Increasingly stringent data privacy regulations limit the development of person re-identification (ReID) because person ReID training requires centralizing an enormous amount of data that contains sensitive personal information. To address this problem, we introduce federated person re-identification (FedReID)—implementing federated learning, an emerging distributed training method, to person ReID. FedReID preserves data privacy by aggregating model updates, instead of raw data, from clients to a central server. Furthermore, we optimize the performance of FedReID under statistical heterogeneity via benchmark analysis. We first construct a benchmark with an enhanced algorithm, two architectures, and nine person ReID datasets with large variances to simulate the real-world statistical heterogeneity. The benchmark results present insights and bottlenecks of FedReID under statistical heterogeneity, including challenges in convergence and poor performance on datasets with large volumes. Based on these insights, we propose three optimization approaches: (1) we adopt knowledge distillation to facilitate the convergence of FedReID by better transferring knowledge from clients to the server, (2) we introduce client clustering to improve the performance of large datasets by aggregating clients with similar data distributions, and (3) we propose cosine distance weight to elevate performance by dynamically updating the weights for aggregation depending on how well models are trained in clients. Extensive experiments demonstrate that these approaches achieve satisfying convergence with much better performance on all datasets. We believe that FedReID will shed light on implementing and optimizing federated learning on more computer vision applications.

CCS Concepts: • Computing methodologies → Distributed algorithms; Object identification; Matching; • Information systems → Top-k retrieval in databases;

Additional Key Words and Phrases: Federated learning, person re-identification
1 INTRODUCTION

Person re-identification (ReID) aims to match the same person appearing in disjoint camera views. It has received considerable attention because of its wide applications in business and public security, such as customer trajectory analysis and criminal investigation [22]. Person ReID has achieved outstanding performance [4, 45, 54], attributed to the advances of deep neural networks (DNNs) [13, 21].

However, the increasing concerns of data privacy protection limit the development of person ReID [8]. DNN-based approaches are data-hungry, relying on centralizing a sizable amount of data to achieve high performance [58]. Training images of person ReID contain sensitive personal information, which could reveal the identity and location of individuals. Centralizing these images imposes potential privacy leakage risks. Hence, it is crucial to navigate the development of person ReID under the premise of privacy protection.

Federated learning (FL), an emerging distributed training technique, has empowered many applications with privacy-preserving mechanisms [18], such as healthcare applications [5, 41] and consumer products [36, 37]. FL preserves data privacy by training models collectively with decentralized clients. These clients, instead of transferring raw data, only transfer training updates to a central server. This reduces privacy leakage risks, as raw data are kept locally. Despite the advantages of FL, implementing FL to person ReID and optimizing its performance are largely overlooked; such implementation possibility is only mentioned in the work of Hao et al. [12], but their study does not present dataset or benchmark results.

In this work, we propose Federated Person Re-identification (FedReID), a new person ReID training paradigm to enable multimedia researchers to train models with privacy guaranteed. Besides privacy protection, FedReID possesses other advantages: reducing communication overhead of uploading plenty amount of data [35], adapting models in clients to local scenes, and obtaining a holistic model that generalizes in diverse scenarios. A usage example of FedReID is video surveillance across communities or districts, where multiple entities collaborate to learn a generalized model without revealing their private video surveillance data.

However, implementing FL to person ReID is not trivial—statistical heterogeneity is a major challenge of FedReID in real-world scenarios [25]: (1) data is in non-identical and independent distribution (non-IID) [56] because data collected from different cameras could have significant discrepancies in resolution, illumination, and angles, and (2) data volume is unbalanced with varied pedestrian flow in different locations. Although some studies illustrate that non-IID harms the training convergence and model performance in tasks like image classification [56], the impact of statistical heterogeneity on FedReID has not yet been explored.

This work aims to optimize FedReID under statistical heterogeneity via benchmark analysis. We start by constructing a new benchmark, FedReIDBench, with nine representative ReID datasets and a specially designed algorithm for FedReID (Section 3). In the benchmark, a server coordinates nine clients (each containing a dataset) to conduct training on their local data and aggregates training updates iteratively. We then conduct benchmark analysis (Section 4), revealing that statistical heterogeneity leads to performance degradation and difficulty in convergence. We end by proposing three performance optimization methods: client clustering (CC) (Section 5.1) and dynamic weight adjustment (Section 5.3) to elevate performance, and knowledge distillation (KD)
(Section 5.2) to facilitate convergence. Specifically, CC groups clients with similar data distribution and aggregate training updates within each group. KD uses a public dataset to transfer knowledge from clients to the server more effectively. In addition, weight adjustment dynamically updates the weights of clients’ training updates in server aggregation. Extensive experiments demonstrate the effectiveness of the benchmark and the significance of the optimization approaches. We believe that FedReID will shed light on implementing and optimizing FL to more computer vision applications.

In summary, we make the following contributions:

- We construct a new benchmark for FedReID, simulating real-world scenarios of statistical heterogeneity with nine representative person ReID datasets.
- We provide useful insights and investigate potential bottlenecks of FedReID by analyzing the benchmark results.
- We propose three performance optimization methods: KD to facilitate convergence, as well as CC and dynamic weight adjustment to elevate performance.
- We extensively evaluate these optimization methods to demonstrate their effectiveness.

2 RELATED WORKS

2.1 Person ReID

The objective of person ReID is to retrieve the identity of interest from disjoint camera views. It is an important computer vision task that is widely applied in public security, such as video surveillance [58]. The advances of the DNN have greatly improved the performance of person ReID by learning better feature representations compared to traditional hand-crafted feature operations [29, 31, 46, 52]. Over the years of development, the community has constructed many person ReID datasets [15, 28, 50, 57, 59]. These datasets are collected from various locations with different camera views. The majority of person ReID studies focus on extracting better feature representations by improving the architecture of DNNs [22, 54]. They rely on the assumption that data, collected from different cameras in various locations, can be centralized to a central server. However, centralizing plenty of images of individuals raises potential privacy leakage risks. Different from previous approaches, we proposed FedReID—a new training paradigm for ReID to learn ReID models from decentralized data. FedReID mitigates potential privacy leakage issues, as data is not transferred to a central server.

2.2 Federated Learning

FL is an emerging distributed training technique that trains models with decentralized clients coordinated by a central server [18].

