A Appendix

A.1 Extended Related Works

Federated Learning As stated earlier, federated learning is a learning paradigm where a centralized model is learnt from data distributed across clients without collecting the data at a single storage space. It has a variety of applications in the real world including but not limited to predictive health models [Brisimi et al., 2018], communication between vehicles [Samarakoon et al., 2018], learning words [**?**], and next-word prediction [Hardy et al., 2017]. FL involves several issues. The most important problems being heterogeneity in the data across clients [Karimireddy et al., 2021, Ghosh et al., 2021, Li et al., 2020] and communication bandwidth [McMahan et al., 2017, Rothchild et al., 2020] as mentioned earlier. Other problems in FL that are being focused on are fairness [Li et al., 2020, Mohri et al., 2019], protocols for federated learning settings [Aledhari et al., 2020], privacy and security [Bonawitz et al., 2016, Agarwal et al., 2018], and personalization [Dinh et al., 2021, Hanzely et al., 2020].

Algorithms The early algorithms on federated learning (Local SGD) involves each client undergoing 1 round of SGD (E=1) before parameter aggregation [Rosenblatt and Nadler, 2016]. The most common algorithms in federated learning is FedAVG [McMahan et al., 2017]. To reduce the communication rounds required to attain an ideal global model, they introduce multiple SGD steps on the local data before aggregation of client models. When the data across clients is heterogeneous, as each client advances towards their global optima before the aggregation, each client moves away from the optima of the global model to that of their local model. This is called "client drift" by Karimireddy et al. [2021]. They introduce control variates at the clients to reduce this drift from the global model at the previous round. The same has been described as "gradient bias " by Yao et al. [2020]. They propose a controllable meta update as well to the global model to tackle this problem. Li et al. [2020] introduces a proximal term at the client loss functions to limit this divergence of the client models by keeping the client model close to the global model. It also incorporates partial work at clients or "stragglers". Another major problem in FL is the heterogeneity in the number of client epochs. With different local iterations, the progress of clients towards their objective varies. This results in a bias when the client parameters are aggregated to approximate the global model. FedNova [Wang et al., 2020] introduces a normalized averaging of client gradients for the global model update. Reddi et al. [2021] introduces federated versions of adaptive optimizers ADAM, Yogi, ADAGrad. It introduces momentum parameters at the global model aggregation.

OOD Generalization The classic formulation of machine learning derived from statistical learning theory involves independently and identically distributed data samples. When this distribution differs in the test samples, it is considered to be "covariate shift". The idea of learning models to perform comparably on data from such distribution is described as "domain adaptation" or "out-of-distribution generalization". According to causality, there exists elements in data that connects causes and effects. They are expected to be invariant under varying external circumstances. Parascandolo et al. [2020] and many other OOD generalization works assumes that learning these invariant mechanisms enables learning a model robust to distribution shifts.

OOD generalization methods like Invariant Risk Minimization(IRM) [Arjovsky et al., 2020], Risk Extrapolation(REx) [Krueger et al., 2021], and gradient starvation by Spectral Decoupling [Pezeshki et al., 2020] involves a penalty weight to selectively learn the invariant features from all predictive features. Francis et al. [2021] has explore the idea of OOD Generalization in a split federated learning setting by applying IRM on the server model. The AND-Mask proposed by Parascandolo et al. [2020] uses a consistency score based on agreement across environments to selectively learn the invariant features. The agreement is based on the sign of the gradients. Shahtalebi et al. [2021] smoothens the AND-Mask to include the magnitude of the gradients so as to incorporate the blind-spots in the AND-Mask.

A.2 Gradient Masked SCAFFOLD

This section shows the gradient masked version of SCAFFOLD [Karimireddy et al., 2021].

Algorithm 2 Gradient Masked SCAFFOLD

ServerUpdate:

Initialize w_0
for each server epoch, t = 1,2,3,... **do**
  Choose C clients at random
  for each client in C, n **do**
    $w_t^n, \Delta_c^n = \text{ClientUpdate}(w_{t-1}, \Delta_c)$
    $\Delta_t^n = \frac{n_k}{\sum_{n=1}^{N} n_k}(w_t^n - w_{t-1})$
  end for
  $\Delta_t = \sum_{n=1}^{N} \Delta_t^n$
  $\Delta_c = \frac{1}{N}\sum_{n=1}^{N} \Delta_c^n$
  $mask = \tilde{m}_\tau(\{\Delta_t^n\}_{n=1..C})$
  $w_t = w_{t-1} - \eta_g * mask \odot \Delta_t$
end for

ClientUpdate(w, c):

Initialize $w_0 = w$
$c_i = c_i^+$
for each local client epoch, i=0,1,2,3,..,n **do**
  $g_i = \nabla_{w_i} L(w_i)$
  $w_{i+1} = w_i - \eta_c \, g_i - c_i + c$
end for
$c_i^+ = (i)g_i(x)$or $(ii)c_i - c + \frac{1}{K\eta_l}(x - y_i)$
return w_{i+1} , $c_i^+ - c_i$ to server

A.3 Proof of Lemma 1

Lemma 1 Let all clients, $L_{n\in N}$ have L-Lipschitz gradients and let the global learning rate, $\eta_g \leq \frac{1}{L}$. After k global iterations or communication rounds, gradient masked federated algorithms (algorithms using the proposed gradient masked averaging instead of arithmetic mean) visits at least one θ such that $||m_\tau \odot \nabla L(\theta^*)|| \leq O(\frac{1}{k})$.

