REDUCING FORGETTING IN FEDERATED LEARNING WITH TRUNCATED CROSS-ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

In Federated Learning, a global model is learned by aggregating model updates computed from a set of client nodes, each having their own data. A key challenge in federated learning is the heterogeneity of data across clients whose data distributions differ from one another. Standard federated learning algorithms perform multiple gradient steps before synchronizing the model, which can lead to clients overly minimizing their local objective and diverging from other client solutions, particularly in the supervised learning setting. We demonstrate that in such a setting, individual client models experience the "catastrophic forgetting" phenomenon with respect to other client data. We propose a simple yet efficient approach that modifies the cross-entropy objective on a per-client basis such that classes outside a client's label set are shielded from abrupt representation change. Through extensive empirical evaluations, we demonstrate that our approach can greatly alleviate this problem, especially in the most challenging federated learning settings with high heterogeneity, low participation, and large numbers of clients.

1 INTRODUCTION

Federated Learning (FL) is a distributed machine learning paradigm in which a shared global model is learned from a decentralized set of data located at a number of independent client nodes McMahan et al. (2017), Konečnỳ et al. (2016). Due to communication constraints, federated learning algorithms typically perform many local gradient update steps before synchronizing. In realistic settings client data often have non-iid distributions, creating additional challenges in FL training. Client drift, a phenomenon in which client solutions severely "drift" from an optimal global solution following multiple local update steps Karimireddy et al. (2020), is one such problem. Approaches to address this inherent challenge often take the form of modifications to existing optimization algorithms to more effectively achieve the objective (Kairouz et al., 2021).

Continual Learning (CL) McCloskey & Cohen (1989) is another emerging research area that studies a learner being presented with a sequence of tasks. In a manner similar to how FL clients have their own non-iid data (often with different class distributions), different tasks in supervised continual learning typically contain data drawn from different distributions. In CL the problem of catastrophic forgetting is typically a focus of study since after learning a new task we want the model to retain its knowledge of previous tasks. A number of methods have been proposed in the literature to combat this problem.

In the case of supervised learning, we can draw a connection between the catastrophic forgetting problem and client drift. Consider one "round" of federated learning, in which C random clients are selected and sent a copy of the current global model. Each client then performs a number of local update steps to optimize the objective on their local data. A round ends with an update to the global model achieved by the aggregating the updates from each client. In a typical round, clients receive a model that has been previously derived from training on other clients data. However, as local training proceeds, the model becomes increasingly biased towards a given client, and will experience a catastrophic forgetting with respect to other data from other clients, which is drawn from distinctly different distributions. Naturally, aggregating models that have deviated from a joint solution (compatible with all clients) is more likely to lead to degraded results with respect to the global objective.

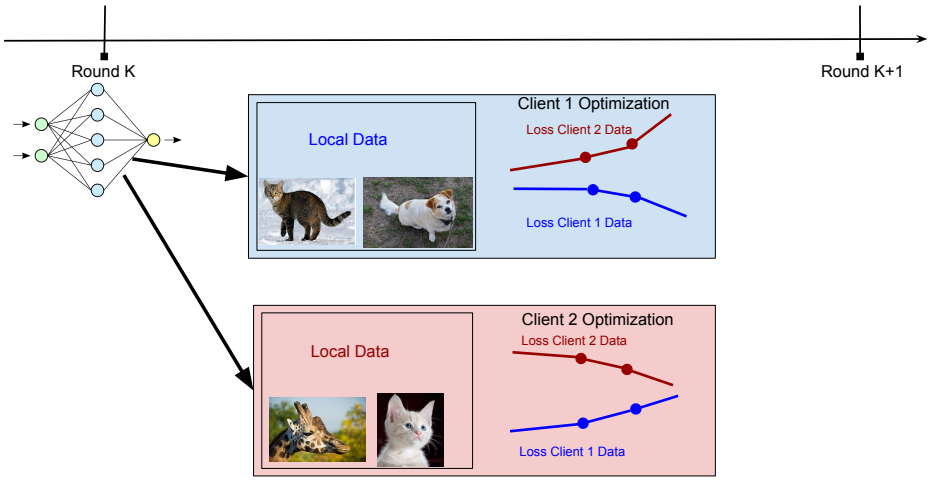


Figure 1: Illustration of the catastrophic forgetting problem within a round of Federated Learning on heterogeneous data. A global model with knowledge of all classes is sent to clients which increase loss on their local data distribution but tend to simultaneously decrease performance on other clients data. This leads to poor aggregation and overall performance. Mitigating catastrophic forgetting at the client level can lead to improved performance in such settings.

We denote the above problem as *local client forgetting*. Reducing client forgetting would moderate the increase in loss with respect to other clients for individual client models and thereby improve an individual client models loss over the combined data. Therefore, tackling local client forgetting can reduce client drift. We can consider invoking the vast literature of methods for controlling catastrophic forgetting which can then allow us to reduce client drift. Although a great deal of methods have been proposed Kirkpatrick et al. (2017); Li & Hoiem (2017); Chaudhry et al. (2019); Schwarz et al. (2018); Davari et al. (2022) they have are largely impractical in the FL setting. Experience Replay methods Chaudhry et al. (2019) would require access to other clients data, violating the primordial data communication constraints of FL. Similarly, many regularization methods such as EWC require communicating additional information and moreover typically require many steps to converge Aljundi et al. (2019) due to the additional conflicting objectives. This computational constraint can hurt convergence of the FL algorithm, a key desiderata. For the supervised continual learning setting Caccia et al. (2022); Ahn et al. (2021) proposed a modification of the standard cross entropy objective function that truncates the softmax denominator, removing terms corresponding to classes from old tasks. This simple approach allows to mitigate catastrophic forgetting by reducing the bias on the model to avoid predicting old classes.

