A Appendix: Proofs

A.1 Proofs

Lemma. 1 *Let $\mathbf{P}_{\mathcal{A}}$ denote a matrix whose entry in row a and column k is $p(A = a|K = k)$ (i.e. the prior of group a in client k). Then, given a solution to the minimax problem across clients*

$$h^*, \boldsymbol{\lambda}^* \in \arg\min_{h\in\mathcal{H}} \max_{\boldsymbol{\lambda}\in\Delta^{|\mathcal{K}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\lambda}}} [\ell(h(X), Y)], \tag{10}$$

$\exists\ \boldsymbol{\mu}^* = \mathbf{P}_{\mathcal{A}}\boldsymbol{\lambda}^*$ that is solution to the following constrained minimax problem across sensitive groups

$$h^*, \boldsymbol{\mu}^* \in \arg\min_{h\in\mathcal{H}} \max_{\boldsymbol{\mu}\in\mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\mu}}} [\ell(h(X), Y)], \tag{11}$$

where the weighting vector $\boldsymbol{\mu}$ is constrained to belong to the simplex subset defined by $\mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1} \subseteq \Delta^{|\mathcal{A}|-1}$. In particular, if the set $\Gamma = \left\{\boldsymbol{\mu}' \in \mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}\text{: } \boldsymbol{\mu}' \in \arg\min_{h\in\mathcal{H}} \max_{\boldsymbol{\mu}\in\Delta^{|\mathcal{A}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\mu}}} [\ell(h(X), Y)]\right\} \neq \emptyset$, then $\boldsymbol{\mu}^* \in \Gamma$, and the minimax fairness solution across clients is also a minimax fairness solution across demographic groups.

Proof. The objective for optimizing the global model for the worst mixture of client distributions is:

$$\min_{h\in\mathcal{H}} \max_{\boldsymbol{\lambda}\in\Delta^{|\mathcal{K}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\lambda}}} [l(h(X), Y)] = \min_{h\in\mathcal{H}} \max_{\boldsymbol{\lambda}\in\Delta^{|\mathcal{K}|-1}} \sum_{k=1}^{|\mathcal{K}|} \lambda_k \mathop{\mathbb{E}}_{\mathcal{D}_k} [l(h(X), Y)], \tag{12}$$

given that $\mathcal{D}_{\boldsymbol{\lambda}} = \sum_{k=1}^{|\mathcal{K}|} \lambda_k p(X, Y|K = k)$. Since $p(X, Y|K = k) = \sum_{a\in\mathcal{A}} p(A = a|K = k)p(X, Y|A)$ with $p(A = a|K = k)$ being the prior of $a \in \mathcal{A}$ for client k, and $p(X, Y|A = a)$ is the distribution conditioned on the sensitive group $a \in \mathcal{A}$, Eq. (12) can be re-written as:

$$\begin{aligned}
&\min_{h\in\mathcal{H}} \max_{\boldsymbol{\lambda}\in\Delta^{|\mathcal{K}|-1}} \sum_{k=1}^{|\mathcal{K}|} \lambda_k \sum_{a\in\mathcal{A}} p(A = a|K = k) \mathop{\mathbb{E}}_{p(X,Y|A=a)} [l(h(X), Y)] = \\
&\min_{h\in\mathcal{H}} \max_{\boldsymbol{\lambda}\in\Delta^{|\mathcal{K}|-1}} \sum_{a\in\mathcal{A}} \mathop{\mathbb{E}}_{p(X,Y|A=a)} [l(h(X), Y)] \Big(\sum_{k=1}^{|\mathcal{K}|} p(A = a|K = k)\lambda_k \Big) = \\
&\min_{h\in\mathcal{H}} \max_{\boldsymbol{\mu}\in\mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}} \sum_{a\in\mathcal{A}} \mu_a \mathop{\mathbb{E}}_{p(X,Y|A=a)} [l(h(X), Y)].
\end{aligned} \tag{13}$$

Where we defined $\mu_a = \sum_{k=1}^{|\mathcal{K}|} p(A = a|K = k)\lambda_k$, $\forall a \in \mathcal{A}$, this creates the vector $\boldsymbol{\mu} = \mathbf{P}_{\mathcal{A}}\boldsymbol{\lambda} \subseteq \mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}$. It holds that the set of possible μ vectors satisfies $\mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1} \subseteq \Delta^{|\mathcal{A}|-1}$, since $\mathbf{P}_{\mathcal{A}} = \big\{\{p(A = a|K = k)\}_{a\in\mathcal{A}}\big\}_{k\in\mathcal{K}} \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{K}|}$, with $\sum_{a\in\mathcal{A}} p(A = a|K = k) = 1\ \forall k$ and $\boldsymbol{\lambda} \in \Delta^{|\mathcal{K}|-1}$.

