Bidirectional Hierarchical Federated Optimisation

Alex Iacob
aai30@cam.ac.uk

1 Overview

Federated Learning (FL) is a distributed Machine Learning (ML) paradigm allowing multiple clients to train a shared collaborative model without communicating private data. It was introduced by McMahan et al. [45] as a means of reducing communication costs and lessening the privacy concerns of storing sensitive data in a centralised location, following the principles of focused collection and data minimisation outlined in the White House [61] privacy report. These properties have led to FL applications with large cohorts of small edge devices, such as mobile keyboard prediction [16] for Android phones, and settings with larger entities subject to privacy requirements, such as hospitals [53]. These two settings are distinguished by Kairouz et al. [26] as cross-device and cross-silo FL.

The growth in the preponderance of Federated Learning since the publication of McMahan et al. [45] can be ascribed to two primary trends. First, an increase in the privacy requirements of consumers and legal frameworks has put pressure on technology companies. This pressure drove interest in privacy-preserving ML at major corporations such as Google [45, 16, 14, 25], Microsoft [58, 11], Meta [21, 46], and Apple [49]. Second, ML has extended to domains with strict privacy requirements such as healthcare [53, 51, 48], Human Activity Recognition (HAR) [54, 47] or collaborations between competing corporations [64, 42]. Moreover, the emergence of Large Language Models (LLMs) [4] has made accessing private language corpora advantageous, leading to the development of Federated Natural Language Processing (NLP) [39]. Similarly, the release of openly available LLM pre-trained weights [57] allows collaboration between entities with low computational resources using FL frameworks [3, 32, 17].

While the field has enjoyed abundant scientific and industry attention, the privacy and communication benefit it provides cause significant challenges in efficiently scaling and evolving federated systems. Crucially, the compromise of training a single global model is unsuitable when unusual clients require partial or complete personalisation of the model to their local data distribution.

This work proposes addressing the challenges above by constructing hierarchical tree-like federated network structures that allow bidirectional and potentially cyclical dataflow where each leaf is a client, and each internal node is a server capable of training on proxy public data. As a result, levels in the tree closer to the leaves are more personalised to the specific client population of a subtree, and those closer to the root provide more generalisable models. We call this approach Bidirectional Hierarchical Federated Learning (B-HFL). Furthermore, we allow leaf clients in these structures to execute asynchronous training using persistent models to account for temporal shifts in their data distributions to facilitate evolution.

1.1 Motivation

In its standard form, FL operates directly on clients using a centralised server to distribute model parameters and then aggregate them after client training; this process is repeated for multiple rounds. However, data in FL is subject to attributes such as client geographic location, sensor hardware, and behaviour. Due to these factors, the federated distribution violates the Independent and Identically Distributed (IID) assumption. Such *data heterogeneity* [26, sec. 3.1] is interwoven with *systems heterogeneity* [26, sec. 7.2] since clients have different computational abilities and network speeds. Additionally, the communication costs of transmitting model parameters between servers and clients are non-trivial. Since data heterogeneity makes obtaining a single global model efficient on all client data distributions unfeasible, we are concerned with creating arbitrary levels of personalisation in the form of Hierarchical Federated Learning in a manner that improves learning efficiency and allows such systems to evolve.

1.1.1 Efficiency

Efficiency and scalability have been at the centre of FL research since Hard et al. [16] applied FL to mobile keyboard prediction at Google. Building on top of Hard et al. [16], Bonawitz et al. [6] showed that

FL could be used to train models over tens of millions of smartphones. However, despite the optimistic billion-device forecasts of Bonawitz et al. [6], several limitations to the efficiency of FL emerged. These limitations are threefold: (a) synchronous FL can only effectively use hundreds of devices every round, (b) federated training is considerably slower than centralised training, (c) user devices are unreliable, leading to dropout and stragglers. These limitations received further attention in the empirical evaluation of Charles et al. [7].

Charles et al. [7] show that the performance of FL does not scale as expected when the number of clients trained every round increases despite previous theoretical work [28] indicating the contrary. Their experimental results show that the primary limitation of increasing cohort size under Non-IID settings is the miss-alignment of client models, indicated by a near-zero cosine similarity between updates. This miss-alignment limits the impact of each round, causes diminishing returns to increasing cohort size, and results in an inability to learn efficiently from client data in parallel. Thus, given that FL algorithms are highly parallel, scalability in FL is strongly limited by the ability to learn from clients on a per-sample basis. Furthermore, while the original investigations of Bonawitz et al. [6], Charles et al. [7] were cross-device, the problem of efficiently learning from clients also applies to cross-silo settings.

1.1.2 Evolution

The datasets of clients forming a federated network are generally not static. Clients may delete data immediately after generation, periodically, or ad-hoc based on memory needs or owner requests. Furthermore, the characteristics of newly added data can shift over time in either a gradual or immediate manner. For example, in Image Recognition tasks, seasonal transitions can shift captured images slowly, while changing locations or upgrading the camera hardware may lead to discrete changes. This problem is known as dataset shift [26, sec. 3.1] and represents *in-client* heterogeneity rather than the more common *cross-client* heterogeneity. Synchronous Federated Learning algorithms [45, 50, 59, 33, 34] assume that the clients only operate on the federated model received at the start of a round. Even works which maintain persistent local models, such as (Ditto) [35], assume that this persistent model is only used within FL rounds. Thus, current approaches cannot capture changes in the data distribution of a client. Asynchronous Federated Learning systems [62, 46, 8], such as Meta's PAPAYA [21], do allow clients to train outside round boundaries. However, they similarly assume that clients only train on the latest copy of the federated model they can access when they pull from the server.

1.2 Proposal Summary

This proposal builds upon the work done by Iacob et al. [22] and Iacob et al. [23] on personalised and hierarchical FL. The proposed system communicates data in a tree-based structure as shown in Fig. 1. Crucially, model parameters can flow bidirectionally, and nodes can apply partial updates from their parents via aggregation. Furthermore, each node can weight children and parent parameters differently while using methods such as the adaptive server optimisers [50] or training-based methods [35, 30, 67, 66]. Adaptive algorithms are particularly relevant as they allow each node in the tree to distinguish itself based on its previous state without necessitating additional parameter tuning. Finally, in the case where client cohorts are meaningfully clustered, this structure may allow a drastic increase in the sample efficiency of the system as each cluster decides how to optimise the generalisation-personalisation trade-off [2]. The potential contributions to the field include:

1. A family of efficient and scalable hierarchical FL algorithms allowing fine-grained control over personalisation and generalisation from the global root to fully-personalised leaves.