Benchmarks. To facilitate the development of FL, researchers have published several benchmarks and datasets: LEAF [3] is the first benchmark for FL research, containing federated datasets for image classification and natural language processing tasks; Streets [34] is a real-world image dataset collected from street cameras for object detection; and OARF [17] is a benchmark that aims to facilitate a wide range of FL applications, such as trend prediction, recommendation, and sentiment analysis. However, different from these tasks, person ReID is a retrieval task in which no existing benchmark contains related datasets. In this work, we construct a new FL benchmark that simulates real-world scenarios of FedReID.

Algorithm. The most known algorithm for FL is **Federated Averaging (FedAvg)** [35]. It defines an iterative training process in which clients send trained local models to a server and the server sends back the aggregated global model to clients. The benchmarks mentioned previously adopt FedAvg as the standard algorithm. However, FedAvg requires all clients to have identical models.
Table 1. Statistics of Nine Datasets in Our Proposed Benchmark, FedReIDBench

| Datasets         | # Cameras | #I Ds  | # Images | #I Ds  | # Images | # I Ds | # Images |
|------------------|-----------|--------|----------|--------|----------|--------|----------|
| MSMT17 [50]      | 15        | 1,041  | 32,621   | 3,060  | 11,659   | 3,060  | 82,161   |
| DukeMTMC-reID [59]| 8         | 702    | 16,522   | 702    | 2,228    | 1,110  | 17,611   |
| Market-1501 [57] | 6         | 751    | 12,936   | 750    | 3,368    | 751    | 19,732   |
| CUHK03-NP [30]   | 2         | 767    | 7,365    | 700    | 1,400    | 700    | 5,332    |
| PRID2011 [15]    | 2         | 285    | 3,744    | 100    | 100      | 649    | 649      |
| CUHK01 [28]      | 2         | 485    | 1,940    | 486    | 972      | 486    | 972      |
| VIPeR [11]       | 2         | 316    | 632      | 316    | 316      | 316    | 316      |
| 3DPeS [2]        | 2         | 93     | 450      | 86     | 246      | 100    | 316      |
| iLIDS-VID [49]   | 2         | 59     | 248      | 60     | 98       | 60     | 130      |

Note: These datasets have large variances in data volume, decreasing from top to bottom.

It is not suitable for FedReID because clients could have varied classifiers. Therefore, we propose an enhanced algorithm, **Federated Partial Averaging (FedPav)**.

Statistical heterogeneity. Statistical heterogeneity—non-IID and unbalanced data—is a major challenge of FL [18, 25]. In traditional distributed training [9, 44], data in multiple nodes of cloud clusters are IID. Data in multiple FL clients, however, could be heterogeneous. To address this challenge, some studies focus on optimizing training in clients [1, 19, 24, 26, 55], whereas recent work [55] requires extra communication by sharing features among clients. However, some studies optimize the aggregation process in the server [47, 48, 61, 62]. In addition, several studies share voluntary or public data between the server and clients [53, 56]. These methods are validated on small datasets [3, 7, 20], and thus may not be directly applicable to the challenging scenario of FedReID. In this work, we introduce three optimization methods targeting the statistical heterogeneity of FedReID via in-depth benchmark analysis.

This work is an extension of our previous conference version [63]. The main improvements are as follows: (1) we introduce a new performance optimization method—CC; (2) we integrate CC with the previously proposed weight adjustment method, achieving the best performance; (3) we conduct more performance evaluations for comparison with the benchmark results and the proposed optimization methods; and (4) we provide more comprehensive descriptions for the proposed optimization methods. Despite that another work [51] also studied FedReID after our conference work [63], it focuses more on adapting to unseen domains, whereas we aim to address the statistical heterogeneity revealed from our benchmark analysis.

### 3 FEDERATED PERSON REID BENCHMARK

This section introduces a new FL benchmark for person ReID, FedReIDBench. This benchmark comprises nine representative datasets, two possible implementation architectures, one enhanced algorithm, and several performance evaluation metrics.

#### 3.1 Datasets

We construct the benchmark dataset with nine representative person ReID datasets as shown in Table 1. It contains a total of 224,064 images of 17,991 identities. These datasets are collected at multiple locations (or countries) and published by different organizations at different times. They not only vary in the number of images, identities, and camera views but also differ in image resolution, illumination, and scenes.

The variances in these datasets simulate the statistical heterogeneity in real-world scenarios: the disparity of data volumes represents the unbalanced data problem, and the domain

ACM Trans. Multimedia Comput. Commun. Appl., Vol. 19, No. 1s, Article 38. Publication date: January 2023.
discrepancies among datasets represent the non-IID problem. Unlike centralized training where data is IID, statistical heterogeneity makes training even more challenging.

3.2 Architectures

Figure 1(a), (b) illustrate two architectures for possible implementation scenarios of FedReID: edge-cloud architecture and device-edge-cloud architecture. In both architectures, the cloud represents the central server connecting to multiple edges.

Edge-cloud architecture. In this architecture, cameras are the edges that directly connect with the server to conduct FL. The server coordinates these cameras to train models with locally collected images. This architecture significantly reduces privacy leakage risks, as the data always stays at the edges. However, deployment of this architecture requires cameras to have enough computation power and storage capability. A real-world application of this architecture would be video surveillance for a community with multiple cameras on different streets.

Device-edge-cloud architecture. This is a three-layer hierarchical architecture. Edge servers are in the middle layer. On the one hand, they construct local training datasets by gathering images from multiple camera views, which is similar to how datasets in the benchmark are collected. On the other hand, edge servers collaboratively perform FL with their local datasets under the coordination of the server. A good illustration of this architecture would be multiple communities collaborating to learn person ReID models, where each community has an edge server collecting data from multiple cameras.

3.3 Algorithm

The standard FL algorithm FedAvg [35] is not suitable for FedReID because it requires identical model structures in all clients. The model structure of the benchmark is ID-discriminative embedding (IDE) [58], which is a common baseline for DNN-based person ReID. This model structure consists of a backbone and a classifier: the backbone is ResNet-50 [13] in our FedReIDBench; the classifier is a linear layer whose dimension depends on the number of identities of clients. Since
Algorithm 1: Federated Partial Averaging (FedPav)

Input: Total data volume \( n \), client \( k \)'s data volume \( n_k \), local epoch \( E \), batch size \( B \)
Output: Global model \( w^T \), local models of each client \( w^T_k \)

1. Server:
   - initialize \( w^0 \);
   - for each round \( t = 0 \) to \( T-1 \) do
     1. \( S_t \leftarrow \) (random set of \( K \) clients selected from \( N \) clients);
     2. for each client \( k \in S_t \) in parallel do
        1. \( w_{k}^{t+1} \leftarrow \) Client\( (w^t, k, t) \);
     3. \( w^{t+1} \leftarrow \sum_{k \in C_t} \frac{n_k}{\sum_{k \in C_t} n_k} w_{k}^{t+1} \);
   - return best global model \( w^T \), best client \( k \)'s local model \( w^T_k \);

2. Client \( (w, k, t) \):
   - \( v \leftarrow \) (initialize classifier \( v \) if \( t = 0 \), otherwise retrieve it from local store);
   - for each epoch \( e = 0 \) to \( E-1 \) do
     1. \( (w, v) \leftarrow (w, v) - \eta \nabla L((w, v); b) \);
   - store \( v \);
   - return \( w \);

the number of identities could vary among clients, their classifiers could differ in clients. Hence, we adopt the enhanced algorithm for FedReID: FedPav [63].

