Knowledge-Aware Federated Active Learning with Non-IID Data

Yu-Tong Cao[1], Ye Shi[2], Baosheng Yu[1], Jingya Wang[2], Dacheng Tao[1]

[1] Sydney AI Centre, School of Computer Science, The University of Sydney

[2] ShanghaiTech University

ycao5602@uni.sydney.edu.au, shiye@shanghaitech.edu.cn, baosheng.yu@sydney.edu.au,

wangjingya@shanghaitech.edu.cn, dacheng.tao@gmail.com

Abstract

Federated learning enables multiple decentralized clients to learn collaboratively without sharing local data. However, the expensive annotation cost on local clients remains an obstacle in utilizing local data. In this paper, we propose a federated active learning paradigm to efficiently learn a global model with a limited annotation budget while protecting data privacy in a decentralized learning manner. The main challenge faced by federated active learning is the mismatch between the active sampling goal of the global model on the server and that of the asynchronous local clients. This becomes even more significant when data is distributed non-IID across local clients. To address the aforementioned challenge, we propose Knowledge-Aware Federated Active Learning (KAFAL), which consists of Knowledge-Specialized Active Sampling (KSAS) and Knowledge-Compensatory Federated Update (KCFU). Specifically, KSAS is a novel active sampling method tailored for the federated active learning problem, aiming to deal with the mismatch challenge by sampling actively based on the discrepancies between local and global models. KSAS intensifies specialized knowledge in local clients, ensuring the sampled data is informative for both the local clients and the global model. Meanwhile, KCFU deals with the client heterogeneity caused by limited data and non-IID data distributions by compensating for each client's ability in weak classes with the assistance of the global model. Extensive experiments and analyses are conducted to show the superiority of KAFAL over recent state-of-the-art active learning methods. Code is available at https://github.com/ycao5602/KAFAL.

1. Introduction

Federated learning is a decentralized paradigm that allows collaborative learning of local devices to attain a powerful global model in a central server through aggregation without accessing local data [25, 35]. Most federated learn-
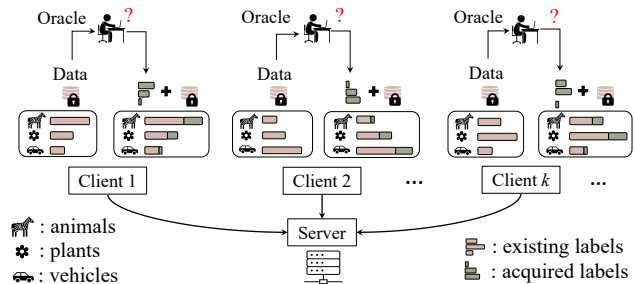

Figure 1. The primary federated active learning framework with non-IID data. Each client maintains an active learning loop to select informative data for annotation with a limited annotation budget. We show each model's existing labelled data in different classes with pink bars and the newly acquired labels with green bars. Clients specialize in different classes due to non-IID data distributions.

ing methods consider supervised learning scenarios with fully annotated training data on each local client. However, the high annotation cost has been a challenge for real-world federated learning scenarios, e.g., large-scale medical data located in different medical institutions while medical specialists for data annotation are very limited in each institution. In this paper, we consider a new federated active learning paradigm, which aims to not only protect data privacy but also make the most of the very limited annotation budget on each local client for decentralized model training. An illustration of the proposed federated active learning with non-IID data framework is shown in Fig. 1.

In federated active learning, we aim to attain a powerful global model on the server by sampling only local data and training the model on each local client. A straightforward solution for federated learning is to directly apply off-the-shelf active learning methods to each client. Specifically, existing methods can mainly be categorized into diversity-based [42, 2], uncertainty-based [49, 44, 51, 45, 23, 8, 3], and discrepancy-based. [43, 6]. Therefore, we actively sample based on either the statistics from each client model or the downloaded global model. However, the former ap-

proach may yield benefits primarily for local clients, while the latter might result in the loss of valuable information during aggregation, even if the selected data is advantageous for the global model on the server. Through experiments in subsequent sections, we demonstrate that active sampling with the global model on the server struggles to derive benefits due to this indirect process.

A major challenge in federated active learning is therefore, the mismatch between the active sampling goal of the clients and that of the model on server caused by asynchronous models. What makes it even more challenging is the statistical heterogeneity resulting from the non-IID data distributions on clients in a typical federated learning setting [35, 52, 17, 29]. Ideally, the models can synchronize with sufficiently many aggregations from local clients to the model on the server. However, the communication costs usually make the above-mentioned solution impractical [35]. Therefore, the model parameters of each client and the global model vary due to non-IID distributions, leading to a higher degree of mismatch between the sampling goals.

To address the aforementioned challenge, we propose a federated active learning scheme, namely **Knowledge-Aware Federated Active Learning (KAFAL)**. It comprises two key components, Knowledge-Specialized Active Sampling and Knowledge-Compensatory Federated Update. **Knowledge-Specialized Active Sampling (KSAS)** is a new active sampling strategy, where each client model learns to intensify its specialized knowledge in order to annotate universally informative data that benefit both the clients and the global model. Specifically, we compute the intensified discrepancy between the client and global model outputs based on the specialized knowledge of each client. In addition, the insufficiency of labelled training data together with the statistical heterogeneity caused by non-IID data can degrade the federated update quality, e.g., clients may perform weakly for certain classes. Aggregating these clients, extra communications are required to achieve convergence. Therefore, we further devise a new update rule, **Knowledge-Compensatory Federated Update (KCFU)**, by compensating for weak classes (or low-frequency classes) on each client through knowledge distillation from the global model. The main contributions of this paper are as follows:

- We explore a rarely studied problem, federated active learning with non-IID data, which aims at efficiently learning a global model with a limited annotation budget on each client under a heterogeneous federated learning framework. Notably, we reveal the main challenge in federated active learning is the mismatch between the active sampling goal of the clients and that of the server caused by asynchronous models.

- We introduce a federated active learning paradigm,

known as KAFAL, with a novel active sampling method KSAS and a novel federated update method KCFU to handle the aforementioned challenge. KSAS is designed to sample universally informative data by computing the intensified discrepancies between the clients' and the global model's outputs based on the specialized knowledge of each client. KCFU is devised to deal with data heterogeneity by compensating for weak classes using knowledge distillation from the global model.

- We conduct extensive experiments on different benchmarks to demonstrate the superiority of the proposed method, where comprehensive ablation studies are also provided to validate the design of the proposed KAFAL.

