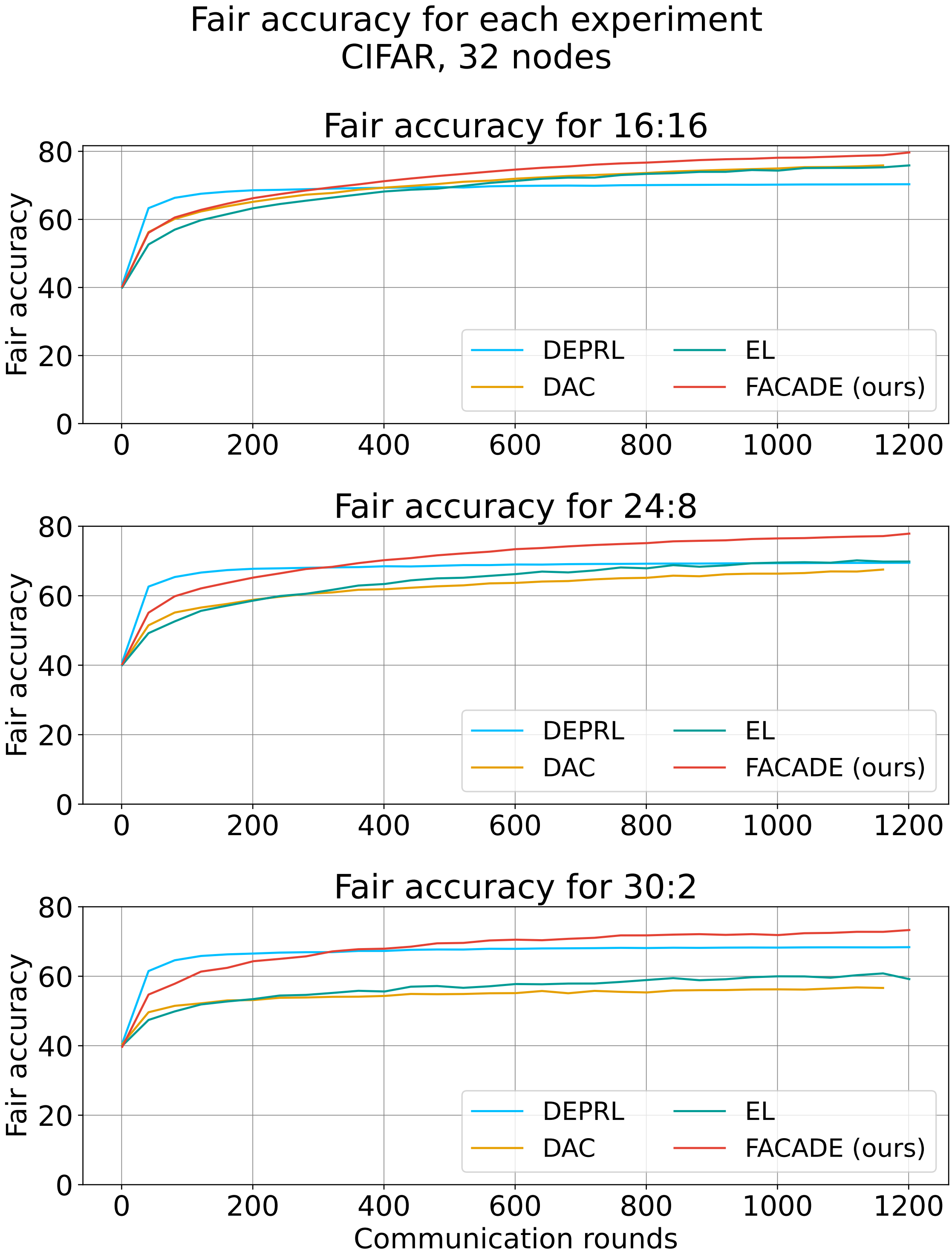


Figure A.1: Fair Accuracy (↑ is better) obtained on CIFAR-10.

