Bidirectional Hierarchical Federated Learning

Alex Iacob
aai30@cam.ac.uk

1 Introduction

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. [65] as a means of reducing communication costs and lessening the privacy concerns of storing sensitive data in a centralised location, following the principle of data minimisation outlined in the White House [90] privacy report. These properties have led to FL applications with large cohorts of small edge devices, e.g., mobile keyboard prediction [27] for Android phones, and settings with larger entities subject to privacy requirements, e.g., hospitals [79]. Kairouz et al. [41] distinguish them as cross-device and cross-silo FL.

The growth in the preponderance of Federated Learning since the publication of McMahan et al. [65] 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 [65, 27, 24, 40], Microsoft [86, 17], Meta [34, 69], and Apple [72]. Second, ML has extended to domains with strict privacy requirements such as healthcare [79, 76, 71], Human Activity Recognition (HAR) [81, 70] or collaborations between competing corporations [96, 62]. Moreover, the emergence of Large Language Models (LLMs) [6] has made accessing private language corpora advantageous, leading to the development of Federated Natural Language Processing (FedNLP) [56]. Similarly, the release of openly available LLM pre-trained weights [85] allows collaboration between entities with low computational resources using FL frameworks [4, 46, 28].

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, training a single global model is unsuitable when unusual clients require partial or complete personalisation of the model to their local data distribution.

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* [41, sec. 3.1] is interwoven with *systems heterogeneity* [41, 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.

Efficiency and scalability have been at the centre of FL research since Hard et al. [27] applied FL to mobile keyboard prediction at Google. Building on top of Hard et al. [27], Bonawitz et al. [8] 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. [8], 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. [10].

Charles et al. [10] show that the performance of FL does not scale as expected when the number of clients trained every round increases despite previous theoretical work [43] 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 a 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 is highly parallel, its scalability is limited by the ability to learn from clients on a per-sample basis. Asynchronous Federated Learning systems [94, 69, 12, 19], such as Meta's PAPAYA [34], show promise in improving efficiency and scalability; however, they introduce the new issues of staleness and high update variance.

Evolving FL systems is also a major challenge. 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 change gradually or immediately. For example, seasonal transitions shift captured images slowly, while changing locations leads to discrete changes. This problem is known as dataset shift [41, sec. 3.1] and represents *intra-client* heterogeneity rather than the common *inter-client* heterogeneity. Even works which maintain persistent local models [51, 3, 16, 26] assume that this model is only used within FL rounds, obfuscating such shifts.

Structure: Given the challenges of FL and shortcomings of previous work, detailed in **??**, the proposed research for my PhD aims to *provide flexible personalisation for highly efficient and scalable FL systems* by exploring the research questions outlined in Section 10. To achieve this goal, **??** propose a new family of FL algorithms called Bidirectional Hierarchical Federated Learning (B-HFL) based on Algorithm 1. Building on these foundations, I outline the research directions of my PhD that B-HFL enables in Section 13. Finally, I summarise already completed work **??** and provide a detailed timeline for the PhD in **??**.

Notice: The following subsection provides a *summary* of B-HFL and the contributions it enables.

2 Summary of Proposed Research

Bidirectional Hierarchical FL (B-HFL) will address the aforementioned personalisation, efficiency, and evolution challenges by constructing hierarchical federated network structures that allow bidirectional and potentially cyclical dataflow where each leaf is a client and each internal node is a server. 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. Furthermore, since B-HFL treats nodes homogeneously, every intermediary node can operate independently like a standard FL server, using synchronous or asynchronous execution of its sub-nodes depending on constraints.

This proposal builds upon the work done by Iacob et al. [35] and Iacob et al. [36] on personalised and hierarchical FL. The proposed system communicates data in as shown in Algorithm 1 and 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 [75], model-interpolation [16, 26], or training-based methods [51, 45, 99, 98]. Adaptive algorithms and model interpolation are particularly relevant as they allow each node in the tree to distinguish itself based on its previous state without necessitating additional parameter tuning. Furthermore, since each edge-server controls fewer clients, the diminishing effects of increasing cohort sizes are avoided. 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, 58]. The potential contributions to the field of Federated Learning include:

1. A family of efficient algorithms with minute control of personalisation and generalisation, capable of achieving communication efficiency in hierarchical networks.
2. The investigation of three techniques enabled by B-HFL: (a) allowing leaf nodes 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 general information, (c) constructing vertical connections in the tree, similar to residual connections [29], to allow customisable dataflow without changing the underlying communication infrastructure.
3. Extensive empirical evaluations considering scenarios with or without meaningful client clusters in language and image/speech recognition tasks leading to intended publication 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.

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) [65] trains the global model locally, 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.

4 Heterogeneity

Non-IID data has been shown to impact both practical accuracies [100, 31] and theoretical convergence bounds [52]. It is thus worth detailing some forms of heterogeneity that Kairouz et al. [41] 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 [8, 48] or operate on stale parameters in asynchronous FL [93, 34]. In addition, network or device unreliability creates dropout, which requires oversampling clients [8] 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 [15]; 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 [45], PackNet [63], and Learning without Forgetting [54] are designed for task-incremental settings where class labels are known, small amounts of previous data may still be available for specialised use cases [45], 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.

5 Privacy

While privacy in FL does not represent a main research direction for the proposed research, it is one of the primary concerns of the field. As such, any approaches which attempt to tackle the main challenges of FL must do so while accounting for their privacy implications.

Since FL keeps training data locally stored on the client, it offers more privacy than standard ML approaches. However, previous works [22, 5, 73, 102] have shown that models trained in a federated fashion may allow for partial or complete reconstruction of their training data. From the perspective of the proposed research, privacy serves as a test for the feasibility of a particular method to be applied in an FL context. For example, methods which require detailed knowledge of the data each client [100, 67, 91, 37] may be rejected as impractical. Similarly, the ability of a system to support FL privacy techniques like Secure Aggregation (SecAgg) [7, 39, 80] or Differential Privacy (DP) [20, 89, 66, 62, 40] is relevant.