FedPav allows models in clients to be only partially identical. For FedReID, FedPav enables clients to use the same backbone but different identity classifiers for FL, as shown in Figure 1(c). The training process is similar to FedAvg except that clients only transfer the identical part of models to the central server for aggregation.

Algorithm 1 summarizes FedPav. We aim to obtain a holistic global model and personalized local models for clients at the end of the training. Each training round \( t \) of FedPav contains four steps: (1) distribution: the central server chooses a fraction (\( K \) out of \( N \)) of clients for current round of training and distributes the global model \( w^t \) to these clients; (2) local training: each client \( k \) initializes the backbone \( w^t_k \) using the global model parameters and trains the model with a local dataset for \( E \) local epochs with \( B \) batch size; (3) upload: each client \( k \) uploads the trained backbone \( w^{t+1}_k \) to the server; and (4) aggregation: the server generates a new global model \( w^{t+1} \) by aggregating updates from clients with weighted average. The training stops after iterating these four steps for \( T \) rounds. After training, we use the global model \( w \) to evaluate convergence and generalization, and use local models \( w_k \) to evaluate how well models adapt to local scenarios.

3.4 Performance Evaluation Metrics

We evaluate FedReID in two aspects: accuracy and communication cost.

Accuracy. The cumulative matching characteristics curve and mean Average Precision (mAP) [58] are standard person ReID evaluation metrics. Given an image as a query, person ReID matches it in a gallery of images based on similarity. Cumulative matching characteristics measures the probability that the query identity is in the top-\( k \) most similar matched gallery images. We consider \( k = \{1, 5, 10\} \) in the benchmark, representing the rank-1 accuracy, rank-5 accuracy, and rank-10 accuracy. In addition, we report the mAP of all queries.

Communication cost. Since FL requires iterative communication between a server and multiple clients, we also consider the communication costs. The total communication cost is \( T \times 2 \times M \), where \( T \) is the number of communication rounds and \( M \) is the transmission message size (model size). \( 2 \times M \) is the communication cost of each round, considering both uploading and downloading from clients.
3.5 Reference Implementation

To facilitate ease of use and reproducibility, we open source referenced implementation in GitHub.\(^1\) It includes data preprocessing, proposed algorithm, and optimization methods. We plan to integrate it to EasyFL \([60]\) in the future. In addition, we provide the experimental settings as follows.

Learning rate. The initialized learning rates were different for the identity classifier and the backbone: 0.05 for the identity classifier and 0.005 for the backbone. The learning rate schedulers of both are the same with step size 40 and gamma 0.1. In addition, the learning rate for the server fine-tuning in KD is 0.0005.

Optimizer. We use Stochastic Gradient Decent (SGD) as the optimizer. The optimizer is set with weight decay $5e^{-4}$ and momentum 0.9.

FL settings. The default settings of FL algorithms are as follows: batch size $B = 32$, local epoch $E = 1$, and total training rounds $T = 300$.

4 BENCHMARK ANALYSIS

In this section, we present the results of extensive experiments on the benchmark. We investigate the performance of two architectures, the impact of different federated settings, and the impact of statistical heterogeneity.

We initialize the backbone with ResNet-50 \([13]\) parameters pre-trained on ImageNet \([10]\). For hyperparameters, we use batch size $B = 32$ and local epoch $E = 1$ to train $T = 300$ communication rounds by default.

4.1 Edge-Cloud Architecture

In the edge-cloud architecture, each camera is a client. Since each person ReID dataset contains data from several camera views, we simulate FedReID in this architecture by assigning data of the same camera view to one client. As a dataset is divided into several clients by camera views, we term it the federated-by-camera scenario.

To understand FedReID performance in the federated-by-camera scenario, we compare it with two other settings. The first is the federated-by-identity scenario: we divide one dataset into partitions for multiple clients, where each client includes one partition that contains an equal number of identities. The number of clients equals the number of camera views. The second is centralized training: training with data merged from multiple cameras, which can be considered as the upper bound. For example, the Market-1501 dataset \([57]\) contains six camera views with 751 identities. In the federated-by-identity scenario, we divide it into six clients, where each client includes 125 non-overlapping identities. The centralized training means training with the Market-1501 dataset.

Table 2 presents the comparisons of global models of different settings on two datasets: CUHK03-NP \([30]\) and Market-1501 \([57]\). Compared with the federated-by-identity scenario or centralized training, the federated-by-camera scenario performs much worse. This indicates that learning from only one camera view is infeasible to obtain a generalized model in person ReID, where the evaluation is based on images from multiple camera views. Hence, even though industrial cameras have enough computation and storage capacity to support edge-cloud architecture, the device-edge-cloud architecture could be more adequate for FedReID because each client learns cross-camera knowledge. All other experiments in the article are conducted based on the device-edge-cloud architecture.

\(^1\)https://github.com/cap-ntu/FedReID.
Table 2. Performance Comparison of the Federated-by-Camera Scenario, Federated-by-Identity Scenario, and Centralized Training on CUHK03-NP and Market-1501 Datasets

| Dataset     | # Clients | Settings        | Rank-1 | Rank-5 | Rank-10 | mAP  |
|-------------|-----------|-----------------|--------|--------|---------|------|
| CUHK03-NP   | 2         | Federated-by-camera | 11.21  | 19.14  | 25.71   | 11.11|
|             |           | Federated-by-identity    | **51.71** | **69.50** | **76.79** | **47.39** |
|             |           | Centralized training    | 49.29  | 68.86  | 76.57   | 44.52|
| Market-1501 | 6         | Federated-by-camera     | 61.13  | 74.88  | 80.55   | 36.57|
|             |           | Federated-by-identity    | 85.69  | 93.44  | 95.81   | 66.36|
|             |           | Centralized training    | **88.93** | **95.34** | **96.88** | **72.62**|

Note: The federated-by-camera scenario achieves the worst performance, indicating that edge-cloud architecture could be inadequate for FedReID.

