FedPD: Federated Open Set Recognition with Parameter Disentanglement

Chen Yang[1] Meilu Zhu[1] Yifan Liu[2] Yixuan Yuan[1,2*]
[1]City University of Hong Kong [2]The Chinese University of Hong Kong
{cyang53, meiluzhu2}-c@my.cityu.edu.hk 1155195605@link.cuhk.edu.hk yxyuan@ee.cuhk.edu.hk

Abstract

*Existing federated learning (FL) approaches are deployed under the unrealistic closed-set setting, with both training and testing classes belong to the same set, which makes the global model fail to identify the unseen classes as 'unknown'. To this end, we aim to study a novel problem of federated open-set recognition (FedOSR), which learns an open-set recognition (OSR) model under federated paradigm such that it classifies seen classes while at the same time detects unknown classes. In this work, we propose a parameter disentanglement guided federated open-set recognition (**FedPD**) algorithm to address two core challenges of FedOSR: cross-client inter-set interference between learning closed-set and open-set knowledge and cross-client intra-set inconsistency by data heterogeneity. The proposed FedPD framework mainly leverages two modules, i.e., local parameter disentanglement (LPD) and global divide-and-conquer aggregation (GDCA), to first disentangle client OSR model into different subnetworks, then align the corresponding parts cross clients for matched model aggregation. Specifically, on the client side, LPD decouples an OSR model into a closed-set subnetwork and an open-set subnetwork by the task-related importance, thus preventing inter-set interference. On the server side, GDCA first partitions the two subnetworks into specific and shared parts, and subsequently aligns the corresponding parts through optimal transport to eliminate parameter misalignment. Extensive experiments on various datasets demonstrate the superior performance of our proposed method.*

1. Introduction

Deep learning algorithms rely on the availability of large-scale data to achieve remarkable performance. However, in reality, data is scattered across different organizations and difficult to integrate into a centralized dataset, owing to increasing privacy and ethical concerns, especially for those sensitive data such as location-based services or health information [22]. To break this dilemma, federated learning (FL) [5, 23, 20] provides a privacy-preserving paradigm that allows local clients to collaboratively train a shared global model without data sharing.

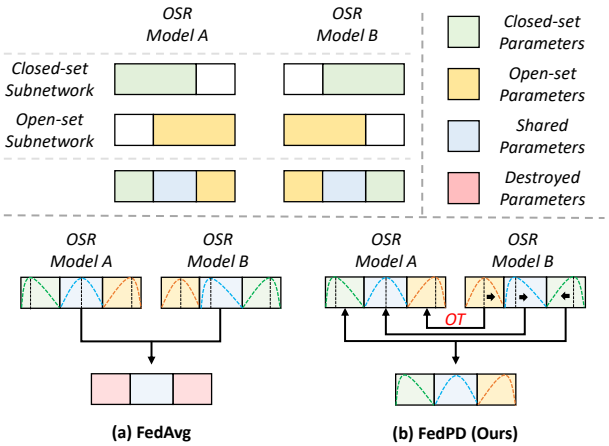

Figure 1. Parameter disentanglement on FedOSR models and comparison of various aggregation strategies on FedOSR setting. (a) **FedAvg** simply aggregates multiple OSR models, leading to model collapse; (b) **Our FedPD** first aligns corresponding close-specific, open-specific and shared parts by optimal transport (OT), then aggregates aligned parameters. The curves inside the network box represent different parameter distribution of each parts.

Although FL has recently achieved promising progress, existing FL works [23, 20, 5] are generally evaluated in a closed-set scenario, where the categories of training and testing samples are identical. The closed-set setting is irrational since unknown classes may appear at the test time and would be classified into known classes. This problem seriously impedes the deployment of FL models in many real-world applications due to enormous risk, such as clinical diagnosis and autonomous driving. Current open-set recognition (OSR) methods [3, 43, 2, 6] attempt to improve the

*Yixuan Yuan is the corresponding author.
This work was supported by Hong Kong Research Grants Council(RGC) General Research Fund 1422062214204321, and Innovation and Technology Commission-Innovation and Technology Fund ITS/100/20.

ability of models in recognizing unknown classes, but they are designed for the centralized setting. In this work, we represent the first effort to formulate a challenging and unexplored problem of Federated Open Set Recognition (FedOSR). FedOSR aims to unite multiple distributed clients to learn a global model and reduce privacy as well as security risk, which not only exactly classifies known classes but also recognizes unknown classes in the testing stage.

Directly applying existing OSR methods into the FL setting for FedOSR mainly undergoes two troublesome challenges. The first challenge lies in the cross-client inter-set interference between learning closed-set and open-set knowledge. According to the previous study [34], the partial parameters of a client model are in charge of learning knowledge of known classes, and the rest are related to open set. The known classes-related parameters of a client is probably polluted by the open set-related parameters from other clients after server communication, leading to the performance degradation on closed set. Similarly, open set-related parameters of a client are also affected by closed-set knowledge of other clients. In this situation, a unknown samples would be possibly misclassified into known classes. The second one is cross-client intra-set inconsistency by data heterogeneity. Even though we aggregate corresponding closed-related parameters of OSR models from different clients, these parameters are still misaligned due to the permutation invariance property of neural networks and data heterogeneity [37]. Aggregation of local client parameters directly at the server can result in inconsistent models among the clients, leading to significant divergence of client models. This inconsistency issue can cause slow and unstable convergence [19], ultimately resulting in sub-optimal performance of the entire FL system [17, 37].

To achieve FedOSR, we conquer these intractable challenges from a new perspective, i.e., parameter disentanglement. Based on the lottery ticket hypothesis [7, 11, 36], we divide parameters of a client model into a closed-set subnetwork and an open-set subnetwork. These two subnetworks have their own specific parameters, which are only related to known classes and unknown classes respectively. Meanwhile, they also share partial parameters since known and unknown samples might have some similar patterns [35]. The parameter disentanglement of client models can preserve the high performance of closed set by reducing the interference from open-set subnetworks. As shown in Fig. 1, the closed-set subnetworks and the open-set subnetworks of different client models distribute in different positions with some overlaps. Directly aggregating all client OSR models by FedAvg [23] on the server side may encounter the parameter misalignment problem and lead to model collapse as shown in Fig. 1 (a). These destroyed parameters are transmitted to clients and slow down the convergence of the federated system due to bad model initialization to the next training step. Therefore, aligning these subnetworks before model aggregation is a crucial step to solve the inconsistency problem of parameter distributions.

