Appendix for *Semi-Variance Reduction for Fair Federated Learning*

A PROOFS

Lemma 1. *For any model parameter θ, the gradient of the global objective $F(\theta)$ defined in equation 3 can be expressed as*

$$\nabla F(\theta) = \sum_{i=1}^{n} w_i(\theta)\nabla f_i(\theta), \;\; w_i(\theta) = \frac{1}{n} + \frac{2\beta(f_i(\theta) - \overline{f}(\theta))}{n}, \;\; \overline{f}(\theta) = \frac{\sum_i f_i(\theta)}{n}. \tag{8}$$

Proof. From equation 3 and with $p_i = \frac{1}{n}$, we have:

$$\begin{aligned} n\nabla F(\theta) &= \sum_i \nabla f_i(\theta) + 2\beta \sum_i \Big[\Big(f_i(\theta) - \overline{f}(\theta)\Big)\Big(\nabla f_i(\theta) - \nabla \overline{f}(\theta)\Big)\Big] \\ &= \sum_i \nabla f_i(\theta) + 2\beta \sum_i \Big[\Big(f_i(\theta) - \overline{f}(\theta)\Big)\nabla f_i(\theta) - \Big(f_i(\theta) - \overline{f}(\theta)\Big)\nabla \overline{f}(\theta)\Big] \\ &= \sum_i \Big(1 + 2\beta(f_i(\theta) - \overline{f}(\theta))\Big)\nabla f_i(\theta) - 2\beta \sum_i \Big(f_i(\theta) - \overline{f}(\theta)\Big)\nabla \overline{f}(\theta) \\ &= \sum_i \Big(1 + 2\beta(f_i(\theta) - \overline{f}(\theta))\Big)\nabla f_i(\theta) \end{aligned} \tag{15}$$

Hence,

$$\nabla F(\theta) = \sum_i \frac{1 + 2\beta(f_i(\theta) - \overline{f}(\theta))}{n}\nabla f_i(\theta) \tag{16}$$

□

Derivation of equation 6

$$\begin{aligned} \sum_i f_i + \beta \sum_i \left| f_i(\theta) - \frac{1}{n}\sum_j f_j(\theta)\right|^2 &= \sum_i f_i + \beta \sum_i \left| \frac{n-1}{n} f_i(\theta) - \frac{1}{n}\sum_{j\neq i} f_j(\theta)\right|^2 \\ &= \sum_i f_i + \frac{\beta}{n^2}\sum_i \left| \sum_{j\neq i}(f_i(\theta) - f_j(\theta))\right|^2 \\ &\leq \sum_i f_i + \frac{\beta}{n^2}\sum_i \sum_{j\neq i} \left| f_i(\theta) - f_j(\theta)\right|^2 \\ &= \sum_i f_i + \frac{2\beta}{n^2}\sum_{j\neq i} \left| f_i(\theta) - f_j(\theta)\right|^2. \end{aligned} \tag{17}$$

Lemma 2. *In each communication round between the clients and the server, let $>_C$ denote the set of clients whose local loss function is greater than the average loss function $\overline{f}(\theta)$. For any model parameter θ, the gradient of the global objective $F(\theta)$ defined in equation 9 can be expressed as*

$$\nabla F(\theta) = \sum_{i=1}^{n} w_i(\theta)\nabla f_i(\theta), \tag{11}$$

where:

$$\overline{f}(\theta)=\frac{\sum_i f_i(\theta)}{n},\quad w_i(\theta)=\begin{cases}\frac{1}{n}+\frac{2\beta(f_i(\theta)-\overline{f}(\theta))}{n}-\frac{2\beta\sum_{j\in>\mathcal{C}}(f_j(\theta)-\overline{f}(\theta))}{n^2}, \textit{if } i\in>\mathcal{C}\\ \frac{1}{n}-\frac{2\beta\sum_{j\in>\mathcal{C}}(f_j(\theta)-\overline{f}(\theta))}{n^2}, \textit{if } i\notin>\mathcal{C}\end{cases} \tag{12}$$

Proof. From equation 9 and with $p_i=\frac{1}{n}$, we have:

$$\begin{aligned}n\nabla F(\theta)&=\sum_i\nabla f_i(\theta)+2\beta\sum_{i\in>\mathcal{C}}\Big[\Big(f_i(\theta)-\overline{f}(\theta)\Big)\Big(\nabla f_i(\theta)-\nabla\overline{f}(\theta)\Big)\Big]\\&=\sum_i\nabla f_i(\theta)+2\beta\sum_{i\in>\mathcal{C}}\Big[\Big(f_i(\theta)-\overline{f}(\theta)\Big)\nabla f_i(\theta)-\Big(f_i(\theta)-\overline{f}(\theta)\Big)\nabla\overline{f}(\theta)\Big]\\&=\sum_{i\notin>\mathcal{C}}\nabla f_i(\theta)+\sum_{i\in>\mathcal{C}}\Big(1+2\beta(f_i(\theta)-\overline{f}(\theta))\Big)\nabla f_i(\theta)-2\beta\sum_{i\in>\mathcal{C}}\Big(f_i(\theta)-\overline{f}(\theta)\Big)\nabla\overline{f}(\theta)\end{aligned} \tag{18}$$

The last term in the above equation can be written as:

$$\begin{aligned}&-2\beta\sum_{i\in>\mathcal{C}}\Big(f_i(\theta)-\overline{f}(\theta)\Big)\nabla\overline{f}(\theta) \qquad (19)\\&=-\Big[\frac{2\beta}{n}\Big(\sum_{i\in>\mathcal{C}}f_i(\theta)\Big)\times\Big(\sum_j\nabla f_j(\theta)\Big)\Big]+\Big[\frac{2\beta}{n}\Big(\sum_{i\in>\mathcal{C}}\overline{f}(\theta)\Big)\times\Big(\sum_j\nabla f_j(\theta)\Big)\Big]\\&=-\Big[\frac{2\beta}{n}\Big(\sum_{i\in>\mathcal{C}}f_i(\theta)-\overline{f}(\theta)\Big)\times\Big(\sum_j\nabla f_j(\theta)\Big)\Big]\end{aligned}$$

