Federated Adversarial Debiasing for Fair and Transferable Representations

Junyuan Hong
hongju12@msu.edu
Michigan State University
East Lansing, Michigan, USA

Zhuangdi Zhu
zhuzhuan@msu.edu
Michigan State University
East Lansing, Michigan, USA

Shuyang Yu
yushuyan@msu.edu
Michigan State University
East Lansing, Michigan, USA

Zhangyang Wang
atlaswang@utexas.edu
University of Texas at Austin
Austin, Texas, USA

Hiroko Dodge
dodgeh@ohsu.edu
Oregon Health & Science University
Portland, Oregon, USA

Jiayu Zhou
jiayuz@msu.edu
Michigan State University
East Lansing, Michigan, USA

ABSTRACT

Federated learning is a distributed learning framework that is communication efficient and provides protection over participating users' raw training data. One outstanding challenge of federate learning comes from the users' heterogeneity, and learning from such data may yield biased and unfair models for minority groups. While adversarial learning is commonly used in centralized learning for mitigating bias, there are significant barriers when extending it to the federated framework. In this work, we study these barriers and address them by proposing a novel approach Federated Adversarial DEbiasing (FADE). FADE does not require users' sensitive group information for debiasing and offers users the freedom to opt-out from the adversarial component when privacy or computational costs become a concern. We show that ideally, FADE can attain the same global optimality as the one by the centralized algorithm. We then analyze when its convergence may fail in practice and propose a simple yet effective method to address the problem. Finally, we demonstrate the effectiveness of the proposed framework through extensive empirical studies, including the problem settings of unsupervised domain adaptation and fair learning.

CCS CONCEPTS

• **Computing methodologies** → **Distributed algorithms; Machine learning**; • **Transfer learning**;

KEYWORDS

Federated learning; Adversarial learning; Unsupervised domain adaptation; Fairness

ACM Reference Format:
Junyuan Hong, Zhuangdi Zhu, Shuyang Yu, Zhangyang Wang, Hiroko Dodge, and Jiayu Zhou. 2021. Federated Adversarial Debiasing for Fair and Transferable Representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447548.3467281

KDD '21, August 14–18, 2021, Virtual Event, Singapore

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467281

1 INTRODUCTION

The last decade witnessed the surging adoption of personal devices such as smartphones, smartwatches, and smart personal assistants. These devices directly interface with the users, collect personal data, conduct light-weighted computations, and use machine learning models to offer personalized services. The challenges from privacy concerns of sensitive personal data, limited computational resources, performance issues of localized learning all together lead to the federated learning (FL) paradigm [4, 34]. FedAvg [32], for example, provides an efficient and privacy-aware FL framework. Users train models locally, upload them to a central server iteratively aggregated to form a global model. FL greatly alleviated privacy concerns because the server can only access model parameters from the users instead of the raw data used for training.

One major challenge of FL comes from the user heterogeneity where users provide statistically different data for training local models [5, 9]. Such heterogeneity may come from different sources. For example, the users may collect data under various conditions according to preferential or usages differences. Consider the learning of handwashing behavior from accelerometers of smartwatches, where patterns can drastically change when using different basins worldwide. Such domain shift [19] can lead to negative impacts during knowledge transfer among users [37]. Another common source of heterogeneity comes from the sensitive group information such as age, gender, and social groups, which are variables typically not to be identified during learning. Heterogeneity from this source is often associated with critical fairness issues [6] after deploying the models, where groups with less resource or smaller computation capability may be biased or even ignored during the learning [29], and the resulting global model may perform worse in minority groups.

Adversarial learning [12] has been a powerful approach to mitigate bias in centralized learning, in which an adversarial objective minimizes the information extracted by an *encoder* that can be maximally recovered by a parameterized model, *discriminator*. For example, it has been applied to disentangle task-specific features that may cause negative transfer [27], to perform unsupervised domain adaptation [11, 42], and recently to achieve fair learning [46].

However, there are significant barriers when applying adversarial techniques in FL: 1) Most existing approaches follow a top-down principle. In the context of FL, the adversarial objective requires the server to access the sensitive group variable (e.g., gender) to construct an adversarial loss. This requirement directly violates the privacy consideration design for FL, and users may not want to disclose their sensitive group variables. 2) adversarial learning demands extra information from users for training the adversarial component and imposes an additional computational burden on smart devices that may not be able to afford. 3) besides, it remains unknown how the introduction of an adversarial component would impact the distributed learning behavior (e.g., convergence property) of FL.

To address the challenges mentioned above, we propose a novel adversarial framework for debiasing federated learning following a bottom-up principle, called *Federated Adversarial DEbiasing (FADE)*. Besides the benefits from typical FL on communication efficiency and data privacy, FADE aims to achieve the following goals:

- **Privacy-Protecting**: The learning algorithm conforms to the privacy design of FL and does not require users' group variable to achieve debiasing w.r.t. the group variable.
- **Autonomous**: A user can choose to join and opt-out from the adversarial component anytime (e.g., due to computational budget or privacy budget) while still participate in the regular federated learning.
- **Satisfiable**: Under above restrictions, the distributed learning should output a debiased and accurate model, despite the user heterogeneity and unpredictable user participation.

To achieve these goals, we first propose a generic algorithm for FADE and show that ideally, it can attain the same global optimality as the one by the central algorithm. We then show how its convergence may fail in practice and propose a simple yet effective method to address the problem. Finally, we demonstrate the effectiveness of the proposed framework through extensive empirical studies on various applications.

2 RELATED WORK

Federated Learning (FL) [32] is a distributed learning framework that allows users with different capabilities to collaboratively train a model without sharing their own data. A critical challenge in FL is the heterogeneity among users. Viewing the learning process of FL as knowledge transfer among different users, heterogeneity in user data leads to negative transfer between users and compromises generalization [3]. One idea to alleviate the negative effect from the heterogeneity during the training, is to find the consensus among users. For example, in [10, 14, 21, 26], the consensus on task knowledge is achieved by distillation. In this work, we seek an alternative and efficient approach by adversarial debiasing the users of different groups.

Adversarial Learning has been widely applied in various domains, such as neural language recognition [27], image-to-image (dense) prediction [31], image generation [12], and etc. Conceptually, adversarial learning aims to solve a two-player (or multi-player) game between two adversarial objectives, which typically leads to a min-max optimization problem. Existing approaches can be briefly categorized as: *1) Sample-to-Sample (S2S)* adversarial learning, where the adversarial objective quantifies the difference between synthetic and real samples. Examples include adversarial learning against adversarial attacks [30] and generative adversarial networks [12]. *2) Group-to-Group (G2G)* adversarial learning, which aims to reduce the max discrepancy (bias) between group distributions, for example, adversarial domain adaptation [11], adversarial fairness [46] and adversarial multi-task learning [27]. All these variants assume the availability of adversarial groups in the same computation node, e.g., by aggregating data in Fig. 1a, and thus cannot be directly extended to federated learning to the violation of privacy design (requiring access of the sensitive group information). A recent effort is done by [38] where embeddings of different groups are shared (see Fig. 1b). Nevertheless, both sharing data and embeddings could induce additional privacy risk and communication costs. The proposed FADE eliminated these requirements, leading to private and efficient distributed collaboration between users/groups.

3 FEDERATED ADVERSARIAL DEBIASING

In this section, we first formulate the proposed Federated Adversarial Debiasing (FADE) framework. We work on the standard federated learning problem setting which learns one model from a set of distributed participating users. Users conduct local learning based on their own data and send the parameters of learning models to a server periodically. The server aggregates the local models to form a global model. We assume the users have non-iid data and each user belongs to one of the E user groups as indicated by a group variable (e.g., age, gender, race) that is not to be shared outside of the local learning.

