CDKT-FL: Cross-Device Knowledge Transfer using Proxy Dataset in Federated Learning

Huy Q. Le\textsuperscript{a,1}, Minh N. H. Nguyen\textsuperscript{c,1}, Shashi Raj Pandey\textsuperscript{d}, Chaoning Zhang\textsuperscript{b}, Choong Seon Hong\textsuperscript{a,*}

\textsuperscript{a}Department of Computer Science and Engineering, Kyung Hee University, Yongin-si, 17104, Republic of Korea
\textsuperscript{b}Department of Artificial Intelligence, Kyung Hee University, Yongin-si, 17104, Republic of Korea
\textsuperscript{c}Digital Science and Technology Institute, The University of Danang—Vietnam-Korea University of Information and Communication Technology, Da Nang, 550000, Vietnam
\textsuperscript{d}Department of Electronic Systems, Aalborg University, Aalborg, 9220, Denmark

Abstract

In a practical setting, how to enable robust Federated Learning (FL) systems, both in terms of generalization and personalization abilities, is one important research question. It is a challenging issue due to the consequences of non-i.i.d. properties of client’s data, often referred to as statistical heterogeneity, and small local data samples from the various data distributions. Therefore, to develop robust generalized global and personalized models, conventional FL methods need to redesign the knowledge aggregation from biased local models while considering huge divergence of learning parameters due to skewed client data. In this work, we demonstrate that the knowledge transfer mechanism achieves these objectives and develop a novel knowledge distillation-based approach to study the extent of knowledge transfer between the global model and local models. Henceforth, our method considers the suitability of transferring the outcome distribution and (or) the embedding vector of representation from trained models during cross-device knowledge transfer using a small proxy dataset in heterogeneous FL. In doing so, we alternatively perform

\textsuperscript{*}Corresponding author.

Email addresses: quanghuy69@khu.ac.kr (Huy Q. Le), nhnminh@vku.udn.vn (Minh N. H. Nguyen), srp@es.aau.dk (Shashi Raj Pandey), chaoningzhang1990@gmail.com (Chaoning Zhang), cshong@khu.ac.kr (Choong Seon Hong)

\textsuperscript{1}Equal contribution.
cross-device knowledge transfer following general formulations as 1) global knowledge transfer and 2) on-device knowledge transfer. Through simulations on three federated datasets, we show the proposed method achieves significant speedups and high personalized performance of local models. Furthermore, the proposed approach offers a more stable algorithm than other baselines during the training, with minimal communication data load when exchanging the trained model's outcomes and representation.

**Keywords:** Federated learning, Knowledge distillation, Representation learning

1. Introduction

Nowadays, myriad mobile applications incorporate on-device AI modules to utilize the user’s data and facilitate new experiences on their smartphones, such as AI-powered cameras, Extended Reality (XR) applications, and intelligent assistants (Samsung, 2021). However, data privacy concerns have been an *Achille’s heels* in the centralized machine learning (ML) paradigm. Thus, in recent years, there has been a sudden surge in developing federated learning (FL) frameworks that are able to limit data privacy leakage (Kairouz et al., 2019; McMahan et al., 2017; Lim et al., 2020; Li et al., 2021a; Nguyen et al., 2021, 2022; Thwal et al., 2024). In FL, a set of devices are connected with a central node (edge/cloud server) to collaboratively train a learning model without sharing their local data (Kairouz et al., 2019). In doing so, the FL schemes require sharing only the learning model parameters with the server; hence, the user data privacy can be kept intact. However, such a federated training scheme raises concerns about the heterogeneity of computing devices and distributed data samples' properties, leading to poor learning performance and fluctuation of the trained models across devices. To fill this research gap, recent FL algorithm designs are departing from heterogeneous FL settings towards personalized FL, which considers the performance of the global model and the personalized client models. In *heterogeneous* FL, the underlying data distribution is not identical across devices, i.e., non-i.i.d. Hence, the solution to the global empirical risk minimization problem, i.e., the trained global model, will not *personalize* well for each client. Meanwhile, the local training process in FL algorithms with their small private datasets often results in biased client models that produce high accuracy on the local data but with low generalization capabilities. Also, the client model parameters
and data features are diverse across devices. Such as in FedAvg (McMahan et al., 2017), the aggregation of model parameters based on the conventional coordinates-based averaging scheme from highly skewed data could slow down the learning process due to extensive divergence of model parameters to be averaged (Yu et al., 2020).

In this paper, we redesign knowledge transfer mechanisms in FL and examine the underlying fundamental correlation between the learning performances of FL algorithms. Unlike previous FL methods (McMahan et al., 2017; Li et al., 2020; Acar et al., 2021; Guzzo et al., 2023) that adopt the parameter aggregation scheme, our framework introduces the knowledge transfer mechanism on both sides to update the models without exchanging weight parameters of learning models between server and local devices. Instead of sending parameters, the models only need to exchange knowledge, such as the outcomes or representations. Specifically, we propose CDKT-FL, a novel cross-device knowledge transfer mechanism that leverages the model outcomes and (or) representation from clients using small proxy data to achieve robust generalized global and personalized models. To that end, we show that the proposed method achieves significant speedups and obtains better personalized performance of local models in the fixed users scenario and the subset of users selection scenario. Furthermore, we show that our approach offers better stability than other baselines and tackles privacy leakage issues with minimal communication data load when exchanging the trained model’s outcomes and (or) representation solely. Our main contributions are:

- In the typical FL algorithms such as FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), the global model is updated via model aggregation (i.e., parameters averaging), which is then broadcast back to clients and used as the initial model for local learning. However, such approaches ignore the fundamentals of interleaved properties of data across clients, thus resulting in poor generalization. To address this research gap, in this work, we propose cross-device knowledge transfer (CDKT) with two mechanisms: 1) Generalized Model Construction for transferring knowledge from client models to the global model, and 2) On-Device Learning for transferring the generalized knowledge of global model to client models.

- In doing so, we study the knowledge transfer mechanism using generic distance regularizers to align the knowledge between the models. Unlike knowledge distillation approaches, the transferable knowledge is the
outcomes (i.e., activation results $z$ of the final layer) and (or) embedding representation features (i.e., activation results $e$ of the intermediate layer) given the proxy samples. We assume this proxy data is of small size and available by clients to follow a practical scenario where the system can pre-collect a small amount of labeled data at the beginning.

- Our experiments reveal that (i) the outcome distribution and embedding features’ representation of the proxy data shared by all devices and servers are transferable and (ii) quickly improves the global model performance and personalized performance of the client models. Unlike many existing works, we evaluate the personalized performance of local models with specialization (i.e., performance on local testing data), and generalization capabilities (i.e., cross-device testing performance). These evaluation metrics provide more insights into the behaviors of the heterogeneous FL systems with different transferring methods and learning settings.

- The proposed CDKT-FL algorithm enables the flexibility of heterogeneous learning model design and knowledge transfer mechanism for future personalized FL applications. CDKT also improves stability and faster convergence while requiring fewer communication data with the outcome and representation knowledge than full complex model parameters compared to the FedAvg algorithm. These observations provide solid motivation to study knowledge transfer approaches and related metrics in FL. To the best of our knowledge, even though there are existing KD-based algorithms, our general proposed knowledge transfer approach and metrics advance the applicability of KD-based algorithms. Further, different from existing works, we leverage the true labels that can fix the wrong prediction from weak models in the early stage of the training process and improve the generalization of models.

