

No One Left Behind: Inclusive Federated Learning over Heterogeneous Devices

Ruixuan Liu
School of Information,
Renmin University of
China
Beijing, China
ruixuan.liu@ruc.edu.cn

Fangzhao Wu*
Microsoft Research Asia
Beijing, China
wufangzhao@gmail.com

Chuhan Wu
Department of Electronic
Engineering, Tsinghua
University
Beijing, China
wuchuhan15@gmail.com

Yanlin Wang
Microsoft Research Asia
Beijing, China
yanlwang@microsoft.com

Lingjuan Lyu
Sony AI
Tokyo, Japan
lingjuan.lv@sony.com

Hong Chen
School of Information,
Renmin University of
China
Beijing, China
chong@ruc.edu.cn

Xing Xie
Microsoft Research Asia
Beijing, China
xingx@microsoft.com

ABSTRACT

Federated learning (FL) is an important paradigm for training global models from decentralized data in a privacy-preserving way. Existing FL methods usually assume the global model can be trained on any participating client. However, in real applications, the devices of clients are usually heterogeneous, and have different computing power. Although big models like BERT have achieved huge success in AI, it is difficult to apply them to heterogeneous FL with weak clients. The straightforward solutions like removing the weak clients or using a small model to fit all clients will lead to some problems, such as under-representation of dropped clients and inferior accuracy due to data loss or limited model representation ability. In this work, we propose InclusiveFL, a client-inclusive federated learning method to handle this problem. The core idea of InclusiveFL is to assign models of different sizes to clients with different computing capabilities, bigger models for powerful clients and smaller ones for weak clients. We also propose an effective method to share the knowledge among local models with different sizes. In this way, all the clients can participate in FL training, and the final model can be big and powerful enough. Besides, we propose a momentum knowledge distillation method to better transfer knowledge in big models on powerful clients to the small models on weak clients. Extensive experiments on many real-world benchmark datasets demonstrate the effectiveness of InclusiveFL in learning accurate models from clients with heterogeneous devices under the FL framework.

*The corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).
KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08...\$15.00
<https://doi.org/10.1145/3534678.3539086>

CCS CONCEPTS

• **Computing methodologies** → **Distributed computing methodologies**; **Distributed artificial intelligence**.

KEYWORDS

Federated learning, Heterogeneous device, Knowledge distillation

ACM Reference Format:

Ruixuan Liu, Fangzhao Wu, Chuhan Wu, Yanlin Wang, Lingjuan Lyu, Hong Chen, and Xing Xie. 2022. No One Left Behind: Inclusive Federated Learning over Heterogeneous Devices. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, August 14–18, 2022, Washington, DC, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3534678.3539086>

1 INTRODUCTION

Nowadays, huge amounts of data are generated every day from a wide range of devices such as mobile phones, autonomous vehicles, or servers [26]. Exchanging data from different sources to train machine learning models is attractive but challenging under the privacy regularization such as GDPR¹. Based on the principles of focused collection and data minimization to avoid privacy issues, federated learning (FL) [25] emerges as a privacy-aware paradigm to train a global model while keeping decentralized data stay on local devices. In each round, sampled clients pull a copy of the global model and optimize models in parallel on local datasets. Then, the server aggregates local models to update the global model. Hence, conventional federated learning methods [25, 31] make an essential assumption that all clients have sufficient local resources to train models with the same architecture.

However, as shown in Fig. 1, in the real world, devices of clients are usually heterogeneous and they may have the significantly different computing power and memory size [16, 37, 47]. Although a large global model (e.g., BERT with 12-layer Transformer [18, 23]) is promising to achieve a good performance, it is difficult to apply it in heterogeneous FL because weak clients that are constrained by hardware resources cannot afford the cost of local training. A

¹<https://gdpr-info.eu/>

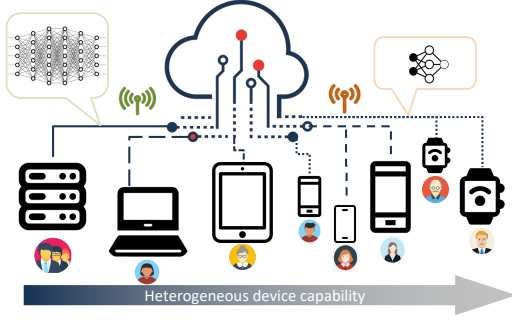


Figure 1: InclusiveFL over heterogeneous devices.

straightforward way [1] is to drop weak clients and only aggregate parameters from powerful clients with sufficient resources. But excluding data on weak devices in training leads to fairness issues [13, 28] because weak clients would be underrepresented in the final global model. Furthermore, the loss of data on weak clients causes an inferior accuracy [29], especially when the amount of weak clients is enormous (e.g., in less developed areas). Alternatively, a client-inclusive baseline is choosing a small global model to fit the minimum capability that all clients can offer. However, the representation ability of the final global model would be largely limited by the small model architecture [24].

Due to the limitations of the above methods, it is attractive to include all clients with heterogeneous devices while utilizing the capabilities of large models. Thus, an intuitive way is training bigger local models on powerful clients and smaller local models on weak clients. To exchange knowledge over heterogeneous models, several works [5, 20, 36] utilize the knowledge distillation technique [15] to enhance the global model with an ensemble of local predictions. However, they assume clients share a public proxy dataset, which is impractical for weak clients with a limited memory. A recent work *HeteroFL* [7] does not require a proxy dataset and proposes to train heterogeneous local models by averaging the overlapped sub-matrices of all parameter matrices across various models. However, this method is not beneficial for retaining large models' knowledge in smaller models because the width pruning operation breaks the original model structure. In addition, even the same sub-matrix may have different behaviors in the small and large models due to the model structure differences, which may lead to a suboptimal performance due to the mismatch of feature spaces. Moreover, simply sharing parts of parameters across different models cannot effectively transfer useful knowledge encoded by strong models to other weaker models [15]. Thus, how to learn models with different sizes and aligned feature spaces via effective knowledge transfer is a challenging problem in FL with heterogeneous devices.

