Privacy-Preserving Adaptive Re-Identification without Image Transfer

Hamza Rami[1,2], Jhony H. Giraldo[1], Nicolas Winckler[2], and Stéphane Lathuilière[1]

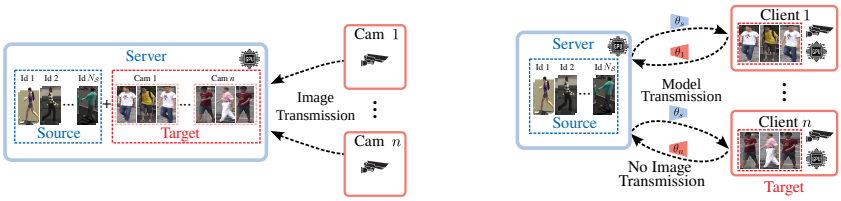
[1] LTCI, Télécom Paris, Institut Polytechnique de Paris
[2] Atos

Abstract. Re-Identification systems (Re-ID) are crucial for public safety but face the challenge of having to adapt to environments that differ from their training distribution. Furthermore, rigorous privacy protocols in public places are being enforced as apprehensions regarding individual freedom rise, adding layers of complexity to the deployment of accurate Re-ID systems in new environments. For example, in the European Union, the principles of "*Data Minimization*" and "*Purpose Limitation*" restrict the retention and processing of images to what is strictly necessary. These regulations pose a challenge to the conventional Re-ID training schemes that rely on centralizing data on servers. In this work, we present a novel setting for privacy-preserving Distributed Unsupervised Domain Adaptation for person Re-ID (DUDA-Rid) to address the problem of domain shift without requiring any image transfer outside the camera devices. To address this setting, we introduce Fed-Protoid, a novel solution that adapts person Re-ID models directly within the edge devices. Our proposed solution employs prototypes derived from the source domain to align feature statistics within edge devices. Those source prototypes are distributed across the edge devices to minimize a distributed Maximum Mean Discrepancy (MMD) loss tailored for the DUDA-Rid setting. Our experiments provide compelling evidence that Fed-Protoid outperforms all evaluated methods in terms of both accuracy and communication efficiency, all while maintaining data privacy.

Keywords: Person Re-ID · Unsupervised Domain Adaptation · Federated Learning

1 Introduction

Person Re-Identification (*Re-ID*) is a crucial task in computer vision, aimed at identifying specific individuals from a collection of images taken by various cameras [52]. The ability to perform Re-ID accurately and efficiently is essential for advancing intelligent surveillance systems and enhancing public safety. Recent years have witnessed remarkable progress in Re-ID performance, thanks to the adoption of deep learning techniques [50]. However, applying these approaches to data that is visually different from their training set results in a performance



(a) Traditional UDA for Person Re-ID (UDA-Rid) **(b)** Privacy-preserving Distributed UDA for person Re-ID (DUDA-Rid)

Fig. 1: In traditional Unsupervised Domain Adaptation (UDA) as depicted in Fig. (a), images are transmitted to a centralized server, which combines the unlabeled target images with the annotated source samples to train a model. In contrast, Distributed UDA for person re-identification (DUDA-Rid) shown in Fig. (b) keeps target images exclusively on edge devices. The learning process is divided between the server and cameras, the latter being equipped with local computational resources (). Only model parameters are exchanged between the clients and the server.

drop [33]. Annotating new data for each distinct environment is often infeasible, prompting previous studies to introduce Unsupervised Domain Adaptation (*UDA*) methods for person Re-ID.

UDA methods [14,15,49], combine a well-annotated dataset (*source domain*) with an unlabeled dataset (*target domain*), as illustrated in Fig. 1a. The objective of UDA methods is to train a model that can perform effectively in a new environment. Despite remarkable advancements in recent years [14,15], applying UDA to person Re-ID (*UDA-Rid*) encounters privacy concerns due to the need to collect and store images of individuals in public areas. Rigorous privacy regulations in many countries restrict technology providers from retaining images of people. For example, within the European Union, the General Data Protection Regulation (GDPR) obligates technology providers to adhere to the principles of "*Data Minimization*" [7] and "*Purpose Limitation*" [8], requiring that personal data be processed only when it is necessary for a designated purpose. These general principles prompt the following question: *What minimal data usage is truly "necessary" for Re-ID systems?*

An initial answer to this question emerges from recent studies [39, 40] that have developed approaches for UDA-Rid, focusing on eliminating the necessity for storing images. These approaches align with privacy regulations thereby clarifying GDPR's practical implications. However, these methods typically require transferring all captured images to a central server, which also poses privacy challenges [9]. Our work explores an alternative perspective on the question of minimal data usage: *Is transferring images outside the cameras truly "necessary" for Re-ID?* Our goal is to demonstrate that adaptation can be performed exclusively within edge devices, ensuring no image data is transmitted beyond its capture point as illustrated in Fig. 1b. This paradigm provides a privacy-compliant solution while leveraging the benefits of advanced Re-ID models.

To avoid the need for transmitting images, we approach this privacy-preserving Distributed UDA for person Re-ID (DUDA-Rid) task as a feder-

ated learning problem which inherently entails two interconnected challenges: (i) training the model in a distributed setup, and (ii) addressing the domain gap between the source and target datasets. Therefore, the key challenge behind the proposed setting is to simultaneously tackle the domain gap while working within a federated learning framework.

To jointly address the privacy and domain shift challenges in DUDA-Rid, we introduce a novel Federated Prototype-based learning for person Re-ID (*Fed-Protoid*) algorithm that enables domain adaptation without transmitting any image over the camera network. Fed-Protoid integrates a pseudo-labeling framework within the federated learning setup, and we propose a distributed version of the Maximum Mean Discrepancy (*MMD*) technique to enhance alignment between the source and target domains. Usually, MMD is calculated in a reproducing kernel Hilbert space using the kernel trick, which involves comparing source and target samples. Instead, we compute source prototypes and only share these prototypes with clients to adhere to privacy constraints. This approach for domain adaptation achieves high adaptation capabilities while keeping communication requirements to a minimum. Fed-Protoid readily outperforms all evaluated methods for DUDA-Rid in various challenging conditions in real-to-real and synthetic-to-real tasks. Furthermore, we show that using self-supervised pre-training [12] coupled with a Vision Transformer (ViT) significantly enhances performance across most scenarios for DUDA-Rid. We refer to this architecture as Fed-Protoid++.

Our main contributions can be summarized as follows:

- To our knowledge, we are the first to introduce and address the DUDA-Rid problem.
- We introduce a novel Fed-Protoid algorithm that uses prototypes to jointly address distributed learning and domain shift in DUDA-Rid. To this end, we propose a distributed version of the MMD loss to solve the domain gap in the federated setting.
- We further propose a Fed-Protoid++, which uses ViT and recent self-supervised pre-training techniques to achieve additional gains[3].

2 Related Work

Domain adaptation for person Re-ID. The current methods for domain adaptation can be broadly classified into three categories. The first is the *domain translation-based* methods [1, 16, 30], which use style transfer techniques such as CycleGAN [56] to modify the source domain to match the appearance of the target set. Recent studies in this category have focused on enhancing the translation process via self-similarity preservation [4] or camera-specific translation [55]. These types of methods are not well-suited for the DUDA-Rid problem since current federated learning methods with generative models are limited to toy datasets such as MNIST or CIFAR-10 [22, 41].