To that end, we discuss some related works in Section 2. We present our proposed method and formulate generalized model construction, on-device learning and propose a CDKT-FL algorithm in Section 3. We validate the effectiveness of our proposed algorithm for both specialization and generalization performance of the client and global models compared to several state-of-the-art FL algorithms in Section 4. Finally, Section 5 concludes the paper.
2. Related Works

**Structural alignment in heterogeneous FL:** One of the promising research directions to resolve unmatched parameters of learning models in FL is using the structural alignment methods (Mostafa, 2019; Wang et al., 2020; Yu et al., 2020; Zhang et al., 2021b; Sun et al., 2023; Xiong et al., 2023). The representation matching in (Mostafa, 2019) and “permutation invariance” in FedMA (Wang et al., 2020) are proposed to evaluate parameter similarity across local models and reorganize the permutation accordingly via matching schemes; thus, obtaining parameters that could be averaged together in the model aggregation. In (Yu et al., 2020), the authors proposed structure-information alignment based on the feature-oriented interpretation and regulation method to ensure explicit feature information allocation in different neural network structures. Based on the feature interpretation, the proposed regulation method adjusts the local model architecture according to their data and task distribution at the very early training stage and continuously maintains structure and information alignment. To address the domain generalization problem in FL, the authors in (Zhang et al., 2021b) design the federated adversarial learning approach to measure and align the distributions among different source domains by matching each distribution to a reference distribution in a distributed manner. Similarly, Sun et al. (2023) tackle the domain generalization problem using feature distribution matching, which employs a federated voting mechanism to generate pseudo-labels for the target domain by aggregating the consensus of clients to fine-tune a global model. Recently, Xiong et al. (2023) proposed the FedDM method, which utilizes iterative distribution matching to learn a surrogate function. Instead of sending local model updates, FedDM sends synthesized data to the server, resulting in improved communication efficiency and effectiveness. In this manner, structural alignment methods showed their efficiency in improving the FL performance and applicability in various heterogeneous settings.

**Common Representation Learning:** This is a popular method to handle the uneven distributions and representations of different modalities or tasks (Bengio et al., 2013; Huang et al., 2017; Peng and Qi, 2019; Xu et al., 2020). In this approach, the training of shared features across tasks/modalities constructs a common representation; thereby, cross-modal common representational learning allows knowledge transfer across different modalities (Bengio et al., 2013). In doing so, the heterogeneous data from different modalities resulting in diverse representations are projected into a common
space. Then, the cross-modal knowledge transfer can be performed by reducing the discrepancy between representations of pairwise cross-modal data from high-level layers (Huang et al., 2017). Furthermore, it can preserve the relevant semantics and allow heterogeneous data from different modalities to be correlated more easily (Peng and Qi, 2019; Xu et al., 2020). Recently, a similar design, the FedRep algorithm (Collins et al., 2021), was applied in FL that alternatively trains the shared low-dimensional representation, which is aggregated across clients by averaging and private local heads to improve the learning performance of client models. In contrast to FedRep, the approach presented in (Oh et al., 2022) introduces FedBABU, an algorithm that focuses on learning and aggregating the body of the model while keeping the head randomly initialized during local updates.

The convergence of the alignment methods and knowledge distillation: Even though the alignment approaches are widely studied in the representation learning literature, the knowledge distillation (KD) approach has emerged as one of the promising alignment approaches. KD approach demonstrated superior performance via knowledge transfer using outcome distribution (i.e., soft targets) from the teacher to student models (Hinton et al., 2015). The alignment of outcome distribution using Kullback–Leibler (KL) divergence helps the student model imitate and learn from a teacher model or ensemble of models. KD was first introduced in (Buciluă et al., 2006; Ba and Caruana, 2014) and then developed by (Hinton et al., 2015) to compress the smaller learning model from a large model. In KD, the soft targets \( z_t \) and \( z_s \) are the softmax outputs of the student and teacher model, respectively. The loss function of the student model between the prediction \( z_s \) and ground-truth label \( y_s \) is the cross-entropy loss denoted as \( \ell_{CE} \). In addition, the distillation loss using the KL divergence is defined as

\[
\ell_{KD} = \tau^2 KL(z_t, z_s),
\]

where \( \tau \) is the temperature parameter that softens the output values. The higher the value of the temperature, the softer the outputs of learning models. Therefore, the loss function is used to train the student model is defined as

\[
\ell_{\text{student}} = \ell_{CE} + \alpha \ell_{KD},
\]

where \( \alpha \) is the hyper-parameter that controls the penalty KD loss.
In (Zhang et al., 2018), the authors introduced a deep mutual learning strategy with an ensemble of student models to collaboratively learn and teach each other throughout the training process. Furthermore, KD was recently adopted to transfer the knowledge between the global and local models and exhibited promising results in FL. The alignment of outcome distribution with KL divergence in KD helps the global model and local models to transfer their learning knowledge and further improve the performance of the learning models. To overcome communication cost and heterogeneity issues, the authors in (Jeong et al., 2018) applied KD with the average logits from clients using their private datasets. Balancing the private datasets with data augmentation can further improve the learning performance. The works (Li and Wang, 2019; Bistritz et al., 2020; Le et al., 2021) leveraged a large public dataset in the local knowledge distillation using the aggregation/consensus of soft targets from clients as the teacher knowledge. In (Lin et al., 2020), the authors proposed ensemble distillation for global model fusion, enabling different sizes, numerical precision, or structure of learning models. It is possible to apply KD in the global and local model updates, as shown in (He et al., 2020), but the server requires sequentially distilling the knowledge from each client and vice versa. However, the authors do not consider the proxy dataset as well as the knowledge aggregation. Recently, the authors in (Zhang et al., 2021a) introduced the personalized knowledge transfer mechanism for personalized FL utilizing proxy data to update the local soft predictions for each local client. In addition to utilizing proxy data for knowledge aggregation from local models, (Zhu et al., 2021; Zhang et al., 2022) employs a lightweight generator to combine the knowledge of client models in a data-free manner. Moreover, another innovative approach known as split learning has garnered attention in federated learning research (Vepakomma et al., 2018; Singh et al., 2019; Thapa et al., 2022). Unlike traditional FL methods, split learning employs a strategy where knowledge is shared using activations and gradients, ensuring privacy preservation in collaborative learning. Split learning primarily involves the division of the neural network model into two segments: device models for each device and a central server model that receives activations from the devices as input. The process includes transmitting activations, referred to as smashed data, from the split layer, known as the cut layer, of the client-side network to the server and receiving the gradients of the smashed data resulting from server-side operations. Unlike split learning, CDKT has the global learning model having generalized capabilities and playing a role in knowledge aggregation from client models.
Figure 1: An overview of Cross-Device Knowledge Transfer (CDKT) scheme in FL, integrating two mechanisms: 1) Generalized Model Construction for transferring knowledge from client models to the global model, and 2) On-Device Learning for transferring the generalized knowledge of global model to client models.

while preserving the good personalized performance of local models leveraging proxy data.

Inspired by the discussion on alignment methods and knowledge distillation approaches, it is fundamental to find the answer to the following research question: “what transferable knowledge and regularizers should be used in knowledge transfer between the global model and local models in FL?” Answering this question opens opportunities to provide more efficient knowledge transfer mechanisms in FL.