To tackle these challenges in FedOSR, we propose a novel parameter disentanglement guided federated OSR (FedPD) algorithm in this paper, which effectively addresses the local parameter misalignment problem occurred on the global model aggregation. Specifically, we design a local parameter disentanglement strategy (LPD) to firstly decouple an OSR model into two subnetworks: an open-set subnetwork and a closed-set subnetwork by task-related metrics. To overcome the parameter misalignment caused by simply parameter averaging on whole client OSR models, we propose a global divide-and-conquer aggregation (GDCA) method to firstly divide two subnetworks into specific parts and shared parts, then align corresponding parameter components by optimal transport [13, 30] and aggregate them. As shown in Fig. 1 (b), our FedPD enables reasonable model aggregation and reliable global model to boost federated training.

The major contribution of this paper are summarized as follows:

- We address a practical FL problem, namely **Federated Open-Set Recognition (FedOSR)**. To the best of our knowledge, this is the first work to improve the ability of detecting novel category for federated models.
- We propose a novel **Parameter Disentanglement guided Federated algorithm (FedPD)** to solve parameter misalignment problem in FedOSR.
- On the client side, we introduce the Local Parameter Disentanglement (LPD) approach, which leverages task-related importance on model parameters to decouple the local OSR model into a closed-set subnetwork and an open-set subnetwork.
- On the server side, we design a Global Divide-and-Conquer Aggregation (GDCA) strategy to partition the two subnetworks into specific and shared parts, align the corresponding parts via optimal transport, and subsequently fuse them to alleviate the misalignment problem in FedOSR.

2. Related Work

2.1. Federated Learning

Federated learning [23, 20, 38, 37, 19, 29, 24, 42] provides a promising privacy-preserving solution for multi-site data collaboration, which develops a global model from decentralized datasets by aggregating the parameters of each local client while keeping data locally. Representatively, McMahan [23] proposed the popular FedAvg algorithm

for communication-efficient federated training of deep networks. There are two lines to improve FedAvg: improvement on local training and on global aggregation.

Regarding improving local training, FedProx [19] introduced a proximal term to the clients' objective, which regulates the local updates to be closer to the initial global model. Meanwhile, MOON [18] proposed a model contrastive loss that corrects local updates by maximizing the agreement of the representation learned by the current local model and the global model, and minimizing the agreement of the representation learned by the current local model and the previous local model.

As for studies on improving the global aggregation phase, FedMA [37] utilizes Bayesian non-parametric methods to match and average weights in a layer-wise manner. To preserve personalization of local clients, FedBN [20] aggregates parameters except BN layers on the server side. Chen [5] proposed to aggregate client model parameters on the frequency domain. Even if these works [37, 21] try to solve parameter misalignment, they are applied to the closed-set recognition task, which can't be directly transferred to open-set recognition due to complex parameter composition in open-set recognition.

2.2. Open Set Recognition

To deploy the classification models to real-world scenario with high stability, open-set recognition (OSR) [40, 41, 33, 3, 43] was proposed to classify known classes while detect unknown classes at the same time. Recent deep learning-based OSR methods can be classified into three categories: discriminative-based models, prototype-based models and generative-based models.

Discriminative model-based methods calibrate the classification logistics to detect unknown samples. Softmax scores are initially utilized to identify out-of-distribution data by argmax thresholding. OpenMax [2] improves softmax scores with an OpenMax layer and fits outputs probabilities with Weibull distributions.

Prototype-based methods [4, 3, 26] apply prototype learning to identify unknown samples on the feature space. ARPL [3] enhanced prototype learning with generated fake samples to achieve prediction-level and feature-level detection. Even if prototype-based methods show outstanding performance on open-set recognition, they are not suitable to be applied on FL since the uploaded prototype may cause leakage of privacy.

Generative model-based methods generate unknown samples using GANs [10] and autoencoders [1] to help the classifier learning the decision boundary between known and unknown distributions. OSRCI [27] utilized GAN [10] architecture to generate counterfactual examples. PROSER [43] set up the open space between class boundaries to keep classes far from each other based on manifold mixup. Generally, there are a closed-set loss based on supervision from known samples and an open-set loss by generated unknown samples or boundary constrains [9, 32].

3. Problem Definition

We begin with formal definition of Federated Learning (FL) and Open-Set Recognition (OSR). Then we define Federated Open-Set Recognition (FedOSR) and its challenges.

Open-Set Recognition: In the standard open-set recognition, the model is trained with a labelled closed training set $\mathcal{D}_{train} = \{(x_i, y_i)\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{C}$, where $\mathcal{X}$ is the input images and $\mathcal{C}$ is the set of 'known' classes. On the testing phase, the testing set $\mathcal{D}_{test} = \{(x_i, y_i)\}_{i=1}^{M} \subset \mathcal{X} \times (\mathcal{C} \cup \mathcal{U})$ contains both seen classes $\mathcal{C}$ and unseen classes $\mathcal{U}$. In addition to returning the distribution $p(y|x, y \in \mathcal{C})$ over known classes, the model also returns a score $\mathcal{O}(y \in \mathcal{C}|x)$ to indicate whether or not the test sample belongs to any of the known classes. Since generative model-based approaches shows superior performance, we utilize these methods as our baseline. There are two loss components of generative model-based approaches: closed-set loss based on supervision from known samples and an open-set loss by generated unknown samples or boundary constrains:

$$\mathcal{L}_{cls} = \mathcal{L}_{close} + \lambda \cdot \mathcal{L}_{open}, \tag{1}$$

where $\mathcal{L}_{close}$ is the cross entropy loss between model prediction and known ground truth, $\mathcal{L}_{open}$ is to constrain open space or generated unknown samples.

