A Appendix

A.1 Evolution of Invariant Neurons

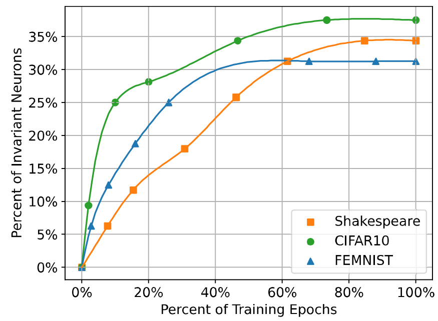


Figure 6: The percentage of 'invariant' neurons as the number of training rounds vary

In this section, we provide an example that some neurons in the server are trained quickly and vary only below a threshold in later iterations. Figure 6 shows the percentages of invariant neurons as the number of training epochs increases. Even after only 30% of the training rounds are completed, 15%-30% of the neurons become invariant across CIFAR10, FEMNIST, and Shakespeare datasets For this example, we choose thresholds of 180%, 10%, and 500%, respectively, for these three datasets and compute their invariant neurons. Sending invariant neurons over to the straggler provides no utility; therefore, these neurons can be dropped. Our work FLuID builds upon this insight.

A.2 Choosing Suitable Threshold

Each model has different characteristics in terms of the magnitude of neuron updates. Therefore, choosing different threshold values results in a different number of neurons classified as invariant. We expanded on our initial findings and studied the effect of threshold value on the number of invariant neurons during training. As expected, a higher threshold value leads to a higher percentage of invariant neurons. Table 3 presents the percentage of invariant neurons observed at different threshold values, and the overall training accuracy of the FEMNIST model, using a sub-model size of 0.75 for the stragglers.

Table 3: Threshold vs accuracy results

Threshold value (%)	**Percentage of Invariant Neurons (%)**	**Accuracy (%)**
1	3	80.1
3	6	80.3
5	13	80.5
7	18	80.7
8	22	80.7
10	31	80.5

We observe that to obtain the desired accuracy and mitigate performance bottlenecks of stragglers, it is critical to choose the threshold that has the closest number of invariant neurons as the number of neurons to be dropped for the sub-model. The FLuID framework can automatically tune the threshold for the desired model based on the straggler performance, as described in Section 5.

A.3 Impact of Sub-Model Size on Training Time

In this section, we present evidence that there is a linear relationship between client training time and sub-model size. We evaluated the training time of 5 Android-based mobile phones from 2018 to 2020 outlined in Table 1. The training time is expressed in the percentage of the training time for the

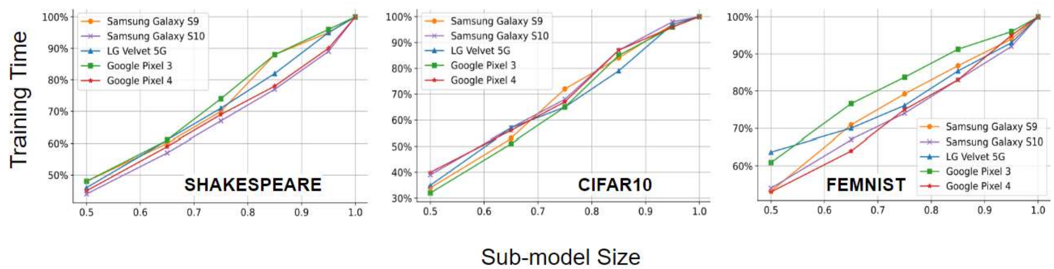


Figure 7: Linear relationship between training time and model size across all three datasets

full model size ($r = 1.0$). Across CIFAR10, FEMNIST, and Shakespeare, the training time of all five mobile clients decreases linearly as the sub-model size decreases and falls within 10% of the sub-model size. Using this insight, FLuID selects a sub-model size r as the available sub-model, the size that's closest to the inverse of $Speedup$.

A.4 Additional Experiments with Scalability Studies.