3. Cross-Device Knowledge Transfer in Federated Learning

In a general scheme of FL algorithms (Reddi et al., 2020; Li et al., 2021b), the global model is updated via a gradient-based optimizer or model parameter aggregation in the server, while the local/personalized model in user device $n$ is updated using their private dataset $D_n$. We first define the neural network with the prediction outcomes (i.e., $z = h \circ w(x)$), where $h$ is the projection head, which is a small neural network (i.e., the last fully connected layers; a small classifier). We denote the embedding features of the representation sub-net parameters $w$ for the image $x$ as $e = w(x)$. Hence, the learning model parameters $w, h$ can be trained and used to extract the embedding features (i.e., $e_n$) and (or) classes’ prediction distribution (i.e., $z_n$) of input images at device $n$. In this work, instead of using model parameters from clients
in every round to aggregate into the global model, cross-device knowledge
transfer is being used to transfer the collective knowledge from all clients
to the server and construct a generalized model. We illustrate the overview
of our proposed CDKT-FL scheme in Fig. 1. Note that the global model
can be updated using an additional regularizer to align its knowledge with
the collective knowledge of clients. In doing so, it is necessary to have a
fixed small proxy data $D_r$ such that all models can access and provide their
learning knowledge as the outcomes (i.e., $z$) and (or) embedding features (i.e.,
e). In addition, the knowledge transfer also helps the global model transfer
its generalization capability to all client models. This way, firstly, we can
reduce communication data load and also further enhance users’ data privacy,
which is often compromised during full parameters exchange against reverse
engineering tricks. Secondly, since knowledge transfer is sufficient to transfer
learning capabilities, the learning model architectures can differ between the
server model (with more layers) and client models (with fewer layers).

The proposed CDKT-FL algorithm (Alg. 1) enables the knowledge transfer
between clients and global models in the FL setting. At each global round
t, the server broadcasts the knowledge from the updated global model to
all clients. Then, each client performs $K$ local epochs with its private and
proxy dataset based on the gradient of the loss function in the on-device
learning problem (8), with the local learning rate $\eta$. Note that we use mini-
batch of private data in the cross-entropy loss, whereas proxy data is in the
on-device CDKT regularizer. Then, the clients send their latest knowledge
(i.e., $z_n$ and (or) $e_n$) to the server. At the global level, the server builds the
generalized model by solving the global learning problem (7). For this, we
use a gradient-based update for $R$ global epochs with the global learning rate $\gamma$. In particular, to perform CDKT-FL, we design the knowledge transfer in
FL with two general problems: generalized model construction and on-device
learning in the following subsections.

3.1. Generalized Model Construction

In the generalized model construction problem, the main target is utilizing
the proxy data (i.e., $D_r$) and collective knowledge from all devices to build
the global model. If the transferable knowledge is only the outcome of
client models, then the problem is in the form of knowledge distillation (KD)
(Hinton et al., 2015). Similar to the design of (Li and Wang, 2019), we
consider the averaged results of the outcomes as the collective knowledge
that results in the equivalent performance of the learning problem. This
Algorithm 1 CDKT-FL Algorithm

1: **Input:** Number of global rounds $T$, number of local epochs $K$ and number of global epochs $R$ for global model update
2: **for** $t = 0, \ldots, T - 1$ **do**
3: **On-Device Learning:** Each device $n$ receives the generalized knowledge $g_s$ from the server and updates local model with $D_n, D_r$;
4: **for** $k = 0, \ldots, K - 1$ **do**
5: We loop for each batch of private and proxy data (i.e., $S_n$ and $S_r$):
6: \[ w_t^n = w_t^n - \eta \nabla c(S_n, S_r); \] (3)
7: **end for**
8: Device $n$ extracts and sends the local knowledge $g_n$ given the proxy dataset to the server;
9: **Generalized Model Construction:** Server updates the global model with the received local knowledge from the selected devices:
10: **for** $r = 0, \ldots, R - 1$ **do**
11: We loop through batches of proxy data (i.e., $S_r$):
12: \[ w_t^g = w_t^g - \gamma \nabla s(S_r); \] (4)
13: **end for**

approach, however, requires less computation than the sum of individual KL loss between the student model (global model) and each teacher model (client model) in ensemble KD. In the following, we review the KD method and discuss the proposed cross-device knowledge transfer method.

**KD Method:**
\[ \ell_s(D_r) = \ell_{CE}(z_s|D_r) + \beta \ell_{KD}(z_s, \sum_{n \in N} \frac{1}{N} z_n | D_r), \] (5)

where $\beta$ is the parameter to control the trade-off between proxy data loss and the KD regularizer; $z$ are the soft targets of the learning models.

**Cross-device knowledge transfer Method (CDKT):** We formulate the generalized model construction problem with the global CDKT regularizer using a generic function $d$. Thereby, the proposed formulation measures the
discrepancy between the knowledge of $n$ client models $(g_1, \ldots, g_n)$ and the knowledge of server model $g_s$ as

$$
\ell_s(D_r) = \ell_{CE}(z_s|D_r) + \beta d(g_s, g_1, \ldots, g_n|D_r).
$$

(6)

The collective knowledge from clients are constructed as the averaged results of the outcomes $z$ and (or) embedding features $e$. To improve the limitation in transferred knowledge from clients, we adjust the collective outcomes $z$ of clients by averaging it with the true labels $y_r$ when using their final outcomes.

$$
\ell_s(D_r) = \ell_{CE}(z_s|D_r) + \beta d\left(e_s, \sum_{n \in N} \frac{1}{N} e_n|D_r\right) + \beta d\left(z_s, \lambda y_r + (1 - \lambda) \sum_{n \in N} \frac{1}{N} z_n|D_r\right),
$$

(7)

where $\lambda \in [0, 1]$, and the parameter $\beta$ allows the server to control the trade-off between proxy data loss and the global CDKT regularizer. Here, we note that various distance functions $d$ can be used for knowledge transfer, such as Norm, KL divergence, and Jensen-Shannon (JS) divergence (Kullback, 1997). A generic distance function $d$ in CDKT regularizer can be adopted with the three following distance metrics.

**Norm2 distance:**

$$
d_{norm}(g_n, g_s|D_r) = \sum_{x \in D_r} \|g_n(x) - g_s(x)\|_2,
$$

where $g_n(x)$ is the prediction outcome (or embedding features) of client $n$ and $g_s(x)$ is the knowledge from the global model given data sample $x$.

**Kullback–Leibler (KL) divergence:**

$$
d_{KL}(g_n, g_s|D_r) = \sum_{x \in D_r} \sum_{c} P(g_{s,c}|x) \frac{P(g_{s,c}|x)}{P(g_{n,c}|x)},
$$

where $P(g_{n,c}|x)$ is the prediction outcome (or embedding features) of client $n$ and $P(g_{s,c}|x)$ is the knowledge from the global model of class $c$ given data sample $x$.

**Jashon-Shannon (JS) divergence:**

$$
d_{JS}(g_n, g_s|D_r) = \frac{1}{2} d_{KL}(g_n, g_m|D_r) + \frac{1}{2} d_{KL}(g_s, g_m|D_r)
$$

11
where \( g_m = \frac{1}{2}(g_n + g_s) \).

In this work, we explored a range of distance functions as regularizers to optimize the performance of our collaborative learning framework. First, we aim to identify the most effective regularizers for achieving stable performance in global and local contexts. By experimenting with different regularizers, we aimed to fine-tune the model’s behavior to suit the specific demands of each scenario, ensuring optimal performance on a global scale while also accommodating the unique characteristics of local clients. Second, it is essential to note that our collaborative learning framework is flexible and adaptable. Note that the knowledge transfer that is reflected in the regularizer of the global and local learning objectives plays a vital role in the CKDT algorithm to exchange the learning capabilities between models.