2. The investigation of three complimentary techniques enabled by such hierarchical structures: (a) allowing leaf clients to maintain persistent local models training asynchronously to tackle dataset shift, (b) making any node in the tree capable of training with a proxy dataset to inject more general information, (c) constructing additional vertical connections in the tree similar to residual connections [18] to allow highly customisable dataflow without changing the underlying communication infrastructure.

3. Extensive empirical evaluations considering scenarios with or without meaningful client clusters in language and speech recognition tasks leading to publications at ICLR or MLSys. This publication will be followed up by a work intended for MobiCom investigating asynchronous training on resource-constrained devices with dataset shift using the Raspberry Pi FL cluster at Cambridge ML Systems.

2 Previous work

The proposal in this document emerged as a natural consequence of research on Personalised Federated Learning and Hierarchical Federated Learning I began during my MPhil in Advanced Computer Science and the first year of my PhD.

Iacob et al. [22] investigated the trade-off between generalisation and personalisation, which is at the heart of this work, from the perspectives of Fair Federated Learning and its interactions with local adaptation (fine-tuning) of the federated model post-training. Since Fair Federated Learning attempts to construct a more uniform accuracy distribution for the federated model over the local test sets of clients, the expectation was to either reduce the need for personalization or to provide a better starting point from which to carry it out. The experimental results showed that Fair FL brings no benefits and potential downsides towards later personalization and led to the proposal of a Personalisation-aware FL algorithm that attempts to anticipate the common regularises used during fine-tuning throughout the FL process.

Iacob et al. [23] evaluated the performance of Federated Human Activity Recognition [54] when trained using multimodal data gathered from different sensor types at increasing levels of privacy. It showed that grouping clients based on the type of sensor that produced their training set effectively mitigated the impacts of privacy being required at a human subject, environment, and sensor level simultaneously. It was a direct precursor to Bidirectional Hierarchical Federated Learning as it relied on a two-tiered model structure where each client trained both a group-level model and the global federated model using a mutual learning approach [67]. This work was later extended to consider the adaptability of such two-tiered systems to the addition of a new sensor type (group) into the federated; the extension was submitted to the MobiUK symposium. Mutual learning was chosen to relate the group-level and global models since it allowed divergent architectures that only shared the output layer. However, despite its success, this training method requires clients to have a high amount of data and local epochs to train both models. The expensive nature of the procedure prompted a move towards a model-averaging approach.

Both of the previous works were implemented in the Flower [3] FL framework; however, the scale of experimentation required for fully validating B-HFL would be unfeasible on the publicly available simulation engine. As such, I have contributed to research on a new engine that doubles Flower simulations' throughput by intelligent ML-based client placement on GPUs. The paper has"High-throughput Simulation of Federated Learning via Resource-Aware Client Placement" has been submitted to Mobicom and is pending review. All the mentioned works are available as appendices to this proposal.

3 Background and Related Work

The standard FL objective can be modelled as seen in Eq. (1)

$$\min_{\theta} F(\theta) = \sum_{c \in C} p_c F_c(\theta) \ , \tag{1}$$

where F is the federated objective, C is the client set, θ is the model, and F_c is the loss of client c weighted by their fraction of the total number of examples p_c. This formulation assumes that a single global model is being trained without regard for the distribution of its performance across client datasets. Federated Averaging (FedAvg) [45] trains the global model locally on clients, for each round t it sums the update $\theta_t^c - \theta_t$ from client c weighted by p_c with the previous model θ_t using learning rate η, as seen in Eq. (2)

$$\theta_{t+1} = \theta_t + \eta \left(\sum_{c \in C} p_c \left(\theta_t^c - \theta_t \right) \right) \ . \tag{2}$$

The inability to colocate client data and the need to construct rough mixtures of model parameters as a compromise represent the leading causes of FL-specific challenges.

3.1 Heterogeneity

Non-IID data has been shown to impact both practical accuracies [68, 19] and theoretical convergence bounds [36]. It is thus worth detailing some forms of heterogeneity that Kairouz et al. [26] identify. The most commonly addressed form is quantity skew caused by clients having different amounts of data available. Standard FL algorithms effectively address Quantity skew via a simple reweighing (Eq. (2)). The other frequently-considered type of heterogeneity is label-distribution skew which is quantity skew per class. While these forms of heterogeneity have been most investigated, situations where features and labels are not related in the same manner across clients are far more pathological and may require some form of clustering or personalisation to tackle. In the worst-case scenario, each client may represent an entirely different task, as in Multi-Task Learning, with potentially no overlap in their solution space.

System (hardware) heterogeneity Devices within the federated network may differ regarding computational ability, storage, network speed, and reliability. They may also differ from themselves at a different point in time as their battery power, network connection, or operational mode vary. Importantly, variations in data-generating hardware, such as sensors, are linked to data heterogeneity. However, system heterogeneity and device unreliability harm the FL process independently of data. For example, slower hardware may result in straggling clients which elongate rounds in synchronous FL or operate on stale parameters in asynchronous FL. In addition, network or device unreliability creates dropout, which requires oversampling clients [6] and harms the effectiveness of maintaining client state across rounds.

Dataset Shift and Continual Learning Allowing ML models to participate in lifelong learning effectively is the goal of continual learning [10]; however, applying continual learning to the FL context is problematic for two primary reasons. First, the optimisation objective (Eq. (1)) intends to find a compromise model across all clients and cannot precisely fit all their data. Consequently, if the dataset of one client shifts independently of the whole network, the federated model will find it hard to adapt. Second, continual learning techniques such as Elastic-weight Consolidation [31], PackNet [43], and Learning without Forgetting [37] are designed for task-incremental settings where class labels are known, small amounts of previous data may still be available for specialised use cases [31], and there may even be different output heads for each task. The privacy requirements of FL make such solutions difficult at the level of the federated network without the addition of persistent local storage.

3.2 Federated Learning Efficiency

It is now worth expanding on the trends that Charles et al. [7] discovered. Those that limit the efficiency of FL in Non-IID settings where clients perform multiple SGD steps are of particular interest. Three significant effects can be observed. First, highly heterogeneous clients may cause sudden reductions in accuracy when their models are aggregated. Second, larger cohorts bring diminishing improvements in final accuracy and speed of convergence. Third, larger cohorts decrease data efficiency as more examples are needed for every accuracy gain.