Proof. Assume $\eta_g \leq \frac{1}{L}$ and consider a pseudo loss $\tilde{L}$ obtained by averaging the client loss functions as mentioned above. Assume all client loss functions, $L_{n\in N}$ have L-Lipschitz gradients. This imples, the pseudo loss will have L-Lipschitz gradients.

Assume that each client updates with full batch gradient descent towards the minima of its local objective.

Using Taylor expansion around θ,

$$L(\theta^{k+1}) \;\leq\; L(\theta^k) - \eta_g < \nabla L(\theta^t), m_\tau \odot \nabla L(\theta^t) > + \frac{L\eta_g^2}{2}||m_\tau \odot \nabla L(\theta^t)||^2$$

$$\leq L(\theta^k) - (\eta_g - \frac{L\eta_g^2}{2}) + \frac{L\eta_g^2}{2}||m_\tau \odot \nabla L(\theta^t)||^2$$

If we seek $\eta_g - \frac{L\eta_g^2}{2} \geq \frac{\eta_g}{2}$, then $\eta \leq \frac{1}{L}$ as assumed earlier.

$$L(\theta^{k+1}) - L(\theta^k) \leq \frac{\eta_g}{2}||m_\tau \odot \nabla L(\theta^t)||^2$$

Summing over k global model updates,

$$\sum_{t=0}^{k} \frac{\eta_g}{2} ||m_\tau \odot \nabla L(\theta^t)||^2 \leq L(\theta^0) - L(\theta^k) \leq L(\theta^0)$$

$$\Rightarrow min_{t=0,1,\cdots,k} \leq \frac{1}{k} \sum_{t=0}^{k-1} \frac{\eta_g}{2} ||m_\tau \odot \nabla L(\theta^t)||^2 \leq \frac{2L(\theta^0)}{\eta_g k}$$

Hence there exists an iteration such that $||m_\tau \odot \nabla L(\theta^t)||^2 \leq O(\frac{1}{k})$

A.4 Experiments

This section contains the details about the model used, datasets considered, hyperparameters, and other plots that were used for our experiments and discussions.

A.4.1 Model

All the experiments were performed on a CNN based architectures at the clients. These models consists of 2 convolution layers and 3 fully connected layers. The kernel size used is 5 for both convolutional layers. The models used for testing on MNIST, FMNIST, FeMNIST, CIFAR, and Rotated MNIST has 6 output channels for the first convolutional layer and 16 output channels for the second convolutional layer. The model used for testing on CMNIST has 32 output channels for the first convolutional layer and 16 output channels for the second convolutional layer. At the end of each convolution, a max pooling layer is used to reduce the size of feature maps. The output from the last maxpooling is then passed to 3 fully connected layers with dropout at the end. A dropout probability of 0.5 is used throughout. A ReLU activation function is used upon the fully connected layers as well. In addition to this, we used SGD as the optimiser for training the individual client models.

A.4.2 Datasets and Samples

IID and Non-IID We simulate IID distribution of data by assigning samples from all classes to each client. The same idea is used for the IID version of all datasets - MNIST, FMNIST, FEMNIST, CIFAR, CMNIST and RotMNIST. For the Non-IID distribution, we use the label distribution skew. That is, each client or party will have samples from 2 classes only. That is, a 2 label distribution skew. The tables below give the sample details. The label distribution skew ensures that the distribution varies across clients. It simulates the heterogeneity in the real world federated dataset.

Model, algorithms, and hyperparameters For all the datasets and algorithms, we fix a simple CNN model as used by [Li et al.]. We analyze gradient masking on non-adaptive optimizers like FedAVG [McMahan et al., 2017] and FedProx [Li et al., 2020], and adaptive optimizers like FedADAM and FedYogi [Reddi et al., 2021]. We also analyze the masking on SCAFFOLD. We compare the performance of these algorithms with the proposed mask to their original version without the mask.