[3] Code available: https://github.com/ramiMMhamza/Fed-Protoid

The second category is based on *domain-invariant* feature learning. Shan *et al*. [27] proposed a framework for Re-ID by minimizing the distribution variation of the source's and target's mid-level features based on the MMD loss. Huang *et al*. [21] designed a novel domain adaptive module to separate the feature map, while Liu *et al*. [32] introduced a coupling optimization method for domain adaptive person Re-ID. Despite their effectiveness, these methods assume unrestricted access to the target domain on the server, relying on continuous image transmission and storage between cameras and the central server, an assumption that conflicts with privacy constraints in real-world applications.

The third category is the *pseudo-labeling* methods that utilize an iterative process alternating between clustering and fine-tuning [2, 11, 28, 44, 51]. Fan *et al*. [10] finetuned the Re-ID model using the cluster indexes as labels. Several extensions have been made to this framework such as Self-Similarity Grouping (*SSG*) [13], Mutual-Mean Teaching (*MMT*) [14], and Self-paced Contrastive Learning (*SpCL*) [15]. SSG [13] assigns different pseudo-labels to global and local features, MMT [14] employs a teacher-student framework with two student networks, and SpCL [15] gradually constructs more reliable clusters to refine a hybrid memory containing both source and target images. We opt for the pseudo-labeling framework as it outperforms previous techniques on most datasets and since it is compatible with our DUDA-Rid setting. Nevertheless, naively using a pseudo-labeling framework like MMT in the federated scenario incurs high communication costs. Therefore, we design our approach to reduce communication requirements between the clients and the server. Furthermore, our pseudo-labeling approach is enhanced with an explicit feature alignment mechanism based on MMD minimization.

Federated learning for person Re-ID. Federated Learning (FL) [36] aims at learning separately from multiple models trained on edges local data. FL restricts the sharing of data between clients and the server, as well as between clients to protect data privacy. FL has been applied to various computer vision tasks like image segmentation [31], classification [24], and person Re-ID [58]. Federated Averaging (*FedAvg*) [36] was first proposed by McMahan *et al*. based on averaging local models trained with local data and redistributing the averaged server model to the edges. Since FedAvg requires all the models in the edges to be identical to the server model, Federated Partial Averaging (*FedPav*) [58] was proposed to leverage only the common part of the clients' models (for example the backbones). In this work, we adapt the FedPav to include also the weights of the model being trained on the labeled source domain.

FL has also been investigated in the task of person Re-ID. FedReID [59] was first proposed to solve the task of supervised person Re-ID, which incorporates the FedPav optimization technique. A second work that also tackles the problem of FL in person Re-ID is FedUnReID [57], where the authors proposed an adaptation of the well-known unsupervised baseline for person Re-ID BUC [28]. In this spirit, FedUCA [29] was recently introduced to address the challenge of FL for person Re-ID. The authors draw inspiration from CAP [47], adopting both inter- and intra-camera losses to update a memory bank for each client.

These methods focus on the setting of federated by dataset. This setting represents client-edge architecture, where clients are defined as the edge servers. Each edge server collects and processes images from a network of multiple cameras. In contrast, our work focuses on a more restricted federated setting which does not allow the transmission of images between the cameras and any edge server. Finally, adapting FedUCA to our context is impractical. This is because, in our setting, each client possesses images from a single camera device, rendering the optimization of the inter-camera loss unfeasible.

Prototypical learning. The concept of prototypes in modern machine learning was first introduced in the field of few-shot learning to learn a metric space where classification can be performed by computing distances to prototype representations of each class [43]. Following this spirit, prototypical networks were applied to various computer vision tasks, such as semantic segmentation [5,37] and continual learning [17,42]. Prototypical learning has also made its way into federated learning, initially applied to diverse domains unrelated to person Re-ID. For instance, Federated Prototype learning (*FedProto*) [45] strives to align features globally using prototypes. Classifier Calibration with Virtual Representations (*CCVR*) [35] generates virtual features by leveraging an approximated Gaussian mixture model. More recently, Federated Prototypes Learning (*FPL*) [20] incorporates cluster prototypes and unbiased prototypes to mitigate the domain gap between the data in the server and clients. Notably, these previous methods are tailored for scenarios where prior information about the number of classes is available, such as in MNIST and CIFAR-10 datasets. Fed-Protoid is the first attempt to leverage prototypes in DUDA-Rid, which brings new challenges due to the unsupervised nature of the problem.

3 Federated Prototype-based Re-ID

Problem definition. The objective of this work is to train a model $F_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ to identify individuals in a collection of n cameras deployed in a target environment. To this end, we have at our disposal n unlabeled datasets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$ associated to each camera-client. Each dataset is composed of N_i training samples (images): $\mathcal{D}_i = \{\boldsymbol{x}_j^{(i)}\}_{j=1}^{N_i}$. Each target dataset $\mathcal{D}_i$ is confined to its respective edge camera device and cannot be transmitted, with each camera functioning as a client that interacts solely with a centralized server. We also have an annotated source dataset $\mathcal{S} = \{(\boldsymbol{x}_j^{\mathcal{S}}, \boldsymbol{y}_j^{\mathcal{S}})\}_{j=1}^{N_s}$ available on the server, where N_s represents the number of instances in the source dataset. The main challenge in this DUDA-Rid setting is to align the distributions of the different clients with the source domain in a distributed and privacy-preserving manner, *i.e.*, without sharing images at any point.

In classical UDA-Rid joint learning, the training objective commonly involves two main loss terms: the source domain loss $\mathcal{L}_s$, and the target domain loss $\mathcal{L}_t$. In non-distributed UDA-Rid, learning is commonly performed via the minimization of a linear combination of both source and target domain datasets as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{S}}\mathcal{L}_s(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) + \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\mathcal{L}_t(\boldsymbol{\theta}, \boldsymbol{x}), \tag{1}$$

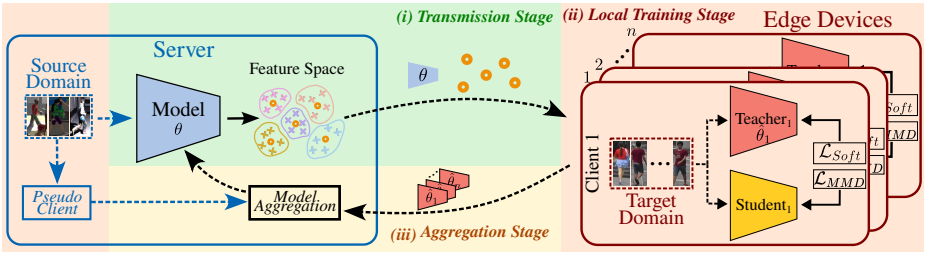


Fig. 2: The pipeline of Fed-Protoid. Our algorithm aggregates n edge-client models and one pseudo-client model in the server. Source prototypes are computed with the aggregated model. The prototypes and aggregated model are then distributed to all edge devices for local unsupervised training. This local training on each client involves cross-entropy, triplet, and Maximum Mean Discrepancy (MMD) loss functions.

where $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$. Typically, this loss is minimized using stochastic gradient descent. However, in our DUDA-Rid setting, the gradient of this total loss cannot be estimated without important communication costs. This is because the source term can be accessible only on the server via the source model, which we designate as the *pseudo-client*. Meanwhile, each device i is limited to compute only its local target loss term: $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_i}\mathcal{L}_t(\boldsymbol{\theta}, \boldsymbol{x})$. In the following, we outline our training strategy to minimize the total loss $\mathcal{L}$ in a distributed manner. Additionally, we describe the specifications of each loss term to facilitate communication-efficient and robust learning.

