CFD: Communication-Efficient Federated Distillation via Soft-Label Quantization and Delta Coding

Felix Sattler*, Arturo Marban, Roman Rischke, and Wojciech Samek*, *Member, IEEE* †‡

***Abstract*—Communication constraints are one of the major challenges preventing the wide-spread adoption of Federated Learning systems. Recently, Federated Distillation (FD), a new algorithmic paradigm for Federated Learning with fundamentally different communication properties, emerged. FD methods leverage ensemble distillation techniques and exchange model outputs, presented as soft labels on an unlabeled public data set, between the central server and the participating clients. In this work, we investigate FD from the perspective of communication efficiency by analyzing the effects of active distillation-data curation, soft-label quantization, and delta-coding techniques. Based on the insights gathered from this analysis, we present Compressed Federated Distillation (CFD), an efficient Federated Distillation method. Extensive experiments, on Federated image classification and language modeling problems, at different levels of data heterogeneity, demonstrate that our method can reduce the amount of communication necessary to achieve fixed performance targets by more than two orders of magnitude when compared to FD, and by more than four orders of magnitude when compared to parameter averaging based techniques like Federated Averaging.**

***Index Terms*—federated learning, distributed training, machine learning, deep learning, efficient communication**

I. Introduction

As many cases of data leakage and misuse in recent times have demonstrated, the centralized processing of personal user data in the "cloud" (e.g., for training deep learning models) is associated with a high privacy risk for the data donors. To address this issue, recently a novel distributed training paradigm called Federated Learning (FL) emerged.

FL [1][2][3] allows multiple entities to jointly train a machine learning model on their combined data, without any of the participants having to reveal their potentially privacy sensitive data to a centralized server. Federated Learning achieves this, by processing the data on the local devices and only communicating sanitized or encrypted information about the underlying patterns to other devices and the server.

Besides improving privacy, FL comes with many other benefits such as improved security [4], autonomy [5], and efficiency [6] due to its distributed nature and on-device processing.

FL is typically performed between mobile and internet of things (IoT) devices, which are often severely hardware constrained, geographically scattered, and have only access to limited and costly communication channels like metered mobile networks. Thus, to harness the ever-growing amounts of privacy-sensitive data collected by these devices, there is a great need for efficient and scalable FL solutions.

One of the most challenging obstacles in Federated Learning, is the communication bottleneck induced by frequently exchanging training information between the participating clients over limited bandwidth channels. For instance, the communication of local gradients, which are the basic unit of information for gradient descent based distributed training methods like distributed SGD, requires $\mathcal{O}(|\theta|)$ bits of information, where $|\theta|$ is the model size. Over the course of multiple thousands of training rounds the communication overhead can grow to hundreds of Gigabytes for modern large-scale neural-network models with millions of parameters. Consequently, if communication bandwidth is limited or communication is costly, Federated Learning can become unproductive or even completely unfeasible.

To address this issue, different algorithmic approaches have been proposed under the umbrella of efficient Federated Learning. In this work, we closely examine the recently proposed framework of Federated Distillation [7][8][9][10][11][12][13] with respect to its communication properties and introduce a set of improvements, which reduce communication in both the upstream and the downstream, without negatively affecting the training performance. More concretely, we make the following contributions:

- We conduct a qualitative and quantitative comparison between the communication properties of two popular algorithmic frameworks for Federated Learning, namely Federated Averaging [1][14][15] and Federated Distillation [7][11][8].
- We perform a thorough analysis of the communication properties of Federated Distillation at different levels of data heterogeneity by investigating the effects of distillation data set size as well as active data selection strategies on the training performance.
- We develop a novel quantization mechanism and delta coding method to compress the soft-labels exchanged in Federated Distillation before communication.
- We address the issue of compressing downstream communication via a novel dual distillation technique.
- Finally, we perform extensive experiments on large-scale convolutional neural networks and transformer models, which demonstrate that our compression method can

*Corresponding authors: F. Sattler and W. Samek.

†This work was supported by the Federal Ministry of Education and Research (BMBF) through the BIFOLD - Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037I), and the EU's Horizon 2020 project COPA EUROPE.

‡F. Sattler, A. Marban, R. Rischke, and W. Samek are with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: felix.sattler@hhi.fraunhofer.de, wojciech.samek@hhi.fraunhofer.de).

reduce communication by more than $\times 100$ as compared to Federated Distillation and more than $\times 10000$ as compared to Federated Averaging.

The remainder of this manuscript is organized as follows: In section II, we describe the two major algorithmic frameworks in Federated Learning, namely Federated Averaging and Federated Distillation, compare them w.r.t. their communication properties, and review existing techniques for communication reduction in both frameworks. In section III, we provide a literature review of the key developments in Federated Learning reasearch in general and Federated Distillation in particular. In section IV, we thoroughly investigate ways to reduce the communication in Federated Distillation by systematically addressing all components that contribute to the total communication load. In section V, we condense the gathered insights and propose Compressed Federated Distillation (CFD), a novel communication-efficient Federated Distillation scheme. Finally, in section VI, we compare the communication properties of CFD with those of regular Federated Distillation and Federated Averaging on a variety of Federated Learning benchmarks featuring large-scale convolutional and transformer neural networks, before concluding in section VII.

II. Algorithmic Frameworks for Federated Learning

To solve Federated Learning problems, two algorithmic frameworks have been proposed, which drastically differ with respect to their communication properties. Figure 1 gives an overview of these frameworks and compares them w.r.t. to the flow of computation and communication.

A. Federated Averaging

The classical algorithmic approach to Federated Learning problems is Federated Averaging [1] (Figure 1, illustration on the left). In Federated Averaging the training is conducted in multiple communication rounds following a three step protocol:

1) In the beginning of each round, the central server selects a subset of the client population and broadcasts to them a common model initialization θ.
2) Starting from the common initialization, the selected clients individually perform iterations of stochastic gradient descent over their local data to improve their local models resulting in an updated model θ_i on every client.
3) The updated models are then communicated back to the server, where they are aggregated (e.g., by an averaging operation) to create a global model, which is used as initialization point for the next communication round.

Every communication round of Federated Averaging thus involves the upstream and downstream communication of a complete parametrization of the jointly trained model θ between all participating clients and the server. In many practical applications, these neural network parametrizations may contain multiple millions to billions of individual parameters. For instance, the widely popular ResNet-50 [16] contains over 23 million parameters. For natural language processing tasks, even larger models are used, with the famous GPT-3 [17] reaching 175 billion parameters. Generally, both theoretical [18], [19] and empirical [20] evidence suggests that the performance of neural network models correlates positively with their size.

For large-scale models like the ones described above, the communication overhead of running the Federated Averaging algorithm can become a prohibitive bottleneck. Although a wide variety of methods to reduce the communication overhead in Federated Averaging have been proposed, including approaches that reduce the frequency of communication [1], use client sampling [1], [21], neural network pruning [22], message sparsification [23], [24], [25] and other lossy [26], [27], [28], [24], [29] and lossless compression techniques [30], [31], the fundamental issue of scaling to larger models persists.

B. Federated Distillation

The recently proposed Federated Distillation [7], [12], [11] (Figure 1, illustration on the right) takes an entirely different approach to communicating the knowledge obtained during the local training. Instead of communicating the parameterization of the locally trained model θ_i to the server, in Federated Distillation the knowledge is communicated in the form of soft-label predictions on records of a public distillation data set X^{pub} according to

$$Y_i = \{f_{\theta_i}(x) | x \in X^{pub}\}. \tag{1}$$

Hereby f_{θ_i} is the (neural network) locally trained model parameterized by θ_i. Prior work [12][32] has shown that this public distillation data needs to only roughly follow a similar distribution as the privacy-sensitive client data. Hence, a wide variety of data sets may be suitable to pose as distillation data. For instance, in many Federated computer vision problems, extremely large image corpora like ImageNet [33] are publicly available. Likewise, for natural language processing problems, public text corpora like WiKiText [34] can be found. While this public data is typically unfit for training a task-specific model due to missing label information, it can still be useful in Federated Distillation pipelines.

Different variations of Federated Distillation have been proposed that vary w.r.t. their communication properties. To fully appreciate the communication-saving benefits of Federated Distillation, it is necessary to avoid communication of model parametrizations at all stages of Federated training. Therefore, we consider the following version of the Federated Distillation protocol for which each communication round consists of the following five steps:

1) At the beginning of every Federated Distillation round, a subset of the client population is selected for participation and synchronizes with the server by downloading aggregated soft-labels, Y^{pub}, on the public data set.
2) The participating clients update their local models by performing model distillation using the downloaded soft-label information. All stochasticity in the distillation process is controlled via random seeds to ensure that all clients end up with the same distilled model θ.
3) The participating clients improve the distilled model by training on their private local data, resulting in improved models θ_i on every client.

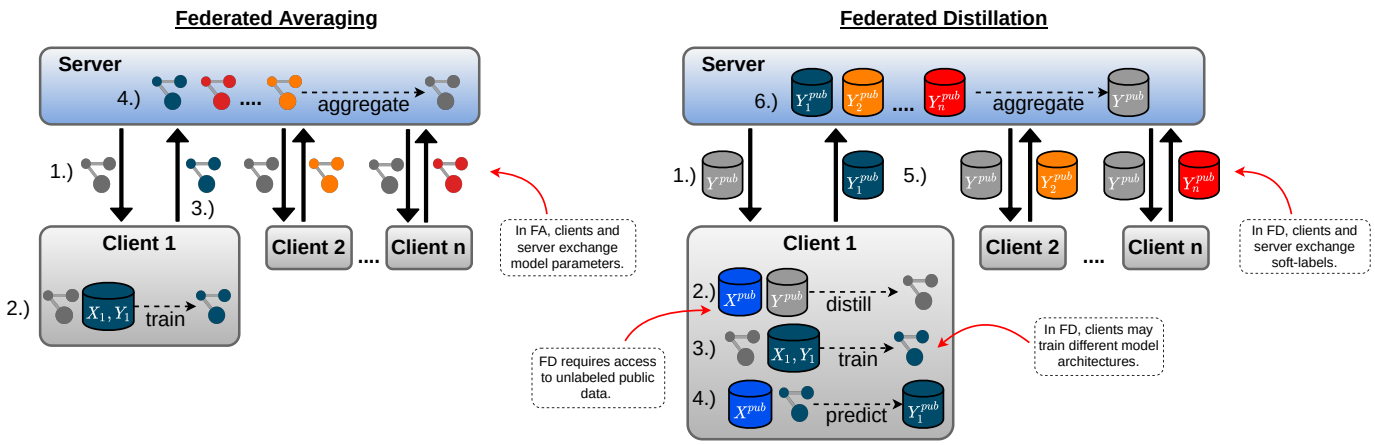


Fig. 1: The flow of data and computations in Federated Averaging and Federated Distillation. In Federated Averaging, the model parameters θ are used to transfer the training information between clients and the server. In Federated Distillation, soft-label predictions Y^{pub} on a common public data set X^{pub} are used to convey the same information.