Then, from the equivalence in Equation 13 we have that, given any solution

$$h^*, \boldsymbol{\lambda}^* \in \arg\min_{h\in\mathcal{H}} \max_{\boldsymbol{\lambda}\in\Delta^{|\mathcal{K}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\lambda}}} [\ell(h(X), Y)], \tag{14}$$

then $\boldsymbol{\mu}^* = \mathbf{P}_{\mathcal{A}}\boldsymbol{\lambda}^*$ is solution to

$$h^*, \boldsymbol{\mu}^* \in \arg\min_{h\in\mathcal{H}} \max_{\boldsymbol{\mu}\in\mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\mu}}} [\ell(h(X), Y)], \tag{15}$$

and

$$\mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\mu}^*}} [\ell(h^*(X), Y)] = \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\lambda}^*}} [\ell(h^*(X), Y)]. \tag{16}$$

In particular, if the space defined by $\mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}$ contains any group minimax fair weights, meaning that the set $\Gamma = \left\{\boldsymbol{\mu}' \in \mathbf{P}_{\mathcal{A}}\Delta^{|\mathcal{K}|-1}\text{: } \boldsymbol{\mu}' \in \arg\min_{h\in\mathcal{H}} \max_{\boldsymbol{\mu}\in\Delta^{|\mathcal{A}|-1}} \mathop{\mathbb{E}}_{\mathcal{D}_{\boldsymbol{\mu}}} [\ell(h(X), Y)]\right\}$ is not empty, then it follows that any $\boldsymbol{\mu}^*$ (solution to Equation 15) is already minimax fair with respect to the groups $\boldsymbol{\mu}^* \in \Gamma$. And therefore the client-level minimax solution is also a minimax solution across sensitive groups.

□

Lemma. 2 *Consider our federated learning setting (Figure 1, right) where each entity k has access to a local dataset $\mathcal{S}_k = \bigcup_{a\in\mathcal{A}} \mathcal{S}_{a,k}$ and a centralized machine learning setting (Figure 1, left) where there is a single entity that has access to a single dataset $\mathcal{S} = \bigcup_{k\in\mathcal{K}} \mathcal{S}_k = \bigcup_{k\in\mathcal{K}} \bigcup_{a\in\mathcal{A}} \mathcal{S}_{a,k}$ (i.e. this single entity in the centralized setting has access to the data of the various clients in the distributed setting).*

Then, Algorithm 1 and Algorithm 2 (in supplementary material, Appendix B) lead to the same global model provided that learning rates and model initialization are identical.

Proof. We will show that FedMinMax, in Algorithm 1 is equivalent to the centralized algorithm, in Algorithm 2 under the following conditions:

1. the dataset on client k, in FedMinMax is $\mathcal{S}_k = \bigcup_{a\in\mathcal{A}} \mathcal{S}_{a,k}$ and the dataset in centralized MinMax is $\mathcal{S} = \bigcup_{k\in\mathcal{K}} \mathcal{S}_k = \bigcup_{k\in\mathcal{K}} \bigcup_{a\in\mathcal{A}} \mathcal{S}_{a,k}$, and
2. the model initialization θ^0, the number of adversarial rounds T,[5], learning rate for the adversary η_μ and learning rate for the learner η_θ, are identical for both algorithms.

This can then be immediately done by showing that steps lines 3-7 in Algorithm 1 are entirely equivalent to step 3 in Algorithm 2. In particular, note that we can write:

$$\begin{aligned}
\hat{r}(\theta, \boldsymbol{\mu}) &= \sum_{a\in\mathcal{A}} \mu_a \hat{r}_a(\boldsymbol{\theta}) \\
&= \sum_{a\in\mathcal{A}} \mu_a \sum_{k\in\mathcal{K}} \tfrac{n_{a,k}}{n_a} \hat{r}_{a,k}(\boldsymbol{\theta}) \\
&= \sum_{a\in\mathcal{A}} \mu_a \tfrac{n}{n_a} \tfrac{1}{n} \sum_{k\in\mathcal{K}} n_{a,k} \hat{r}_{a,k}(\boldsymbol{\theta}) \\
&= \sum_{a\in\mathcal{A}} w_a \tfrac{1}{n} \sum_{k\in\mathcal{K}} \tfrac{n_{a,k}}{n_k} n_k \hat{r}_{a,k}(\boldsymbol{\theta}) \\
&= \sum_{k\in\mathcal{K}} \tfrac{n_k}{n} \sum_{a\in\mathcal{A}} w_a \tfrac{n_{a,k}}{n_k} \hat{r}_{a,k}(\boldsymbol{\theta}) \\
&= \sum_{k\in\mathcal{K}} \tfrac{n_k}{n} \hat{r}_k(\boldsymbol{\theta}, \boldsymbol{w})
\end{aligned} \tag{17}$$