3.1 Overview of Fed-Protoid

Figure 2 shows the pipeline of Fed-Protoid for the DUDA-Rid setting. Our algorithm aggregates n client models along with the pseudo-client in a distributed setting. It adheres to standard practices in FL and functions in rounds. Each round is composed of three stages: (i) **transmission stage**: the aggregated model is distributed to every client and pseudo-client; (ii) **local training stage**: each client, as well as the pseudo-client, adapts their local model; (iii) **aggregation stage**: the local models are transmitted back to the server for aggregation.

At the beginning of each new round, the *transmission stage* also includes the transfer of source prototypes that are later used for source-target alignment. The aggregated model $F_{\boldsymbol{\theta}}$ computes the features of the source samples and the prototypes of each individual as the centroid of its feature representations. The prototypes of $\mathcal{S}$ are then transmitted along with the aggregated model $F_{\boldsymbol{\theta}}$ to all clients. Note that we assume the server utilizes either synthetic data or real data gathered in compliance with relevant legislation. Consequently, the transmission of source prototypes does not breach the privacy-preserving constraints.

In the *local training stage*, we use a teacher-student architecture to adapt $\boldsymbol{\theta}$ to the unlabeled target dataset $\mathcal{D}_i$ on each device i, and to the labeled source dataset $\mathcal{S}$. The server updates the pseudo-client via supervised training, while the local adaptation on each client involves cross-entropy, triplet, and MMD loss

functions. Considering the use of the cross-entropy loss and the variation of the number of identities for each client, we add to each local device i a personalized classifier head $\mathcal{C}_i$. This classifier is designed to match the number of classes to the respective number of identities in each client, including the pseudo-client.

Finally, in the *aggregation stage*, the server gathers and aggregates the n client models $\{F_{\hat{\boldsymbol{\theta}}_1}, F_{\hat{\boldsymbol{\theta}}_2}, \ldots, F_{\hat{\boldsymbol{\theta}}_n}\}$ obtained in the *local training stage* and the model $F_{\hat{\boldsymbol{\theta}}_s}$ trained on the source dataset using a weighted average sum as follows:

$$\boldsymbol{\theta} = \alpha\hat{\boldsymbol{\theta}}_s + (1-\alpha)\sum_{i=1}^{n} w_i\hat{\boldsymbol{\theta}}_i, \tag{2}$$

where $\{\hat{\boldsymbol{\theta}}_s, \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \ldots, \hat{\boldsymbol{\theta}}_n\}$ are the parameters of the client models after adaptation, α is the weight contribution of the pseudo-client model $\hat{\boldsymbol{\theta}}_s$, and w_i is the weight assigned to the ith client model given by $w_i = \frac{N_i}{\sum_{i=1}^{n} N_i}$.

3.2 Teacher-student architecture

All clients, encompassing the pseudo-client, employ the same teacher-student architecture. This framework is chosen for its effectiveness in enabling self-training techniques, which have been shown to yield optimal performance in UDA-Rid scenarios. While self-training is not required in the source domain due to its labeled nature, the use of the teacher-student framework favors similar training dynamics across both clients and the pseudo-client, facilitating more efficient model aggregation.

For simplicity, we assume here that we are in the ith client. Firstly, we initialize at each round the parameters of the teacher model $\bar{\boldsymbol{\theta}}_i$ and student model $\boldsymbol{\theta}_i$ with the parameters of the aggregated model $\boldsymbol{\theta}$. During adaptation, the student model is updated through the minimization of the target loss function $\mathcal{L}_t(\cdot)$ which are later detailed in Sections 3.3 and 3.4. After back-propagation through the student, we use the Exponential Moving Average (*EMA*) parameters update [23, 46] to compute the teacher model. At every iteration t, the parameters $\bar{\boldsymbol{\theta}}_i^{(t+1)}$ of the teacher model are given by:

$$\bar{\boldsymbol{\theta}}_i^{(t+1)} = \tau\bar{\boldsymbol{\theta}}_i^{(t)} + (1-\tau)\boldsymbol{\theta}_i, \tag{3}$$

where $\tau \in [0, 1)$ is a weighting factor. The model $\hat{\boldsymbol{\theta}}_i$, which is sent back to the server for model aggregation, is assigned to the final teacher model $\bar{\boldsymbol{\theta}}_i^{(t)}$.

3.3 Prototype estimation and server training

Source prototypes. In the *transmission stage*, the server sends prototypes to all the target clients. These prototypes are defined as the mean feature representation for each identity from the source domain. Formally, the prototype $\boldsymbol{p}_k$ of the kth identity is given by:

$$\boldsymbol{p}_k = \frac{1}{|\mathcal{S}_k|}\sum_{l\in\mathcal{S}_k} F_{\boldsymbol{\theta}}\left(\boldsymbol{x}_l^{\mathcal{S}}\right) \;\; \forall\; 1 \leq k \leq K, \tag{4}$$

where K is the number of identities in $\mathcal{S}$, $\mathcal{S}_k \subset \mathcal{S}$ is the set of images of the kth identity, and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset \ \forall i \neq j$. With enough diverse identities and images per identities from the source domain, the set of all source prototypes can serve as an approximation of the source domain distribution which can be transmitted with little cost. Subsequently, we use them to align the source and target distributions in the edge devices in the *local training stage*.

Pseudo-client loss. The source domain is treated as a pseudo-client in the *local-training stage*. Since the pseudo-client has access to the source domain dataset with labeled samples $\mathcal{S} = \{(\boldsymbol{x}_j^{\mathcal{S}}, \boldsymbol{y}_j^{\mathcal{S}})\}_{j=1}^{N_s}$, we can compute a supervised source loss $\mathcal{L}_s$ for the jth sample as:

$$\mathcal{L}_s(\boldsymbol{x}_j^{\mathcal{S}}, \boldsymbol{y}_j^{\mathcal{S}}) = \mathcal{L}_{CEs} + \mathcal{L}_{Tris}, \tag{5}$$

with

$$\mathcal{L}_{CEs} = \beta_1 \mathcal{L}_{CE}\left(\mathcal{C}_s \circ F_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{\mathcal{S}}), \boldsymbol{y}_j^{\mathcal{S}}\right) + \beta_2 \mathcal{L}_{CE}\left(\mathcal{C}_s \circ F_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{\mathcal{S}}), \bar{\mathcal{C}}_s \circ F_{\bar{\boldsymbol{\theta}}_s}(\boldsymbol{x}_j^{\mathcal{S}})\right),$$
$$\mathcal{L}_{Tris} = \gamma_1 \mathcal{L}_{Tri}\left(F_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{\mathcal{S}}), \boldsymbol{y}_j^{\mathcal{S}}\right) + \gamma_2 \mathcal{L}_{Tri}\left(F_{\boldsymbol{\theta}_s}(\boldsymbol{x}_j^{\mathcal{S}}), F_{\bar{\boldsymbol{\theta}}_s}(\boldsymbol{x}_j^{\mathcal{S}})\right),$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, $\bar{\mathcal{C}}$ is the teacher classifier head, $\mathcal{L}_{Tri}$ is the triplet loss, $\beta_1 + \beta_2 = 1$, and $\gamma_1 + \gamma_2 = 1$.

3.4 Local training on edge devices

We now detail the *local training stage* for the clients. A key difficulty of the target domain training is the estimation of the number of identities from an unlabeled set of images $\mathcal{D}_i = \{\boldsymbol{x}_j^{(i)}\}_{j=1}^{N_i}$. To this end, we apply the pseudo-labeling technique [28, 44, 51] consisting of an iterative process between clustering with the DBSCAN [6] method and fine-tuning. After this pseudo-labeling process we get an augmented dataset $\tilde{\mathcal{D}}_i = \{\boldsymbol{x}_j^{(i)}, \tilde{\boldsymbol{y}}_j^{(i)}\}_{j=1}^{N_i}$, where $\tilde{\boldsymbol{y}}_j^{(i)}$ is the pseudo-label associated to the jth sample.

