Appendix

A Additional Experiments

A.1 Hyperparameter Sensitivity

To study how sensitive the hyperparameter tuning process is to different degrees of non-identicalness in FL settings, we perform experiments on CIFAR-10/100 datasets with a grid of hyperparameters.[1] Following [2], we define the effective learning rate for `FedAvgM` as $\eta_{\text{eff}} = \eta/(1-\beta)$. For all values of Dirichlet concentration α, we sweep over learning rate $\eta_{\text{eff}} \in \{10^{-3}, 10^{-2.5}, \ldots, 10^{0}\}$ and momentum $1-\beta \in \{10^{-2.5}, 10^{-2}, \ldots, 10^{0}\}$.

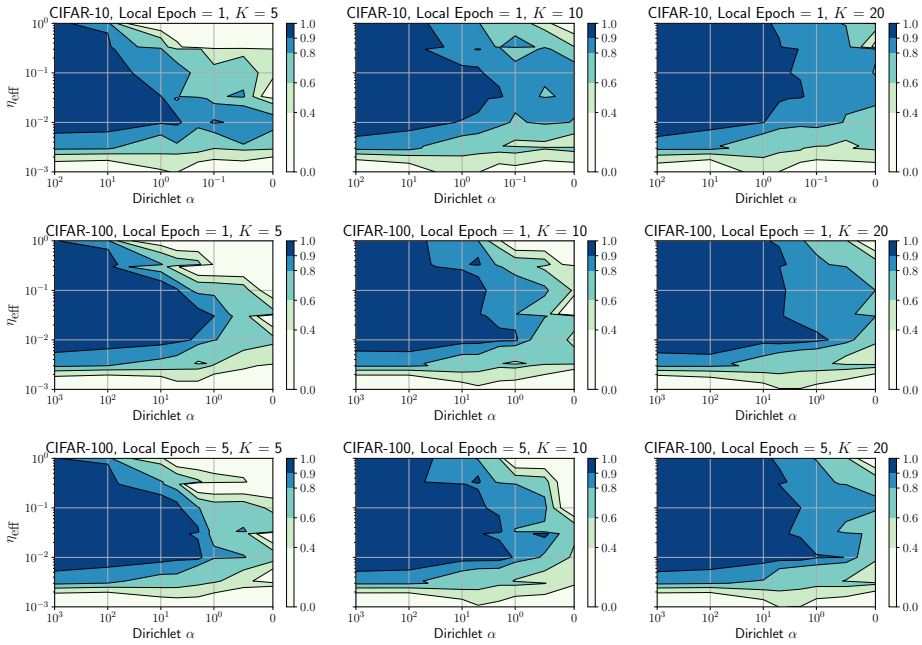

Fig. 1: **Relative Accuracy of `FedAvgM` on CIFAR Datasets.** Darker shades denote regions of higher relative accuracy. $\eta_{\text{eff}} = \eta/(1-\beta)$ is the effective learning rate, and K is the reporting goal out of 100 clients. Note that data split is increasingly non-identical to the right.

In Figure 1 we show the effect of using different η_{eff} on the relative accuracy with each grid point showing the best result over all (β, η) combinations that give the same η_{eff}.

[1] CIFAR experiments in the main text are tuned over the the same grid.

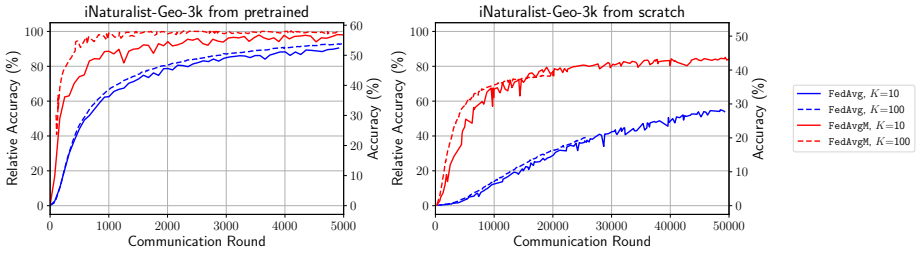

Fig. 2: **Learning Curves from ImageNet Pretraining and from Scratch.** On the left vertical axis is the relative accuracy while on the right is the absolute accuracy. Two plots are rescaled to have the full span of 100% relative accuracy.