In this paper, we propose a client-inclusive federated training solution *InclusiveFL* that can train a large model over devices with heterogeneous capabilities, and address the above key challenges. *InclusiveFL* assigns models of different sizes to clients with different computing capabilities, bigger models for powerful clients and smaller ones for weak clients. To eliminate the mismatch between

small and large models, we propose to share shallow bottom model layers in the largest model with other smaller models. It has been empirically investigated that the lower layers are most critical for retaining the pre-trained knowledge in downstream tasks [32], and the feature spaces among different layers have much consistency [8]. We propose a layer-wise heterogeneous aggregation method to update the parameters of shared bottom layers. Furthermore, since the small model on weak clients may not be strong enough, we propose to transfer the knowledge learned by large models on powerful clients to small models on weak clients via a momentum distillation method [46]. Intuitively, by encouraging the top encoder layer in small models to imitate the behavior of top encoder layers in a larger model, we can achieve effective knowledge transfer in *InclusiveFL*. Contributions of this paper are summarized as follows:

- We propose *InclusiveFL* for effective federated learning over heterogeneous client devices. Clients with different computing capabilities are assigned models of different sizes, and all of them can contribute to learning a large and powerful global model.
- We propose an effective method to share the knowledge in the heterogeneous models from heterogeneous clients. Besides, we propose a momentum knowledge distillation method to boost knowledge sharing from large models to small models.
- We conduct extensive experiments to verify the effectiveness of the proposed method in learning an accurate global model from heterogeneous clients in the FL framework.

2 BACKGROUND AND RELATED WORK

Federated learning [25] is a machine learning technique that trains an algorithm across multiple decentralized edge devices (cross-device) [14, 27, 48] or servers holding local data samples (cross-silo) [6, 41, 45], without exchanging them. Conventional federated learning protocol typically adopts *FedAvg* [25] to aggregate local parameters as follows:

$$w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k,$$

where $n = \sum_{k=1}^K n_k$ denotes the total number of local samples. For a better training convergence, *FedAdam* [31] is recently proposed to use adaptive methods on the server and SGD on the clients. For each round, a sampled client i computes the local update g_t^i after iterating over several rounds of local SGDs. Then, the server averages local updates $g_t \leftarrow \frac{1}{|S|} \sum_{i \in |S|} g_t^i$ and updates the global model as follows:

$$\begin{aligned} m_t &\leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &\leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\ w_{t+1} &\leftarrow w_t + \eta \frac{m_t}{\sqrt{v_t} + \tau}. \end{aligned}$$

Basic federated learning methods make an essential assumption that all clients have sufficient local resources to train models with the same architecture. However, weak clients cannot even compute over the global model when local hardware is constrained. Previous works [5, 20] provide solutions for federated learning with heterogeneous local models based on knowledge distillation [15].

A server student model is updated by distilling from the ensemble of predictions [21]. Alternatively, local models are optimized sequentially [20] or jointly [5] over the augmented dataset and the local private dataset. However, it is hard to guarantee the domain and quality of the public data. And it is impractical for weak clients with limited hardware to store the public dataset.

Recently, *HeteroFL* [7] removes this assumption and conduct parameter-averaging over heterogeneous models. Authors scale model architectures for clients with variant local resources with variant widths. For each matrix $W \in \mathbb{R}^{d_g \times k_g}$ in the large model, they reduce the parameters for a smaller model as the upper left sub-matrix $W_s = W[:d_s, :k_s]$ with size $d_s \times k_s$. To exchange knowledge, heterogeneous models share the overlapped matrix with a parameter averaging. However, *HeteroFL* has limitations of a biased initial pruning and parameter mismatch across heterogeneous models. Instead, we reduce number of layers for small models and propose methods to solve these problems.

Other related works that consider weak capabilities include solutions to reduce the communicational cost [20, 33]. Several works aim to mitigate the up-link communication bottlenecks from client to server [10, 19, 22, 39, 40]. Local model updates can be sparsified, quantized or randomly sub-sampled before uploading to the central server. Konečný et al. [19] performs a lossy compression on the model updates and reduces the error incurred by the subsequent quantization with the randomized Hadamard transform. To reduce the server-to-client communication, Caldas et al. [3] extends the idea of dropout [35] and drops a fixed number of activation units at each full-connected layer before distributing a local model to each client. Bouacida et al. [2] adopts an adaptive strategy and selects the dropped activation based on the activation score map. It should be noted that local optimization is conducted on decompressed models. Also, above works do not consider federated training over heterogeneous models and are compatible to our solution when aggregating local models with the same architecture in *InclusiveFL*.

3 METHODOLOGY

In this section, we introduce our solution *InclusiveFL*, which trains a global model across client devices with heterogeneous capabilities. We will first introduce the layer-wise heterogeneous aggregation framework in *InclusiveFL*, and then introduce details of our proposed momentum distillation in federated learning for more effective knowledge exchange in the heterogeneous setting.

3.1 Layer-wise Heterogeneous Aggregation Framework

The overview of *InclusiveFL* framework is shown in Fig. 2. Without loss of generality, we denote three groups that hold different device capabilities as weak, medium, and strong clients. Our core idea is to assign models of different sizes to clients with different computing capabilities, bigger models for strong clients and smaller models for weak clients. We reduce the number of parameters for weaker clients by decreasing the depth of network. Since the bottom layers are most critical for retraining the pre-trained knowledge [32], the sub-model with pre-trained bottom layers would benefit from the pre-trained parameters, which results in better initialization for small models. Accordingly, the clients in weak, medium or strong

Algorithm 1 Framework of *InclusiveFL*.

