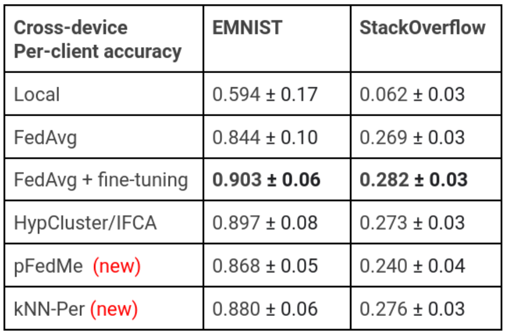
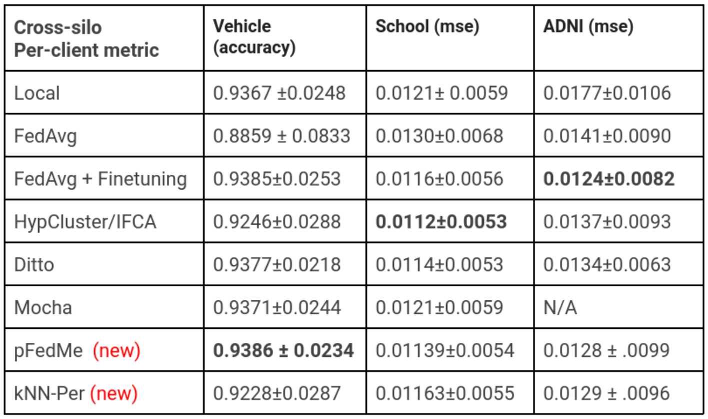
Extra experiments



Cross-device Per-client accuracy	EMNIST	StackOverflow
Local	0.594 ± 0.17	0.062 ± 0.03
FedAvg	0.844 ± 0.10	0.269 ± 0.03
FedAvg + fine-tuning	**0.903 ± 0.06**	**0.282 ± 0.03**
HypCluster/IFCA	0.897 ± 0.08	0.273 ± 0.03
pFedMe (new)	0.868 ± 0.05	0.240 ± 0.04
kNN-Per (new)	0.880 ± 0.06	0.276 ± 0.03



Cross-silo Per-client metric	Vehicle (accuracy)	School (mse)	ADNI (mse)
Local	0.9367 ±0.0248	0.0121± 0.0059	0.0177±0.0106
FedAvg	0.8859 ± 0.0833	0.0130±0.0068	0.0141±0.0090
FedAvg + Finetuning	0.9385±0.0253	0.0116±0.0056	**0.0124±0.0082**
HypCluster/IFCA	0.9246±0.0288	**0.0112±0.0053**	0.0137±0.0093
Ditto	0.9377±0.0218	0.0114±0.0053	0.0134±0.0063
Mocha	0.9371±0.0244	0.0121±0.0059	N/A
pFedMe (new)	**0.9386 ± 0.0234**	0.01139±0.0054	0.0128 ± .0099
kNN-Per (new)	0.9228±0.0287	0.01163±0.0055	0.0129 ± .0096

Figure 1: We will add two more stateless algorithms to the revised version of our paper: **pFedMe** (Dinh et al., "Personalized Federated Learning with Moreau Envelope", NeurIPS 2020), a regularizer-based algorithm (aka mean-regularized MTL); and **kNN-Per** (Marfoq et al., "Personalized Federated Learning through Local Memorization", ICML 2022), a very recent kNN-based algorithm that shows good performance compared to previous methods. Methods with the best mean per-client metric are highlighted (note that School and ADNI uses MSE metric, so the lower the better). Similar to Table 2 and Table 3 in our paper, the standard deviation is computed across the clients, which can be viewed as a **fairness** metric.