2. Related Work

2.1. Federated Learning

Federated learning is a learning paradigm that allows decentralized training of a model on the central server with training data distributed over a number of local clients in a non-IID manner [25, 35, 18, 37, 19, 4, 30, 13]. Specifically, Konevcny et al. [25] first introduced the term and proposed a method, FedAvg, to aggregate the client models, which was later improved by FedAvgM to accumulate model updates with momentum [18, 19]. Federated learning has also been discussed in more practical views, such as federated multi-task learning [34], federated domain adaptation [40, 48], federated continual learning [50], semi-supervised federated learning [22, 47], and unsupervised federated learning [31]. Specifically, Jeong et al. [22] considered the deficiency of data labels in federated learning and proposed a semi-supervised solution. Ahn et al. [1] and Kim et al. [24] discussed a federated active learning paradigm, while they only considered the less realistic IID data scenario. To the best of our knowledge, we are the first to explore the active data sampling problem in the non-IID federated learning framework.

2.2. Active Learning

Existing active learning methods can be categorized into diversity-based, uncertainty-based, and discrepancy-based methods. Specifically, diversity-based methods [42, 2, 38] select representative and diverse data points that span the data space for query. Sener et al. [42] proposed a core-set approach that selects the most representative core-set from the data pool using k-center algorithms. Recently, Ash et al. [2] proposed to actively select the data points that produce gradients with diverse directions. Uncertainty-based methods [49, 44, 51, 45, 23, 3] estimate the uncertainty of predictions using different metrics and select data

points accordingly. Despite being simple to use, these methods cannot be directly applied in federated active learning, without the mismatch between each client model and the global model being handled. Some recent methods explicitly measure the informativeness of data points instead of directly calculating the uncertainty metrics [49, 44, 51, 45, 8]. Specifically, Sinha et al. [44] utilized an extra variational auto-encoder to select data points that are less likely to be distributed in the labelled pool for querying. Despite effective sampling, these methods require extra modules for sampling with an increased computational cost. Discrepancy-based methods [43, 9, 7, 36, 6] pass data points through an ensemble of models, namely a committee, and select the data points that cause large discrepancy within the committee. Freund et al. [9] proposed to randomly pick two models in the committee that are consistent for labelled data and then use them to sample from unlabelled data. Multiple models usually make discrepancy-based methods stable, but also increase the computational cost. This partially explains why they become less popular with the rise of deep active learning. It is costly to fit them in federated active learning.

Many recent methods have also been proposed to enable active learning in more challenging settings, e.g., low-budget active learning [33], biased-data active learning [14], semi-supervised active learning [11, 20] and cross-domain active learning [10, 32]. Our work also considers applying active learning in a more practical decentralized federated learning setting where local data privacy is protected. Chen et al. [5] proposed a novel automated learning system for distributed active learning that requires a shared labelled set. Furthermore, Goetz et al. [12] considered active learning in a federated learning framework that studies how to select clients actively. Our work, instead, considers the active sampling of data on each local client in federated learning.

3. Method

In this section, we first describe the problem setting of federated active learning and then introduce two main components, i.e., KSAS and KCFU in the proposed KAFAL.

3.1. Problem Setting

We illustrate the overview of the federated active learning framework in Fig. 1 and sum up the proposed KAFAL algorithm in Alg. 1. In federated active learning, we keep K local client models parametrized with $\{\boldsymbol{\omega}_i\}_{i=1}^{K}$ and one global model on central server parametrized with $\boldsymbol{\Omega}$. Each client model i is optimized using its local training dataset $\mathcal{D}_i$. Different from standard federated learning, federated active learning annotates a subset of data samples on each client with a local active learning loop. The training set $\mathcal{D}_i$ for client i is divided into a labelled set $\mathcal{D}_i^L$ and an unlabelled set $\mathcal{D}_i^U$. In each communication round, a fraction $R \in (0, 1]$ of the total K clients are first randomly selected

---

Algorithm 1 Knowledge-Aware Federated Active Learning

Data: local datasets $\{\mathcal{D}_k^L\}_{i=1}^{K}$ and $\{\mathcal{D}_i^U\}_{i=1}^{K}$
Input: T, R, sampling budgets $\{b_i\}_{i=1}^{K}$
Parameter: $\boldsymbol{\Omega}$, $\{\boldsymbol{\omega}_i\}_{i=1}^{K}$

1: **for** active round a=1 to A **do**
2:    **Federated Update: KCFU**
3:    Initialize the global model with $\boldsymbol{\Omega}^0$
4:    **for** communication round $t = 1$ to T **do**
5:      $S_t \leftarrow$ Random subset of $\lceil R \cdot K \rceil$ clients.
6:      **for** client $k \in S_t$ **do**
7:        Download the global model's parameters $\boldsymbol{\Omega}^t$
8:        Copy to the client $\boldsymbol{\omega}_k^t \leftarrow \boldsymbol{\Omega}^t$
9:        $\boldsymbol{\omega}_k^{t+1} \leftarrow$ LocalUpdate($\boldsymbol{\omega}_k^t; \mathcal{D}_k^L, \mathcal{D}_k^U$)
10:       Upload the local model parameters to the server
11:      **end for**
12:      **for** client $k' \notin S_t$ **do**
13:        Keep the client model unchanged $\boldsymbol{\omega}_{k'}^{t+1} \leftarrow \boldsymbol{\omega}_{k'}^t$
14:      **end for**
15:      Aggregate the clients with Eq. (7) to update $\boldsymbol{\Omega}^{t+1}$
16:      **for** each client $k \in S_t$ **do**
17:        Download $\boldsymbol{\Omega}_k^{t+1}$ and save as $\hat{\boldsymbol{\Omega}}_k$
18:      **end for**
19:    **end for**
20:    **Active Sampling: KSAS**
21:    **for** client $i = 1$ to K **do**
22:      **for** each unlabelled data $x \in \mathcal{D}_i^U$ **do**
23:        **for** class $y \in \mathbb{C}$ **do**
24:          Compute $P_y^i(\boldsymbol{x})$ on class y using Eq. (1)
25:          Compute $Q_y^i(\boldsymbol{x})$ on class y using Eq. (2)
26:        **end for**
27:        Compute $D^i(\boldsymbol{x})$ using Eq. (3)
28:      **end for**
29:      Send b_i unlabelled data points with the largest D to the oracle for annotation
30:      Remove the annotated data in $\mathcal{D}_i^U$ and add to $\mathcal{D}_i^L$
31:    **end for**
32: **end for**
33: **Return** $\{\mathcal{D}_i^L\}_{i=1}^{K}$ and $\{\mathcal{D}_i^U\}_{i=1}^{K}$