Hence,

$$\begin{aligned}n\nabla F(\theta)=&\sum_{i\in>\mathcal{C}}\Big(1+2\beta(f_i(\theta)-\overline{f}(\theta))-\frac{2\beta}{n}\Big(\sum_{j\in>\mathcal{C}}f_j(\theta)-\overline{f}(\theta)\Big)\Big)\nabla f_i(\theta)\\&+\sum_{i\notin>\mathcal{C}}\Big(1-\frac{2\beta}{n}\Big(\sum_{j\in>\mathcal{C}}f_j(\theta)-\overline{f}(\theta)\Big)\Big)\nabla f_i(\theta)\end{aligned} \tag{20}$$

Therefore,

$$\begin{aligned}\nabla F(\theta)=&\sum_{i\in>\mathcal{C}}\Big(\frac{1+2\beta(f_i(\theta)-\overline{f}(\theta))-\frac{2\beta}{n}\Big(\sum_{j\in>\mathcal{C}}f_j(\theta)-\overline{f}(\theta)\Big)}{n}\Big)\nabla f_i(\theta)\\&+\sum_{i\notin>\mathcal{C}}\Big(\frac{1-\frac{2\beta}{n}\Big(\sum_{j\in>\mathcal{C}}f_j(\theta)-\overline{f}(\theta)\Big)}{n}\Big)\nabla f_i(\theta)\end{aligned} \tag{21}$$

□

Lemma 3. *Assuming $P_i(x,y)=P_i(x|y)P_i(y)=P(x|y)P_i(y)$ for $i\in\{1,\dots,n\}$, for any model parameter θ, `Semi-VRed` global objective $F(\theta)$ defined in equation 9 can be expressed as*

$$F(\theta)=\sum_{j=1}^{C}\overline{P}(j)\overline{\ell}_j(\theta)+\frac{\beta}{n}\sum_{i=1}^{n}\Big(\sum_{j=1}^{C}[P_i(j)-\overline{P}(j)]\overline{\ell}_j(\theta)\Big)_+^2, \tag{13}$$

where $\overline{P}(j)=\frac{\sum_{i=1}^{n}P_i(j)}{n}$ is the marginal distribution of class j in the global dataset.

Proof. From equation 9 and with $p_i = \frac{1}{n}$, we have:

$$\begin{aligned}
\overline{f}(\theta) &= \sum_{i=1}^{n} \frac{f_i(\theta)}{n} = \frac{1}{n} \sum_{i=1}^{n} \Big[\mathbb{E}_{(x,y)\sim p_i(x,y)} [\ell(h(x,\theta), y)] \Big] \\
&= \frac{1}{n} \sum_{i=1}^{n} \Big[\sum_{j=1}^{C} p_i(j) \times \mathbb{E}_{(x,y)\sim p(x|y=j)} [\ell(h(x,\theta), j)] \Big] \\
&= \frac{1}{n} \sum_{i=1}^{n} \Big[\sum_{j=1}^{C} p_i(j) \overline{\ell}_j(\theta)] \Big] \\
&= \sum_{j=1}^{C} \Big[(\frac{\sum_{i=1}^{n} p_i(j)}{n}) \overline{\ell}_j(\theta) \Big] \\
&= \sum_{j=1}^{C} \overline{p}(j) \overline{\ell}_j(\theta), \qquad (22)
\end{aligned}$$

where $\overline{p}(j) = \frac{\sum_{i=1}^{n} p_i(j)}{n}$ is the ratio of data points with label j in the global dataset. Similarly, we have:

$$f_i(\theta) = \sum_{j=1}^{C} p_i(j) \overline{\ell}_j(\theta). \qquad (23)$$

By plugging in the above equivalences for $f_i(\theta)$ and $\overline{f}(\theta)$ into equation 9, we get to equation 13. □

Algorithm 2: FedAvg

Input: global epoch T, client number n, loss function f_i, number of samples n_i for client i, number of total samples N, initial global model θ_0, local step number K_i for client i, learning rate η
for $t = 0, 1 \dots T-1$ **do**
  randomly select $\mathcal{S}_t \subseteq [n]$
  $\theta_t^{(i)} = \theta_t$ for $i \in \mathcal{S}_t$, $N = \sum_{i \in \mathcal{S}_t} n_i$
  for i *in* $\mathcal{S}_t$ **do** `// in parallel`
    starting from $\theta_t^{(i)}$, take K_i local SGD steps on f_i, with learning rate η, to find $\theta_{t+1}^{(i)}$
  $\theta_{t+1} = \sum_{i \in \mathcal{S}_t} \frac{n_i}{N} \theta_{t+1}^{(i)}$
Output: global model θ_T

In algorithm 2, we have reported the `FedAvg` algorithm for easier reference. Due to the similarity of `VRed`'s objective function to that of `FedAvg`, we build its convergence proof on top of the proof for `FedAvg` in Zhang et al. (2022a). We borrow the following theorem on convergence of `FedAvg` from the same work. We refer the reader to the work for the detailed proof.