4) Using the locally trained model θ_i, the clients compute soft-labels Y_i^{pub} on the public data and send them to the server.
5) The server aggregates the soft-labels for the next communication round.

Variations of this protocol have been proposed in [7], [8] and [11].

As demonstrated in recent studies [7], [12], [11], Federated Distillation has several advantages over Federated Averaging: First, as model information is aggregated by means of distillation, Federated Distillation allows the participating clients to train different model architectures. This gives additional flexibility in settings where clients have heterogeneous hardware constraints. Federated Distillation also benefits from increased robustness, as adversarial or malicious clients can not directly influence the parametrization of the jointly trained model (only indirectly via their soft-labels). Furthermore, Federated Distillation has favorable privacy properties, as in contrast to parameter averaging-based Federated Learning algorithms, it is not directly vulnerable to model inversion attacks [35][36]. The most significant advantage, however, arises from the fact that Federated Distillation has a completely different communication profile than Federated Averaging. While the upstream and downstream communication in every round of Federated Averaging scales with the size of the jointly trained neural network as

$$\mathtt{b} \in \mathcal{O}(|\theta|)) \tag{2}$$

in Federated Distillation, communication scales with the product of the distillation data set size $|X^{pub}|$ and the number of different classes $\dim(\mathcal{Y})$ as

$$\mathtt{b} \in \mathcal{O}(|X^{pub}|\dim(\mathcal{Y})). \tag{3}$$

This can put FD at an advantage in applications where large neural networks are trained, as is the case for instance in natural language processing and computer vision tasks (among many other application).

Nevertheless, Federated Distillation is still communication intensive, especially for large multi-class tasks where sizable distillation data sets are used.

The aim of this work is thus to further improve the communication efficiency in FD, by exploring a variety of communication reduction techniques. Our efforts will culminate in the development of our Compressed Federated Distillation (CFD) method, a novel compression technique for FD based on soft-label quantization, delta coding and dual distillation.

III. Related Work

Federated Distillation: Albeit their novelty, Federated Distillation techniques have been used in several existing works already. In the following we present a comprehensive overview on these existing techniques. Most relevant for the studied multi-round protocol for diverse models in this paper is the protocol proposed by Itahara et al. [11], which is based on ideas from Jeong et al. [7] and mostly follows the steps described in section II-B with the sole exception that client models are required to participate in every round and are not kept synchronized during local distillation by means of random seeds. The similar protocol by Jeong et al. [7] and Seo et al. [37] is instead based on locally accumulated logits per *label*, which are aggregated by the server. Furthermore, instead of exploiting these global logits for refining the local models by direct distillation, they are used for regularizing the local training in the next round. Similarly, Bistritz et al. [38] use distillation on an unlabelled public data set for regularizing on-device learning in a peer-to-peer network. Guha et al. [39] propose a one-shot distillation method for convex models, where the server distills the locally optimized client models in a single round based on an unlabelled data set.

The recently proposed FedMD by Li and Wang [8] and Cronus by Chang et al. [40] also address knowledge distillation in Federated Learning through aggregated logits for a public data set. In FedMD, the clients train in each round first on the public data set and then on the private data set for personalization and communicate afterwards their model output

on the public data set to the server, where the aggregation of the uploaded logits for the next round is performed. For the initial pretraining in FedMD the public data set is required to be labelled, whereas in the communication rounds after initialization the aggregated logits from the clients serve as soft-labels for the public data set. In Cronus, however, each client uses the local data set and the soft-labelled public data set jointly for local training.

Lin et al. [12] apply ensemble distillation on top of Federated Averaging to refine the global server model resulting in fewer communication rounds compared to benchmark Federated Averaging methods. Although leveraging the power of ensemble distillation for robust model fusion and data augmentation, their method, called FedDF, is based on the classical Federated Averaging protocol with all the mentioned consequences w.r.t. the communication-efficiency.

Chen and Chao [13] introduce FedBE, where the server creates Bayesian model ensembles based on the uploaded client models, instead of directly averaging the client models as in FedAvg, and uses an unlabelled data set to distill one global student model from a Bayesian model ensemble created from the teacher models. This global model is transferred back to the clients as initialization for the next round of local training. Their approach however is more closely related to classical Federated Averaging than to Federated Distillation, since all clients have to train the same model architecture and the model parameters are communicated up- and downstream.

Ahn et al. [9] and Oh et al [10] also investigate hybrid approaches that use both Federated Distillation and Federated Averaging techniques. Ahn et al. focuses on wireless communication aspects of Federated Distillation while Oh et al. address privacy aspects via a Mixup data augmentation strategy.

While quantization techniques have been widely applied in Federated Averaging [14][15][41][42], we are not aware of any prior work that aims to improve the communication efficiency of the Federated Distillation process by means of quantizing the soft-label information. We initiate this study in the hope to foster further research in this direction, for this new Federated Learning paradigm.

Data Heterogeneity in Federated Learning: Federated Learning is typically performed between distributed mobile or IoT devices, which locally and independently generate private data based on their particular environment and usage patterns. As a consequence, Federated Learning problems are typically characterised by statistically heterogeneous client data [1]. It is well known, that conventional FL algorithms like Federated Averaging [1] perform best on statistically homogeneous data and suffer severely in this ("non-iid") setting [43], [44]. A number of different studies [45], [43], [25] have tried to address this issue, but relevant performance improvements so far have only been possible under strong assumptions. For instance [43] assume that the server has access to *labeled* public data from the same distribution as the clients, other approaches [25] require high-frequent communication, with up to thousands of communication rounds, between server and clients, which might be prohibitive in a majority of FL applications where communication channels are intermittent and slow. A different line of research, which aims to address data heterogeneity in FL via meta- and multi-task learning. Here, separate models are trained for each client [46], [47] or clients are grouped into different clusters with similar distributions [48], [49].

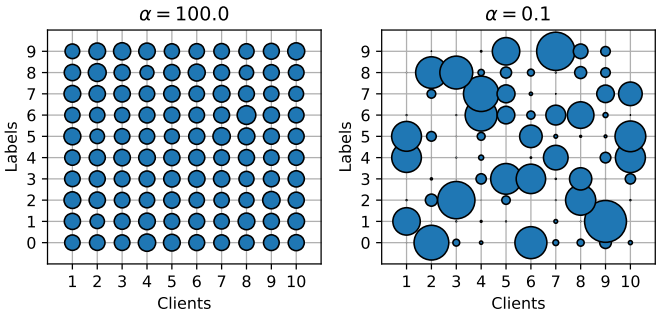


Fig. 2: Illustration of the Dirichlet data splitting strategy we use throughout the paper, exemplary for a Federated Learning setting with 10 Clients and 10 different classes. Marker size indicates the number of samples held by one client for each particular class. Lower values of α lead to more heterogeneous distributions of client data. Figure adapted from [50][12].

IV. Investigating the Communication Properties of Federated Distillation

In this section we investigate the communication properties of Federated Distillation. The total amount of communication necessary to transfer the soft-label information in each round is given by the product of the distillation data set size and the average amount of bits required to store the value of one soft-label

$$\mathtt{b}_{total} = |X^{pub}| \times (H(Y_i) + \eta). \tag{4}$$

Hereby $H(Y_i)$ is the entropy of the soft-labels, and η indicates the coding inefficiency. In conventional Federated Distillation as proposed in [7], [11], the soft-label information is stored at 32-bit floating-point precision, and thus, we have

$$\mathtt{b}_{total} = |X^{pub}| \times \dim(\mathcal{Y}) \times 32 \text{ bit}. \tag{5}$$

Following eq. (4), a reduction of the communication overhead can be achieved by either

(a) reducing the size of the distillation data set,
(b) reducing the entropy of the soft-labels, or
(c) improving the efficiency of the coding technique.

In this section, we will look at all three of these determining factors and investigate their relative impact on the Federated Learning performance.

In the preliminary experiments performed in this section, we consider Federated Learning settings with 20 clients among which we split the training data according to a Dirichlet distribution [50], as illustrated in Figure 2. More details on the experiment setup can be found in section VI.

A. Distillation Dataset Size & Active Learning Strategies

As the communication overhead in Federated Distillation is directly proportional to the number of data points used for

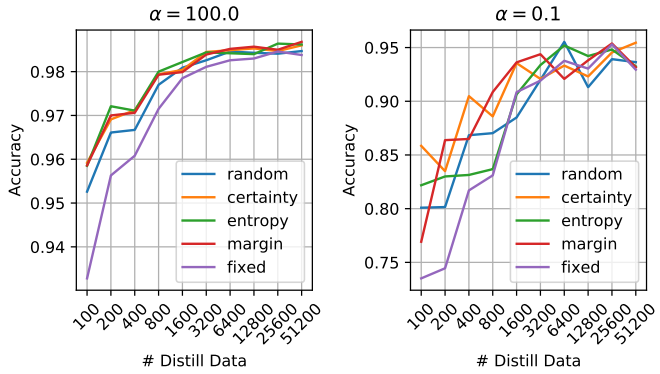


Fig. 3: Effect of distillation data set size with different (active) selection strategies using LeNet on MNIST.