These behaviours are approximately analogous to the well-known efficiency and generalisation limitations of large-batch training in centralised ML [27]. Charles et al. [7] find that data efficiency issues are caused by decreasing pseudo-gradient norms with increased cohort sizes and by the near-orthogonality of client updates following multiple steps of local training. The authors also find that adaptive optimisers fare better as cohort sizes grow due to scale invariance, making them particularly attractive aggregation algorithms.

3.2.1 Adaptive Federated Optimisation

Of particular relevance to this proposal are Federated Averaging with Server Momentum (FedAvgM) [20] and the more general Federated Adaptive Optimisation (FedOPT) [50]. They extend the concepts of momentum and adaptive optimisation [12, 29, 52] to Federated Learning on the *server-side* by treating client updates as pseudo-gradients and maintaining information across rounds on server-side accumulators. This structure allows such strategies to minimise the impact of individual rounds by averaging their pseudo-gradients and derived quantities with those of previous rounds. Since the outcome of individual rounds is highly variable based on the combination of clients selected and the model's current state, such techniques offer a more consistent optimisation trajectory.

Specifically, following the account provided by Reddi et al. [50] as shown in Eq. (3)

$$\Delta_t = \frac{1}{|C|}\sum_{c\in C}(\theta_t^c - \theta_t) \tag{3a}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\Delta_t \tag{3b}$$

$$v_t = \beta_2 v_t + (1-\beta_2)\Delta_t^2 \tag{3c}$$

$$\theta_{t+1} = \theta_t + \eta\frac{m_t}{\sqrt{v_t}+\tau} \tag{3d}$$

for a given round t and federated model θ_t each client c in the selected set C trains the model locally to construct a personalised version θ_t^c. The pseudo-gradient Δ_t is then computed by averaging the differences between these personalised and federated models as shown in Eq. (3a). All operations on tensors are element-wise including division between tensors.

The first-moment accumulator m_t can then be constructed as the weighted average of the previous accumulator m_t and Δ_t using weight β_1 as shown in Eq. (3b). Thus, the pseudo-gradient of the current round is smoothed by those of the previous rounds decayed using β_1. Similarly, for the version of FedOpt based on Adam [29] the second-moment accumulator v_t keeps track of the element-wise second power of the pseudo-gradient denoted by Δ_t^2 as shown in Eq. (3c). These two accumulators are then used to compute the updated model for the next round θ_{t+1} using the server learning rate η as shown in Eq. (3d). Notably, the term $\sqrt{v_t}$ refers to the element-wise square root; it is used to normalise model parameters and make the algorithm scale-invariant to the pseudo-gradient. Finally, τ controls the adaptivity of FedOPT.

FedOPT presents several promising properties in the context of hierarchical FL. First, Reddi et al. [50] show it is highly resilient to the exact choice of hyperparameters, including learning rate, compared to standard FedAvg and FedAvgM. Second, their scale-invariance partially addresses the issues observed by Charles et al. [7] regarding the near-zero pseudo-gradients caused by the near-orthogonality of client updates. Third, they provide a means of automatically differentiating the learning rates of multiple servers based on the state of their accumulators without having to carry out hyperparameter tuning.

3.3 Related Work

To tackle the inherent trade-off between optimising for the average global performance versus the performance on the data of a specific client which can be seen in Eq. (1), two overall directions emerged in the literature. The first, exemplified by Fair Federated Learning [33], attempts to modify the importance of a client in the federated objective function to change the final model's effectiveness for that client. The second relaxes the single global model requirement by personalising the federated model [65, 55, 68], maintaining persistent fully-local models alongside it [35], clustering clients based on similarity [44, 13], or building hierarchies [41, 1]. Since the proposed B-HFL family of algorithms falls in the second camp, this section shall detail the most closely related work and present its limitations. Finally, the desired properties of the federated system and their relation to previous work are summarised in Table 1.

3.3.1 Personalised Federated Learning

Fully personalised FL refers to creating one model per client in addition to the global one. The most common means of achieving this is by local adaptation, or fine-tuning, of the federated model after training [65] with the potential additions of techniques such as Knowledge Distillation [67] or Elastic-weight Consolidation [31]. However, this two-stage optimisation is challenging to implement in an FL lifecycle where the federated model may need additional training after the adaptation phase has already been carried out. Furthermore, it provides no middle ground between the global and local models, which hurts the ability of such systems to integrate new clients, which may be incapable of fine-tuning.

A more recent approach is represented by Ditto [35] for settings where clients are visited frequently and can maintain state across rounds. Ditto allows clients to maintain a persistent local model and train it alongside the federated one during FL rounds. The two models are connected by incorporating the l_2 distance between their weights within the loss function of the local one. However, despite its proven benefits of fairness and robustness, persistent local models still face the challenges of traditional personalised models. Finally, they do not address dataset shifts within the client, as they only operate during training rounds.

3.3.2 Hierarchical Federated Learning and Clustering

The most relevant subfield of FL to our proposal is Hierarchical Federated Learning (HFL) introduced by Liu et al. [41]. Their proposed HierFAVG algorithm was developed primarily to handle the communication challenges of traditional cloud-based FL. In order to obtain scales of millions of participating clients [16, 6], FL systems relied on cloud infrastructure to connect devices over a wide geographic area and thus incurred additional latency. This trade-off was considered worthwhile since the larger populations were necessary for convergence, and edge servers, while capable of fast client communication, could not draw on a sufficient data pool. Liu et al. [41] argue that a two-level structure resolves the tensions between edge servers close to the clients and cloud servers. Abad et al. [1] propose an identical algorithm for heterogeneous celluar networks where edge servers are small cell base stations, and a central macro base station replaces the cloud server. Similarly to Liu et al. [41], Abad et al. [1] focus on reducing communication costs and go further in this direction by utilising update sparsification techniques [40, 56].

Clustering clients is an orthogonal synergistic technique that attempts to group participants based on a similarity metric. These clusters are constructed using various approaches, from clustering the model parameters directly as done in Ouyang et al. [47] do or using the loss of clients when assigned to a specific cluster as Mansour et al. [44] and Ghosh et al. [13] do. Clusters may also exist naturally based on characteristics like geographic location or language.

Previous works in HFL show a series of limitations. The HierFAVG algorithm directly extends FedAvg [45] by allowing the cloud server to treat edge servers as clients. However, because Liu et al. [41] and Abad et al. [1] only consider communication efficiency, they do not allow the edge servers to maintain greater personalisation and instead replace their model entirely during cloud-aggregation. Furthermore, their system does not consider asynchronicity, proxy training, or multi-level hierarchies. Regarding clustering, the available algorithms fail to obtain the desired trade-off between generalisation and personation. Standard clustering algorithms in FL assume data-sharing between clusters is unnecessary and do not directly map onto a hierarchical communication structure. Finally, they are not meant to provide a single global model besides the cluster models for applications where it would be beneficial.