Target client loss. In the edge devices, the teacher model generates soft labels that guide the student model to be less confident about the hard pseudo-labels [10]. This results in a refinement of the wrong predictions of the student model. Specifically, for a given target client dataset $\mathcal{D}_i$, the local loss function $\mathcal{L}_i$ in a mini-batch is given by:

$$\mathcal{L}_i = \frac{1}{m} \sum_{j \in \mathcal{D}_{i,m}} \mathcal{L}_p(\boldsymbol{x}_j^{(i)}) + \lambda \mathcal{L}_{MMD}\left(\mathcal{D}_{i,m}, \mathcal{P}_m\right), \tag{6}$$

where $\mathcal{D}_{i,m} \subseteq \mathcal{D}_i$ is the set of images in the mini-batch with $|\mathcal{D}_{i,m}| = m$, $\mathcal{L}_p(\boldsymbol{x}_j^{(i)})$ is a pseudo-label loss for the jth sample, λ is a weighting factor, and $\mathcal{L}_{MMD}\left(\mathcal{D}_{i,m}, \mathcal{P}_m\right)$ is the MMD loss between $\mathcal{D}_{i,m}$ and a subset of the prototypes $\mathcal{P}_m \subseteq \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K\}$ with $|\mathcal{P}_m| = m$. The pseudo-label loss $\mathcal{L}_p(\boldsymbol{x}_j^{(i)})$ is the same as in (5), but since the true labels are not available in the clients, we use the pseudo-labels $\tilde{\boldsymbol{y}}_j^{(i)}$ instead. The local loss is used to update the student parameters $\boldsymbol{\theta}_i$.

Personalized pseudo-epoch. A significant challenge in federated learning scenarios is determining the optimal number of training epochs for each client. This decision is crucial to achieve the best balance between learning efficiency and transmission overhead. In our task, this problem is also crucial to prevent overfitting in clients with only a few identities or images. To ensure equal usage of all identities within a client during a federated training round, we introduce the *Personalized Pseudo-Epoch* (*PPE*).

For a specific client i, let K_i represent the count of identities in $\mathcal{D}_i$ as identified by the DBSCAN algorithm. In every iteration, mini-batches are constructed by randomly selecting I identities. From each chosen identity, B images are sampled, as in previous works [14, 15]. Consequently, we define the number of iterations required for one PPE as $P_i = \frac{K_i}{I}$. By doing so, we ensure that, during a federated training round, each identity is presented an equal number of times, irrespective of the varying number of identities present in each client's dataset.

4 Experiments and Results

In this section, we detail our experimental setup, covering datasets, implementation details, and evaluation metrics. Subsequently, we compare Fed-Protoid against two categories of approaches: (i) FL + UDA, wherein we adapt the UDA methods MMT [14] and SpCL [15] to DUDA-Rid, and (ii) federated learning approaches for person Re-ID, namely FedReID [59] and FedUnReID [57]. Finally, we conduct a series of ablation studies to (i) demonstrate the efficacy of the transformer-based architecture coupled with self-supervised pre-training (Fed-Protoid++), (ii) confirm the suitability of the MMD loss, and (iii) validate the teacher-student architecture and aggregation choice.

4.1 Experimental setup

Datasets. We evaluate our method in real-to-real and synthetic-to-real scenarios. For the source domain, we use two datasets:

- *MSMT* (MS) [49] includes videos from 15 cameras. The training set has 32,621 images of 1,042 identities, while the test set comprises 11,659 query images and 82,161 gallery images from 3,060 identities.
- *RandPerson* (RP) [48] is a synthetic dataset containing 8,000 identities and 132,145 images.

For the target domain, we consider the following datasets:

- *Market* (M) [53] has 1,501 identities captured by six cameras. It includes 32,668 images, with 12,936 training images from 751 identities and 19,732 test images from the remaining 750 identities.
- *CUHK03-np* (C) [25] comprises 14,097 photos of 1,467 individual identities where each identity is recorded by two cameras. We use the new protocol [54] which consists of splitting the dataset into 767 identities for training and 700 identities for testing. In testing, each query identity is selected by both cameras to ensure the evaluation of the cross-camera Re-ID.

Evaluation protocol. In the DUDA-Rid setting, we assume the cameras are equipped with embedded devices that can train the teacher-student models of the clients. To mimic this scenario, we split *Market* into six clients and *CUHK03-np* into two clients, where each client contains images from a single camera viewpoint. We adopt the commonly used metrics for evaluation in person Re-ID [14, 15]: mean Average Precision (mAP) and CMC Rank-1 [53] accuracies. During each round of the federated learning, each client performs a number of PPEs. Therefore, we compute the metrics on a separate test set related to the target domain using the aggregated model from the server. We report for each method the highest average mAP and Rank-1, with the number of rounds required to reach these top scores.

Implementation details. For a fair comparison with the state-of-the-art methods, we follow the common practices in the UDA for person Re-ID field by adopting ResNet-50 [18] pre-trained on ImageNet [3] as a backbone. We train every method for 800 rounds of federated learning. Except for FedUnReID, where we follow its implementation details and set the training number of rounds to 200. We stop the training process upon observing any signs of divergence, specifically when there is a considerable decline in the test mAP over the training rounds. We present a sensibility analysis of the hyper-parameters of Fed-Protoid in the supplementary material.

To stress the practicality of the adopted setting, we also consider a variant of Fed-Protoid called Fed-Protoid++, where we employ a stronger backbone architecture and leverage as initialization a model pre-trained on a large-scale Re-ID dataset. Concerning the architecture, we transition from the traditional ResNet-50 to a ViT [19] backbone. We complement the backbone improvement with the adoption of self-supervised pre-trained models on the large-scale unlabeled dataset LUPerson [34].

4.2 Comparison with the State of the Art

Since Fed-Protoid is the first method that addresses DUDA-Rid, we adapt various methods initially designed for other settings to facilitate the comparison.

Competitive methods. We assess the performance of Fed-Protoid against two federated frameworks for Re-ID: FedReID [59] and FedUnReID [57]. On one hand, FedReID is a Fully Supervised (FS) method that uses dynamic weight adjustment, knowledge distillation, and FedPav as its aggregation rule. The original study of FedReID also explores our dataset partition in the edge devices, *i.e.*, each client contains images from a single camera. However, FedReID requires the target dataset $\mathcal{D}_i$ to be labeled, while we do not have such a constraint. On the other hand, FedUnReID is a framework that adapts the Purely Unsupervised (PU) baseline Bottum-Up-Clustering (BUC) [28] for Federated person Re-ID. We also compare Fed-Protoid with FedReID+$\mathcal{S}$ and FedUnReID+$\mathcal{S}$. These variants are improved versions of the original frameworks where we initialize the models with supervised source pre-training, offering a fairer comparison with Fed-Protoid that leverages the source domain's knowledge.

Table 1: Comparison of mAP, Rank-1 accuracy, and number of rounds (#R) for four adaptation configurations. The different methods range from Fully Supervised (FS) and Purely Unsupervised (PU) to Unsupervised Domain Adaptation (UDA). *The communication cost for a single round in MMT is four times greater than that in the other ResNet-based models.