3.2. On-device Learning

Using a similar design, the on-device learning problem helps the client models to improve their generalization capabilities by imitating the generalized knowledge from the global model. In this approach, the client \( n \) utilizes the private dataset \( D_n \) and also the generalized knowledge obtained from the global model using the proxy dataset \( D_r \). In particular, we formulate the on-device loss function for each device \( n \) as follows:

\[
\ell_n^o(D_n, D_r) = \ell_{CE}^o(z_n|D_n) + \alpha d(g_n, g_s|D_r) \\
= \ell_{CE}^o(z_n|D_n) + \alpha d(\epsilon_n, \epsilon_s|D_r) + \alpha d(z_n, \lambda y_s + (1 - \lambda) z_s|D_r) \quad (8)
\]

where \( \alpha \) is the parameter to control the trade-off between the local learning objective’s biased knowledge and generalization capabilities. Further, the on-device CDKT regularizer helps to reduce the bias and over-fitting issues of local models when training on the small private dataset.

4. Experiments

4.1. Setting

In this section, we validate the efficacy of the CDKT-FL algorithm with the Fashion-MNIST (Xiao et al., 2017), CIFAR-10, and CIFAR-100 (Krizhevsky et al., 2009) datasets for handwritten digits, fashion images, and object recognition, respectively. We conduct the experiments with 10 selected clients in each round, where each client has median numbers of data samples at 70.5, 966.5, and 1163.5 with Fashion-MNIST, CIFAR-10, and CIFAR-100.
respectively. To simulate the properties of non-i.i.d data distribution, we set each client’s data from random 20 classes in 100 classes for CIFAR-100 dataset and only two classes in total 10 classes with the other datasets and use 20% of private data samples for evaluating the testing performance. Thereby, the private datasets are unbalanced and have a small number of training samples. We provide the data distributions among clients of different datasets in the Appendix. Further, the small proxy datasets have all classes data with, respectively, 330, 4200, and 4200 samples for Fashion-MNIST and CIFAR-10 datasets. They are just 0.5% and 7% of the overall number of samples and practically accessible by service providers. The global CNN model in the server consists of two convolution layers in Fashion-MNIST dataset, whereas three convolution layers are used in the CIFAR-10 and CIFAR-100 dataset, followed by the pooling and fully connected layers (described in Table A.3 and Table A.4 in Appendix). The client CNN models have a similar design or one CNN layer fewer than the global model one in the heterogeneous model setting. The learning parameters, such as learning rates and trade-off parameters, are tuned to achieve good results for different settings of algorithms: the number of local epochs $K = 2$, global epochs $R = 2$, and the batch size is 20.

**Evaluation metrics:** To the best of our knowledge, FL approaches focus more on the learning performance of the global model, while personalized FL algorithms evaluate the learning performance of the local models. In this work, to better capture the learning behavior of FL algorithms, we conduct evaluations with the specialization (C-Spec) and generalization (C-Gen) performance of client models on average, defined by the accuracy and F1 score in their local testing data only, and the collective testing data from all clients, respectively. To summarize the performance of local models, we calculate the average of C-Gen and C-Spec performance as the C-Per metric in terms of accuracy and F1 score. Using only C-Spec or C-Gen does not fully capture the performance of personalized models. However, considering both metrics allows us to have a complete view of the specialization and the generalization capabilities of personalized models. Therefore, the introduced averaging metric C-Per provides a balance between both capabilities of the personalized models. Meanwhile, the Global metric is the accuracy of the global model for the collective testing data from all clients.

**Implementation:** We develop our CDKT-FL algorithm on Pytorch library (Paszke et al., 2019), which is a robust open-source machine learning framework developed by Facebook. All the experiments are deployed on our server with one NVIDIA GeForce GTX-1080 Ti GPU using CUDA version 11.2 and Intel
Table 1: The comparison of average accuracy across three datasets over a range of 90 to 100 rounds.

|                  | Fashion-MNIST | CIFAR-10 | CIFAR-100 |
|------------------|----------------|----------|-----------|
|                  | Global C-Per   | Global C-Pers | Global C-Per |
| Fixed Users       |                |           |            |
| No Transfer       | 83.72          | 59.09     | 55.92      | 17.27 | 18.76 |
| FedAvg            | 82.56          | 76.41     | 59.85      | 52.95 | 15.53 | 17.13 |
| Scaffold          | 67.67          | 59.59     | 48.41      | 47.83 | 13.48 | 13.20 |
| MOON              | 83.28          | 75.83     | 60.83      | 53.09 | 17.37 | 17.67 |
| FedDyn            | 83.26          | 85.08     | 58.00      | 59.24 | 14.03 | 20.23 |
| CDKT (Rep)        | 84.88          | 86.06     | 58.44      | 59.25 | 17.75 | 21.81 |
| CDKT (Full)       | 85.81          | 81.66     | 58.91      | 55.83 | 17.84 | 21.93 |
| CDKT (RepFull)    | 86.51          | 84.08     | 59.09      | 57.74 | 18.49 | 22.22 |
| Subset of Users   |                |           |            |
| No Transfer       | 78.19          | 58.98     | 52.17      | 46.75 | 16.47 | 8.13  |
| FedAvg            | 69.44          | 70.62     | 48.83      | 47.20 | 12.25 | 12.57 |
| Scaffold          | 68.00          | 67.88     | 40.76      | 43.06 | 12.29 | 12.12 |
| MOON              | 70.05          | 70.70     | 53.05      | 52.15 | 16.79 | 16.69 |
| FedDyn            | 76.03          | 76.11     | 55.71      | 51.91 | 16.37 | 16.49 |
| CDKT (Rep)        | 78.50          | 80.90     | 55.67      | 53.92 | 16.72 | 15.74 |
| CDKT (Full)       | 79.22          | 79.02     | 56.60      | 54.35 | 17.33 | 15.76 |
| CDKT (RepFull)    | 79.94          | 81.39     | 56.57      | 54.03 | 17.60 | 17.74 |

Core i7-7700K 4.20GHz CPU with sufficient memory for model training.

**Baselines and CDKT settings:** For comparison, we evaluate different settings of the CDKT-FL algorithm and compared them with serveral state-of-the-art approaches including (1) FedAvg (McMahan et al., 2017), (2) SCAFFOLD (Karimireddy et al., 2020), (3) MOON (Li et al., 2021a) and (4) FedDyn (Acar et al., 2021). In **No Transfer** setting, the clients use the private DS, and the server uses the proxy DS. Accordingly, the setting **Rep, Full, and RepFull** stand for using only the embedding features of representation, outcomes of full models, and both in the global knowledge transfer, respectively. Furthermore, different distance functions were also validated, such as KL, JS, and Norm-2 (N). The setting KL-N indicates the use of KL divergence in the global CDKT regularizer, while the on-device CDKT regularizer uses Norm-2.
Figure 2: Comparative Analysis of CDKT-FL under various settings against other baselines using the Fashion-MNIST dataset in Fixed Users scenario.