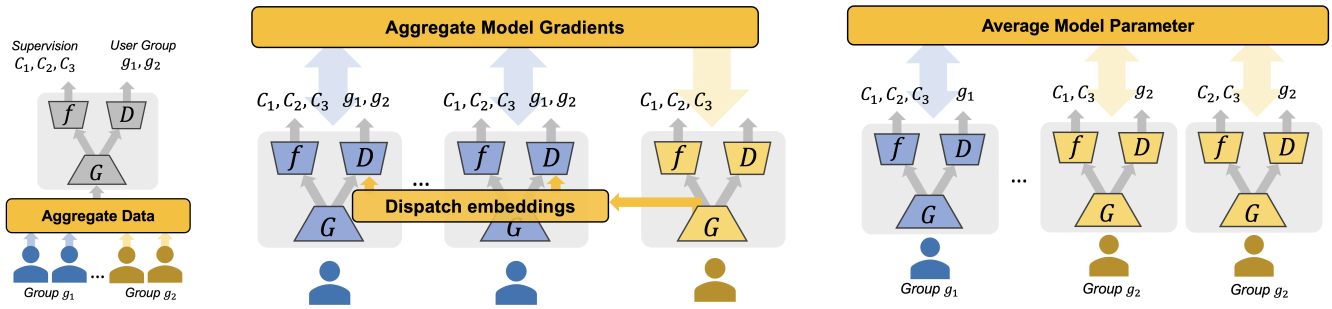
The model of each user consists of three components: a decoder f for the learning task (e.g., classification target), an encoder G, and a group discriminator D, as illustrated in Fig. 1c. In the two-group setting (a data point belongs to either group 0 and 1), D outputs a scalar in $(0, 1)$ approximating the probability of an input data point x belong to the group 0. More generally, for E groups, we use a softmax mapping in the last layer of D which outputs an E-dimensional vector. The FADE objective learns f, D, G by:

$$\min_{f,G} \mathcal{L}(f, G) = \sum\nolimits_{g=1}^{E} \sum\nolimits_{i=1}^{m_g} L_{i,g}(f, G), \tag{1}$$

$$L_{i,g}(f, G) = L_i^{\text{task}}(f, G) + \lambda \max_{D} L_{i,g}^{\text{adv}}(G, D), \tag{2}$$

where $L_i^{\text{task}}(f, G)$ is the task loss for the i-th user, $L_{i,g}^{\text{adv}}(G, D)$ is the adversarial loss, and m_g is the number of users in group g. Note that we absorb the variable model D into $L_{i,g}$ in Eq. (2), and the objective is still an optimization over f, D, G. For classification tasks, the task loss can be defined as $L_i^{\text{task}}(f, G) \triangleq \mathbb{E}_{(x,y)\sim p_i(x,y)}[\mathcal{E}(f(G(x)), y)]$, where $\mathcal{E}$ denotes the cross-entropy loss and p_i is the data distribution of user i. The adversarial loss is defined as $L_{i,g}^{\text{adv}}(G, D) \triangleq \mathbb{E}_{x\sim p_i(x)}[\log D_g(G(x))]$, where $D_g(G(x))$ is the g-th output of the softmax vector. The optimal solution for the min-max problem is the *adversarial balance* when D is unable to tell the difference of $G(x)$ among groups. For the two-group case, the adversarial loss can be modified as:

$$L_{i,g}^{\text{adv}}(G, D) = \mathbb{E}_{x\sim p_i(x)}\big[\mathbb{I}(g = 0) \log D(G(x)) + \mathbb{I}(g = 1) \log(1 - D(G(x)))\big], \tag{3}$$



(a) Central (b) Federated with shared embeddings [38] (c) Federated Adversarial Debiasing (proposed)

Figure 1: Illustrations of different adversarial learning frameworks for debiasing. f, D and G are classifier (task model), discriminator and encoder, respectively. C_1, C_2, C_3 represents the task supervisions, for example, ground-truth classes, in the corresponding users. g_1 and g_2 represents the two groups of users. The encoders are adversarially trained such that the embeddings are informative for distinguishing C_1, C_2, C_3 but not g_1, g_2. The proposed FADE tackles a more challenging problem than other two because of isolated and non-sharing group/user data (or embeddings) and class-wise non-iid users within groups.

where $\mathbb{I}(\cdot)$ is the indicator function.

One fundamental difference between traditional adversarial learning and FADE is that FADE only has one group data in the loss function. Hence, users have no sense of what an adversary (a user from other groups) looks like. Directly optimizing this objective may fail in finding the right direction towards convergence. In the worst case, the optimal solution may not be the adversarial balance. In the next section, we will provide principled analysis to the adversarial balance that is achievable under appropriate conditions.

We summarize the server and user update strategies in Algorithms 1 and 2. The server is responsible for aggregating users' models and dispatching the global models to users. Meanwhile, users train the received global model and the adversarial component using local data. Note that we use the reversal gradient strategy to implement the min-max optimization in Algorithm 1. Our algorithm enjoys the two nice properties:

Autonomous: Different from vanilla FL, FADE allows the users to decide whether or not to join the learning of the discriminator D at each iteration. A user can opt-in the discriminator learning at a low frequency or completely opt-out when privacy becomes a concern or learn with restrictive computational resources. For example, in the adversarial domain adaptation setting [38] where some users have supervision and some others not, some supervised user may not want to help unsupervised users. FADE will significantly reduce the communication cost and privacy risk overhead involved by cutting down the interactions form these users.

Privacy: In the proposed FADE framework, the group label g will be restricted to local learning and the group debiasing is done through the discriminator model D. Thus, users will not be able to obtain the other users' sensitive attributes including the group variable. Moreover, following [33], the privacy of FADE can be strictly protected by directly injecting *Differential-Privacy* noise during the gradient descent procedure.

Algorithm 1 FADE User Update

Input: f, G, D received from server, learning rate η, adversarial parameter λ, user data distribution p_i

$f_0, D_0, G_0 = f, D, G$
for $t = 1, \ldots, K$ **do**
  Sample a batch by $x \sim p_i(x)$ or $(x, y) \sim p_i(x, y)$
  $z \leftarrow G(x)$
  $\nabla_f \leftarrow \frac{\partial L_i^{\text{task}}}{\partial f}$, $\nabla_D \leftarrow \frac{\partial L_i^{\text{adv}}}{\partial D}$
  if adversarial game D is accepted by user i **then**
    $\nabla_G \leftarrow \frac{\partial z}{\partial G}\left(\frac{\partial L_i^{\text{task}}}{\partial z} + \lambda \frac{\partial L_i^{\text{adv}}}{\partial z}\right)$
    $D_{t+1} \leftarrow D_t + \eta \nabla_D$
    $G_{t+1} \leftarrow G_t - \eta \nabla_G$
  else
    $\nabla_G \leftarrow \frac{\partial z}{\partial G} \frac{\partial L_i^{\text{task}}}{\partial z}$
    $D_{t+1} \leftarrow D_t$
    $G_{t+1} \leftarrow G_t - \eta \nabla_G$
  $f_{t+1} \leftarrow f_t - \eta \nabla_f$
Output: $f_{K+1}, G_{K+1}, D_{K+1}$

Algorithm 2 FADE Server Aggregation

Input: Initial models f, D, G, momentum parameter β

for $t \in 1, \cdots, T_{\max}$ **do**
  Select m active users uniformly at random into $\mathcal{A}$
  Broadcast $\theta_t = (f_t, G_t, D_t)$ to m users
  for user i in $\mathcal{A}$ in parallel **do**
    User updates by Algorithm 1
  Aggregate $\{\theta_t^k = (f_t^i, G_t^i, D_t^i)\}_{i=1}^m$ and average

$$\theta_{t+1} \leftarrow \beta \sum_{i=1}^m \frac{n_i}{N} \theta_t^i + (1 - \beta)\theta_t$$

Output: f_t, G_t, D_t

4 OPTIMALITY ANALYSIS

Despite the fact that FADE enables autonomous and improves privacy in learning, it is critical to ask if the algorithm gives a satisfiable solution and what is the optimal solution of Eq. (1). Remarkably, FADE differs from traditional adversarial learning by Eq. (3), where only one group is used to evaluate the adversarial objective. This imposes a unique challenge in learning as it may compromise the convergence of learning. Below we give formal analysis of the optimality when Algorithm 2 is iterated with users from two groups in

non-zero probability. Since most of multi-group adversarial problems can be transformed into two-group problems, we focus on discussing the two-group case for the ease of analysis.