because

$$\hat{r}_k(\boldsymbol{\theta}, \boldsymbol{w}) = \sum_{a\in\mathcal{A}} \frac{n_{a,k}}{n_k} w_a \hat{r}_{a,k}(\boldsymbol{\theta}), \text{ with } w_a = \frac{\mu_a}{\frac{n_a}{n}}, \text{ and } \hat{r}_a(\boldsymbol{\theta}) = \sum_{k\in\mathcal{K}} \frac{n_{a,k}}{n_a} \hat{r}_{a,k}(\boldsymbol{\theta}). \tag{18}$$

Therefore, the following model update:

$$\boldsymbol{\theta}^t = \sum_{k\in\mathcal{K}} \frac{n_k}{n} \boldsymbol{\theta}_k^t = \sum_{k\in\mathcal{K}} \frac{n_k}{n} \left(\boldsymbol{\theta}^{t-1} - \eta_\theta \nabla_\theta \hat{r}_k(\boldsymbol{\theta}^{t-1}, \boldsymbol{w}^{t-1})\right) \tag{19}$$

associated with step in 7, at round t of Algorithm 1, is entirely equivalent to the model update

$$\boldsymbol{\theta}^t = \boldsymbol{\theta}^{t-1} - \eta_\theta \nabla_\theta \hat{r}(\boldsymbol{\theta}^{t-1}, \boldsymbol{w}^{t-1}) \tag{20}$$

associated with step in line 3 at round t of Algorithm 2, provided that $\boldsymbol{\theta}^{t-1}$ is the same for both algorithms.

[5] In the federated Algorithm 1, we also refer to the adversarial rounds as communication rounds.

It follows therefore by induction that, provided the initialization $\boldsymbol{\theta}^0$ and learning rate η_θ are identical in both cases the algorithms lead to the same model. Also, from Eq. 18, we have that the projected gradient ascent step in line 4 of Algorithm 2 is equivalent to the step in line 10 of Algorithm 1. $\square$

B Appendix: Experiments

B.1 Experimental Details

Datasets. For the experiments we use the following datasets:

- **Synthetic.** Let $\mathcal{N}$ and Ber be the normal and Bernoulli distributions. The data were generated assuming the group variable $A \sim Ber(\frac{1}{2})$, the input features variable $X \sim \mathcal{N}(0,1)$ and the target variable $Y|X, A = a \sim Ber(h_a^*)$, where $h_a^* = u_a^l \mathbb{1}[x \leq 0] + u_a^h \mathbb{1}[x > 0]$ is the optimal hypothesis for group $A = a$. We select $\{u_0^h, u_1^h, u_0^l, u_1^l\} = \{0.6, 0.9, 0.3, 0.1\}$. As illustrated in Figure 2, left side, the optimal hypothesis h is equal to the optimal model for group $A = 0$.
- **FashionMNIST.** FashionMNIST is a grayscale image dataset which includes $60,000$ training images and 10,000 testing images. The images consist of 28×28 pixels and are classified into 10 clothing categories. In our experiments we consider each of the target categories to be a sensitive group too.

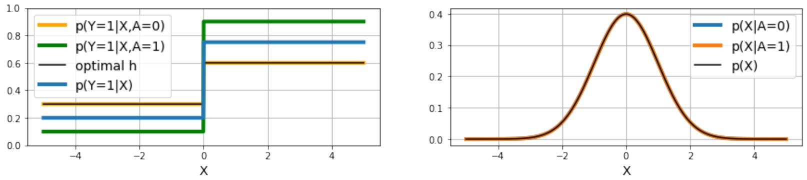

Figure 2: Illustration of the optimal hypothesis h and the conditional distributions $p(Y|X)$ and $p(X|A)$ for the generated synthetic dataset.