4.2 Device-Edge-Cloud Architecture

In the device-edge-cloud architecture, edge servers collect data from multiple cameras and conduct FedReID with a central server. Since each of the benchmark datasets consists of data from multiple camera views, we simulate this scenario with nine clients—each client contains one unique dataset of the benchmark datasets. In all experiments, we choose nine clients to participate in training.

Under this architecture, we consider two types of models produced from FedReID training: (1) *local model*: the specialized models trained after \( E \) local epochs in clients before uploading to the server in each training round, and (2) *global model*: the generalized model obtained in the server by aggregating models uploaded from clients.

To understand the performance of FedReID, we compare global and local models with the other two models: (1) *standalone training*: the models trained in clients with their own dataset (without participating in FL), and (2) *centralized training*: the model trained using the combination of all benchmark datasets, simulating conventional person ReID training that centralizes datasets. Centralized training can be treated as the upper bound of FedReID, whereas FedReID is meaningful for a client only when the performance of global or local models is better than standalone training.

4.2.1 Impact of Federated Settings. We first investigate the performance of FedReID (the global model) using the FedPav algorithm under different federated settings, including batch size \( B \) and local epochs \( E \).

Batch size reflects the trade-off between computation power consumption and model accuracy. With the same local epochs, a larger batch size reduces computation time because the training can better take advantage of the parallelism provided by the client hardware. (Computation is fully utilized as long as \( B \) is large enough.) Figure 2(a) compares the rank-1 accuracy of FedPav using different batch sizes \( B = \{32, 64, 128\} \), under the setting that local epochs \( E = 1 \) and communication rounds \( T = 300 \). Smaller batch size generally achieves better performance in most datasets but consumes higher computation.

Local epochs reflect the trade-off between the communication cost and model accuracy. The total training epochs \( E_{\text{total}} \) can be calculated with \( E_{\text{total}} = T \times E \), where \( T \) is the communication rounds and \( E \) is the number of local epochs. By fixing the total training epochs for a fair comparison, smaller \( E \) means larger communication rounds \( T \), requiring higher communication costs. In addition, we compare the rank-1 accuracy of different numbers of local epochs in Figure 2(b). Despite that \( E = 5 \) performs worse than \( E = 10 \) in several datasets, smaller numbers of local epochs \( E \) generally result in better performance. The smallest number of local epoch \( E = 1 \) achieves much
Optimizing Performance of FedReID: Benchmarking and Analysis

Fig. 2. Performance comparison of different batch sizes (a) and different local epochs (b). Batch size $B = 32$ and local epoch $E = 1$ achieve better overall performance than other settings. We ran a total of 300 epochs for these experiments.

Table 3. Rank-1 Accuracy Comparison of the Global Model and Local Models Obtained from FedReID, Standalone Training, and Centralized Training

| Method          | MSMT17 | DukeMTMC | Market | CUHK03 | PRID2011 | CUHK01 | VIPeR | 3DPeS | iLIDS-VID |
|-----------------|--------|----------|--------|--------|----------|--------|-------|-------|-----------|
| Centralized training | 54.6   | 84.2     | 91.7   | 64.0   | 80.0     | 89.7   | 65.5  | 82.1  | 80.6      |
| Standalone training   | 49.6   | 80.1     | 88.9   | 49.3   | 55.0     | 69.0   | 27.5  | 65.4  | 52.0      |
| Global model         | 41.0   | 74.3     | 83.4   | 31.7   | 37.7     | 73.4   | 48.1  | 69.2  | 79.9      |
| Local model          | 48.3   | 78.1     | 83.6   | 39.5   | 50.7     | 80.7   | 52.0  | 80.6  | 84.7      |

Note: FedReID effectively improves the performance on small datasets. However, it performs worse than standalone training on large datasets due to statistical heterogeneity. We ran the experiment with $B = 32$ and $E = 1$.

better performance than $E = 5$ and $E = 10$ in all datasets, and it requires the highest communication cost, indicating the trade-off between communication costs and model accuracy.

4.2.2 Impact of Statistical Heterogeneity. The statistical heterogeneity hinders the convergence and performance of FedReID. Specifically, non-IID causes difficulty in convergence, and both non-IID and unbalanced data limit the performance of FedReID.

Later, Figure 4 shows that FedPav does not converge well, as the accuracy (of the global model) fluctuates throughout training. We argue that it is mainly due to non-IID data of nine clients. As datasets in clients have domain discrepancy (e.g., illumination, resolution, scenes), aggregating them simply by weighted average leads to unstable and unpredictable results. As a result, it causes difficulty in selecting a representative global model for other scenarios. We report the accuracy by averaging the three best global models throughout training, evaluated every 10 rounds.

Furthermore, Table 3 compares the performance of the global and local models obtained from FedReID with standalone and centralized training. The results are twofold: on the one hand, standalone training outperforms both the global and local models in large datasets such as DukeMTMC-reID [59] and CUHK03-NP [30]; on the other hand, both the global and local models outperform standalone training in small datasets such as VIPeR [11] and 3DPeS [2], and even outperform centralized training in the iLIDS-VID dataset [49]. These results indicate that although clients with larger datasets do not benefit from FedReID, the ones with smaller datasets gain significant improvement. We interpret the results from two perspectives: (1) for clients with large datasets, they dominate in server aggregation as the weights for aggregation are positively correlated with data volumes, causing less gain from others; (2) for clients with small datasets, they learn from other clients more effectively because their models are not well trained.

Another observation from Table 3 is that local models outperform the global model in all datasets. As the global model is produced by aggregating local models, we argue that non-IID data causes performance degradation in the server aggregation. Better aggregation methods can be considered to better transfer knowledge from local models to the global model.

ACM Trans. Multimedia Comput. Commun. Appl., Vol. 19, No. 1s, Article 38. Publication date: January 2023.
5 PERFORMANCE OPTIMIZATION

In this section, we first propose three methods to address the problems caused by statistical heterogeneity: CC, KD, and dynamic weight adjustment. Then, we present experimental results of these optimization methods compared with standalone training and the benchmark results.

5.1 Client Clustering

To tackle the performance degradation caused by non-IID data in server aggregation of all clients, we propose to aggregate clients with similar data distributions. As discussed in Section 4.2.2, local models outperform the global model in all datasets. The global model is obtained by aggregating local models, so the performance drop mainly sources from the aggregation of clients with diverse data distributions. To tackle this problem, we propose CC to split clients into several groups based on their data distributions and aggregate models within each group in the server.