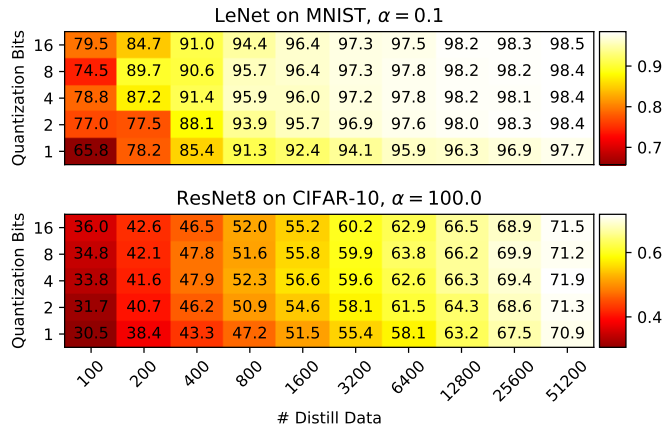


Fig. 4: Effect of distillation data set size and quantization strength on training performance in Federated Distillation using LeNet on MNIST at $\alpha = 0.1$ and ResNet-8 on CIFAR-10 at $\alpha = 100.0$.

distillation, restricting the size of the distillation data is the most straight-forward way to reduce communication. It is commonly known however, that in machine learning (and deep learning in particular), the size of the training data set has strong impact on the generalization capacity of any trained classifier [51]. The machine learning discipline of active learning has developed techniques to systematically select samples from a larger pool of data for training with the goal to achieve higher performance with fewer samples of data. Here, we adapt four popular active learning techniques to the setting of Federated Distillation and compare their performance when used to select distillation data sets of different sizes. Let

$$\text{top}_n[x \mapsto \Psi(x)] : \mathcal{D} \to \mathcal{D} \tag{6}$$

be the operator that maps a data set to one of its subsets of size n, by selecting the top n elements according to the criterion $x \mapsto \Psi(x)$. Then, we can define the "entropy", "certainty", and "margin" selection strategies as follows:

$$D_n^{entropy} = \text{top}_n[x \mapsto H(f_\theta(x))](X^{pub}) \tag{7}$$

$$D_n^{certainty} = \text{top}_n[x \mapsto -\max(f_\theta(x))](X^{pub}) \tag{8}$$

$$D_n^{margin} = \text{top}_n[x \mapsto \max_2(f_\theta(x)) - \max(f_\theta(x))](X^{pub}) \tag{9}$$

Hereby, $H(p) = -\sum_i p_i \log(p_i)$ denotes the entropy, $\max(p)$ represents the maximum value in the vector of probabilities p, and $\max_2(p) = \max(p \setminus \{\arg\max(p)\})$ denotes the second-largest element of p. For instance $D_n^{certainty}$ selects those n data points from X^{pub} for which the maximum likelihood prediction $\max(f_\theta(x))$ is assigned the lowest certainty. We also consider the selection strategy of picking n data-points at random in each round.

In each communication round of Federated Distillation, we select a subset of n data points for distillation, according to one of the above strategies based on the model θ, which was used in the previous round. The results of this experiment are shown in Figure 3. As we can see, the performance of Federated Distillation strongly depends on the size of the distillation data set. On the other hand, the effect of using active learning strategies to systematically select data points is rather low. While in the i.i.d regime ($\alpha = 100.0$) the active learning strategies slightly improve the Federated Distillation performance, the situation is rather unclear in the non-i.i.d regime ($\alpha = 0.1$). From these results, we conclude that in most situations, the performance gains obtained by using active learning strategies do not justify the additional computational overhead incurred by these techniques (i.e., evaluating $f_\theta(x)$ on the entire accessible distillation data). Thus, in the remainder of this manuscript, we will restrict our analysis to randomly selected distillation data sets of fixed size.

B. Soft-Label Quantization

Quantization is a popular technique to reduce communication and has been successfully applied in Federated Averaging to reduce the size of the parameter updates [14], [24], [29]. Quantization techniques, however, so far have not been applied to Federated Distillation. Here we consider constrained uniform quantization to reduce the entropy of the communicated soft-labels. Let $p \in \mathcal{Y}$ be a vector of soft-label probabilities. Then, we obtain the quantized soft-label q via constrained uniform quantization [52] as follows

$$q = \mathcal{Q}_b(p) = \arg \min_{\substack{q_i \in \{\frac{l}{2^b-1}, l \in 0,..,2^b-1\} \\ \sum_i q_i = 1}} \|q - p\|_1 \tag{10}$$

The optimization problem above can be solved in log-linear time. In case the optimization problem in (10) does not have a unique solution, we randomly break the tie. As can be easily seen, for $b = 1$, the quantization operator $\mathcal{Q}_b$ is equivalent to the maximum vote:

$$\mathcal{Q}_1(p)_i = \begin{cases} 1 & \text{if } i = \arg\max(p) \\ 0 & \text{else} \end{cases} \tag{11}$$

Constrained uniform quantization as defined above reduces the number of bits required to communicate any vector of probabilities from 32-bits $\times$ dim($\mathcal{Y}$) to b-bits $\times$ dim($\mathcal{Y}$).

TABLE I: Effect of the client population size on the sensitivity to soft-label quantization. Displayed is the maximum Accuracy achieved when training ResNet-8 on Cifar-10 for 50 communication rounds (mean and standard deviation over 3 independent runs). The level of data heterogeneity is set to $\alpha = 1.0$.

# Clients	Quantization Bits 1	2	4	8	16
10	72.9±0.6	73.0±0.4	**73.3**±0.8	72.8±0.5	72.7±0.6
20	69.8±0.6	**70.2**±0.6	69.8±0.5	**70.2**±0.4	70.0±0.6
80	**62.6**±0.3	62.0±0.3	62.0±0.4	62.5±0.2	61.7±0.1

Figure 4 shows the effect of different distillation data set sizes and quantization levels on the model Accuracy after a fixed number of communication rounds. From this data, we notice two interesting trends. Firstly, we observe that, while reducing the number of quantization bits by half has the same effect on the communication overhead as reducing the size of the distillation data by the same amount, the former strategy has a much lower impact on the training performance. This result holds across different levels of quantization and distillation data set sizes. Second, as the size of the distillation data set increases, the harmful effects of quantization vanish. For instance, in the MNIST data set (Figure 4, top plot), the experimental findings suggest that for $n \geq 6400$ distillation data points, the model performance remains strong for any quantization level (with a small Accuracy degradation at the highest compression levels). Similar effects can be observed when training ResNet-8 the CIFAR-10 data set (Figure 4, bottom plot), where in some cases higher compression rates even lead to slight improvements in Accuracy. Moreover, notice that in both data sets, when the maximum number of distillation samples are available ($n = 51200$), the strongest compression level (i.e., 1-bit quantization) only incurs less then 1% Accuracy degradation on both data sets.

As can be seen in Table I, repeating these experiments at varying client population sizes yields similar results. While larger client populations suffer from reduced convergence speed at all quantization levels due to the higher degree of distribution in the system, the number of clients does not seem to have an effect on the quantization sensitivity.

These results indicate that as a means for reducing communication, quantization should be strictly preferred over distillation data set reduction, especially if one has access to a large distillation data set. In the following, we will concentrate our analysis on 1-bit quantization using the compression operator $\mathcal{Q}_1$.

C. Lossless Compression via Delta Coding

In this section we investigate efficient lossless coding techniques to minimize the size of the compressed soft-label representations. As shown in eq. (11), applying the compression operator $\mathcal{Q}_1$ to a vector of probabilities p results in a one-hot vector of size $\dim(\mathcal{Y})$. As this one-hot vector can also be represented by an integer number between 1 and $\dim(\mathcal{Y})$, a straight-forward encoding process would comprise

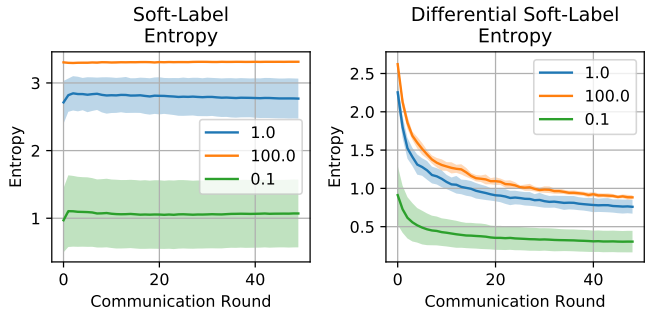


Fig. 5: Evolution of the soft-label entropy over multiple communication rounds when training ResNet-8 on the CIFAR-10 data set, at different levels of data-heterogeneity α. Left: When communicating compressed soft-labels directly, the entropy stays approximately constant over the course of training. Right: In contrast, when using differential soft-labels, the entropy steadily decreases as training progresses.

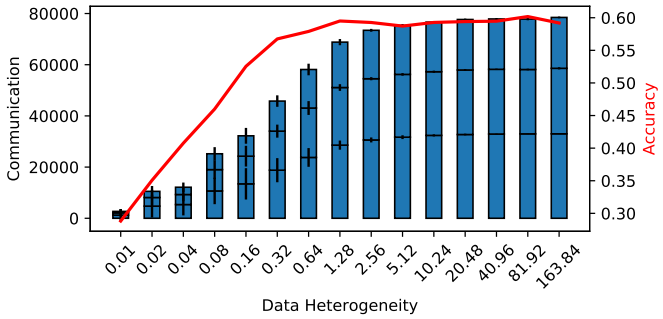


Fig. 6: Upstream communication (vertical bars) and model Accuracy (red curve), at different levels of data heterogeneity α, for ResNet-8 trained on CIFAR-10. Communication varies by more than an order of magnitude between the most homogeneous ($\alpha = 163.84$) and the most heterogeneous ($\alpha = 0.01$) setting.

of communicating

$$\tilde{Y}_i = \{\mathcal{Q}_1(f_{\theta_i}(x)) | x \in X^{pub}\} \tag{12}$$

as an array of $|X^{pub}|$ integer values, using up $|X^{pub}| \times \log_2(\dim(\mathcal{Y}))$ bits of data in total.

This however is only an upper bound on the true entropy $H(\tilde{Y}_i)$, which highly depends on the distribution of max predictions in $\tilde{Y}_i$. Figure 5 (left) shows the development of $H(\tilde{Y}_i)$ over the course of 50 communication rounds for a Federated Learning problem with 20 clients training ResNet-8 on CIFAR-10 at different levels of data heterogeneity. As we can see, the true entropy is well below the theoretical maximum of $\log_2(10)$ (for CIFAR-10 we have $\dim(\mathcal{Y}) = 10$) and decreases with increasing heterogeneity α, down to around $H(\tilde{Y}_i) \approx 1$ at $\alpha = 0.1$. This behaviour is expected, as the labels in the client training data, and consequently also their predictions $\tilde{Y}_i$, get more concentrated with increasing heterogeneity in the data. Additional knowledge about the distribution of $\tilde{Y}_i$ can be used to further reduce the entropy.