Secure aggregation is a form of Secure Multi-party Aggregation and was introduced to FL by Bonawitz et al. [7]. It operates by having all clients generate and share secrets. The clients then mask their models

using random noise in such a manner that the server can construct the true average of client updates without knowing the true weights of any individual client. Such techniques require multiple clients to participate in aggregation concurrently and have $\mathcal{O}(C^2)$ communication cost where C is the number of clients whose models are being aggregated. This excludes, for example, fully asynchronous federated learning as proposed by Xie et al. [93].

Differential Privacy in FL [66] is formally defined as shown in Eq. (3)

$$\Pr[M(d) \in S] \leq e^{\epsilon} \Pr[M(d') \in S] + \delta \,, \tag{3}$$

where M is a probabilistic model, S is the output set of that model, d is the dataset used to train the model and d' is an adjacent dataset. Two datasets are adjacent in FL if they can be formed by adding or subtracting the local dataset of one client. Finally, (ϵ, δ) bound the similarity of outputs between two models trained with or without a specific client. Since DP comes with an inherent privacy-accuracy trade-off, the most relevant factor for its usability is whether an FL system can be scaled to operate over sufficiently large populations of clients. Larger populations allow productive training while offering a low ϵ by limiting the contribution of individual clients.

6 Federated Learning Efficiency

It is now worth expanding on the trends that Charles et al. [10] 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. More recently, Zhou et al. [101] have shown that while the federated model successfully keeps the client models in a common basin and achieves the lowest loss, it can drift from the optimum across rounds due to cohort heterogeneity. Furthermore, they also show that the efficacy of large cohorts in reducing the variance of aggregate updates is limited by the degree of data heterogeneity present.

These behaviours are approximately analogous to the well-known efficiency and generalisation limitations of large-batch training in centralised ML [42]. Charles et al. [10] 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.

6.1 Adaptive Federated Optimisation

Of particular relevance to the proposed research are Federated Averaging with Server Momentum (FedAvgM) [32] and the more general Federated Adaptive Optimisation (FedOPT) [75]. They extend the concepts of momentum and adaptive optimisation [18, 44, 77] 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, such techniques offer a more consistent optimisation trajectory.

Specifically, following the account provided by Reddi et al. [75] as shown in Eq. (4)

$$\Delta_t = \frac{1}{|C|} \sum_{c \in C} (\theta_t^c - \theta_t) \tag{4a}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \Delta_t \tag{4b}$$

$$v_t = \beta_2 v_t + (1 - \beta_2) \Delta_t^2 \tag{4c}$$

$$\theta_{t+1} = \theta_t + \eta \frac{m_t}{\sqrt{v_t} + \tau} \tag{4d}$$

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. (4a). 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. (4b). 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 [44] the second-moment accumulator v_t keeps track of the second power of the pseudo-gradient denoted by Δ_t^2 as shown in Eq. (4c). 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. (4d). The term $\sqrt{v_t}$ normalises model parameters, making the algorithm scale-invariant to the pseudo-gradient. Finally, τ controls adaptivity.

FedOPT presents several promising properties in the context of hierarchical FL. First, Reddi et al. [75] 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. [10] regarding the near-zero pseudo-gradients caused by the near-orthogonality of client updates. Third, they provide a means of automatically differentiating multiple servers based on accumulator state without hyperparameter tuning.

7 Asynchronous Federated Learning

Together with the previously mentioned adaptive federated optimization, asynchronous FL [93, 69, 34, 94, 12] represents another promising means of improving the overall efficiency of FL generally and the presently proposed system specifically by improving concurrency. In the context of FL, concurrency refers to *the number of clients training simultaneously*.

The federated cohort size trained during a round controls the system's concurrency for standard synchronous FL. Besides the data-efficiency issues discussed in Section 6, this round-based design introduces two factors which limit effective concurrency. First, the concurrency of a round decreases as clients finish training. Second, stragglers with slow hardware or large datasets elongate a round, usually addressed through oversampling [8] or a time cut-off.

Asynchronous FL was proposed by Xie et al. [93] as an alternative means of tackling stragglers in FL besides the standard oversampling method introduced by Bonawitz et al. [8]. In its fully asynchronous form, it functions by allowing each client to update the global federated model when they finish training. Thus, it removes client update averaging and allows the system to maintain high concurrency. However, clients may return stale updates at round t generated by training the model from round τ. As such, Xie et al. [93] and future works [69, 34] utilise a staleness function $s(t-\tau)$ in the aggregation to compensate, as seen in Eq. (5)

$$\theta_{t+1} = \theta_t + \eta\left(s(t-\tau)\,\theta_\tau^c\right) \ . \tag{5}$$

The benefits of the fully asynchronous approach are countermanded by its sensitivity to data heterogeneity and inability to properly utilise cohort-based techniques such as Secure Aggregation [7]. Up to a limit [10, 101, 69, 34], using cohort-based aggregation in synchronous FL imposes a variance-reduction effect which limits the impact of highly heterogeneous clients. Xie et al. [93] must instead adopt the local regularised of FedProx [48] to constrain model divergence.

To regain the variance-reduction benefits and cohort-based techniques of synchronous FL, Nguyen et al. [69] introduce FedBuff. FedBuff brings a conceptually minor but practically crucial change to async FL by introducing a buffer of size K, which holds updates until it is filled. After the buffer is full, it averages the pseudo-gradients weighted by their staleness and updates the model. Unlike the fully synchronous approaches, concurrency remains decoupled from K, which only controls how often a new model version is created. Huba et al. [34] build upon FedBuff in deployment at Meta and show that it can bring significant improvements in convergence time by updating the model more frequently for the same concurrency and not waiting for stragglers. Huba et al. [34] also show FedBuff results in a more uniform accuracy distribution over clients than oversampling since it incorporates updates from such stragglers.

Asynchronous FL can be combined with adaptive optimisation, as done by Huba et al. [34], communication-efficient methods like dropout regularisation [19], and is entirely compatible with the hierarchical FL, as will be shown in Section 8.2 and **??**.