Consider the case when each group only has one user. The data distributions for the two users are p_1 and p_2, respectively. We single out the min-max optimization in Eq. (1) as:

$$\min_G \max_D \mathbb{E}_{p_1}[\log D(G(x))] + \mathbb{E}_{p_2}[\log(1 - D(G(x)))].$$

For simplicity, we denote $G(x)$ by z and slightly abuse $p_1(x)$ by $p_1(z)$ in our discussion. Hence, we can define:

$$\mathbf{D}_{p_1,p_2} = \max_D \mathbb{E}_{p_1}[\log D(z)] + \mathbb{E}_{p_2}[\log(1 - D(z))],$$

which is the maximal discrepancy between $p_1(z)$ and $p_2(z)$ that D can characterize. Now, we can rewrite the min-max problem as $\min_G \mathbf{D}_{p_1,p_2}(G)$ which minimizes the distribution distance over z. Alternatively, we can formulate it by $\min_{p_1,p_2} \mathbf{D}_{p_1,p_2}$ since p_1 and p_2 are parameterized by G.

Because users may participate federated learning at varying frequencies, we use an auxiliary random variable $\xi_i \in \{0, 1\}$ for $i = 0, 1$ to denote whether the user is active for training. We assume ξ_i is subject to the Bernoulli distribution, $B(1, \alpha_i)$. Plug ξ_i into $\mathbf{D}_{p_1,p_2}$ to obtain $\mathbf{D}_{p_1,p_2} = \max_D \mathbb{E}_{p_1}[\xi_1 \log D(z)] + \mathbb{E}_{p_2}[\xi_2 \log(1 - D(z))]$ and take expectation:

$$\begin{aligned}\tilde{\mathbf{D}}_{p_1,p_2} &\triangleq \mathbb{E}_{\xi_1,\xi_2}[\mathbf{D}_{p_1,p_2}] \\ &= \max_D \mathbb{E}_{p_1}[\alpha_1 \log D(z)] + \mathbb{E}_{p_2}[\alpha_2 \log(1 - D(z))]. \end{aligned} \quad (4)$$

Therefore, our problem is transformed as minimizing $\tilde{\mathbf{D}}_{p_1,p_2}$.

Note that with p_1 and p_2 given, the solution of the maximization in $\tilde{\mathbf{D}}_{p_1,p_2}$ is:

$$D^*_{\alpha_1,\alpha_2}(z) = \frac{\alpha_1 p_1(z)}{\alpha_1 p_1(z) + \alpha_2 p_2(z)}, \quad (5)$$

with which we can derive the optimality sufficiency as below.

Theorem 4.1. *The condition $p_1(z) = p_2(z)$ is a sufficient condition for minimizing $\tilde{\mathbf{D}}_{p_1,p_2}$ and the minimal value is $\alpha_1 \log \alpha_1 + \alpha_2 \log \alpha_2 + (\alpha_1 + \alpha_2) \log(\alpha_1 + \alpha_2)$.*

Theorem 4.1 shows that even if some users are inactive, the distribution matching, $p_1 = p_2$, remains a sufficient optimality condition. We remark that the above result can be generalized to multiple users when all users are iid and ξ_i represent the ratio of group i in users. In addition, we notice Theorem 4.1 does not guarantee a stable convergence or exclude other undesired solutions. We discuss these issues in the following.

4.1 The effect of imbalanced groups

Although Theorem 4.1 shows the optimality of the matched distribution, the optimization may still fail to converge especially when one group of users are relatively inactive, e.g., $\alpha_1 \ll \alpha_2$. When $\alpha_1 \ll \alpha_2$ or reverse, we call the situation as *imbalanced groups*. The imbalanced groups happens because the users are free to quit or joint the training. From Eq. (5), we observe that $D^*(x)$ will be less sensitive to changes of $p_1(x)$ if $\alpha_1 \ll \alpha_2$, and vice versa. Meanwhile, $\log D^*(x) \to -\infty$ and $\mathbf{D}_{p_1,p_2}$ approaches the minimum even if p_1 and p_2 are quite different.

Theorem 4.2. *Let ϵ be a positive constant. Suppose $|\log p_1(x) - \log p_2(x)| \le \epsilon$ for any x in the support of p_1 and p_2. Then we have $\tilde{\mathbf{D}}_{p_1,p_2} = O(\alpha_1 \epsilon/(\alpha_1 + \alpha_2))$ when $\alpha_1 \ll \alpha_2$.*

Theorem 4.2 reveals that the imbalance between groups could greatly reduce the sensitivity of the discrepancy ϵ between p_1 and p_2. A less sensitive discriminator will ignore the minor differences between groups. The importance of discrepancy sensitivity for the adversarial convergence was also discussed in [2]. It is easy to see the negative impact of the low sensitivity: **1)** higher communication cost incurs due to more communication rounds are required to check the discrepancy; **2)** the optimization possibly fails to converge due to vanished gradients (scaled by α_1).

4.2 Squared adversarial loss

In Eq. (4), when $\alpha_1 \to 0$ and $\alpha_2 \to 1$, we notice that $\tilde{\mathbf{D}}_{p_1,p_2}$ approaches 0 while $\mathbb{E}_{p_1}[\log D(z)] \to -\infty$. In other words, the large value of $\mathbb{E}_{p_1}[\log D(z)]$ is neglected due to its coefficient α_1. To re-emphasize the value, a heuristic method is to increase the weight when $|\mathbb{E}_{p_1}[\log D(z)]|$ is large. Thus, we propose to replace $L^{\text{adv}}_{i,g}(G, D)$ by:

$$L^{\text{adv}}_{i,g,2}(D, G) = -\frac{1}{2}\left(L^{\text{adv}}_{i,g}(G, D)\right)^2, \quad (6)$$

which we call *squared adversarial loss*. We can write the corresponding discrepancy $\tilde{\mathbf{D}}^{(2)}_{p_1,p_2}$ as:

$$\min_D \alpha_1 \mathbb{E}^2_{p_1}[\log D(z)] + \alpha_2 \mathbb{E}^2_{p_2}[\log(1 - D(z))].$$

Though we derive the squared adversarial loss in a heuristic manner, the loss can be explained in the view of resource-fair federated learning [22]. Because the adversarial objective pays more attention to the frequent group, we can interpret the problem as the unfairness between groups. Following [22], we generalize our adversarial loss function as:

$$L^{\text{adv}}_{i,g,2}(D, G) \triangleq (-1)^{q-1} \frac{1}{q} \mathbb{E}_x\left[\ell^q_k(D, G; x)\right], \quad (7)$$

where $q \ge 1$. If $q = 1$, the loss degrades to the vanilla one.

4.3 The effect of non-iid users

It is well-known that typical federated learning approaches suffer from very heterogeneous users since they sample data from very different distributions. The adversarial objective captured and decreases the group heterogeneity by design. Another kind of heterogeneity is related to the users' tasks. We argue that the heterogeneity is natural and could be essential for the task discriminability but may be accidentally eliminated by adversarial learning. For example, three users are non-iid by three classes. After FADE training, the non-iid users collapse to the similar distributions due to the wrong sense of the group discrepancy.