We train for 10k/20k communication rounds with CIFAR-10/100 respectively.

Within each individual contour plot, it can be seen that the accuracy consistently drops with increased non-identicalness, and the set of hyperparameters yielding high performance becomes smaller. In general, we find an effective learning rate $\eta_{\text{eff}} = 10^{-2}$ works well in many situations.

Across different report goals K, a larger K enables good performance over a wider range of η_{eff}. This result is unsurprising, since with more clients reporting in, the server observes more data and hence obtains gradients with less variance. The number of local epochs does not affect the choice of hyperparameters much in our experiments (see last two rows of Figure 1). Interestingly, while CIFAR-10 and CIFAR-100 have different numbers of classes and centralized learning accuracy, they exhibit very similar characteristics in terms of *relative accuracy* (the overall shape of plots in Figure 1 is similar).

A.2 The Effect of Pretraining

Pretraining large visual models (e.g., using ImageNet) is very common in centralized training. It is likely to be even more beneficial in federated settings, where extra computation rounds could be prohibitively time consuming. In some cases, however, it may be necessary or desirable to train from scratch. In this section, we investigate the feasibility of training large federated visual classification models without pretraining[2].

We perform experiments using iNaturalist-Geo-3k with a combination of settings including the FL algorithm (`FedAvg`/`FedAvgM`) and report goal K. Since training from random initialization and from pretrained weights converge to different final test accuracy, we use *relative accuracy* for evaluating FL algorithms' progress relative to the corresponding centralized learning upperbounds.

[2] Note that in the main text, the smaller CIFAR10/100 experiments are trained from scratch, but the larger iNaturalist and Landmarks experiments use an ImageNet pretrained MobileNetV2.

Table 1: **Communication Rounds to Reach Relative Accuracy.** Note that models have different centralized learning accuracy (51.4% from scratch and 57.9% from pretrained). The multipliers are calculated row-wise, using Rounds@10% as the baseline. Experiments that do not reach the target relative accuracy even after t rounds is marked $> t$.

Data	Method	Initialization	K	Rounds@Relative Accuracy		
				10 %	50 %	90 %
Geo-3k	`FedAvg`	pretrained	10	165 (1.0×)	669 (4.1×)	4912 (29.8×)
	`FedAvg`	pretrained	100	165 (1.0×)	567 (3.4×)	3780 (22.9×)
	`FedAvgM`	pretrained	10	79 (1.0×)	249 (3.2×)	1505 (19.1×)
	`FedAvgM`	pretrained	100	**60** (1.0×)	**116** (1.9×)	**420** (6.9×)
	`FedAvg`	scratch	10	9005 (1.0×)	39236 (4.4×)	> 50k
	`FedAvg`	scratch	100	7793 (1.0×)	> 20k	> 20k
	`FedAvgM`	scratch	10	1463 (1.0×)	5788 (4.0×)	> 50k
	`FedAvgM`	scratch	100	977 (1.0×)	3733 (3.8×)	> 20k

From Figure 2, we see that FL with pretraining requires orders of magnitude fewer communication rounds for convergence and yields higher final relative accuracy than training from scratch. Table 1 further shows the rounds needed to reach 10%, 50%, and 90% relative accuracy. We see that `FedAvgM` is able to accelerate convergence significantly, with a report goal $K = 100$ it takes 94% (977 $\rightarrow$ 60) fewer rounds than `FedAvg` to reach 10% relative accuracy when starting from pretrained model weights. We also see that `FedAvgM` has a much steeper learning curve, reaching 90% relative accuracy in 6.9× the rounds needed to reach 10% (compared to 20× for `FedAvg`).