Server executes:

```

1: Initialize  $\theta_0^c, f_0^a, f_0^b, f_0^c, m_0^b \leftarrow 0, m_0^c \leftarrow 0$ 
2: Set shared layers  $\theta_0^a \leftarrow \theta_0^{c,1:L_a}, \theta_0^b \leftarrow \theta_0^{c,1:L_b}$ 
3: Get models  $w_0^a \leftarrow \theta_0^a \circ f_0^a, w_0^b \leftarrow \theta_0^b \circ f_0^b, w_0^c \leftarrow \theta_0^c \circ f_0^c$ 
4: for  $t \in \{1, \dots, T\}$  do
5:    $U_t \leftarrow$  (randomly sample a portion of  $r$  users)
6:    $U_t^a, U_t^b, U_t^c \leftarrow$  (group  $U_t$  into the same device type)
7:    $n_a, n_b, n_c \leftarrow$  (the number of users for each group)
8:   // Homomorphic aggregation
9:   for device type  $j \in \{a, b, c\}$  do
10:    for user  $i \in U_t^j$  do
11:      Pull the sub-global model  $w_t^j$ 
12:       $g_t^{i,j} \leftarrow \text{LocalUpdate}(w_t^j, D_i)$   $\triangleright D_i$  is the local dataset
13:    end for
14:     $g_t^j \leftarrow \frac{1}{|U_t^j|} \sum_{i \in U_t^j} g_t^{i,j}$ 
15:    if  $j$  is not  $c$  then
16:       $k \leftarrow$  (the device type that is greater than  $j$ )
17:       $g_t^{j,L_j} \leftarrow \beta \cdot m_{t-1}^k + (1 - \beta) \cdot g_t^{j,L_j}$ 
18:    end if
19:     $w_{t+1}^j \leftarrow \text{FedAdam}(g_t^j, w_t^j)$   $\triangleright w_{t+1}^j$  includes  $\theta_{t+1}^j \circ f_{t+1}^j$ 
20:    if  $j$  is not  $a$  then
21:       $L \leftarrow$  (the number of layers in the next smaller model)
22:       $m_t^j \leftarrow \frac{1}{L_j - L + 1} \cdot \sum_{l \in [L - L_j, L_j]} g_t^{j,l}$ 
23:    end if
24:  end for
25:  // Heterogeneous aggregation
26:   $\theta_{t+1}^{c,l} \leftarrow \sum_{i \in \{a,b,c\}} \frac{n_i}{n_a + n_b + n_c} \theta_{t+1}^{i,l}$  for  $l \in [1, \dots, L_a - 1]$ 
27:   $\theta_{t+1}^{c,l} \leftarrow \sum_{i \in \{b,c\}} \frac{n_i}{n_b + n_c} \theta_{t+1}^{i,l}$  for  $l \in [L_a, \dots, L_b - 1]$ 
28:  Update  $\theta_{t+1}^a \leftarrow \theta_{t+1}^{c,1:L_a-1} \circ \theta_{t+1}^{a,L_a}, \theta_{t+1}^b \leftarrow \theta_{t+1}^{c,1:L_b-1} \circ \theta_{t+1}^{b,L_b}$ 
29:   $w_{t+1}^a \leftarrow \theta_{t+1}^a \circ f_{t+1}^a, w_{t+1}^b \leftarrow \theta_{t+1}^b \circ f_{t+1}^b, w_{t+1}^c \leftarrow \theta_{t+1}^c \circ f_{t+1}^c$ 
30: end for
```

client groups own small, medium or a large local models with L_a, L_b, L_c layers, where $L_a < L_b < L_c$. In general, we conduct a homomorphic aggregation within the same client group and denote the aggregated model as the sub-global models w^a, w^b , and w^c . Then we run a heterogeneous aggregation across different client groups to update all sub-global models.

As shown in Algorithm 1, in the t^{th} round of communication, the server first randomly samples a subset of clients with heterogeneous devices. The sampled clients are grouped by the device type as U_t^a, U_t^b and U_t^c and the proportion $n_a : n_b : n_c$ for each device type is dynamic for each round. Then, for a client i in the group of device j , we conduct the local optimization with SGD as in conventional FL [25] and get the local update as $g_t^{i,j}$. Within each client group, we average the update as g_t^j , where $j \in \{a, b, c\}$. Then we update the sub-global model into w_{t+1}^j with the averaged updates and the momentum m_{t-1}^k to merge local knowledge from clients with the same device type. The momentum m_{t-1}^k is distilled from a larger model on client group k and we explain details in Section 3.2.

With sub-models $w_{t+1}^a, w_{t+1}^b, w_{t+1}^c$ after the homomorphic aggregation over local models with an identical architecture, we start the heterogeneous aggregation from Line 25 in Algorithm 1 to exchange knowledge across variant model architectures. It should be

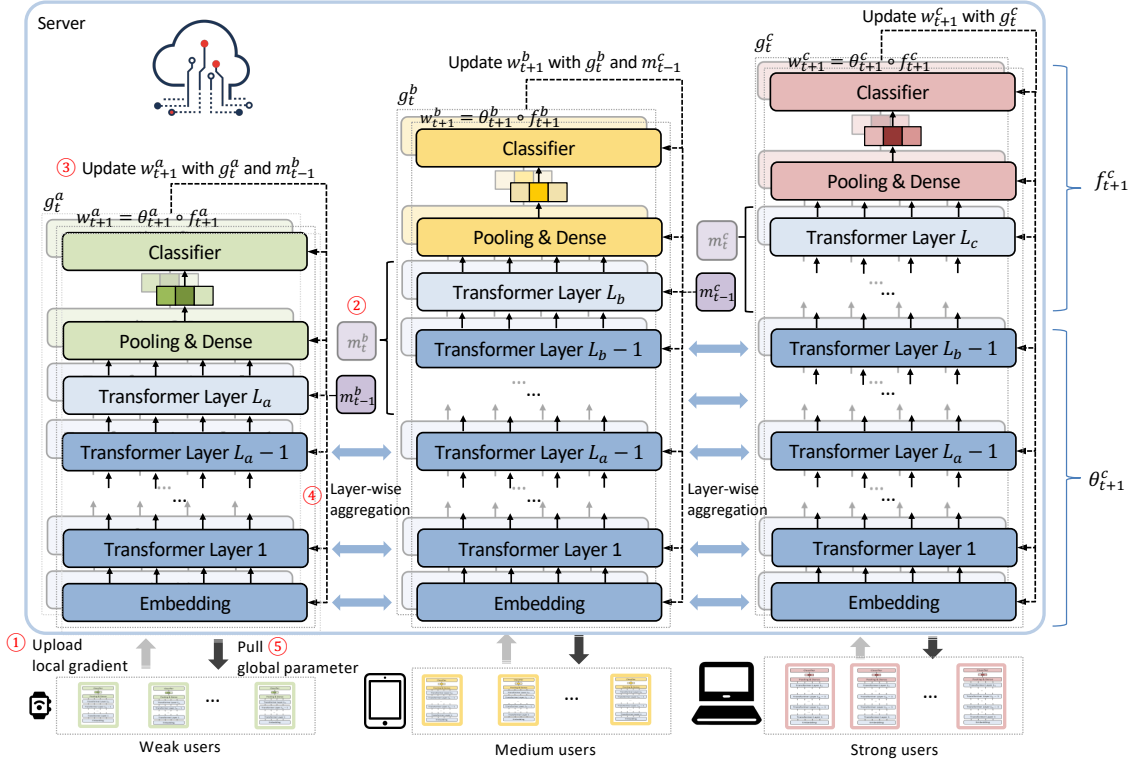


Figure 2: Overview of *InclusiveFL*.