Figure 3(a) depicts the process of CC with the following steps: (1) we extract features $f_k$ from one batch data (32 samples) of a public person ReID dataset\(^2\) using the trained model $w_k$ from client $k$; (2) we adopt a clustering algorithm to cluster these features into multiple groups; (3) we aggregate models of clients within each group, obtaining a global model in each group; and (4) we use the global model of each group to update local models of clients within that group for the next training round. In Figure 3(a), we cluster clients into two groups: one group contains clients $\{1, 4\}$ and another one contains clients $\{2, 3, 5\}$, based on their features $f$. Then, we aggregate $w_1$ and $w_4$ to obtain global model $w_{c1}$, and we aggregate $w_2$, $w_3$, and $w_5$ to obtain $w_{c2}$. At the start of next training round, we update local models of clients $\{1, 4\}$ with $w_{c1}$ and local models of clients $\{2, 3, 5\}$ with $w_{c2}$. CC obtains multiple global models after training, so we focus on evaluating personalized local models $w_k$ of each client $k$.

In this way, we use the features as a proxy to measure the similarity of data distributions among clients. The intuition behind CC is that the clients clustered into the same group share more similar data distributions. The choice of the clustering algorithm is important for the overall performance of this method. We utilize a hierarchical clustering algorithm, FINCH [39], to cluster clients based on similarities of features extracted from their models. Regarding each client as a cluster at the start, we group the clients that are first neighbors; two clients are first neighbors if their features have the shortest distance (cosine similarity) or they share the same first neighbor. FINCH merges first neighbors in each clustering step. In our scenario, since nine clients would be merged into one cluster after two to three clustering steps, we only cluster for one step per communication round. As a result, the server would have two to three clusters, where each cluster contains two to seven clients. FINCH is able to deliver good clustering results without prior knowledge of the targeted number of clusters.

5.2 Knowledge Distillation

Besides CC, we adopt KD to elevate performance and improve the convergence of FedReID. Since local models outperform the global model, this suggests that local models contain more knowledge than the global model—simple server aggregation could not effectively aggregate knowledge from local models. KD is a method proposed by Hinton et al. [14] to transfer knowledge from a teacher model to a student model, where the teacher model contains more knowledge than the student model. We adopt KD to better transfer knowledge from local models to the global model, regarding clients as teachers and the server as the student.

\(^2\)The public person ReID dataset is shareable among the server and clients. This dataset can be unlabeled.
After clients finish local training and upload models, we apply KD with a public shared dataset \( D_{\text{shared}} \) in the server. Figure 3(b) illustrates the additional steps required from KD. In the first step, the server uses each trained model \( w_k \) of client \( k \) to generate soft labels \( \ell_k \) using samples of \( D_{\text{shared}} \). These soft labels represent the knowledge of clients’ models. In the second step, apart from model aggregation, the server aggregates these soft labels with \( \ell = \frac{1}{K} \sum_{k \in S} \ell_k \). In the third step, the server fine-tunes the global model with \( D_{\text{shared}} \) and corresponding labels \( \ell \) to learn the distilled knowledge.

5.3 Weight Adjustment

In addition to tackling the performance degradation caused by non-IID data, we propose to dynamically update the weights for aggregation to curb the adverse effect of unbalanced data. As discussed in Section 4.2.2, the weights of server aggregation are inappropriate. The formula for server aggregation [35, 63] is \( w^{t+1} = \frac{n_k}{n} w^{t+1}_k \), where \( n \) is the total data volume and \( n_k \) is the data volume of client \( k \). The weights of local models depend on the data volume of clients—larger datasets lead to larger weights. Since data volumes have large discrepancies among datasets, large datasets dominate in the server aggregation. For example, the weight of the largest dataset (MSMT17 [50]) is around 40%, whereas the weight of smallest dataset (iLIDS-VID [49]) is only 0.3%. Models from smaller datasets are almost negligible in aggregation. Such unbalanced data volume hampers clients with large datasets to effectively learn from others. Hence, we introduce a novel weight adjustment method to obtain more suitable weights for weighted average in aggregation.

Cosine distance weight. We introduce cosine distance weight (CDW) to substitute the weight of data volumes. CDW adjust the weights for aggregation dynamically in each round, based on how well models are trained in clients. It is measured by changes in features extracted from models before and after training. Such changes are calculated by cosine distance. In particular, in each training round, client \( k \) downloads and trains on the global model \( w^t_k \) from the server to obtain a new local model \( w^{t+1}_k \). Figure 3(c) demonstrates our method to calculate the new weight with \( w^t_k \) and \( w^{t+1}_k \), with the following steps: (1) client \( k \) extracts logits \( g^t_k \) with a random batch data \( D_{\text{batch}} \) using \((w^t_k, v^t_k)\); (2) client \( k \) obtains a new local model \((w^{t+1}_k, v^{t+1}_k)\) after local training; (3) client \( k \) extracts features \( g^{t+1}_k \) with \( D_{\text{batch}} \) using \((w^{t+1}_k, v^{t+1}_k)\); and (4) we calculate the cosine distance of

\[
\cos^2 \theta = 1 - \frac{g^t_k \cdot g^{t+1}_k}{||g^t_k|| ||g^{t+1}_k||}
\]

These labels are termed soft labels, as they are the predicted labels, not the actual labels, of the dataset.

\((w^t_k, v^t_k)\) is the concatenation of global model \( w^t_k \) and local classifier \( v^t_k \).
these two logits \(g^t_k\) and \(g^{t+1}_k\), with following formula:

\[
d_k^{t+1} = 1 - \frac{g^t_k \cdot g^{t+1}_k}{\|g^t_k\| \cdot \|g^{t+1}_k\|},
\]

where the cosine distance \(d_k^{t+1}\) of each client \(k\) is pushed to the server. The server uses the following formula to obtain the new weight:

\[
p_k^{t+1} = \frac{d_k^{t+1}}{\sum_{k \in S_t} d_k^{t+1}},
\]

where the server uses \(p_k^{t+1}\) to replace \(\frac{n_k}{n}\) as the new weight for aggregation. The intuition of CDW is that clients whose local trainings are more effective should contribute more to the aggregation. The cosine distance \(d_k\) measures the scale of changes in local training that updates model \(w_k\) to \(w_k^{t+1}\).

5.4 Combinations of Optimization Method

We can achieve even better performance by combinations of these three optimization methods: CC, KD, and CDW. We consider only combining CDW and CC, and CDW and KD.