Experimental Setting and Model Architectures. For all the datasets, we use three-fold cross validation to compute the means and standard deviations of the accuracies and risks, with different random initializations. Also note that each client's data is unique, meaning that there are no duplicated examples across clients. We assuming that every client is available to participate at each communication round for every method. For q-FedAvg we use $q = \{0.2, 5.0\}$. The learning rate of the classifier for all methods is $\eta_\theta = 0.1$ and for the adversary in AFL and FedMinMax we use $\eta_\mu = \eta_\lambda = 0.1$. The local iterations for FedAvg and q-FedAvg are $E = 15$. For AFL and FedMinMax the batch size is equal to the number of examples per client while for FedAvg and q-FedAvg is equal to 100. For the synthetic dataset, we use an MLP architecture consisting of four hidden layers of size 512 and in the experiments for FashionMNIST we used a CNN architecture with two 2D convolutional layers with kernel size 3, stride 1 and padding 1. Each convolutional layer is followed with a maxpooling layer with kernel size 2, stride 2, dilation 1 and padding 0. All models were trained using Brier score loss function. A summary of the experimental setup is provided in Table 3.

Software & Hardware. The proposed algorithms and experiments are written in Python, leveraging PyTorch [23]. The experiments were realised using 1 × NVIDIA Tesla V100 GPU.

B.2 Additional Results

Experiments on FashionMNIST. In the *Partial access to Sensitive Groups (PSG)* setting, we distribute the data across 40 participants, 20 of which have access to groups *T-shirt*, *Trouser*, *Pullover*, *Dress* and *Coat* and the other 20 have access to *Sandal*, *Shirt*, *Sneaker*, *Bag* and *Ankle Boot*. The data distribution is unbalanced across clients since the size of local datasets differs among clients (i.e. $n_i \neq n_j \forall i, j \in \mathcal{K}, i \neq j$). In the *Equal access to Sensitive Groups (ESG)* setting, the 10 classes are equally distributed across the clients, creating a scenario where each client has access to the same amount of data examples and groups (i.e. $n_i = n_j \forall i, j \in \mathcal{K}, i \neq j$ and

Setting	Method	η_θ	Batch Size	Loss	Hypothesis Type	Epochs	η_μ or η_λ
ESG,SSG	AFL	0.1	n_k	Brier Score	MLP	-	0.1
ESG,SSG	FedAvg	0.1	100	Brier Score	MLP	15	-
ESG,SSG	q-FedAvg	0.1	100	Brier Score	MLP	15	-
ESG,SSG	FedMinMax (ours)	0.1	n_k	Brier Score	MLP	-	0.1
ESG,SSG	Centalized Minmax	0.1	n_k	Brier Score	MLP	-	0.1
ESG,SSG,PSG	AFL	0.1	n_k	Brier Score	CNN	-	0.1
ESG,SSG,PSG	FedAvg	0.1	100	Brier Score	CNN	15	-
ESG,SSG,PSG	FedMinMax (ours)	0.1	n_k	Brier Score	CNN	-	0.1
ESG,SSG,PSG	Centalized Minmax	0.1	n_k	Brier Score	CNN	-	0.1

Table 3: Summary of parameters used in the training process for all experiments. Epochs refers to the local iterations performed at each client,n_k is the number of local data examples in client k, η_θ is the model's learning rate and η_μ or η_λ is the adversary learning rates.

$n_{a,i} = n_{a,j} \forall i, j \in \mathcal{K}, a \in \mathcal{A}, i \neq j$). Finally, in the *Single access to Sensitive Groups (SSG)* setting, every client owns only one sensitive group and each group is distributed to only 4 clients. Again, the local datasets are different, $n_i \neq n_j \forall i, j \in \mathcal{K}, i \neq j$, creating an unbalanced data distribution.

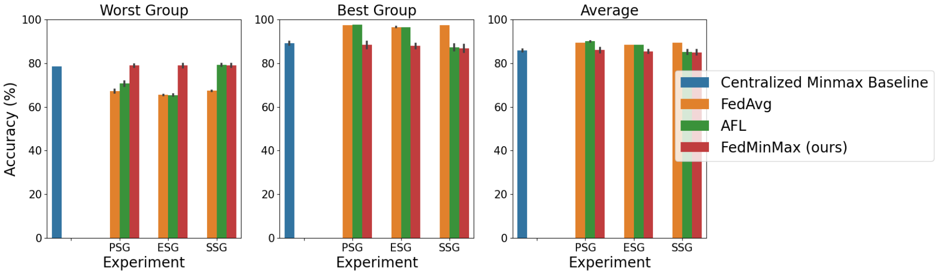

Figure 3: Worst Group, Best Group and Average accuracies for AFL, FedAvg and FedMinmax across different federated learning scenarios on the FashionMNIST dataset.