In federated learning, as local client optimization proceeds for many gradient steps, optimizing the terms in the cross entropy corresponding to classes not present in that client's data distribution will quickly enforce not present classes to be forgotten by the local model. Furthermore, as discussed in Lesort (2022), it can cause spurious features to emerge. Inspired by these observations and the correspondence to the continual learning case, we can consider adapting the solution of Caccia et al. (2022); Ahn et al. (2021) to FL.

In this work we thus propose a simple approach for heterogeneous federated learning which corrects each clients loss function based on its class distribution. We show that this approach can drastically reduce client level forgetting in the heterogeneous setting and lead to substantially improved overall global model convergence and final performance that is also more robust to optimization hyperparameters and normalization layers.

2 Related Work

Federated Learning The most commonly used baseline in federated learning is the FedAvg algorithm proposed by McMahan et al. (2017). Communication costs between two nodes is orders

of magnitude larger than communication costs between processor and memory on the same node, making communication efficiency in federated learning of upmost importance Konečnỳ et al. (2016). FedAvg reduces communication costs by allowing clients to train multiple iterations successively. Convergence of FedAvg has been widely studied for both i.i.d Stich (2018), Wang & Joshi (2018) and non i.i.d settings Li et al. (2020). Under non i.i.d settings, convergence deteriorates as a function of increasing heterogeneity Hsu et al. (2019). Reddi et al. (2020) studied approaches for introducing adaptive algorithms into the server updates to accelerate FL algorithms.

Non-Heterogeneous Data Partitions and Client Drift One significant challenge encountered when training on decentralized data is heterogeneity of samples across clients. Realistically, partitions contain data generated under different conditions which can reasonably be expected to create different local distributions at each client. For example, smartphones containing images of sailboats will be more concentrated in coastal regions than in desert regions. Data at each client are sampled from these local distributions creating different local objectives. When clients progress too far towards minimizing their own objectives, local models drift apart, degrading the performance of the shared global model and slowing down convergence(Yao et al., 2021; Li et al., 2019; Karimireddy et al., 2020). Several attempts have been made to alleviate client drift through various methods. One approach centers around knowledge distillation to regulate local training, Zhu et al. (2021) and Lin et al. (2020) ensemble information about the global data distribution and disseminate it to clients via models trained at the server. These methods possess the added risk of privacy attacks and while Zhu et al. (2021) take steps to mitigate this risk, their method requires the existence of an unlabeled dataset which may not be practical in all settings. Other approaches attempt to regularize updates at the client level. Karimireddy et al. (2020) propose SCAFFOLD, an algorithm to control client drift by use of control norms which modify the client gradients. Li et al. (2020) add a proximal term to the local objective to limit the impact of variation in local updates. Tenison et al. (2022) propose a gradient masking technique that modifies the aggregation of updates on the server side. These approaches are aimed at modifying the federated optimization to improve communication efficiency, convergence, and robustness to heterogeneity. On the other hand our proposal in this work is a modification of the loss function on a per-client basis, and takes advantage of the structure of the commonly used cross-entropy loss. Since it modifies the objective functions locally, this approach is compatible with any federated optimization method in the literature.

Continual Learning Continual learning (CL) is a process by which tasks are learned sequentially over a period of time. The learner retains knowledge of previous tasks and leverages that prior knowledge to learn new tasks Chen & Liu (2018). CL is made difficult by the fact that neural networks suffer from catastrophic forgetting, in which learning a new task overrides weights learned from past training, thus degrading model performance on previously learned tasks McCloskey & Cohen (1989). Several families of methods have been developed to mitigate the catastrophic forgetting. In the first class of methods, architecture based approaches Schwarz et al. (2018) that attempt to grow or modify an architecture over time to expand its knowledge. In the second class of methods approaches which store some subset of old data for "rehearsal" are applied Lopez-Paz & Ranzato (2017); Chaudhry et al. (2019); Rebuffi et al. (2017). Finally, a third class of methods considers regularization Kirkpatrick et al. (2017).

A federated continual learning setting has been considered in the literature Yoon et al. (2021). Here each client in the federated network continuously collects data. Our work on the other hand considers the standard FL setting where each client maintains a fixed set of data and draws connections to a notion of forgetting across clients to motivate a modification of the loss function. Shoham et al. (2019) have used ideas from continual learning to propose FedCurv based on the EWC algorithm Kirkpatrick et al. (2017) from continual learning. FedCurv requires sending additional information and is not compatible with all FL methods. Along this line Xu et al. (2022) also proposed an approach inspired from rehearsal methods, generating pseudo data and adding an additional regularization term. This requires an expensive pseudo data generating procedure and can increase local training time.

Normalization in FL Batch normalization (BN) is a commonly employed machine learning tactic that normalizes features by the mean and variance computed across a mini batch. It has been shown to help with generalization and stabilize optimization (Ioffe & Szegedy, 2015). A limitation of BN is that under certain conditions, the mean and standard deviation used at test time may differ

significantly from those used in training. Scenarios including small batch sizes or non-i.i.d batch distributions such as those in heterogeneous FL have been noted to suffer from significant performance degradation when using BN (Reddi et al., 2020). Group normalization (GN) is not dependent on batch statistics, and is effective at mitigating the effect of model performance degradation induced by skewed data partitions and small batch sizes (Hsieh et al., 2020; Wu & He, 2018). Recent works using FedAvg and any of its variants typically avoid using BN (Diao et al., 2020).