---

as a subset S_t, which simulates the real-world scenarios that some local devices may be offline from time to time. After that, the selected clients first download $\boldsymbol{\Omega}$ from the server to initialize $\{\boldsymbol{\omega}_k\}_{k \in S_t}$, and then conduct local update based on $\{\mathcal{D}_k\}_{k \in S_t}$. The updated $\{\boldsymbol{\omega}_k\}_{k \in S_t}$ will be uploaded to the central server and aggregated to update $\boldsymbol{\Omega}$. The training process terminates after T communication rounds. After that, a batch of unlabelled data is sampled from each $\mathcal{D}_i^U$, sent to the local oracle for annotation, and added to the labelled data pool $\mathcal{D}_i^L$ for each client i. The sampling budget for the client i is b_i. The active sampling process is repeated for A times, where A is set according to need.

The training sets $\{\mathcal{D}_i\}_{i=1}^{K}$ follow non-IID distributions. All client models share the same architecture with the global model to synchronize model parameters between the client and server. Just like in federated learning, transferring the local data $\{\mathcal{D}_i\}_{i=1}^{K}$ across clients (or server) is prohibited in federated active learning. The objective of federated active learning is to actively annotate local data with limited budgets to improve the overall model performance without violating data privacy.

3.2. Knowledge-Specialized Active Sampling

Given the mismatch problem in federated active learning, informative data on each client may not be that informative to the global model due to the non-IID data distributions, meaning that using only one of them for active sampling is therefore not reliable. Computing the model discrepancy between each client and the global model allows us to consider both aspects in active sampling. But alone is insufficient. Data from rare classes in each local dataset can cause large discrepancies between the client and the global model. However, they are usually uninformative to the global model and can hardly contribute to the client model's updates. Being rare locally makes their contributions limited in the gradient computation. Furthermore, during aggregation, the global model may not find them as informative as they are to the clients. Hence, we propose to enable each client to intensify its specialized knowledge (common class knowledge) in the computation of discrepancy to sample more informative data containing specialized knowledge. We introduce the Knowledge-Specialized KL-Divergence as follows. On top of a symmetrized KL-Divergence [21, 27], our Knowledge-Specialized KL-Divergence further incorporates a Knowledge-Specialized component to accentuate each client's specialized knowledge. The Knowledge-Specialized probability of client i being predicted to class y is formulated as:

$$P_y^i(\boldsymbol{x}) = \frac{n_{i,y}^\lambda \exp\Big(g_y(\boldsymbol{x}; \boldsymbol{\omega}_i)\Big)}{\sum_{c \in \mathbb{C}} n_{i,c}^\lambda \exp\Big(g_c(\boldsymbol{x}; \boldsymbol{\omega}_i)\Big)}, \quad (1)$$

where x is an unlabelled data point sampled from $\mathcal{D}_i^U$, $g_y(\boldsymbol{x}; \boldsymbol{\omega}_k)$ is the prediction score at the y-th class, $\mathbb{C}$ indicates the set of all classes, $n_{i,y}$ is the number of data points that belong to class y in $\mathcal{D}_i^L$, and λ is a hyperparameter which controls the knowledge-specialized level. We name $n_{i,y}^\lambda$ as the knowledge weight which indicates the client's knowledge in each class. Similarly, the knowledge-specialized probability of the global model predicted to be class y can be defined as:

$$Q_y^i(\boldsymbol{x}) = \frac{n_{i,y}^\lambda \exp\Big(g_y(\boldsymbol{x}; \hat{\boldsymbol{\Omega}}_i)\Big)}{\sum_{c \in \mathbb{C}} n_{i,c}^\lambda \exp\Big(g_c(\boldsymbol{x}; \hat{\boldsymbol{\Omega}}_i)\Big)}, \quad (2)$$
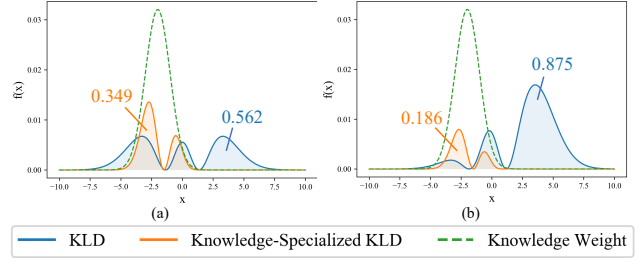

Figure 2. Illustration of how Knowledge-Specialized KL-Divergence intensifies specialized knowledge compared to the standard KL-Divergence. The blue and orange lines integrate to be KL-Divergence and the knowledge-specialized KL-Divergence computed from the same pair of distributions. The blue and orange numbers show the integrated areas of the blue and orange curves in each image, respectively.

where $\hat{\boldsymbol{\Omega}}_i$ is a copy of global model parameters downloaded from the server to client i. The knowledge-specialized KL-Divergence is defined as:

$$D^i(\boldsymbol{x}) = \sum_{y \in \mathbb{C}} \Big(P_y^i(\boldsymbol{x}) \ln \frac{P_y^i(\boldsymbol{x})}{Q_y^i(\boldsymbol{x})} + Q_y^i(\boldsymbol{x}) \ln \frac{Q_y^i(\boldsymbol{x})}{P_y^i(\boldsymbol{x})}\Big), \quad (3)$$

where x is data from the unlabelled pool of client i. The knowledge-specialized KL-Divergence focuses on each client's specialized knowledge and selects more informative data points from its specialized classes for labelling. In the Knowledge-Specialized probabilities, $\{n_{i,c}\}_{c \in \mathbb{C}}$ serve to amplify the KL-Divergence on class c if the class is considered to contain the client's specialized knowledge.

