Table 6: EMNIST model.

Layer	Output Shape	Parameters
Input	$(28, 28, 1)$	0
Conv2d	$(28, 28, 32)$	832
MaxPool2d	$(14, 14, 32)$	0
Conv2d	$(14, 14, 64)$	51264
GroupNorm	$(14, 14, 64)$	128
MaxPool2d	$(7, 7, 64)$	0
Flatten	3136	0
Dense	512	1606144
Dense	62	31806

A Datasets

CIFAR-10 The CIFAR-10 dataset [Krizhevsky et al., 2009] consists of images with 3 channels of 32 x 32 pixels each. Each pixel is represented by an int8. This dataset contains 50,000 training examples, and 10,000 test examples. The 10 possible labels are equally represented among the images (with 6,000 images per label). While this dataset does not have a natural partition by client, we follow [Reddi et al., 2020], and use the approach described in [Hsu et al., 2019] to federate it. We randomly partition the training data into 500 clients, each receiving 100 examples. We apply Latent Dirichlet Allocation over the labels, and each client draws a multinomial distribution over the labels from a symmetric Dirichlet distribution with parameter α. This method results in an IID split when $\alpha \to \infty$, while each client tends to draw a single label when $\alpha \to 0$. We set $\alpha = 1$ in our experiments. We preprocess the images by randomly cropping each channel of the training images to shape $(24, 24)$, and randomly flipping them horizontally. The test images are centrally cropped to the same dimensions. We normalize the pixel values by centering them, and dividing them by their standard deviations.

EMNIST We use a federated version of the EMNIST dataset, where the characters are partitioned by their authors [Caldas et al., 2019a]. There are 62 possible characters, split among 3,400 clients.

Stack Overflow The Stack Overflow dataset contains the body text of questions and answers from the Stack Overflow website. This data is available for 342,477 training clients with 135,818,730 examples, 38,758 validation clients with 16,491,230 examples, and 204,088 test clients with 16,586,035 examples. It is naturally partitioned into clients by the post authors. The train clients only have examples timestamped before 2018-01-01 UTC, and the test clients examples from after 2018-01-01 UTC. The validation clients are held out from both train, and test sets. Users with less than 100 training examples are filtered out. We limit our vocabulary to the 10,000 most frequent words, and map the other words to an out-of-vocabulary bucket. We restrict each client to the first 256 sentences. We pad and truncate each sentence to ensure it has a length of 20 words.

B Models

We use the CNN model described in [McMahan et al., 2017a] to train EMNIST, adding a group normalization layer after the second convolution. A full description of this model is given in table 6. We freeze the first dense layer in our experiments with the corresponding partially trainable model.

We train ResNet-18 on CIFAR-10, where we replaced the batch normalizations by group normalizations, following the observation by [Hsieh et al., 2020] that this change results in better behavior in non-idd settings. To partially train this model, we freeze the convolutional layers of the residual blocks in increasing order.

We perform next word prediction on the Stack Overflow dataset using a three layer Transformer architecture [Vaswani et al., 2017]. The token embeddings have a dimension set to 96, and the hidden dimension of the feed forward network (FFN) block to 2,048. The multi-head attentions have 8 heads, each based on 12-dimensional key, query, value vectors. The activation used is ReLU, and the

Table 7: Experiment details.

Task	Training rounds	Clients per round	Batch Size	Client Optimizer
EMNIST	1500	20	16	SGD
CIFAR-10	1500	10	128	SGDM
SO NWP	5000	32	16	Adam

Table 8: DP-FTRL experiment details.

Task	Training rounds	Report Goal	Batch Size
SO NWP	1600	100	16

dropout rate is set to 0.1 during training. The partially trainable models are derived by freezing the hidden layers of the FFN.

C Experiment Details

We run our experiments using the Tensorflow Federated framework [Ingerman and Ostrowski, 2019], on GPUs. We conduct 5 runs for each setting, with different random seeds, except in our DP-FTRL experiments, which we ran only once. We present the average final accuracies, $\pm$ the standard deviations over these runs. For each experiment, we use the SGD optimizer for the server updates.

We summarize more experimental details in tables 7 and 8.

C.1 Hyperparameters

We present the hyperparameters used in our experiments.

Non DP experiments Table 9 details the optimization parameters we use in all our experiments, except for the DP-FTRL ones. The server momentum value used in CIFAR-10 is 0.9.

DP-FTRL experiments The DP-FTRL experiments were run on Stack Overflow only. We used the SGD optimizer on the clients, and SGD with a momentum value of 0.9 on the server. We set the clipping norm to 0.3. We performed a grid search on the learning rates for each noise multiplier value, and report the best accuracies. The grid search was performed on the following values:

$$\eta_{client} \in \{10^{-1.5}, 10^{-1.0}, 10^{-0.5}\}$$
$$\eta_{server} \in \{10^{-1.5}, 10^{-1.0}, 10^{-0.5}, 10^{0.0}, 10^{0.25}\}$$

C.2 Frozen/Trainable parameters

We now detail the layers that were frozen in each set of experiments.

EMNIST The majority of the parameters in the model we use to do character recognition on EMNIST is contained in the dense layer following the convolutional blocks. We freeze this layer when experimenting with a partially trained version of that model.

Table 9: Optimizers.

Task	Server Optimizer	Client Learning rate	Server learning rate
EMNIST	SGD	0.05	0.5
CIFAR-10	SGDM	$10^{-0.5}$	$10^{-1.0}$
SO NWP	Adam	0.1	0.03

Table 10: Frozen ResNet-18 convolutional blocks.

Trainable parameters (%)	Frozen convolutional blocks
100	None
26.25	0
8.07	$0, 1$
3.47	$0, 1, 2$
2.16	$0, 1, 2, 3$

Table 11: Partially frozen encoder layers in SO NWP.

Trainable parameters (%)	Partially frozen encoder blocks
100	None
91.3	2
82.6	$1, 2$
73.8	$0, 1, 2$

CIFAR-10 We vary the percentage of trainable parameters in the Resnet-18 model we train on CIFAR-10 by increasing the number of frozen convolutional blocks, as shown in table 10.

Stack Overflow The Transformer architecture we use in our Stack Overflow Next Word Prediction experiments contains three encoder layers. We vary the proportion of trainable parameters by freezing the first layer of the FFNs of these encoders, as detailed in table 11.

C.3 Runtimes

We report the average training time per round from our TFF simulation experiments. We filtered out aberrant values by removing any value deviating by more than one standard deviation of the average in one run.