3 Background and Methods

Federated optimization refers to the optimization problem implicit to federated learning Konečnỳ et al. (2016). In federated optimization, training data is distributed and optimization occurs over K clients with each client $k \in 1, ..., K$ having data $\mathbf{X}_k$ drawn from distribution D_k. We define $n_k = |\mathbf{X}_k|$ and $n = \sum_{k=1}^{K} n_k$ for n samples. The data $\mathbf{X}_k$ at each node may be drawn from different distributions and/or may be unbalanced with some clients possessing more training samples than others. The typical objective function for federated optimization is given by

$$\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}(\mathbf{w}, \mathbf{X}_k), \tag{1}$$

with $\mathcal{L}(\mathbf{w}, \mathbf{X}_k)$ measuring client k's local objective, and $\mathbf{w}$ representing the global parameters. In this work we will restrict ourselves to the common case where $\mathcal{L}$ is the cross entropy loss. There are many possible variations of FL algorithms. In general, they follow the same structure as FedAvg McMahan et al. (2017), which proceeds as follows :

- **client selection:** for a set of K clients, $K * C$ are selected at each round $\{t_i\}_{i=1}^{T}$, where $0 < C \leq 1$ is a pre-determined fraction.
- **client updates:** At the beginning of round, client models are initialized with the current weights of the server model. Each client selected for the round performs E local iterations of SGD.
- **server update:** The weights of the individual client models are aggregated to form an update to the shared global model.

Truncated Cross-Entropy Consider a neural network $f : \mathcal{R}^D \to \mathcal{R}^C$ where C is the total number of classes. The standard cross entropy is given $\mathcal{L}_{CE}(\mathbf{X}_k, \mathbf{Y}_k, \mathbf{w}) = -\sum_{\mathbf{x}\in\mathbf{X}} \log \frac{\exp(f_{\mathbf{w}}(\mathbf{x})_{y(\mathbf{x})})}{\sum_{c\in\mathcal{C}} \exp(f_{\mathbf{w}}(\mathbf{x})_c)}$. Here $y(\mathbf{x})$ is the label of $\mathbf{x}$ and $\mathcal{C}$ is the set of all classes available to the clients. We now consider the truncated cross entropy

$$\mathcal{L}_{TCE}(\mathbf{X}_k, \mathbf{Y}_k, \mathbf{w}) = -\sum_{\mathbf{x}\in\mathbf{X}} \log \frac{\exp(f_{\mathbf{w}}(\mathbf{x})_{y(\mathbf{x})})}{\sum_{c\in\mathcal{C}(\mathbf{Y}_k)} \exp(f_{\mathbf{w}}(\mathbf{x})_c)}, \tag{2}$$

where the denominator is a function of the labels for the client data $\mathbf{Y}_k$. Specifically, in our work we consider $\mathcal{C}(\mathbf{Y}_k)$ to be the set unique labels for the client. Intuitively, aggressively optimizing $\mathcal{L}_{CE}(\mathbf{X}_k, \mathbf{Y}_k)$ through multiple gradient steps during a client round can lead to a drastic increase in $\mathbb{E}_{x,y\sim D_{j\neq k}}[l_{CE}(\mathbf{x}, \mathbf{y})]$ where D_j are the distribution of clients other than client k. The truncated cross-entropy approach, on the other hand, modifies the original local objective function to avoid excessive pressure that drives up the loss of other client data. Indeed, classes not present at the current client are ignored by the local optimization. This in turn forces each client to learn by adapting the model's internal representation of the classes present in its training data, rather than abruptly shifting representations of classes outside its training set Caccia et al. (2022). We will demonstrate empirically in the sequel that this leads to a reduction in one notion of forgetting defined below.

Local client forgetting We formalize the notion of local client forgetting discussed in Sec 1 for classification problem. Denoting the accuracy on a client k's local test data $Acc_k(\mathbf{w})$, with $\mathbf{w}$ the model parameters. We can consider the local client forgetting $F_{ki} = Acc_k(\mathbf{w}_t^i) - Acc_k(\mathbf{w}_{t-1})$. Here $\mathbf{w}_t^i$ refers to the model of client i at round t (before the aggregation step) and $\mathbf{w}_{t-1}$ the global model at the end of round $t-1$. Furthermore, we can define an average forgetting for a client k's model $F_k = \frac{1}{K-1} \sum_{i\neq k} F_{ki}$. In the sequel we will study these quantities for a standard FL setting.