Theorem 2 (FedAvg). *Denote $p_i = \frac{n_i}{N}$. Given Assumption 1, assume that the local learning rate satisfies $\eta \leq \frac{1}{6LK_i}$ for any $i \in [n]$ and*

$$\eta \leq \frac{1}{L} \sqrt{\frac{1}{24(e-2)(\sum_i p_i^2)(\sum_i K_i^4)}}. \qquad (24)$$

Running FedAvg for T global epochs we have:

$$\min_{0 \leq t \leq T-1} \mathbb{E} \|\nabla F(\theta_t)\|^2 \leq \frac{12}{(11\gamma - 9)\eta} \left(\frac{F_0 - F^*}{T} + \Psi_\sigma \right),$$

with $\gamma = \sum_i p_i K_i$ *for full participation and* $\gamma = \min_i K_i$ *for partial participation,* $F_0 = F(\theta_0)$, $F^* = \min_\theta F(\theta)$ *the optimal value, and*

$$\Psi_\sigma = \frac{\eta}{2}\Big(\sum_{i=1}^n p_i^2\Big)\Big[\sum_{i=1}^n K_i^2(\sigma_{l,i}^2 + 2\sigma_g^2) + 2(e-2)\eta^2 L^2 \sum_{i=1}^n K_i^3\left(\sigma_{l,i}^2 + 6K_i\sigma_g^2\right)\Big].$$

Based on the above theorem for `FedAvg`, we now prove the convergence of our `VRed` algorithm.

Theorem 1 (`VRed`). *Denote* $\widetilde{L} = (L + \beta ML + 2\beta L_0^2)$ *and* $p_i = \frac{n_i}{N}$. *Given Assumptions 1 and 2, assume that the local learning rate satisfies* $\eta \leq \frac{1}{6\widetilde{L}K_i}$ *for any* $i \in [n]$ *and:*

$$\eta \leq \frac{1}{L}\sqrt{\frac{1}{24(e-2)(\sum_i p_i^2)(\sum_i K_i^4)}} \tag{14}$$

By running Algorithm 1 for T *global epochs we have:*

$$\min_{0\leq t\leq T-1} \mathbb{E}\|\nabla F(\theta_t)\|^2 \leq \frac{12}{(11\gamma - 9)\eta}\left(\frac{F_0 - F^*}{T} + \widetilde{\Psi}_\sigma\right),$$

with $\gamma = \sum_i p_i K_i$ *for full participation and* $\gamma = \min_i K_i$ *for partial participation,* $F_0 = F(\theta_0)$, $F^* = \min_\theta F(\theta)$ *the optimal value, and*

$$\widetilde{\Psi}_\sigma = \frac{\eta}{2}\Big(\sum_{i=1}^n p_i^2\Big)\Big[\sum_{i=1}^n K_i^2(\widetilde{\sigma}_{l,i}^2 + 2\widetilde{\sigma}_g^2) + 2(e-2)\eta^2 L^2 \sum_{i=1}^n K_i^3\left(\widetilde{\sigma}_{l,i}^2 + 6K_i\widetilde{\sigma}_g^2\right)\Big],$$

where $\widetilde{\sigma}_{l,i}^2 = \big(2(2\beta\mu+1)^2\sigma_{l,i}^2 + 8\beta^2 M^2 L_0^2\big)$ *and* $\widetilde{\sigma}_g = \big((2\beta\mu+1)\sigma_g + 2\beta M L_0\big)$ *and* $\mu = \Sigma_i p_i f_i(\theta_0)$.

Proof. We first rewrite a simplified version of the `VRed` objective function (equation 3) in the following.

$$F(\theta) = \sum_i p_i G_i(\theta) = \sum_i p_i\big(f_i(\theta) + \beta\left(f_i(\theta) - \mu\right)^2\big), \tag{25}$$

where $G_i(\theta) = f_i(\theta) + \beta\left(f_i(\theta) - \mu\right)^2$, and also, $\mu = \Sigma_i p_i f_i$ is fixed during clients local computations. In the beginning of each communication round, we update μ for the next round of local computations. In other words:

$$\mu_{t+1} = \sum_{i=1}^n p_i f_i(\theta_{t+1}). \tag{26}$$

With these notations, it suffices to find the constants in Assumption 1 for $G_i(\theta)$.

Smoothness We have:

$$\begin{aligned}\nabla^2 G_i(\theta) &= \nabla^2 f_i(\theta) + 2\beta((f_i(\theta) - \mu)\nabla^2 f_i(\theta) + \nabla f_i(\theta)\nabla f_i(\theta)^\top)\\ &\preceq L + 2\beta(\frac{M}{2}\nabla^2 f_i(\theta) + \nabla f_i(\theta)\nabla f_i(\theta)^\top)\\ &\preceq (L + \beta ML + 2\beta L_0^2)I,\end{aligned} \tag{27}$$

where in the last line we used Assumption 2 and the following:

$$\begin{aligned}\|\nabla f_i(\theta)\nabla f_i(\theta)^\top\|_{sp} &= \sup_{\|u\|=1}\sup_{\|v\|=1}\langle\nabla f_i(\theta)\nabla f_i(\theta)^\top u; v\rangle\\ &= \sup_{\|u\|=1}\sup_{\|v\|=1}(\nabla f_i(\theta)^\top u)^\top\nabla f_i(\theta)^\top v = \|\nabla f_i(\theta)\|^2 \leq L_0^2,\end{aligned} \tag{28}$$

where in the second line we used Cauchy–Schwarz inequality and Assumption 2. Hence, from eq. (27), we conclude that $G_i(\theta)$ is Lipschitz smooth:

$$\|\nabla G_i(\theta) - \nabla G_i(\theta')\| \leq (L + \beta ML + 2\beta L_0^2)\|\theta - \theta'\|. \tag{29}$$