It is not desirable to combine CC and KD because they both enhance the server aggregation. On the one hand, KD only fine-tunes a single global model, whereas CC contains multiple global models.
ALGORITHM 3: FedPav with Knowledge Distillation and Cosine Distance Weight

**Input:** Total data volume \( n \), client \( k \)'s data volume \( n_k \), local epoch \( E \), batch size \( B \), shared dataset \( D_{shared} \)

**Output:** Global model \( w^T \), local models of each client \( w^T_k \)

1. **Server:**
   - initialize \( w^0 \);
   - for each round \( t = 0 \) to \( T - 1 \) do
     - \( S_t \leftarrow \) (random set of \( K \) clients selected from \( N \) clients);
     - for each client \( k \in S_t \) in parallel do
       - \( w^{t+1}_k, d_k \leftarrow \text{Client}(w^t, k, t) \);
       - \( \ell_k \leftarrow \) (predict soft labels with \( w^{t+1}_k \) and \( D_{shared} \));
       - \( d \leftarrow \sum_{k \in S_t} d_k \);
       - \( w^{t+1}_k \leftarrow \sum_{k \in C_t} d_k w^{t+1}_k \);
       - \( \ell \leftarrow \frac{1}{K} \sum_{k \in S_t} \ell_k \);
     - \( w^{t+1} \leftarrow \) (fine-tune \( w^{t+1} \) with \( D_{shared} \) and soft labels \( \ell \));
   - return best global model \( w^T \), best client \( k \)'s local model \( w^T_k \);

2. **Client** \( (w, k, t) \):
   - \( v \leftarrow \) (initialize classifier \( v \) if \( t == 0 \), otherwise retrieve it from local store);
   - \( (w^t, v^t) \leftarrow (w, v) \);
   - for each local epoch \( e = 0 \) to \( E - 1 \) do
     - for \( b \in \) batches of local data in size \( B \) do
       - \( (w, v) \leftarrow (w, v) - \eta \nabla L((w, v); b) \);
     - \( g, g' \leftarrow \) (extract logits with \( (w, v) \) and \( (w^t, v^t) \) using \( B_{shared} \));
     - \( d = 1 - \frac{g' - g}{\|g\|\|g'\|} \);
   - store \( v \);
   - return \( w, d \);

models. On the other hand, both methods address the non-IID problem: KD aims to further improve the global model, whereas CC tends to elevate the performance of local models. Hence, we do not consider the combination of these two methods.

Since CDW tackles unbalanced data volume, either the combination of CDW and CC or the combination of CDW and KD addresses statistical heterogeneity with non-IID and unbalanced data problems. To combine CDW with CC or KD, we just need to replace the original weights in the server aggregation process with the new weights. As CC has no single global model, combining it with CDW aims to achieve better local models; as KD further fine-tunes the global model, combining it with CDW aims to achieve a better global model. We summarize these two combinations in Algorithms 2 and 3.

5.5 Evaluation

We present the empirical evaluation of these performance optimization approaches compared with the benchmark and stand-alone training. By default, we conduct these experiments with batch size \( B = 32 \) and local epoch \( E = 1 \). For both CC and KD, we adopt an additional unlabeled dataset—CUHK02 [27]. This dataset is regarded as a public dataset that is shareable among clients and the server. The CUHK02 dataset is an extension of the CUHK01 dataset. It includes 7,264 images of 1,816 identities collected from six camera views. We first evaluate the effectiveness of KD and the combination of CDW and KD by monitoring performance changes of global models as training proceeds. Figure 4 shows the performance
Fig. 4. Convergence and performance (rank-1 accuracy and mAP) comparison of the benchmark results, KD, and the combination of KD and CDW. KD effectively facilitates the convergence of FedReID. In addition, the combination of KD and CDW not only facilitates convergence but also effectively improves performance. These experiments were run with batch size $B = 32$ and local epoch $E = 1$.

Table 4. Increase in Rank-1 Accuracy of Benchmark Results, CC, CDW, and Combination of CC and CDW, When Compared with Standalone Training

| Methods      | MSMT17 | DukeMTMC | Market | CUHK03-NP | PRID2011 | CUHK01 | VIPeR | 3DPeS | iLIDS-VID |
|--------------|--------|----------|--------|-----------|----------|--------|-------|-------|-----------|
| Benchmark    | −1.3   | −2.0     | −5.4   | −9.8      | −4.3     | +11.6  | +24.5 | +15.2 | +32.7     |
| FedNova [48] | −2.1   | −2.8     | −4.4   | −14.6     | 0.0      | +9.9   | +24.4 | +12.6 | +35.8     |
| FedProx [26] | −0.1   | −1.6     | +1.0   | −6.4      | −1.0     | +12.5  | +24.1 | +7.7  | +34.7     |
| CC           | +2.4   | −1.3     | +0.1   | +3.9      | +6.0     | +9.3   | +4.1  | −1.2  | +16.3     |
| CDW          | +4.0   | +1.3     | +1.4   | +1.2      | +7.3     | +13.8  | +26.0 | +16.3 | +30.3     |
| CC & CDW     | +4.1   | +3.8     | +2.0   | +2.2      | +13.0    | +6.1   | +28.2 | +6.5  | +28.6     |

Note: CC effectively improves the performance on larger datasets, and CDW effectively elevates the performance on all datasets. In addition, the combination of CC and CDW achieves the best overall performance, especially on the larger datasets. These experiments were run with batch size $B = 32$ and local epoch $E = 1$.

changes (either rank-1 accuracy or mAP) of KD, the combination of CDW and KD, and the benchmark results on eight datasets. Compared with the benchmark results, training with KD achieves much better convergence; KD can also lead to higher performance, especially when datasets in clients share similar data distributions with the public shared dataset. For example, we use the CUHK02 dataset as the shared dataset, so the accuracy of the global models on the CUHK03-NP and CUHK01 datasets is better than the benchmark results. Moreover, training with the combination of KD and CDW achieves outstanding performance on almost all datasets—better than the benchmark results or training with KD. These results indicate that the combination of KD and CDW is able to obtain the best generalized global model that is transferable to other scenarios.