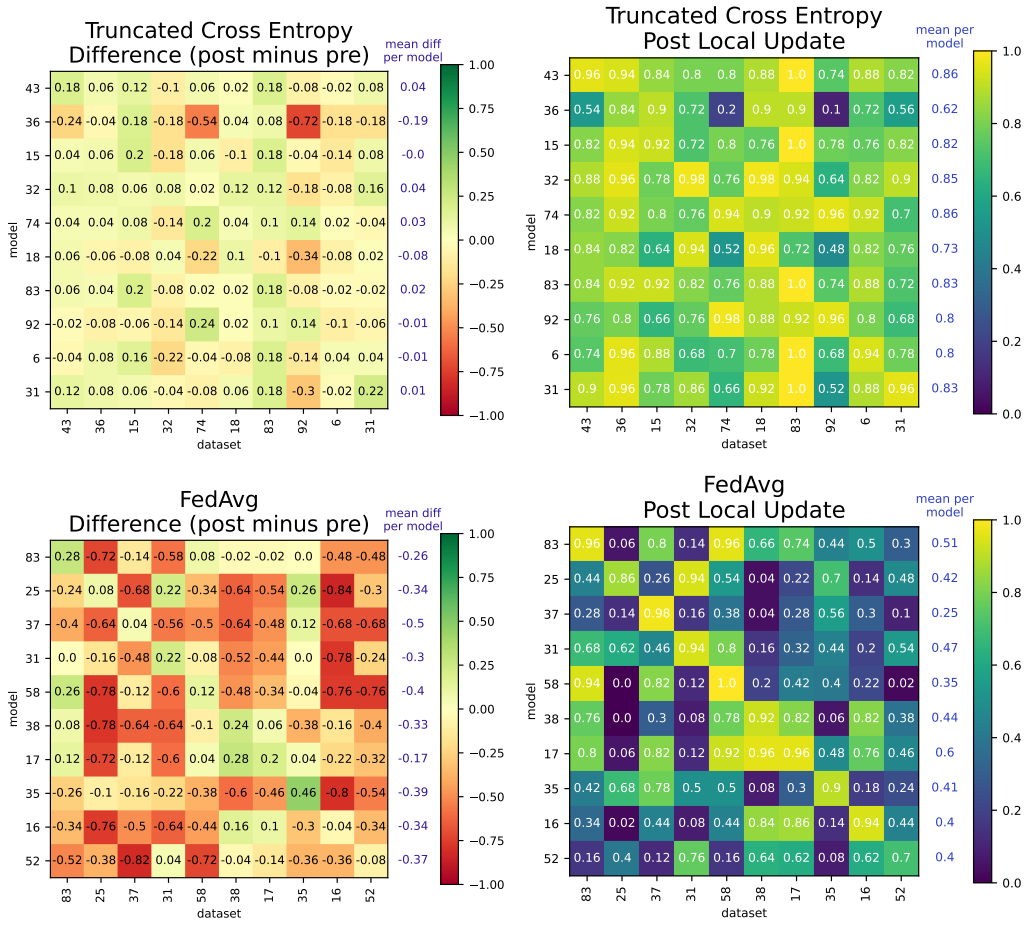


Figure 2: We show the local client forgetting for a given round with and without TCE. Prior to a local training round, each client model is an identical copy of the server model. After local training, local models have diverged according to their own local objectives. Each clients model is evaluated on its own dataset and the datasets of each other client selected for the round both before and after local training. On the left set of maps, the x and y axes contain the indices of each client selected for the round and the value F_{ik} in the heat map indicates the forgetting of the model of the k^{th} client evaluated on the i^{th} client's dataset. The final column gives F_k, the average forgetting over all clients. On the right we also show the accuracy of each client's model on the other client's data. We note that in some cases the accuracy can completely collapse on other client's data (particularly when they don't overlap in any classes). We see that TCE (top) significantly reduces forgetting across clients.

4 Experiments

In this section, we present the empirical results for the TCE framework. We start by analyzing the notion of forgetting in the context of standard FedAVG, and then show how the application of the Truncated Cross-Entropy objective can substantially resolve this. Subsequently, we study how TCE can enhance standard FL algorithms like FedAVG, improving their overall performance as well as making them more robust to key hyperparameters, namely the learning rate and the choice of normalization technique.

Datasets and Data Partitions We utilize CIFAR-10, CIFAR-100 Krizhevsky & Hinton (2009) and FEMNIST Caldas et al. (2018) datasets for our experiments, each of these datasets come pre-separated into training and testing sets. Our primary evaluations consider 100 clients and each client requires their own training and validation sets according to their own unique distribution.

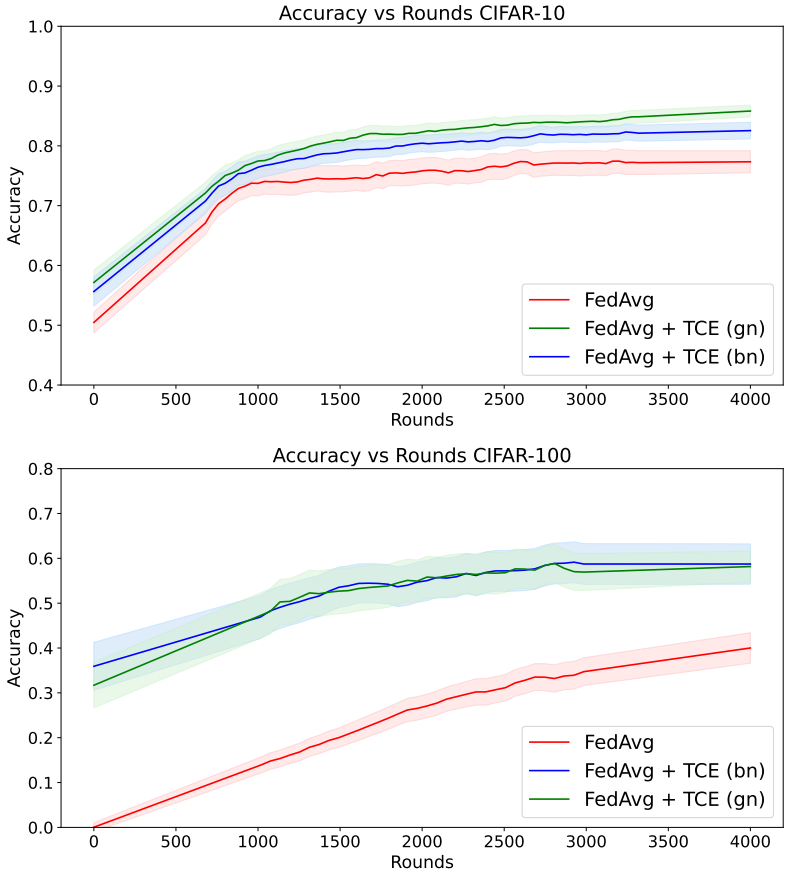


Figure 3: We show the performance over rounds in a highly heterogenous setting with FedAVG, FedAVG+TCE. We observe that TCE consistently gives the best performance. For FedAVG we use group norm in our models as batchnorm performs poorly in the FL setting. On the other hand TCE allows group norm and batchnorm variants to perform equally well.