Variance constants For the global variance between clients gradients, we have:

$$
\begin{aligned}
&\|\nabla G_i(\theta) - \nabla G_j(\theta)\| \\
&= \big\|\big(\nabla f_i(\theta) + 2\beta(f_i(\theta)-\mu)\nabla f_i(\theta)\big) - \big(\nabla f_j(\theta) + 2\beta(f_j(\theta)-\mu)\nabla f_j(\theta)\big)\big\| \\
&\leq \|\nabla f_i(\theta) - \nabla f_j(\theta)\| + 2\beta\|(f_i(\theta)-\mu)\nabla f_i(\theta) - (f_j(\theta)-\mu)\nabla f_j(\theta)\| \\
&\leq \|\nabla f_i(\theta) - \nabla f_j(\theta)\| + 2\beta\|f_i(\theta)\nabla f_i(\theta) - f_j(\theta)\nabla f_j(\theta)\| + 2\beta\mu\|\nabla f_i(\theta) - \nabla f_j(\theta)\| \\
&\leq (2\beta\mu+1)\|\nabla f_i(\theta) - \nabla f_j(\theta)\| + 2\beta\|f_i(\theta)\nabla f_i(\theta) - f_j(\theta)\nabla f_j(\theta)\| \\
&\leq (2\beta\mu+1)\|\nabla f_i(\theta) - \nabla f_j(\theta)\| + 2\beta f_i(\theta)\|\nabla f_i(\theta)\| + 2\beta f_j(\theta)\|\nabla f_j(\theta)\| \\
&\leq (2\beta\mu+1)\sigma_g + 2\beta f_i(\theta)\|\nabla f_i(\theta)\| + 2\beta f_j(\theta)\|\nabla f_j(\theta)\| \\
&\leq (2\beta\mu+1)\sigma_g + \beta M\|\nabla f_i(\theta)\| + \beta M\|\nabla f_j(\theta)\| \\
&\leq (2\beta\mu+1)\sigma_g + 2\beta M L_0, \qquad (30)
\end{aligned}
$$

where in line seven we used Assumption 1, and in lines eight and nine, we used Assumption 2. Therefore, we have:

$$
\|\nabla G_i(\theta) - \nabla G_j(\theta)\|^2 \leq \bigg((2\beta\mu+1)\sigma_g + 2\beta M L_0\bigg)^2. \qquad (31)
$$

For the local variance term, we define $\varphi(t) = t + \beta(t-\mu)^2$. Similar to the derivation of equation 30, we have:

$$
\begin{aligned}
&\|\nabla G_i(\theta) - \nabla(\varphi \circ \ell_S)(\theta)\| \\
&= \big\|\big(\nabla f_i(\theta) + 2\beta(f_i(\theta)-\mu)\nabla f_i(\theta)\big) - \big(\nabla \ell_S(\theta) + 2\beta(\ell_S(\theta)-\mu)\nabla \ell_S(\theta)\big)\big\| \\
&\leq (2\beta\mu+1)\|\nabla f_i(\theta) - \nabla \ell_S(\theta)\| + 2\beta f_i(\theta)\|\nabla f_i(\theta)\| + 2\beta \ell_S(\theta)\|\nabla \ell_S(\theta)\| \\
&\leq (2\beta\mu+1)\|\nabla f_i(\theta) - \nabla \ell_S(\theta)\| + 2\beta M L_0, \qquad (32)
\end{aligned}
$$

where in line four, we used Assumption 2. By taking the square on both sides and the expectation over $S \sim \mathcal{B}_i^b$, we get:

$$
\begin{aligned}
\mathbb{E}_{S\sim\mathcal{B}_i^b}\|\nabla G_i(\theta) - \nabla(\varphi\circ\ell_S)(\theta)\|^2 &\leq \mathbb{E}_{S\sim\mathcal{B}_i^b}\bigg((2\beta\mu+1)\|\nabla f_i(\theta) - \nabla\ell_S(\theta)\| + 2\beta M L_0\bigg)^2 \\
&\leq \mathbb{E}_{S\sim\mathcal{B}_i^b}\bigg(2(2\beta\mu+1)^2\|\nabla f_i(\theta) - \nabla\ell_S(\theta)\|^2 + 8\beta^2 M^2 L_0^2\bigg) \\
&= \bigg(2(2\beta\mu+1)^2\sigma_{l,i}^2 + 8\beta^2 M^2 L_0^2\bigg). \qquad (33)
\end{aligned}
$$

In the third line, we used $(a+b)^2 \leq 2(a^2+b^2)$. We also used Assumption 1 in the same line.

□

B EXAMPLES

We borrow the following example on class imbalance in `FL` from Shen et al. (2022) to provide a better understanding of lemma 3. The following example shows an extreme class imbalance, which `Semi-VRed` can handle efficiently.

Example 1. *Let u be the uniform distribution over the existing C classes. Also, let δ_c be the Dirac distribution of class c. Now, without loss of generality, lets assume that $C = 2$ (binary classification problem). For the n existing clients, we have:*

$$
p_i(y) = \begin{cases} \alpha u + (1-\alpha)\delta_1 & \text{if } i = 1 \\ \alpha u + (1-\alpha)\delta_2 & \text{if } i \in \{2,\dots,n\} \end{cases} \qquad (34)
$$

Accordingly, we have:

$$p_i(1) = \begin{cases} 1 - \frac{\alpha}{2} & \text{if } i = 1 \\ \frac{\alpha}{2} & \text{if } i \in \{2, \dots, n\} \end{cases} \tag{35}$$

$$p_i(2) = \begin{cases} \frac{\alpha}{2} & \text{if } i = 1 \\ 1 - \frac{\alpha}{2} & \text{if } i \in \{2, \dots, n\} \end{cases} \tag{36}$$

Therefore,

$$f_i(\theta) = \begin{cases} (1 - \frac{\alpha}{2})\overline{\ell}_1(\theta) + \frac{\alpha}{2}\overline{\ell}_2(\theta) & \text{if } i = 1 \\ \frac{\alpha}{2}\overline{\ell}_1(\theta) + (1 - \frac{\alpha}{2})\overline{\ell}_2(\theta) & \text{if } i \in \{2, \dots, n\} \end{cases} \tag{37}$$