Visualization. To better visualize how Knowledge-Specialized KL-Divergence intensifies specialized knowledge compared to the standard KL-Divergence, we use continuous distributions to simulate model predictions and compute the divergences in Fig. 2. Note that the knowledge weight curves represent a continuous version of our knowledge weights. For clarity, we only show the KLD and Knowledge-Specialized KLD and omit the distribution curves in the figure. (a) and (b) can be viewed as global-local discrepancies from two different inputs on the same client model since the KLD values are different and the knowledge weights are the same. Although (a) has a smaller KLD, its knowledge-specialized KLD is larger, meaning that if we used KLD for sampling, (a) is less likely to be sampled. On the other hand, if our proposed knowledge-specialized KLD is used, (a) is more likely to be sampled than (b). What makes the results different is the knowledge weight. It intensifies the client's specialized knowledge and suppresses the less reliable divergence contributed by unfamiliar knowledge of the client. More of the model difference in (a) is caused by specialized knowledge (peak area of knowledge weight) other than that in (b). More analyses are provided in the supplementary (A.10).
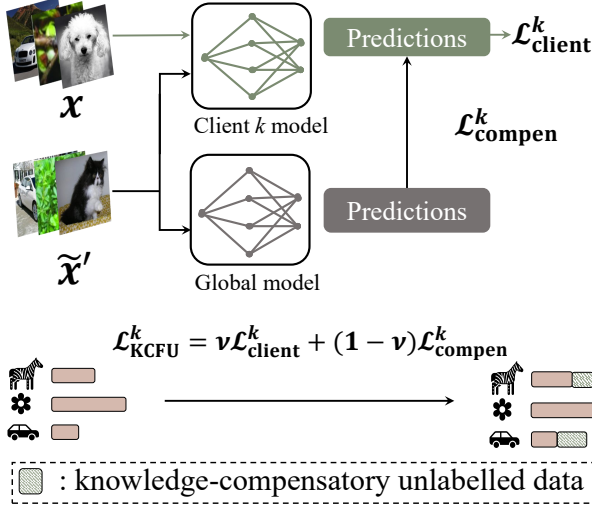
Figure 3. KCFU compensates for each client's ability on weak classes through knowledge distillation from the global model. Unlabelled data are used in the process.

3.3. Knowledge-Compensatory Federated Update

An overview of knowledge-compensatory federated update (KCFU) is shown in Fig. 3. The local data on each client follows its own realistic data distributions [18], thus leaving non-uniform class distribution on each client. Besides, our KSAS which tends to annotate data with specialized-knowledge further introduces imbalance in labelled data. Therefore, on top of the standard FedAvg [35], we introduce a balanced classifier and a knowledge-compensatory strategy.

Local Update with Balanced Loss. Our balanced classifier on each client optimizes with a balanced cross-entropy loss [41] to deal with the imbalanced local data distribution:

$$\mathcal{L}_{\text{client}}^k = -\log \frac{n_{k,y}\exp\Big(g_y(\boldsymbol{x};\boldsymbol{\omega}_k)\Big)}{\sum_{c\in\mathbb{C}} n_{k,c}\exp\Big(g_c(\boldsymbol{x};\boldsymbol{\omega}_k)\Big)}. \qquad (4)$$

In our experiment, each labelled set $\mathcal{D}_k^L$ starts from only a small proportion of the local dataset on the client k. We demonstrate with experiments in later sections that, with the imbalanced class distributions and the small size of training data, a simple local client update using cross-entropy loss is not enough for training. The balanced loss allocates more weight to data from rare classes and less weight to data from common classes to deal with the problem. It prevents the model from becoming biased towards common classes during training.

Remark: Although our balanced loss (Eq. (4)) looks similar in formulation compared with the aforementioned knowledge-specialized probabilities, i.e., Eq. (1) and (2),

they are designed for various purposes and function differently. Eq. (1) and (2) are designed to compute the active sampling scores, and Eq. (4) is a loss that updates the client models. The knowledge-specialized probabilities magnify the KL-Divergence computed from common classes for sampling, while the balanced loss magnifies the loss computed from rare classes.

Global-to-Local Knowledge Compensation. Due to the extreme limitation and non-uniformity of local data labels, the clients can perform weakly on rare classes. The weak classes of clients differ depending on the data distributions. Such statistical heterogeneity of clients can be harmful in model aggregation. To compensate for the clients' knowledge on the weak classes, we further introduce an extra loss $\mathcal{L}_{\text{compen}}$. Since the global model aggregates parameters of local clients, they usually have a more balanced performance over different classes. On classes where each client considers to be rare, the global model is likely to perform better than the client. Hence, it is reasonable to design the knowledge-compensation process which conducts knowledge distillation from the global model to the clients using unlabelled data. We later show with experiments that $\mathcal{L}_{\text{compen}}$ can save the communication cost via boosting the convergence. The loss $\mathcal{L}_{\text{compen}}$ on client k can be evaluated as follows. We first sample unlabelled data $\boldsymbol{x}'$ from $\mathcal{D}_k^U$. Then we compute the logits $\boldsymbol{z} = g(\boldsymbol{x}';\boldsymbol{\Omega})$ and the pseudo label $y' = \arg\max_c g_c(\boldsymbol{x}';\boldsymbol{\Omega})$ with the downloaded global model. The loss weight can be computed as $\boldsymbol{\Gamma}(\boldsymbol{x}') = \frac{\sum_{c\in\mathbb{C}} n_{k,c}}{n_{k,y'}}$, and the compensation loss is then defined as:

$$\mathcal{L}_{\text{compen}}^k = \boldsymbol{\Gamma}(\boldsymbol{x}') \cdot \text{KL}\Big(\sigma(\boldsymbol{z})\|\sigma(g(\boldsymbol{x}';\boldsymbol{\omega}_k))\Big), \qquad (5)$$

where σ stands for the softmax function and KL-divergence $\text{KL}(p,q) = p\ln\frac{p}{q}$. Note that no gradient is computed for the global model Ω, only. As the unlabelled data falls in the same distribution as the labelled data, rare classes in labelled data are usually still rare in unlabelled data. To make the most of the compensation loss, we further propose to augment the training with mixed unlabelled data $\tilde{\boldsymbol{x}}' = \beta\boldsymbol{x}'_1 + (1-\beta)\boldsymbol{x}'_2$, where β is a mixing weight sampled from a beta distribution. $\boldsymbol{x}'_1$ and $\boldsymbol{x}'_2$ are randomly sampled from the unlabelled batch. $\Gamma(\tilde{\boldsymbol{x}}')$ is similarly mixed as $\beta\Gamma(\boldsymbol{x}'_1) + (1-\beta)\Gamma(\boldsymbol{x}'_2)$. The compensation loss then becomes $\mathcal{L}_{\text{compen}}(\tilde{\boldsymbol{x}}';\tilde{\boldsymbol{z}},\boldsymbol{\omega}_k)$ with $\tilde{\boldsymbol{z}} = g(\tilde{\boldsymbol{x}}';\boldsymbol{\Omega})$. Therefore, the complete loss $\mathcal{L}_{\text{KCFU}}^k$ to update client k is:

$$\mathcal{L}_{\text{KCFU}}^k = \nu\mathcal{L}_{\text{client}}^k + (1-\nu)\mathcal{L}_{\text{compen}}^k, \qquad (6)$$

where ν is a tradeoff hyperparameter. We show the detailed local update algorithm in Alg. 2.