To facilitate this, the entire training set is separated into equally sized non-i.i.d partitions using the Dirichlet distribution parameterized $\alpha = 0.1$, similar to the method of Hsu et al. (2019). These client partitions are then further separated into training (90%) and validation (10%) sets for each client. For example, 100 clients being trained using CIFAR-10 which contains 50 000 training samples would each have 500 of these training samples. Of those 500 samples, 450 would be used for local model updates and 50 would be used exclusively for validation.

Settings Clients are sampled without replacement for each round but can be selected again in subsequent rounds. The fraction of clients sampled is 10% for CIFAR-10 and FEMNIST datasets and 1% for CIFAR-100. In each case our primary evaluations train a ResNet-18 for over 4000 communication rounds for 3 local epochs, and a mini-batch of size 64. We use SGD as our optimizer, with weight decay of 1×10^{-4} following Yao et al. (2021), Hsu et al. (2019). Experiments done without truncated cross entropy (FedAvg) replace batch normalization with group normalization Hsieh et al. (2020), experiments using truncated cross entropy are run twice, once with batch normalization and once with group normalization.

Validation Throughout the training process the global model is evaluated periodically on the aggregation client validation sets to gauge overall training progress. A model is evaluated on the training set only once, after the completion of the entire 4000 rounds of training. This value is the accuracy reported in the validation statistics. At lower values of α, as client distributions become more skewed, there can be significant changes in accuracy between training runs Hsu et al. (2019).

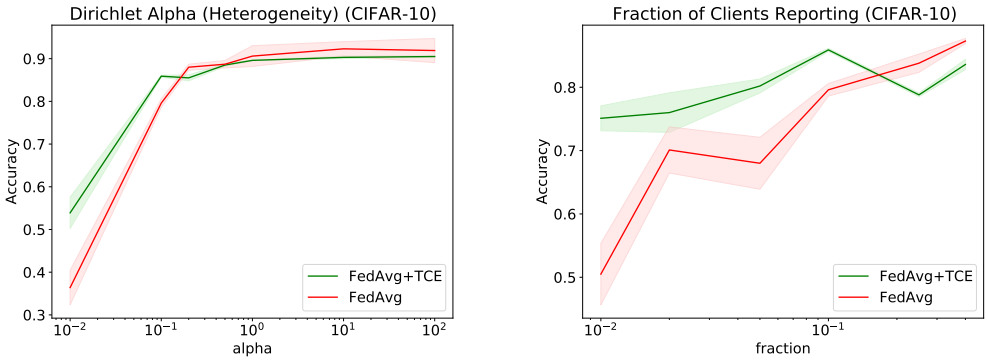
Method	*Hyper-params* lr	norm	*Dataset* CIFAR-10	CIFAR-100	FEMNIST
FEDAVG	0.7	group	NC	NC	NC
FEDAVG+TCE (OURS)			NC	NC	NC
FEDAVG+TCE (OURS)		batch	0.805 ± 0.039	0.350 ± 0.037	0.803
FEDAVG	0.5	group	0.777 ± 0.033	0.223 ± 0.059	0.695
FEDAVG+TCE (OURS)			0.751 ± 0.039	0.382 ± 0.036	0.700
FEDAVG+TCE (OURS)		batch	0.832 ± 0.028	0.354 ± 0.019	0.831
FEDAVG	0.3	group	0.77 ± 0.022	0.324 ± 0.148	0.801
FEDAVG+TCE (OURS)			0.836 ± 0.029	0.438 ± 0.041	0.787
FEDAVG+TCE (OURS)		batch	0.836 ± 0.010	0.309 ± 0.060	**0.850**
FEDAVG	0.1	group	0.79 ± 0.017	0.365 ± 0.131	0.805
FEDAVG+TCE (OURS)			0.846 ± 0.010	0.451 ± 0.046	0.806
FEDAVG+TCE (OURS)		batch	0.841 ± 0.020	0.397 ± 0.044	**0.850**
FEDAVG	0.07	group	0.751 ± 0.020	0.486 ± 0.036	0.805
FEDAVG+TCE (OURS)			0.832 ± 0.029	0.429 ± 0.024	0.774
FEDAVG+TCE (OURS)		batch	$\mathbf{0.859 \pm 0.009}$	0.478 ± 0.024	0.836
FEDAVG	0.05	group	0.742 ± 0.087	0.456 ± 0.031	0.829
FEDAVG+TCE (OURS)			0.834 ± 0.011	0.484 ± 0.030	0.791
FEDAVG+TCE (OURS)		batch	0.852 ± 0.010	0.506 ± 0.008	0.830
FEDAVG	0.03	group	0.787 ± 0.011	0.486 ± 0.013	0.735
FEDAVG+TCE (OURS)			$\mathbf{0.859 \pm 0.004}$	0.520 ± 0.006	0.732
FEDAVG+TCE (OURS)		batch	0.850 ± 0.018	0.513 ± 0.030	0.813
FEDAVG	0.01	group	0.742 ± 0.065	0.429 ± 0.052	0.780
FEDAVG+TCE (OURS)			0.825 ± 0.004	$\mathbf{0.524 \pm 0.010}$	0.809
FEDAVG+TCE (OURS)		batch	0.845 ± 0.015	0.479 ± 0.015	0.791
FEDAVG	0.007	group	0.739 ± 0.030	0.440 ± 0.041	0.780
FEDAVG+TCE (OURS)			0.751 ± 0.005	0.501 ± 0.018	0.814
FEDAVG+TCE (OURS)		batch	0.747 ± 0.013	$\mathbf{0.524 \pm 0.010}$	0.782