Next, we evaluate the effectiveness of CC, CDW, and the combination of these two methods by comparing the performance of their local models. Table 4 shows the increase in rank-1 accuracy of several methods when compared with standalone training on nine datasets. Although FedNova [48] and FedProx [26] slightly improve the performance of the smallest dataset (iLIDS-VID), they are incapable to elevate the performance of large datasets, like our benchmark method. We further analyze the results in three folds. First, CC effectively mitigates the drawback of the benchmark, improving the performance on larger datasets such as MSMT17 [50]. This is
Table 5. Rank-1 Accuracy Comparison of Our Proposed Methods (CC & CDW and KD & CDW) with the Existing Approaches on Two New Datasets: CAVIAR [6] and GRID [33]

| Datasets   | DSTML | UMDL | CrossGrad | MLDG | SSDAL | DIMN | Decentralized [51] | CC & CDW | KD & CDW |
|------------|-------|------|-----------|------|-------|------|-------------------|----------|----------|
| CAVIAR [6] | 28.2  | 41.6 | –         | –    | –     | –    | 45.6              | 46.8     | 53.2     |
| GRID [33]  | –     | –    | 9.0       | 15.8 | 22.4  | 29.3 | 24.2              | 30.0     | 36.8     |

Note: Our trained models outperform the existing methods on both datasets without extra fine-tuning. These results demonstrate the generalization ability of our methods.

because the dominance of larger datasets over smaller datasets is reduced as they are clustered into different groups in aggregation. Most of the time, CC creates two clusters: the first one contains clients with PRID2011, CUHK03-NP, VIPeR, 3DPeS, and iLIDS-VID datasets; the second one contains clients with MSMT17, DukeMTMC-reID, Market-1501, and CUHK01 datasets. Second, CDW outperforms the standalone training on all datasets. This indicates that CDW effectively addresses the unbalanced data problem such that all clients are beneficial in FL. Third, the combination of CDW and CC further elevates the performance in most datasets. Although such combination produces slight decreases on smaller datasets compared with CDW, it significantly further improves the performance of larger datasets. It increases the motivation of clients with larger datasets to participate in FL.

Last, we demonstrate the generalization ability of our methods by comparing existing methods on the CAVIAR4REID [6] and GRID [33] datasets. Specifically, we compare with the unsupervised cross-domain fine-tuning methods DSTML [16] and UMDL [38]; the unsupervised generalization methods CrossGrad [40], MLDS [23], SSDAL [43], and DIMN [42]; and recent work [51]. For evaluation on CAVIAR4REID, we follow the work of Liu et al. [32] and Peng et al. [38] to randomly select 36 identities that appeared on two camera views. The GRID dataset contains 250 identities from two camera views. For both datasets, we use images of one camera view as the query and another one as the gallery. Table 5 shows that our proposed FedReID with optimizations (CC & CDW and KD & CDW) outperforms all existing methods on rank-1 accuracy on both datasets; KD & CDW achieves especially good performance. Note that we do not fine-tune trained models on these two evaluation datasets. These results further illustrate the significance of our methods.

6 CONCLUSION

In this article, we presented FedReID, a new paradigm of person ReID training with decentralized data. To investigate the challenges of FedReID, we construct a new benchmark to simulate real-world scenarios. Based on the results and insights from benchmark analysis, we proposed three optimization approaches to elevate performance. We proposed CC and KD to address the non-IID problem and introduced CDW to address the unbalanced data problem. Empirical results demonstrated that the combination of CDW and CC achieves the best local models, and the combination of CDW and KD achieves the best global model, among all methods. In the future, we plan to investigate the system heterogeneity challenges among clients. We also plan to extend FedReID to support unsupervised learning.

REFERENCES

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. 2020. Federated learning based on dynamic regularization. In Proceedings of the International Conference on Learning Representations.

[2] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. 2011. 3DPeS: 3D people dataset for surveillance and forensics. In Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding (J-HGBU’11). ACM, New York, NY, 59–64. https://doi.org/10.1145/2072572.2072590
[3] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konecný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. LEAF: A benchmark for federated settings. CoRR abs/1812.01097 (2018). http://arxiv.org/abs/1812.01097.

[4] Tianlong Chen, Shaoqin Deng, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. 2019. ABD-Net: Attentive but diverse person re-identification. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 8351–8361.

[5] Yiqiang Chen, Xin Qin, Jingdong Wang, Chaohui Yu, and Wen Gao. 2020. FedHealth: A federated transfer learning framework for wearable healthcare. IEEE Intelligent Systems 35 (2020), 83–93.

[6] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. 2011. Custom pictorial structures for re-identification. In Proceedings of the British Machine Vision Conference (BMVC’11).

[7] Gregory Cohen, Saed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN’17). IEEE, Los Alamitos, CA, 2921–2926.

[8] Bart Custers, Alan M. Sears, Francien Dechesne, Ilina Georgieva, Tommaso Tani, and Simone van der Hof. 2019. EU Personal Data Protection in Policy and Practice. Springer.

[9] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc’Aurelio Ranzato, et al. 2012. Large scale distributed deep networks. In Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, 1223–1231. http://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks.pdf.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition.

[11] Douglas Gray and Hai Tao. 2008. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In Proceedings of the European Conference on Computer Vision. 262–275.

[12] Tianshu Hao, Yunyou Huang, Xu Wen, Wanling Gao, Fan Zhang, Chen Zheng, Lei Wang, et al. 2018. Edge AI-Bench: Towards comprehensive end-to-end edge computing benchmarking. In Proceedings of the 2018 BenchCouncil International Symposium on Benchmarking, Measuring, and Optimizing.

[13] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’16). 770–778.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In Proceedings of the NIPS Deep Learning and Representation Learning Workshop. http://arxiv.org/abs/1503.02531.

[15] Martin Hirzer, Peter Roth, Csaba Beleznai, and Horst Bischof. 2011. Person re-identification by descriptive and discriminative classification. In Proceedings of the Scandinavian Conference on Image Analysis (SCIA’11). 91–102.

[16] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. 2015. Deep transfer metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 325–333.

[17] Sixu Hu, Yuan Li, Xu Liu, Qinbin Li, Zhaomin Wu, and Bingsheng He. 2020. The OARF benchmark suite: Characterization and implications for federated learning systems. arXiv preprint arXiv:2006.07856 (2020).

[18] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, et al. 2019. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019).

[19] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In Proceedings of the International Conference on Machine Learning. 5132–5143.

[20] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report. University of Toronto, Toronto, Ontario.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (2012), 1097–1105.

[22] Qingming Leng, Mang Ye, and Qi Tian. 2019. A survey of open-world person re-identification. IEEE Transactions on Circuits and Systems for Video Technology 30, 4 (2019), 1092–1108.

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. 2018. Learning to generalize: Meta-learning for domain generalization. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence.

[24] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 10713–10722.

[25] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine 37 (2020), 50–60.