Method	Type	MS → M			MS → C			RP → M			RP → C		
		mAP	Rank-1	#R	mAP	Rank-1	#R	mAP	Rank-1	#R	mAP	Rank-1	#R
FedReID [59]	FS	38.9	61.9	800	11.6	11.7	750	38.9	61.9	800	11.6	11.7	750
FedReID+$\mathcal{S}$	FS	39.5	63.8	790	12.0	12.3	800	40.0	64.4	800	11.4	11.6	780
FedUnReID [57]	PU	19.5	43.6	190	6.8	7.0	170	19.5	43.6	190	6.8	7.0	170
FedUnReID+$\mathcal{S}$	PU	31.0	61.7	170	10.5	11.1	170	31.5	31.8	170	10.6	11.6	160
FedAvg+SpCL [15]	UDA	39.1	67.3	8	19.7	18.9	1	36.1	62.9	9	21.2	21.6	3
FedPav+MMT* [14]	UDA	45.8	73.6	70	22.4	21.9	9	30.2	58.9	9	19.0	19.7	9
Fed-Protoid (ours)	UDA	51.0	76.8	288	23.8	23.1	22	39.2	66.4	22	25.1	24.7	253
Fed-Protoid++ (ours)	UDA	**61.7**	**82.6**	170	**43.8**	**42.4**	24	**45.2**	**71.8**	186	**25.7**	**24.9**	212

Since our DUDA-Rid setting combines both UDA and FL, we extend our comparison to include Fed-Protoid against UDA methods for person Re-ID. For the UDA methods, we adapt the state-of-the-art pseudo-labeling approaches SpCL and MMT to suit the DUDA-Rid setting. In this process, during each federated learning round, we send copies of these UDA frameworks to all the edge clients for local training. Additionally, for a fair comparison with Fed-Protoid, we train in the server the pseudo-source client on the labeled source domain $\mathcal{S}$. The aggregation rule for these adapted UDA methods, denoted FedAvg+SpCL and FedPav+MMT is consistent with Eq. (2). The main objective of this comparison is to evaluate the effectiveness of traditional UDA methods when confronted with privacy constraints, where the target domain is distributed over multiple edge devices (cameras).

Quantitative results and discussions. Table 1 reports the best mAP accuracy and CMC Rank-1 score alongside the number of rounds (#R) required to achieve these top scores. We include two real-to-real configurations MS → M, MS → C, and two synthetic-to-real configurations RP → M, RP → C.

Fed-Protoid demonstrates good results against the supervised and unsupervised federated learning methods for person Re-ID, FedReID, and FedUnReID as shown in Table 1. For example, Fed-Protoid obtains 23.8 of mAP in MS → C, outperforming FedReID with 11.6 mAP and FedUnReID with 6.8 mAP. More interestingly, Fed-Protoid reaches this performance after only 22 rounds, whereas FedReID and FedUnReID require 750 and 170 rounds, respectively. Fed-Protoid also reaches superior performance in the RP → C configuration with 25.1 mAP compared to the other federated learning methods. We also evaluate the improved versions FedReID+$\mathcal{S}$ and FedUnReID+$\mathcal{S}$ where both models start with a pre-training on the source domain. Even though starting from the source pre-trained models improves slightly the original models' performances, they are below the performances obtained by Fed-Protoid in almost all configurations.

Fed-Protoid also improves significantly the performance of the adapted UDA baselines SpCL and MMT as shown in Table 1. In MS → M, SpCL and MMT

achieve a mAP of 39.1 and 45.8, respectively, while Fed-Protoid achieves a mAP accuracy of 51. This observation can also be generalized to the synthetic-to-real configurations like RP $\rightarrow$ M, where Fed-Protoid reaches a performance of 39.2 mAP, while SpCL and MMT achieve 36.1 and 30.2 mAP, respectively. Even though Fed-Protoid requires more communication rounds to reach its optimal performance compared to MMT, it is important to notice that Fed-Protoid transmits approximately only a quarter of the data weights per round. This is because the MMT architecture sends four backbones to the server, whereas Fed-Protoid needs to share only one (the teacher model) and the transmission cost of the prototypes is almost negligible compared to the weights of the models. Overall, Fed-Protoid is more effective in the DUDA-Rid scenario than the adapted UDA baselines SpCL and MMT as shown in Table 1.

Fed-Protoid++ Recent work [34] has shown the suitability and effectiveness of self-supervised pre-training methods for transformer-based methods [19] in person Re-ID, yielding substantial enhancements across a variety of Re-ID benchmarks. In the context of our DUDA-Rid setting, the performance of Fed-Protoid++ is consistent with the aforementioned findings as shown in Table 1. Particularly, transitioning from the ResNet-50 to a ViT backbone pre-trained in a self-supervised way leads to remarkable performance enhancements in all the configurations. For instance, we observe an increase in the mAP from 51 to 61.7 in MS $\rightarrow$ M. Similarly, we have an improvement from 39.2 to 45.2 in RP $\rightarrow$ M in the mAP, showing Fed-Protoid++ enhanced effectiveness. The improvement in the performance of using transformer-based models in person Re-ID comes from three main reasons [19]: (i) the multi-head self-attention effectively captures long-range dependencies and drives the model to focus on diverse human-body parts, (ii) transformer-based models have the ability to extract fine-grained features which is essential in person Re-ID, and (iii) the rich variety and volume of the LUPerson dataset provide the model with the capability of extracting more robust features that are generalizable across small downstream datasets. We perform an ablation study in Section 4.3 to empirically validate these points.

Training dynamics. In Fig. 3, we illustrate the progression of the mAP of the different methods in the MS $\rightarrow$ M configuration. Notably, there is a difference in the evolution of the mAP between the methods designed for the FL FedReID+$\mathcal{S}$ and FedUnReID+$\mathcal{S}$, and the UDA-based methods FedAvg+SpCL and FedPav+MMT. Specifically, while FedReID+$\mathcal{S}$ and FedUnReID+$\mathcal{S}$ exhibit a consistent improvement during the training, this trend is not mirrored in the performance of FedAvg+SpCL and FedPav+MMT. In fact, both UDA-based methods tend to converge rapidly at the early stage of FL training. This is because initially, the local models are relatively close to the source model, allowing for easier leveraging of the source domain knowledge in the first rounds of FL. However, as training progresses, the local models start diverging from the source model, leading to a decrease in performance. Conversely, our methods demonstrate a stable progression, effectively managing to mitigate domain shift during training. This highlights the effectiveness of our approach in maintaining consistent performance in the DUDA-Rid setting.

Table 2: Impact of the backbone architecture and pre-training datasets on Fed-Protoid.

Backbone	Pre-tr.	Warm-up	MS → M	MS → C
ResNet-50	ImageNet	✗	41.5	23.7
ResNet-50	ImageNet	✓	**51.0**	**23.8**
ResNet-50	LUPerson	✗	44.0	13.6
ResNet-50	LUPerson	✓	46.0	16.0
ViT (S)	ImageNet	✓	52.4	27.5
ViT (S)	LUPerson	✗	59.7	23.9
ViT (S)	LUPerson	✓	**61.7**	**43.8**

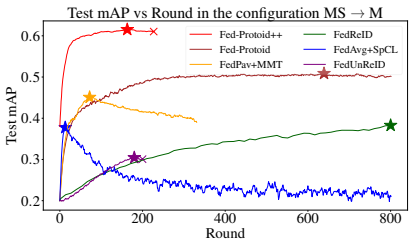


Fig. 3: Test mAP vs Round (MS → M). ★ denotes the maximum mAP.