Table 1: Accuracy results of FedAVG with and without TCE for different settings of client learning rates as well as normalization layer settings. We observe that for many learning rate settings TCE consistently improves performance, as well as having the highest overall accuracy by a large margin. Note that results of FedAVG with batchnorm are not included as the model often fails to train from any lr settings or yields very poor performance. On the other hand TCE combined with batchnorm based models is competitive with group norm. NC indicated cases for which the algorithm did not converge

We focus our analysis on the highly heterogeneous case $\alpha = 0.1$, which can also lead to higher variance in the results, particularly for smaller datasets such as CIFAR-10 and CIFAR-100. We thus run each training 3 times to reduce variance of the results.

4.1 FORGETTING DURING A FEDERATED ROUND

We study the local client forgetting at different optimization rounds of federated learning. In Figure 2 we show F_{ki} (the heatmap) and F_k (the last column) for the set of participating clients for the respective round for FedAVG, when using both TCE and CE. We observe that the forgetting is very high when using standard CE, as no extra steps are taken to control forgetting. On the other hand TCE is able to greatly control the local client forgetting. This leads to better overall performance after aggregation. The results are shown for round 2800 of training CIFAR-10. However, the observation is further confirmed for other rounds as shown in the Appendix Sec A.2. Having observed that TCE can indeed reduce the local client forgetting we now study its effect on the aggregated models.



(a) (b)

Figure 4: Ablations for (a) data heterogeniety (b) number of participating clients. We observe that (a) TCE gives improvements in cases where data is highly heterogenous (b) TCE is most useful with low number of participating clients.

4.2 Evaluations on CIFAR-10,CIFAR-100 and FEMNIST

We now evaluate TCE in combination with FedAVG on CIFAR-10, CIFAR-100, and FEMNIST datasets, demonstrating it can greatly improve model performance for a number of datasets under various settings. Figure 3 shows the best performing models among FedAVG and FedAVG+TCE for CIFAR-10 and CIFAR-100. From these results, we observe FedAvg+TCE is robust to the normalization layer unlike vanilla FL methods which typically does poorly when using batch normalization. In general we conclude that for both CIFAR-10 and CIFAR-100, TCE substantially improves performance for any training budget. We see a similar substantial performance increase for FedAvg+TCE when compared to vanilla FedAvg with the FEMNIST dataset (Table 1) which supports our claims.

Robustness to learning rate and normalization In Table 1 we summarize a detailed analysis of model performance for the set of learning rates $\eta = \{0.007, 0.01, 0.03, 0.05, 0.07, 0.1, 0.3, 0.5, 0.7\}$. Additionally, we vary the normalization method between batch norm and group norm for TCE. With regard to the normalization method, we observe comparable performance using either batch norm or group norm which indicates the truncated cross entropy method helps to mitigate performance degradation typically observed when using batch norm in heterogeneous FL Reddi et al. (2020). In general, we observe that across all client learning rates TCE provides performance improvement using either normalization setting. We particularly notice improvements by TCE even at higher learning rates, where vanilla FedAVG can collapse or under-perform. We also remark that not only is overall performance higher for FedAVG+TCE but it's performance deviates less from the case of it's best performing hyperparameters as various settings are changed. This analysis is further supported by Table 2 in the Appendix.

4.3 Ablations

We now further study the behavior of TCE in combination with FedAVG under different data distributions and client participation settings. In Figure 4 ablations for the CIFAR-10 setting are shown where we ablate one setting at a time. The learning rate for the ablation studies is set according to the best performing learning rate determined in Table 1 *i.e.* 0.1 for FedAvg and 0.03 for FedAvg+TCE with group norm. We observe particular settings of α and the fraction of clients selected at each round, for which TCE provides particular improvement.

Parameter α of the Dirichlet Distribution The Dirichlet distribution is parameterized by α. As α approaches 0 the client distribution will become increasingly likely to contain only one label and as α increases the client distribution will be increasingly i.i.d.. Similar to Hsu et al. (2019), we investigate $\alpha = \{0.01, 0.1, 0.2, 0.5, 1, 10, 100\}$ with $\alpha = 100$ considered i.i.d.. We observe that as α increases and the data becomes homogeneous, the gap between TCE and CE shrinks. This

observation is due to the fact that most clients will have most of the classes and thus the approaches become nearly equivalent. On the other hand we observe that reducing α to 0.01, an heterogeneous case, more extreme than that studied in our primary experiments ($\alpha = 0.1$), then the performance improvement gap further widens between FedAvg and FedAvg+TCE.

Fraction of Participating Clients For the fraction of participating clients, C we observe the largest performance gap between FedAvg+TCE and FedAvg when the number of participating clients is very low. This result is significant since it mimics most closely potential real world settings. We hypothesize this effect is because unless we control forgetting, non-participating clients will have their data distributions forgotten and unlike participating clients will be unable to contribute their updated to the global model to counteract this forgetting. As the fraction of clients selected at each round increases, we observe the performance gap between the two methods narrow since more clients will have the opportunity to be selected at each round and "remind" the model of their data distributions.