Federated Open-Set Recognition: We then extend conventional open-set recognition to Federated Open-Set Recognition (FedOSR). Given K local clients $\{\mathcal{S}^l\}_{l=1}^{K}$ with the same known classes $\mathcal{C}$ and a global central server G, for the federated round t, every client $\mathcal{S}^k$ will receive the same global model weights G_{t-1} from the central server and update the model with their local data $\mathcal{D}_k$ for E epochs. The central server then collects the local parameters $\mathcal{S}_t^k$ from all clients and aggregates them to update the global model G_t. This process repeats until the global model converges. In this paper, we consider the most popular federated averaging algorithm (FedAvg) [23], which aggregates the local parameters with weights of each local dataset to update the global model $G = \frac{1}{K}\sum_{k=1}^{K} \mathcal{S}^k$.

Challenges: Based on the conclusion that closed-set ability is related to certain parameters of an OSR model [34], the simple merging of OSR models may result in the mixing of closed-related parameters from one client with closed-unrelated parameters from other clients, thereby rendering the related parameters ineffective. Furthermore, even if we aggregate the corresponding closed-related parameters of different OSR models, they may still be misaligned due to the inconsistent distribution of locations. These two challenges pose difficulties in globally aggregating OSR models.

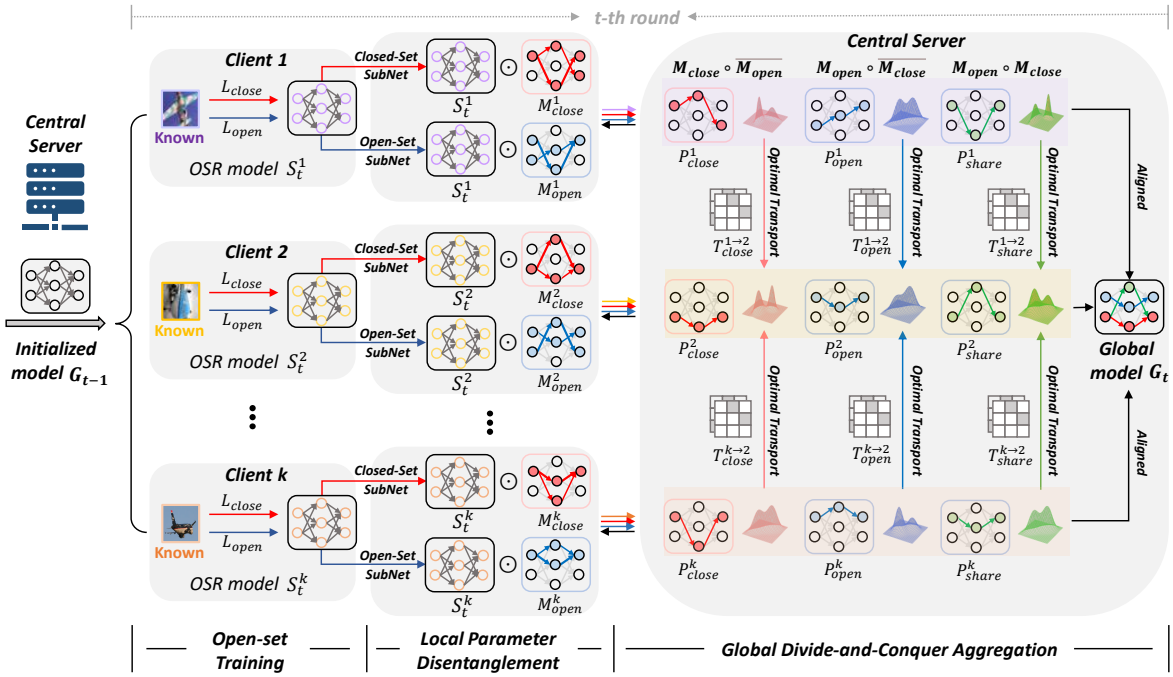

Figure 2. Framework of the proposed FedPD. It consists of a local parameter disentanglement strategy (LPD) and a global divide-and-conquer aggregation approach (GDCA). On the client side, an OSR model is generated with closed-set loss and open-set loss on samples of known classes. Then, LPD decouples the local OSR model into a closed-set subnetwork and an open-set subnetwork by task-related parameter importance. On the server side, GDCA first extracts close-specific, open-specific and shared parameters from the uploaded subnetworks, and then align the corresponding parts by optimal transport and fuse them to generate global model.

4. The Proposed FedPD

The overview of our method is depicted in Fig. 2. To address the FedOSR requirements, our method solves parameter misalignment via a local parameter disentanglement (LPD) strategy (Section 4.1) and a global divide-and-conquer aggregation (GDCA) approach (Section 4.2). For the federated round t, every client $\mathcal{S}^k$ will receive the same global model weights G_{t-1} from the last round and update the model with their local known data $\mathcal{D}_k$ by a closed-set loss and an open-set loss. The LPD then decouples the updated local OSR model S_t^k into a closed-set subnetwork $\mathcal{M}_{close}^k$ and an open-set subnetwork $\mathcal{M}_{open}^k$. The local model $\mathcal{S}^k$ together with two subnetworks $\mathcal{M}_{close}^k$ and $\mathcal{M}_{open}^k$ are uploaded to the server for global divide-and-conquer aggregation. Specifically, the central server first divides two subnetworks into specific parts $\mathcal{P}_{close}^k$, $\mathcal{P}_{open}^k$ and shared part $\mathcal{P}_{share}^k$, then aligns corresponding parts of all clients by optimal transport. At last, the aligned models are averaged to generate the global model G_t.

4.1. Local Parameter Disentanglement

To address the parameter misalignment problem caused by cross-client inter-set inference, we propose a local parameter disentanglement (LPD) strategy to analyze parameter components in FedOSR and decouple an OSR model into a closed-set subnetwork and an open-set subnetwork. Specifically, motivated by the lottery ticket hypothesis [7, 11, 36] which shows that only partial parameters are significant for generalization, we find that partial parameters of an OSR model are important to closed-set classification and some parameters are significant to open-set detection.