4.3 Ablation studies

On the effectiveness of the backbones and pre-training datasets. The first ablation study focuses on the impact of the different modifications done to design Fed-Protoid++. Table 2 shows the ablation study for different backbones, pre-training strategies, and warm-up. For the backbones, we have the option to use either the classical ResNet-50 or ViT Small (S). For the pre-training dataset and strategy, we can either use fully supervised on ImageNet or self-supervised in LUPerson. The warm-up consists of adding an additional supervised pre-training on the source domain $\mathcal{S}$. We observe in Table 2 that adopting the ViT backbone combined with an appropriate pre-training dataset significantly enhances the performance. Fed-Protoid corresponds to a ResNet-50 backbone pre-trained on ImageNet, while Fed-Protoid++ corresponds to a ViT model pre-trained on the large-scale LUPerson dataset in a self-supervised way.

A key finding in Table 2 is that using ViT (S) instead of ResNet-50 with the same pre-training strategy consistently results in performance improvements. For instance, when comparing ResNet-50 and ViT (S) pre-trained on ImageNet, the mAP slightly improves from 51 to 52.4 in MS → M, and from 23.8 to 27.5 in MS → C. This suggests that the ViT-based backbone learns more robust features in the target domain. Additionally, ViT (S) captures more generalizable features when pre-trained in a self-supervised way thanks to the large and diverse set of unlabeled images in LUPerson. As a final remark, the warm-up generally enhances the performances across all the scenarios and configurations.

On the effectiveness of the teacher-student framework. The integration of the teacher-student architecture gives multiple possibilities to the design of Fed-Protoid. Table 3 shows the results of the ablation study where (i) we do not have the teacher-student framework in the pseudo-client, (ii) we have the teacher-student framework in the pseudo-client and we transmit the students for aggregation, and (iii) we have the teacher-student framework in the pseudo-client and we transmit the teachers for aggregation. Table 3 shows a considerable drop in performance when the teacher model is omitted from the pseudo-client in both MS → M and MS → C. Specifically, the mAP decreases from 51 to 37.4 in MS → M, and from 23.8 to 22.5 in MS → C, underscoring the crucial role of the teacher model in the pseudo-client. These findings align with our claim in Section

Table 3: Ablation study of the teacher-student framework and the choice of the transmitted model.

Teach.-Stud. on $\mathcal{S}$	Transmission	MS → M	MS → C
✗	Teacher	37.4	22.5
✓	Student	<u>49.3</u>	**23.8**
✓	Teacher	**51.0**	**23.8**

Table 4: Impact of the kernel function choice for the Maximum Mean Discrepancy (MMD) loss.

MMD	Kernel	MS → M	MS → C
✗	–	42.6	22.4
✓	Linear	38.1	22.1
✓	Order 2	27.8	13.1
✓	Gaussian	**51.0**	**23.8**

3.2 regarding the use of the teacher-student architecture in the pseudo-client to keep similar training dynamics with the other clients. The teacher-student architecture gives another alternative of transmitting the student instead of the teacher models for aggregation. Table 3 illustrates that this alternative yields reasonable performance. However, using students for aggregation falls marginally short of the performance achieved by aggregating the teacher models.

On the effectiveness of the MMD loss. The MMD loss, serving as a measure of domain discrepancy, offers a variety of options for the reproducing kernel Hilbert space where we minimize the distance between source prototypes and target feature representations. Table 4 shows a comparison between different kernel functions including linear, order 2, and Gaussian kernels. The linear kernel minimizes the mean average of prototypes and target features distributions, while the order 2 kernel minimizes the mean average and the standard deviation of these distributions. Table 4 suggests that the linear and order 2 kernels are not effective in the DUDA-Rid setting. This can be attributed to potentially biased estimations of the true mean (linear) and variance (2nd order) within relatively small and diverse batches of images. Furthermore, using the MMD loss with a Gaussian kernel achieves superior performance in all cases, including when MMD loss is not used at all. We further evaluate the MMD's effectiveness by examining its performance with limited prototypes and comparing the proposed distributed MMD with the original MMD in the supplementary material, demonstrating its robustness against device storage and communication limitations and proving it to be effective and suitable for our setting.

5 Conclusion

In this paper, we presented a novel approach for the task of UDA for person Re-ID that addresses both problems of domain shift and privacy preservation. Our method Fed-Protoid learns a person Re-ID model across multiple edge devices without transmitting target images from the cameras where they were captured. By integrating a teacher-student architecture and a source-client model, trained in the server side on labeled source domain, and introducing a distributed version of the Maximum Mean Discrepancy (MMD) loss, Fed-Protoid ensures effective domain adaptation with the target clients while eliminating the need for explicit inter-camera learning and keeping communication requirements minimal.

References

1. Chen, Y., Zhu, X., Gong, S.: Instance-guided context rendering for cross-domain person re-identification. In: ICCV (2019)
2. Delorme, G., Xu, Y., Lathuilière, S., Horaud, R., Alameda-Pineda, X.: CANU-ReID: a conditional adversarial network for unsupervised person re-identification. In: ICPR (2021)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
4. Deng, W., Zheng, L., Kang, G., Yang, Y., Ye, Q., Jiao, J.: Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In: CVPR (2018)
5. Dong, N., Xing, E.P.: Few-shot semantic segmentation with prototype learning. In: BMVC (2018)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD (1996)
7. European Parliament and Council of the European Union: General data protection regulation (GDPR) (2016), chapter 2, Article 5.c
8. European Parliament and Council of the European Union: General data protection regulation (GDPR) (2016), chapter 2, Article 5.b
9. European Parliament and Council of the European Union: General data protection regulation (GDPR) (2016), chapter 5, Article 44-49
10. Fan, H., Zheng, L., Yan, C., Yang, Y.: Unsupervised person re-identification: Clustering and fine-tuning. ACM TOMM (2018)
11. Feng, H., Chen, M., Hu, J., Shen, D., Liu, H., Cai, D.: Complementary pseudo labels for unsupervised domain adaptation on person re-identification. IEEE TIP (2021)
12. Fu, D., Chen, D., Bao, J., Yang, H., Yuan, L., Zhang, L., Li, H., Chen, D.: Unsupervised pre-training for person re-identification. CVPR (2021)
13. Fu, Y., Wei, Y., Wang, G., Zhou, Y., Shi, H., Huang, T.S.: Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In: ICCV (2019)
14. Ge, Y., Chen, D., Li, H.: Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In: ICLR (2020)
15. Ge, Y., Chen, D., Zhu, F., Zhao, R., Li, H.: Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In: NeurIPS (2020)
16. Ge, Y., Zhu, F., Chen, D., Zhao, R., Wang, X., Li, H.: Structured domain adaptation with online relation regularization for unsupervised person re-id. IEEE TNNLS (2022)
17. Han, X., Dai, Y., Gao, T., Lin, Y., Liu, Z., Li, P., Sun, M., Zhou, J.: Continual relation learning via episodic memory activation and reconsolidation. In: ACL (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
19. He, S., Luo, H., Wang, P., Wang, F., Li, H., Jiang, W.: Transreid: Transformer-based object re-identification. In: ICCV (2021)
20. Huang, W., Ye, M., Shi, Z., Li, H., Du, B.: Rethinking federated learning with domain shift: A prototype view. In: CVPR (2023)
21. Huang, Y., Peng, P., Jin, Y., Xing, J., Lang, C., Feng, S.: Domain adaptive attention model for unsupervised cross-domain person re-identification. arXiv preprint arXiv:1905.10529 (2019)

