G SUPPLEMENTARY MATERIALS

G.1 LEARNING RATE TREND UNDER DIFFERENT DATA HETEROGENEITY

Heterogeneity Index	Batch Size / Number of Data Points			
	5/400	10/400	20/800	30/800
0.15	0.021	0.04	0.041	0.061
0.50	0.032	0.065	0.071	0.105
0.75	0.042	0.086	0.081	0.125

Table 2: Optimal learning rate under different data heterogeneity levels, batch sizes, and data sizes.

Here we demonstrate how the learning rate for the clients varies across data distribution and quantity. We observe that with a constant number of data points and batch size, increasing heterogeneity (i.e. *HI*) leads to higher optimal learning rate. Similarly, with a constant *HI*, increasing number of data points and batch size leads to increasing learning rate.

G.2 TRAINING AND PROXY DATASET SETUP

Dataset	**Model**	**Train/Test split**	**Clients Total/Per Round**	**Global LR /Batch Size**	**Training Rounds**
FEMNIST	2 conv 2 dense	49,644/6,200	192/10	0.004/8	2000
Cifar100	Resnet18	50,000/10,000	50/5	0.045/16	1000
Cifar10	4 conv 2 dense	50,000/10,000	50/5	0.05/16	500
F-MNIST	2 conv 2 dense	50,000/10,000	50/5	0.002/8	500

Table 3: Training Setup.

We perform our experiments using the popular image classification datasets Cifar100, Cifar10 and Fashion-MNIST. We also use the widely used FEMNIST dataset, which is a handwritten digit and character image classification dataset made specifically for benchmarking Federated Learning applications. It contains 62 classes and around 800,000 images split into 3,550 clients. We sample from it using the seed and sample found in their official repository in github[1]. Since it does not have a separate evaluation dataset, we use the same setup in Caldas et al. (2018) and derive a balanced test dataset of size 6,200 by randomly sampling 100 datapoints per class from the unused datapoints.

The set of proxy datasets are uniformly sampled from their full training datasets. Note that this sampled dataset is removed from the full dataset. Therefore, all proxy datasets have no overlap with either training nor testing datasets. Proxy datasets in FEMNIST, CIFAR10, CIFAR100, and FashionMNIST contain 5000, 5000, 5000, and 4000 samples, respectively. The remaining training datasets (after removing the sampled proxy datasets for training) are 44664, 45000, 45000, and 46000 samples, respectively.

We control the different data distributions within each client by splitting them into groups and subgroups. We first create 6 groups of clients by splitting them equally (e.g. in FEMNIST, 192 clients are split into 32 clients per group), and assign each of these devices to get 100/200/400/600/800/1000 data points respectively. We further split these groups into 4 more evenly split subgroups (e.g. in FEMNIST, 32 clients get split into 4 groups of 8). These groups are then assigned *HI*s of 0.2, 0.4, 0.6 and 0.8.

G.3 ROBUSTNESS TO DATA HETEROGENEITY METRICS

Apart from *HI*, state-of-the-art papers also use other methods of quantifying heterogeneity such as *Gaussian* and *Possion* distribution sampling (for both quantity and distribution heterogeneity) Reddi et al. (2020); Zawad et al. (2021). To demonstrate that *FedTune* works with other distribution metrics, we compare the accuracy increase we get after applying *FedTune* compared to Global Tuning. We do this across the heterogeneity types *HI*, *Gaussian*, and *Poisson*, and the results are presented in Figure 6.

[1] https://github.com/TalwalkarLab/leaf/

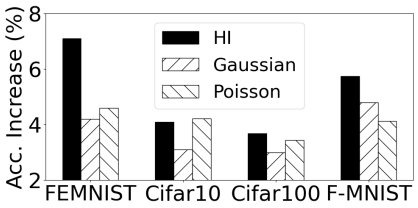

Figure 6: Final test accuracy comparison between global, *FedTune* with transferred dataset and *FedTune* on original dataset.

We observe that across all datasets, *FedTune* results in varying degrees of accuracy improvement for all different types of data heterogeneity metrics. This demonstrates that our framework is robust to different types of data distribution metrics.

G.4 Compatibility with Other Heterogeneity-aware FL Optimization

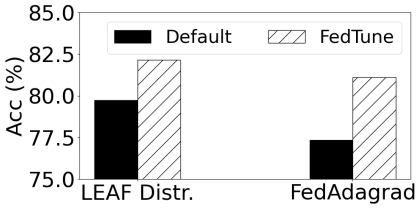

Figure 7: Final test accuracy comparison between global and *FedTune* when using LEAF's (Caldas et al. (2018)) default distribution (*LEAF Distr.*) and when used with and without *FedTune*.

We perform additional experiments to demonstrate *FedTune* is compatible with other state-of-the-art heterogeneity-aware optimizations in FL. In Figure 7, the first set of bars show the comparison of test accuracy at convergence between global tuning (*Default*) and *FedTune* when using LEAF's default distribution. We observe that using our customized hyperparameter tuning can achieve an accuracy improvement of around 2.3%. The second set of bars show the change in accuracy when using FedAdagrad (Reddi et al. (2020)) by itself (*Default*) versus adding *FedTune* on top of it (*FedTune*). We observe that with the help of *FedTune*, the final accuracy is improved around 4%, confirming that *FedTune* and FedAdagrad are complementary to each other and can be combined to achieve an even better performance.

G.5 Sensitivity Analysis of HRT Size against accuracy

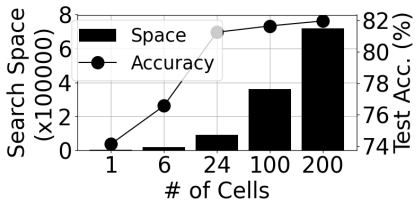

Figure 8: Sensitivity analysis of the hyperparameter search space as a function of HRT cells against test accuracy and cost using FEMNIST.

Figure 8 shows the results of the sensitivity analysis of how the number of cells in the HRT impacts the search space. For example, 1 cell means that only one set of hyperparameters is used to train the full system, i.e., a global tuning set. As we increase the number of cells, there is a drastic increase in the total search space, making it expensive to tune. Figure 8 shows how the final test accuracy for FEMNIST changes with varying number of HRT cells for our approach. It is clear

that the benefits of increasing the number of cells after 24 diminish greatly while the search space keeps on increasing. Thus, in our experiments, an HRT with 24 cell blocks strikes a good balance between search cost and accuracy. Specifically, we use HIs of $0.2, 0.4, 0.6, 0.8$ and data quantities of $100, 200, 400, 600, 800, 1000$ in our evaluation.