No One Left Behind: Inclusive Federated Learning over Heterogeneous Devices

Ruixuan Liu¹, Fangzhao Wu², Chuhan Wu³, Yanlin Wang², Lingjuan Lyu⁴, Hong Chen¹, Xing Xie²
¹Renmin University of China, Beijing 100872, China
²Microsoft Research Asia, Beijing 100080, China
³Tsinghua University, Beijing 100084 China
⁴Sony AI, Tokyo 108-0075, Japan
{ruixuan.liu, chong}@ruc.edu.cn, {wufangzhao, wuchuhan15}@gmail.com
lingjuan.lv@sony.com,{yanlwang,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 multiple local models with different sizes. In this way, all the clients can participate in the model learning in FL, 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 the proposed method in learning accurate models from clients with heterogeneous devices under the FL framework.

KEYWORDS
Federated learning, Heterogeneous device, Knowledge distillation

Figure 1: InclusiveFL over heterogeneous devices.

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 [24]. Exchanging data from different sources to optimize machine learning models is attractive but challenging under the privacy regularization such as GDPR⁴. Based on the principles of focused collection and data minimization to avoid privacy issues, federated learning (FL) [23] 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 local models in parallel on local datasets. Then, the server aggregates local models to update the global model. Hence, conventional federated learning methods [23, 29] 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, the devices of clients are usually heterogeneous and they may have the significantly different computing power and memory size [15, 34, 44]. Although a large global model (e.g., BERT with 24-layer Transformer [17, 21]) 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 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

⁴https://gdpr-info.eu/
fairness issues [12, 26] because weak clients would be underrepresented in the final global model. Furthermore, the loss of data on weak clients causes an inferior accuracy [27], 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 [22].

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 [4, 19] utilize the knowledge distillation technique [14] 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 limited memory to store locally. A recent work HeteroFL [6] does not require a proxy dataset and proposes to train heterogeneous local models by averaging overlapped parameters of each sub-matrix across large and small models. However, this method may lead to serious mismatch problems between small and large models because the width pruning operation breaks the original model structure, which is not beneficial for retaining large models’ knowledge in smaller models. In addition, 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 [14]. Thus, how to learn models with different sizes and aligned feature spaces via effective knowledge transfer is a challenging problem in FL on 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 smaller models. It has been empirically investigated that the lower layers are most critical for retaining the pre-trained knowledge in downstream tasks [30], and the feature spaces among different layers have much consistency [7]. 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 [43]. 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 the learning of a global powerful large 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 [23] is a machine learning technique that trains an algorithm across multiple decentralized edge devices (cross-device) [13, 25, 45] or servers holding local data samples (cross-silo) [5, 38, 42], without exchanging them. Conventional federated learning protocol typically adopts FedAvg [23] to aggregate local parameters as follows:

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

where \( n = \sum_{k=1}^{K} n_k \) denotes the total number of local samples. For a better training convergence, FedAdam [29] 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 \( g_i^t \) after iterating over several rounds of local SGDs. Then, the server averages local gradients \( g_t \leftarrow \frac{1}{|S|} \sum_{i \in S} g_i^t \) and updates the global model as follows:

\[
\begin{align*}
    m_t &\leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t \\
    v_t &\leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\
    w_{t+1} &\leftarrow w_t - \eta \frac{m_t}{\sqrt{v_t} + \epsilon}
\end{align*}
\]

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 [4, 19] provide solutions for federated learning with heterogeneous local models based on knowledge distillation[14]. They assume that all clients share a public proxy dataset and propose to ensemble knowledge from clients by aggregating local predictions. Then the global prediction is distributed to all clients as augmented dataset for local training. Local models are optimized sequentially [19] or jointly Chang et al. [4] over the augmented dataset and the local private dataset. However, it is impractical for weak clients with limited hardware to store the public dataset.