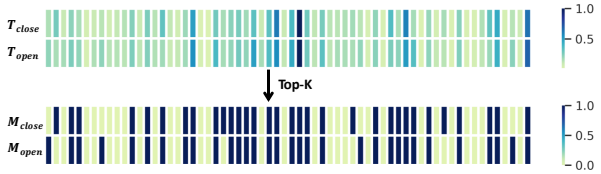

Figure 3. Parameter distribution of closed-set subnetwork and open-set subnetwork on an OSR model on MNIST dataset.

The Lottery Ticket Hypothesis We first review the lottery ticket hypothesis [7, 11, 36], which generates subnetwork to achieve better generalization. For subtask $\mathbf{t}$, if parameter h_i is essential to it, the change of loss would be large once we remove h_i (i.e., $h_i = 0$) [25]. We define the difference value $\Omega^{\mathbf{t}}$ to represent the importance score of the parameter as shown in Eq. 2.

$$\Omega^{\mathbf{t}}(h_i) = |\mathcal{L}^{\mathbf{t}}(H, h_i = 0) - \mathcal{L}^{\mathbf{t}}(H, h_i)|, \tag{2}$$

where H refers to other parameters except h_i. Since it's inefficient to evaluate the importance by parameter traversal,

we approximate it with Taylor Expansion [39] and obtain:

$$\Omega^{\mathbf{t}}(h_i) = |\frac{\partial \mathcal{L}^{\mathbf{t}}(H, h_i)}{\partial h_i} h_i| = |\nabla \mathcal{L}_{\mathbf{t}}(h_i) \times h_i|. \quad (3)$$

After deriving the importance score of parameters for subtask $\mathbf{t}$ based on input $(\mathcal{X}_{\mathbf{t}}, \mathcal{Y}_{\mathbf{t}})$, parameters with the higher score are selected as the subnetwork for $\mathbf{t}$. It can be indicated by a mask M_t, where $M^{\mathbf{t}}(h_i) = 1$ if h_i belongs to the subnetwork, and $M^{\mathbf{t}}(h_i) = 0$ otherwise.

Parameter Disentanglement on OSR Based on the lottery ticket hypothesis, we apply parameter disentanglement to decouple an OSR model into closed-set subnetwork and open-set subnetwork. Specifically, given a local OSR model S^k, we define two task-related metric $\mathcal{T}_{close}$ and $\mathcal{T}_{open}$ to judge the importance of each parameter contributing to closed-set classification and open-set detection based on the Eq. 3:

$$\mathcal{T}^k_{close}(i) = |\nabla \mathcal{L}_{close}(\omega_i) \times \omega_i|, i \in [m_*]\,. \quad (4)$$

$$\mathcal{T}^k_{open}(i) = |\nabla \mathcal{L}_{open}(\omega_i) \times \omega_i|, i \in [m_*]\,. \quad (5)$$

where m_* is the parameter number of a module in an OSR network and w_i is the corresponding parameters of the module. The larger the value of $\mathcal{T}(i)$ is, the more this parameter contributes to the task-related loss function.

After deriving the importance score $\mathcal{T}_{close}$, $\mathcal{T}_{open}$ of parameters for closed-set loss and open-set loss, we choose the top-K highest scores as most valuable weights and set them as 1 with the rest as 0 to generate closed-set subnetwork and open-set subnetwork respectively:

$$\mathcal{M}^k_{close}(i) = \begin{cases} 1, \mathcal{T}^k_{close}(i) > \delta_{close} \\ 0, otherwise \end{cases} \quad (6)$$

$$\mathcal{M}^k_{open}(i) = \begin{cases} 1, \mathcal{T}^k_{open}(i) > \delta_{open} \\ 0, otherwise \end{cases} \quad (7)$$

where δ_{close} and δ_{open} are the threshold to filter out redundant parameters and we choose the threshold based on ratio of parameter numbers. Here we set the masking ratio as 0.5 based on our experimental observation.

We visualize the subnetworks in the first convolution layer as illustrated in Fig. 3. It's obvious that open-set and closed-set subnetworks hold different distribution.

Parameter Disentanglement on FedOSR Given a set of local OSR models $\{\mathcal{S}^k\}^K_{k=1}$, we apply parameter disentanglement on these models, and plot closed-set subnetworks $\{\mathcal{M}^k_{close}\}^K_{k=1}$ and open-set subnetworks $\{\mathcal{M}^k_{open}\}^K_{k=1}$ as shown in Fig. 4. It illustrates that there exists parameter misalignment in both closed-set subnetworks and open-set subnetworks among these clients. Simply aggregating the client OSR modes into one global model may ignore the complex parameter composition and lead to model collapse due to unmatched averaging. For example, the parameter averaging on a closed-set subnetwork and an open-set subnetwork on the same position may generate chaotic neuron weights. This phenomenon motivates us to develop a new model aggregation approach for FedOSR.

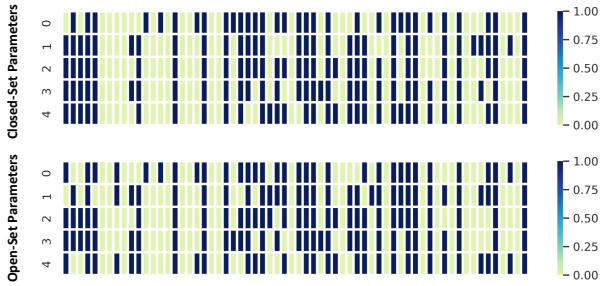

Figure 4. Parameter distribution of closed-set parameters and open-set parameters in federated framework on HDR-FL dataset. The horizontal direction of the picture represents channel numbers, and the vertical of it denotes client numbers.

4.2. Global Divide-and-Conquer Aggregation

During model communication on the server side, parameter misalignment problem will significantly destroy the global model, thus providing bad model initialization for local training in next step. To further remedy the aggregation catastrophe by cross-client intra-set inconsistency, we design a global divide-and-conquer aggregation (GDCA) method to first divide an OSR model into three non-overlapping parts: close-specific parameters, open-specific parameters and shared parameters, and then align and aggregate the corresponding parts respectively.