noted that we decompose the model w into a shared module θ and an isolated module f for each group. The shared module includes bottom encoding layers θ and the isolated module includes the last encoder layer and top task-specific layers, such as the pooling and dense layer and the classifier layer in Fig. 2. *InclusiveFL* is a general framework for other tasks by only changing the architecture of the top layers f . Thus, we first exclude the updated f_{t+1} in the heterogeneous parameter aggregation and conduct the standard *FedAvg* for each overlapped layer in shared module θ_{t+1}^c , as shown in Line 26 and 27. Specifically, we use the notation $\theta_{t+1}^{i,l}$ to indicate the l^{th} encoder layer in the model for device type $j \in \{a, b, c\}$. Similarly, layers from 1 to $L_a - 1$ in the sub-global model with type c is denoted as $\theta_{t+1}^{c,1:L_a-1}$. Combining with the within-group updated top layers f_{t+1} for $\{a, b, c\}$ in Line 29, the server derives the updated sub-global model and distributes to clients for the next round.

Intuitively, the bottom layers can be shared because they can capture basic features across heterogeneous models. Also, the overlapped layers across three models have the same initialization, which help to align them keep aligned from the beginning of training. We are motivated to isolate the last encoder layer and top layers for two considerations. First, isolating top layers help mitigate the parameter mismatch in each sub-global model. For the example of a text classification task in Fig. 2, the bottom $L_a - 1$ layers in w^a will produce a sequence of hidden states as the input to a pooling and dense layer for getting a sentence embedding. However, the output of the bottom $L_a - 1$ layers in w_b can be further extracted with other

$L_b - L_a + 1$ layers before the pooling and dense layer. Thus, the top parameters behave differently due to the heterogeneous model architectures. If sharing f_{t+1}^a with f_{t+1}^b and f_{t+1}^c , the averaged model applied to w^a would lack several layers of feature extraction, which would lead to a poor performance of w_a . Second, isolating top layers makes *InclusiveFL* capable of training separate local model with different target labels. For example in the cross-silo scenario such as federated named entity recognition [11], medical data stored at different medical institutions may have different characteristics and annotation criterion, which leads to personalized top layers.

3.2 Momentum Distillation

Now we introduce details of the momentum distillation for accelerating the heterogeneous federated training, as shown from Line 15 to 23 in Algorithm 1. Intuitively, a larger model is more capable of extracting knowledge from data than a small model. Thus, we are motivated to distill knowledge from larger models to smaller models. The knowledge distillation [15] in the centralized setting typically uses a hidden loss to align hidden representations of a small student model and a large teacher model. However, the hidden loss does not apply to the heterogeneous federated learning with data separated for the small model and teacher model.

Instead, we notice that the key to align small models and large models when bottom layers are shared is to align the behaviors of the isolated encoder layer in a small model to the corresponding top layers in the larger model. Hence, taking the small model and

the medium model for instance, we calculate a gradient momentum as the average over updates for a stack of layers from L_a to L_b in the medium sub-global model. For the averaged update g_t^a , we inject a distilled gradient momentum m_{t-1}^b to the update g_t^{a,L_a} of the isolated encoder layer L_a , which helps the L_a layer in the small model imitate the block of layer L_a to L_b in the medium model. Similarly, we use the averaged update from the large model as the gradient momentum for the layer L_b in the medium model. In general, the distillation transfers knowledge from a larger model to a smaller model without the limitation on the number of model types. The gradient momentum is initialized as zero and applied with a momentum factor β , which is an important hyper-parameter to tune the strength of the distillation.

4 EXPERIMENTS

In this section, we evaluate *InclusiveFL* for federated learning over heterogeneous devices on IID and non-IID datasets and compare with both naive baselines and HeteroFL [7]. To simulate the heterogeneous device scenario, we set up three types of clients: weak clients, medium clients and powerful clients. Accordingly, they can store and compute over 4-layer, 8-layer and 12-layer transformer models, respectively.

Datasets. Our experiments are conducted on the text classification task in the GLUE benchmark and token classification task (NER) on three medical corpora. GLUE benchmark [38] is a collection of multiple natural language understanding tasks, including MNLI [44] (inference), SST-2 [34] (sentimental analysis), MRPC [9] (paraphrase detection), CoLA [42] (linguistic acceptability), QNLI [30] (inference), QQP² (question-answering), RTE³ (inference), and STSB [4] (textual similarity). The generality makes the GLUE dataset a standard benchmark to evaluate NLU models.

Three medical datasets includes SMM4H [43], CADEC [17] and ADE [12], which are widely used to evaluate NER tasks. The statistics of the medical datasets are shown in Table 1.

Table 1: Statistics of the medical NER datasets.