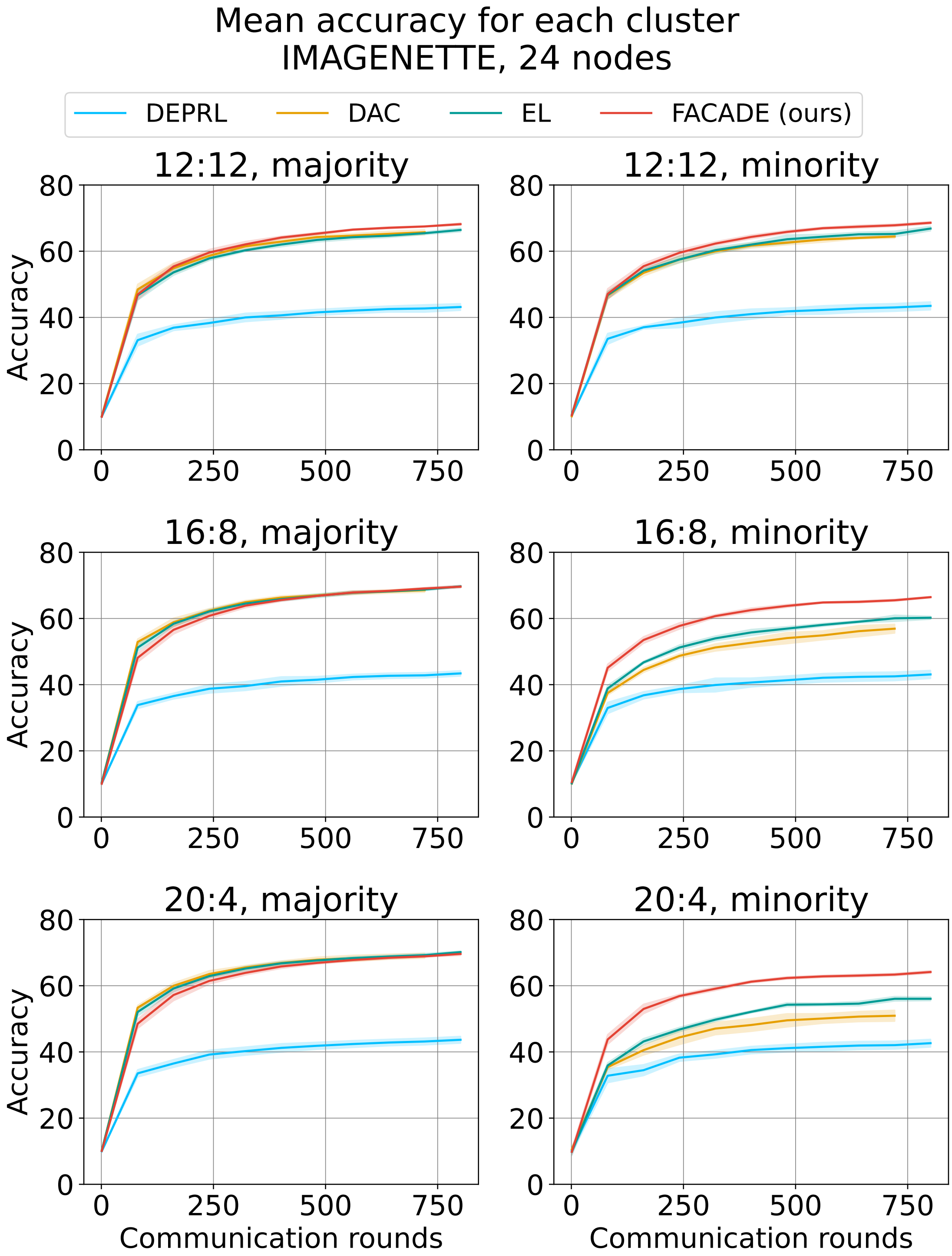
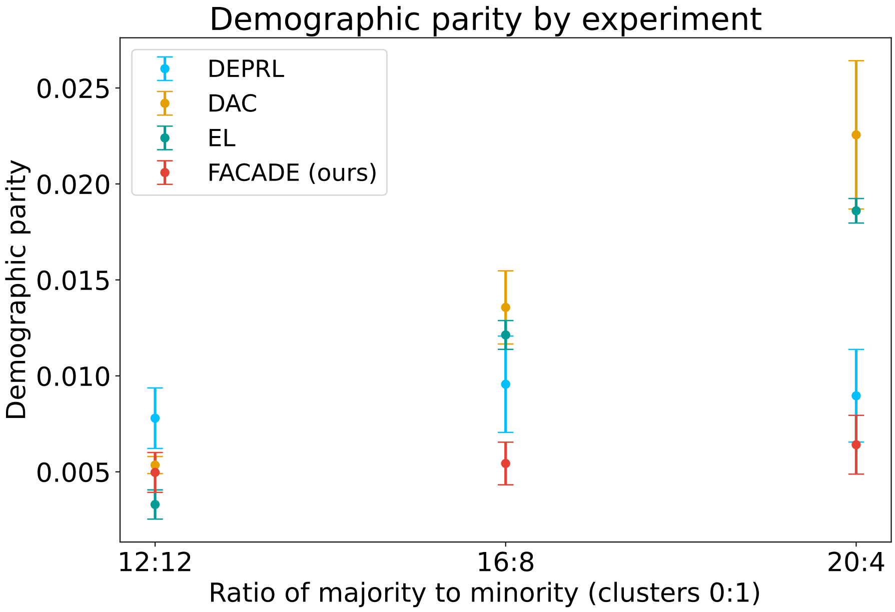