Figure 3: Comparative Analysis of CDKT-FL under various settings against other baselines using the Fashion-MNIST dataset in Subset of Users scenario.
Table 2: The comparison of average F1 score (weighted) across three datasets over a range of 90 to 100 rounds.

|                  | Fashion-MNIST |         | CIFAR-10 |         | CIFAR-100 |         |
|------------------|---------------|---------|----------|---------|-----------|---------|
|                  | Global | C-Per | Global | C-Per | Global | C-Per |
| No Transfer      | 86.65 | 58.82 | 64.17 | 48.24 | 20.68 | 14.44 |
| FedAvg           | 81.75 | 75.53 | 66.46 | 51.85 | 18.19 | 12.79 |
| Scaffold         | 77.73 | 58.92 | 57.52 | 46.14 | 15.40 | 10.19 |
| MOON             | 85.53 | 79.96 | 69.03 | 56.79 | 20.81 | 12.97 |
| FedDyn           | 85.11 | 85.97 | 65.93 | 62.34 | 19.72 | 21.54 |
| CDKT (Rep)       | 87.04 | 89.85 | 67.15 | 62.40 | 23.60 | 21.67 |
| CDKT (Full)      | 86.37 | 84.52 | 67.21 | 58.32 | 23.37 | 19.65 |
| CDKT (RepFull)   | 86.69 | 82.82 | 68.15 | 60.16 | 24.68 | 21.30 |

|                  | Fashion-MNIST |         | CIFAR-10 |         | CIFAR-100 |         |
|------------------|---------------|---------|----------|---------|-----------|---------|
|                  | Global | C-Per | Global | C-Per | Global | C-Per |
| No Transfer      | 85.65 | 57.92 | 64.84 | 35.97 | 21.57 | 4.98 |
| FedAvg           | 73.91 | 75.65 | 61.54 | 58.48 | 12.50 | 12.33 |
| Scaffold         | 75.51 | 83.19 | 51.34 | 50.37 | 14.04 | 13.59 |
| MOON             | 74.53 | 76.90 | 65.00 | 62.72 | 18.85 | 13.84 |
| FedDyn           | 81.33 | 81.55 | 64.32 | 60.64 | 18.54 | 14.23 |
| CDKT (Rep)       | 86.41 | 85.96 | 65.69 | 62.41 | 22.61 | 14.45 |
| CDKT (Full)      | 86.39 | 83.35 | 67.07 | 60.68 | 22.86 | 12.94 |
| CDKT (RepFull)   | 86.46 | 85.90 | 66.63 | 63.01 | 22.67 | 13.10 |

### 4.2. Experimental Results

In summary, we obtained experimental results on three datasets for different settings in the CDKT-FL algorithm and baselines with the median testing accuracy and F1 score from round 90 to 100, as in Table 1 and Table 2. There are two principal scenarios: 1) the Fixed Users scenario, where all 10 users participate in every communication round, and (2) the Subset of Users scenario, in which 10 randomly selected users from a total of 50 users in each communication round. Compared to the other baselines, in terms of accuracy metric, CDKT-FL outperforms 10% and 5% with Fashion-MNIST, 6% and 2% with CIFAR-10, 2% and 1% with CIFAR-100 for scenario (1) and scenario (2) respectively, according to C-Per performance. In addition to the main target to improve personalized performance, in both metrics, CDKT-FL also significantly improves the performance of the global model in most of the settings as shown in Table 1 and Table 2. According to our detailed figures, in scenario...
Figure 4: Effects of trade-off parameters in CDKT-FL with Fashion-MNIST dataset in Fixed Users scenario.

(1), CDKT-FL obtains a higher C-Gen performance, a comparable C-Spec performance of client models, and a comparable or higher Global performance in all three datasets. In scenario (2), our proposed algorithm achieves higher C-Spec, Global performance and comparable C-Gen performance. We provide the detailed results of C-Spec and C-Gen in the Appendix. In principle, CDKT helps the client models improve C-Gen and slightly improve the Global performance than that of no transfer setting. In addition, as the final performance metric, the CDKT-FL algorithm exhibits better stability and faster convergence while improving the C-Per. It preserves high Global performance in most settings. These attributes also enable the client or global models to be usable early after 50 global rounds. As expected, in the subset of users selection scenario, other baseline show an extensive fluctuation due to irregularity in learning parameters across the client models stemming from statistical heterogeneity.

In Fig. 2 and Fig. 3, we evaluate different settings knowledge transfer schemes of CDKT-FL to compare with other baselines in both of the scenarios on Fashion-MNIST dataset. In scenario (1), as shown in Fig. 2, experimental
results demonstrate that our proposed schemes outperform other baselines in terms of speedups C-Gen and Global performance. The Rep+KL-N setting achieves the highest C-Gen performance with 75.37% but slightly lower C-Spec than that of MOON. In scenario (2), Fig. 3 shows that the CDKT-FL algorithms with three settings significantly improve C-Spec, Global and more stable than other approaches. We also provide the figures for CIFAR-10 and CIFAR-100 in the Appendix.

We next evaluate the effects of the trade-off parameters $\alpha$ and $\beta$ using to control the compensation between learning loss and the CDKT regularizer as shown in Fig. 4 and Fig. 5 in both scenarios, respectively. As a result, when increasing $\alpha$, we can boost C-Gen performance; however, that slightly degrades the C-Spec performance. Such learning behavior shows the interesting compensation effect between C-Gen and C-Spec performances, such as till some threshold the specialized capability may decrease as increasing the generalized capability of client models. On the other hand, $\beta$ does not show significant effects on the performance, however, when $\beta$ is too small, it makes the global performance is fluctuated due to less contribution from CDKT
Figure 6: Convergence analysis of CDKT-FL under various settings against other baselines using the Fashion-MNIST dataset.
Figure 7: Performance analysis of CDKT-FL with different distance metrics on the Fashion-MNIST dataset.

Figure 8: Performance analysis of CDKT-FL with heterogeneous models on the Fashion-MNIST dataset.
regularizers and transferred knowledge. Finally, we evaluate the effectiveness of the CDKT-FL algorithm using heterogeneous local models, i.e., using one CNN layer less than global model. The CDKT-FL algorithm with smaller client models achieves comparable performance in Fashion-MNIST dataset but slight degradation in the other datasets.

Additionally, In Fig. 6, we plot the box plot for different settings of CDKT-FL to compare with other baselines in terms of stability in the last 20 rounds in both scenarios. As can be seen, our algorithm obtains a very stable convergence with better performance while other methods depict the extensive fluctuation, especially in the subset of users scenario. For Fashion-MNIST, we also evaluate the effect of different distance metrics in CDKT-regularizers in both scenarios in Fig. 7. We observe that the setting KL-N obtains stable and higher performance among other metrics. In the KL-N setting, KL divergence is used in the global CDKT regularizer, while the on-device CDKT regularizer uses Norm. Finally, we observe that our method obtains the comparable performance when using heterogeneous models in Fig. 8 with just a slight degradation in fixed users scenario.

5. Conclusion

This work bridges the gap between existing alignment methods and knowledge distillation to develop a general form of cross-device knowledge transfer in FL. Through extensive experiments on numerous scenarios, the performance of client and global models are validated for heterogeneous FL. The detailed results following extensive simulations on three datasets have shown the advantages of the proposed algorithms. Compared to the other methods, we demonstrate that the CDKT-FL algorithm is more stable and can speed up the global model performance and higher personalized performance of client models. In addition to the learning performance, the proposed algorithm enables (i) the flexibility of heterogeneous model architectures, (ii) reduces communication data with the outcome, and (or) representation knowledge instead of full model parameters compared to the conventional FL algorithms, and (iii) tackles privacy leakage issues (via reverse engineering techniques) of current FL algorithms. As a trade-off, the proposed algorithm requires an additional small proxy dataset for transferring knowledge. Finally, we advocate that the knowledge transfer method works as a hybrid design with recent advanced techniques to improve learning performance.
References

Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V., 2021. Federated learning based on dynamic regularization. arXiv preprint arXiv:2111.04263.