Dataset	# Sentences	Entity Types	# Entity
SMM4H	3,824	ADE (1,707)	1,707
ADE	4,483	ADE (5,678), Drug (5,076), Dosage (222)	10,976
CADEC	7,683	ADE (5,937), Drug(1,796), Disease(282), Finding(425), Symptom(268)	8,535

Evaluation Protocols. For the evaluation on cross-device [26] federated learning with IID data, we randomly split the training dataset of each GLUE benchmark to 1000 clients and report the best performance of the evaluation dataset. By default, we report the matched accuracy for MNLI, the Matthew’s correlation for CoLA, Pearson correlation for STSB, F1-score for MRPC and the accuracy metric for all other tasks. Higher value indicates better performance. To evaluate over non-IID data distribution, we take each medical

corpus as the local dataset of one client (e.g. a hospital) and simulate the cross-silo [26] federated setting. We set SMM4H as the local data for a weak client, ADE as the local data for a medium client and CADEC as the local data for the powerful client. We validate the generality of this setting by rotating datasets and client types and draw similar observations. For each client, we randomly take 80% of the local dataset as the training dataset and the rest as the evaluation dataset. We use the sequeval framework⁴ to report the maximum averaged F1-score, precision, recall and accuracy on the evaluation dataset to indicate the overall performance or the performance over each type of entity for each client. Results for the two settings are reported by averaging over 5 independent repeats.

Baselines. We compare our proposed *InclusiveFL* solution with the following naive baselines:

- *AllLarge*: an ideal baseline to indicate the performance upper bound of training over heterogeneous devices. We remove the resource limitation on weak and medium clients and conduct federated training over all clients participant with a 12-layer RoBERTa model.
- *AllSmall*: a naive baseline to include all clients with heterogeneous devices and conduct training over the smallest model with 4 layers.
- *ExclusiveFL*: a naive baseline where weak and medium clients are dropped off in federated training due to the resource limitation on the largest model.
- *Local*: a naive baseline for cross-silo setting where each client trains the model with the size that fits their local resources.
- *HeteroFL* [7]: the most related work for training across all clients over heterogeneous devices. The number of parameters in smaller models are reduced in width by cropping each parameter matrix of the largest model into sub-matrix with various reduction radio.

Hyper-Parameter Settings. For federated training over heterogeneous devices, we assign each client a client type with equal probability before training and fix it during training by default. The effect of different ratios of device types is investigated in our hyper-parameter analysis. For a fair comparison, we keep the number of parameters for each model type in *HeteroFL* equal to 4-layer, 8-layer and 12-layer models in other baselines by reducing the hidden size from 768 to 456, 624 for weak and medium clients respectively. For *InclusiveFL*, we set the momentum factor $\beta = 0.2$ for all GLUE benchmark and $\beta = 0.5$ for medical NER datasets.

For GLUE dataset, we randomly sample a fraction of 0.02 clients in each round with $R = 1000$. For NER datasets, we conduct 5 local steps with batch size 64 in each communication round with $R = 100$. We set the same learning rate for each type of the model over all baselines. We use FedAdam [31] as the global aggregation protocol over homogeneous models for all baselines.

4.1 Performance Evaluation

We evaluate the effectiveness of proposed methods by comparing the averaged best performance that each method can achieve after convergence within 1,000 training rounds. Then we evaluate the efficiency and analyze the reason for improvement by showing the

²<https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs>

³<https://huggingface.co/datasets/glue/viewer/rte/train>

⁴<https://github.com/chakki-works/sequeval>

Table 2: Performance comparison of federated learning methods on the GLUE benchmark for text classification tasks. Performance is evaluated on the final large model in a global inference scenario. Bold faces indicate the best method for inclusive FL training with heterogeneous devices.

	Inclusive?	COLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSB	Avg.
All-Large	N/A	63.03	86.48	91.50	92.09	91.49	76.12	94.43	90.60	85.72
ExclusiveFL	No	37.77	85.98	89.87	91.24	89.47	62.17	94.06	89.26	79.98
All-Small	Yes	34.91	78.83	82.50	85.93	79.37	58.94	90.14	83.68	74.29
HeteroFL	Yes	8.15	31.83	81.51	62.70	73.79	52.71	84.98	30.54	53.28
InclusiveFL	Yes	54.85	86.36	91.42	91.76	90.55	66.14	94.17	89.94	83.15

Table 3: Performance comparison of federated learning methods on medical datasets for named entity recognition. Performances are evaluated for each named entity on each local model (SMM4H with 4-layer model, ADE with 9-layer model and CADEC with 12-layer model) in the local inference scenario. Best method for heterogeneous training is highlighted in bold face.

Methods	Inclusive?	SMM4H	ADE			CADEC					Avg.
		ADE	Drug	ADE	Dose	ADE	Symptom	Drug	Disease	Finding	
AllLarge	N/A	55.08	95.21	80.91	18.97	71.23	43.37	90.43	33.55	29.85	57.62
Local	No	46.77	95.12	80.14	11.25	44.15	19.90	53.53	20.58	16.53	43.11
AllSmall	Yes	49.56	92.76	75.67	13.51	64.82	26.71	87.91	21.07	20.23	40.89
HeteroFL	Yes	27.94	73.90	37.57	11.57	59.90	26.92	84.58	25.77	19.88	42.51
InclusiveFL	Yes	49.90	95.49	80.34	13.91	71.10	40.91	90.21	39.34	22.97	56.02