Recently, HeteroFL [6] 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 \( W \in \mathbb{R}^{d_x \times k_x} \) in the large model, they reduce the parameters for a smaller model as the upper left sub-matrix \( W_c = W[: d_c \times k_x] \) with size \( d_c \times k_x \). 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 the 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 communication cost [19, 31]. Several works aim to mitigate the up-link communication bottlenecks from client to server [9, 18, 20, 36, 37]. Local model updates can be sparsified, quantized or randomly sub-sampled before uploading to the central server. Konečný et al. [18] 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. [2] extends the idea of dropout [33] and drops a fixed number of activation units at each full-connected layer before distributing a local model to each client. Reddi et al. [29] adopts an adaptive strategy and selects the dropped activation based on the activation score map. It should be noted that reducing server-to-client communication requires a full-model optimization and cannot include devices with insufficient memory. 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 [30], 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 client groups own small, medium or a large local models with $L_w$, $L_m$, $L_c$ layers, where $L_w < L_m < L_c$. 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 $U^a_t$, $U^b_t$ and $U^c_t$ 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 [23] and get the local gradient as $g^j_t$. Within each client group, we average the gradient as $g^j_t$, where $j \in \{a, b, c\}$. Then we update the sub-global model into $w^j_t$ with the averaged gradients and the momentum $m^j_{t-1}$ to merge local knowledge from clients with the same device type. The momentum $m^j_{t-1}$ is distilled from a larger model on client group $k$ and we explain details in Section 3.2.

With sub-models $w^a_{t-1}$, $w^b_{t-1}$, $w^c_{t-1}$ 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 noted that we decompose the model $w$ into a shared module $\theta$ and an isolated module $f$, as shown in Line 26. Specifically, we use the notation $\theta^a_{t-1}$ to indicate the $t^{th}$ 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 $\theta^c_{t-1}$. Combining with the within-group updated top layers $f_{t-1}$ for $\{a, b, c\}$ in Line 29, the server derives the updated model $\theta^a_{t}$.
sub-global model and distributes to the according clients for the next training rounds.

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 example of a text classification task in Fig. 2, the bottom $L_a - 1$ layers in $w^d$ 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 $f_{+1}$ with $f_{+1}^b$ and $f_{+1}^t$, the averaged model applied to $w^d$ 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 [10], medical data stored at different medical institutions may have different characteristics and annotation criterion, which leads to distinct top layers that cannot be shared.

![Figure 2: Overview of InclusiveFL.](image)

### 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 [14] 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 $\theta^t_{+1}$ as the average over the gradient for a stack of layers from $L_a$ to $L_b$ in the medium sub-global model. For the averaged gradient $\theta^t_{+1}$, we inject a distilled gradient momentum $m^b_{+1}$ to the gradient $g^b_{+1}$ of the isolated encoder layer $L_a$, which helps the $L_a$ layer in the small model imitate the block of layer $L_b$ to $L_a$ in the medium model. Similarly, we use the averaged gradient from the large model as the gradient momentum for the layer $L_b$ in the medium model. In general, the distillation transfer 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 $\beta$, 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 [6]. 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 [35] is a collection of multiple natural language understanding tasks, including MNLI [41] (inference), SST-2 [32] (sentimental analysis), MRPC [8] (paraphrase detection), CoLA [39] (linguistic acceptability), QNLI [28] (inference), QQP (question-answering), RTE 3 (inference), and STSB [3] (textual similarity). The generality makes the GLUE dataset a standard benchmark to evaluate NLU models.

Three medical datasets include SMM4H [40], CADEC [16] and ADE [11], which are widely used to evaluate NER tasks. The statistics of the medical datasets are shown in Table 1.

| Dataset  | # Sentences | Entity Types | # Entity |
|----------|-------------|--------------|---------|
| SMM4H    | 3,824       | ADE (1707)   | 1,707   |
| ADE      | 4,483       | ADE (5678), Drug (5076), Dosage (222) | 10,976  |
| CADEC    | 7,683       | ADE (5937), Drug(1796), Disease(282), Finding(425), Symptom(268) | 8,535   |

Evaluation Protocols. For the evaluation on cross-device [24] 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 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 [24] 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 seqeval framework\footnote{https://github.com/chakki-works/seqeval} 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.
- DropWeak: 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: 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 ratio.