Ba, J., Caruana, R., 2014. Do deep nets really need to be deep?, in: Advances in Neural Information Processing Systems.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1798–1828. doi:10.1109/TPAMI.2013.50.

Bistritz, I., Mann, A., Bambos, N., 2020. Distributed distillation for on-device learning, in: Advances in Neural Information Processing Systems, pp. 22593–22604.

Bucilă, C., Caruana, R., Niculescu-Mizil, A., 2006. Model compression, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 535–541.

Collins, L., Hassani, H., Mokhtari, A., Shakkottai, S., 2021. Exploiting shared representations for personalized federated learning, in: Proceedings of the 38th International Conference on Machine Learning, pp. 2089–2099.

Guzzo, A., Fortino, G., Greco, G., Maggiolini, M., 2023. Data and model aggregation for radiomics applications: Emerging trend and open challenges. Information Fusion, 101923.

He, C., Annavaram, M., Avestimehr, S., 2020. Group knowledge transfer: Federated learning of large cnns at the edge, in: Advances in Neural Information Processing Systems.

Hinton, G., Vinyals, O., Dean, J., 2015. Distilling the knowledge in a neural network, in: NIPS Deep Learning and Representation Learning Workshop.

Huang, X., Peng, Y., Yuan, M., 2017. Cross-modal common representation learning by hybrid transfer network, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 1893–1900.
Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.L., 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv:1811.11479.

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

Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T., 2020. Scaffold: Stochastic controlled averaging for federated learning, in: International conference on machine learning, PMLR. pp. 5132–5143.

Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images.

Kullback, S., 1997. Information theory and statistics. Courier Corporation.

Le, H.Q., Shin, J.H., Nguyen, M.N., Hong, C.S., 2021. Distilling knowledge in federated learning, in: 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 196–201. doi:10.23919/APNOMS52696.2021.9562670.

Li, D., Wang, J., 2019. Fedmd: Heterogenous federated learning via model distillation, in: The 2nd International Workshop on Federated Learning for User Privacy and Data Confidentiality.

Li, Q., He, B., Song, D., 2021a. Model-contrastive federated learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10713–10722.

Li, T., Hu, S., Beirami, A., Smith, V., 2021b. Ditto: Fair and robust federated learning through personalization, in: International Conference on Machine Learning, PMLR. pp. 6357–6368.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V., 2020. Federated optimization in heterogeneous networks, in: Proceedings of Machine Learning and Systems 2020, pp. 429–450.

Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., Miao, C., 2020. Federated learning in mobile edge networks: A comprehensive survey. IEEE Communications Surveys & Tutorials 22, 2031–2063.
Lin, T., Kong, L., Stich, S.U., Jaggi, M., 2020. Ensemble distillation for robust model fusion in federated learning, in: Advances in Neural Information Processing Systems, pp. 2351–2363.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, pp. 1273–1282.

Mostafa, H., 2019. Robust federated learning through representation matching and adaptive hyper-parameters. arXiv:1912.13075.

Nguyen, M.N., Pandey, S.R., Dang, T.N., Huh, E.N., Tran, N.H., Saad, W., Hong, C.S., 2022. Self-organizing democratized learning: Toward large-scale distributed learning systems. IEEE Transactions on Neural Networks and Learning Systems.

Nguyen, M.N., Pandey, S.R., Thar, K., Tran, N.H., Chen, M., Bradley, W.S., Hong, C.S., 2021. Distributed and democratized learning: Philosophy and research challenges. IEEE Computational Intelligence Magazine 16, 49–62.

Oh, J., Kim, S., Yun, S.Y., 2022. FedBABU: Toward enhanced representation for federated image classification, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=HuaYQfggn5u.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32.

Peng, Y., Qi, J., 2019. Cm-gans: Cross-modal generative adversarial networks for common representation learning. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 15, 1–24.

Reddi, S.J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., McMahan, H.B., 2020. Adaptive federated optimization, in: International Conference on Learning Representations.

Samsung, 2021. Smarter future powered by ai. samsung.com/semiconductor/minisite/exynos/technology/ai/.

24
Singh, A., Vepakomma, P., Gupta, O., Raskar, R., 2019. Detailed comparison of communication efficiency of split learning and federated learning. arXiv preprint arXiv:1909.09145.

Sun, Y., Chong, N., Ochiai, H., 2023. Feature distribution matching for federated domain generalization, in: Asian Conference on Machine Learning, PMLR. pp. 942–957.

Thapa, C., Arachchige, P.C.M., Camtepe, S., Sun, L., 2022. Splitfed: When federated learning meets split learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 8485–8493.

Thwal, C.M., Nguyen, M.N., Tun, Y.L., Kim, S.T., Thai, M.T., Hong, C.S., 2024. Ondev-lct: On-device lightweight convolutional transformers towards federated learning. Neural Networks 170, 635–649.

Vepakomma, P., Gupta, O., Swedish, T., Raskar, R., 2018. Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564.

Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y., 2020. Federated learning with matched averaging, in: International Conference on Learning Representations.

Xiao, H., Rasul, K., Vollgraf, R., 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747.

Xiong, Y., Wang, R., Cheng, M., Yu, F., Hsieh, C.J., 2023. Feddm: Iterative distribution matching for communication-efficient federated learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16323–16332.

Xu, X., Lin, K., Gao, L., Lu, H., Shen, H.T., Li, X., 2020. Learning cross-modal common representations by private-shared subspaces separation. IEEE Transactions on Cybernetics.

Yu, F., Zhang, W., Qin, Z., Xu, Z., Wang, D., Liu, C., Tian, Z., Chen, X., 2020. Heterogeneous federated learning. arXiv:2008.06767.

Zhang, J., Guo, S., Ma, X., Wang, H., Xu, W., Wu, F., 2021a. Parameterized knowledge transfer for personalized federated learning. Advances in Neural Information Processing Systems 34, 10092–10104.
Appendix A. Supplemental Results

Appendix A.1. Cross Entropy Loss:

\[ \ell_{CE}(z|D_r) = - \sum_{x \in D_r} \sum_c P(y_c|x) \ln P(z_c|x), \]

where \( P(z_c|x) \) is the prediction outcome probability and \( P(y_c|x) \) is the target label of class \( c \) given data sample \( x \).

Appendix A.2. CIFAR-10 and CIFAR-100 results

Similar to the learning behaviors with the Fashion-MNIST dataset in the main manuscript, CIFAR-10 and CIFAR-100 exhibit faster convergence of the C-Gen performance, comparable Global and C-Spec performance and its stability with different settings as shown in Fig. A.9 and Fig. A.11, respectively, in scenario (1). Regarding scenario (2), as shown in Fig. A.10 and Fig. A.12, FedDyn and MOON depict the highest C-Gen performance in two datasets respectively, meanwhile CDKT-FL outperforms all baselines in terms of Global and C-Spec performance. Also, other methods extensively fluctuate since the learned models differ highly between clients. Moreover, as shown in Fig. A.13 and Fig. A.14, using smaller learning models at clients with two CNN layers while three CNN layers in the server model show a slight degradation due to lower learning capacities in both datasets. Additionally,
we plot the box plot for different settings of CDKT-FL to compare with other methods in terms of stability in both scenarios, as shown in Fig. A.15 and Fig. A.16. Similar to the Fashion-MNIST dataset, our algorithm achieves a stable convergence in the last 20 rounds, while other baselines show significant fluctuation.