Algorithm 2 LocalUpdate($\boldsymbol{\omega}_k; \mathcal{D}_k^L, \mathcal{D}_k^U$)

Data: $\mathcal{D}_k^L, \mathcal{D}_k^U$
Input: epochs, batches, communication round t, learning rate η
Parameter: $\boldsymbol{\omega}_k$

1: **for** $e = 1$ to epochs **do**
2: **for** $b = 1$ to batches **do**
3:   Sample a batch $\{(\boldsymbol{x}, y)\} \subseteq \mathcal{D}_k^L$
4:   Compute $\mathcal{L}_{\text{client}}^k$ using Eq. (4)
5:   **if** t equals 1 **then**
6:    $\boldsymbol{\omega}_k \leftarrow \boldsymbol{\omega}_k - \eta \nabla \mathcal{L}_{\text{client}}^k$
7:   **else**
8:    Sample a batch $\{\boldsymbol{x'}\} \subseteq \mathcal{D}_k^U$
9:    Construct a mixed batch $\{\tilde{\boldsymbol{x}}'\}$
10:    Find $\boldsymbol{z}, \boldsymbol{\Gamma}(\tilde{\boldsymbol{x}}')$ for each $\tilde{\boldsymbol{x}}'$
11:    Compute $\mathcal{L}_{\text{compen}}^k$ using Eq. (5)
12:    Compute $\mathcal{L}_{\text{KCFU}}^k$ with Eq. (6)
13:    $\boldsymbol{\omega}_k \leftarrow \boldsymbol{\omega}_k - \eta \nabla \mathcal{L}_{\text{KCFU}}^k$
14:   **end if**
15: **end for**
16: **end for**
17: **Return** $\boldsymbol{\omega}_k$

Global Aggregation After local updates of clients, they are uploaded to the server and aggregated as follows:

$$\boldsymbol{\Omega}^t = \sum_{k \in S_t} \frac{N_k}{\sum_{j \in S_t} N_j} \boldsymbol{\omega}_k^t, \qquad (7)$$

where $N_k = \sum_{c \in \mathbb{C}} n_{k,c}$ indicates the number of data points in local labelled data pool $\mathcal{D}_k^L$. Lastly, we can formulate the overall objective as:

$$\underset{\{\boldsymbol{\omega}_k\}_{k \in S_t}, \boldsymbol{x} \sim \{\mathcal{D}_k^L\}_{k \in S_t}, \boldsymbol{x}' \sim \{\mathcal{D}_k^U\}_{k \in S_t}}{\arg \min} \mathcal{L}_{\text{KCFU}}, \qquad (8)$$

where $\mathcal{L}_{\text{KCFU}} = \sum_{k \in S_t} \frac{N_k}{\sum_{j \in S_t} N_j} \mathcal{L}_{\text{KCFU}}^k$, $\{\mathcal{D}_k^L\}_{k \in S_t}$ is achieved via active learning loops.

4. Experiments

In this section, we mainly conduct experiments on three image classification datasets, CIFAR10/100 [26] and MNIST [28], that are popular in both active and federated learning. Additionally, we also apply our method in a more realistic scenario by conducting medical image classification with NIH Chest X-Ray dataset [46]. Specifically, CIFAR10 and CIFAR100 contain $60,000$ images from 10 and 100 classes, respectively, including $50,000$ training images and $10,000$ testing images. Results and details of MNIST are shown in the supplementary (A.8). For the server and client models, we utilize ResNet-8 [16] as the model architecture.

We implement the method with Pytorch [39]. We use $K = 10$ clients in our experiments. In each communication round, $R = 80\%$ of the clients are selected at random to update locally. The hyperparameter λ is set to 1. More details are given in the supplementary (A.1). To distribute non-IID data to different clients, we follow Hsu et al. [18] and draw $q \sim \text{Dir}(\alpha p)$ from a Dirichlet distribution. p stands for the global prior class distribution over all classes, and $\alpha > 0$ is a concentration parameter that controls the level of IID. When $\alpha \to \infty$, the data distributions are identical to the global class distribution. When $\alpha \to 0$, each client will be allocated data from only one class. In our main results, we set $\alpha = 0.1$. We also show the results from $\alpha = 0.3$ and $\alpha = 1$. The different CIFAR10 data distributions are shown in the supplementary (A.1).

For active learning loops, we start by randomly selecting 10% data from $\mathcal{D}_i$ as the labelled pool $\mathcal{D}_i^L$ of client i. This is around 500 labelled data for each client. For each sampling cycle, the budget b_i on each client is 5% of the total local data $\mathcal{D}_i$. We sample for $A = 5$ times until the labelled data amount reaches 35% of all data for each client. We repeat each experiment 5 times with different random seeds and average the results to get a final result.

4.1. Comparison with Active Learning Methods

We compare our KAFAL with 8 other active learning methods and show the results in Fig. 4(a)(b). All methods are fit into the federated active learning framework using the same model architectures following the same training steps for fair comparison. For all baselines, we use the KCFU loss (Eq. (6)) for local update. FedAvg is used for aggregation for all methods. We categorize our baselines into five types. (I) We compare with uncertainty-based methods: entropy and top-2 margin scores (Margin). Entropy is calculated as $H(p) = -p \cdot \log(p)$, where p is the Softmax output. The top-2 margin score calculates the margin between the largest prediction score and the second largest prediction score over all classes for each data point. Unlabelled data with the lowest top-2 margin scores will be sampled for annotation. Here we compute the uncertainty scores on each client model after local update. (II) We compare with a special uncertainty-based method which explicitly learns the data loss with extra modules, Learning Loss for Active Learning (LL4AL) [49]. We train a loss prediction module for each client model. (III) We also compare with diversity methods: Core-set [42], BADGE [2], and ALFA-Mix [38]. We sample on each client model using diversity. (IV) Results from a previous discrepancy-based method, Query-by-Committee (QBC) [7], is also compared with. We use 3 models on each client for QBC. (V) Finally, we compare with random sampling results.