To prove the existence of user-collapsed solution for FADE, we consider $z \sim p(z|T = t)$, or simply $z \sim p(z|t)$, where t is a discrete hidden variable related to users' tasks. For example, each user has one class of samples in classification tasks. Then t is the corresponding class. In addition, we define $\hat{p}_1(z) = \frac{1}{m}\sum_{t=1}^m p(z|t)$ which is a p.d.f. For simplicity, we assume all users always participate the

learning, i.e., $\alpha_i = 1$ for all users. Hence, we can obtain $\mathbf{D}_{p_1,p_2}$ as

$$\max_D \sum_{t=1}^m \mathbb{E}_{p(z|t)}[\log D(z)] + \mathbb{E}_{p_2}[\log(1 - D(z))]$$
$$= \max_D m\mathbb{E}_{\hat{p}_1(z)}[\log D(z)] + \mathbb{E}_{p_2}[\log(1 - D(z))],$$

whose maximizer is given by: $D^*(z) = \frac{m\hat{p}_1(z)}{m\hat{p}_1(z)+p_2(z)}$. Use similar derivations as in Theorem 4.1, we can show that $\hat{p}_1(z) = p_2(z)$ is a sufficient optimality condition, which implies:

$$\sum_{t=1}^m p(z|t) = mp_2(z). \tag{8}$$

First, we can still obtain $p_1(z)\sum_{t=1}^m p(t|z)/p(t) = mp_2(z)$ from Eq. (8) where we use $p(z|t) = p_1(z)\frac{p(t|z)}{p(t)}$. If $\sum_{t=1}^m \frac{p(t|z)}{p(t)} = m$, then we can get the vanilla solution, $p_1(z) = p_2(z)$.

Except for the vanilla solution, a trivial solution to Eq. (8) is $p(z|t) = p_2(z)$. However, the solution could hurt the task utility since it may eliminate the inherent difference between tasks. For instance, if t represents the classification label, the solution will vanish the discriminability of the representation z. We call the scenario as the *user collapse*. It worth noticing that user collapse could happen even if the p_1 and p_2 are matched.

4.4 Mitigate user collapse

Since there are arbitrarily many solutions to $\sum_{t=1}^m \frac{p(t|z)}{p(t)} = m$, we need to constraint the feasible solutions such that the collapsed solution will be eliminated. Notice $\frac{p(t|z)}{p(t)} = \frac{p(t,z)}{p(z)p(t)}$ is related to the mutual information between t and z. Conceptually, we can modify the adversarial loss to:

$$\hat{L}^{\text{adv}}_{i,g,2}(D, G) = L^{\text{adv}}_{i,g,2}(D, G) + I(G(x); t|i),$$

where $I(G(x); t|i)$ is the mutual information conditioned on user i. Because mutual information is hard to estimate in practice (especially given few samples), we provide some surrogate solutions.

If the t represents the class labels and supervision is available, then $I(G(x); t|i)$ is already encouraged by L^{task}. If supervision is not available, we may maximize the entropy of the output of classifier f such that the correlation between user's tasks and representations will not disappear during training. Useful techniques were previously exploited for unsupervised domain adaptation, e.g., [28], and we defer the technique details to Section 5.2.

4.5 Privacy risks from malicious FADE users

Our analysis suggests the feasibility of using adversarial training in the federated setting. The distribution matching is achievable under variety of cases including imbalanced groups, although the success rate may vary. But such power also implies potential privacy overhead associated with FADE. Consider a malicious user i who wants to steal data from others, FADE can match $p_i(x)$ with a victim's data $p_j(x)$. The empirical study in [16] also discussed the risk where a malicious attacker may take advantage of the discriminator to steal other users' data. Our results in Theorem 4.1 theoretically show that the attack is possible in general. During the learning of the adversarial discriminator, injecting predefined noise is known to be effective to defend such attacks [41]. Meanwhile, users could quit or frequently opt-out the federated communication when the privacy budget (quantified by noise and Differential Privacy metric [7]) is low. Based on Theorem 4.2, when more and more users opt-out the communication, the adversary's discriminator can hardly sense one victim's distribution.

5 EXPERIMENTS ON UNSUPERVISED DOMAIN ADAPTATION

In this section, we evaluate the FADE algorithms on Unsupervised Domain Adaptation (UDA) [10, 24, 38]. UDA aims to mitigate the domain shift between supervised and unsupervised data such that the trained classifiers can generalize to unlabeled data. We call the supervised user (domain) as the source user (domain). Each domain may include multiple users.

Related work. [38] is among the first to discuss the adversarial UDA under federated constraint, through sharing the embedding of samples. However, we consider a more challenging problem, a federated adversarial learning *without* sharing data. Recently, learning without access to the source data has gained increasing attention. [24] (SHOT) considered the domain adaptation only using the source-domain model which surprisingly outperformed most traditional UDA with source supervisions. However, its success relies on the pre-matched representation distribution (but not well discriminated) by batch normalization (BN) layers. In the FADE setting, the BN layers will fail to match representations since the local estimate of their mean will be easily biased. In addition, in [10], distillation is used to avoid directly passing data. Differing from [10], FADE is more efficient since it does not need to upload all models from source domain to target domain. For example, if M_s users (M_t) in source (target) domain take part in training, sending models will involves M_sM_t communication. Instead, FADE only use $M_s + M_t$ times to communicate between domains.

Network architectures. We adopt the same network architecture as the state-of-the-art of UDA [23]. As presented in Fig. 4, we first use a backbone network to extract sample features. Specifically, we use modified LeNet [28] for digit recognition, ResNet50 [15] for Office and Office-Home datasets, and ResNet101 for the VisDA-C dataset. We use an one-layer bottleneck to reduce the feature dimension. After the bottleneck, we get a representation of 256-dimension. A single fully-connected layer is used for classification at the end. The discriminators are small-scale networks to match the capability of the classifiers. The networks and algorithms are implemented using PyTorch 1.7. The ResNet backbones pre-trained on ImageNet are retrieved from the torchvision 0.8 package.

5.1 Digit recognition with imbalanced groups

As discussed in Section 4.1, group imbalance could result in the mismatch of group distributions. Here, we empirically evaluate the effect of the imbalanced groups on convergence, adversarial losses and utility performance.

Digit dataset is a standard UDA benchmark built on digit images collected from different environments. 10 digits, from 0 to 9, are included. We follow the UDA protocol of [17] and use two subsets: MNIST and USPS. MNIST dataset contains 60,000 training images and 10,000 testing 28×28 gray-scale images. USPS consists of 7291 training and 2007 testing 16×16 gray-scale images. We augment the USPS training set by resizing and random rotation.

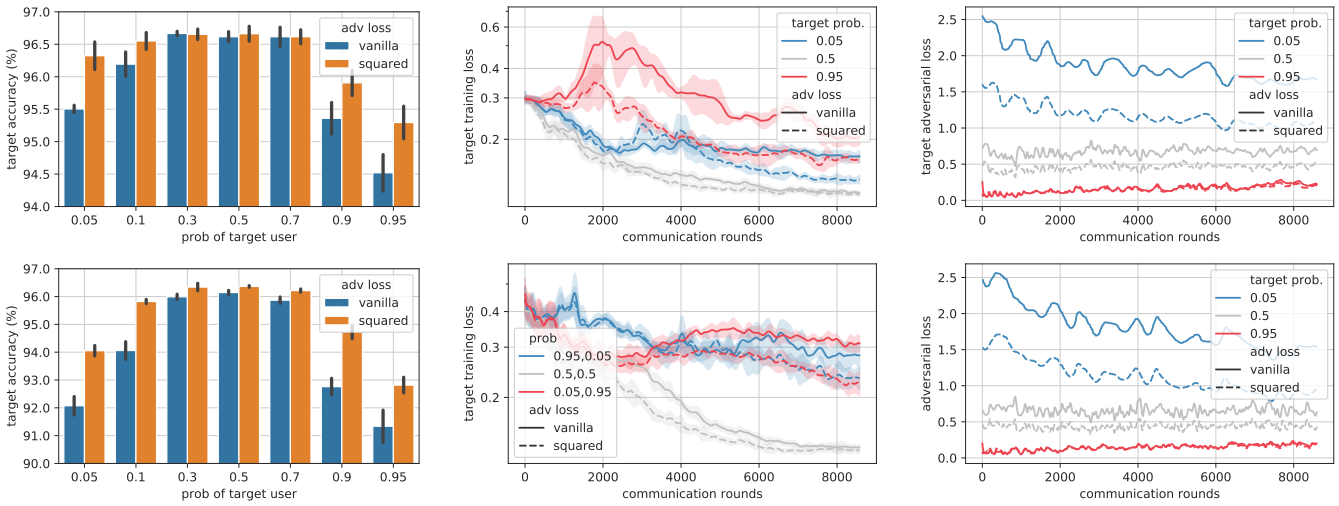


Figure 2: Comparison of vanilla adversarial loss versus the squared adversarial loss on MNIST-to-USPS (top) and USPS-to-MNIST (bottom) UDA. We vary the probability of target users. For both UDA experiments, the SOTA central methods [23] can achieve over 98% accuracies. From left to right, the columns are target domain accuracies, classification losses and adversarial losses of target domain users.