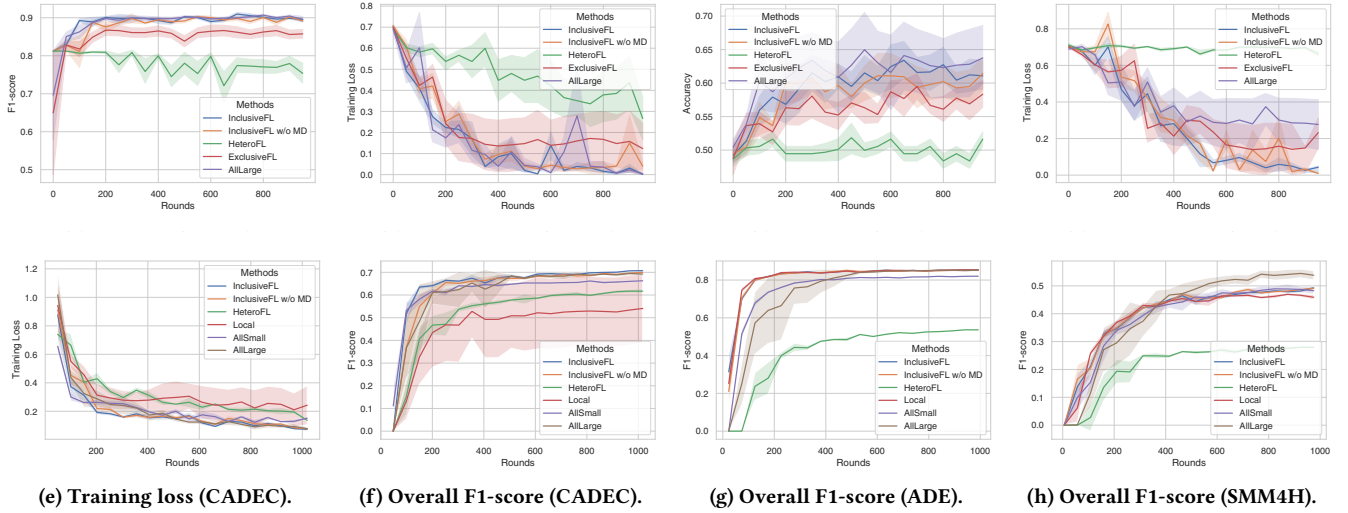


Figure 3: Federated training convergence and best performance. Error band shows the standard error of 5 independent repeats.

Table 4: Averaged number of rounds for achieving the best performance in Table 2.

Methods	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST2	STSB	NER
InclusiveFL-w/o MD	630	665	616	600	850	620	810	870	90
InclusiveFL	580	570	515	585	740	510	520	805	75

convergence curve and the average rounds required to achieve the best performance.

Effectiveness. Results of IID setting on GLUE datasets and results of non-IID setting on medical NER datasets are shown in

Table 2 and 3 respectively. We observe that for all datasets, *AllLarge* achieves the overall best performance, which indicates that training a large global model with all clients' data can achieve the optimal

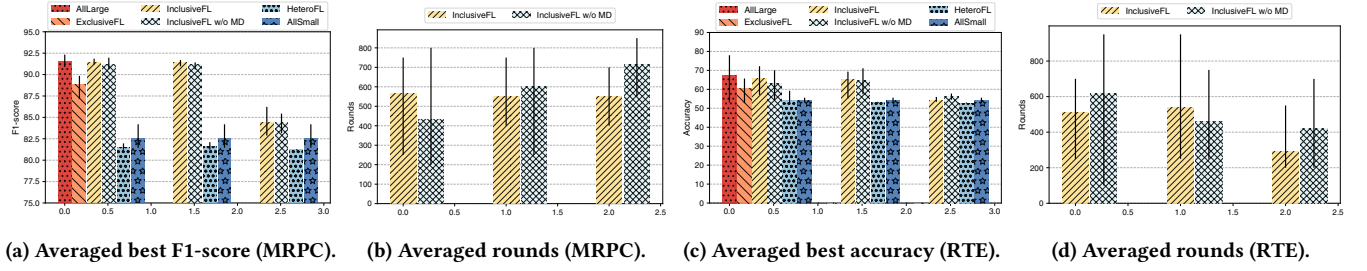


Figure 4: Local inference performance in cross-device federated learning.

performance. However, it is impractical for training participants with less resources to train large models locally.

For all other approaches that can be implemented across heterogeneous devices, our proposed approach *InclusiveFL* achieves the best performance. It is reasonable because involving large models (*InclusiveFL*) is better than only utilizing small models (*AllSmall*) and training over all data (*InclusiveFL*) is better than only training over partial data (*ExclusiveFL* / *Local*). This observation also validates the effectiveness of our proposed momentum distillation.

We find that the most related work *HeteroFL* [7] performs worse than the naive baselines *AllSmall* and *ExclusiveFL* on all GLUE datasets. And the averaged performance of *HeteroFL* is also inferior to *AllSmall* and *ExclusiveFL* on medical datasets as shown in Table 3. This indicates that the method of sharing the top-left sub-matrix across heterogeneous devices in *HeteroFL* may work for simple models with shallow layers (such as CNN in [7]) but not for more complex deep models (such as 12-layer RoBERTa). Furthermore, we observe in Fig. 3 that the starting performance of local models on weak and medium clients *InclusiveFL* is better than *HeteroFL*, which indicates that the small model in *InclusiveFL* has a better parameter initialization than *HeteroFL*. Thus, one reason of the inferior utility is that cropping the network in *HeteroFL* cannot fully utilized the pre-trained parameters while a layer-wise sharing in *InclusiveFL* can better maintain the utility of a pre-trained model naturally.

In addition, we observe that *ExclusiveFL* performs better than *AllSmall* on GLUE benchmark and *Local* performs worse than *AllSmall*. Thus, the performance gain from including more data and the gain from utilizing a large global model depends on the task and are hard to trade-off. This observation enhances the contribution of *InclusiveFL* for finding an effective way to aggregate over heterogeneous models and utilize performance gain from both inclusive data training and large models.

Efficiency. By observing the convergence curve in Fig. 3, we find *InclusiveFL* is approaching the performance of *AllLarge* with a faster speed than *InclusiveFL-w/o MD*, especially for the largest model with 12 layers. For the IID setting as shown in Fig. 3a, *InclusiveFL* achieves the F1-score of 89.43 within 400 rounds while *InclusiveFL-w/o MD* requires 550 rounds. For the non-IID setting in Fig. 3f, *InclusiveFL* achieves the accuracy of 60.5 within 650 while *InclusiveFL-w/o MD* requires 950 rounds. For all datasets, we list the averaged number of rounds required by *InclusiveFL* with or without momentum distillation to achieve the best performance in Table 2 and 3. Thus, we can conclude that the proposed momentum

Table 5: Influence of different proportions of device types on the CoLA dataset.

	1:1:1	1:2:7	7:2:1
AllLarge	63.03	63.03	63.03
AllSmall	34.91	34.91	34.91
ExclusiveFL	37.77	59.15	36.87
HeteroFL	8.15	9.60	N/A
InclusiveFL w/o MD	52.69	60.35	N/A
InclusiveFL	54.85	61.23	34.96*

distillation accelerates the performance of the largest global model by requiring less communication rounds.

Discussion on the local model inference. To further investigate the reason for performance improvement, we observe the evaluation results on the local model for each group of clients. For the cross-silo setting, local model inference is essential because each client has her own top layers of classifier for fitting local targets. For the medium model in Fig. 3g and small model in 3h, we find the effect of boosting convergence is less obvious but still exists and leads to a higher performance in Table 3 when the local model (8-layer for ADE and 4-layer for SMM4H) converges.