8 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 [49], 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 [97, 82, 100], maintaining persistent fully-local models alongside it [3, 16, 26, 51], clustering clients based on similarity [64, 23, 58], or building hierarchies [59, 1, 67]. 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 B-HFL and its relation to previous work are summarised in Table 1.

8.1 Personalised Federated Learning

Personalised Federated Learning (PFL) refers to a class of FL algorithms which intends to tackle the Non-IID distribution of client data by creating models which better match the distribution of specific clients or groups of clients. This strategy differs from standard FL, which attempts to create the best compromise model. Thus, PFL approaches lie between creating one global model and personalising on a per-client basis, with clustering approaches offering a compromise.

8.1.1 Per-client Personalisation

Fully personalised FL refers to creating one model per client in addition to the global one. The most common means of achieving this is a local adaptation (fine-tuning) of the federated model after training [97, 64, 13]. Local adaptation is potentially combined with techniques such as Knowledge Distillation [99] or Elastic-weight Consolidation [45] for the explicit purpose of combating catastrophic forgetting [21]. However, this two-stage optimisation is challenging to implement in an FL lifecycle where the federated model may need additional training after local adpatation. Furthermore, it provides no middle ground between global and local models, which hurts the ability to integrate new clients as they may be incapable of fine-tuning.

An alternative approach is represented by Ditto [51] 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. Model interpolation [64, 16, 26] and techniques which maintain local personalisation layers [3, 53] also rely on clients being capable of maintaining a persistent state. The model interpolation approaches of Mansour et al. [64] and Deng et al. [16] rely on optimising local models with a mixture parameter which is adaptively tuned through SGD. On the other hand, Looples Gradient Descent [26] allows clients to probabilistically take steps towards either local training or partially averaging the federated model into their local one. FInally, split approaches such as those of Arivazhagan et al. [3] and Li et al. [53] train models locally but only average and update the layers up to a cut layer q, with the intuition that earlier layers represent more general feature extractors and later layers require heavier personalisation.

However, despite the proven benefits to local performance [3, 16] as well as fairness and robustness [51], maintaining a persistent state still faces the challenges of traditional personalised models in terms of incorporating new clients with the additional cold-start problem of initialising their local state. Moreover, for cross-device settings with low levels of participation, which arise both due to client availability and the small cohort sizes used in practice [8, 10], local client state may become stale across rounds. Finally, neither fine-tuning nor persistent-state techniques address dataset shifts within the client, as they only operate during training or adaptation rounds.

8.1.2 Clustering

Clustering clients is a technique that attempts to group participants based on a similarity metric, with two predominant variants being used in FL. First, since directly clustering clients based on their data is unfeasible, standard clustering approaches such as K-means [60] or Hierarchical Agglomerative Clustering [38] need to operate over embeddings representing the local distribution of a client. The natural choice for such an embedding in FL is using locally-trained models (or pseudo-gradients) directly, as done in the one-shot K-means algorithm of Ghosh et al. [23], in Sattler et al. [78], and in the hierarchical clustering algorithm of Briggs et al. [9]. In such solutions, the distance function between models will be the l_n norm of model differences [23, 9] or a similarity metric such as the cosine similarity [78, 58, 9] computed over flattened parameters. While locally trained models are widely available and more admissible from a privacy perspective to a raw representation like prototypes [83], using the entire model is computationally expensive and potentially uninformative. As Wang et al. [87] indicates, two models may represent the same concept in different sets of weights. This issue can be addressed using an explicit encoder [33].

Second, because the methods above are computationally expensive and challenging to apply dynamically for low-participation cross-device FL, Ghosh et al. [23] and Mansour et al. [64] concomitantly proposed a loss-based form of clustering for FL. Their algorithms are iterative and operate by maintaining K

hypothesis models, which are communicated to clients. Clients then get assigned to the cluster model where they have the lowest training loss, and then the models of clients who are self-selected to a specific cluster are averaged to produce the new cluster models. The procedure then repeats for several rounds or until convergence; at this point, each cluster is separated and runs FL independently However, despite the dynamic nature of such clustering being well-fit to FL, the increase in communication and computation brought by having each client interact with K models is significant and hard to reconcile with other FL approaches. Furthermore, it is unclear how K should be chosen without access to client data.

A practical FL system for clustering which addresses issues in both approaches, Auxo, was proposed by Liu et al. [58]. Auxo creates an initial set of clusters using K-means with cosine similarity, then it dynamically adjusts the clusters by splitting them hierarchically if it heuristically detects that the split would reduce heterogeneity in the population. Clients join clusters based on an exploration-exploitation trade-off where the reward is computed based on the distance of the gradient produced by training a client on a specific cluster and the cluster average. This reward can be propagated to unexplored clusters due to the hierarchical structure, with clusters having a distance metric based on the number of steps to their most recent ancestor.

All the available clustering algorithms, including Auxo, fail to obtain the desired trade-off between generalisation and personation because they do not continuously share information between clusters. In the case of Auxo, because of the hierarchical cluster splitting, ancestors provide the initialisation weights for new clusters but become utterly disconnected afterwards. As such, creating more clusters results in them having access to increasingly fewer clients, which always hurts generalisation and may harm the performance of the cluster clients directly if they do not possess enough aggregate samples to train a high-quality model. Thus, the decision is only reasonable when the reductions in data heterogeneity are sufficient to compensate for the smaller population. One exception is the weight-sharing iterative clustering algorithm proposed by Ghosh et al. [23], which serves as a middle point between a two-layer hierarchical solution and multi-task learning with personalisation layers; however, it still suffers from the drawback of having to train K versions of the model on each client. Finally, clustering algorithms are not meant to provide a single global model or intermediary models besides cluster models, even for applications where it would be beneficial to have both a good default and customised experiences.

Clusters may also exist naturally based on characteristics like location or language, which become relevant if the clients and server controlling them are geographically correlated.

8.2 Hierarchical Federated Learning

The most relevant subfield of FL to our proposal is Hierarchical Federated Learning (HFL) introduced by Liu et al. [59]. 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 [27, 8], 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. [59] 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. [59], Abad et al. [1] focus on reducing communication costs and go further in this direction by utilising update sparsification techniques [57, 84]. To further improve communication efficiency Luo et al. [61] propose a resource allocation framework which assigns clients to edge servers to optimise costs.