Table 1: Gap analysis table showing proposed system's properties and overlap with closely related work.

Related Work	Hierarchical Structure	Personalisation	Allows Persistent Models	General Group Models	Meaningful Group Models	Asynchronous Work
Local Adaptation		✓				
Ditto		✓	✓			
Clustering						
HieFAVG	✓			✓	✓	
Asynchronous FL						✓
Bidirectional Hierarchical FL	✓	✓	✓	✓	✓	✓

4 Proposal

Given the shortcomings of traditional hierarchical FL systems, this work proposes Bidirectional Hierarchical Federated Learning (B-HFL), an alternative family of methods that optimize data and communication efficiency. This is achieved by using the hierarchical structure to organize communication between servers and control the dissemination of training parameters through the following design choices:

1. While previous methods such as HierFAVG [41, 1] entirely replace the edge-server and client models after global aggregation takes place, B-HFL performs partial aggregation between a children node and their parent, which allows children to maintain their local weights while incorporating global information. We propose modeling this in two phases:
   (a) **Leaf-to-root aggregation:**:clients finish training, and their information is propagated up the tree. Each internal node has an internal parameter T_n, which determines after how many rounds it sends its updates to the parent. This value is equivalent to local client epochs and may be the same for all nodes at a given tree level or independently set per node.
   (b) **Root-to-leaf aggregation:** After a node has received and aggregated the training result from some or all of its children, it propagates its parameters down their subtree. The cost of this propagation is proportional to the depth of the subtree; however, the connection speed between internal nodes can be assumed to be higher than that of the clients to edge servers.
2. Internal nodes within the hierarchical structure can train on proxy datasets to regularise training as done by Guha et al. [15], Zhao et al. [68]. Proxy training is especially relevant for language modelling as large public corpora are available. In order to avoid operating on stale parameters, the natural point to add such training is after leaf-to-root aggregation reaches the node and before root-to-leaf aggregation takes place. However, the latency incurred from such training may be too large. In that case, it can operate on stale parameters asynchronously while its subtrees execute.
3. All nodes may be allowed to operate synchronously or asynchronously concerning other nodes on the same level if necessary during leaf-to-root aggregation. For leaves (clients) under the control of an edge-server, this is equivalent to traditional asynchronous FL [62]. For an internal node, the same federated asynchronous strategies [46, 21] can be applied when receiving models from the child nodes, with client execution being replaced by the execution of the entire subtree.

Expressly, parameters aggregated from the leaf nodes (clients) up through the tree are fine-tuned to relevant local data. In contrast, parameters transmitted from parents to children are averaged over more numerous populations. When servers cover meaningfully clustered clients, these populations may be less related (e.g., covering multiple languages). Furthermore, if internal nodes are allowed to train on proxy datasets, they inject additional training into the federated models and provide regularisation for the entire tree. In traditional FL approaches, training on the server directly controlling the clients can impose overly strong regularisation; however, in B-HFL, higher nodes in the tree already represent a global picture and have limited impact at the leaves as their influence gets diluted through multiple intermediary nodes. Finally, allowing each client to maintain a persistent model across rounds and aggregate with their parents rather than entirely replacing their model makes them identical to any other node except for not having children.

Since not all nodes in the tree are required to be capable of training, it is worth distinguishing models which have been optimised via additional learning rather than mere aggregation. Specifically, training data being available may enable more efficient learning-based aggregation methods such as mutual learning [67] or l_2-based regularisation [35]. Additionally, updates constructed via training directly may offer a better optimisation signal. Thus, this work proposes adding dataflows directly between training nodes (e.g., clients and the root) while using the underlying hierarchical communication structure, like residual connection in ResNet [18]. For example, the system could allow the K client updates of each server with the highest absolute value to pass all the way to the root, where they may be merged via either training or adaptive optimisation with independent accumulator states. This sort of vertical connection provides highly dynamic and potentially cyclic dataflow. Another avenue worth exploring is allowing nodes, especially clients, to train asynchronously using their persistent model. This would permit clients to account for local dataset shift using well-known techniques from the Continual Learning literature [10, 37, 31].

Algorithm 1 Recursive algorithm for a generic version of B-HFL. Each node $q \in Q$ has an associated persistent model W_q, number of executing rounds T_q, children nodes C_q, leaf-to-root learning rate $\eta^{\uparrow}$, root-to-leaf learning rate $\eta^{\downarrow}$. "Residual" edges are kept between nodes and their ancestors/descendants in $AncRes/DescRes$ with the models being accumulated in the lists of lists $R^{\uparrow}$ and $R^{\downarrow}$.

1: **Require** $Q, W, T, C, \eta^{\uparrow}, \eta^{\downarrow}, \eta^{l}, D, E$ ▷ lists indexed over all the nodes in Q
2: **Require** $R^{\uparrow}, R^{\downarrow}$ ▷ list of lists of models that a node q receives from children/ancestors
3: **Require** $AncRes, DescRes$ ▷ list of "residual" connections to descendants/ancestors
4: **Require** TRAIN, NODEOPT, SELECTRESIDUALS
5: **procedure** EXECUTENODE(ϕ, q)
6:     **if** $q = \emptyset$ **then return** $\emptyset$ ▷ error checking
7:     $\theta_0 \leftarrow W_q$ ▷ handle root
8:     **if** $\phi \neq \emptyset$ **then**
9:         $\theta_0 \leftarrow$ NODEOPT$(W_0, [\phi], R^{\downarrow}_q, q, \eta^{\downarrow}_q)$ ▷ aggregate parent $[\phi]$ and "residuals" from ancestors
10:     **for each** round $t \leftarrow 1, \ldots, T_q$ **do**
11:         $\theta_t =$ TRAIN$(\theta_t, D_q, E_q, \eta^{l}_q)$ ▷ train (sync or async) parameters on node data
12:         **for each** node $d \in DescRes_q$ **do**
13:             $R^{\downarrow}_d \leftarrow [\theta_t]$
14:         $S \leftarrow$Sample a subset from q's set of children C_q
15:         **for each** node $c \in S$ **do**
16:             $\theta^c_t \leftarrow$ EXECUTENODE(θ_t, c) ▷ non-blocking, returns a future
17:         **for each** node $a \in AncRes_q$ **do**
18:             $R^{\uparrow}_a \leftarrow$ SELECTRESIDUALS$([\theta^c_t \ \forall c \in S])$
19:         $\theta_{t+1} \leftarrow$ NODEOPT$(\theta_t, [\theta^c_t \ \forall c \in S], R^{\uparrow}_q, \eta^{\uparrow}_q)$ ▷ aggregate children and "residuals"
20:     $W_q \leftarrow \theta_{T_q}$ ▷ update persistent node model
21:     **return** θ_{T_q}
22: EXECUTENODE$(\emptyset, root)$

Algorithm 1 describes B-HFL as a recursive algorithm starting from the root of the system. It assumes that the model training TRAIN, and node aggregation NODEOPT procedures are provided. All variables are indexed per-node and assumed to be provided by the implementation. The "residual" connections are adjacency lists between nodes and their ancestors/descendants in $AncRes/DescRes$. The algorithm treats all nodes homogeneously with distinctions in execution only for the root.