Table 1: Averaged classification UDA accuracies (%) on Office and OfficeHome dataset with 3 non-iid target users and 1 source user. Underlines indicate the occurrence of non-converged results. Standard deviations are included in brackets.

Method	A→D	A→W	D→A	D→W	W→A	W→D	Re→Ar	Re→Cl	Re→Pr	Avg.
Federated methods										
Source only	79.5	73.4	59.6	91.6	58.2	95.8	67.0	46.5	78.2	72.2
non-iid target users w/ 20 (Office) or 45 (OfficeHome) classes per user										
FADE-DANN	85.4 (1.9)	81.8 (1.8)	43.1 (33)	97.7 (0.5)	64.8 (0.5)	99.7 (0.2)	46.4 (37)	34.9 (27)	78.8 (0.1)	70.3
FADE-CDAN	**92.3 (1.2)**	**91.6 (0.5)**	**65.9 (9.3)**	**98.9 (0.2)**	**70.2 (0.8)**	**99.9 (0.1)**	70.3 (1.6)	54.9 (4.6)	**82.2 (0.1)**	80.7
FedAvg-SHOT	83.6 (0.5)	83.1 (0.5)	64.7 (1.4)	91.7 (0.2)	64.7 (2.2)	97.4 (0.5)	**70.7 (0.5)**	**55.4 (0.5)**	80.1 (0.3)	76.8
iid target users										
FADE-DANN	84.2 (1.5)	81.3 (0.4)	66.3 (0.3)	97.5 (1.2)	59.4 (10.6)	99.9 (0.2)	67.3 (0.9)	51.3 (0.4)	79.0 (0.6)	76.2
FADE-CDAN	**93.6 (0.8)**	**92.2 (1.3)**	**71.2 (1.0)**	**98.7 (0.4)**	**71.3 (0.7)**	**100 (0.0)**	70.6 (1.3)	55.1 (1.0)	**82.3 (0.2)**	81.7
FedAvg-SHOT	**96.3 (0.5)**	**94.3 (1.1)**	70.9 (2.0)	98.4 (0.4)	**72.7 (0.9)**	99.8 (0.0)	**74.8 (0.3)**	**60.0 (0.1)**	**84.9 (0.2)**	83.6
Central methods										
ResNet [15]	68.9	68.4	62.5	96.7	60.7	99.3	53.9	41.2	59.9	67.9
Source only [23]	80.8	76.9	60.3	95.3	63.6	98.7	65.3	45.4	78.0	73.8
DANN [11]	79.7	82.0	68.2	96.9	67.4	99.1	63.2	51.8	76.8	76.1
CDAN [28]	92.9	**94.1**	71.0	**98.6**	69.3	**100**	70.9	56.7	81.6	81.7
SHOT [23]	**94.0**	90.1	**74.7**	98.4	**74.3**	99.9	**73.3**	**58.8**	**84.3**	**83.1**

Setup. We assume 2 users from source and target domain, respectively. In each round, we select one user with predefined probability. For example, the case that source and target users are of 0.05 and 0.95 probability implies severe imbalance. If a user/group has high probability, that means the user/group will actively participate in the adversarial learning and the other will activate less. The experiment can also be generalized to multiple users in same frequency while one domain has more users. Both situations imply the imbalance between two groups. In experiments, we fix the batch size to 32 and run one user per communication round. In total, we train the users for global 8600 rounds. In each global round, the users will train locally for 10 iterations. Experiments are repeated 3 times with three fixed seeds. At the beginning, we train the models with adversarial coefficient $\lambda = 0$ when all source users are involved until the classification loss converges. Then, we follow [11, 23] to use the decaying schedule of learning rates and schedule the adversarial coefficient λ from 0 to 1.

Results are reported in Fig. 2. Left two figures show the negative impact of imbalanced groups. When the imbalance is severe (large or small target probability), the drop in target accuracies is more obvious. In the middle pane, the convergence curves of imbalanced groups fluctuate more significantly and fail to converge. In the last pane, the imbalanced cases have large adversarial losses which barely decrease by federated iterations. It explains why the

corresponding classification tasks fail to converge. When using the squared adversarial losses, the ignored adversarial losses of low-frequent users are reduced during federated learning. Meanwhile, the convergence of utility losses are faster. Thus, the negative impact of imbalanced groups is mitigated.

5.2 Object recognition with non-iid users

In Section 4.3, we prove that the non-iid distribution of users will lead to a trivial solution which may lose the natural discrepancy between users. For federated classification learning where each user only has a partial set of classes, the loss of user discrepancy will make the representations non-discriminative to classes. Here, we conduct experiments to reveal the impact of the non-iid users.

Dataset. We adopt three object recognition datasets, Office [40] (small size), Office-Home [44] (medium size) and VisDA-C [39] (large size), including image of office products. The former two are standard benchmarks widely used for UDA. The Office dataset contains three domains: Amazon (A), DSLR (D) and Webcam (W) with 2817, 498, 795 images, respectively. 31 object classes of images are taken under different office environments (corresponding to domains). The Office-Home datasets have 65 categories and 4 domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr), and Real-World images (Re) with 2427, 4365, 4439 and 4357 images, respectively. The VisDA dataset is a challenging large-scale benchmark. The source domain comprises 12-way synthetic classification data. In total, 1.5×10^5 images are synthesized by rendering 3D models and are adapted to $55,000$ unlabeled real-world images.

Setup. In total, 4 users are generated from two domain datasets. First, we let the single source domain user with all classes. Second, we generate 3 non-iid target domain users with partial set of classes following the standard federated setting [32]. For Office dataset, each user has 20 classes and adjacent users have consecutive classes with 10-class stride. For instance, user 1 has class 0 to 20 and user 2 has class 10 to 30. For OfficeHome dataset, each user has 45 classes with 20-class stride. For VisDA-C dataset, each user has 5 classes with 4-class stride. All users in the same domain will have the same number of samples. We select 2 users per communication round when training on OfficeHome. For VisDA-C dataset, we adopt 1 user per round. In this case, the major difficulty comes from non-iid distributions of users conditioned on the subset of classes. In experiments, the parameters for SHOT follows [23]. Details of network architectures and learning rate schedules are discussed in Appendix B.

Baselines. We compare different UDA methods extended by FADE upon the presence of non-iid users. **DANN** [11] is the first work on adversarial domain adaptation based on which many recent methods are developed. **CDAN** [28] is the first to condition the discriminator prediction on the estimated classes, which aligns with our purpose to maximize the mutual information between user (related to classes) and representation. **SHOT** [23] (extended by FedAvg [32]) is the current state-of-the-art method in domain adaptation which does not use source data, assuming approximately mitigated domain shift.

Results. We summarize the results in Tables 1 and 2. Note that the straightforward extension of DANN without constraints will suffer from the user heterogeneity. Therefore, we observe catastrophic failures by DANN, for example, D→A with only a low accuracy. This kind of failures happens when both D (498) is of less samples than A (2817). The possible reason is that the discriminators fail to sense the position of target domain batches which is a small ratio of all target-domain samples and changes frequently by iterations. In comparison, when regulated by estimated classes, SHOT and methods combined with SHOT perform better. Notably, because SHOT relies on BN states to mitigate domain shift, its accuracies are much worse than its central version. Since SHOT can provide pseudo supervisions which conditions on the estimated users' local classes, DANN+SHOT outperforms DANN. In reverse, DANN helps SHOT to mitigate the domain shift. We further explore CDAN+SHOT, which conditions group discrimination on local classifier outputs (correlated to users' classes). As a result, CDAN+SHOT outperforms other methods and is close to the central version of CDAN. Plus, CDAN+SHOT achieves the best average accuracies when the number of users per round varies from 1 to 4. Remarkably, in the hardest case where only one user is trained per round, CDAN+SHOT gains the best accuracies on 8 out of 9 tasks. In a more challenging large-scale VisDA-C dataset, CDAN+SHOT also shows its advantage against other baselines (see Table 2). We note that adversarial methods are more robust to the non-iid users.