22. Kortoçi, P., Liang, Y., Zhou, P., Lee, L.H., Mehrabi, A., Hui, P., Tarkoma, S., Crowcroft, J.: Federated split gans. In: ACM MobiComWorkshops (2022)
23. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: ICLR (2017)
24. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. MLSys (2020)
25. Li, W., Zhao, R., Xiao, T., Wang, X.: DeepReID: Deep filter pairing neural network for person re-identification. In: CVPR (2014)
26. Liao, S., Shao, L.: TransMatcher: Deep image matching through transformers for generalizable person re-identification. NeurIPS (2021)
27. Lin, S., Li, H., Li, C.T., Kot, A.C.: Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification. In: BMVC (2018)
28. Lin, Y., Dong, X., Zheng, L., Yang, Y.: A bottom-up clustering approach to unsupervised person re-identification. In: AAAI (2019)
29. Liu, J., Zhuang, W., Wen, Y., Huang, J., Lin, W.: Optimizing federated unsupervised person re-identification via camera-aware clustering. In: MMSP (2022)
30. Liu, J., Zha, Z.J., Chen, D., Hong, R., Wang, M.: Adaptive transfer network for cross-domain person re-identification. In: CVPR (2019)
31. Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: CVPR (2021)
32. Liu, X., Zhang, S.: Domain adaptive person re-identification via coupling optimization. In: ACM MM (2020)
33. Luo, H., Gu, Y., Liao, X., Lai, S., Jiang, W.: Bag of tricks and a strong baseline for deep person re-identification. In: CVPRW (2019)
34. Luo, H., Wang, P., Xu, Y., Ding, F., Zhou, Y., Wang, F., Li, H., Jin, R.: Self-supervised pre-training for transformer-based person re-identification. arXiv preprint arXiv:2111.12084 (2021)
35. Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., Feng, J.: No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. NeurIPS (2021)
36. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS (2017)
37. Nguyen, K., Todorovic, S.: Feature weighting and boosting for few-shot segmentation. In: ICCV (2019)
38. Ni, H., et al.: Part-aware transformer for generalizable person re-identification. In: ICCV (2023)
39. Rami, H., Giraldo, J.H., Winckler, N., Lathuilière, S.: Source-guided similarity preservation for online person re-identification. In: WACV (2024)
40. Rami, H., Ospici, M., Lathuilière, S.: Online unsupervised domain adaptation for person re-identification. In: CVPRW (2022)
41. Rasouli, M., Sun, T., Rajagopal, R.: FedGAN: Federated generative adversarial networks for distributed data. arXiv preprint arXiv:2006.07228 (2020)
42. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR (2017)
43. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. NeurIPS (2017)
44. Song, L., Wang, C., Zhang, L., Du, B., Zhang, Q., Huang, C., Wang, X.: Unsupervised domain adaptive re-identification: Theory and practice. Pattern Recognition (2020)

45. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: FedProto: Federated prototype learning across heterogeneous clients. In: AAAI (2022)
46. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS (2017)
47. Wang, M., Lai, B., Huang, J., Gong, X., Hua, X.S.: Camera-aware proxies for unsupervised person re-identification. In: AAAI (2021)
48. Wang, Y., Liao, S., Shao, L.: Surpassing real-world source training data: Random 3D characters for generalizable person re-identification. In: ACM MM (2020)
49. Wei, L., Zhang, S., Gao, W., Tian, Q.: Person transfer GAN to bridge domain gap for person re-identification. In: CVPR (2018)
50. Wu, C., Ge, W., Wu, A., Chang, X.: Camera-conditioned stable feature generation for isolated camera supervised person re-identification. In: CVPR (2022)
51. Ye, M., Li, J., Ma, A.J., Zheng, L., Yuen, P.C.: Dynamic graph co-matching for unsupervised video-based person re-identification. IEEE TIP (2019)
52. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.H.: Deep learning for person re-identification: A survey and outlook. IEEE TPAMI (2022)
53. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: ICCV (2015)
54. Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking person re-identification with k-reciprocal encoding. In: CVPR (2017)
55. Zhong, Z., Zheng, L., Li, S., Yang, Y.: Generalizing a person retrieval model hetero- and homogeneously. In: ECCV (2018)
56. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV (2017)
57. Zhuang, W., Wen, Y., Zhang, S.: Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In: ACM MM (2021)
58. Zhuang, W., Wen, Y., Zhang, X., Gan, X., Yin, D., Zhou, D., Zhang, S., Yi, S.: Performance optimization of federated person re-identification via benchmark analysis. In: ACM MM (2020)
59. Zhuang, W., Wen, Y., Zhang, X., Gan, X., Yin, D., Zhou, D., Zhang, S., Yi, S.: Performance optimization of federated person re-identification via benchmark analysis. In: ACM MM (2020)

Privacy-Preserving Adaptive Re-Identification without Image Transfer : Supplementary Materials

Hamza Rami[1,2], Jhony H. Giraldo[1], Nicolas Winckler[2], and Stéphane Lathuilière[1]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris
[2] Atos

In this supplementary material, we provide results and analysis of additional experiments and present additional details about the Fed-Protoid.

- We provide the pseudo-code of Fed-Protoid to give more details about its algorithmic structure.
- We present a set of experiments focused on the source prototypes, initially showing the significance of utilizing the global model for their computation instead of the pseudo client. Subsequently, we highlight the impact of reducing the number of source prototypes in the performance of Fed-Protoid.
- We include a detailed analysis regarding the sensibility of the hyper-parameters of Fed-Protoid.
- We compare the distributed MMD with the original MMD and some Domain Generalization (DG) Re-ID methods.

1 Fed-Protoid: Algorithm

For completeness, we detail the Fed-Protoid algorithm as follows:

2 Additional experiments on the source prototypes: computation and communication

2.1 Impact of Global model in source prototype computation

In this section, we present experimental results for both Fed-Protoid and Fed-Protoid++, showing the advantages of utilizing the global model for computing source prototypes. The results shown in Table 1 indicate that across the two configurations MS $\rightarrow$ M and MS $\rightarrow$ C, we obtain superior performance when the prototypes are derived from the global model instead of the pseudo-client model. The effectiveness of using the global model in prototype computation can be attributed to its ability to bridge the gap between the source and target domain distributions. Essentially, the prototypes generated by the global model represent a median distribution that lies between those of the source and target domains. This intermediary positioning facilitates more efficient optimization of

Algorithm 1 Fed-Protoid algorithm

Input: n unlabeled datasets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$, an annotated dataset $\mathcal{S}$, and the source pre-trained weights $\boldsymbol{\theta}_s$
Initialize model $F_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}_s$
for each training round **do**
    Transmission Stage:
    Transmit $F_{\boldsymbol{\theta}}$ and source prototypes to all clients
    Local Training Stage:
    Update pseudo-client model $F_{\hat{\boldsymbol{\theta}}_s}$ using $\mathcal{S}$ and $\mathcal{L}_s$
    for each client i **do**
        Update client model $F_{\hat{\boldsymbol{\theta}}_i}$ using local dataset $\mathcal{D}_i$ and $\mathcal{L}_i$
    end for
    Aggregation Stage:
    Aggregate models using equation $\boldsymbol{\theta} = \alpha\hat{\boldsymbol{\theta}}_s + (1-\alpha)\sum_{i=1}^{n} w_i\hat{\boldsymbol{\theta}}_i$
end for
Output: Trained federated model $F_{\boldsymbol{\theta}}$

the Maximum Mean Discrepancy (MMD). Instead of directly aligning the target domain with the source domain, the global model provides features that are equidistant to both domains. Consequently, this approach converges all distributions towards a central, unified distribution, rather than skewing them towards the source domain distribution alone.