1. For the root, use the persistent model as the initial federated model θ_0. [Line 6]
2. **Root-to-leaf aggregation:** UseNODEOPT to aggregate the persistent local model with the parent model ϕ and the models in "residual" connections from ancestors $R^{\downarrow}_q$ using $\eta^{\downarrow}_q$. [Line 9]
3. Begin executing federated rounds. [Line 10]
4. Train θ_t on the potentially empty dataset D_q using the local learning rate η^{l}_q for E_q local epochs. *This is where edge clients and servers with proxy datasets would execute training.* [Line 11]

5. Add the current ancestor model θ_t to the $R_d^{\downarrow}$ accumulator of every descendent to which a "residual" connection exists. [Line 12 to Line 13]

6. Sample node subset S for execution. In the case of the edge servers, the sampled set's size would equal the client cohort size. For internal nodes $S = C_q$. For a leaf node (client) $S = \emptyset$. [Line 14]

7. Recursively execute the nodes in the subtree of all selected children using θ_t. [Line 15 to Line 16]

8. Select a series of children models θ_t^c and send them to the $R_a^{\uparrow}$ accumulator of every ancestor to which a "residual" connection exists. [Line 17 to Line 18]

9. **Leaf-to-root aggregation:** UseNODEOPT to aggregate θ_t with the chidlren models $[\theta_t^c \ \forall c \in S]$ and the models in "residual" connections from descendants $R_q^{\uparrow}$ using $\eta^{\uparrow}$. [Line 19]

10. After federated training, update the persistent model W_q with the most recent federated model θ_{T_q} and then return θ_{T_q}.[Line 20 to Line 21]

"Residual" connections from descendants to ancestors may send multiple child models based (e.g., the K models representing the largest updates) directly or after an independent aggregation procedure. On the other hand, "residual" connections from ancestors to descendants only need to send one model. The most relevant example of a NODEOPT procedure is FedOPT (Eq. (3)) [50]. FedOPT can be adapted to handle residual connections by adding a second accumulator state and averaging the input from the "residuals". The synchronicity of TRAIN is defined concerning the execution of child nodes. If training is synchronous, it must complete before child nodes begin execution. If async, the model sent to a child would be θ_t prior to training, and the post-training θ_t would be used during leaf-to-root aggregation. When async training is used, it must be accounted for during the aggregation procedure with a potential staleness factor.

The system may bring several potential benefits:

1. Can accommodate nodes having different aggregation methods, learning rates, dynamic optimiser states for leaf-to-root and root-to-leaf aggregation. Similarly to the number of rounds T, parameters related to aggregation may be independent or set on a per-tree or per-level basis.

2. Smaller cohorts for each edge-server avoids the issue of decreasing pseudo-gradients norms noticed by Charles et al. [7], as does cluster clients prior to edge-server assignment.

3. While persistent local models are known to work well in cross-silo FL, this hierarchical structure makes them relevant in cross-device settings by potentially allowing a much larger number of clients to be sampled every thus permitting them to be visited more than once.

4. Can naturally integrate Secure Aggregation [5, 24] at the level of each edge-server. As first noted by Bonawitz et al. [6], this reduces additional communication cost of training C clients with Secure Aggregation from $\mathcal{O}(C^2)$ to $\mathcal{O}(C^2/M)$ where M is the number of edge-servers. Secure Aggregation and Differential Privacy [60] only need to be applied at the lowest level of the tree.

4.1 Example System

An example of a B-HFL system, which would be the primary deliverable of this proposal, may be seen in Fig. 1. The central server controls a proxy dataset used to train after it performs aggregation. Intermediary servers perform only aggregation. All servers send their updates to the parent after every round.

Each node, including the clients, runs at-least two stateful FedOPT server optimizers with separate learning rates, one for the leaf-to-root aggregation with the averaged pseudo-gradient Δ_t and one for parent aggregation. Even if the same leaf-to-root learning rate $\eta^{\uparrow}$ and root-to-leaf learning rate $\eta^{\downarrow}$ were to be used for all nodes in the tree or at a given level, the independent server optimiser states would distinguish the aggregation procedure of their node based on historical trends.

The residual connections serve different functions between the leaf-to-root and root-to-leaf stages. For the upward stage, they collect the client update with the highest absolute value from all edge servers, thus sending one additional model to the central server for each edge-server. The central server may then maintain independent optimiser states for each incoming "residual" connection. For the downward stage, they provide the edge servers with a chance to directly benefit from the training of the central server without having to rely on the models of the intermediary servers. While this last component is somewhat superfluous in the small hierarchy shown by Fig. 1, it would prove highly relevant for profound structures. For example, for deep hierarchies, parameters that receive extra training at the central server might get averaged several times before reaching the edge servers and thus influencing the leaves.