Learning to divide Based on the results of local parameter disentanglement, an OSR model can be decoupled into a closed-set subnetwork and an open-set subnetwork as shown in Fig. 4. Since there are overlaps between these two subnetworks, we further divide an OSR model into three non-overlapping parts: close-specific parameters, open-specific parameters and shared parameters.

$$\mathcal{P}_{close} = \mathcal{M}_{close} \odot \overline{\mathcal{M}_{open}}, \quad (8)$$

$$\mathcal{P}_{open} = \mathcal{M}_{open} \odot \overline{\mathcal{M}_{close}}, \quad (9)$$

$$\mathcal{P}_{share} = \mathcal{M}_{open} \odot \mathcal{M}_{close}, \quad (10)$$

where $\overline{\mathcal{M}}$ is the negation of the binary mask $\mathcal{M}$. Compared with two subnetworks, the three non-overlapping parts are more suitable to deal with since the overlapping parameters may cause conflicts after alignment.

Learning to conquer We disentangle the OSR model of each client into three parts and align the corresponding parts before aggregation. The neuron weights are considered as a distribution, and we use optimal transport (OT) to achieve

distribution alignment as shown in Fig. 2. OT is a technique used to solve distribution matching problems by finding a minimal effort solution to transport a given mass of dirt into a given hole. It has been successfully applied to various problems such as domain adaptation and GANs. We compute the transport map layer by layer to achieve alignment between two models $\boldsymbol{W}_A$ and $\boldsymbol{W}_B$.

Taking two models $\boldsymbol{W}_A$ and $\boldsymbol{W}_B$ as an example, we align parameters of $\boldsymbol{W}_A$ on $\boldsymbol{W}_B$ by channel-wise distribution matching. Let us assume that we are at one convolution layer $\boldsymbol{W}^{(\ell)} \in (\mathcal{C}_{in}^{\ell}, \mathcal{C}_{out}^{\ell}, F^{\ell})$ and the previous layers have already been aligned, where $F = k \times k$ is the square of filter size. The transport matrix of the last convolution layer $\boldsymbol{W}^{(\ell-1)} \in (\mathcal{C}_{in}^{\ell-1}, \mathcal{C}_{out}^{\ell-1}, F^{\ell-1})$ is denoted as $\boldsymbol{T}^{(\ell-1)} \in (\mathcal{C}_{out}^{\ell-1}, \mathcal{C}_{out}^{\ell-1})$. Since the output of last layer has been permuted by $\boldsymbol{T}^{(\ell-1)}$, we conduct post-multiplying on current layer with transport map of previous layer such that the order of current layer input $\mathcal{C}_{out}^{\ell-1}$ can match the order of $\mathcal{C}_{in}^{\ell}$. Then the current layer can be transformed as:

$$\widehat{\boldsymbol{W}}_A^{(\ell,\ell-1)} \leftarrow \boldsymbol{W}_A^{(\ell)^\top} \boldsymbol{T}^{(\ell-1)}, \tag{11}$$

where $\widehat{\boldsymbol{W}}_A^{(\ell,\ell-1)}$ is the post-processed layer, and $\boldsymbol{W}_A^{(\ell)}$ is transposed as $(\mathcal{C}_{in}^{\ell}, F^{\ell}, \mathcal{C}_{out}^{\ell})$ to achieve matrix multiplication.

With current layer permuted, we compute the optimal transport map $\boldsymbol{T}^{(\ell)}$ between $\widehat{\boldsymbol{W}}_A^{(\ell,\ell-1)}, \boldsymbol{W}_B^{(\ell)}$, i.e., $\boldsymbol{T}^{(\ell)},\ d \leftarrow \mathrm{OT}(\widehat{\boldsymbol{W}}_A^{(\ell,\ell-1)}, \boldsymbol{W}_B^{(\ell)})$, where d denotes the obtained Wasserstein-distance. We use this transport map $\boldsymbol{T}^{(\ell)}$ to align the neurons weights of the first model ($\boldsymbol{W}_A$) with respect to the second ($\boldsymbol{W}_B$),

$$\widetilde{\boldsymbol{W}}_A^{(\ell,\ell-1)} \leftarrow \boldsymbol{T}^{(\ell)^\top} \widehat{\boldsymbol{W}}_A^{(\ell,\ell-1)}, \tag{12}$$

where $\widetilde{\boldsymbol{W}}_A^{(\ell,\ell-1)}$ is the aligned layer from model $\boldsymbol{W}_A$ to $\boldsymbol{W}_B$. To simplify this alignment process, we define $\widetilde{\boldsymbol{W}}_A = OT(\boldsymbol{W}_A, \boldsymbol{W}_B)$. Through the alignment of model parameters, model $\boldsymbol{W}_A$ will only change the orders of feature channels without affecting the model prediction.

When we align the parameters of the OSR model for all clients, we can choose any client as the target client and align other clients with the target client. For example, we use the second client as the target client as shown in Fig. 2. The alignment process can be denoted as:

$$\widetilde{\mathcal{S}}_{close}^{k} = OT(\mathcal{S}^k \odot \mathcal{P}_{close}^{k}, \mathcal{S}^2 \odot \mathcal{P}_{close}^{2}), \tag{13}$$

$$\widetilde{\mathcal{S}}_{open}^{k} = OT(S^k \odot \mathcal{P}_{open}^{k}, S^2 \odot \mathcal{P}_{open}^{2}), \tag{14}$$

$$\widetilde{\mathcal{S}}_{share}^{k} = OT(\mathcal{S}^k \odot \mathcal{P}_{share}^{k}, \mathcal{S}^2 \odot \mathcal{P}_{share}^{2}). \tag{15}$$

After that, the global model G can be represented as:

$$G = \frac{1}{K}\Big(\sum_{k=1}^{K} \widetilde{S}_{close}^{k} + \sum_{k=1}^{K} \widetilde{S}_{open}^{k} + \sum_{k=1}^{K} \widetilde{S}_{share}^{k}\Big). \tag{16}$$

Algorithm 1 summarizes the FedPD algorithm.