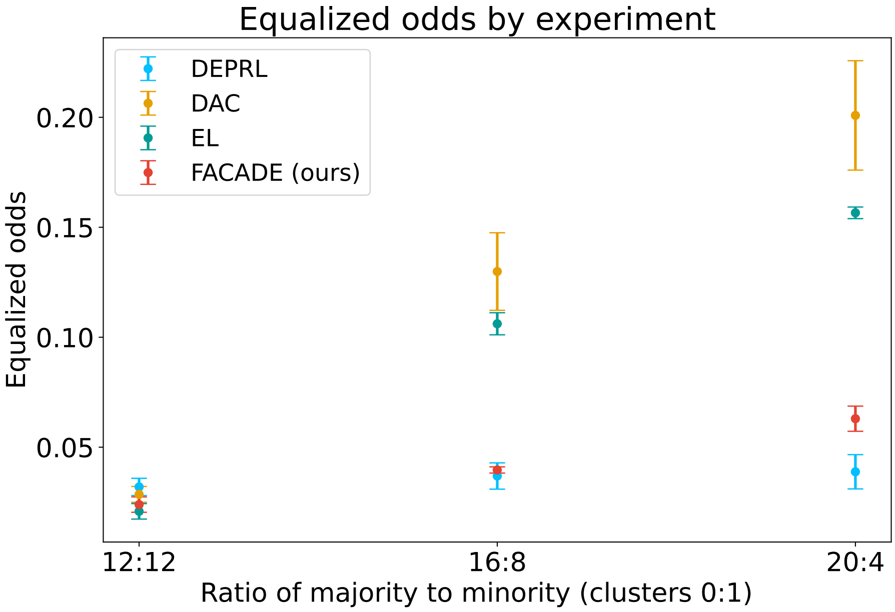