Hence,

$$\overline{f}(\theta) = \Big(\frac{\alpha}{2} + \frac{1-\alpha}{n}\Big)\overline{\ell}_1(\theta) + \Big(\frac{\alpha}{2} + \frac{(1-\alpha)(n-1)}{n}\Big)\overline{\ell}_2(\theta) \tag{38}$$

Clearly, we can see that if $\alpha \approx 0$ and n is large, then $\overline{\ell}_1(\theta)$, which is the loss over the minority data will have a small weight, which leads to $\overline{\ell}_1(\theta)$ being larger than $\overline{\ell}_2(\theta)$ and poor performance on the minority class 1. Now, if we rewrite the `Semi-VRed` *objective function (equation 9), we have:*

$$F(\theta) = \Big(\frac{\alpha}{2} + \frac{1-\alpha}{n}\Big)\overline{\ell}_1(\theta) + \Big(\frac{\alpha}{2} + \frac{(1-\alpha)(n-1)}{n}\Big)\overline{\ell}_2(\theta) + \frac{\beta(n-1)^2(1-\alpha)^2}{n^3}\Big(\overline{\ell}_1(\theta) - \overline{\ell}_2(\theta)\Big)^2 \tag{39}$$

For $\alpha \approx 0$:

$$F(\theta) \approx \overline{\ell}_2(\theta) + \frac{\beta}{n}\Big(\overline{\ell}_1(\theta) - \overline{\ell}_2(\theta)\Big)^2, \tag{40}$$

which improves $\overline{\ell}_2(\theta)$, thanks to its regularization term. Hence, the performance of client 1 and consequently, fairness in the system will improve.

C Experimental setup

In this section, we provide more experimental details that are deferred from the main paper. For each experiment, we report the average result obtained from three runs with different random seeds. For our experiments, we used an internal GPU server with six NVIDIA Tesla P100. The experiments last about 4 weeks in total.

C.1 Datasets and models

In this subsection, we describe the datasets we use in our experiments. For all the datasets we use a batch size of 64.

CIFAR-10/100 (Krizhevsky et al., 2009) are two image classification datasets vastly used in the literature as benchmark datasets. Each of these datasests contains 50000 sample images with 10/100 balanced classes for CIFAR-10 and CIFAR-100, respectively. We use Dirichlet allocation (Wang et al., 2019) to distribute the data among 50 clients with label shift: we split the set of samples from class k ($\mathcal{S}_k$) to n subsets ($\mathcal{S}_k = \mathcal{S}_{k,1} \cup \mathcal{S}_{k,2} \cup \ldots \cup \mathcal{S}_{k,n}$), where n is the number of clients and $\mathcal{S}_{k,j}$ corresponds to the client j. We do the split based on Dirichlet distribution with parameter 0.05 (`Dir(0.05)`). When the split is done for all classes, we gather the samples corresponding to each client from different classes: assuming there are C classes in total $\mathcal{S}_{1,j} \cup \mathcal{S}_{2,j} \cup \ldots \cup \mathcal{S}_{C,j}$ is the data allocated to the client j. Having allocated the data of each client, we split them into train and

test set for each client. The train-test split ratio is 50-50 and 60-40 for CIFAR-10 and CIFAR-100, respectively.

CINIC-10 (Darlow et al., 2018) is another benchmark vision dataset that we use in our experiments. There are a total of 270,000 sample images, which we distribute with label shift between 50 clients based on `Dir(0.5)` distribution Wang et al. (2019). We then randomly split the data of each client into train and test sets with split ratio 50-50.

StackOverflow (The Tensorflow Federated Authors, 2019) is a language dataset consisting of Shakespeare dialogues for the task of next word prediction. There is a natural heterogeneous partition of the dataset and we treat each speaking role as a client. We filter out the clients (speaking roles) with less than 10,000 samples from the original dataset and randomly select 20 clients from the remaining. Finally, we split the data of each client into train and test sets with a ratio of 50-50.

Table 2 provides a summary of the datasets we used and the models used for each of them.

Table 2: Details of the experiments and the datasets. ResNet-18: residual neural network (He et al., 2016). GN: Group Normalization (Wu & He, 2018); RNN: Recurrent Neural Network; LSTM: Long Short-Term Memory layer; FC: fully connected layer.

Dataset	Train samples	Test samples	Partition method	clients	Model
CIFAR-10	24959	25041	`Dir(0.05)`	50	ResNet-18 + GN
CIFAR-100	39445	10555	`Dir(0.05)`	50	ResNet-18 + GN
CINIC-10	134713	134966	`Dir(0.5)`	50	ResNet-18 + GN
StackOverflow	109671	109621	realistic partition	20	RNN (1 LSTM + 2 FC)

C.2 Algorithms and their hyperparameters

We use most recent fair `FL` algorithms existing in the literature as our baseline algorithms, including: `FedAvg` (McMahan et al., 2017), `q-FFL` (Li et al., 2020c), `AFL` (Mohri et al., 2019), `PropFair` (Zhang et al., 2022a), `TERM` (Li et al., 2020a), `GiFair` (Yue et al., 2021). For each pair of dataset and algorithm, we find the best learning rate based on a grid search. In the following, we have reported the learning rate grid we use for each dataset:

- CIFAR-10: {`1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2`};
- CIFAR-100: {`1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2`};
- CINIC-10: {`1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2`};
- StackOverflow: {`1e-2, 5e-2, 1e-1, 5e-1, 1`}.

The best learning rate used for each (dataset, algorithm) pair is reported in Table 4.