Algorithm 1 FedPD

SERVER OPERATIONS
Inputs: Round number T, Set of clients K
Output: Global OSR model G

for $t = 0, 1, \ldots, T-1$ **do**
  for client $k \in K$ **in parallel do**
    $\mathcal{S}^k, \mathcal{M}_{close}^k, \mathcal{M}_{open}^k \leftarrow$ CLIENTOPERATIONS(G_{t-1})
  end for
  $\mathcal{P}_{close}^k = \mathcal{M}_{close}^k \odot \overline{\mathcal{M}_{open}^k}$ ▷ Eq. 8
  $\mathcal{P}_{open}^k = \mathcal{M}_{open}^k \odot \overline{\mathcal{M}_{close}^k}$ ▷ Eq. 9
  $\mathcal{P}_{share}^k = \mathcal{M}_{close}^k \odot \mathcal{M}_{open}^k$ ▷ Eq. 10
  $\widetilde{S}_{close}^k = OT(S^k \odot \mathcal{P}_{close}^k, S^2 \odot \mathcal{P}_{close}^2)$ ▷ Eq. 13
  $\widetilde{S}_{open}^k = OT(S^k \odot \mathcal{P}_{open}^k, S^2 \odot \mathcal{P}_{open}^2)$ ▷ Eq. 14
  $\widetilde{S}_{share}^k = OT(S^k \odot \mathcal{P}_{share}^k, S^2 \odot \mathcal{P}_{share}^2)$ ▷ Eq. 15
  $G_t \leftarrow \frac{1}{K}(\sum_{k=1}^{K} \widetilde{S}_{close}^k + \sum_{k=1}^{K} \widetilde{S}_{open}^k + \sum_{k=1}^{K} \widetilde{S}_{share}^k)$
end for
Obtain global OSR model G

CLIENT OPERATIONS
Input: Model weights G_{t-1}
Output: Updated local OSR model weights $\mathcal{S}_t^k$,
closed-set subnetwork $\mathcal{M}_{close}^k$, open-set subnetwork $\mathcal{M}_{open}^k$

for *epoch* $e = 0, 1 \ldots, E-1$ **do**
  for batch $\{x, y\} \in D_m$ **do** ▷ Local dataset D_m
    $\mathcal{L}_{local} = \mathcal{L}_{close}(x, y) + \mathcal{L}_{open}(x, y)$
    $\mathcal{T}_{close}^k = |\nabla \mathcal{L}_{close}(\omega) \times \omega|$ ▷ Eq.4
    $\mathcal{T}_{open}^k = |\nabla \mathcal{L}_{open}(\omega) \times \omega|$ ▷ Eq.5
    $\mathcal{M}_{close}^k, \mathcal{M}_{open}^k = TopK(\mathcal{T}_{close}^k, \mathcal{T}_{open}^k)$
    $\mathcal{S}_t^k \leftarrow update(\mathcal{S}_t^k, \mathcal{L}_{local})$ ▷ Gradient descent
  end for
end for
Send model weights $\mathcal{S}_t^k$ and subnetworks $\mathcal{M}_{close}^k, \mathcal{M}_{open}^k$ to the server

5. Experiment

5.1. Datasets and Evaluation

We conduct extensive experiments on both heterogeneous federated learning benchmark Handwritten digital recognition FL Dataset (HDR-FL) and homogeneous federated learning benchmark CIFAR-10. The closed-set classification and open-set detection performances are evaluated by accuracy (ACC) and AUROC (AUC) respectively.

HDR-FL: It consists of five datasets: MNIST [16], SVHN [28], USPS [12], SynthDigits [8] and MNIST-M [8]. These datasets are 10-class handwritten digital images from various scenarios. Each dataset is set as a client for Non-IID FedOSR. To achieve open-set recognition, six classes are chosen to be known and four classes are to be unknown classes. We keep the same known classes and unknown classes for all clients.

CIFAR-10: It contains 60000 images in 10 classes, with 6000 images per class [15]. We first divide 10 classes into known classes and unknown classes, then split them into five equal parts to construct homogeneous federated setting. Specifically, we try different ratio between the known and the unknown to validate our method ($e.g.$ 4:6, 6:4 and 8:2).

Table 1. Performance comparisons between our method and other baseline methods on HDR-FL benchmark.

Methods	Closed-set ACC						Open-set AUC					
	MNIST	SVHN	USPS	Synth	MNIST-M	Avg	MNIST	SVHN	USPS	Synth	MNIST-M	Avg
SoftMax	96.69	76.06	97.97	83.99	83.69	87.68	77.38	65.41	88.80	70.53	66.24	73.67
OpenMax[2]$_{(CVPR'16)}$	95.54	63.13	97.97	81.15	76.57	82.87	77.78	57.37	89.21	69.78	60.88	71.00
RPL[4]$_{(ECCV'20)}$	98.33	77.22	98.13	84.76	86.32	88.95	77.87	66.70	89.53	73.17	73.23	76.10
PROSER[43]$_{(CVPR'21)}$	98.04	75.81	97.20	86.92	84.93	88.58	83.63	65.57	90.28	71.76	69.75	76.20
ARPL[3]$_{(TPAMI'21)}$	97.17	69.83	96.44	84.64	84.22	86.46	85.65	59.79	92.53	69.70	68.17	75.16
DIAS[26]$_{(ECCV'22)}$	97.50	70.66	97.62	86.11	86.81	87.74	82.90	67.33	90.69	77.48	71.92	78.06
SSB[35]$_{(ICLR'22)}$	96.41	60.12	97.46	82.82	74.86	82.33	88.53	57.68	90.45	73.40	70.01	76.01
FedPD$_{(Ours)}$	**98.73**	**78.06**	**98.56**	**89.32**	**90.14**	**90.96**	**90.98**	**69.46**	**93.31**	**79.43**	**73.64**	**81.36**

Table 2. Performance comparisons between our method and other baseline methods on CIFAR-10 benchmark.