For FedAdam and FedYogi, we fix the first momentum parameter $\beta_1 = 0.9$ and the second momentum parameter $\beta_2 = 0.99$ as per Reddi et al. [2021]. For uniformity across datasets and algorithms in in-distribution testing, we fix batch size, $B = 32$ and $\eta_l = 0.01$ for all clients and $\eta_g = 1$ for non-adaptive optimizers including SCAFFOLD [Li et al.] and $\eta_g = 0.1$ for adaptive optimizers [Reddi et al., 2021]. The hyperparameter τ is fixed by hyperparameter tuning on CIFAR-10 with respect to FedYogi and the same is used across all datasets and algorithms. $\tau \in [0, 1]$ represents the minimum majority fraction of gradients along the dominant direction or sign for the gradient parameter to be considered consistent across clients in FL. For example, $\tau = 0.5$ implies that there should be at least 50% more clients along the dominant direction than that along the submissive direction. Since a validation data is unavailable in a cross-device FL setting, we tune to the hyperparameters that have minimum average training loss over the last 20 rounds of training as done by Reddi et al. [2021]. Each algorithm is run for 500 communication rounds or until convergence, whichever comes first.

Table 3: This table shows the label distribution across clients for an IID setting. Each client will have randomly chosen examples from all 10 classes. This represent the 10 class setting in MNIST. We have considered 3 clients for the table. The same pattern would be present across all clients.

	0	1	2	3	4	5	6	7	8	9
Client 1	585	643	591	550	571	561	631	628	620	620
Client 2	589	691	593	628	553	526	588	640	602	590
Client 3	531	697	595	627	557	557	596	626	581	633
.....										

Table 4: This table shows the label distribution skew used by us for our experiments for a non-IID data distribution across clients. This represents the 10 class setting in MNIST. We have taken 3 clients. The same pattern would be present across all clients.

	0	1	2	3	4	5	6	7	8	9
Client 1	2894	2247								
Client 2		2246	1962							
Client 3				1962		1371				
.....										

Table 5: Average test performance (% accuracy) over the last 20 rounds of training under a FL setting where the data is distributed i.i.d across 10 clients. The results compare the algorithms to their gradient masked version on MNIST, FMNIST, CIFAR-10, and FEMNIST datasets. Across all datasets and algorithms, gradient masked aggregation had higher accuracy than their naive aggregation counterpart.

	FedAVG		**FedProx**		**SCAFFOLD**		**FedAdam**		**FedYogi**	
	AVG	GMA	AVG	GMA	AVG	GMA	AVG	GMA	AVG	GMA
MNIST	97.49	**98.94**	97.9	**98.93**	98.64	**98.92**	98.71	**99.24**	98.63	**99.3**
FMNIST	85.49	**89.5**	86.58	**89.48**	88.03	**89.65**	88.31	**89.14**	87.86	**89.47**
CIFAR-10	46.78	**61.15**	60.71	**60.78**	40.26	**58.16**	49.61	**58.68**	45.31	**61.12**
FEMNIST	97.71	**99.28**	98.27	**99.05**	97.88	**99.35**	99.13	**99.24**	99.16	**99.32**

Federated OOD datasets The federated OOD datasets we use is the Rotated MNIST similar to that used by Francis et al. [2021] in a split learning setting. Each digit is rotated at an angle from -90 to +90 degrees (except 45) and the test data contains samples with 45 degrees. The spurious mechanism here is the angle at which the data is rotated. Samples for the dataset is give in the figure 2. For the federated CMNIST dataset, we use the colored MNIST dataset used by Arpit et al. [2019]. We distribute the dataset across client as per the IID and Non-IID distribution sampling mentioned above. The spurious mechanism here is the foreground and background colors. there are four each per class. These spurious mechanisms at the train will not be present at the test. Train and test samples are given in Figure 3.

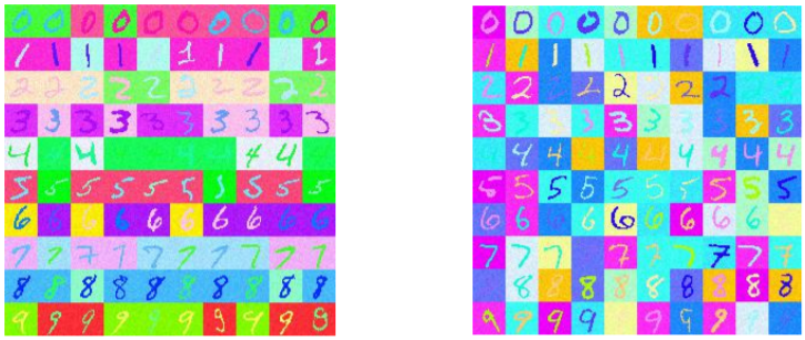
A.4.3 Communication Rounds

Communication rounds is the number of times the global model is undated. At each communication round, the client models send their model parameters to the server where they are aggregated to approximate the global model. These global parameters are then send to the clients. It was observed that gradient masking converged faster than naive averaging in IID data distributions as well. However, it was less significant compared to non-IID setting.