Figure A.2: Average accuracy (↑ is better) for the majority cluster (left) and the minority (right) obtained on Imagenette.



(a) Demographic parity



(b) Equalized odds

Figure A.3: (up) demographic parity (↓ is better) and (down) equalized odds (↓ is better) obtained for on Imagenette. FACADE is fairer than all the baselines (if we exclude DePRL)

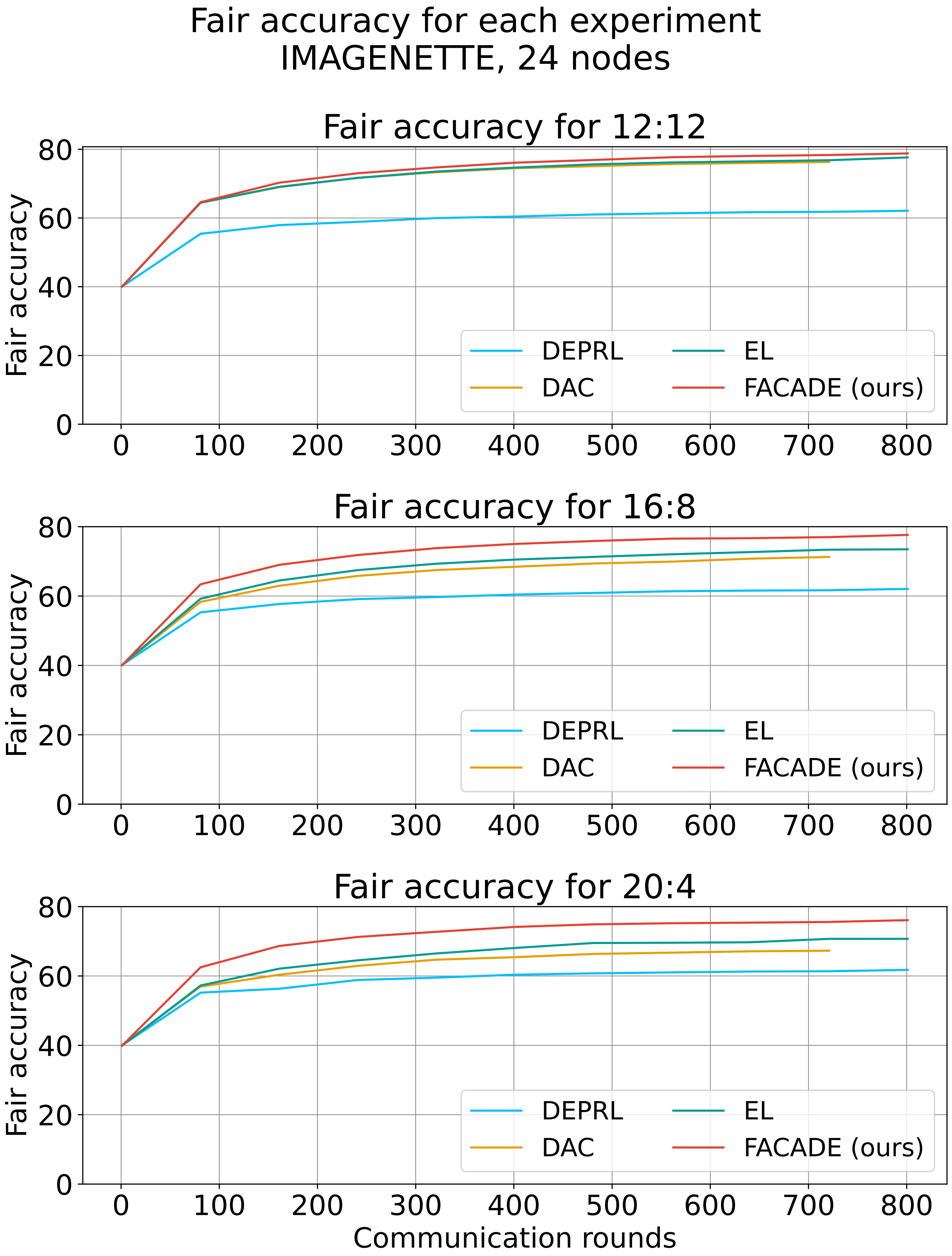


Figure A.4: Fair Accuracy (↑ is better) obtained on Imagenette.