Method	**Known=4**		**Known=6**		**Known=8**	
	Closed-set ACC	Open-set AUC	Closed-set ACC	Open-set AUC	Closed-set ACC	Open-set AUC
SoftMax	83.23 ±0.28	65.70±0.14	83.14±0.38	72.00±0.44	72.17±0.41	61.11±0.45
OpenMax[2]$_{(CVPR'16)}$	83.24 ±0.08	65.96±0.14	84.56±0.24	81.69±0.59	72.72±0.40	61.52±0.38
RPL[4]$_{(ECCV'20)}$	81.23 ±0.11	65.10±0.05	78.78±0.43	68.72±0.43	72.98±0.33	61.36±0.32
PROSER[43]$_{(CVPR'21)}$	84.15 ±0.13	69.04±0.07	85.77±0.48	80.69±0.27	70.50±0.21	60.48±0.41
ARPL[3]$_{(TPAMI'21)}$	83.91 ±0.09	69.02±0.12	86.54±0.53	79.83±0.69	73.63±0.37	66.78±0.64
DIAS[26]$_{(ECCV'22)}$	84.85 ±0.04	70.32±0.09	87.74±0.16	81.66±0.25	74.53±0.37	67.75±0.34
SSB[35]$_{(ICLR'22)}$	84.27±0.06	68.85±0.11	86.04±0.70	82.41±0.33	75.54±0.18	67.97±0.23
FedPD$_{(Ours)}$	**86.28±0.07**	**71.50±0.07**	**89.43±0.22**	**85.07±0.50**	**75.87±0.16**	**69.12±0.45**

5.2. Implementation Details

On local training, we apply PROSER [43] for open-set training with a closed-set loss and an open-set loss. On global aggregation, we utilize FedAvg [23] to average OSR models for comparison methods. For handwritten digital recognition, we apply a six-layer CNN. During the training process, we utilize the SGD optimizer [31] with learning rate 10^{-2} for closed-set loss and 10^{-4} for open-set loss, we set batch size to 32 and training epochs to 100. Global model is updated every epoch by FedAvg [23] aggregation. For CIFAR-10 Dataset, we use WideResNet for classification. Networks are trained by Adam optimizer [14] with batch size of 128. The learning rates of closed-set loss and open-set loss are initialized as 10^{-1} and 10^{-3} respectively. The communication is conducted after every E = 5 epochs in local training until reaching T = 250 epochs in total. All experiments of these two benchmarks are performed on NVIDIA 2080Ti card with Pytorch library. Detailed model architecture for both benchmarks is shown in the supplementary material.

5.3. Comparison with state-of-the-arts

We compare the performance of FedPD with the state-of-the-art OSR methods, including SoftMax, OpenMax [2], RPL [4], PROSER [43], ARPL [3], DIAS [26], SSB [35]. These comparison methods are implemented by FedAvg [23] on each client OSR models. Our FedPD utilizes the popular generative-based method PROSER [43] for local open-set training.

HDR-FL: As shown in Table 1, our FedPD outperforms existing OSR approaches with a large margin not only in closed-set classification but also in open-set detection. In addition, our FedPD achieves consistent improvements on all clients. Specifically, our method can surpass existing approaches with a promising 90.60% average closed-set ACC and 80.78 average open-set AUC, outperforming the state-of-the-art OSR method DIAS [26] with 2.76% in ACC and 2.72% in AUC. It validates that our method could enable better global model aggregation for open-set recognition, which verifies the effectiveness of our divide-and-conquer approach to address parameter misalignment in FedOSR. Moreover, some generate-based methods (e.g. SSB [26]) may encounter serious model collapse problem due to unmatched parameter of local OSR models, leading to 8.17% performance gap in average ACC.

CIFAR-10: Comparison results on CIFAR-10 benchmark are shown in Table 2. To validate the stability of our method, we conduct experiments on different ratios between known classes and unknown classes. In these three setting, our FedPD achieves the best open-set AUC of 71.70%, 85.07% and 69.12%. The consistent performance improvement over different openness demonstrate the effectiveness of our FedPD to promote the ability of detecting novel category for federated models.

5.4. Ablation Analysis of Our Method

5.4.1 Effectiveness of LPD

Table 4. Comparison to different network splitting methods.

Splitting Method	None	Grad	Grad × Weight (Ours)
Closed-set ACC	85.61	89.04	90.96
Open-set AUC	76.19	80.95	81.36

To demonstrate the advantage of local parameter disentanglement, we compare it with no splitting and network splitting by grad as shown in Table 4. It shows that conducting network splitting according to task gradients results in large

Table 3. Ablation study for key components.

Methods	closed-set ACC						Open-set AUC					
	MNIST	SVHN	USPS	Synth	MNIST-M	Avg	MNIST	SVHN	USPS	Synth	MNIST-M	Avg
Baseline + FedAvg	98.04	75.81	97.20	86.92	84.93	88.58	83.63	65.57	90.28	71.76	69.75	76.20
Baseline + FedMA	97.21	60.72	98.22	86.18	85.72	85.61	83.09	66.47	88.81	73.13	69.46	76.19
Baseline + FedPD (Ours)	**98.73**	**78.06**	**98.56**	**89.32**	**90.14**	**90.96**	**90.98**	**69.46**	**93.31**	**79.43**	73.64	**81.36**
Ours w/o Divide	98.39	77.60	98.22	88.65	89.66	90.50	89.56	69.91	92.72	79.05	72.67	80.78
Ours w/o Conquer	97.43	72.74	97.62	85.82	85.22	87.76	84.41	65.78	91.60	72.51	70.81	77.02

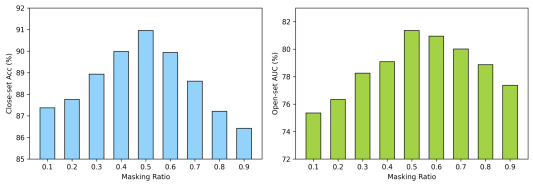

Figure 5. Ablation study for the masking ratio of the parameter disentanglement.

Figure 6. Parameter distribution of closed-set subnetworks and open-set subnetworks on CIFAR-10 dataset.

performance gain. Combining gradient and weight information makes better model decoupling, which corresponds to our theoretical analysis in Eq. 3.