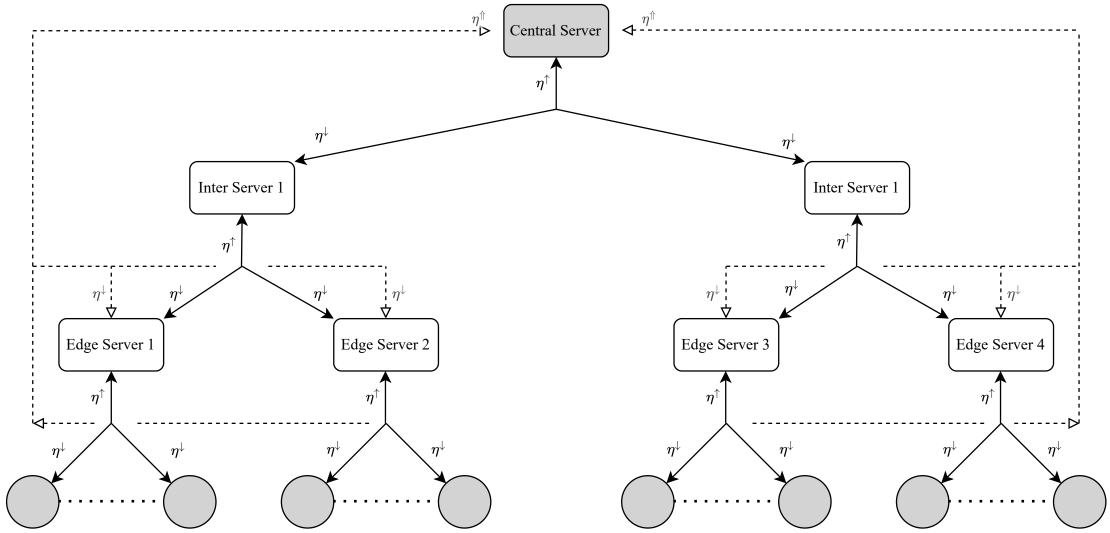

Figure 1: Diagram of an example B-HFL system. Solid lines represent communication links, while dashed lines represent conceptual "residual" connections using the underlying links. Nodes capable of training, such as clients or the central server with a proxy dataset, are in grey. When model parameters propagate up, nodes merge the incoming pseudo-gradients and update their model using the leaf-to-root learning rate $\eta^{\uparrow}$. The same happens when parameters flow from parents to children nodes with learning rate $\eta^{\downarrow}$. Since the dashed lines communicate 0 to K models, $\eta^{\Uparrow}$ may represent 0 to K aggregations using a $\eta^{\downarrow}$ learning rate.

5 Plan and Timeline

The presented family of Bidirectional Hierarchical Federated Learning algorithms will be developed during the PhD period and will form part of the final PhD thesis. In addition, before the final thesis, it offers opportunities for conference publications that significantly contribute to Federated Learning. Given the novelty of FL in general and hierarchical FL in particular, there is ample room for further developments in the structure of B-HFL as the fields mature.

The summer period of the end of my first year of the PhD shall be dedicated to implementing the example version of B-HFL in the Flower [3] FL framework affiliated with our research group. The framework is currently tuned to standard FL settings and would require heavy API modifications to execute and simulate hierarchical FL effectively. However, the previous work on group-level models for Federated Human Activity Recognition of Iacob et al. [23] and the effective FL simulation engine I contributed to can be the basis for implementing and streamlining the process.

The autumn Michaelmas Term of my second year will have as a main objective the publication of a conference paper based on the example system proposed in Section 4.1. ICLR and MLSys would be appropriate venues. Given the growing importance of LLMs, and the trade-offs recently discovered by Agarwal et al. [2] in terms of their generalization and personalization abilities with or without pre-trained weights, they represent a natural application for the proposed hierarchical system. Moreover, multi-language text prediction provides a naturally clustered FL application corresponding to real-world scenarios where countries have independent edge servers for FL and must collaborate at a continental and global level. The study would use a large multi-lingual BERT model [9] together with two multi-language datasets [e.g., 38, 63] for training. One dataset will be partitioned by language, and the other will be kept as a proxy dataset at the central server in Fig. 1. The study's goals would be to compare the final accuracy of each model at every level of the hierarchy on the client test sets and the centralised test set created from the proxy dataset. The expectation would be for the model performance on the data of a specific client to be proportional to their proximity to that client in the tree. Alternatively, for the proxy test set and the union of all client test sets, accuracy should be proportional to the proximity to the central server. In addition, ablation studies on the "residual" connections, adaptive optimization, or persistent local models will also be performed with efficiency comparisons between node-execution asynchronicity at different levels of the tree. Finally, if time allows, the paper could include other naturally-clustered tasks, such as speech recognition for multilingual data or algorithmic clustering of a standard dataset.

Following the publication of this work, a natural extension during Lent and Easter terms would be to tackle a setting where clients continuously generate and delete data with limited local storage. The example system would be extended to allow asynchronous training on all nodes, including the leaves, which run parallel to the actual FL component. Each client would generate a data stream while having a fixed internal memory to operate on during training. Real resource constraints and asynchronicity can be modelled using the Raspberry Pi FL cluster at Cambridge ML Systems. This work would likely be intended for MobiCom, the same venue we submitted the Flower simulation engine to, or another systems-oriented conference.

References

[1] Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gündüz, and Özgür Erçetin. Hierarchical federated learning ACROSS heterogeneous cellular networks. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 8866–8870. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9054634. URL https://doi.org/10.1109/ICASSP40776.2020.9054634. Cited on page 5, Cited on page 6

[2] Ankur Agarwal, Mehdi Rezagholizadeh, and Prasanna Parthasarathi. Practical takes on federated learning with pretrained language models. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 454–471. Association for Computational Linguistics, 2023. URL https://aclanthology.org/2023.findings-eacl.34. Cited on page 2, Cited on page 9

[3] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D. Lane. Flower: A friendly federated learning research framework. *CoRR*, abs/2007.14390, 2020. URL https://arxiv.org/abs/2007.14390. Cited on page 1, Cited on page 3, Cited on page 9

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL https://arxiv.org/abs/2108.07258. Cited on page 1

[5] Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *CoRR*, abs/1611.04482, 2016. URL http://arxiv.org/abs/1611.04482. Cited on page 8

[6] Kallista A. Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloé Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. In Ameet Talwalkar, Virginia Smith, and Matei Zaharia, editors, *Proceedings of Machine Learning and Systems 2019, MLSys 2019, Stanford, CA, USA, March 31 - April 2, 2019*. mlsys.org, 2019. URL https://proceedings.mlsys.org/book/271.pdf. Cited on page 1, Cited on page 2, Cited on page 4, Cited on page 5, Cited on page 8

[7] Zachary Charles, Zachary Garrett, Zhouyuan Huo, Sergei Shmulyian, and Virginia Smith. On large-cohort training for federated learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20461–20475, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ab9ebd57177b5106ad7879f0896685d4-Abstract.html. Cited on page 2, Cited on page 4, Cited on page 5, Cited on page 8

[8] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In Xintao Wu, Chris Jermaine, Li Xiong, Xiaohua Hu, Olivera Kotevska, Siyuan Lu, Weija Xu, Srinivas Aluru, Chengxiang Zhai, Eyhab Al-Masri, Zhiyuan Chen, and Jeff Saltz, editors, *2020 IEEE International Conference on Big Data (IEEE BigData 2020), Atlanta, GA, USA, December 10-13, 2020*, pages 15–24. IEEE, 2020. doi: 10.1109/BigData50022.2020.9378161. URL https://doi.org/10.1109/BigData50022.2020.9378161. Cited on page 2