Since Federated Distillation is empirically known to converge [12][8], we can further expect there to be a growing overlap between the predictions made by a client in the current round T and those made in the previous round $T-1$.

High agreement between consecutive data points in a stream of data is a phenomenon commonly encountered in communication. The effect for instance can also be found in video data, where consecutive frames are often highly correlated. The canonical technique to exploit this pattern is differential coding (resp. delta coding or predictive coding) [53], which relies on only communicating "new" information in order to achieve higher compression rates.

We apply lossless delta coding to the quantized predictions of two consecutive rounds $\tilde{Y}^t$ and $\tilde{Y}^{t-1}$ by setting

$$(\hat{Y}^t)_l = \begin{cases} (\tilde{Y}^t)_l & \text{if } (\tilde{Y}^t)_l \neq (\tilde{Y}^{t-1})_l \\ 0 & \text{else} \end{cases} \quad \forall l \tag{13}$$

and measuring the entropy (in slight abuse of notation this assumes an arbitrary but fixed ordering of the set $\tilde{Y}$ and the same distillation data set X^{pub} to be used in all rounds). It should be noted, that all of the information contained in $\tilde{Y}^t$ can be retained from $\hat{Y}^t$ by comparing with the previous message $\hat{Y}^{t-1}$. This only requires minor additional bookkeeping by the central server (which is typically assumed to have access to strong computational resources).

Figure 5 (right) shows the development of the entropy of the differential updates $H(\hat{Y}_i)$. As we can see, the differential soft-label entropy behaves exactly as predicted and $H(\hat{Y}_i)$ is lower than $H(\tilde{Y}_i)$ from the first round on and smoothly decreases over the course of training. We note that, curiously, the development of the differential soft-label entropy over time can be very accurately predicted via the functional relation $H(\hat{Y}^t) \approx ct^{-d}$ for some constants c, d. We were able to replicate this behaviour across different model architectures and Federated Learning settings, hinting at an interesting underlying mathematical relationship, which could be the subject of future studies.

Figure 6 explores in more detail the influence of data heterogeneity on the amount of communication. It displays the upstream communication in the first three rounds of Federated Distillation with quantization and differential soft-label encoding. The resulting model Accuracy is also given (indicated by the red curve). As we can see, the amount of communication monotonically decreases when lowering the value of α (thus increasing the data heterogeneity), with more than an order of magnitude difference between the most homogeneous ("iid") setting at $\alpha = 163.84$ and the most heterogeneous setting at $\alpha = 0.01$ (note that homogeneity saturates for large values of α and values $\alpha \geq 100$ lead to an almost perfectly iid split of data).

D. Efficient Downstream Communication

So far we have only considered the upstream communication from the clients to the server. While in most Federated Learning settings with mobile and IoT devices, the uplink channel is more constrained than the downlink channel, it is still desirable to reduce the downstream communication as much as possible.

TABLE II: Effect of the initialization on the maximum accuracy achieved in Federated Distillation after 50 communication rounds. Displayed are the mean and standard deviation of the accuracy computed from 3 experiments, with a client participation rate of 20%. Our proposed dual-distillation approach closes the gap between random model initialization and the (infeasible) initialization from the previous model state.

	α	Init Previous	Init Random	Dual Distill
MNIST	100.0	**98.1**±0.0	96.7±0.1	97.8±0.0
	1.0	**97.9**±0.1	96.4±0.1	97.5±0.1
	0.1	92.6±1.4	90.1±1.5	**92.7**±1.2
CIFAR10	100.0	70.2±0.4	68.5±0.1	**74.9**±0.2
	1.0	68.2±0.6	66.0±0.6	**72.4**±0.7
	0.1	51.1±3.5	48.2±2.3	**56.1**±3.0

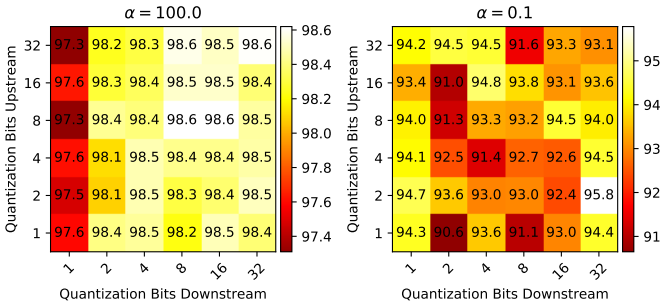


Fig. 7: Effect of different levels of upstream and downstream quantization on the training performance of LeNet on MNIST data set, at homogeneous ($\alpha = 100.0$) and heterogeneous ($\alpha = 0.1$) data settings. Displayed is the maximum Accuracy achieved after 20 communication rounds.

Compressing the down-link in Federated Learning however is challenging, as clients will run out of sync if they are not initialized with the same model in every round (the importance of a common model initialization has been famously demonstrated in [1]). If the participation rate is below 100% and clients do not participate in every round, state information (like the model state θ) becomes stale. To keep the client models synchronized under these conditions, clients need to either download the latest master-model θ from the server in every round (resulting in high downstream communication) or alternatively randomly re-initialize their local models in every round using a common random seed (resulting in performance degradation).

To illustrate this point, Table II shows the maximum Accuracy achieved after 50 communication rounds of Federated Distillation with three different client initialization schemes and three different neural networks at varying levels of data heterogeneity. As we can see, initializing the client models randomly before distillation ("Init Random") achieves worse performance than using the distilled model from the previous round ("Init Prev.") as the initialization point. To close the performance gap, we propose a novel dual distillation technique, which avoids de-synchronization of client models at arbitrary participation rates.

Let I_t be the set of clients participating in on particular

communication round t and

$$Y^{pub} = \frac{1}{|I_t|} \sum_{i \in I_t} \tilde{Y}_i^{pub} \tag{14}$$

be the corresponding set of aggregated soft-labels. In dual distillation, instead of directly sending the aggregated soft-labels to the clients, the server first performs a distillation step of it's own

$$\theta_S^t \leftarrow \text{train}(\theta_S^{t-1}, X^{pub}, Y^{pub}) \tag{15}$$

using the model θ_S^{t-1}, which was distilled in the previous round, as initialization point. This way the training information stored in θ_S^{t-1} is not lost. Then, the server computes soft-labels using the newly distilled model,

$$Y_S^{pub} = \{f_{\theta_S}(x) | x \in X^{pub}\} \tag{16}$$

and sends them to the clients. Starting from a random initialization, the participating clients then distill from the server predictions to mimic the server model

$$\theta \leftarrow \text{train}(\theta_0, X^{pub}, Y_S^{pub}) \tag{17}$$

This way the clients are indirectly initialized with all the accumulated training information stored in θ_S, before going into the next round of local training.

This allows us now to communicate soft-labels in upstream and downstream and appreciate the resulting communication savings in both directions, without loss of Accuracy (cf. Table II). To further reduce the amount of downstream communication, we can also quantize the server soft-labels Y_S^{pub} before communication, using the same constrained compression operator $\mathcal{Q}_{b_{down}}$ that we used in the upstream. We emphasize that dual distillation is only necessary if client participation is below 100%. Furthermore in Federated Learning the server is typically assumed to have much stronger computational resources than the clients, thus the workload of running an additional server model can mostly be neglected.

Figure 7 shows the effects of different levels of upstream and downstream quantization on the training performance of LeNet trained on MNIST, using Federated Distillation after 20 communication rounds. As we can see in the i.i.d. setting with $\alpha = 100.0$, downstream quantization appears to have a slightly stronger effect on the model performance than upstream quantization, with a maximum Accuracy drop of 1% at the highest quantization level, i.e., $b_{down} = 1$. In contrast, in the non-i.i.d. setting with $\alpha = 0.1$, it is more difficult to observe such a trend. Here, the strongest levels of upstream and downstream compression outperform the uncompressed FD. Thus, it appears that using quantization in both directions, upstream and downstream, is a promising technique for reducing communication.

V. Compressed Federated Distillation

In this section, we combine the insights of the previous section and propose Compressed Federated Distillation (CFD). CFD extends the conventional Federated Distillation framework by the following five techniques:

Algorithm 1: Compressed Federated Distillation

1 **init:** Set upstream and downstream precision b_{up} and b_{down}. Every client, C_i, holds a different local data set, $D_i = (X_i, Y_i)$, as well as the common public data set, X^{pub}, with size $|X^{pub}| = n$.
2 Iterate over T communication rounds:
3 **for** $t = 1, .., T$ **do**
4 Iterate over $|I_t|$ participating clients:
5 **for** $i \in I_t \subseteq \{1, .., [\text{Number of Clients}]\}$ ***in parallel*** **do**
6 Client C_i does:
7 • $\theta \leftarrow$ random_init() # Initialize
8 **if** $t > 1$ **then**
9 • $\text{download}_{S \rightarrow C_i}(\tilde{Y}_S^{pub})$
10 • $\theta \leftarrow \text{train}(\theta, X^{pub}, \tilde{Y}_S^{pub})$ # Distillation
11 **end**
12 • $\theta_i \leftarrow \text{train}(\theta, X_i, Y_i)$ # Local Training
13 • $Y_i^{pub} \leftarrow f_{\theta_i}(X^{pub})$ # Compute Soft-Labels
14 • $\tilde{Y}_i^{pub} \leftarrow \mathcal{Q}_{b_{up}}(Y_i^{pub})$ # Compress Soft-Labels
15 • $\text{upload}_{C_i \rightarrow S}(\tilde{Y}_i^{pub})$ # Upload
16 **end**
17 Server S does:
18 • $Y^{pub} \leftarrow \frac{1}{|I_t|} \sum_{i \in I_t} \tilde{Y}_i^{pub}$ # Aggregate
19 • $\theta_S \leftarrow \text{train}(\theta_S, X^{pub}, Y^{pub})$ # Server Distillation
20 • $Y_S^{pub} \leftarrow f_{\theta_S}(X^{pub})$ # Compute Soft-Labels
21 • $\tilde{Y}_S^{pub} \leftarrow \mathcal{Q}_{b_{down}}(Y_S^{pub})$ # Compress Soft-Labels
22 **end**
23 **return** θ_S

1) **Distill data curation (Alg. 1 - 1)**: We select a fixed random subset X^{pub} of the available distillation data for training. This subset is not varied over the course of training.
2) **Upstream quantization (Alg. 1 - 14)**: We reduce the bit-width of the client soft-labels by applying the constrained quantization operator $\mathcal{Q}$ (eq. (10)).
3) **Delta coding (Alg. 1 - 14)**: The quantized soft-labels are encoded using an efficient arithmetic entropy coding technique, like CABAC [54]. Additionally, we use delta coding (eq. (13)), to further reduce the entropy of the quantized soft-label information $\tilde{Y}_i$.
4) **Dual distillation (Alg. 1 - 19, 20)**: In every round, we distill a server model θ_S from the aggregated soft-labels. This server model accumulates training information from all previous communication rounds. The clients are then trained to match the predictions of this server model. This avoids loss of information in settings where clients do not participate in every round.
5) **Downstream quantization (Alg. 1 - 21)**: We apply constrained quantization $\mathcal{Q}$ also to the predictions of the server model before sending them down to the clients. The clients then, starting from a random initialization, are trained to mimic the predictions of the server model.