Previous works in HFL show a series of limitations. The HierFAVG algorithm directly extends FedAvg [65] by allowing the cloud server to treat edge servers as clients. However, because Liu et al. [59] 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. The work of Wang et al. [88], RFL-HA, combines hierarchical aggregation and clustering in a mixed scenario of peer-to-peer and client-server FL where powerful clients take on the role of edge servers and perform aggregation before transmitting their models to the cloud for asynchronous aggregation. However, their clustering procedure is meant to optimise communication efficiency first and foremost. It thus does not exploit the personalisation advantages of combining clustering and hierarchical FL.

Mhaisen et al. [67] do consider scenarios where the data distribution of edge servers is taken into account and propose optimal user-edge assignment. Specifically, they allow edge servers to contain clients with a Non-IID distribution and use FedSGD [65] to counteract its effects. To obtain communication efficiency

without sacrificing convergence at the cloud server, they attempt to maintain an IID distribution across edge servers and apply FedAvg at the cloud server level. While promising, their work requires complete knowledge of the distribution of each client in order to realise edge-server assignment. Furthermore, it assumes that edge servers have sufficiently low communication latency to efficiently train with FedSGD despite the original work of McMahan et al. [65] showing FedSGD to be up to two orders of magnitude slower than FedAvg in terms of convergence speed.

The current approaches to hierarchical Federated Learning cannot adequately tackle data heterogeneity because their solve objective is constructing a single consensus model to which the entire federated network is meant to converge. This is opposite to the concern of previously discussed clustering systems incapable of effectively sharing information across clusters. To address this, it is necessary to simultaneously tackle the construction of both generalisable and more personalised models while flexibly tunning the generalisation-personalisation trade-off.

Table 1: Gap analysis showing B-HFL's intended properties and overlap with related work.

Related Work	Hierarchical	Per-client Personalisation	Allows Persistent State	Group Models	Meaningful Groups	Allows Async	Scalable	Private
Local Adaptation [97, 64, 13]		✓					✓	✓
Persistent Models/Layers [51, 16, 3]		✓	✓					✓
Standard Clustering [23, 9, 78, 64]				✓	✓			✓
Auxo [58]				✓	✓		✓	✓
HieFAVG [59, 1, 61]	✓			✓			✓	✓
Optimal User-edge Assignment [67]	✓			✓	✓			
RFL-HA [88]	✓			✓		✓	✓	✓
Asynchronous FL [93]						✓		✓
FedBuff [69, 34]						✓	✓	✓
Bidirectional Hierarchical FL	✓	✓	✓	✓	✓	✓	✓	✓

9 Proposal

Given the shortcomings of traditional hierarchical FL systems, this work proposes Bidirectional Hierarchical Federated Learning (B-HFL), an alternative family of methods that optimises data and communication efficiency while allowing flexible degrees of personalisation. This section constructs the algorithmic framework upon which the rest of the PhD thesis will be built.

10 Research Questions

The proposed research aims to answer the following research questions during the PhD:

1. Can using multiple servers with small cohorts *outperform large-cohort single-server FL by avoiding the data efficiency problems identified by Charles et al. [10]?*
2. If the need for convergence to a global model is removed, *can hierarchical FL effectively address the trade-off between generalisation and personalisation with Non-IID data?*
3. If all nodes are treated homogeneously and servers are allowed to train on proxy data, *can the regularisation strength be more effectively controlled with the hierarchical structure?*
4. Can the structure of the network itself be used to *enhance or design aggregation strategies without major harm to communication efficiency?*
5. Do persistent models and/or continual asynchronous learning allow nodes to *tackle dataset shift and obtain a greater degree of personalisation?*

All of these questions are intertwined within the hierarchical structure itself, and, unlike previous work in hierarchical FL, they are all predicated on the abandonment of a single global model as the explicit goal of FL. They can all be reduced to the following question: *Can a hierarchical FL structure allow us to better utilise node data on both clients and servers while maintaining the communication efficiency which made FL practical in the first place?*

As shown in Table 1 and discussed in Section 8, previous approaches in the field are not flexible enough to simultaneously allow trade-offs in terms of personalisation, sample efficiency and asynchronous training or execution. As such, research in the following proposed family of algorithms would significantly contribute to the field.

11 System Design

The fundamental design of the FL systems proposed in this work allows 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 [59, 1] entirely replace the edge-server and client models after global aggregation takes place, B-HFL performs partial aggregation between a child node and their parent. This allows children to maintain their local weights while incorporating global information. It proceeds in two phases:
   (a) **Leaf-to-root aggregation:** clients finish training, and their information is propagated up the tree. Each internal node has a parameter T_n, which determines after how many rounds it sends its updates to the parent. This value is equivalent to local client epochs during SGD and may be the same for a tree level or independent 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 propagation cost is proportional to the depth; however, communication between internal nodes can be assumed to be faster than between clients and edge servers.
2. All nodes may be allowed to execute synchronously or asynchronously concerning other nodes on the same level if necessary during leaf-to-root aggregation. For leaves (clients), this is equivalent to traditional asynchronous FL [94]. For an internal node, the same federated asynchronous strategies [69, 34] can be applied when receiving models from the children, with client training replaced by executing the subtree rooted at the child node.
3. Internal nodes within the hierarchical structure can train on proxy datasets to regularise training as done by Guha et al. [25], Zhao et al. [100]. 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 for such training is after leaf-to-root aggregation and before root-to-leaf aggregation. However, the latency incurred from such training may be too large. In that case, it can train parameters asynchronously while the subtree executes.