Appendix A.3. Model Architecture

In Table A.3 and Table A.4, we provide a summary of model architecture that we used in our experiments. For Fashion-MNIST, we use the CNN model with two convolution layers, whereas three convolution layers are used in CIFAR-10 and CIFAR-100 datasets, followed by max pooling and fully connected layers.

Appendix A.4. Data Partition

We provide the data partition of three datasets in Fig. A.17 and Fig. A.18. We generate the non-i.i.d properties by assigning data from randomly 2 classes out of the total classes to each client for Fashion-MNIST and CIFAR-10 datasets and 20 classes in 100 classes for CIFAR-100 datasets.

Appendix A.5. Detailed of C-Gen and C-Spec performance

We provide the detailed results of average accuracy and F1-score of C-Gen and C-Spec performance for scenario (1) in Table A.5 and Table A.6 and scenario (2) in Table A.7 and Table A.8.

Appendix A.6. Loss vs. Global Rounds Analysis

We analyze the impact of different loss functions on the training process and model performance regarding average local loss and global loss on three datasets, as shown in Fig. A.19 and Fig. A.20. We observe that JSD and KL converge quickly in terms of average local loss, whereas KL-N shows the best convergence in terms of global loss. In both two cases, N performs the slowest decline. In our experiments, although JSD illustrates the solid convergence in terms of loss value, KL-N exhibits the best accuracy in most scenarios.

Appendix A.7. Scalability

To assess the impact of increasing the number of local users, we conducted experiments under two settings: (1) partitioning the dataset into 50 clients and randomly sampling 25 clients for each training round, and (2) partitioning the dataset into 100 clients with a random sample of 20 clients per training
round. The median testing accuracy for these scenarios is presented in Table A.9, Table A.10 and Table A.11, while the convergence in the number of communication rounds is depicted in Fig. A.21, Fig. A.22 and Fig. A.23. The results highlight distinct patterns in Global and C-Per performance across all datasets. In terms of Global performance, an increase in the sample fraction from 0.2 to 0.5 prompts the global model to generalize across a larger set of clients, leading to a slight decrease in performance. On the other hand, an increase in the number of clients results in comparable Global performance after a greater number of communication rounds. Turning to C-Per performance, elevating the sample fraction from 0.2 to 0.5 tends to enhance C-Per outcomes as the subset of clients undergoes fewer replacements. Increasing the sample fraction provides clients with more opportunities to participate in the training process, thereby leading to improved C-Per performance. However, increasing the number of clients with a consistent fraction leads to a decline in C-Per performance, attributed to the more frequent replacement of the subset of clients. These findings affirm the scalability of our approach and its ability to handle diverse numbers of local devices while preserving performance.
Figure A.10: Comparative analysis of CDKT-FL under various settings against other baselines using the CIFAR-10 dataset in Subset of Users scenario.

Table A.3: Detailed information of the CNN architecture used in our experiment for Fashion-MNIST dataset.

| Layer | Parameter & Shape & Hyper-parameters |
|-------|--------------------------------------|
| conv1 | Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)) |
| pool1 | MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1) |
| dropout1 | Dropout(p=0.4) |
| conv2 | Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)) |
| pool2 | MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1) |
| dropout2 | Dropout(p=0.4) |
| fc1  | Linear(in_features=32*7*7, out_features=512) |
| fc2  | Linear(in_features=512, out_features=10) |
Table A.4: Detailed information of the CNN architecture used in our experiment for CIFAR-10 and CIFAR-100 datasets.

| Layer | Parameter & Shape & Hyper-parameters |
|-------|---------------------------------------|
| conv1 | Conv2d(3, 16, kernel_size=(3,3), stride=(1,1), padding=(1, 1)) |
| pool1 | MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1) |
| conv2 | Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)) |
| pool2 | MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1) |
| conv3 | Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)) |
| pool3 | MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1) |
| fc1   | Linear(in_features=64*4*4, out_features=512) |
| fc2   | Linear(in_features=512, out_features=64) |
| dropout | Dropout(p=0.5) |
| fc3   | Linear(in_features=64, out_features=num_classes, bias=True) |

Table A.5: The median of test accuracy across three datasets over a range of 90 to 100 rounds in Fixed Users scenario.

|                      | Fashion-MNIST | CIFAR-10 | CIFAR-100 |
|----------------------|---------------|----------|-----------|
|                      | Global  | C-Gen | C-Spe | Global  | C-Gen | C-Spe | Global  | C-Gen | C-Spe |
| No Transfer          | 83.72  | 19.58 | 98.80 | 55.92  | 22.74 | 78.11 | 17.27   | 8.29   | 29.23 |
| FedAvg               | 82.56  | 53.05 | 98.14 | 59.85  | 26.24 | 79.66 | 15.53   | 7.58   | 26.67 |
| Scaffold             | 67.67  | 20.58 | 98.60 | 48.41  | 21.40 | 74.25 | 13.48   | 5.87   | 20.52 |
| MOON                 | 83.28  | 52.35 | 99.30 | 60.83  | 26.62 | 79.55 | 17.37   | 7.17   | 27.62 |
| FedDyn               | 83.26  | 74.58 | 95.58 | 58.00  | 38.63 | 79.85 | 14.03   | 10.68  | 29.78 |
| CDKT (Rep)           | 84.88  | 75.37 | 96.74 | 58.44  | 40.59 | 77.91 | 17.75   | 12.27  | 31.34 |
| CDKT (Full)          | 85.81  | 65.19 | 98.14 | 58.91  | 37.93 | 73.74 | 17.84   | 11.06  | 32.80 |
| CDKT (RepFull)       | **86.51**| 69.70 | 98.37 | 59.09  | 39.88 | 75.59 | **18.49**| 11.69  | 32.76 |
Figure A.11: Comparative analysis of CDKT-FL under various settings against other baselines using the CIFAR-100 dataset in Fixed Users scenario.

Figure A.12: Comparative analysis of CDKT-FL under various settings against other baselines using the CIFAR-100 dataset in Subset of Users scenario.
Figure A.13: Performance analysis of CDKT-FL with heterogeneous models on the CIFAR-10 dataset.

Figure A.14: Performance analysis of CDKT-FL with heterogeneous models on the CIFAR-100 dataset.
Figure A.15: Convergence analysis of CDKT-FL under various settings against other baselines using the CIFAR-10 dataset.
(a) Fixed Users Scenario

(b) Subset of Users Scenario

Figure A.16: Convergence analysis of CDKT-FL under various settings against other baselines using the CIFAR-100 dataset.
Figure A.17: Visualization of the data distribution on three datasets for 10 clients in the Fixed Users scenario. The color bar denotes the number of data samples. Each rectangle defines the number of samples in each class of each client.

Figure A.18: Visualization of the data distribution on three datasets for 50 clients in the Subset of Users scenario. The color bar denotes the number of data samples. Each rectangle defines the number of samples in each class of each client.