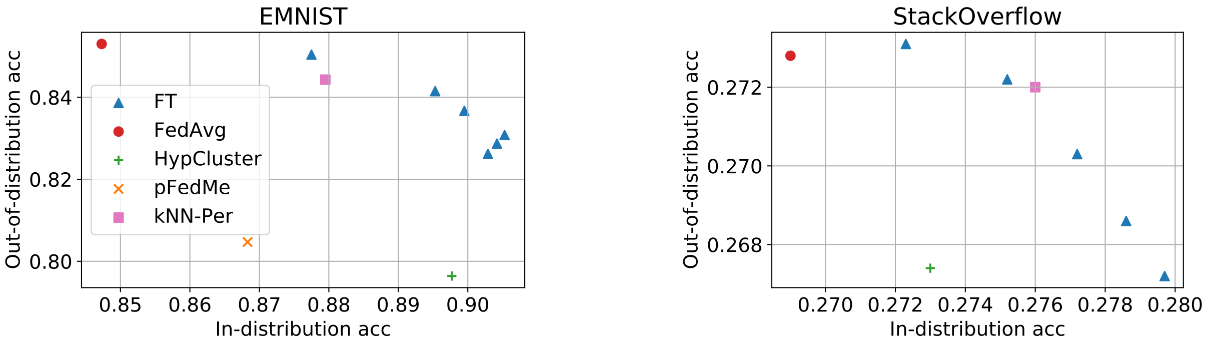
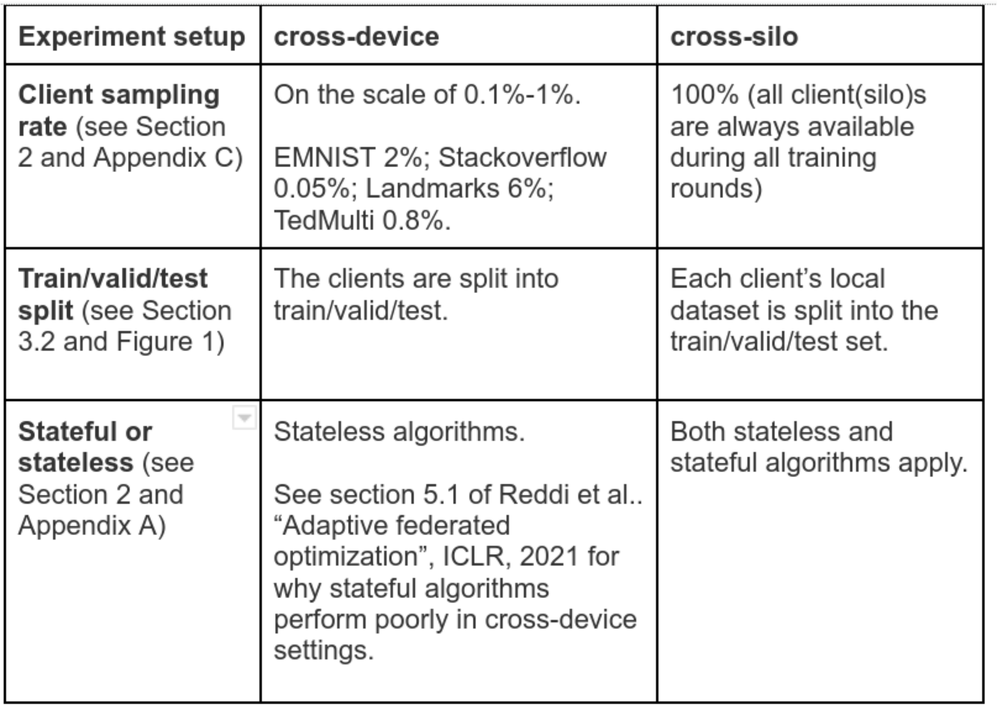
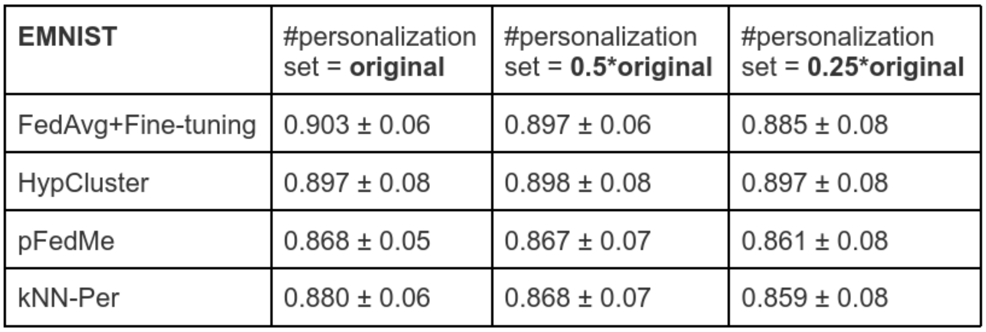


Figure 2: We performed additional experiments to illustrate the sensitivity to distribution shifts issue (aka catastrophic forgetting). Specifically, we evaluate each **test** client's personalized model over the in-distribution test set (i.e., this client's original local evaluation set) and the out-of-distribution test set (which contains examples sampled from the global distribution across all clients). pFedMe results on the StackOverflow dataset is 0.24 (in-distribution) and 0.21 (out-of-distribution), which is significantly worse than others, and hence, is omitted in the figure. For FedAvg+Fine-tuning (FT), we plot the results for FT epochs $1, 3, 5, 10, 15, \ldots$. Increasing the FT epochs usually gives better in-distribution accuracy but lower out-of-distribution accuracy.



Experiment setup	**cross-device**	**cross-silo**
Client sampling rate (see Section 2 and Appendix C)	On the scale of 0.1%-1%. EMNIST 2%; Stackoverflow 0.05%; Landmarks 6%; TedMulti 0.8%.	100% (all client(silo)s are always available during all training rounds)
Train/valid/test split (see Section 3.2 and Figure 1)	The clients are split into train/valid/test.	Each client's local dataset is split into the train/valid/test set.
Stateful or stateless (see Section 2 and Appendix A)	Stateless algorithms. See section 5.1 of Reddi et al.. "Adaptive federated optimization", ICLR, 2021 for why stateful algorithms perform poorly in cross-device settings.	Both stateless and stateful algorithms apply.

Figure 3: One of our contributions is to clearly separate the two federated settings in our experiments: cross-device and cross-silo. As mentioned in our paper and also summarized in the above table, these two settings affect three things: 1) clients sample rate per round; 2) how to split the data into the training/validation/test set; and 3) which algorithms are appropriate.



EMNIST	#personalization set = **original**	#personalization set = **0.5*original**	#personalization set = **0.25*original**
FedAvg+Fine-tuning	0.903 ± 0.06	0.897 ± 0.06	0.885 ± 0.08
HypCluster	0.897 ± 0.08	0.898 ± 0.08	0.897 ± 0.08
pFedMe	0.868 ± 0.05	0.867 ± 0.07	0.861 ± 0.08
kNN-Per	0.880 ± 0.06	0.868 ± 0.07	0.859 ± 0.08

Figure 4: We performed an ablation study on EMNIST by reducing the number of samples used to personalize the model. Recall that each **test** client has two local datasets: a personalization set and an evaluation set. The personalization set is used to: fine-tune the model for FedAvg+Fine-tuning; select the best (among k models) model for HypCluster; learn a regularized model for pFedMe; and find the nearest neighbors for kNN-Per. We reduce the size of the personalization set but keep the evaluation set unchanged. As shown in the above table, while every personalization algorithm's accuracy is decreasing with smaller personalization set, HypCluster's accuracy is pretty robust to this change. The main reason is that HypCluster only uses the personalization set to select the best model while the other algorithms use it to learn a new model.