Thus, the objective function of FL from Eq. (1) is modified for B-HFL as described in Eq. (6)

$$\min_{\theta} F_q(\theta) = \alpha_q f_q(\theta) + \beta_q F_{D_q}(\theta) + \gamma_q F_{A_q}(\theta) \tag{6a}$$

$$F_{D_q}(\theta) = \sum_{d \in D_q} p_d F_d(\theta) \tag{6b}$$

$$F_{A_q}(\theta) = \sum_{a \in A_q} p_a F_a(\theta) \tag{6c}$$

$$f_q(\theta) = \frac{1}{|\Omega|} \sum_{j \in \Omega_q} f_q^j(w) \tag{6d}$$

where each node q in the tree attempts to find the model θ which minimizes its local objective f_q, that of its descendants F_{D_q}, and ancestors F_{A_q} using weights $\alpha_q, \beta_q, \gamma_q$. The objective of the descendants and ancestors are recursively described while the local objective f_q is defined by performance of the model θ on the local node dataset Ω_q. In the case of a leaf node, only its local objective and that of the ancestors matter, while for the root, only its local objective and that of the descendants matter. If an internal node lacks a proxy dataset, only F_{D_q} and F_{A_q} are optimized. All leaf nodes are expected to have local datasets.

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. Keeping persistent models and repeatedly re-aggregating them is a more structured version of the Iterative Moving Averaging (IMA) applied by Zhou et al. [101].

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 [99], l_2-based regularisation [51] or model interpolation with adaptive weights [16, 64]. Additionally, updates constructed via training directly may offer a better optimisation signal similarly to methods trying to build diverse ensembles [47] or diverse models for parameter averaging [74]. 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 [29]. 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 [15, 54, 45].

Algorithm 1 describes B-HFL recursively starting from the system's root. 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 $\mathrm{AncRes}/\mathrm{DescRes}$.

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 rounds T_q, child 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 $\mathrm{AncRes}/\mathrm{DescRes}$ with models accumulated in the 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** $\mathrm{AncRes}, \mathrm{DescRes}$ ▷ 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: $\theta_0 \leftarrow \text{NODEOPT}(q, W_0, [\phi], R^{\downarrow}_q, q, \eta^{\downarrow}_q)$ ▷ aggregate parent $[\phi]$ and "residuals"
10: **for each** round $t \leftarrow 1, \ldots, T_q$ **do**
11: **for each** node $d \in \mathrm{DescRes}_q$ **do**
12: $R^{\downarrow}_d \leftarrow [\theta_t]$
13: $S \leftarrow$Sample a subset from q's set of children C_q
14: **for each** node $c \in S$ **do**
15: $\theta^c_t \leftarrow \text{EXECUTENODE}(\theta_t, c)$ ▷ sync/async
16: **for each** node $a \in \mathrm{AncRes}_q$ **do**
17: $R^{\uparrow}_a \leftarrow \text{SELECTRESIDUALS}(q, [\theta^c_t \ \forall c \in S])$
18: $\theta'_t \leftarrow \text{NODEOPT}(q, \theta_t, [\theta^c_t \ \forall c \in S], R^{\uparrow}_q, \eta^{\uparrow}_q)$ ▷ aggregate children and "residuals"
19: $\theta_{t+1} = \text{TRAIN}(\theta'_t, D_q, E_q, \eta^{l}_q)$ ▷ train (sync/async) parameters on node data
20: $W_q \leftarrow \theta_{T_q}$ ▷ update persistent node model
21: **return** θ_{T_q}
22: EXECUTENODE($\phi = \emptyset, q = root$)

In its natural language form, Algorithm 1 operates as follows:

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 node model with the parent model ϕ and those in "residual" connections from ancestors $R^{\downarrow}_q$ using $\eta^{\downarrow}_q$. [Line 9]
3. Begin executing federated rounds. [Line 10]

4. Add ancestor model θ_t to $R_d^{\downarrow}$ for descendants with "residual" connections. [Line 11 to 12]
5. Sample node subset S for execution. For edge servers, $|S|$ would equal the client cohort size. For non-edge servers $S = C_q$ while for a leaf node (client) $S = \emptyset$. [Line 13]
6. Recursively execute nodes in the subtree of selected children, sending θ_t. [Line 14 to 15]
7. Select and send children models θ_t^c to $R_a^{\uparrow}$ for ancestors with "residual" connections. [Line 16 to 17]
8. **Leaf-to-root aggregation:** UseNODEOPT to aggregate θ_t with the children models $[\theta_t^c \ \forall c \in S]$ and those in "residual" connections from descendants $R_q^{\uparrow}$ using $\eta^{\uparrow}$. [Line 18]
9. Train θ_t on the potentially empty dataset D_q using local learning rate η_q^l for E_q epochs. *This is where edge clients and servers with proxy datasets would execute training.* [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 21]

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+1} 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.

"Residual" connections from descendants to ancestors may send multiple children models (e.g., the K models representing the largest updates) directly or after a "residual" aggregation procedure which merges them. 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. (4)) [75]. FedOPT can be adapted to handle residual connections by adding a second accumulator state and averaging the input from the "residuals". Bidirectional Hierarchical FL may bring several potential benefits:

1. Can accommodate nodes with different aggregation methods, learning rates, dynamic optimiser states for leaf-to-root and root-to-leaf aggregation. Similarly to the number of rounds, aggregation parameters may be independent or set per tree or level.
2. Smaller cohorts for each edge-server avoids the issue of decreasing pseudo-gradients norms noticed by Charles et al. [10], as does clustering clients during 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 more significant number of clients to be sampled per round, thus visiting them more than once.
4. Can naturally integrate Secure Aggregation [7, 39] at the level of each edge-server. As first noted by Bonawitz et al. [8], this reduces the 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 [89] only need to be applied at the lowest level of the tree.

12 Example System

An example of a B-HFL system, which would be the first deliverable for the second year of my PhD, 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. Servers send updates to the parent after every round.

Each node, including the clients, runs at-least two stateful FedOPT server optimisers with separate learning rates, one for the leaf-to-root aggregation 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 central server uses model interpolation with a mixture parameter [16] adapted to its proxy data.

The residual connections serve different functions between the leaf-to-root and root-to-leaf stages. For the upward stage, they collect the $K = 1$ client update with the highest absolute value, thus sending one additional model to the central server per edge server. For the downward stage, they allow the edge servers to directly benefit from the central server's training without relying on averaged intermediate models. While this last component is somewhat superfluous in the small hierarchy shown by Fig. 1, it may prove relevant for profound structures. For example, in 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.