Figure A.19: Average local loss versus global rounds on three datasets
Figure A.20: Global loss versus global rounds on three datasets

(a) Fashion-MNIST  
(b) CIFAR-10  
(c) CIFAR-100

Table A.6: The median of F1 score (weighted) across three datasets over a range of 90 to 100 rounds in Fixed Users scenario.

|                  | Fashion-MNIST | CIFAR-10 | CIFAR-100 |
|------------------|---------------|----------|-----------|
|                  | Global  | C-Gen  | C-Spe  | Global  | C-Gen  | C-Spe  | Global  | C-Gen  | C-Spe  |
| No Transfer      | 86.65   | 18.92  | 98.72  | 64.17   | 20.66  | 75.82  | 20.68   | 6.32   | 22.56  |
| FedAvg           | 81.75   | 52.45  | 98.62  | 66.46   | 25.62  | 78.07  | 18.19   | 2.96   | 22.62  |
| Scaffold         | 77.73   | 19.99  | 97.85  | 57.52   | 19.97  | 72.31  | 15.40   | 0.73   | 19.66  |
| MOON             | 85.53   | 61.29  | 98.63  | 69.03   | 32.98  | 80.60  | 20.81   | 2.26   | 23.69  |
| FedDyn           | 85.11   | 75.58  | 96.36  | 65.93   | 45.02  | 79.66  | 19.72   | 2.26   | 31.26  |
| CDKT (Rep)       | 87.04   | 81.61  | 98.09  | 67.15   | 47.68  | 77.12  | 23.60   | 12.82  | 30.26  |
| CDKT (Full)      | 86.37   | 70.64  | 98.40  | 67.21   | 42.24  | 74.40  | 23.37   | 7.37   | 31.93  |
| CDKT (RepFull)   | 86.69   | 66.14  | 99.49  | 68.15   | 43.59  | 76.73  | 24.68   | 9.71   | 32.89  |
Table A.7: The median of test accuracy across three datasets over a range of 90 to 100 rounds in Subset of Users scenario.

| Method          | Fashion-MNIST | CIFAR-10 | CIFAR-100 |
|----------------|---------------|----------|-----------|
|                | Global | C-Gen | C-Spe | Global | C-Gen | C-Spe | Global | C-Gen | C-Spe |
| No Transfer    | 78.19  | 19.09 | **98.86** | 52.17  | 17.14 | **76.35** | 16.47  | 3.04  | 13.22 |
| FedAvg         | 69.44  | 65.33 | 75.90 | 48.83  | 41.69 | 52.70 | 12.25  | 10.81 | 14.34 |
| Scaffold       | 68.00  | 62.36 | 73.40 | 40.76  | 37.97 | 48.16 | 12.29  | 10.89 | 13.35 |
| MOON           | 70.05  | 66.53 | 74.87 | 53.05  | 46.82 | 57.48 | 16.79  | 14.77 | 18.61 |
| FedDyn         | 76.03  | 72.60 | 79.62 | 55.71  | 47.84 | 55.98 | 16.37  | 14.59 | 18.39 |
| CDKT (Rep)     | 78.50  | **69.70** | 92.10 | 55.67  | 38.33 | 69.52 | 16.72  | 7.41  | 24.07 |
| CDKT (Full)    | 79.22  | 62.86 | 95.17 | **56.60** | 38.02 | 70.69 | 17.33  | 7.76  | 23.76 |
| CDKT (RepFull) | **79.94** | 67.82 | 94.96 | 56.57  | 38.85 | 69.22 | **17.60** | 10.32 | **25.15** |

Table A.8: The median of F1 score (weighted) across three datasets over a range of 90 to 100 rounds in Subset of Users scenario.

| Method          | Fashion-MNIST | CIFAR-10 | CIFAR-100 |
|----------------|---------------|----------|-----------|
|                | Global | C-Gen | C-Spe | Global | C-Gen | C-Spe | Global | C-Gen | C-Spe |
| No Transfer    | 85.65  | 18.35 | **97.48** | 64.84  | 11.04 | 60.90 | 21.57  | 1.75  | 8.21  |
| FedAvg         | 73.91  | 70.76 | 80.54 | 61.54  | 53.08 | 63.28 | 12.50  | 10.90 | 13.76 |
| Scaffold       | 75.51  | 78.79 | 87.59 | 51.34  | 46.96 | 53.77 | 14.04  | **12.96** | 14.22 |
| MOON           | 74.53  | 71.70 | 82.09 | 65.00  | **58.56** | 66.88 | 18.85  | 12.62 | 15.06 |
| FedDyn         | 81.33  | **79.19** | 83.91 | 64.32  | 56.09 | 65.18 | 18.54  | 12.59 | 15.87 |
| CDKT (Rep)     | 86.41  | 76.51 | 95.41 | 65.69  | 48.86 | **75.96** | 22.61  | 7.17  | **21.73** |
| CDKT (Full)    | 86.39  | 70.30 | 96.40 | **67.07** | 46.68 | 74.67 | **22.86** | 7.12  | 18.76 |
| CDKT (RepFull) | **86.46** | 75.51 | 96.29 | 66.63  | 51.75 | 74.27 | 22.67  | 7.52  | 18.69 |
Table A.9: The median of test accuracy of CDKT-FL on Fashion-MNIST dataset in different settings of clients.

|                      | 50 clients, sample fraction = 0.2 | 50 clients, sample fraction = 0.5 | 100 clients, sample fraction = 0.2 |
|----------------------|-----------------------------------|-----------------------------------|-----------------------------------|
|                      | 100 rounds                        | 100 rounds                        | 100 rounds                        |
|                      | Global                            | C-Per                            | Global                            | C-Per                            | Global                            | C-Per                            |
| CDKT (Rep)           | 18.50                             | 80.90                            | 77.83                             | 82.42                            | 79.23                             | 79.02                            |
| CDKT (Full)          | 78.22                             | 79.02                            | 78.91                             | 82.04                            | 79.33                             | 79.41                            |
| CDKT (RepFull)       | 79.94                             | 81.49                            | 78.92                             | 83.32                            | 80.23                             | 79.69                            |

Table A.10: The median of test accuracy of CDKT-FL on CIFAR-10 dataset in different settings of clients.

|                      | 50 clients, sample fraction = 0.2 | 50 clients, sample fraction = 0.5 | 100 clients, sample fraction = 0.2 |
|----------------------|-----------------------------------|-----------------------------------|-----------------------------------|
|                      | 100 rounds                        | 100 rounds                        | 100 rounds                        |
|                      | Global                            | C-Per                            | Global                            | C-Per                            | Global                            | C-Per                            |
| CDKT (Rep)           | 55.67                             | 53.92                            | 53.72                             | 57.66                            | 54.06                             | 47.27                            |
| CDKT (Full)          | 56.60                             | 54.35                            | 54.78                             | 56.05                            | 56.12                             | 47.41                            |
| CDKT (RepFull)       | 56.59                             | 54.03                            | 54.93                             | 55.86                            | 55.12                             | 49.42                            |

Table A.11: The median of test accuracy of CDKT-FL on CIFAR-100 dataset in different settings of clients.

|                      | 50 clients, sample fraction = 0.2 | 50 clients, sample fraction = 0.5 | 100 clients, sample fraction = 0.2 |
|----------------------|-----------------------------------|-----------------------------------|-----------------------------------|
|                      | 100 rounds                        | 100 rounds                        | 100 rounds                        |
|                      | Global                            | C-Per                            | Global                            | C-Per                            | Global                            | C-Per                            |
| CDKT (Rep)           | 16.72                             | 15.74                            | 16.92                             | 18.58                            | 17.17                             | 16.28                            |
| CDKT (Full)          | 17.33                             | 17.76                            | 17.58                             | 20.80                            | 17.95                             | 16.47                            |
| CDKT (RepFull)       | 17.60                             | 17.74                            | 17.60                             | 18.78                            | 17.97                             | 16.28                            |
Figure A.21: Convergence of CDKT-FL with different settings of clients Fashion-MNIST dataset
Figure A.22: Convergence of CDKT-FL with different settings of clients with CIFAR-10 dataset
Figure A.23: Convergence of CDKT-FL with different settings of clients with CIFAR-100 dataset