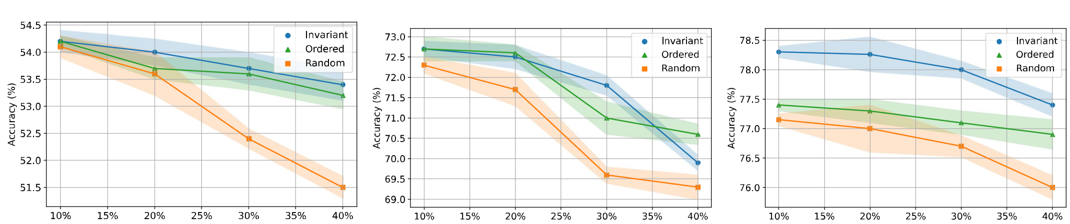
In this section, we show that in the case that the network has multiple stragglers, FLuID does not assume that all stragglers have similar capabilities or select a sub-model for all stragglers based on the slowest device. FLuID can select sub-model sizes for each straggler client based on each client's own capabilities. In this experiment, we cluster devices of similar capabilities into four groups of sub-model sizes. Table 4 presents the accuracy when stragglers are assigned to 4 equal-sized clusters (sub-model size 0.65, 0.75, 0.85, 0.95). The overall accuracy generally lies between assigning sub-model sizes of 0.75 and 0.85 for all stragglers. This way, FLuID can achieve a higher training accuracy with a shorter training time, even with stragglers that are initially more than 35% slower than non-straggler devices.

Table 4: Accuracy comparison of Random Dropout, Ordered Dropout, and Invariant Dropout as we cluster stragglers into different sub-model size groups.

	Random	Ordered	Invariant
CIFAR10	71.7	72.3	72.7
FEMNIST	77.5	77.4	78.2
Shakespeare	53.8	53.9	54.1

A.5 Additional Experiments with Varying Straggler Percentages

In this section, we show that FLuID is capable of handling multiple ratios of stragglers. We have run additional experiments to explore the impact of different ratios of stragglers. One common trend we observed across state-of-the-art techniques and FLuID is that accuracy decreases as the ratio of stragglers is increased as more of the devices are now being trained only on the sub-model. Nonetheless, in all the cases, Invariant Dropout offers the highest accuracy because it is aware of the neuron gradient changes and only drops the least changing neurons. The accuracy results of varying the straggler ratios while using 0.75-sized sub-models are summarized in Figure 8.



(a) Shakespeare - LSTM (50 Clients) (b) CIFAR10 - VGG9 (100 Clients) (c) FEMNIST - CNN (100 Clients)

Figure 8: Accuracy of varying the straggler ratios from 10% to 40% with 0.75 sub-model size

A.6 Scalability of FLuID with Client Sampling

As the federated learning system scales, FL servers sample a subset of clients participating in each training round. At any point in training, FLuID is capable of recalibrating stragglers and supports dynamic changes during runtime. The ability to adjust to a different set of clients and identify stragglers in every training round enables FLuID to easily incorporate client sampling into its process. We scaled FLuID to 1,000 clients with the FEMNIST dataset for 500 global training rounds. We run with a client sampling ratio of 10%, as used by the prior works in federated learning spaces such as FjORD [HLA+21]. We present the accuracy results in Table 5 against each sub-model size for Invariant Dropout and the baseline techniques. Invariant Dropout maintains a better accuracy profile than the baselines even when scaled up to 1000 clients while incorporating client sampling.

Table 5: Accuracy comparison of Random Dropout, Ordered Dropout, and Invariant Dropout as for FEMNIST with 1000 clients and client sampling of 10%.

	r=0.95	r=0.85	r=0.75	r=0.65	r=0.40
Random	87.9	87.5	87.5	86.9	85.7
Ordered	87.8	88.0	87.5	87.3	87.0
Invariant	88.1	88.2	88.0	87.7	87.2