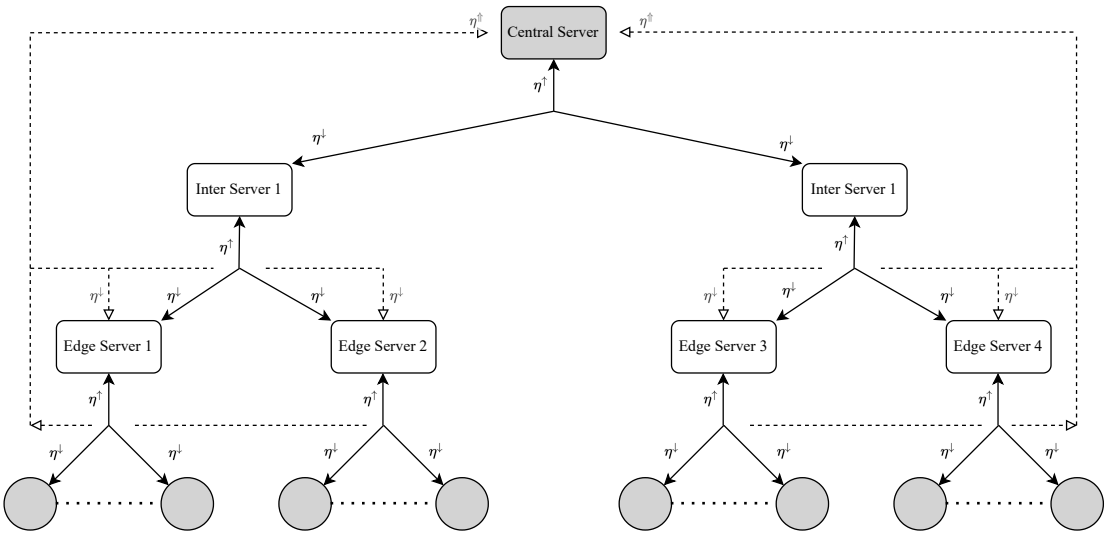

Figure 1: Example B-HFL system. Solid lines represent communication links, and dashed lines represent conceptual "residual" connections. Nodes capable of training, such as clients and the central server with proxy data, are in grey. When model parameters propagate up, nodes merge incoming pseudo-gradients and update their model with learning rate $\eta^{\uparrow}$. The same happens when parameters flow from parents to child 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 $\eta^{\uparrow}$ learning rate.

13 Research Directions

Algorithm 1 imposes sufficient structure to create a new family of hierarchical FL algorithms, opening a series of research directions. These directions can be primarily divided into three types: (a) node aggregation procedures filling in the NODEOPT function, (b) residual selection procedures filling in the SELECTRESIDUALS function, and (c) clustering algorithms which decide the edges of each node in the tree. Both node aggregation and "residual" selection are expected to be set for types of nodes or levels of the tree, as allowing each node to have a separate aggregation procedure would be difficult to manage practically. Regarding clustering, relations imposed by the physical communication links and purely conceptual ones must be distinguished. For example, a physical edge server may represent multiple internal nodes in the tree if multiple clusters fall under its geographical reach. While conceptual relations are flexible to the data properties of clients and proxy datasets, physical links are unalterable.

Node Aggregation Procedures Relevant node aggregation procedures can either be those developed for standard FL, replacing clients with child nodes, or procedures that can take advantage of the unique graph structure or available proxy data. FedOPT [75] and Iterative Moving Averaging [101] are examples of standard FL algorithms that offer unique *implicit* benefits for this hierarchical structure because they maintain stateful accumulators or lists of previous models, respective, which permit every server to be distinguished. The smaller number of children of non-edge servers may also enable much more costly aggregation procedures, which are limited in standard FL due to many edge clients involved. For example, internal nodes having potential proxy data allows them to use model interpolations with adaptive interpolation rates, as proposed by Deng et al. [16], optimised to the proxy data in order to balance the influence of updates from child nodes, parents nodes, and from proxy training. Other data-dependent techniques include fine-tuning while constraining the l_2 norm [51], EWC [45, 97] or KD [99, 97]. Aggregation procedures can also *explicitly* consider the links between nodes. A simple example of this second type is aggregation considering the distance between an ancestor and a descendant connected by a "residual" connection. For example, more complex procedures may combine the known hierarchical topology with similarity metrics to create a distance matrix between internal node models and perform graph message passing [92], as proposed by Chen et al. [11]. In such message passing, the models of all neighbouring nodes would be interpolated based on their distance in the hierarchy, cosine similarity or both, extending previous approaches to more than two models [16].

Residual Selection Procedures For "residuals" to be useful during aggregation, they must contain information that is not already evident in the parameters of their parent. As mentioned, it is well-known that averaging is an imperfect means of aggregating models trained on Non-IID data [65, 10, 52, 48] as the

directions of different pseudo-gradients may conflict or even cancel each other. As such, in the case of leaf-to-root "residuals", parameters may be selected on the basis of simple metrics like an l_n norm with the assumption that larger absolute values correspond to more informative gradients, the per-sample loss they have averaged on the local data, or based on their relation to each other (e.g, cosine similarity).

Restructuring Algorithms Finally,

14 Completed 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. All the mentioned works are available as appendices to this proposal.

15 Fairness and Personalisation

Iacob et al. [35] 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.

Personalisation-aware FL functions in two phases, first it allows standard FL training to progress unimpeded. After near-convergence, it injects common personalisation regularises such as Knowledge Distillation [30, 99] or Elastic-weight Consolidation [45] into the local client loss function where the reference model is taken to be the federated model from the start of the round. This allows the model to learn from the distributions of highly heterogeneous clients without harming performance on the overall federated network which enables a better distribution of accuracy over clients without the harm to average performance that Fair FL is known to bring [49, 50]. While more effective than Fair FL, this regularisation-based approach is still limited by the goal of training a single global model without any intermediary level of personalisation between the federated model and fine-tuned local models.