Table 1: Ablation study of the choice of the model that computes the source prototypes.

Method	Source Prototypes computed with	MS → M	MS → C
Fed-Protoid	Pseudo-client	43.1	23.6
	Global model	**51.0**	**23.8**
Fed-Protoid++	Pseudo-client	60.5	43.7
	Global model	**61.7**	**43.8**

2.2 The impact of the number of source prototypes

The following experiments aim to assess the effectiveness of Fed-Protoid in scenarios with stronger memory and communication limitations. Such situations occur when a large source domain with many identities is deployed in the server, resulting in an increased number of prototypes, thus requiring higher communication bandwidth. For instance, the synthetic RP dataset contains $8,000$ identities which is 8 times the number of identities in the MS dataset. To address this challenging scenario, we propose investigating whether a simple uniform

sub-sampling of prototypes reduces the transmission cost without impacting the ReID performance. We evaluate Fed-Protoid with a reduced number of source prototypes in both configurations: RP→M and RP→C. Fig. 1 shows that Fed-Protoid remains effective despite a significant decrease in the number of source prototypes for the MMD optimization on edge devices. Fed-Protoid shows a stable performance when the number of source prototypes varies between 2% and 100% in RP→M and between 20% and 10% in RP→C configuration. Therefore, we argue that Fed-Protoid is robust against device storage and communication limitations.

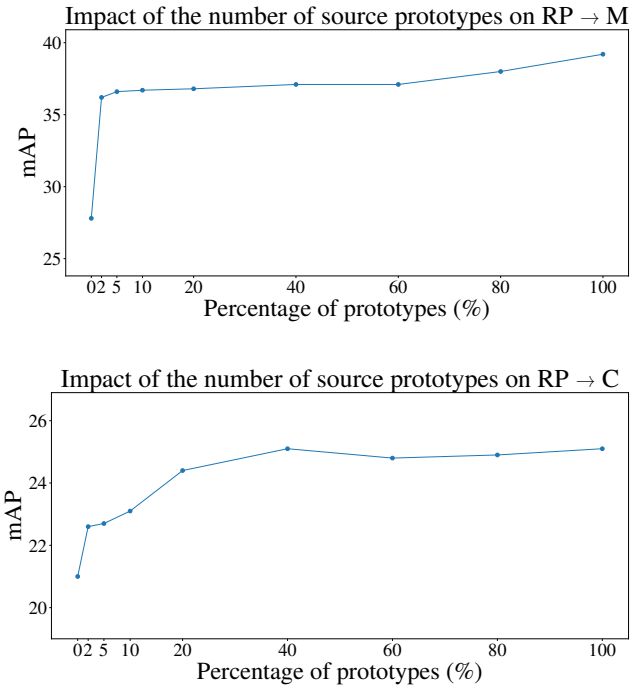


Fig. 1: The impact of the number of source prototypes in the Fed-Protoid performance in two configurations: RP→M and RP→C

3 Variability of Fed-Protoid and hyper-parameters

3.1 Variability of the performance of Fed-Protoid across different initialization

We conduct multiple experiments of our Fed-Protoid and Fed-Protoid++ with three different seeds in all the configurations presented in the main paper. We

report in Table 2 the mean of the mAP and Rank-1 across those runs. Alongside these metrics, we also report the standard deviation to illustrate the variability in the results. We can state that we have consistency and minimal variance across all the different configurations, demonstrating the robustness and reliability of our method under different initializations.

Table 2: Standard deviation of both Fed-Protoid and Fed-Protoid++ with varying seeds.

Configuration	**Fed-Protoid**		**Fed-Protoid++**	
	mAP	**Rank-1**	**mAP**	**Rank-1**
MS → M	$51.0_{\pm 0.3}$	$76.8_{\pm 0.2}$	$61.7_{\pm 0.2}$	$82.6_{\pm 0.1}$
MS → C	$23.8_{\pm 0.2}$	$23.1_{\pm 0.1}$	$43.8_{\pm 0.2}$	$42.4_{\pm 0.4}$
RP → M	$39.2_{\pm 0.1}$	$66.4_{\pm 0.0}$	$45.2_{\pm 0.1}$	$71.8_{\pm 0.1}$
RP → C	$25.1_{\pm 0.4}$	$24.7_{\pm 0.3}$	$25.7_{\pm 0.2}$	$24.9_{\pm 0.1}$

3.2 Hyper-parameters ablation study

Fig. 2 illustrates the results of the sensitivity analysis conducted on the hyper-parameters of Fed-Protoid. In this analysis, all hyper-parameters except the specific one under investigation are maintained at their default values. We observe that changing the hyper-parameters β_1 and γ_1 results in a slight impact on the accuracy with only minimal variances, showing that our method is stable and robust. As for λ, which is the hyper-parameter controlling the importance of the MMD loss in the final objective, we determined that a value of 0.1 yields optimal results and have therefore set it to this fixed value for subsequent experiments.

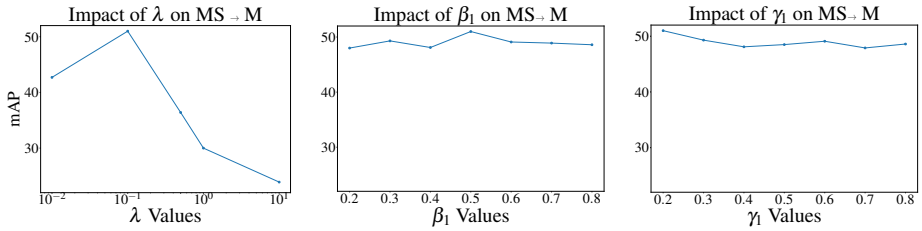


Fig. 2: Ablation study on the sensibility of the different hyper-parameters of Fed-Protoid.

4 Distributed MMD vs. Original MMD

Tab. 3 compares the distributed MMD with the original MMD. For the original MMD, we use the same DUDA-Rid setting, but the MMD loss is computed over the entire target dataset. The results show that the distributed MMD outperforms the original MMD in both configurations. Note that the original MMD violates DUDA-Rid privacy constraints, making it unsuitable for our problem.

Table 3: Comparison between original and distributed MMD.

Method	**MS → M**		**RP → M**	
	mAP	**Rank-1**	**mAP**	**Rank-1**
Fed-Protoid + orig. MMD	47.1	75.3	30.2	59.2
Fed-Protoid + dist. MMD (ours)	**51.0**	**76.8**	**39.2**	**66.4**

5 Comparison with DG and additional experiments.

Tab. 4 includes additional results from two SOTA methods in DG Re-ID: TransMatcher [26] and PAT [38]. We compare these methods with Fed-Protoid (ViT) presented in Tab. 2 of the main paper. In all configurations, Fed-Protoid (ViT) outperforms the other methods. These results are further improved using Fed-Protoid++, which incorporates the LUP large-scale dataset during pre-training instead of being initialized by ImageNet.

Table 4: Comparison between Fed-Protoid and DG methods.

Method	**Type**	**MS → M**		**MS → C**	
		mAP	**Rank-1**	**mAP**	**Rank-1**
TransMatcher [26]	DG	52.0	80.1	22.5	23.7
PAT [38]	DG	47.3	72.2	25.1	24.2
Fed-Protoid (ViT) (ours)	UDA	52.4	80.6	27.5	26.6
Fed-Protoid++ (ours)	UDA	**61.7**	**82.6**	**43.8**	**42.4**