In the cross-device setting, the local model inference is also necessary when the inference is conducted on personal devices. As shown in Fig. 4, the final global model of *ExclusiveFL* can only be deployed on devices of powerful clients while client-inclusive federated learning solutions (*InclusiveFL*, *HeteroFL*) and *AllSmall* can be deployed on other clients with less storage and computation power. We notice that *InclusiveFL* performs best on each sub-global model, which validates the advantage of *InclusiveFL* in local inference. This also explains the performance advantage than another heterogeneous aggregation method *HeteroFL* in a global inference scenario, because better sub-global models lay the foundation for a better large inference model. In addition, the averaged rounds for achieving the best F1-score for MRPC and best accuracy for RTE is less for *InclusiveFL* than *InclusiveFL-w/o MD*, which shows the advantage of the proposed momentum distillation.

4.2 Ablation Study

To validate the principle of “No one left behind”, we conduct the ablation study in Fig. 5 by ignoring each type of devices in heterogeneous federated learning on both GLUE benchmark and medical NER datasets. For non-IID setting, *InclusiveFL-w/o weak* for SMM4H,

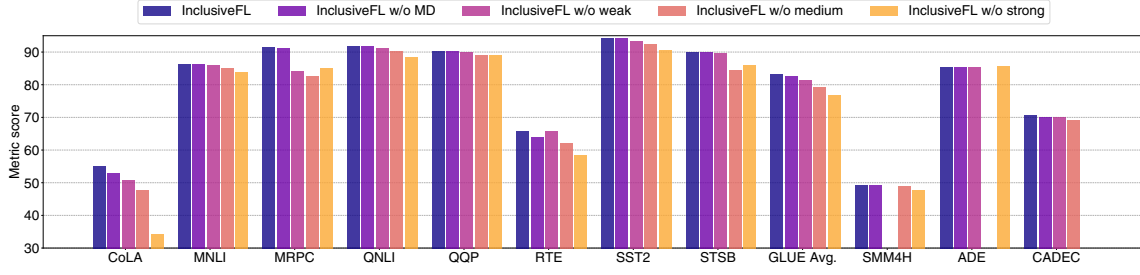
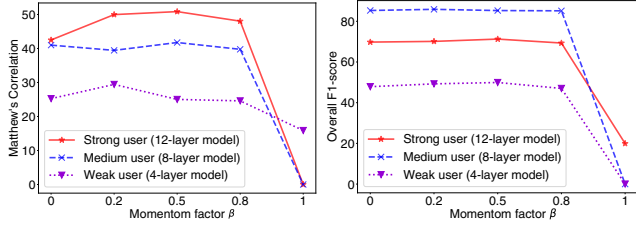


Figure 5: No one left behind: Ablation study of ignoring every single party for GLUE benchmark and medical NER datasets.



(a) CoLA with *InclusiveFL**. (b) Medical NER with *InclusiveFL*

Figure 6: Influence of momentum distillation β .

InclusiveFL-w/o medium for ADE and *InclusiveFL-w/o strong* for CADEC are not applicable because each client has different local token classification task and cannot be trained without the local dataset. Among the three device types, we notice that *InclusiveFL w/o strong* performs worse than *InclusiveFL-w/o medium* and *InclusiveFL-w/o weak*, which indicates that the contribution of powerful clients, medium clients and weak clients decrease progressively. powerful clients with more resources can learn knowledge with a larger model and contribute more. We can observe that the client-inclusive federated learning (*InclusiveFL* and *InclusiveFL-w/o MD*) perform better than excluding each type of device, which validates our motivation that every single device should be included and contribute to federated training.

4.3 Hyper-Parameter Analysis

At last, we analyze two important hyper-parameters for *InclusiveFL*.

Device type proportion. To investigate the effectiveness of *InclusiveFL* under different proportions of strong, medium, and weak devices, we present results with 1:2:7, 7:2:1 and a default setting of 1:1:1 on the CoLA dataset in Table 5. In general, increasing the proportion of strong devices results in a better performance of *ExclusiveFL* and *HeteroFL*. We can observe that the performance advantage of *InclusiveFL* is consistent across different device proportions, compared with naive baselines and *HeteroFL*. But we notice that when the powerful client group is the minority (e.g., 7:2:1), the convergence of the largest model is slow and aggregating over heterogeneous devices is not stable. Thus, we indicate one result in Table 5 with a superscript of * when only adopt the momentum distillation without sharing layer-wise parameters across heterogeneous models. And the performance is slightly better than the best baseline *ExclusiveFL* with a Matthew's correlation of 36.87.

Momentum factor β . Then we tune the momentum factor β of *InclusiveFL* with various choices of $\{0, 0.2, 0.5, 0.8, 1\}$. This parameter indicates the weight of gradient from larger model when applying the homophobic gradients to update each sub-global. And $\beta = 0$ equals the version of *InclusiveFL-w/o MD*. We conduct evaluation on each type of sub-global model for the two cases: 1) *InclusiveFL** that only applies the momentum distillation without heterogeneous parameter exchange on CoLA dataset in the IID cross-device setting. 2) *InclusiveFL* on Medical NER dataset in non-IID cross-silo setting. In Fig. 6, we indicate results that are superior than *ExclusiveFL*, *AllSmall* and *HeteroFL* with the circle dot. We notice that the optimal value of β is around 0.5 for powerful clients under both settings. The model on medium and weak devices also performs well when $\beta = 0.2$ or 0.5. The performance of a $\beta > 0.8$ may be ruined by injecting too much momentum from larger model to smaller model's gradients.