On both CIFAR10 and CIFAR100, our KAFAL achieves state-of-the-art results (Fig. 4(a)(b)). The margins between
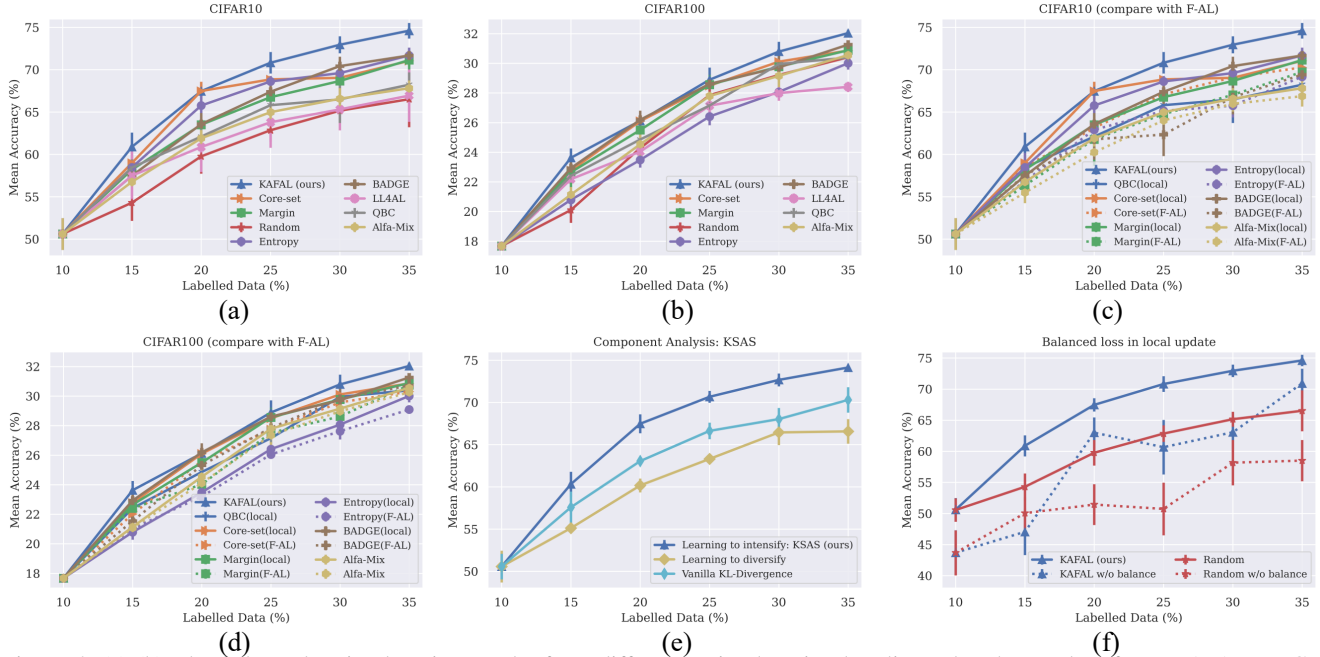
Figure 4. (a)-(b) The federated active learning results from different active learning baselines plus the results of our KAFAL on CI-FAR10/100 with $\alpha = 0.1$. (c)-(d) Comparing our KAFAL with the federated active learning baseline F-AL on CIFAR10/100 with $\alpha = 0.1$. (e) Component analyses of KSAS and KCFU on CIFAR10. (f) Results with balanced loss (Eq. (4)) and standard cross-entropy loss on CIFAR10. For all figures, the error bars show the standard deviation of results across 5 runs.

KAFAL and other baselines become larger with the increase of labelled data. On CIFAR10, our method eventually achieve a margin of around 3% compared to BADGE, Entropy, Margin, and Core-set. LL4AL, although quite competitive in standard active learning, does not perform well in the federated active setting. Besides, LL4AL and QBC update extra model parameters of sizes $0.015M$ and $0.156M$ for each client, when each client's model size for the rest methods is only $0.078M$. On CIFAR100, the margins are less significant compared to results from CIFAR10. Probably because the 10 times of classes in CIFAR100 makes it a much more difficult task, especially consider the limited amount of labelled data for each client. Some of the methods perform poorer than Random, possibly because Random naturally diversifies in sampling. It is worth noting that in CIFAR10 and CIFAR100, the full-set federated learning results are 72.93% and 37.35%. On CIFAR10, the full-set result is lower than our KAFAL result with 35% labelled data. This could be because our strategy selects only the most informative data for annotation and avoids data redundancy. On CIFAR100, the full-set result is around 5% higher than our KAFAL result with 35% labelled data, indicating that 35% data is not enough to represent a 100-class dataset. Notably, we also evaluate each client with the test set and show the analysis in the supplementary (A.1) for a complete picture of the performance of our KAFAL.

4.2. Comparison with Sampling by Global Model

As we mentioned in previous sections, for some sampling methods, it is possible to compute the sampling criteria either on the local client after local updates or on the downloaded aggregated global model. Using the global model for sampling is also the main idea of F-AL [1], a federated active learning method for IID data. Among the baselines we compared with in Sec. 4.1, we found Core-Set, Margin, Entropy, BADGE, and Alfa-Mix to be qualified to compute sampling scores on either the clients or the downloaded global model. We show the experiment results on CIFAR10 and CIFAR100 in Fig. 4(c)(d). The solid lines show the results from using locally updated client models to compute sampling criteria. These are also the results presented in Fig. 4(a). The dashed lines represent the results from using the downloaded global model for computing sampling scores. There is a clear drop in performance moving from client model statistics to global model statistics. Additionally, we compare our method with QBC. Our method combines local and global with discrepancy-based sampling and QBC is a local disagreement-sampling method. This experiment demonstrates the challenge in federated active learning where the sampling aims of the clients mismatch with that of the global model. It also shows that even if we sample informative data points directly using the downloaded global model, the information cannot be fully utilized to benefit the global model through aggregation.

Table 1. Number of rounds of different federated update ways to achieve the same accuracy as running KCFU runs for 15 rounds.

Method	CIFAR10	CIFAR100
Baseline	35	39
KCFU w/o mix	25	27
KCFU	**15**	**15**

F-AL, which is initially proposed for IID federated active learning, does not suit the task of federated active learning with non-IID data.