16 Hierarchical multimodal Federated Human Activity Recognition

Iacob et al. [36] evaluated the performance of Federated Human Activity Recognition [81] 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 [99]. 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 federation; 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 the more flexible and potentially data-free methods (such as FedOPT with persistent models) considered in this proposal.

17 Simulation efficiency

Both of the previous works were implemented in the Flower [4] FL framework; however, the scale of experimentation required for fully validating B-HFL would be unfeasible on the publicly available simulation engine in the case of cross-device scenarios. The previous Virtual Client Engine (VCE) of Flower used the Ray [68] distributed execution engine for simulation, which is designed for few long-running ML tasks rather than the numerous and short-running client training of FL. Furthermore, Ray does not have a means of forcefully freeing GPU-memory allocated by an ML framework like PyTorch without killing a process. This resulted in slow training times, instability and frequent disk spillage.

To correct these issues, I have contributed to research on a new engine that doubles Flower simulations' throughput by intelligent ML-based client placement on GPUs. Since clients in FL are heterogeneous in terms of workloads, pre-processing pipelines and dataset size this required more than mere load-balancing based on sample count. The system functions by actively placing clients to workers on specific GPUs based on a log-linear model which estimates client training time on a given piece of hardware based on historic data. The workers then perform local averaging in order to minimise communication. This system results in up-to 400% improvements in FL training time compared to other FL frameworks like FedScale [46], original Flower [4], and Flute [17]. The paper "High-throughput Simulation of Federated Learning via Resource-Aware Client Placement" will be submitted to MLSys.

18 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.

19 Second year plan

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 [4] 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. [36] 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 12. 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 [14] together with two multi-language datasets [e.g., 55, 95] 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.

20 Third and fourth year plan

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 6, Cited on page 7, Cited on page 8, Cited on page 9

[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 14

[3] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *CoRR*, abs/1912.00818, 2019. URL http://arxiv.org/abs/1912.00818. Cited on page 2, Cited on page 6, Cited on page 8

[4] 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 13, Cited on page 14

[5] Abhishek Bhowmick, John C. Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *CoRR*, abs/1812.00984, 2018. URL http://arxiv.org/abs/1812.00984. Cited on page 3

[6] 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

[7] 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 https://www.usenix.org/conference/osdi18/presentation/nishihara. Cited on page 3, Cited on page 5, Cited on page 11

[8] 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 3, Cited on page 5, Cited on page 6, Cited on page 7, Cited on page 11

[9] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–9. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9207469. URL https://doi.org/10.1109/IJCNN48605.2020.9207469. Cited on page 6, Cited on page 8

[10] 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 1, Cited on page 4, Cited on page 5, Cited on page 6, Cited on page 8, Cited on page 11, Cited on page 12

[11] Fengwen Chen, Guodong Long, Zonghan Wu, Tianyi Zhou, and Jing Jiang. Personalized federated learning with a graph. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 2575–2582. ijcai.org, 2022. doi: 10.24963/ijcai.2022/357. URL https://doi.org/10.24963/ijcai.2022/357. Cited on page 12

[12] 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 1, Cited on page 5

[13] Gary Cheng, Karan N. Chadha, and John C. Duchi. Fine-tuning is fine in federated learning. *CoRR*, abs/2108.07313, 2021. URL https://arxiv.org/abs/2108.07313. Cited on page 6, Cited on page 8

[14] 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 14

[15] 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 3, Cited on page 10

[16] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *CoRR*, abs/2003.13461, 2020. URL https://arxiv.org/abs/2003.13461. Cited on page 2, Cited on page 6, Cited on page 8, Cited on page 10, Cited on page 11, Cited on page 12

[17] 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, Cited on page 14

[18] 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

[19] Chen Dun, Mirian Hipolito Garcia, Chris Jermaine, Dimitrios Dimitriadis, and Anastasios Kyrillidis. Efficient and light-weight federated learning via asynchronous distributed dropout. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pages 6630–6660. PMLR, 2023. URL https://proceedings.mlr.press/v206/dun23a.html. Cited on page 1, Cited on page 5

[20] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006. doi: 10.1007/11787006_1. URL https://doi.org/10.1007/11787006_1. Cited on page 3

[21] Robert French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3:128–135, 05 1999. doi: 10.1016/S1364-6613(99)01294-2. Cited on page 6

[22] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c4ede56bbd98819ae6112b20ac6bf145-Abstract.html. Cited on page 3

[23] 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 6, Cited on page 7, Cited on page 8

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

[25] 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 9

[26] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *CoRR*, abs/2002.05516, 2020. URL https://arxiv.org/abs/2002.05516. Cited on page 2, Cited on page 6

[27] 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 7

[28] 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

[29] 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 10

[30] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531. Cited on page 13

[31] 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

[32] 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

[33] Li Huang, Andrew L. Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *J. Biomed. Informatics*, 99, 2019. doi: 10.1016/j.jbi.2019.103291. URL https://doi.org/10.1016/j.jbi.2019.103291. Cited on page 6

[34] 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 3, Cited on page 5, Cited on page 8, Cited on page 9

[35] 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 13

[36] 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 13, Cited on page 14

[37] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *CoRR*, abs/1811.11479, 2018. URL http://arxiv.org/abs/1811.11479. Cited on page 3

[38] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. doi: 10.1080/01621459.1963.10500845. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500845. Cited on page 6

[39] 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 3, Cited on page 11

[40] 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, Cited on page 3

[41] 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

[42] 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

[43] 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 1

[44] 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

[45] 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, Cited on page 3, Cited on page 6, Cited on page 10, Cited on page 12, Cited on page 13

[46] 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, Cited on page 14

[47] Stefan Lee, Senthil Purushwalkam, Michael Cogswell, Viresh Ranjan, David J. Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2119–2127, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/20d135f0f28185b84a4cf7aa51f29500-Abstract.html. Cited on page 10

[48] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze, editors, *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. mlsys.org, 2020. URL https://proceedings.mlsys.org/book/316.pdf. Cited on page 3, Cited on page 5, Cited on page 12