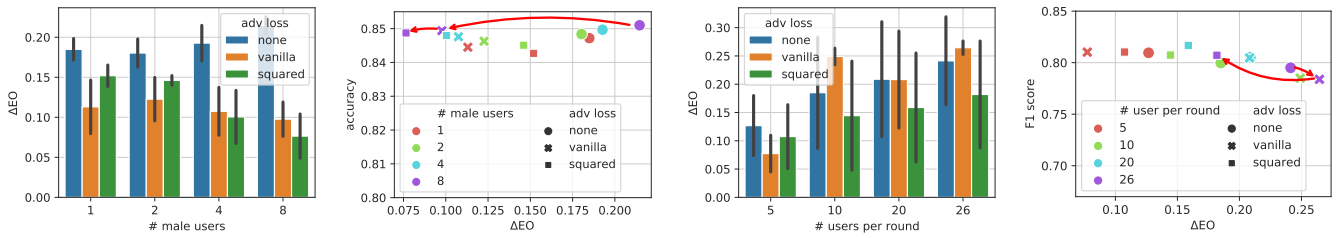
Table 2: Comparison of target accuracies on Visda-C dataset.

Methods	Source only	DANN	SHOT	CDAN
Central	46.6	57.6	82.9	73.9
FADE	54.3	56.4	69.2	73.1 (+SHOT)

6 EXPERIMENTS ON FAIR FEDERATED LEARNING

The fair federated learning is motivated by the imbalanced groups in training. For example, when vendor rallies people to use their software and train model with locally collected data, the global model may be biased by the majority, e.g., male users. When a user from another gender uses the software, she/he may find that the model performs poorly. As a result, the minority group vanishes while majority continues to dominate. Thus, a method actively debiasing w.r.t. the groups will be essential to defend the tendency.

Related work. The fairness in federated learning was first discussed in [22] where users are thought to have different capability for computation. Fairness was enforced by increasing the weights of large loss, which was less optimized. In this experiment, we consider the unfairness brought by the difference of group distributions. With FADE, we use a discriminator locally to justify whether the user's representations are biased from the other group. Related central algorithms have been exploited [8, 29, 45]. To the best of our knowledge, we are the first to encourage such group-based fairness in federated setting. Importantly, our method preserve the privacy of group variables. The concerns of the privacy of group variables was previously discussed [13]. In [13], Hashimoto *et al.* assumes the group membership and number of groups are unknown to the central learning server, when users interact with the system and contribute data. Our FADE extends the setting to a distributed



(a) Results on Adult by varying number of male users.

(b) Results on MCI data by varying number of users per round.

Figure 3: FADE with/without adversarial losses. In each subfigure, *left* is fairness measured by ΔEO where smaller values indicates better fairness; *right* is the trade off between fairness and utility where left-top is the preferred balance.

framework where the private group information is still unknown to other parties including the aggregation server.

We utilize the Equalized Odds (ΔEO) to evaluate the degree of fairness. Consider a binary classifier $f : \mathcal{Z} \to \{0, 1\}$ predicting label variable y when representations ($z \in \mathcal{Z}$) are sampled from two groups. We denote the conditional p.d.f. $p(z|g, y)$ as $z_{g,y}$ which shapes the probability of z at group g and class y. An algorithm is said to be fair if the positive ΔEO (defined below) is close to 0.

$$\Delta\text{EO} \triangleq \left|\mathbb{E}_{z_{0,0}}[f(z)] - \mathbb{E}_{z_{1,0}}[f(z)]\right| + \left|\mathbb{E}_{z_{0,1}}[1 - f(z)] - \mathbb{E}_{z_{1,1}}[1 - f(z)]\right| \quad (9)$$

which comprises the absolute difference in false positive rates and the absolute difference in false negative rates.

6.1 Fair adult income prediction

Dataset. We evaluate our algorithm on the UCI Adult dataset[1] which is a standard benchmark for fair classification. The dataset consists of over 40,000 vector samples from the 1994 US Census. Each sample includes 14 attributes predicting if his/her income is over 50,000 dollars.

Setup. We adversarially disentangle the unfair representations from the gender. When keeping the total data size fixed, we construct one female user and vary the number of male users. Each synthesized user evenly split the samples in the group. We run FADE for 8,000 communication rounds. In every round, 2 users are selected to train for 1 local iteration on a batch of 64 samples. The accuracies and fairness are evaluated on the left-out 10% samples. The network structure is in Fig. 5. We set hyper-parameters as $\beta = 0.5$ and the initial learning rate as 10^{-3}.

Results are depicted in Fig. 3a. Without adversarial training, the unfairness is aggravated when the imbalance between groups worsens. When more male users are involved, the squared adversarial loss is able to further improve the fairness. Instead, the vanilla adversarial learning performs better when the two groups are balanced. Both adversarial losses will maintain the utility performance close to the non-adversarial method.

6.2 Fair MCI detection

Dataset. Mild Cognition Impairment (MCI) is the pre-symptom of Alzheimer's Disease (AD) which typically happens on elders. Early detection of MCI is important for prevention of AD occurrence and treatment [1, 43]. Details of the dataset is comprised in Appendix B.3 where females forms the majority group (over 94%). The prediction task here is to classify the disease condition, Normal Cognition (NC) or MCI, based on the daily activities (walking speed, etc.).

Setup. In the original dataset, there are 88 users with different number of samples. We notice the imbalance between NC and MCI users will greatly degrade the model quality. To focus on our fairness task, we manually select 26 users such that 13 users was diagnosed as NC at least once and the other 13 ones are stable MCI patients. Because male users are much fewer than female ones, we prefer to select male users when balancing the two classes. After downsampling, users have 39 samples on average. Among the 26 users, there are 6 males and 20 females in total. Details of features, preprocessing and network architectures are deferred to Appendix B.3. We set hyper-parameters as $\beta = 0.5$, the initial learning rate as 10^{-2} and batch size as 16. In the 700 communication rounds, we first train without adversarial losses for 400 rounds and then schedule the λ and learning rates as the Adult experiments.

Results. We compare the convergence of the training F_1-score (utility) and ΔEO (fairness) by varying the number of users per round. As shown in Fig. 3b, the unfairness is obvious with ΔEO over 0.2 when no adversarial losses are used. We see that the vanilla adversarial loss failed to debias in most cases. In contrast, the squared adversarial loss stably debias the unfair performance in all cases. When the number of users per round is less than 10, even the non-adversarial loss is more fair. The natural debiasing could be attributed to the random selection of users, which breaks the domination of one group in a short span.

7 CONCLUSION

In this work, we propose a unified framework for federated adversarial learning called FADE. Our framework preserves the user privacy and allows user to freely opt-in/out the learning of the adversarial component. To our best knowledge, we are the first to study the properties of adversarial learning in the federated setting. We presented the potential challenge and solution for the FADE, and identified a gap between FADE and its centralized counterpart as an open question for our future work.

Acknowledgement

This material is based in part upon work supported by the National Science Foundation under Grant IIS-1749940, EPCN-2053272, Office of Naval Research N00014-20-1-2382, and National Institute on Aging (NIA) R01AG051628, R01AG056102, P30AG066518, P30AG024978, RF1AG072449.

[1] https://archive.ics.uci.edu/ml/datasets/adult

REFERENCES

[1] P. S. Aisen, S. Andrieu, C. Sampaio, M. Carrillo, Z. S. Khachaturian, B. Dubois, H. H. Feldman, R. C. Petersen, E. Siemers, R. S. Doody, S. B. Hendrix, M. Grundman, L. S. Schneider, R. J. Schindler, E. Salmon, W. Z. Potter, R. G. Thomas, D. Salmon, M. Donohue, M. M. Bednar, J. Touchon, and B. Vellas. 2011. Report of the Task Force on Designing Clinical Trials in Early (Predementia) AD. *Neurology* 76, 3 (Jan. 2011), 280–286.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. (Jan. 2017).