We show a comparison of the worst group *Shirt*, the best group *Trousers* and the average accuracies in Figure 3. FedMinMax enjoys a similar accuracy to the Centralized Minmax Baseline, as expected. AFL has similar performance FedMinMax in SSG, where across client fairness implies group fairness, in line with Lemma 1, and FedAvg has similar worst, best and average accuracy, across federated settings. An extended version of group risks is shown in Table 4.

Setting	Method	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
ESG	AFL	0.239±0.003	0.046±0.0	0.262±0.001	0.159±0.001	0.252±0.004	0.06±0.0	0.494±0.004	0.067±0.001	0.049±0.0	0.07±0.001
	FedAvg	0.243±0.003	0.046±0.0	0.262±0.001	0.158±0.003	0.253±0.002	0.061±0.0	0.492±0.003	0.068±0.0	0.049±0.0	0.069±0.0
	FedMinMax (ours)	0.261±0.006	0.191±0.016	0.256±0.027	0.217±0.013	0.223±0.031	0.207±0.027	**0.307±0.01**	0.172±0.016	0.193±0.021	0.156±0.011
SSG	AFL	0.267±0.009	0.194±0.023	0.236±0.013	0.226±0.012	0.262±0.012	0.201±0.026	**0.307±0.003**	0.178±0.033	0.205±0.025	0.162±0.021
	FedAvg	0.227±0.003	0.039±0.001	0.236±0.004	0.143±0.003	0.232±0.003	0.051±0.001	0.463±0.003	0.067±0.0	0.041±0.0	0.063±0.001
	FedMinMax (ours)	0.269±0.012	0.2±0.026	0.238±0.017	0.231±0.013	0.252±0.034	0.2±0.024	**0.309±0.011**	0.177±0.03	0.205±0.032	0.169±0.013
PSG	AFL	0.244±0.007	0.032±0.001	0.257±0.066	0.122±0.006	0.209±0.098	0.045±0.002	0.425±0.019	0.059±0.001	0.041±0.001	0.062±0.001
	FedAvg	0.229±0.008	0.039±0.0	0.236±0.004	0.142±0.002	0.232±0.003	0.052±0.001	0.464±0.011	0.067±0.001	0.042±0.001	0.063±0.001
	FedMinMax (ours)	0.263±0.013	0.177±0.026	0.228±0.011	0.21±0.019	0.238±0.025	0.182±0.03	**0.31±0.008**	0.16±0.027	0.184±0.031	0.154±0.018
Centalized Minmax Baseline		0.259±0.01	0.173±0.015	0.239±0.051	0.213±0.008	0.24±0.063	0.182±0.024	**0.311±0.006**	0.168±0.018	0.18±0.013	0.151±0.012

Table 4: Brier score risks for FedAvg, AFL and FedMinmax across different federated learning settings on FashionMNIST dataset. Extension of Table 2.

B.3 Centralized MinMax Algorithm

We provide the centralized version of FedMinMax in Algorithm 2.

Algorithm 2 CENTRALIZED MINMAX BASELINE

```
Input: $T$ : total number of adversarial rounds, $\eta_{\boldsymbol{\theta}}$: model learning rate, $\eta_{\boldsymbol{\mu}}$: adversary learning rate, $\mathcal{S}_a$: set of examples for group $a$, $\forall a \in \mathcal{A}$.
Server initializes $\boldsymbol{\mu}^0 \leftarrow \rho = \{|\mathcal{S}_a|/|\mathcal{S}|\}_{a \in \mathcal{A}}$ and $\boldsymbol{\theta}^0$ randomly.
for $t = 1$ to $T$ do
    Server computes $\boldsymbol{\theta}_k^t \leftarrow \boldsymbol{\theta}^{t-1} - \eta_\theta \nabla_\theta \hat{r}(\boldsymbol{\theta}^{t-1}, \boldsymbol{\mu}^{t-1})$
    Server updates
    $\boldsymbol{\mu}^t \leftarrow \prod_{\Delta^{|\mathcal{A}|-1}} \left(\boldsymbol{\mu}^{t-1} + \eta_{\boldsymbol{\mu}} \nabla_{\boldsymbol{\mu}} \langle\, \boldsymbol{\mu}^{t-1}, \hat{r}_a(\boldsymbol{\theta}^{t-1}) \rangle \right)$
end for
Outputs: $\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\theta}^t$
```