We now explain the algorithms we use and how we tune their hyperparameters. We adapt TERM with only client-level fairness ($\alpha > 0$) and no sample-level fairness ($\tau = 0$). We tune the hyperparameter α for each dataset based on a grid search in the grid $\{0.01, 0.1, 0.5, 1\}$. We have reported the best value of α for each dataset in Table 5. For AFL, there are two hyperparameters: γ_w and γ_λ. We tune the learning rate γ_w from the corresponding grid and choose the default value $\gamma_\lambda = 0.1$. For q-FFL, we use the q-FedAvg algorithm (Li et al., 2020c). We also tune the hyperparameter q from the grid $\{0.01, 0.1, 1\}$. We find that for all the used datasets, $q = 0.1$ has the best peformance (as reported in Table 5). We also tried larger values out of the grid and they often lead to divergence of the `q-FFL` algorithm. We adopt the Global GiFair model (Yue et al., 2021), which results in a single global model. In order to have client-level fairness, we treat each client as a group of size 1. For tuning the regularization weight of GiFair (λ), we follow (Yue et al., 2021). As stated in the paper, there is an upper-bound for λ (see §3 in the paper). For our experiments, the upper-bound is $\lambda \leq \min_i\{\frac{w_i}{n-1}\}$, where w_i is the ratio of total samples allocated to the client i and n is the number of clients. We try four different values in the interval and choose the best one. When the number of clients is large, this upper-bound is small, and for all of our datasets, this upper-bound was the best value, as reported in Table 5. We tune M for the PropFair algorithm based on a grid search in $\{2, 3, 4, 5\}$. Finally, for our VRed and Semi-VRed algorithms, we tune the regularization weight β based on grid search on the grid $\{0.01, 0.05, 0.1, 0.2, 0.5, 1\}$. Larger values of β often resulted in

Table 3: Details of the existing fairfl algorithms. f_i is the loss function of the client i.

FL algorithm	Objective	Reference
FedAvg	$\sum_i f_i$	McMahan et al. (2017)
AFL	$\max_i f_i$	Mohri et al. (2019)
q-FFL	$\sum_i f_i^{q+1}$	Li et al. (2020c)
TERM	$\sum_i e^{\alpha f_i}$	Li et al. (2020a)
PropFair	$-\sum_i \log(M - f_i)$	Zhang et al. (2022a)
GiFair	$\sum_i f_i + \lambda \sum_{i \neq j} \lvert f_i - f_j \rvert$	Yue et al. (2021)
VRed	$\sum_i f_i + \beta \sum_i \Big(f_i(\theta) - \frac{1}{n} \sum_j f_j(\theta) \Big)^2$	this work
Semi-VRed	$\sum_i f_i + \beta \sum_i \Big(f_i(\theta) - \frac{1}{n} \sum_j f_j(\theta) \Big)_+^2$	this work

Table 4: The best learning rates used for training each algorithm on different datasets.

Datasets	FedAvg	q-FFL	AFL	TERM	PropFair	GiFair	VRed	Semi-VRed
CIFAR-10	5e-3	5e-3	5e-3	1e-2	1e-2	5e-3	5e-3	5e-3
CIFAR-100	2e-3	2e-3	5e-3	1e-2	1e-2	5e-3	5e-3	5e-3
CINIC-10	1e-2	5e-3	1e-2	1e-2	2e-2	2e-2	5e-3	5e-3
StackOverflow	2e-1	5e-2	5e-2	2e-1	5e-1	2e-1	5e-1	5e-1

the divergence of the algorithms. We have reported the best value of all of the hyperparameters for each dataset in Table 5.

Table 5: The best values of hyperparameters used for different datasets, chosen based on grid search.

Algorithm	CIFAR-10	CIFAR-100	CINIC-10	StackOverflow
q-FFL q	1e-1	1e-1	1e-1	1e-1
TERM α	1e-2	5e-1	5e-1	5e-1
GiFair λ	6e-5	2.6e-4	5e-5	2.4e-3
PropFair M	3	3	5	4
VRed β	5e-1	1e-1	2e-1	1e-1
Semi-VRed β	5e-1	1e-2	2e-1	2e-1

C.3 DETAILED RESULTS

In Table 6, we report detailed results obtained from the algorithms we study in this work. We use a default batch size of 64 for all the experiments. The statistics we report include: 1. the average test accuracy across clients (overall average performance) 2. the standard deviation of test accuracies across clients 3. the lowest (worst) test accuracy among clients 4. the lowest 10% test accuracies 5. the lowest 20% test accuracies 6. the highest 10% test accuracies. For each experiment, we report the average result obtained from three runs with different random seeds. As can be observed, our proposed algorithms VRed and Semi-VRed consistently beat almost all the baseline algorithms across different datasets in terms of various fairness metrics. Also, Semi-VRed can improve the overall average performance (mean test accuracy) for three of the datsets as well.

Following Figure 1, we have compared our proposed algorithms with the baseline algorithms in terms of their worst 20% test accuracies as well and the visualized results are shown in Figure 2.