5 Conclusion

In this paper we took a deeper look at the *local client forgetting* problem. Through extensive experiments, we showed that when a client performs local updates during federated learning, it risks overly optimizing its local objective, which can lead to forgetting on other subsets of data, in turn degrading the performance of the global model. We showed that this phenomenon is especially severe in cases where there is a significant distribution mismatch across clients. First making the connection with the heavily studied catastrophic forgetting problem in Continual Learning, we then proposed a local, client level, modification of the objective function which we call truncated cross entropy, that allows us to mitigate client level forgetting. We demonstrate our method can lead to improved performance when combined with a standard federated learning algorithm, particularly in the regime of highly heterogeneous client datasets and/or when a small percentage of clients are selected at each round. We also demonstrate TCE is more robust to a wide range of hyperparameter settings than vanilla FedAvg. Finally, we showed that addressing this local client forgetting is particularly important in cases of severe data heterogeneity. Future investigation can consider the combination of our modified objective function with a broader range of algorithms designed for federated learning.

References

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ssil: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 844–853, 2021.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/15825aee15eb335cc13f9b559f166ee8-Paper.pdf.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Zhiyuan Chen and Bing Liu. Lifelong supervised learning. In *Lifelong Machine Learning*, pp. 33–74. Springer, 2018.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16712–16721, 2022.

Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Timothée Lesort. Continual feature selection: Spurious features in continual learning. *arXiv preprint arXiv:2203.01012*, 2022.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Irene Tenison, Sai Aravind Sreeramadas, Vaikkunth Mugunthan, Edouard Oyallon, Eugene Belilovsky, and Irina Rish. Gradient masked averaging for federated learning. *arXiv preprint arXiv:2201.11986*, 2022.

Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Chencheng Xu, Zhiwei Hong, Minlie Huang, and Tao Jiang. Acceleration of federated learning with alleviated forgetting in local training. *arXiv preprint arXiv:2203.02645*, 2022.

Dezhong Yao, Wanning Pan, Yutong Dai, Yao Wan, Xiaofeng Ding, Hai Jin, Zheng Xu, and Lichao Sun. Local-global knowledge distillation in heterogeneous federated learning with non-iid data. *arXiv preprint arXiv:2107.00051*, 2021.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pp. 12073–12086. PMLR, 2021.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International Conference on Machine Learning*, pp. 12878–12889. PMLR, 2021.

A APPENDIX

A.1 ADDITIONAL EVALUATION OF HYPERPARAMETER ROBUSTNESS

In Table A.1 we further confirm the robustness of FedAvg+TCE to learning rate. For each dataset used in the experiments we indicate accuracies of models trained using a different learning rate that fall within 2%, 3% and 5% of the best accuracy obtained in training. The best learning rate is indicated in column 2 for convenience.

	best acc (best lr)	lr for acc within 2%	lr for acc within 3%	lr for acc within 5%
CIFAR-10				
FedAvg	0.796 (0.1)	0.1, 0.03	0.3, 0.1, 0.03	0.3, 0.1, 0.07, 0.03
FedAvg+TCE (group)	0.860 (0.03)	0.1, 0.03	0.3, 0.1, 0.07, 0.05, 0.03	0.3, 0.1, 0.07, 0.05, 0.03, 0.01
FedAvg+TCE (batch)	0.859 (0.07)	0.1, 0.07, 0.05, 0.01	0.3, 0.1, 0.07, 0.05, 0.01	0.3, 0.1, 0.07, 0.05, 0.01
CIFAR-100				
FedAvg	0.486 (0.07)	0.07	0.07, 0.03	0.07, 0.05, 0.03, 0.007
FedAvg+TCE (group)	0.524 (0.01)	0.01	0.05, 0.03, 0.01, 0.007	0.05, 0.03, 0.01, 0.007
FedAvg+TCE (batch)	0.524 (0.007)	0.05, 0.03, 0.007	0.05, 0.03, 0.007	0.07, 0.05, 0.03, 0.01, 0.007
FEMNIST				
FedAvg	0.850	0.05	0.05,0.07, 0.1, 0.3	0.007, 0.01, 0.05, 0.07, 0.1, 0.3
FedAvg+TCE (group)	0.814	0.007, 0.01, 0.1	0.007, 0.01,0.05, 0.07, 0.1, 0.3	0.007, 0.01, 0.05, 0.07, 0.1, 0.3
FedAvg+TCE (batch)	0.850	0.05, 0.07, 0.1, 0.3, 0.5	0.05, 0.07, 0.1, 0.3, 0.5	0.05, 0.07, 0.1, 0.3, 0.5, 0.7

Table 2: learning rates where accuracy is within a specified tolerance of the best accuracy. We observe that not only does TCE provide the best accuracy, this accuracy is less sensitive to hyperparameters

A.2 ADDITIONAL FORGETTING STUDIES

In Figures 5, 6, 7 we show additional results for other rounds of training for forgetting. Figure 5 is evaluated after the first rounf of training, Figure 7 is evaluated after the 4000^{th} round of training and Figure 6 is evaluated right in the middle of training, after the 2000^{th} round. We observe in very early rounds since performance is still very low for many client datasets, there is not as much accuracy to destroy, however we still observe several cases where initial accuracy of the model is substantial enough that forgetting is observably more extreme without TCE. By the middle of training at round 2000, we see clear indications of forgetting, the bottom row of Figure 6 corresponding to FedAvg without TCE shows substantially better performance on its own dataset (indicated along the diagonal values). At the completion of training (Figure 7) we see FedAvg+TCE doing much better than FedAvg at overcoming the local client forgetting problem.