The training procedure is illustrated in Figure 8 and formally described in Algorithm 1.

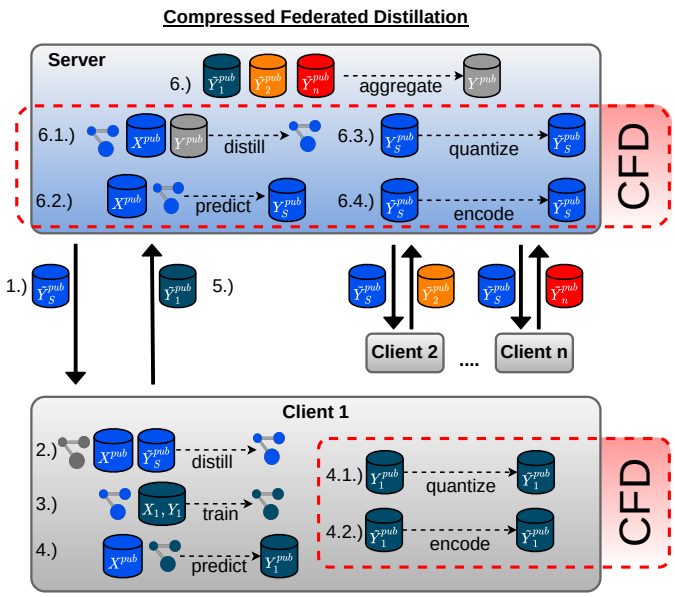


Fig. 8: Our proposed Compressed Federated Distillation method employs distill data curation (see Sec. IV-A), soft-label quantization (4.2., see Sec IV-B) and delta-coding (4.3., see Sec. IV-C) to minimize the communication from the clients to the server. Furthermore, CFD uses dual distillation (6.1., 6.2., see Sec. IV-D) to keep clients synchronized in situations when full client participation in every round can not be ensured. On top of that CFD also uses quantization (6.3.) and delta-coding (6.4.) in the downstream, to reduce the communication from the server to the clients.

The performance of our algorithm in every round t is determined from the distilled model θ_S on the validation data set.

VI. Experiments

In this section we empirically evaluate our proposed Compressed Federated Distillation method and compare its performance against the natural baselines of Federated Averaging [1] and Federated Distillation [11]. The experimental setup is given as follows:

Data sets and models: We evaluate CFD on both Federated image and text classification problems with large scale convolutional and transformer neural networks, respectively. For our image classification problems we experiment with the following combinations of client- and/ distillation data: (MNIST / EMNIST [55]) and (CIFAR-10 / STL-10 [56]). In both cases the distribution of the distillation data deviates from the one of the client data, as it would in realistic Federated Learning scenarios (MNIST contains handwritten digits, EMNIST contains handwritten characters, CIFAR-10 and STL-10 both contain different types of natural images). For our text classification problems we use disjoint splits of the SST2 [57] and AG-News [58] datasets for client training, distillation, and validation, respectively. We train LeNet- [59], VGG-type [60], AlexNet-type [61] and ResNet-type [16] architectures with and without batch-normalization layers. The Alexnet, ResNet-18 and VGG-16 models used in our experiments contain 23.2M, 11.1M, and 15.2M parameters respectively. For our text classification experiments we fine-tune DistilBERT [62], a popular transformer model with approximately 66 Million parameters.

Federated Learning environment and data partitioning: For image classification problems, we consider Federated Learning settings with 20 clients. In all experiments, we split the training data evenly among the clients according to a Dirichlet distribution $D(\alpha)$ following the procedure outlined in [50]. This technique allows us to smoothly adapt the level of non-iid-ness in the client data using the Dirichlet parameter α. We experiment with values for α varying between 100.0 and 0.01. A value of $\alpha = 100.0$ results in almost identical label distributions, while setting $\alpha = 0.01$ results in a split, where the vast majority of data on every client stems from one single class (see Figure 2 for an illustration). For image classifiers, we vary the clients' participation rate (in every round) between 40% and 100%, and train for 50 communication rounds. For language models, we set the number of clients to 10 and the participation rate to 100%, and train for a total of 10 communication rounds. As is standard convention in FL, the validation data follows the clients' training data distribution (and not the distribution of the distillation data).

Optimization details: For the sake of simplicity, in all image classification tasks, we use the popular Adam [63] optimizer with a fixed learning rate of 0.001 across all baselines and for both the distillation and training on local private data. While a dedicated selection of optimizer and optimization hyperparameters might improve performance, our goal here is to give a fair comparison between the different Federated Learning algorithms. For language models, we perform one epoch of distillation with Adam, using a learning rate of 1×10^{-5} and no weight decay. The clients' models in each round are trained for one local epoch with SGD with learning rate and momentum set to 0.001 and 0.9, respectively.

Baselines: We compare the performance of our method, Compressed Federated Distillation (CFD), with respect to the two natural baselines: Federated Averaging (FA) [1] and Federated Distillation (FD) [11]. For CFD, we test two configurations: For CFD-1-32 we only quantize the upstream communication by setting $b_{up} = 1$ and $b_{down} = 32$. For CFD-1-1 we quantize both the upstream and downstream communication and set $b_{up} = 1$ and $b_{down} = 1$. We also investigate the effects of using delta coding (as described in section IV-C). CFD methods that use delta coding are indicated by CFD$_\Delta$.

Evaluation Metrics: As is custom in communication-efficient Federated Learning literature [14][15][25], we report cumulative communication of the different FL methods. Given this general metric, other quantities of interest like wall-clock time or energy consumption can be approximated for any given hardware setup and/ or communication infrastructure. For the Baseline FD and our methods, CFD and CFD$_\Delta$, we measure only the communication of soft-labels Y^{pub} and explicitly ignore the communication cost of transferring the unlabeled public data set X^{pub} to the participating clients. While clients technically need to download this data once before training, it is not subject to the same constraints as the Federated

TABLE III: Upstream and downstream communication in [MB], necessary to achieve accuracy targets in Federated Learning on the CIFAR-10 dataset, across different neural network models and levels of data heterogeneity α. The Federated Learning setting consists of 20 clients and a participation rate of 40%. For the distillation based methods, 80000 data points from the STL-10 dataset are used as distillation data. The number of communication rounds, necessary to achieve the target accuracy is given in parenthesis. A value of "n.a." signifies that the method did not achieve the target accuracy within 50 communication rounds.

Model	Target Accuracy	α	Up/Down	FA	FD	CFD-1-32	CFD$_\Delta$-1-32	CFD-1-1	CFD$_\Delta$-1-1
ResNet-18	0.71	100.0	up	760.35 (17)	44.80 (14)	0.56 (17)	**0.40** (17)	1.36 (41)	0.82 (41)
			down	760.35 (17)	44.80 (14)	54.40 (17)	54.40 (17)	1.36 (41)	**0.39** (41)
	0.68	1.0	up	1028.71 (23)	48.00 (15)	0.37 (13)	**0.28** (13)	0.64 (22)	0.43 (22)
			down	1028.71 (23)	48.00 (15)	41.60 (13)	41.60 (13)	0.72 (22)	**0.34** (22)
	0.45	0.1	up	1520.70 (34)	16.00 (5)	0.09 (7)	**0.08** (7)	0.52 (41)	0.40 (41)
			down	1520.70 (34)	16.00 (5)	22.40 (7)	22.40 (7)	0.99 (41)	**0.92** (41)
VGG-16	0.8	100.0	up	671.16 (11)	32.00 (10)	0.40 (12)	**0.29** (12)	0.76 (23)	0.47 (23)
			down	671.16 (11)	32.00 (10)	38.40 (12)	38.40 (12)	0.76 (23)	**0.24** (23)
	0.78	1.0	up	1281.30 (21)	28.80 (9)	0.38 (13)	**0.28** (13)	0.56 (19)	0.37 (19)
			down	1281.30 (21)	28.80 (9)	41.60 (13)	41.60 (13)	0.62 (19)	**0.27** (19)
	0.48	0.1	up	2928.69 (48)	25.60 (8)	0.11 (9)	**0.09** (9)	0.43 (34)	0.35 (34)
			down	2928.69 (48)	25.60 (8)	28.80 (9)	28.80 (9)	0.77 (34)	**0.75** (34)
AlexNet	0.68	100.0	up	n.a.	89.60 (28)	0.94 (29)	**0.74** (29)	n.a.	n.a.
			down	n.a.	**89.60** (28)	92.80 (29)	92.80 (29)	n.a.	n.a.
	0.64	1.0	up	n.a.	38.40 (12)	0.61 (21)	**0.49** (21)	0.76 (26)	0.62 (26)
			down	n.a.	38.40 (12)	67.20 (21)	67.20 (21)	0.84 (26)	**0.42** (26)
	0.44	0.1	up	n.a.	6.40 (2)	0.09 (6)	**0.08** (6)	0.11 (7)	0.10 (7)
			down	n.a.	6.40 (2)	19.20 (6)	19.20 (6)	0.17 (7)	**0.15** (7)

TABLE IV: Upstream and downstream communication, measured in [MB], required in Federated fine-tuning of DistilBERT to achieve a specific target accuracy on the SST2 and AG-News datasets, at different levels of data heterogeneity α. The number of required communication rounds is given in parenthesis.