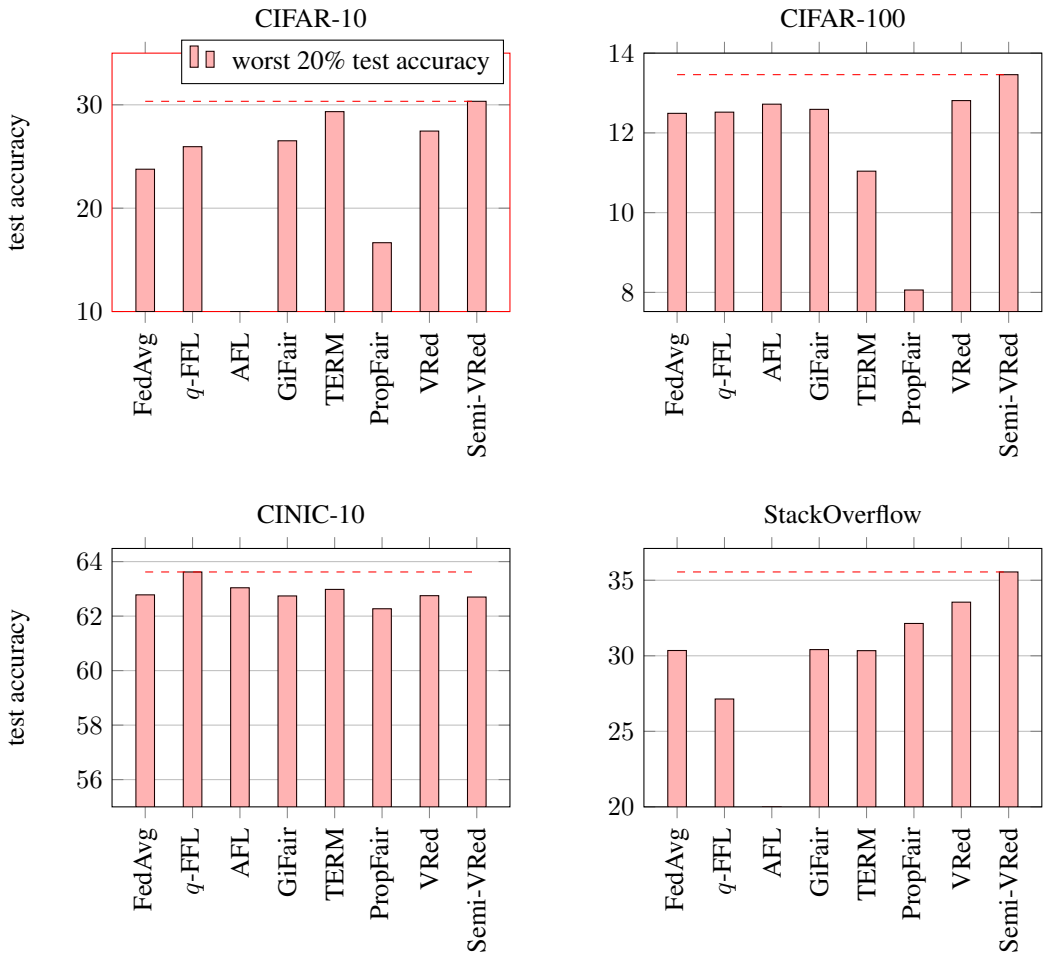

Figure 2: Worst 20% test accuracies for different algorithms. **top left:** CIFAR-10, **top right:** CIFAR-100, **bottom left:** CINIC-10, **bottom right:** StackOverflow. Due to divergence, results for `AFL` on CIFAR-10 and StackOverFlow are not shown. All subfigures share the same legends and axis labels.

Table 6: Comparison among federated learning algorithms on CIFAR-10, CIFAR-100, CINIC-10 and StackOverflow datasets with test accuracies (%) from clients. All algorithms are fine-tuned. **Mean**: the average test accuracy across all clients; **Std**: standard deviation of clients test accuracies; **Worst**: the worst test accuracy among clients; **Worst (10/20%)**: the worst 10/20% test accuracies of clients; **Best (10%)**: the best 10% test accuracies of clients.