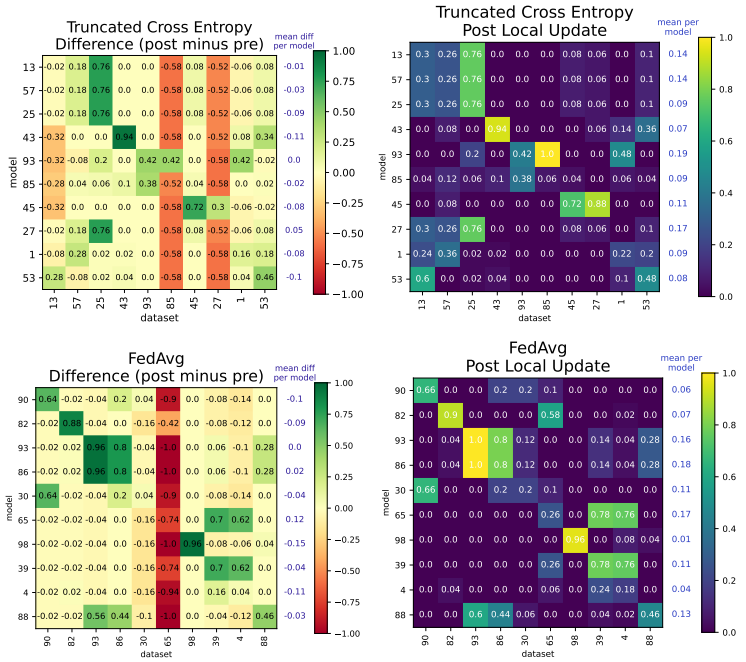


Figure 5: After round 1: The model of each client is evaluated on its own dataset and the datasets of each other client selected for the round both prior to training (right) and after training (center). The difference between the post and prior accuracies is presented on the left.

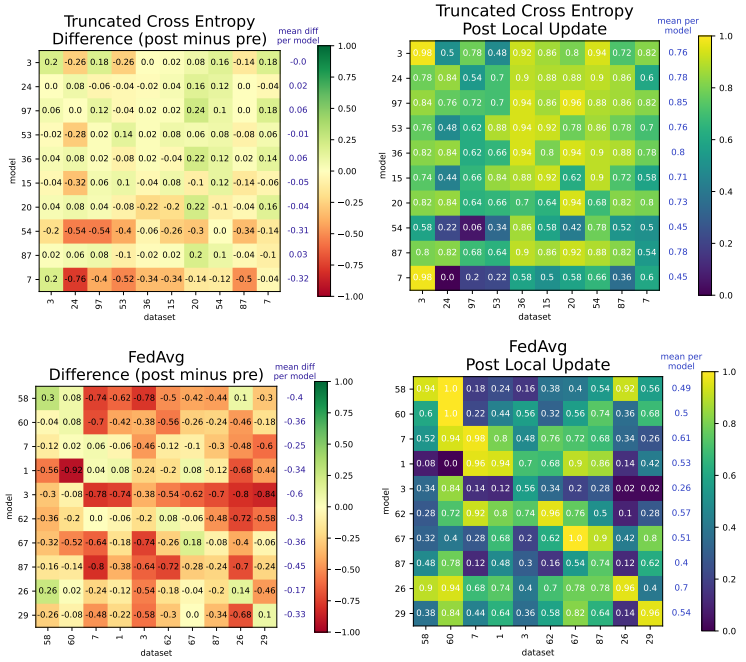


Figure 6: After round 2000: The model of each client is evaluated on its own dataset and the datasets of each other client selected for the round both prior to training (right) and after training (center). The difference between the post and prior accuracies is presented on the left.

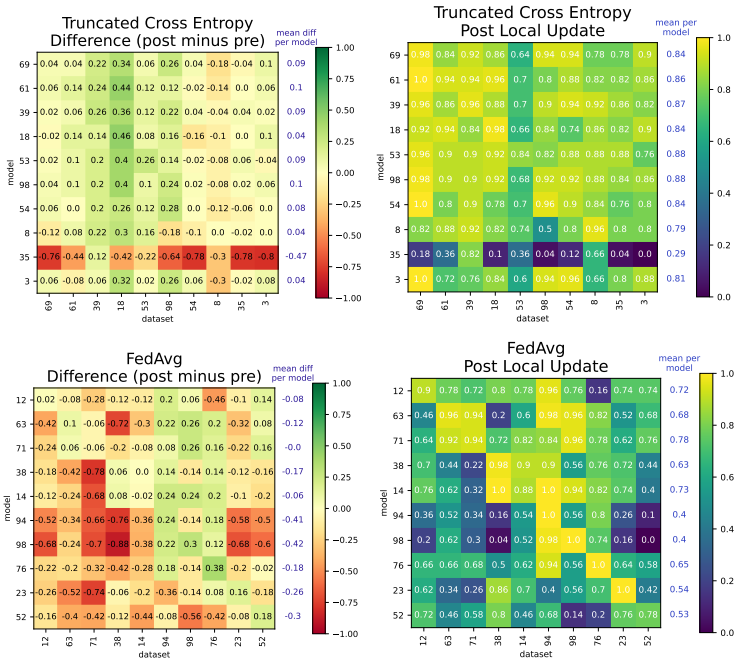


Figure 7: After round 4000: The model of each client is evaluated on its own dataset and the datasets of each other client selected for the round both prior to training (right) and after training (center). The difference between the post and prior accuracies is presented on the left.