Dataset (Target Accuracy)	α	Up/Down	FA	FD	CFD-1-32	CFD$_\Delta$-1-32	CFD-1-1	CFD$_\Delta$-1-1
SST2	100.0	Up	267.820 (1)	0.269 (1)	**0.004** (1)	0.006 (1)	0.029 (7)	0.044 (7)
(0.88)	100.0	Down	267.820 (1)	0.269 (1)	0.269 (1)	0.269 (1)	**0.029** (7)	0.044 (7)
	1.0	Up	803.460 (3)	0.539 (2)	**0.008** (2)	0.012 (2)	0.038 (10)	0.057 (10)
	1.0	Down	803.460 (3)	0.539 (2)	0.539 (2)	0.539 (2)	**0.042** (10)	0.063 (10)
AG-News	100.0	Up	535.640 (2)	1.920 (2)	**0.030** (2)	0.035 (2)	**0.030** (2)	0.035 (2)
(0.91)	100.0	Down	535.640 (2)	1.920 (2)	1.920 (2)	1.920 (2)	**0.030** (2)	0.035 (2)
	1.0	Up	1071.280 (4)	4.800 (5)	0.142 (10)	0.168 (10)	**0.101** (7)	0.119 (7)
	1.0	Down	1071.280 (4)	4.800 (5)	9.600 (10)	9.600 (10)	**0.105** (7)	0.121 (7)

Learning process. In communication-sensitive applications, X^{pub} could already be stored on the devices long before the Federated training process starts, and thus, the timing of its communication is much less critical. Other work [12] also demonstrates that X^{pub} can be automatically generated on the clients using Generative Adversarial Networks.

A. Image Classification Results

We first investigate the communication properties of CFD on image classification benchmarks. Table III shows the amount of upstream and downstream bits, as well as the number of communication rounds, required to achieve fixed Accuracy targets for Alexnet, ResNet-18, and VGG-16 on CIFAR-10, at different levels of data heterogeneity between the clients. The corresponding training curves are given in Figure 9. As we can see, CFD is drastically more communication-efficient than the baselines FA and FD in all tested scenarios. For instance, for VGG-16 and $\alpha = 100.0$, CFD$_\Delta$-1-1 achieves a target Accuracy of 80% by cumulatively communicating only 0.47 MB on average, from the clients to the server, and only 0.24 MB on average, from the server to the clients. This is particularly remarkable, as one single transfer of the parameters of VGG-16 already takes up 61.01 MB. To achieve the same 80% Accuracy target, FA requires 671.16 MB of cumulative communication in both the upstream and the downstream, translating to more than three orders of magnitude in communication savings for CFD. When directly comparing with FD, which requires 32.00 MB, CFD still reduces the communication by about two orders of magnitude. Similar results can be observed for the two other tested neural networks, ResNet-18 and Alexnet. On Alexnet, FA even underperforms CFD w.r.t. to the maximum achieved Accuracy and misses the Accuracy target of 68%.

The communication savings are even larger in the non-iid settings with $\alpha = 0.1$, where FA is known to perform poorly

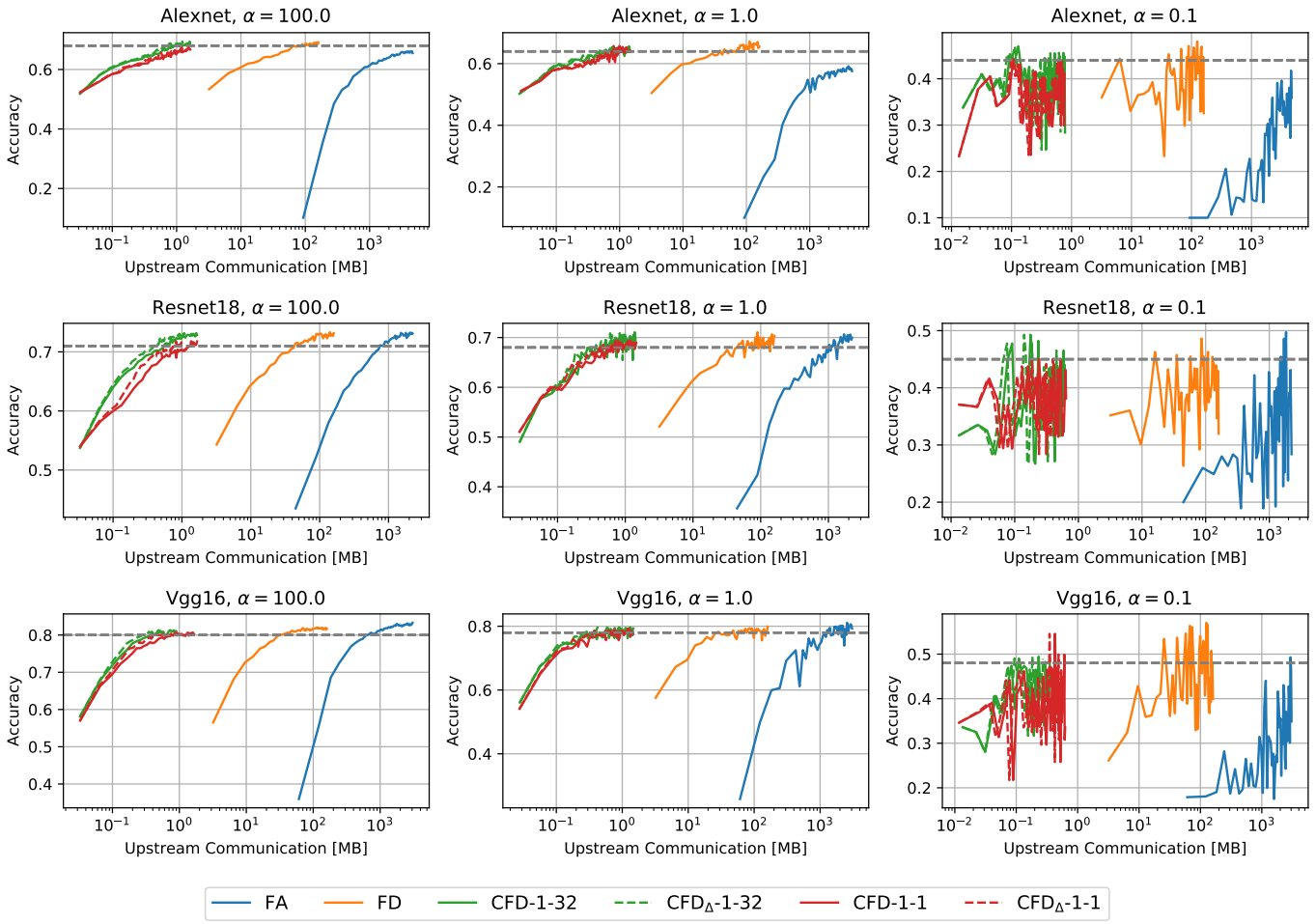
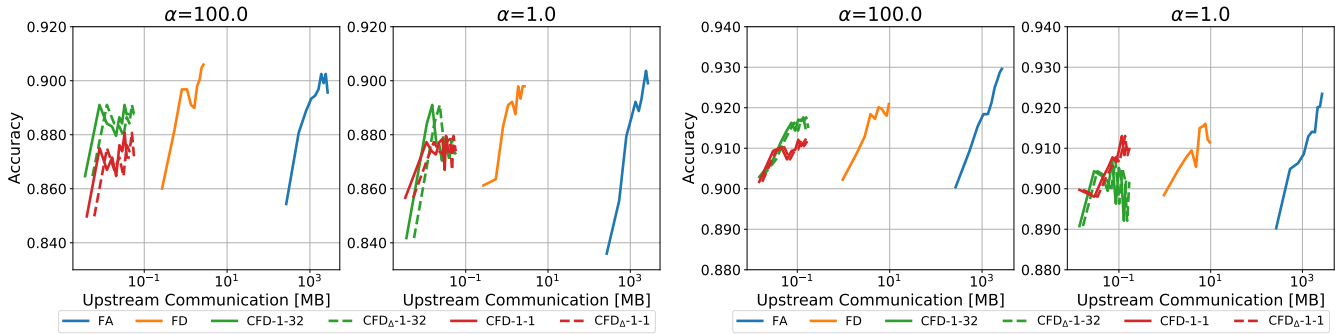


Fig. 9: Model performance as a function of communicated bits for our proposed CFD method and baselines FA and FD in Federated Learning on the CIFAR-10 dataset, across different neural network models and levels of data heterogeneity (α=100.0, 1.0, and 0.1). The federated learning setting consists of 20 clients with a participation rate of 40%. For the distillation based methods, 80000 data points from the STL-10 data set were used as distillation data.



(a) SST2.

(b) AG-News.

Fig. 10: DistilBERT model performance as a function of communicated bits for our proposed CFD method and baselines FA and FD on the SST2 and AG-News data sets. 10 clients are trained at a participation rate of 100%.

[25]. For instance, when training ResNet-18 at $\alpha = 0.1$, FA requires 1520.70 MB to achieve the Accuracy target of 45%. In contrast, CFD$_\Delta$-1-32 requires only 0.08 MB to achieve the same Accuracy, corresponding to a reduction in communication by a factor of $\times$19943.

In all investigated settings, CFD$_\Delta$ methods that use delta

coding are more efficient than those that do not. For instance, for VGG-16 and $\alpha = 1.0$, delta coding can bring down the cumulative upstream-communication required to achieve 78% Accuracy from 0.38 MB to 0.28 MB for CFD-1-32. On the same benchmark, delta coding also reduces the cumulative upstream-communication from 0.56 MB to 0.37 MB for CFD-1-1.

As can be seen in Figure 9, the heavily compressed CFD can keep up with the uncompressed baselines FD and FA w.r.t. maximum achieved Accuracy on most benchmarks. Additional experimental results for Federated Learning scenarios with heterogeneous model architectures can be found in Supplement A and are in line with those obtained when using homogeneous model architectures.