Dataset	Algorithm	Mean	Std	Worst	Worst (10%)	Worst (20%)	Best (10%)
CIFAR-10	FedAvg, Ditto	$43.45_{\pm 0.60}$	$14.33_{\pm 0.62}$	$9.35_{\pm 3.13}$	$18.86_{\pm 0.99}$	$23.77_{\pm 0.70}$	$68.97_{\pm 0.81}$
	q-FFL	$45.46_{\pm 0.74}$	$14.31_{\pm 2.03}$	$18.71_{\pm 3.36}$	$21.23_{\pm 3.06}$	$25.95_{\pm 3.51}$	$72.31_{\pm 2.88}$
	AFL	-	-	-	-	-	-
	GiFair	$45.05_{\pm 0.64}$	$12.93_{\pm 0.44}$	$16.79_{\pm 3.55}$	$22.65_{\pm 2.03}$	$26.52_{\pm 0.76}$	$65.62_{\pm 2.59}$
	TERM	$\mathbf{45.61}_{\pm 1.03}$	$\mathbf{12.24}_{\pm 0.56}$	$13.80_{\pm 5.25}$	$24.89_{\pm 1.37}$	$29.34_{\pm 0.61}$	$68.65_{\pm 1.27}$
	PropFair	$36.95_{\pm 0.21}$	$15.16_{\pm 1.33}$	$1.14_{\pm 1.62}$	$12.49_{\pm 0.28}$	$16.66_{\pm 1.31}$	$66.04_{\pm 4.24}$
	VRed	$44.43_{\pm 0.88}$	$13.05_{\pm 1.32}$	$18.61_{\pm 3.12}$	$24.28_{\pm 2.22}$	$27.46_{\pm 1.56}$	$69.31_{\pm 3.48}$
	Semi-VRed	$45.47_{\pm 0.10}$	$12.58_{\pm 0.23}$	$\mathbf{19.04}_{\pm 6.73}$	$\mathbf{27.08}_{\pm 1.76}$	$\mathbf{30.34}_{\pm 1.05}$	$\mathbf{72.50}_{\pm 0.88}$
CIFAR-100	FedAvg, Ditto	$20.20_{\pm 0.31}$	$6.50_{\pm 0.21}$	$\mathbf{10.36}_{\pm 0.69}$	$11.07_{\pm 0.54}$	$12.49_{\pm 0.51}$	$33.88_{\pm 0.09}$
	q-FFL	$20.25_{\pm 0.11}$	$6.30_{\pm 0.27}$	$9.66_{\pm 0.33}$	$11.09_{\pm 0.67}$	$12.52_{\pm 0.46}$	$33.96_{\pm 0.90}$
	AFL	$18.98_{\pm 0.71}$	$\mathbf{4.91}_{\pm 0.37}$	$9.81_{\pm 0.69}$	$11.31_{\pm 0.18}$	$12.72_{\pm 0.21}$	$28.68_{\pm 1.71}$
	GiFair	$19.81_{\pm 0.32}$	$5.74_{\pm 0.16}$	$9.35_{\pm 0.34}$	$11.19_{\pm 0.24}$	$12.59_{\pm 0.49}$	$32.30_{\pm 0.32}$
	TERM	$18.00_{\pm 0.41}$	$6.05_{\pm 0.18}$	$8.86_{\pm 0.50}$	$10.02_{\pm 0.44}$	$11.04_{\pm 0.51}$	$31.58_{\pm 0.98}$
	PropFair	$14.97_{\pm 0.68}$	$6.44_{\pm 0.34}$	$5.40_{\pm 1.28}$	$7.00_{\pm 1.11}$	$8.06_{\pm 1.07}$	$28.89_{\pm 0.91}$
	VRed	$20.42_{\pm 0.36}$	$6.08_{\pm 0.05}$	$9.43_{\pm 1.01}$	$11.21_{\pm 0.74}$	$12.81_{\pm 0.85}$	$33.59_{\pm 1.11}$
	Semi-VRed	$\mathbf{20.85}_{\pm 0.39}$	$6.26_{\pm 0.18}$	$9.12_{\pm 1.47}$	$\mathbf{11.86}_{\pm 0.74}$	$\mathbf{13.46}_{\pm 0.63}$	$\mathbf{34.57}_{\pm 1.20}$
CINIC-10	FedAvg, Ditto	$86.26_{\pm 0.03}$	$15.20_{\pm 0.07}$	$50.48_{\pm 0.29}$	$56.87_{\pm 0.36}$	$62.78_{\pm 0.16}$	$100.0_{\pm 0.00}$
	q-FFL	$\mathbf{86.63}_{\pm 0.06}$	$\mathbf{14.88}_{\pm 0.08}$	$51.57_{\pm 0.82}$	$57.77_{\pm 0.36}$	$\mathbf{63.62}_{\pm 0.18}$	$\mathbf{100.0}_{\pm 0.01}$
	AFL	$86.45_{\pm 0.12}$	$15.10_{\pm 0.11}$	$51.57_{\pm 0.45}$	$57.58_{\pm 0.29}$	$63.04_{\pm 0.28}$	$100.0_{\pm 0.00}$
	GiFair	$86.28_{\pm 0.11}$	$15.20_{\pm 0.13}$	$49.66_{\pm 1.21}$	$56.97_{\pm 0.29}$	$62.74_{\pm 0.36}$	$100.0_{\pm 0.00}$
	TERM	$86.34_{\pm 0.04}$	$15.12_{\pm 0.01}$	$49.90_{\pm 0.42}$	$57.21_{\pm 0.11}$	$62.98_{\pm 0.04}$	$100.0_{\pm 0.00}$
	PropFair	$86.01_{\pm 0.17}$	$15.34_{\pm 0.19}$	$49.97_{\pm 1.23}$	$56.53_{\pm 0.65}$	$62.27_{\pm 0.55}$	$99.99_{\pm 0.01}$
	VRed	$85.79_{\pm 0.35}$	$15.02_{\pm 0.06}$	$51.57_{\pm 0.50}$	$57.66_{\pm 0.30}$	$62.75_{\pm 0.36}$	$99.98_{\pm 0.01}$
	Semi-VRed	$85.83_{\pm 0.33}$	$14.95_{\pm 0.07}$	$\mathbf{51.59}_{\pm 0.98}$	$\mathbf{58.00}_{\pm 0.21}$	$62.70_{\pm 0.14}$	$99.96_{\pm 0.01}$
StackOverflow	FedAvg, Ditto	$40.34_{\pm 0.06}$	$6.98_{\pm 0.03}$	$25.64_{\pm 0.11}$	$27.12_{\pm 0.06}$	$30.35_{\pm 0.03}$	$49.70_{\pm 0.07}$
	q-FFL	$37.79_{\pm 0.80}$	$7.38_{\pm 0.09}$	$22.54_{\pm 1.03}$	$24.12_{\pm 1.00}$	$27.14_{\pm 0.92}$	$47.06_{\pm 0.66}$
	AFL	-	-	-	-	-	-
	TERM	$40.34_{\pm 0.05}$	$6.96_{\pm 0.06}$	$25.56_{\pm 0.21}$	$27.12_{\pm 0.20}$	$30.41_{\pm 0.12}$	$49.76_{\pm 0.10}$
	GiFair	$40.34_{\pm 0.04}$	$6.97_{\pm 0.03}$	$25.71_{\pm 0.13}$	$27.10_{\pm 0.11}$	$30.34_{\pm 0.08}$	$49.71_{\pm 0.09}$
	PropFair	$41.76_{\pm 0.01}$	$6.80_{\pm 0.05}$	$27.30_{\pm 0.21}$	$28.75_{\pm 0.19}$	$32.14_{\pm 0.10}$	$50.76_{\pm 0.08}$
	VRed	$42.90_{\pm 0.05}$	$6.64_{\pm 0.01}$	$29.08_{\pm 0.09}$	$\mathbf{30.39}_{\pm 0.05}$	$33.55_{\pm 0.05}$	$51.66_{\pm 0.03}$
	Semi-VRed	$\mathbf{42.90}_{\pm 0.03}$	$\mathbf{6.60}_{\pm 0.01}$	$\mathbf{29.10}_{\pm 0.06}$	$30.34_{\pm 0.09}$	$\mathbf{35.55}_{\pm 0.05}$	$\mathbf{51.70}_{\pm 0.04}$