(a)

(b)

Figure 2: Samples from Rotated MNIST (a) Train data. (b) Test data.



(a) (b)

Figure 3: Samples from FedCMNIST (a) Train data. (b) Test data. [Arpit et al., 2019]

A.4.4 Effect of τ

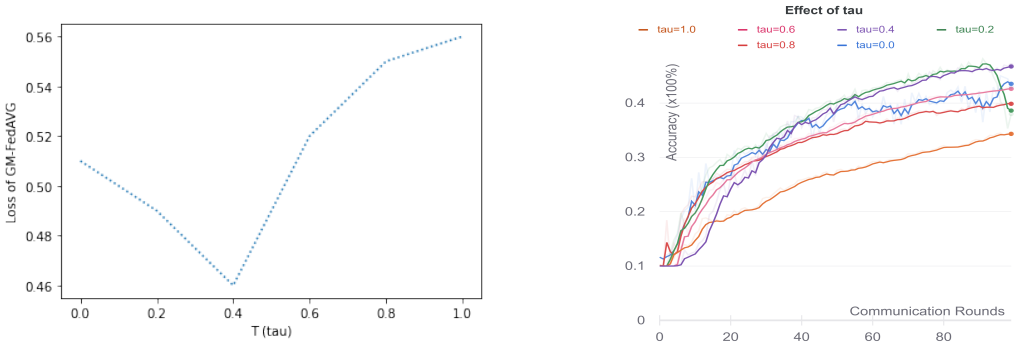
τ is a hyperparameter introduced to threshold agreement across clients. Throughout the experiments we have maintained τ at 0.4. This was obtained from fine-tuning CIFAR-10 using FedYogi optimizer with the gradient masking for 10 clients on the simple CNN model explained above. $\tau = 0.4$ implies that the gradient being considered have a 40% excess or a total of 60% of the client gradients along the dominant direction. From our experiments it was observed that when τ is low, the model underfits with a high test accuracy. It reduces with an increasing τ until 0.4. With further increasing τ, the model overfits increasing the test accuracy again. Precise fine-tuning can help improving the performance further. Furthermore, when $\tau = 0$, gradient masked FedAVG becomes equal to FedAVG with all clients being selected for averaging with weight as 1.

Table 6: Number of communication rounds taken by the algorithms and their gradient masked counterpart to attain a commendable accuracy (as specified beside the dataset) on MNIST, FMNIST, CIFAR-10, and FEMNIST datasets when the data is distributed non-i.i.d across clients. Gradient masking reduces the number of communication rounds in most cases.

	FedAVG		**FedProx**		**SCAFFOLD**		**FedAdam**		**FedYogi**	
	AVG	**GMA**	**AVG**	**GMA**	**AVG**	**GMA**	**AVG**	**GMA**	**AVG**	**GMA**
MNIST (95%) (non-i.i.d)	37	**25**	51	**31**	>100	>100	63	**19**	35	**20**
FMNIST (75%) (non-i.i.d)	118	**28**	98	**74**	>100	>100	**19**	31	**25**	34
CIFAR-10 (45%) (non-i.i.d)	>300	**132**	>300	**111**	>300	>300	>300	**150**	>300	**248**
FEMNIST (90%) (non-i.i.d)	20	**13**	40	**14**	78	**13**	21	**12**	17	**12**

Table 7: Average test performance (% accuracy) over the last 20 rounds of training under a FL setting with the data distribution mentioned alongside the dataset name in the table. The results compare the algorithms to their gradient masked version on FedCMNIST and FedRotMNIST datasets. Note that across all datasets and algorithms, gradient masked aggregation had higher accuracy than their naive aggregation counterpart.

	FedAVG		**FedProx**		**SCAFFOLD**		**FedAdam**		**FedYogi**	
	AVG	**GMA**	**AVG**	**GMA**	**AVG**	**GMA**	**AVG**	**GMA**	**AVG**	**GMA**
MNIST (i.i.d) (97%)	12	**7**	11	**5**	10	**4**	9	9	9	**7**
FMNIST (i.i.d) (85%)	40	**18**	30	**11**	29	**9**	**15**	23	**19**	22
CIFAR-10 (i.i.d) (50%)	95	**65**	35	**25**	30	**25**	90	**38**	>100	**38**
FEMNIST (i.i.d) (97%)	**6**	7	4	4	6	**4**	16	**4**	17	**4**



(a) (b)

Figure 4: (a) Test loss vs. τ/. This plot further confirms the need to fine tune the hyperparameter τ. The underfitting and overfitting of the model with varying τ can be observed in this plot. To further improve the performance, τ can be fine-tuned precisely. (b) Test accuracy of FedYogi with gradient masking on CIFAR-10 for 10 clients across different values of $\tau \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ with 100 communication rounds. It can be noted that the test accuracy is the best when $\tau = 0.4$