B. Language Model Results

Figure 10 shows the convergence speed in terms of communicated bits for different baseline methods on Federated language modelling tasks. We fine-tune, DistilBERT, a popular large-scale transformer model, on the SST2 and AG-News data sets. In these experiments, we consider a Federated Learning setting with 10 clients, 100% participation rate, and total of 10 communication rounds. From this data, we can highlight five important observations. First, while FA tends to achieve slightly higher total Accuracy than the other methods, it also requires several orders of magnitude more upstream communication. Second, FD reduces the communication overhead with respect to FA by $\times 996$ and $\times 279$ in SST2 and AG-News data sets, respectively, at the expense of no more than 2% Accuracy degradation. Third, CFD-1-32 and CFD-1-1 stand out as the most efficient techniques. When compared to FA, CFD achieves communication savings of up to $\times 66955$ on the SST2 data set and $\times 17855$ on the AG-News data set. Fourth, we notice that in this particular set of experiments, delta-coding (see CFD_{Δ}-1-1 and CFD_{Δ}-1-32), slightly increases the communication overhead with respect to regular CFD (in both cases, CFD-1-1 and CFD-1-32). This effect is caused by the small number of classes in the data sets (i.e., SST2 contains 2 classes, while AG-News contains 4 classes), which limits the benefits of delta-coding. Finally, the experimental findings on i.i.d. ($\alpha = 100.0$) and non-i.i.d. ($\alpha = 1.0$) data, show that CFD is robust to changes in the clients' data heterogeneity.

Table IV shows the upstream and downstream communication cost (in MB) necessary to achieve certain Accuracy targets across different levels of data heterogeneity, as well as the required number of communication rounds. As we can see, similar as on the image classification problems, CFD requires several orders of magnitude less communication than FA and FD in both the upstream and downstream to achieve these performance targets. In some situations, this comes at the cost of an increased number of total communication rounds.

For additional results on the effect of the distillation data set size on the performance of CFD, we refer the reader to the supplementary materials B.

VII. Conclusion

In this work we have explored the communication properties of Federated Distillation and shown that drastic compression gains are possible. For instance, on language modelling tasks, we demonstrated that our proposed Compressed Federated Distillation method can reduce the cumulative communication necessary to achieve fixed performance targets from 1071.28 MB to 0.101 MB when compared to the very popular Federated Averaging algorithm. This corresponds to a reduction in communication by over four orders of magnitude. Similar compression rates were obtained in our investigated image classification problems on popular convolutional neural networks. We believe that our findings will help the widespread adoption of Federated Learning in heavily distributed and/or resource-constrained settings.

It is important to note however, that the favorable communication properties of all Federated Distillation methods, like the ones reported in this paper, come at the cost of additional computational overhead caused by the local distillation. This additional computational overhead might be challenging in Federated Learning environments where clients have limited computational resources, or where the number of clients is high and/or the number of data points per client is low. It thus needs to be carefully considered for every application, which of the two paradigms - Federated Averaging or Federated Distillation - is more suitable for the problem at hand.

VIII. Future Work

Federated Distillation is a very promising new way of solving Federated Learning problems, but many aspects are still not fully understood. While it's unique communication properties and the added option for clients to train different local models could make it a popular choice for Federated Learning applications, it is also lacking formal robustness and convergence guarantees so far. Future work could address these open problems and also explore personalization techniques for FD via meta- or multi-task learning [64][49]. Moreover, lazy aggregation mechanisms, as proposed in [65][66], could further improve efficiency of Federated Distillation methods.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[4] C. Ma, J. Li, M. Ding, H. H. Yang, F. Shu, T. Q. Quek, and H. V. Poor, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, 2020.

[5] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, 2020.

[6] F. Sattler, T. Wiegand, and W. Samek, "Trends and advancements in deep neural network communication," *arXiv preprint arXiv:2003.03320*, 2020.

[7] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018. [Online]. Available: http://arxiv.org/abs/1811.11479

[8] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[9] J.-H. Ahn, O. Simeone, and J. Kang, "Wireless federated distillation for distributed edge learning with heterogeneous data," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2019, pp. 1–6.

[10] S. Oh, J. Park, E. Jeong, H. Kim, M. Bennis, and S.-L. Kim, "Mix2fld: downlink federated learning after uplink federated distillation with two-way mixup," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2211–2215, 2020.

[11] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data," *arXiv preprint arXiv:2008.06180*, 2020.

[12] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *arXiv preprint arXiv:2006.07242*, 2020.

[13] H. Chen and W. Chao, "FedDistill: Making bayesian model ensemble applicable to federated learning," *arXiv preprint arXiv:2009.01974*, 2020.

[14] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[15] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[18] P. Kidger and T. J. Lyons, "Universal approximation with deep narrow networks," in *Conference on Learning Theory (COLT)*, ser. Proceedings of Machine Learning Research, vol. 125, 2020, pp. 2306–2327.

[19] K. F. E. Chong, "A closer look at the approximation capabilities of neural networks," in *8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=rkevSgrtPr

[20] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. X. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019, pp. 103–112.

[21] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[22] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 2, 1990, pp. 598–605.

[23] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 440–445.

[24] F. Sattler, S. Wiedemann, K. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," in *International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[25] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, 2020.

[26] M. Courbariaux, Y. Bengio, and J. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015, pp. 3123–3131.

[27] F. Li, B. Zhang, and B. Liu, "Ternary weight networks," *arXiv preprint arXiv:1605.04711*, 2016.

[28] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[29] J. Xu, W. Du, R. Cheng, W. He, and Y. Jin, "Ternary compression for communication-efficient federated learning," *arXiv preprint arXiv:2003.03564*, 2020.

[30] D. Neumann, F. Sattler, H. Kirchhoffer, S. Wiedemann, K. Müller, H. Schwarz, T. Wiegand, D. Marpe, and W. Samek, "DeepCABAC: Plug & play compression of neural network weights and weight updates," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 21–25.

[31] S. Wiedemann, H. Kirchhoffer, S. Matlage, P. Haase, A. Marbán, T. Marinc, D. Neumann, T. Nguyen, H. Schwarz, T. Wiegand, D. Marpe, and W. Samek, "DeepCABAC: A universal compression algorithm for deep neural networks," *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 4, pp. 700–714, 2020.

[32] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "Fedaux: Leveraging unlabeled auxiliary data in federated learning," *arXiv preprint arXiv:2102.02514*, 2021.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[34] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[35] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 603–618.

[36] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?" *arXiv preprint arXiv:2003.14053*, 2020.

[37] H. Seo, J. Park, S. Oh, M. Bennis, and S. Kim, "Federated knowledge distillation," *arXiv preprint arXiv:2011.02367*, 2020.

[38] I. Bistritz, A. J. Mann, and N. Bambos, "Distributed distillation for on-device learning," in *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[39] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," *arXiv preprint arXiv:1902.11175*, 2019.

[40] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer," *arXiv preprint arXiv:1912.11279*, 2019.

[41] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8851–8855.

[42] ——, "Uveqfed: Universal vector quantization for federated learning," *IEEE Transactions on Signal Processing*, 2020.

[43] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[44] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," in *Proceedings of 8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020.

[45] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

[46] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 4424–4434.

[47] H. Wu, C. Chen, and L. Wang, "A theoretical perspective on differentially private federated multi-task learning," *arXiv preprint arXiv:2011.07179*, 2020.

[48] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, "Robust federated learning in a heterogeneous environment," *arXiv preprint arXiv:1906.06629*, 2019.

[49] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2020.

[50] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[51] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[52] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of quantization," *IEEE Transactions on instrumentation and measurement*, vol. 45, no. 2, pp. 353–361, 1996.

[53] K. Sayood, *Introduction to data compression*, 5th ed. Morgan Kaufmann, 2017.

[54] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, 2003.

[55] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: An extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[56] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 215–223.

[57] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013, pp. 1631–1642.

[58] X. Zhang, J. J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015, pp. 649–657.

[59] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 2, 1989, pp. 396–404.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, 2012, pp. 1097–1105.

[62] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[65] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "Lag: Lazily aggregated gradient for communication-efficient distributed learning," *arXiv preprint arXiv:1805.09965*, 2018.

[66] C. B. Issaid, A. Elgabli, J. Park, and M. Bennis, "Communication efficient distributed learning with censored, quantized, and generalized group admm," *arXiv preprint arXiv:2009.06459*, 2020.

Felix Sattler received a M.Sc. degree in computer science, a M.Sc. degree in applied mathematics and a B.Sc. degree in Mathematics all from Technische Universität Berlin. He is currently with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Berlin, Germany. His research interests include distributed machine learning, neural networks and multi-task learning.

Arturo Marban has a B. Eng. degree in Mechatronics (2007) and an M.Sc. degree in Manufacturing Systems (2010) from the Monterrey Institute of Technology and Higher Education, Monterrey, Mexico. In 2018, he received a Ph.D. degree in Automatic Control, Robotics, and Computer Vision from the Polytechnic University of Catalonia, Catalonia, Spain. Afterward, in the same year, he joined the Artificial Intelligence Department at Fraunhofer Heinrich Hertz Institute, Berlin, Germany, where he currently works as a post-doctoral researcher. His research interests include machine learning and neural networks, efficient deep-learning, and computer vision.

Roman Rischke received the M.Sc. degree in business mathematics from Technische Universität Berlin, Berlin, Germany, in 2012, and the Dr. rer. nat. degree in mathematics from Technische Universität München, Munich, Germany, in 2016. He currently works as a post-doctoral researcher in the Artificial Intelligence Department at Fraunhofer Heinrich Hertz Institute, Berlin, Germany. His research interests include discrete optimization under data uncertainty, robust and trustworthy machine learning as well as distributed learning.