[9] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.747. URL https://doi.org/10.18653/v1/2020.acl-main.747. Cited on page 9

[10] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446. Cited on page 4, Cited on page 7

[11] Dimitrios Dimitriadis, Mirian Hipolito Garcia, Daniel Madrigal, Andre Manoel, and Robert Sim. Flute: A scalable, extensible framework for high-performance federated learning simulations, March 2022. URL https://www.microsoft.com/en-us/research/publication/flute-a-scalable-extensible-framework-for-high-performance-federated-learning-simulations/. Cited on page 1

[12] John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, 2011. doi: 10.5555/1953048.2021068. URL https://dl.acm.org/doi/10.5555/1953048.2021068. Cited on page 4

[13] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *IEEE Trans. Inf. Theory*, 68(12):8076–8091, 2022. doi: 10.1109/TIT.2022.3192506. URL https://doi.org/10.1109/TIT.2022.3192506. Cited on page 5

[14] Google. Tensorflow federated, 2019. URL https://www.tensorflow.org/federated. Cited on page 1

[15] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *CoRR*, abs/1902.11175, 2019. URL http://arxiv.org/abs/1902.11175. Cited on page 6

[16] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *CoRR*, abs/1811.03604, 2018. URL http://arxiv.org/abs/1811.03604. Cited on page 1, Cited on page 5

[17] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. Fedml: A research library and benchmark for federated machine learning. *CoRR*, abs/2007.13518, 2020. URL https://arxiv.org/abs/2007.13518. Cited on page 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90. Cited on page 2, Cited on page 7

[19] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip B. Gibbons. The non-iid data quagmire of decentralized machine learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4387–4398. PMLR, 2020. URL http://proceedings.mlr.press/v119/hsieh20a.html. Cited on page 3

[20] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *CoRR*, abs/1909.06335, 2019. URL http://arxiv.org/abs/1909.06335. Cited on page 4

[21] Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. PAPAYA: practical, private, and scalable federated learning. In Diana Marculescu, Yuejie Chi, and Carole-Jean Wu, editors, *Proceedings of Machine Learning and Systems 2022, MLSys 2022, Santa Clara, CA, USA, August 29 - September 1, 2022*. mlsys.org, 2022. URL https://proceedings.mlsys.org/paper/2022/hash/f340f1b1f65b6df5b5e3f94d95b11daf-Abstract.html. Cited on page 1, Cited on page 2, Cited on page 6

[22] Alex Iacob, Pedro Porto Buarque Gusmão, and Nicholas Lane. Can fair federated learning reduce the need for personalisation? In *Proceedings of the 3rd Workshop on Machine Learning and Systems*, EuroMLSys '23, page 131–139, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700842. doi: 10.1145/3578356.3592592. URL https://doi.org/10.1145/3578356.3592592. Cited on page 2, Cited on page 3

[23] Alex Iacob, Pedro Porto Buarque Gusmão, Nicholas Lane, Armand Koupai, Mohammud Bocus, Raul Santos-Rodriguez, Robert Piechocki, and Ryan McConville. Privacy in multimodal federated human activity recognition. In *To be Published in Proceedings of the 3rd On-Device Intelligence Workshop*, MLSys '23, 2023. URL https://sites.google.com/g.harvard.edu/on-device-workshop-23/home?authuser=0. Cited on page 2, Cited on page 3, Cited on page 9

[24] Swanand Kadhe, Nived Rajaraman, Onur Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *CoRR*, abs/2009.11248, 2020. URL https://arxiv.org/abs/2009.11248. Cited on page 8

[25] Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kairouz21b.html. Cited on page 1

[26] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, and et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083. Cited on page 1, Cited on page 2, Cited on page 3

[27] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1oyRlYgg. Cited on page 4

[28] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR, 2020. URL http://proceedings.mlr.press/v108/bayoumi20a.html. Cited on page 2

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980. Cited on page 4, Cited on page 5

[30] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114. Cited on page 2

[31] 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. Cited on page 4, Cited on page 5, Cited on page 7

[32] Fan Lai, Yinwei Dai, Sanjay Sri Vallabh Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha V. Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 11814–11827. PMLR, 2022. URL https://proceedings.mlr.press/v162/lai22a.html. Cited on page 1

[33] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ByexElSYDr. Cited on page 2, Cited on page 5

[34] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=K5YasWXZT3O. Cited on page 2

[35] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6357–6368. PMLR, 2021. URL http://proceedings.mlr.press/v139/li21h.html. Cited on page 2, Cited on page 5, Cited on page 7

[36] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJxNAnVtDS. Cited on page 3

[37] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. Cited on page 4, Cited on page 7

[38] Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Bruce Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Rangan Majumder, and Ming Zhou. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *CoRR*, abs/2004.01401, 2020. URL https://arxiv.org/abs/2004.01401. Cited on page 9

[39] Bill Yuchen Lin, Chaoyang He, Zihang Ze, Hulin Wang, Yufen Hua, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. Fednlp: Benchmarking federated learning methods for natural language processing tasks. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 157–175. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-naacl.13. URL https://doi.org/10.18653/v1/2022.findings-naacl.13. Cited on page 1

[40] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SkhQHMW0W. Cited on page 5

[41] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Client-edge-cloud hierarchical federated learning. In *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*, pages 1–6. IEEE, 2020. doi: 10.1109/ICC40277.2020.9148862. URL https://doi.org/10.1109/ICC40277.2020.9148862. Cited on page 5, Cited on page 6

[42] Lingjuan Lyu, Jiangshan Yu, Karthik Nandakumar, Yitong Li, Xingjun Ma, Jiong Jin, Han Yu, and Kee Siong Ng. Towards fair and privacy-preserving federated deep models. *IEEE Trans. Parallel Distributed Syst.*, 31(11):2524–2541, 2020. doi: 10.1109/TPDS.2020.2996273. URL `https://doi.org/10.1109/TPDS.2020.2996273`. Cited on page 1

[43] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2017. Cited on page 4

[44] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *CoRR*, abs/2002.10619, 2020. URL `https://arxiv.org/abs/2002.10619`. Cited on page 5

[45] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017. URL `http://proceedings.mlr.press/v54/mcmahan17a.html`. Cited on page 1, Cited on page 2, Cited on page 3, Cited on page 6