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A Theory of Learning from Different Domains. *Machine Language* 79, 1-2 (May 2010), 151–175.

[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1175–1191.

[5] Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems*.

[6] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS '12)*. Association for Computing Machinery, Cambridge, Massachusetts, 214–226.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography (Lecture Notes in Computer Science)*. Springer Berlin Heidelberg, 265–284.

[8] Harrison Edwards and Amos Storkey. 2016. Censoring Representations with an Adversary. *ICLR* (March 2016).

[9] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized Federated Learning: A Meta-Learning Approach. In *Advances in Neural Information Processing Systems*.

[10] Hao-Zhe Feng, Zhaoyang You, Minghao Chen, Tianye Zhang, Minfeng Zhu, Fei Wu, Chao Wu, and Wei Chen. 2021. KD3A: Unsupervised Multi-Source Decentralized Domain Adaptation via Knowledge Distillation. *AAAI* (2021).

[11] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *International Conference on Machine Learning*. PMLR, 1180–1189.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]* (June 2014).

[13] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*. PMLR, 1929–1938.

[14] Chaoyang He, Murali Annavaram, and Salman Avestimehr. 2020. Group Knowledge Transfer: Federated Learning of Large CNNs at the Edge. *arXiv:2007.14513 [cs]* (Nov. 2020).

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2016).

[16] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. ACM, New York, NY, USA, 603–618.

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *International Conference on Machine Learning*. PMLR, 1989–1998.

[18] Junyuan Hong, Jeffrey Kaye, Hiroko H. Dodge, and Jiayu Zhou. 2020. Detecting MCI Using Real-Time, Ecologically Valid Data Capture Methodology: How to Improve Scientific Rigor in Digital Biomarker Analyses. *Alzheimer's & Dementia* 16, S5 (2020), e044371.

[19] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. 2008. *Dataset Shift in Machine Learning | The MIT Press*. MIT Press.

[20] Jeffrey A. Kaye, Shoshana A. Maxwell, Nora Mattek, Tamara L. Hayes, Hiroko Dodge, Misha Pavel, Holly B. Jimison, Katherine Wild, Linda Boise, and Tracy A. Zitzelberger. 2011. Intelligent Systems for Assessing Aging Changes: Home-Based, Unobtrusive, and Continuous Assessment of Aging. *The Journals of Gerontology: Series B* 66B, suppl_1 (July 2011), i180–i190.

[21] Daliang Li and Junpu Wang. 2019. FedMD: Heterogenous Federated Learning via Model Distillation. *arXiv:1910.03581 [cs, stat]* (Oct. 2019).

[22] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2019. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*.

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. *International Conference on Machine Learning* (Oct. 2020).

[24] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. 2020. Source Data-Absent Unsupervised Domain Adaptation through Hypothesis Transfer and Labeling Transfer. *ArXiv* (2020).

[25] Ming Lin, Pinghua Gong, Tao Yang, Jieping Ye, Roger L. Albin, and Hiroko H. Dodge. 2018. Big Data Analytical Approaches to the NACC Dataset: Aiding Preclinical Trial Enrichment. *Alzheimer Disease & Associated Disorders* 32, 1 (2018), 18–27.

[26] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. In *Advances in Neural Information Processing Systems*.

[27] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-Task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1–10.

[28] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. 2018. Conditional Adversarial Domain Adaptation. *arXiv:1705.10667 [cs]* (Dec. 2018).

[29] David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. 2018. Learning Adversarially Fair and Transferable Representations. *International Conference on Machine Learning* (2018), 11.

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2019. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv:1706.06083 [cs, stat]* (Sept. 2019).

[31] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. 2019. Attentive Single-Tasking of Multiple Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1851–1860.

[32] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*. 1273–1282.

[33] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*.

[34] P. Mohassel and Y. Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *2017 IEEE Symposium on Security and Privacy (SP)*. 19–38.

[35] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. 2017. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4467–4477.

[36] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the Preferred Inputs for Neurons in Neural Networks via Deep Generator Networks. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 3387–3395.

[37] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (Oct. 2010), 1345–1359.

[38] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. 2019. Federated Adversarial Domain Adaptation. In *International Conference on Learning Representations*.

[39] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. 2017. VisDA: The Visual Domain Adaptation Challenge. *arXiv:1710.06924 [cs]* (Nov. 2017).

[40] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting Visual Category Models to New Domains. In *Proceedings of the 11th European Conference on Computer Vision: Part IV (ECCV'10)*. Springer-Verlag, Berlin, Heidelberg, 213–226.

[41] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. 2019. DP-CGAN: Differentially Private Synthetic Data and Label Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 0–0.

[42] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial Discriminative Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.

[43] B. Vellas, R. Bateman, K. Blennow, G. Frisoni, K. Johnson, R. Katz, J. Langbaum, D. Marson, R. Sperling, A. Wessels, S. Salloway, R. Doody, and P. Aisen. 2015. Endpoints for Pre-Dementia AD Trials: A Report from the EU/US/CTAD Task Force. *The journal of prevention of Alzheimer's disease* 2, 2 (June 2015), 128–135.

[44] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5018–5027.

[45] Christina Wadsworth, Francesca Vera, and Chris Piech. 2018. Achieving Fairness through Adversarial Learning: An Application to Recidivism Prediction. *arXiv:1807.00199 [cs, stat]* (June 2018).

[46] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating Unwanted Biases with Adversarial Learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society (AIES '18)*. Association for Computing Machinery, New York, NY, USA, 335–340.

A PROOFS

Proof of Theorem 4.1. Substitute $D^*_{\alpha_1,\alpha_2}(z)$ into Eq. (4):

$$\begin{aligned}
\tilde{\mathbf{D}}_{p_1,p_2} &= \mathbb{E}_{p_1}[\alpha_1 \log \frac{\alpha_1 p_1(z)}{\alpha_1 p_1(z) + \alpha_2 p_2(z)}] \\
&\quad + \mathbb{E}_{p_2}[\alpha_2 \log \frac{\alpha_2 p_2(z)}{\alpha_1 p_1(z) + \alpha_2 p_2(z)}] \\
&= \alpha_1 \mathrm{KL}\left[p_1 \middle| \frac{\alpha_1 p_1 + \alpha_2 p_2}{\alpha_1 + \alpha_2}\right] \\
&\quad + \alpha_2 \mathrm{KL}\left[p_2 \middle| \frac{\alpha_1 p_1 + \alpha_2 p_2}{\alpha_1 + \alpha_2}\right] \\
&\quad + \alpha_1 \log \alpha_1 + \alpha_2 \log \alpha_2 \\
&\quad + (\alpha_1 + \alpha_2) \log(\alpha_1 + \alpha_2) \\
&\geq \alpha_1 \log \alpha_1 + \alpha_2 \log \alpha_2 \\
&\quad + (\alpha_1 + \alpha_2) \log(\alpha_1 + \alpha_2)
\end{aligned}$$

where the last inequality is from the non-negative property of KL divergence.