5 CONCLUSIONS

In this paper, we propose a client-inclusive framework *InclusiveFL* for federated training over heterogeneous devices and remove the ideal assumption in conventional FL that all clients have sufficient device capability. *InclusiveFL* enables us to train a large global model with contributions from all clients by assigning models of different sizes to clients with different computing capabilities. Thus, we avoid problems caused by ignoring weak clients or training with a small model. Then, we propose an effective method to share the knowledge among multiple local models with different sizes by a layer-wise heterogeneous aggregation. Besides, we propose a momentum knowledge distillation method for better transferring knowledge in big models of strong powerful clients to small models on weak clients. Extensive experiments are conducted on benchmark data and real-world medical corpora with comparisons of both naive baselines and the existing state-of-the-art work under IID and Non-IID settings. Results validate the accuracy improvement of *InclusiveFL* on the large global model as well as smaller models when inference is conducted on local devices.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant numbers 62072460, 62076245, 62172424 and Beijing Natural Science Foundation under Grant number 4212022.

REFERENCES

- [1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingelman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems* 1 (2019), 374–388.
- [2] Nader Bouacida, Jiahui Hou, Hui Zang, and Xin Liu. 2021. Adaptive federated dropout: Improving communication efficiency and generalization for federated learning. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPs)*. IEEE, 1–6.
- [3] Sebastian Caldas, Jakub Konečný, H Brendan McMahan, and Ameet Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210* (2018).
- [4] Daniel Cera, Mona Diabb, Eneko Agirrec, Inigo Lopez-Gazpio, Lucia Speciad, and Basque Country Donostia. 2017. SemEval-2017 Task 1 Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation. (2017).
- [5] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. 2019. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279* (2019).
- [6] Pierre Courtiol, Charles Maussion, Mathi Moari, Elodie Pronier, Samuel Pilcer, Meriem Sefta, and Pierre Manceron. 2019. Deep learning-based classification of mesothelioma improves prediction of patient outcome. *Nature Medicine* 25, 10 (2019), 1519–1526.
- [7] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net. <https://openreview.net/forum?id=TNkPBBYfkXg>
- [8] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305* (2020).
- [9] Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.
- [10] Aritra Dutta, El Houcine Bergou, Ahmed M Abdelmoniem, Chen-Yu Ho, Atal Narayan Sahu, Marco Canini, and Panos Kalnis. 2020. On the discrepancy between the theoretical analysis and practical implementations of compressed communication for distributed deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3817–3824.
- [11] Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2020. FedNER: Privacy-preserving Medical Named Entity Recognition with Federated Learning. (2020). *arXiv:2003.09288* <https://arxiv.org/abs/2003.09288>
- [12] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Julianne Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a Benchmark Corpus to Support the Automatic Extraction of Drug-Related Adverse Effects from Medical Case Reports. *JBI* 45 (Oct. 2012), 885–892. <https://doi.org/10.1016/j.jbi.2012.04.008>
- [13] Weituo Hao, Mostafa El-Khamy, Jungwon Lee, Jianyi Zhang, Kevin J. Liang, Changyou Chen, and Lawrence Carin. 2021. Towards Fair Federated Learning With Zero-Shot Data Augmentation. In *CVPR Workshops*. 3310–3319.
- [14] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).
- [15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).
- [16] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. *NeurIPS* 33 (2020), 9782–9793.
- [17] Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A Corpus of Adverse Drug Event Annotations. *Journal of Biomedical Informatics* 55 (June 2015), 73–81. <https://doi.org/10.1016/j.jbi.2015.03.010>
- [18] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Universal Language Model Fine-tuning for Text Classification* (2018), 278.
- [19] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
- [20] Daliang Li and Junpu Wang. 2019. Fedmd: Heterogeneous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* (2019).
- [21] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. 2020. Ensemble distillation for robust model fusion in federated learning. *NeurIPS* 33 (2020), 2351–2363.
- [22] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. 2020. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. *arXiv:1712.01887 [cs.CV]*
- [23] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692* (2019).
- [24] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the Value of Network Pruning. In *ICLR*.
- [25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
- [26] H Brendan McMahan et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1 (2021).
- [27] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*.
- [28] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.
- [29] Joseph Prusa, Taghi M. Khoshgoftaar, and Naeem Seliya. 2015. The Effect of Dataset Size on Training Tweet Sentiment Classifiers. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. 96–102. <https://doi.org/10.1109/ICMLA.2015.22>
- [30] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018).
- [31] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295* (2020).
- [32] Hassan Sajjad, Fahim Dalvi, Nadir Durani, and Preslav Nakov. 2020. On the effect of dropping layers of pre-trained transformer models. *arXiv preprint arXiv:2004.03844* (2020).
- [33] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *TNNLS* 31, 9 (2019), 3400–3413.
- [34] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.
- [35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* 15, 1 (jan 2014), 1929–1958.
- [36] Lichao Sun and Lingjuan Lyu. 2020. Federated model distillation with noise-free differential privacy. *arXiv preprint arXiv:2009.05537* (2020).
- [37] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S Nikolopoulos. 2016. Challenges and opportunities in edge computing. In *SmartCloud*. IEEE, 20–26.
- [38] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).
- [39] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. 2018. Atomo: Communication-efficient learning via atomic sparsification. *NeurIPS* 31 (2018).
- [40] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient sparsification for communication-efficient distributed optimization. *NeurIPS* 31 (2018).
- [41] Stefanie Wernat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastri, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickers, N Ahmad Aziz, et al. 2021. Swarm learning for decentralized and confidential clinical machine learning. *Nature* 594, 7862 (2021), 265–270.
- [42] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics* 7 (2019), 625–641.
- [43] Davy Weissenbacher, Abeed Sarker, Arjun Magge, Ashlynn Daughton, Karen O'Connor, Michael J. Paul, and Graciela Gonzalez-Hernandez. 2019. Overview of the Fourth Social Media Mining for Health (SMM4H) Shared Tasks at ACL 2019. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*. 21–30. <https://doi.org/10.18653/v1/W19-3203>
- [44] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. 1112–1122.
- [45] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. FedCTR: Federated Native Ad CTR Prediction with Cross Platform User Behavior Data. *ACM Transactions on Intelligent Systems and Technology (TIST)* (2022).
- [46] Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021. NewsBERT: Distilling Pre-trained Language Model for Intelligent News Application. *arXiv:2102.04887 [cs.CL]*
- [47] Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. 2021. Asynchronous federated learning on heterogeneous devices: A survey. *arXiv preprint arXiv:2109.04269* (2021).
- [48] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv:1812.02903* (2018).