[46] John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 3581–3607. PMLR, 2022. URL `https://proceedings.mlr.press/v151/nguyen22b.html`. Cited on page 1, Cited on page 2, Cited on page 6

[47] Xiaomin Ouyang, Zhiyuan Xie, Jiayu Zhou, Jianwei Huang, and Guoliang Xing. Clusterfl: a similarity-aware federated learning system for human activity recognition. In Suman Banerjee, Luca Mottola, and Xia Zhou, editors, *MobiSys '21: The 19th Annual International Conference on Mobile Systems, Applications, and Services, Virtual Event, Wisconsin, USA, 24 June - 2 July, 2021*, pages 54–66. ACM, 2021. doi: 10.1145/3458864.3467681. URL `https://doi.org/10.1145/3458864.3467681`. Cited on page 1, Cited on page 5

[48] Sarthak Pati, Ujjwal Baid, Brandon Edwards, Micah J. Sheller, Hans Shih-Han Wang, G. Anthony Reina, Patrick Foley, Alexey Gruzdev, Deepthi Karkada, Christos Davatzikos, Chiharu Sako, Satyam Ghodasara, Michel Bilello, Suyash Mohan, Philipp Vollmuth, Gianluca Brugnara, Chandrakanth J. Preetha, Felix Sahm, Klaus H. Maier-Hein, Maximilian Zenk, Martin Bendszus, Wolfgang Wick, Evan Calabrese, Jeffrey D. Rudie, Javier E. Villanueva-Meyer, Soonmee Cha, Madhura Ingalhalikar, Manali Jadhav, Umang Pandey, Jitender Saini, John Garrett, Matthew Larson, Robert Jeraj, Stuart Currie, Russell Frood, Kavi Fatania, Raymond Y. Huang, Ken Chang, Carmen Balaña Quintero, Jaume Capellades, Josep Puig, Johannes Trenkler, Josef Pichler, Georg Necker, Andreas Haunschmidt, Stephan Meckel, Gaurav Shukla, Spencer Liem, Gregory S. Alexander, and et al. Federated learning enables big data for rare cancer boundary detection. *CoRR*, abs/2204.10836, 2022. doi: 10.48550/arXiv.2204.10836. URL `https://doi.org/10.48550/arXiv.2204.10836`. Cited on page 1

[49] Matthias Paulik, Matt Seigel, Henry Mason, Dominic Telaar, Joris Kluivers, Rogier C. van Dalen, Chi Wai Lau, Luke Carlson, Filip Granqvist, Chris Vandevelde, Sudeep Agarwal, Julien Freudiger, Andrew Byde, Abhishek Bhowmick, Gaurav Kapoor, Si Beaumont, Áine Cahill, Dominic Hughes, Omid Javidbakht, Fei Dong, Rehan Rishi, and Stanley Hung. Federated evaluation and tuning for on-device personalization: System design & applications. *CoRR*, abs/2102.08503, 2021. URL `https://arxiv.org/abs/2102.08503`. Cited on page 1

[50] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=LkFG3lB13U5`. Cited on page 2, Cited on page 4, Cited on page 5, Cited on page 8

[51] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus H. Maier-Hein, Sébastien Ourselin, Micah J. Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *CoRR*, abs/2003.08119, 2020. URL `https://arxiv.org/abs/2003.08119`. Cited on page 1

[52] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016. URL `http://arxiv.org/abs/1609.04747`. Cited on page 4

[53] Micah J. Sheller, Brandon Edwards, G. Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R. Colen, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):12598, 2020. doi: 10.1038/s41598-020-69250-1. URL `https://doi.org/10.1038/s41598-020-69250-1`. Cited on page 1

[54] Konstantin Sozinov, Vladimir Vlassov, and Sarunas Girdzijauskas. Human activity recognition using federated learning. In Jinjun Chen and Laurence T. Yang, editors, *IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications, ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018, Melbourne, Australia, December 11-13, 2018*, pages 1103–1111. IEEE, 2018. doi: 10.1109/BDCloud.2018.00164. URL `https://doi.org/10.1109/BDCloud.2018.00164`. Cited on page 1, Cited on page 3

[55] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *CoRR*, abs/2103.00710, 2021. URL `https://arxiv.org/abs/2103.00710`. Cited on page 5

[56] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7663–7673, 2018. URL `https://proceedings.neurips.cc/paper/2018/hash/44feb0096faa8326192570788b38c1d1-Abstract.html`. Cited on page 5

[57] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/arXiv.2302.13971. URL `https://doi.org/10.48550/arXiv.2302.13971`. Cited on page 1

[58] Ewen Wang, Ajay Kannan, Yuefeng Liang, Boyi Chen, and Mosharaf Chowdhury. FLINT: A platform for federated learning integration. *CoRR*, abs/2302.12862, 2023. doi: 10.48550/arXiv.2302.12862. URL `https://doi.org/10.48550/arXiv.2302.12862`. Cited on page 1

[59] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=BkluqlSFDS`. Cited on page 2

[60] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.*, 15:3454–3469, 2020. doi: 10.1109/TIFS.2020.2988575. URL `https://doi.org/10.1109/TIFS.2020.2988575`. Cited on page 8

[61] White House. Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy. *Journal of Privacy and Confidentiality*, 4(2), Mar. 2013. doi: 10.29012/jpc.v4i2.623. URL `https://journalprivacyconfidentiality.org/index.php/jpc/article/view/623`. Cited on page 1

[62] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. Asynchronous federated learning on heterogeneous devices: A survey. *CoRR*, abs/2109.04269, 2021. URL `https://arxiv.org/abs/2109.04269`. Cited on page 2, Cited on page 6

[63] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934, 2020. URL `https://arxiv.org/abs/2010.11934`. Cited on page 9

[64] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A sustainable incentive scheme for federated learning. *IEEE Intell. Syst.*, 35(4):58–69, 2020. doi: 10.1109/MIS.2020.2987774. URL `https://doi.org/10.1109/MIS.2020.2987774`. Cited on page 1

[65] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *CoRR*, abs/2002.04758, 2020. URL `https://arxiv.org/abs/2002.04758`. Cited on page 5

[66] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M. Alvarez. Personalized federated learning with first order model optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL `https://openreview.net/forum?id=ehJqJQk9cw`. Cited on page 2

[67] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4320–4328. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00454. URL `http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_Deep_Mutual_Learning_CVPR_2018_paper.html`. Cited on page 2, Cited on page 3, Cited on page 5, Cited on page 7

[68] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018. URL `http://arxiv.org/abs/1806.00582`. Cited on page 3, Cited on page 5, Cited on page 6