[26] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Mazar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. In Proceedings of the 3rd Machine Learning and Systems Conference (MLSys’20). 429–450.

[27] W. Li and X. Wang. 2013. Locally aligned feature transforms across views. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition. 3594–3601.
[28] Wei Li, Rui Zhao, and Xiaogang Wang. 2012. Human reidentification with transferred metric learning. In Computer Vision—ACCV 2012. Lecture Notes in Computer Science, Vol. 7724. Springer, 31–44.

[29] W. Li, R. Zhao, T. Xiao, and X. Wang. 2014. DeepReID: Deep filter pairing neural network for person re-identification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14). 152–159.

[30] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. 2014. DeepReID: Deep filter pairing neural network for person re-identification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’14). 152–159.

[31] Jiawei Liu, Zheng-Jun Zha, Q. I. Tian, Dong Liu, Ting Yao, Qiang Ling, and Tao Mei. 2016. Multi-scale triplet CNN for person re-identification. In Proceedings of the 24th ACM International Conference on Multimedia (MM’16). ACM, New York, NY, 192–196. https://doi.org/10.1145/2964284.2967209

[32] Xiao Liu, Mingli Song, Dacheng Tao, Xingchen Zhou, Chun Chen, and Jiajun Bu. 2014. Semi-supervised coupled dictionary learning for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3550–3557.

[33] Chen Change Loy, Chunchao Liu, and Shaogang Gong. 2013. Person re-identification by manifold ranking. In Proceedings of the 2013 IEEE Conference on Image Processing. IEEE, Los Alamitos, CA, 3567–3571.

[34] Jiahan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang. 2019. Real-world image datasets for federated learning. arXiv:1910.11089 [cs.CV] (2019).

[35] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS’17). 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[36] Khalil Muhammad, Qin Qin Wang, Diarmuid O’Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. 2020. FedFast: Going beyond average for faster training of federated recommender systems. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 1234–1242.

[37] Chaoyue Ni, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, Chengfei Lv, Zhihua Wu, and Guihai Chen. 2020. Billion-scale federated learning on mobile clients: A submodel design with tunable privacy. In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. 1–14.

[38] Peixi Peng, Tao Xiang, Yaowei Wang, Massimiliano Pontil, Shaogang Gong, Tiejun Huang, and Yonghong Tian. 2016. Unsupervised cross-dataset transfer learning for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1306–1315.

[39] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. 2019. Efficient parameter-free clustering using first neighbor relations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8934–8943.

[40] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothy, and Sunita Sarawagi. 2018. Generalizing across domains via cross-gradient training. arXiv preprint arXiv:1804.10745 (2018).

[41] Micah J. Sheller, G. Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. 2018. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In Proceedings of the International MICCAI Brain Lesion Workshop (BrainLes’18). 92–104.

[42] Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. 2019. Generalizable person re-identification by domain-invariant mapping network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 719–728.

[43] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. 2016. Deep attributes driven multi-camera person re-identification. In Proceedings of the European Conference on Computer Vision. 475–491.

[44] P. Sun, Y. Wen, R. Han, W. Feng, and S. Yan. 2022. GradientFlow: Optimizing network performance for large-scale distributed DNN training. IEEE Transactions on Big Data 8, 2 (2022), 495–507.

[45] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. 2018. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In Proceedings of the European Conference on Computer Vision (ECCV’18). 480–496.

[46] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. 2018. Learning discriminative features with multiple granularities for person re-identification. In Proceedings of the 26th ACM International Conference on Multimedia (MM’18). ACM, New York, NY, 274–282. https://doi.org/10.1145/3240508.3240552

[47] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated learning with matched averaging. In Proceedings of the International Conference on Learning Representations. https://openreview.net/forum?id=BkJhql5SDFS

[48] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. arXiv preprint arXiv:2007.07481 (2020).

[49] Tieqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. 2014. Person re-identification by video ranking. In Computer Vision—ECCV 2014. David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, Switzerland, 688–703.
[50] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. 2018. Person transfer GAN to bridge domain gap for person re-identification. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 79–88.

[51] Guile Wu and Shaogang Gong. 2021. Decentralised learning from independent multi-domain labels for person re-identification. Proceedings of the AAAI Conference on Artificial Intelligence 35, 4 (May 2021), 2898–2906. https://ojs.aaai.org/index.php/AAAI/article/view/16396.

[52] Jiwei Yang, Xu Shen, Xinmei Tian, Houqiang Li, Jianqiang Huang, and Xian-Sheng Hua. 2018. Local convolutional neural networks for person re-identification. In Proceedings of the 26th ACM International Conference on Multimedia (MM’18). ACM, New York, NY, 1074–1082. https://doi.org/10.1145/3240508.3240645

[53] Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. 2019. Federated learning with unbiased gradient aggregation and controllable meta updating. In Proceedings of the NIPS Federated Learning for Data Privacy and Confidentiality Workshop.

[54] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. 2021. Deep learning for person re-identification: A survey and outlook. arXiv:2001.04193 (2021).

[55] Lin Zhang, Yong Luo, Yan Bai, Bo Du, and Ling-Yu Duan. 2021. Federated learning for non-IID data via unified feature learning and optimization objective alignment. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 4420–4428.

[56] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-IID data. CoRR abs/1806.00582 (2018). http://arxiv.org/abs/1806.00582.

[57] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. 2015. Scalable person re-identification: A benchmark. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV’15). 1116–1124.

[58] Liang Zheng, Yi Yang, and Alexander G. Hauptmann. 2016. Person re-identification: Past, present and future. CoRR abs/1610.02984 (2016). http://arxiv.org/abs/1610.02984.

[59] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In Proceedings of the IEEE International Conference on Computer Vision.

[60] Weiming Zhuang, Xin Gan, Yonggang Wen, and Shuai Zhang. 2022. EasyFL: A low-code federated learning platform for dummies. IEEE Internet of Things Journal. Early access, January 20, 2022. https://doi.org/10.1109/JIOT.2022.3134842

[61] Weiming Zhuang, Xin Gan, Yonggang Wen, Shuai Zhang, and Shuai Yi. 2021. Collaborative unsupervised visual representation learning from decentralized data. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 4912–4921.

[62] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. 2022. Divergence-aware federated self-supervised learning. In Proceedings of the International Conference on Learning Representations. https://openreview.net/forum?id=oVE1z8NiNE.

[63] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. 2020. Performance optimization of federated person re-identification via benchmark analysis. In Proceedings of the 28th ACM International Conference on Multimedia. 955–963.

Received 14 May 2021; revised 11 March 2022; accepted 10 April 2022