Wojciech Samek (M'13) is head of the Department of Artificial Intelligence and the Explainable AI Group at Fraunhofer Heinrich Hertz Institute, Berlin, Germany. He studied computer science at Humboldt University of Berlin, from 2004 to 2010, and received the Ph.D. degree with distinction from the Technical University of Berlin in 2014. During his studies he was awarded scholarships from the German Academic Scholarship Foundation and the DFG Research Training Group GRK 1589/1, and was a visiting researcher at NASA Ames Research Center, Mountain View, USA. In 2014 he founded the Machine Learning Group at Fraunhofer HHI, which he has directed until 2020. Dr. Samek is associated faculty at the Berlin Institute for the Foundation of Learning and Data (BIFOLD), the ELLIS Unit Berlin and the DFG Graduate School BIOQIC. Furthermore, he is an editorial board member of Pattern Recognition, PLoS ONE and IEEE TNNLS, and an elected member of the IEEE MLSP Technical Committee. He has been serving as an AC for NAACL'21, was a recipient of multiple best paper awards, including the 2020 Pattern Recognition Best Paper Award, and a part of the MPEG-7 Part 17 standardization. His research interest include deep learning, explainable AI, neural network compression, and Federated Learning.

CFD: Communication-Efficient Federated Distillation via Soft-Label Quantization and Delta Coding
- Supplementary Materials -

A. Federated Distillation with Heterogeneous Model Architectures

Our proposed Compressed Federated Distillation approach supports training of heterogeneous model architectures. To evaluate our method in this setting, we simulate a Federated Learning scenario with 30 clients, 10 of which each training ResNet-8, VGG-16 and Alexnet respectively. The results are shown in Figure 11 and are in line with our findings for homogeneous models architectures, which CFS-1-1 (resp. $\text{CFD}_{\triangle}$-1-1) performing en par with uncompressed Federated Distillation across all levels of data heterogeneity.

B. Distillation Dataset size in NLP Tasks

Table V describes the effect of the distillation data set size in the server model performance, during upstream communication. Specifically, 50%, 20%, and 10% of the distillation data set samples are processed, and we report the upstream communication cost (in MB) necessary to achieve a certain target Accuracy. For the SST2 data set, the target Accuracy was set to 0.88, while for the AG-News data set, to 0.91. On the other hand, Figure 12 shows the complete communication cost dynamics (i.e., Accuracy vs. communication cost at every communication round) for these experiments.

*Corresponding authors: F. Sattler and W. Samek.

†This work was supported by the Federal Ministry of Education and Research (BMBF) through the BIFOLD - Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037I), and the EU's Horizon 2020 project COPA EUROPE.

‡F. Sattler, A. Marban, R. Rischke, and W. Samek are with the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: felix.sattler@hhi.fraunhofer.de, wojciech.samek@hhi.fraunhofer.de).

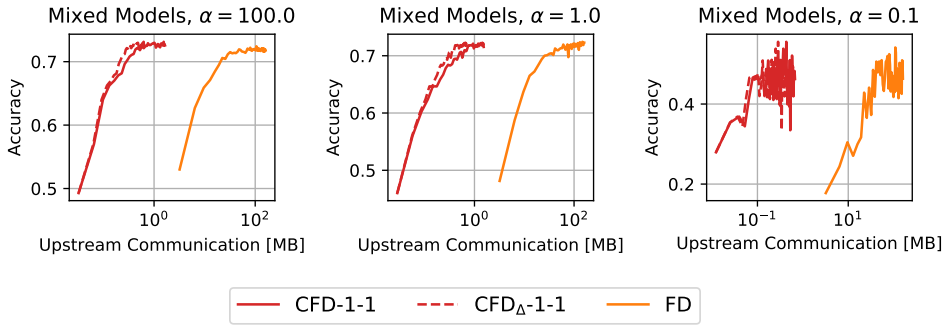
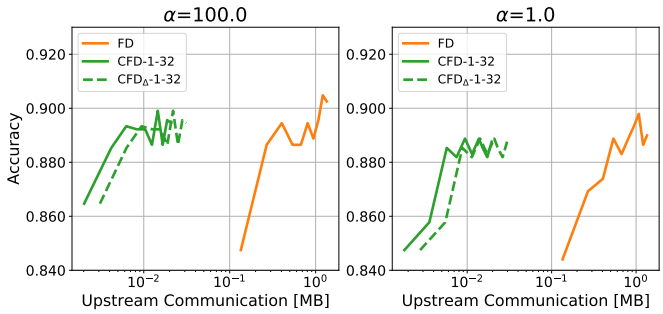


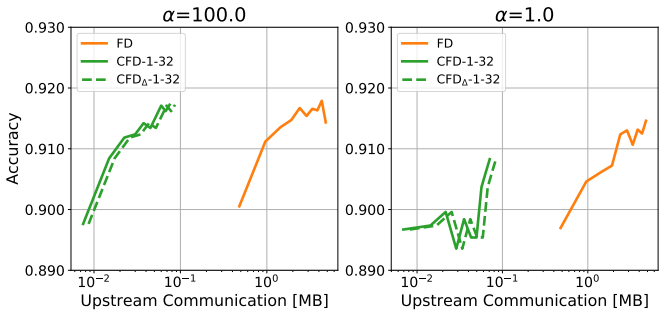
Fig. 11: Model performance as a function of communicated bits for our proposed CFD method and baseline FD in Federated Learning on the CIFAR-10 data set at different levels of data heterogeneity α. The Federated Learning setting consists of a total of 30 clients, 10 of which each training ResNet-8, VGG-16 and Alexnet respectively. 40% of clients participate in every round and 80000 randomly selected data points from the STL-10 data set are used for distillation.

TABLE V: Upstream communication, measured in [MB], required in federated training of DistilBERT to achieve a specific target accuracy on the SST2 and AG-News datasets, using different numbers of distillation data samples (50%, 20%, and 10%) and levels of data heterogeneity ($\alpha = 100.0$ and 1.0). The values inside the parenthesis (next to the communication costs) correspond to the number of communication rounds.

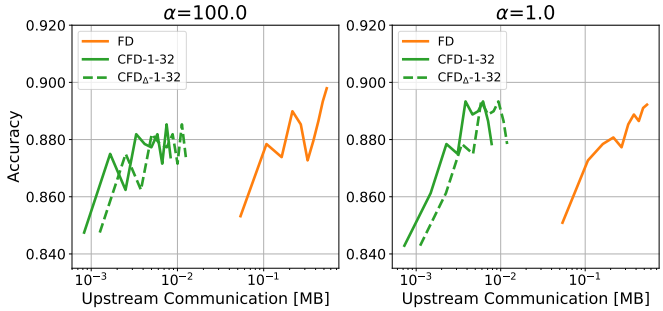
Dataset (Target Accuracy)	Distillation Dataset Size	α	FA	FD	CFD-1-32	CFD_Δ-1-32
SST2 (0.88)	50%	100.0	267.820 (1)	0.135 (1)	0.002 (1)	0.003 (1)
		1.0	803.460 (3)	0.404 (3)	0.004 (2)	0.006 (2)
	20%	100.0	267.820 (1)	0.162 (3)	0.003 (3)	0.004 (3)
		1.0	803.460 (3)	0.162 (3)	0.003 (4)	0.005 (4)
	10%	100.0	267.820 (1)	0.135 (5)	0.001 (2)	0.001 (2)
		1.0	803.460 (3)	0.162 (6)	0.004 (10)	0.006 (10)
AG-News (0.91)	50%	100.0	535.640 (2)	0.480 (1)	0.015 (2)	0.017 (2)
		1.0	1071.280 (4)	1.920 (4)	0.071 (10)	0.084 (10)
	20%	100.0	535.640 (2)	0.384 (2)	0.003 (1)	0.003 (1)
		1.0	1071.280 (4)	1.536 (8)	0.029 (10)	0.034 (10)
	10%	100.0	535.640 (2)	0.384 (4)	0.007 (5)	0.009 (5)
		1.0	1071.280 (4)	0.960 (10)	0.014 (10)	0.017 (10)



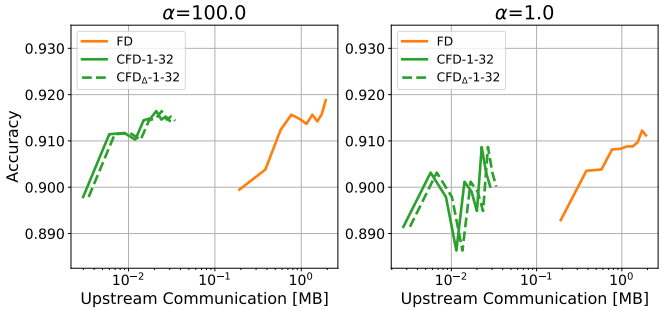
(a) SST2 (50% distillation data samples).



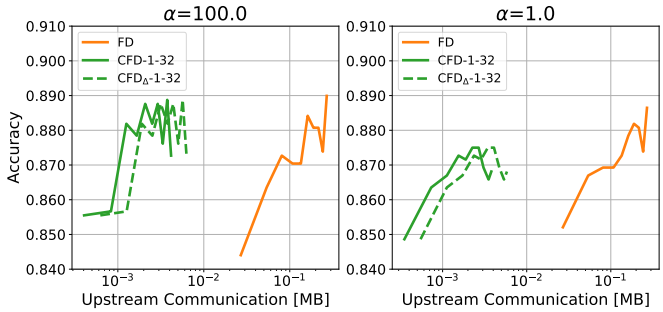
(b) AG-News (50% distillation data samples).



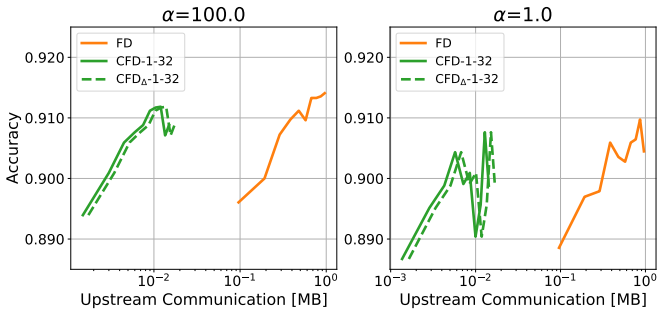
(c) SST2 (20% distillation data samples).



(d) AG-News (20% distillation data samples).



(e) SST2 (10% distillation data samples).



(f) AG-News (10% distillation data samples).

Fig. 12: Communication efficiency (i.e., accuracy vs communication cost) for Federated Learning of DistilBERT, on the SST2 and AG-News datasets, with α = 100.0 and 1.0, using 50%, 20%, and 10% of the disitillation dataset samples (10 clients with 100% participation rate).