[49] 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 5, Cited on page 13

[50] 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 13

[51] 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 6, Cited on page 8, Cited on page 10, Cited on page 12

[52] 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, Cited on page 12

[53] Zhengyang Li, Shijing Si, Jianzong Wang, and Jing Xiao. Federated split BERT for heterogeneous text classification. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*, pages 1–8. IEEE, 2022. doi: 10.1109/IJCNN55064.2022.9892845. URL https://doi.org/10.1109/IJCNN55064.2022.9892845. Cited on page 6

[54] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. Cited on page 3, Cited on page 10

[55] 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 14

[56] 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

[57] 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 7

[58] Jiachen Liu, Fan Lai, Yinwei Dai, Aditya Akella, Harsha V. Madhyastha, and Mosharaf Chowdhury. Auxo: Heterogeneity-mitigating federated learning via scalable client clustering. *CoRR*, abs/2210.16656, 2022. doi: 10.48550/arXiv.2210.16656. URL https://doi.org/10.48550/arXiv.2210.16656. Cited on page 2, Cited on page 6, Cited on page 7, Cited on page 8

[59] 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 6, Cited on page 7, Cited on page 8, Cited on page 9

[60] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28(2):129–136, 1982. doi: 10.1109/TIT.1982.1056489. URL https://doi.org/10.1109/TIT.1982.1056489. Cited on page 6

[61] Siqi Luo, Xu Chen, Qiong Wu, Zhi Zhou, and Shuai Yu. HFEL: joint edge association and resource allocation for cost-efficient hierarchical federated edge learning. *IEEE Trans. Wirel. Commun.*, 19(10):6535–6548, 2020. doi: 10.1109/TWC.2020.3003744. URL https://doi.org/10.1109/TWC.2020.3003744. Cited on page 7, Cited on page 8

[62] 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, Cited on page 3

[63] 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 3

[64] 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 6, Cited on page 8, Cited on page 10

[65] 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 3, Cited on page 7, Cited on page 8, Cited on page 12

[66] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. 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=BJ0hF1Z0b. Cited on page 3, Cited on page 4

[67] Naram Mhaisen, Alaa Awad Abdellatif, Amr Mohamed, Aiman Erbad, and Mohsen Guizani. Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints. *IEEE Trans. Netw. Sci. Eng.*, 9(1):55–66, 2022. doi: 10.1109/TNSE.2021.3053588. URL https://doi.org/10.1109/TNSE.2021.3053588. Cited on page 3, Cited on page 6, Cited on page 7, Cited on page 8

[68] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: A distributed framework for emerging AI applications. In Andrea C. Arpaci-Dusseau and Geoff Voelker, editors, *13th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2018, Carlsbad, CA, USA, October 8-10, 2018*, pages 561–577. USENIX Association, 2018. Cited on page 13

[69] 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 5, Cited on page 8, Cited on page 9

[70] 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

[71] 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

[72] 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

[73] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inf. Forensics Secur.*, 13(5):1333–1345, 2018. doi: 10.1109/TIFS.2017.2787987. URL https://doi.org/10.1109/TIFS.2017.2787987. Cited on page 3

[74] Alexandre Ramé, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/46108d807b50ad4144eb353b5d0e8851-Abstract-Conference.html. Cited on page 10

[75] 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 11, Cited on page 12

[76] 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

[77] 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

[78] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural Networks Learn. Syst.*, 32(8):3710–3722, 2021. doi: 10.1109/TNNLS.2020.3015958. URL https://doi.org/10.1109/TNNLS.2020.3015958. Cited on page 6, Cited on page 8

[79] 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

[80] Jinhyun So, Corey J. Nolet, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E. Ali, Basak Guler, and Salman Avestimehr. Lightsecagg: a lightweight and versatile design for secure aggregation in 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/d2ddea18f00665ce8623e36bd4e3c7c5-Abstract.html. Cited on page 3

[81] 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 13

[82] 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 6

[83] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8432–8440. AAAI Press, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/20819. Cited on page 6

[84] 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 7

[85] 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

[86] 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

[87] 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 6

[88] Zhiyuan Wang, Hongli Xu, Jianchun Liu, He Huang, Chunming Qiao, and Yangming Zhao. Resource-efficient federated learning with hierarchical aggregation in edge computing. In *40th IEEE Conference on Computer Communications, INFOCOM 2021, Vancouver, BC, Canada, May 10-13, 2021*, pages 1–10. IEEE, 2021. doi: 10.1109/INFOCOM42981.2021.9488756. URL https://doi.org/10.1109/INFOCOM42981.2021.9488756. Cited on page 7, Cited on page 8

[89] 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 3, Cited on page 11

[90] 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

[91] Qiong Wu, Xu Chen, Zhi Zhou, and Junshan Zhang. Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring. *IEEE Trans. Mob. Comput.*, 21(8):2818–2832, 2022. doi: 10.1109/TMC.2020.3045266. URL https://doi.org/10.1109/TMC.2020.3045266. Cited on page 3

[92] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386. URL https://doi.org/10.1109/TNNLS.2020.2978386. Cited on page 12

[93] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *CoRR*, abs/1903.03934, 2019. URL http://arxiv.org/abs/1903.03934. Cited on page 3, Cited on page 4, Cited on page 5, Cited on page 8

[94] 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 1, Cited on page 5, Cited on page 9

[95] 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 14

[96] 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

[97] 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 6, Cited on page 8, Cited on page 12

[98] 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

[99] 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 6, Cited on page 10, Cited on page 12, Cited on page 13

[100] 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 6, Cited on page 9

[101] Tailin Zhou, Zehong Lin, Jun Zhang, and Danny H. K. Tsang. Understanding model averaging in federated learning on heterogeneous data. *CoRR*, abs/2305.07845, 2023. doi: 10.48550/arXiv.2305.07845. URL https://doi.org/10.48550/arXiv.2305.07845. Cited on page 4, Cited on page 5, Cited on page 9, Cited on page 12

[102] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14747–14756, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html. Cited on page 3