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 [29] 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 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 performance. However, it is impractical for federated training participants with less resources to conduct the local optimization of large models.

For all other approaches that can be implemented across heterogeneous devices, our proposed approach InclusiveFL achieves...
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. Underlines indicate the performance is superior to baseline methods. Bold faces indicate the best method for training with heterogeneous devices.

| Methods       | Client-inclusive | Agg. type | COLA | MNLI | MRPC | QNLI | QQP | RTE | SST2 | STSB | Avg. |
|---------------|-----------------|-----------|------|------|------|------|-----|-----|------|------|------|
| AllLarge      | N/A             | Homo.     | 63.03| 86.48| 91.5 | 92.09| 91.49| 76.12| 94.43| 90.6 | 85.45|
| DropWeak      | No              | Homo.     | 37.77| 85.98| 88.81| 91.24| 89.47| 62.17| 94.06| 89.26| 79.85|
| AllSmall      | Yes             | Homo.     | 34.91| 78.83| 82.5 | 85.93| 79.37| 58.94| 90.14| 83.68| 74.29|
| HeteroFL      | Yes             | Hete.     | 8.15 | 31.83| 81.51| 62.7 | 73.79| 52.71| 84.98| 30.54| 53.28|
| InclusiveFL-w/o MD | Yes       | Hete.     | 52.69| 86.28| 91.2 | 91.59| **90.34**| 63.9 | 94.04| 89.91| 82.49|
| InclusiveFL   | Yes             | Hete.     | **54.8**| **86.36**| **91.42**| **91.76**| **90.32**| **65.85**| **94.17**| **89.94**| **83.08**|

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. Underlines and bold faces have the same meaning as in Table 2.

| Methods           | SMM4H | ADE | CADEC | Avg. |
|-------------------|-------|-----|-------|------|
|                   | ADE   | Drug | Drug  | Drug |
|                   |       | ADE  | ADE   | Symptom | Drug  | Disease | Finding |
| AllLarge          | 55.08 | 95.21| 80.91 | 18.97 | 71.23 | 43.37 | 90.43 | 33.55 | 29.85 | 57.62 |
| Local             | 46.77 | 95.12| 80.14 | 11.25 | 44.15 | 19.9 | 53.33 | 20.58 | 16.53 | 43.11 |
| AllSmall          | 49.56 | 92.76| 75.67 | 13.51 | 64.82 | 26.71 | 87.91 | 21.07 | 20.23 | 50.25 |
| HeteroFL          | 27.94 | 73.9 | 37.57 | 11.57 | 59.9 | 26.92 | 84.58 | 25.77 | 19.88 | 42.51 |
| InclusiveFL-w/o MD| 49.14 | 95.45| 79.63 | 12   | 69.98 | 34.42 | 90.16 | 38.31 | 20.41 | 54.39 |
| InclusiveFL       | **49.9**| **95.49**| **80.34**| **13.91**| **71.1**| **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**|
the best performance and its variant  InclusiveFL-w/o MD also beats baselines in most cases. 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 (DropWeak / Local). This observation also validates the effectiveness of our proposed momentum distillation.

We find that the most related work HeteroFL [6] performs worse than the naive baselines AllSmall and DropWeak on all GLUE datasets. And the averaged performance of HeteroFL is also inferior to AllSmall and DropWeak 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 [6]) 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 utilize 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 DropWeak 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 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 DropWeak 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, 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 DropWeak 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 strongest baseline DropWeak 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 homophbic gradients to update each sub-global. And β = 0 equals the 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 DropWeak, 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 β = 0.2 or 0.5. The performance of a β > 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.

### REFERENCES

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dmitriy Huba, Alex Ingerman, 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] Sebastian Caldas, Jakub Konečný, H Brendan McMahan, and Anmei Talwalkar. 2018. Expanding the reach of federated learning by reducing client resource requirements. arXiv preprint arXiv:1812.07210 (2018).

[3] Daniel Cera, Mona Diab, Eneko Agirre, Inigo Lopez-Garcio, Lucia Speciad, and Basque Country Donostia. 2017. SemEval-2017 Task 1 Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation. (2017).