Note when $p_1 = p_2$, both KL divergence is 0. Thus, we can conclude that $p_1 = p_2$ is the sufficient condition. □

Proof of Theorem 4.2. For the ease of derivation, we assume α_1 and α_2 are normalized s.t. $\alpha_1 + \alpha_2 = 1$. From $|\log p_1(x) - \log p_2(x)| \leq \epsilon$, we can get

$$\begin{aligned}
e^{-\epsilon} &\leq p_1(x)/p_2(x) \leq e^{\epsilon}, \\
e^{-\epsilon} &\leq p_2(x)/p_1(x) \leq e^{\epsilon}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathrm{KL}\left[p_1 \,|\alpha_1 p_1 + \alpha_2 p_2\right] &= \int_x p_1 \log\left(\frac{p_1}{\alpha_1 p_1 + \alpha_2 p_2}\right) \\
&\leq \int_x p_1 \log\left(\frac{1}{\alpha_1 + \alpha_2 e^{-\epsilon}}\right) \\
&= \epsilon - \log(\alpha_1 e^{\epsilon} + \alpha_2).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathrm{KL}\left[p_2 \,|\alpha_1 p_1 + \alpha_2 p_2\right] &= \int_x p_2 \log\left(\frac{p_2}{\alpha_1 p_1 + \alpha_2 p_2}\right) \\
&\leq \int_x p_2 \log\left(\frac{1}{\alpha_1 e^{-\epsilon} + \alpha_2}\right) \\
&= \epsilon - \log(\alpha_1 + \alpha_2 e^{\epsilon}).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\tilde{\mathbf{D}}_{p_1,p_2} &= \alpha_1[\epsilon - \log(\alpha_1 e^{\epsilon} + \alpha_2)] \\
&\quad + \alpha_2[\epsilon - \log(\alpha_1 + \alpha_2 e^{\epsilon})] \\
&\quad + \alpha_1 \log \alpha_1 + \alpha_2 \log \alpha_2 \\
&= \epsilon - \alpha_1 \log(e^{\epsilon} + \alpha_2/\alpha_1) \\
&\quad - \alpha_2 \log(e^{\epsilon} + \alpha_1/\alpha_2) \\
&\leq O((1 - \alpha_2)\epsilon) \\
&= O(\alpha_1 \epsilon/(\alpha_1 + \alpha_2))
\end{aligned}$$

where we manually add $(\alpha_1 + \alpha_2)$ to normalize α_1. □

B EXPERIMENT DETAILS

B.1 Dynamic schedules

We use dynamic schedules for learning rates and the adversarial parameter λ following previous work [11]. Specifically,

$$\begin{aligned}
\eta_t &= \frac{1}{(1 + 10\frac{t}{T_{\max} K})^{0.75}} \\
\lambda_t &= \frac{2}{1 + \exp(-10t/T_{\max})} - 1
\end{aligned}$$

where K is the number of local iterations and $T_{\max}$ is the number of global rounds. Notably, η_t is schedule locally and λ_t is scheduled globally.

B.2 Network architectures

Federated UDA. The network architectures are presented in Fig. 4.

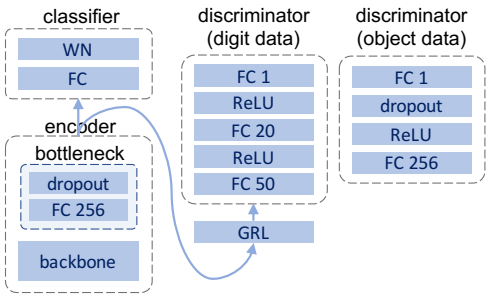


Figure 4: Network architectures for digit and object datasets. WN denotes the weight-norm layer [23] and FC 256 denotes fully-connected layer with 256 units. GRL is the gradient reversal layer [11].

Batch normalization in FADE. During training, we share the parameters of ResNet between users. Notably, in ResNet, batch normalization (BN) layer is densely embedded in different depth. The BN layer is known to be important for transferring between distinct domains, because the hidden representations will be normalized with mean and variance estimated from a batch. Because such estimation could be easily biased by a small batch, running estimation by accumulating results from previous batches is a common practice. Thus, it is also important for all users to get the global estimate of the mean and variance by communication. However, sharing such a running estimate of representation mean and standard variance may leak the private information [35, 36]. For example, given a feature vector at a specific layer, the input image can be reverted using a conditional generative network [35, 36]. Instead of sharing the mean and variance (*BN states*), we keep the values the same as values pre-trained on ImageNet. In addition, we freeze the BN states both during pre-training on the source domain user. In **??**, we compare the transferring of source model with or without frozen BN states during pre-training. It turns out that freezing the BN states will improve the zero-shot transferring in terms of target accuracies.

Fair federated learning. We depict the network architectures for Adult and MCI datasets in Fig. 5. For the Adult dataset, we aim to evaluate the performance of deep networks. Thus, we use a deeper network other than a shallow one for central algorithms [29]. Because of the small size of the MCI dataset, we adopt a

small network architecture where only two layers of LSTM are used for feature extraction and one layer for classification or group identifying.

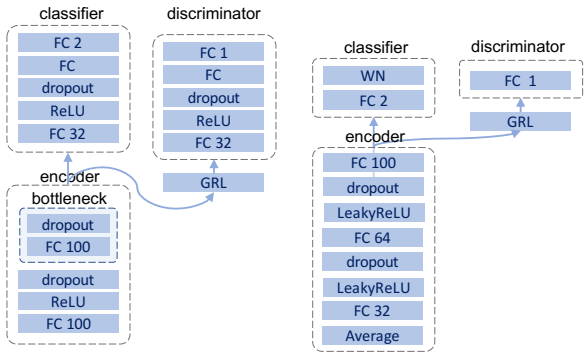


Figure 5: Network architectures for Adult and MCI datasets. LSTM 100 indicates a Long Short-Term Memory (LSTM) cell with 100 hidden units.

B.3 Details of MCI datasets

Dataset Due to the mild symptoms and expansive cost of clinic diagnosis, early detection of MCI is a hard task. To address the challenge, MCI detection models is built on a MCI dataset, which is collected with Intelligent Systems for Assessing Aging Change (ISAAC), a longitudinal cohort study [18, 20]. A total of 152 participants were enrolled beginning in 2017. 12 variables are extracted from the participants' sensor data and clinical diagnoses was done once a year. Meanwhile, four kinds of demographic information are also recorded, including age, gender, education, and ethnicity, which are potentially unfair features for each patient.

Though prior work has shown the effectiveness of machine learning methods in diagnosis prediction [18, 25], the possibility of training such a model fairly in a distributed framework remains unknown. We assume the sensor data can be immediately trained locally and only the trained models are sent to the server. The distributed framework brings in several new challenges. First, users' data are kept locally and many users only have one-class data which makes the local model less discriminative. For example, 13 users are always diagnosed as MCI during his/her recording. Second, it is difficult to do adversarial learning like Fig. 1b. Because the users' group information, e.g., gender, can not be revealed to others, the server has no idea who will be the adversarial group. Therefore, we utilize the FADE framework to tackle these issues as illustrated in Fig. 1c. As far as privacy is concerned, in the ISAAC protocol, the sensor data were collected periodically by engineers such that the user data are kept away from others. But we argue that our extension to federated setting is practical because the data are not directly shared.

Preprocessing. Since the records of some patients are missing due to occasionally off-line of sensor systems, and these incomplete samples can introduce uncertainty in our experiments, we choose to remove some samples according to a certain missing value. To generate samples, hundreds of days of records for each patient will then be sliced by a moving window, and each slice is used as a sample for training or to be predicted. The slicing is done inside each person's sequence without overlap. The time window is moved in a step of 7 days. Only a subsequence of a small enough ratio of missing values will be maintained for the current study. The number of sequences for each patient is related to the amount of data the patient has. For some of the patients, they have only a small number of records. We also remove the samples of those patients to avoid inaccurate prediction.

We have 12 varaibles in total, including gender (Rsex), years of education (Ryrschool), race/ethnicity (Rethnic), age at each date (ageyrs), total computer use (compuse), computer sessions (numcsess), track sensor line (linenum), walks (numwalks), mean walking speed (meanws), upper quartile of walking speed (wsq3), coefficient of var of walking speed (wscv) and std deviation of walking speed (wsstddev). We preprocess special variables in the following specified methods. For linenum which is a sensor metric identity value, its integer values are transformed into a one-hot encoding form that uses the position of a single one to indicate the ID value. RSex and Rethic variables are encoded in the same way. The ages are transformed by 3-bin discretization. All continuous variables are normalized within $[-1, 1]$ by min-max scaling such that no significant variance will occur between different variables and their coefficients could be trained in a numerically robust way.

All the data features are collected in a relatively redundant way, for which they should be carefully selected for better prediction performance. We select features using mutual information, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and the higher value means higher dependency. A special case is the linenum variable which only makes sense when other walking speed features are used. As a result, when a walking feature is selected according to the above metrics, the linenum variable is automatically included.