4.3. Ablation Studies

4.3.1 Component study

To explore the importance of our model components in KAFAL, we separately run experiments to analyze KASA and KCFU. To analyze KSAS, we first replace our knowledge-Specialized KL-Divergence with a vanilla KL-Divergence and observe a 3% to 5% performance drop through the whole sampling process (Fig. 4(c)). We also attempt to sample with a reversed KSAS (learning to diversify), where we replace each $n_{i,y}$ and $n_{i,c}$ in Eq. (1) and (2) with $\frac{1}{n_{i,y}}$ and $\frac{1}{n_{i,c}}$. This prevents the client models from intensifying their specialized knowledge. Instead, it drives the clients to focus on sampling data from rare classes. The results show a significant drop compared to the other two. This further validates our design where each client should intensify its knowledge during active sampling. Data from rare classes can be quite useless in improving the global model on the server.

To analyze the efficiency of KCFU, we count the number of communication rounds of different federated update ways to achieve the same accuracy. The benchmark accuracy is set as the accuracy of running KCFU for 15 rounds. We experiment with the baseline method by replacing KCFU with a vanilla federated update which removes $\mathcal{L}_{compen}$ and updates with $\mathcal{L}_{client}$ (eq. (4)) only. We also compare the results from mixing and not mixing unlabelled data in KCFU. As shown in Tab. 1, KCFU can converge faster than vanilla update no matter mixed data are used, demonstrating the effect of our knowledge-compensatory design which borrows common knowledge from the global model. Mixing data in KCFU further boosts the efficiency. We further experiment by fixing $\mathbf{\Gamma}(\tilde{x}') = \frac{1}{C}$, where C is the number of classes, for $\mathcal{L}_{compen}$ (Eq. (5)). This means we distil knowledge from the downloaded global model without differentiation on all unlabelled data. Unsurprisingly, the performance is very poor. The accuracy reaches only 40.4% with 10% data and the setting aligned with Fig. 4(a).

4.3.2 Local update without the balanced loss

We use a balanced loss (Eq. (4)) for local update of clients. This type of loss is usually the cherry on the top for standard federated learning. This is however not the case in our federated active learning problem. In Fig. 4(f), we show the results using balanced loss (Eq. (4)) and a simple cross-entropy loss (simply replacing $n_{i,y}$ and $n_{i,c}$ in Eq. (4) with 1). We ran the experiment with two methods, our KAFAL and random sampling. From the experiment results, we can see that removing the balanced loss in local update disturbs or almost ruins the learning, a drop of 5% to 10% in performance occurs. Our KAFAL still outperforms random sampling, but the results are highly unstable. This is somewhat foreseeable since each client model starts with a very small amount of data in federated active learning. Despite our learning to intensify on specialized knowledge during sampling, it is still crucial to handle the imbalance of data during local client update using the balanced loss.

4.3.3 Knowledge specialization alternatives

It is an interesting question whether other reweighting techniques can also help achieve knowledge specialization in federated active learning. Here we compare our method with two knowledge specialization alternatives, probability-level specialization and KL-Divergence-level specialization. Results and detailed analyses are presented in the supplementary (A.2). The experimental results show that KAFAL outperforms both of the alternative methods. While probability-level specialization yields an acceptable outcome, KL-Divergence-level specialization fails to produce a reasonable result. One possible reason for this difference is that the probability-level specialization method, like our KAFAL, uses a moderate level of reweighting to adjust the results. In contrast, the KL-Divergence-level specialization method directly reweights the summation in the KL-Divergence calculation, potentially resulting in a level of reweighting that is too strong.

4.3.4 Different non-IID levels

We further explore federated active learning with the non-IID coefficient $\alpha = 0.3$ and $\alpha = 1$ on CIFAR10. We show results and detailed analysis in the supplementary (A.3). A larger α value provides less non-IID distributions for clients, i.e., the distributions across different clients are more similar. Unsurprisingly, compared to our CIFAR10 with $\alpha = 0.1$ results, the results are overall better for $\alpha = 0.3$ and $\alpha = 1$. Our KAFAL is still state-of-the-art for $\alpha = 0.3$ and $\alpha = 1$, but the margins between the results of KAFAL and the rest methods are relatively smaller. This experiment demonstrates that our KAFAL is more competitive with higher levels of non-IID. It validates that intensi-

fying knowledge-specialized data in KAFAL can handle the non-IID distributed data in federated active learning.

4.3.5 Different λ values

The coefficient λ in eq. (1)(2) controls the knowledge-specialized level in KSAS. With larger values of λ, the clients intensify more on their specialized knowledge in active sampling. As we stated, we simply use $\lambda = 1$ in our main experiments. Here we explore more values of λ on CIFAR10 and show the results and detailed analysis in the supplementary (A.4). For λ of values 1, 2, and 3, the difference is not significant. However, the results are clearly poorer for more extreme λ values 0.1 and 10. Therefore, when applying KAFAL, the selection of λ value can be flexible, but the chosen value should be neither too small nor too large.

4.3.6 Learning with more decentralized clients

In previous sections, we explored federated active learning with $N = 10$ clients. To better analyze the problem, we run experiments on CIFAR10 with $N = 20$ and $N = 100$. The labelled data amount still starts with 10% of each training set, meaning that the local dataset on each client is smaller in size. The results and detailed analysis are shown in the supplementary (A.5). Compared with results from using $N = 10$ clients, results for all methods reduce due to the smaller local datasets. Our KAFAL still outperforms the rest methods by a clear margin.

4.3.7 A smaller ratio of clients to update per round

We used $R = 80\%$ in previous experiments. To test how our KAFAL performs with a smaller ratio of clients updated in each communication round, we use $R = 40\%$, meaning that only 40% of the clients are updated in each communication round. Surprisingly, our KAFAL performs even better using $R = 40\%$ compared with using $R = 80\%$, while results from the rest methods all drop. This is possibly because our KAFAL compensates for the knowledge of clients with the global model using KCFU along with actively sampling data by intensifying specialized knowledge using KSAS. The two together enable a faster convergence in global aggregations. Using $R = 40\%$ means each client is trained less compared to using $R = 80\%$ when the communication rounds T is fixed. The rest methods which still actively sample harder data that are likely from less frequent classes cannot utilize these data in training with the smaller R value. Although KCFU is also used for other methods for a fair comparison, it cannot be fully utilized without the knowledge-specialized intensification of KSAS. Detailed results and analyses are shown in the supplementary (A.6).
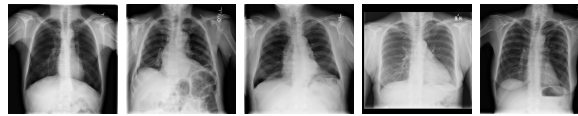

Figure 5. Selected images in NIH Chest X-Ray dataset.