Whilst our results suggest that it is possible to train large federated visual classification models from scratch, doing so efficiently and effectively remains an open challenge with room for improvement.

B CIFAR-10/100 Dataset Details

B.1 Synthetic Clients with Dirichlet Prior

To generate non-identical client datasets from CIFAR-10 and CIFAR-100 [1] datasets, we partition each into 100 clients, with 500 training examples each. We assume every client k has their data independently drawn from the original dataset according to a multinomial distribution $q_k(\cdot)$ of C classes ($q_k(y) \geq 0$ and $\sum_y q_k(y) = 1$).

To synthesize a population of non-identical clients, we draw a multinomial $\boldsymbol{q}_k \sim \mathsf{Dir}(\alpha \boldsymbol{p})$ from a Dirichlet distribution, where $\boldsymbol{p}$ describes a prior class distribution over C classes, and $\alpha > 0$ is a parameter controlling the *concentration*, or identicalness among all clients. α can be used to control the overall homogeneity: $\alpha \rightarrow \infty$ generates clients that are all identical to the prior $\boldsymbol{p}$, while $\alpha \rightarrow 0$ generates clients that tend to hold very sparse labels. After drawing the class distributions $\boldsymbol{q}_k$, for every client k, we sample training examples from CIFAR-10/100 for each class according to $\boldsymbol{q}_k$ *without replacement*. This is to ensure there are no overlapping examples between any two clients.

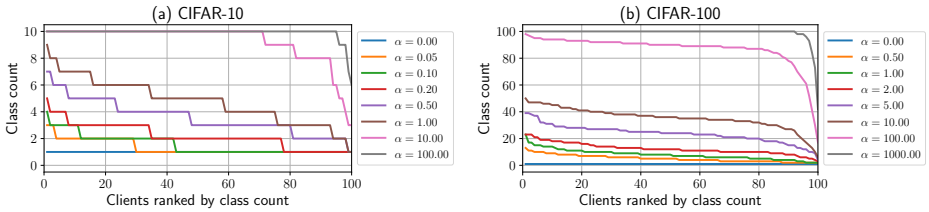

Fig. 3: **CIFAR-10/100 Distribution.** Each curve represents the class counts of clients within a data partitioning synthesized using a Dirichlet concentration parameter α.

Note that by drawing examples without replacement, towards the end of the assignment process, some subset $\mathcal{S}$ of classes can be exhausted earlier than other classes, ending up with a shorter list of available classes from which the client synthesis procedure can continue drawing samples. When this happens, we eliminate $\mathcal{S}$ and enforce the remaining clients to only sample from classes $\{1, 2, \ldots, C\} \setminus \mathcal{S}$ with a multinomial distribution

$$\tilde{q}_k(y) = \begin{cases} 0, & y \in \mathcal{S} \\ q_k(y) \,/\, \left(1 - \sum_{s \in \mathcal{S}} q_k(s)\right), & y \notin \mathcal{S}. \end{cases} \tag{1}$$

For CIFAR-10, we use $\alpha \in \{100, 10, 1, 0.5, 0.2, 0.1, 0.05, 0\}$; for CIFAR-100 we use $\alpha \in \{1000, 100, 10, 5, 2, 1, 0.5, 0\}$. Summary statistics showing the class count over the client population in both datasets is given in Figure 3.

C Experiment Run Time

The federated learning experiments are carried out by simulation with a cluster of NVIDIA Tesla P100 GPUs in parallel. The experiment run time, while highly variable depending on the experimental setup (model complexity, dataset, local steps E, and reporting clients per round K), is roughly 0.5 to 2.0 seconds per communication round per reporting client. This amounts to about 9 GPU-days for a run of Landmarks-User-160k experiment for 5000 rounds with $K = 100$.

References

1. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009) 3
2. Shallue, C.J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., Dahl, G.E.: Measuring the effects of data parallelism on neural network training. arXiv preprint arXiv:1811.03600 (2018) 1