5.4.2 Effectiveness of GDCA

Learning to Divide To validate the advantage of dividing two subnetworks into specific parts and shared parts, we conduct experiments on two different aggregation strategy: align two subnetworks then average (Ours w/o Divide) and align three parts then average (Ours). Comparison results in Table 3 illustrates that aligning three parts outperforms aligning two subnetworks with performance gain of 0.46% in ACC and 0.58% in AUC. It is because that the overlapping parameters between these two subnetworks may have different distribution after alignment, thus leading to ambiguous parameters.

Learning to Conquer To validate the effectiveness of divide-and-conquer aggregation, we compare our method with traditional aggregation methods FedAvg and FedMA. Unlike obvious improvement on closed-set setting, FedMA shows inferior performance than FedAvg as shown in Table 3. It is consistent with our observation that parameter components in open-set models are more complex than those in closed-set models due to multiple optimization directions. Compared with directly averaging three parts with alignment (Ours w/o Conquer), conducting our proposed aggregation strategy brings performance boost on of 3.2% in ACC and 4.34% in AUC, demonstrating that our FedPD can alleviate parameter misalignment problem.

5.4.3 Effect of Masking Ratio

To investigate the effect of masking ratio to the FedOSR performance, we design ablation experiments under HDR-FL setting, as shown in Fig. 5. It is observed that small masking ratio ($e.g.$ 0.1) may filter out too much parameters, leading to empty shared parameter. Large masking ratio ($e.g.$ 0.9) can't choose valuable weights for the closed-set task and the open-set task, thus leading to invalid decoupling. In this paper, we choose suitable masking ratio as 0.5 based on our experimental observation that it achieves the best closed-set ACC and open-set AUC.

5.5. Analysis of Parameter Misalignment

To verify the parameter misalignment problem in FedOSR, we visualize the weights distributions of closed-set subnetwork and open-set subnetwork on CIFAR-10 homogeneous benchmark as shown in Fig. 6. Different clients still holds inconsistent parameter distribution on both closed-set subnetwork and open-set subnetwork. Different from results shown in Fig. 4, where parameter misalignment may come from domain shift in heterogeneous federated datasets. Fig. 6 further emphasize that the parameter misalignment in FedOSR mainly comes from gradient divergence of closed-set loss and open-set loss.

6. Conclusion

In this work, we propose a novel and practical problem of federated open-set recognition (FedOSR) for the first time. To alleviate the parameter misalignment problem in FedOSR, we design a novel parameter disentanglement guided federated algorithm (FedPD). Specifically, on the client side, a local parameter disentanglement is developed to decouple the local OSR models into closed-set subnetworks and open-set subnetworks. On the server side, a global divide-and-conquer aggregation strategy is proposed to divide two subnetworks into specific parts and shared parts, then align corresponding parts by optimal transport and fuse them to generate global model. Extensive experiments on both IID and Non-IID benchmark datasets demonstrate the effectiveness of FedPD.

References

[1] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015. 3

[2] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016. 1, 3, 7

[3] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 1, 3, 7

[4] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *ECCV*, August 2020. 3, 7

[5] Zhen Chen, Meilu Zhu, Chen Yang, and Yixuan Yuan. Personalized retrogress-resilient framework for real-world medical federated learning. In *MICCAI*, pages 347–356. Springer, 2021. 1, 3

[6] Luke Ditria, Benjamin J Meyer, and Tom Drummond. Opengan: Open set generative adversarial networks. In *ACCV*, 2020. 1

[7] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 2, 4

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189. PMLR, 2015. 6

[9] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3614–3631, 2020. 3

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020. 3

[11] Zhongyi Han, Haoliang Sun, and Yilong Yin. Learning transferable parameters for unsupervised domain adaptation. *IEEE Trans. Image Process.*, 2022. 2, 4

[12] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):550–554, 1994. 6

[13] Leonid V Kantorovich. On the translocation of masses. *J. Math. Sci.*, 133(4):1381–1382, 2006. 2

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

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

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[17] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *ICDE*, pages 965–978. IEEE, 2022. 2

[18] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021. 3

[19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020. 2, 3

[20] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *ICLR*, 2021. 1, 2, 3

[21] Chang Liu, Chenfei Lou, Runzhong Wang, Alan Yuhan Xi, Li Shen, and Junchi Yan. Deep neural network fusion via graph matching with applications to model ensemble and federated learning. In *ICML*, pages 13857–13869. PMLR, 2022. 3

[22] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020. 1

[23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 7

[24] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *CVPR*, pages 8397–8406, 2022. 2

[25] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016. 4

[26] WonJun Moon, Junho Park, Hyun Seok Seong, Cheol-Ho Cho, and Jae-Pil Heo. Difficulty-aware simulator for open set recognition. *arXiv preprint arXiv:2207.10024*, 2022. 3, 7

[27] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, pages 613–628, 2018. 3

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6

[29] Jaehoon Oh, SangMook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *ICLR*, 2022. 2

[30] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Found. Trends Mach. Learn.*, 11(5-6):355–607, 2019. 2

[31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 7

[32] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021. 3

[33] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2317–2324, 2014. 3

[34] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *ECCV*, 2022. 2, 3

[35] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: a good closed-set classifier is all you need? In *ICLR*, 2022. 2, 7

[36] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. Exploring domain-invariant parameters for source free domain adaptation. In *CVPR*, pages 7151–7160, 2022. 2, 4

[37] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ICLR*, 2020. 2, 3

[38] Jiacheng Wang, Yueming Jin, and Liansheng Wang. Personalizing federated medical image segmentation via local calibration. *arXiv preprint arXiv:2207.04655*, 2022. 2

[39] Runxin Xu, Fuli Luo, Baobao Chang, Songfang Huang, and Fei Huang. S4-tuning: A simple cross-lingual sub-network tuning method-tuning: A simple cross-lingual sub-network tuning method. In *ACL*, pages 530–537, 2022. 5

[40] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, pages 4016–4025, 2019. 3

[41] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *ECCV*, pages 102–117. Springer, 2020. 3

[42] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *ICLR*, 2020. 2

[43] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *CVPR*, pages 4401–4410, 2021. 1, 3, 7