4.4. Medical Image Classification

We further conduct experiments in a more realistic scenario of X-ray image classification using NIH Chest X-Ray dataset [46]. Some examples are shown in Fig. 5. The task is to categorize thorax diseases using chest X-ray images. The dataset consists of more than 112k images of size 1024×1024. We follow the official training and testing splits. And we exclude images tagged with 'no findings'. The rest data have 14 for different thorax diseases as labels. The training split includes 36024 images and the testing split includes 15735 images. We use ResNet-50 [15] as the backbone of the clients and the global model. We still use $\alpha = 0.1$ as the non-IID coefficient to distribute the client data. 5 clients are used, and 80% are selected for the update at each communication round. We start with 10% labels and use 5% of the whole dataset as the budget. The results are shown in the supplementary (A.7). We compare with four baseline methods (Random, Core-Set, Entropy, and Margin) that the dataset can easily fit in considering the image size and model size. Our KAFAL still achieves state-of-the-art results on this dataset.

5. Conclusion

We have introduced a federated active learning paradigm which allows actively selecting the unlabelled data to efficiently learn a global model given a limited annotation budget in a decentralized learning process. We revealed that the main challenge in federated active learning is the mismatch between the active sampling goals of the global model on the server and each local client due to model differences caused by non-IID data distributions. This paper devised a Knowledge-Aware Federated Active Learning (KAFAL) method for federated active learning with non-IID data. KAFAL computes the discrepancies between client-server models with an intensification on each client's specialized knowledge. It is worth noting that the intensifying process is particularly important to achieve a powerful global model in the non-IID federated learning framework. Moreover, KAFAL also compensates for each client's ability in rare classes to handle data heterogeneity caused by non-IID data during federated updates. Extensive experiments and analyses have validated the superiority of KAFAL over the state-of-the-art active learning methods under the federated active learning framework.

References

[1] Jin-Hyun Ahn, Kyungsang Kim, Jeongwan Koh, and Quanzheng Li. Federated active learning (f-al): an efficient annotation strategy for federated learning. *arXiv preprint arXiv:2202.00195*, 2022. 2, 7

[2] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *ICLR*, 2020. 1, 2, 6

[3] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *CVPR*, 2018. 1, 2

[4] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2021. 2

[5] Xu Chen and Brett Wujek. Autodal: Distributed active learning with automatic hyperparameter selection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3537–3544, 2020. 3

[6] Corinna Cortes, Giulia DeSalvo, Mehryar Mohri, Ningshan Zhang, and Claudio Gentile. Active learning with disagreement graphs. In *ICML*, 2019. 1, 3

[7] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995. 3, 6

[8] Sayna Ebrahimi, William Gan, Dian Chen, Giscard Biamby, Kamyar Salahi, Michael Laielli, Shizhan Zhu, and Trevor Darrell. Minimax active learning. *arXiv preprint arXiv:2012.10467*, 2020. 1, 3

[9] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Information, prediction, and query by committee. In *NIPS*, 1993. 3

[10] Bo Fu, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. Transferable query selection for active domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7272–7281, 2021. 3

[11] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, pages 510–526. Springer, 2020. 3

[12] Jack Goetz, Kshitiz Malik, Duc Bui, Seungwhan Moon, Honglei Liu, and Anuj Kumar. Active federated learning. *arXiv preprint arXiv:1909.12641*, 2019. 3

[13] Xuan Gong, Abhishek Sharma, Srikrishna Karanam, Ziyan Wu, Terrence Chen, David Doermann, and Arun Innanje. Ensemble attention distillation for privacy-preserving federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15076–15086, 2021. 2

[14] Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. Deep active learning for biased datasets via fisher kernel self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9041–9049, 2020. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 9

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 6

[17] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020. 2

[18] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 2, 5, 6

[19] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution. In *European Conference on Computer Vision*, pages 76–92. Springer, 2020. 2

[20] Siyu Huang, Tianyang Wang, Haoyi Xiong, Jun Huan, and Dejing Dou. Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3447–3456, 2021. 3

[21] Harold Jeffreys. Theory of probability. 1939. 4

[22] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. Federated semi-supervised learning with inter-client consistency & disjoint learning. In *International Conference on Learning Representations*, 2020. 2

[23] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *CVPR*, 2021. 1, 2

[24] SangMook Kim, SangMin Bae, Se-Young Yun, and Hwanjun Song. Lg-fal: Federated active learning strategy using local and global models. 2

[25] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. 1, 2

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[27] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

[28] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010. 6

[29] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 2

[30] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020. 2

[31] Nan Lu, Zhao Wang, Xiaoxiao Li, Gang Niu, Qi Dou, and Masashi Sugiyama. Unsupervised federated learning is possible. In *International Conference on Learning Representations*, 2021. 2

[32] Xinhong Ma, Junyu Gao, and Changsheng Xu. Active universal domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8968–8977, 2021. 3

[33] Rafid Mahmood, Sanja Fidler, and Marc T Law. Low budget active learning via wasserstein distance: An integer programming approach. *arXiv preprint arXiv:2106.02968*, 2021. 3

[34] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[35] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 5

[36] Prem Melville and Raymond J Mooney. Diverse ensembles for active learning. In *ICML*, page 74, 2004. 3

[37] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019. 2

[38] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12237–12246, 2022. 2, 6

[39] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[40] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2019. 2

[41] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in Neural Information Processing Systems*, 33:4175–4186, 2020. 5

[42] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 1, 2, 6

[43] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992. 1, 3

[44] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, 2019. 1, 2, 3

[45] Shuo Wang, Yuexiang Li, Kai Ma, Ruhui Ma, Haibing Guan, and Yefeng Zheng. Dual adversarial network for deep active learning. 2020. 1, 2, 3

[46] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, M Bagheri, and R Summers. Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE CVPR*, volume 7, 2017. 6, 9

[47] Zhiguo Wang, Xintong Wang, Ruoyu Sun, and Tsung-Hui Chang. Federated semi-supervised learning with class distribution mismatch. *arXiv preprint arXiv:2111.00010*, 2021. 2

[48] Chun-Han Yao, Boqing Gong, Hang Qi, Yin Cui, Yukun Zhu, and Ming-Hsuan Yang. Federated multi-target domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1424–1433, 2022. 2

[49] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *CVPR*, 2019. 1, 2, 3, 6

[50] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pages 12073–12086. PMLR, 2021. 2

[51] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qingming Huang. State-relabeling adversarial active learning. In *CVPR*, 2020. 1, 2, 3

[52] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. 2