FedGCR: Achieving Performance and Fairness for Federated Learning with Distinct Client Types via Group Customization and Reweighting

Shu-Ling Cheng[1] **Chin-Yuan Yeh**[1,3], **Ting-An Chen**[2,3], **Eliana Pastor**[4] **Ming-Syan Chen**[1,2],
[1]Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
[2]Department of Electrical Engineering, National Taiwan University, Taiwan
[3]Institute of Information Science, Academia Sinica, Taiwan
[4]Department of Control and Computer Engineering, Politecnico di Torino, Italy
{csling, cyyeh, tachen}@arbor.ee.ntu.edu.tw, eliana.pastor@polito.it, mschen@ntu.edu.tw

Abstract

To achieve better performance and greater fairness in Federated Learning (FL), much of the existing research has centered on individual clients, using domain adaptation techniques and redesigned aggregation schemes to counteract client data heterogeneity. However, an overlooked scenario exists where clients belong to distinctive groups, or, *client types*, in which groups of clients share similar characteristics such as device specifications or data patterns. Despite being common in group collaborations, this scenario has been overlooked in previous research, potentially leading to performance degradation and systemic biases against certain client types. To bridge this gap, we introduce *Federated learning with Group Customization and Reweighting (FedGCR)*. FedGCR enhances both performance and fairness for FL with Distinct Client Types, consisting of a *Federated Group Customization (FedGC)* model to provide customization via a novel prompt tuning technique to mitigate the data disparity across different client-types, and a *Federated Group Reweighting (FedGR)* aggregation scheme to ensure uniform and unbiased performances between clients and between client types by a novel reweighting approach. Extensive experiment comparisons with prior FL methods in domain adaptation and fairness demonstrate the superiority of FedGCR in all metrics, including the overall accuracy and performance uniformity in both the group and the individual level. *FedGCR* achieves **82.74%** accuracy and **12.26**(↓) in performance uniformity on the Digit-Five dataset and **81.88%** and **14.88%**(↓) on DomainNet with a domain imbalance factor of 10, which significantly outperforms the state-of-the-art. Code is available at https://github.com/celinezheng/fedgcr.

Introduction

Federated Learning (FL) enables multiple clients to collaborate in training deep learning models while maintaining the privacy of their data by only sharing the model updates after each round of local training (McMahan et al. 2017). While individual clients gain a more sophisticated model effectively trained on the collective data by aggregating each client's model update in a central server, the heterogeneity of the data between each client may induce biased results as well as performance degradation. Efforts have been made to address these issues, such as integrating domain adaptation techniques into the model for enhanced performance (Jiang, Wang, and Dou 2022; Qu et al. 2022; Yoon et al. 2021) and redesigning the aggregation scheme for performance uniformity to promote individual fairness (Li et al. 2020; Xu et al. 2023; Li et al. 2021a).

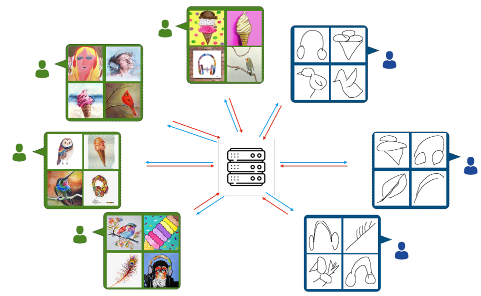

Figure 1: Illustration of the Federated Learning with Distinct Client Types scenario. Clients from a platform hosting paintings (in green) collaborate with another platform, where individual clients possess personal hand-drawn images (in blue), to jointly train an image classifier. The scenario reveals a unique setting where collective differences may pose challenges to traditional FL approaches.

However, existing research has mostly focused on the *individual client* level, neglecting the potential existence of diverse *types of clients* in FL. In Fig. 1, the scenario of Federated Learning with Distinct Client Types (FL-DCT) naturally appears in collaborations between large online platforms, each hosting multiple clients with their respective private data. To illustrate the scenario, consider the setting presented in Figure 1. Clients from a platform hosting data with a certain characteristics (*e.g.*, paintings, illustrated in green in the figure) collaborate with another platform, where individual clients possess images of another domain (*e.g.*, personal hand-drawn images, in blue) to train an image classifier jointly. The collective differences under this setting pose challenges to traditional FL approaches. Recent developments of decentralized social networking protocols such as ActivityPub (Kominers and Wu 2023; Pierce 2023),[1] a decentralized social networking protocol, suggests that future

[1]cf. https://www.w3.org/TR/activitypub/.

FL projects could involve users between large social media platforms. Influenced by factors such as device standards, image processing norms, or platform culture, user data from different platforms may present distinct domain differences, corresponding to disparate and distinct *client-types*, *i.e.*, categories or groupings of clients based on similarities in term of their respective data. Overlooking the client-type differences risks forfeiting opportunities for better performance through type-specific model customization and poses challenges to achieving greater fairness at a collective level, potentially leading to systemic biases against whole groups of clients due to their unique characteristics, which may not be adequately captured or addressed by generic models.

Remarkably, existing research centered on individual clients fails to adequately address the performance and fairness challenges specific to FL with Distinct Client Types. Techniques for individual-based domain adaptation in FL often result in considerable space complexity by pursuing complete personalization for each client (Kulkarni, Kulkarni, and Pant 2020). Such approaches not only fall short of guaranteeing uniform FL performance across all clients (Li et al. 2021b) but also overlook the collective needs of clients belonging to the same type. On the other hand, methods promoting *individual fairness* in FL (Li et al. 2020) are ill-suited for ensuring fairness at the collective level, especially when distinct client types are present. Diverse data distributions and unique characteristics among groups of clients belonging to different types can pose difficulties in achieving uniform performance when addressing them individually. Categorizing clients enables us to handle data heterogeneity and ensure uniform performance across client types.

In this work, we aim to achieve both good performance and better fairness in Federated Learning with Distinct Client Types. Our goal is to enhance performance via type-based customization, while promoting fairness at the *collective* level between types of clients by encouraging a consistent distribution of performance across different client types. To this end, we introduce Federated learning with Group Customization and Reweighting (FedGCR), consisting of the *Federated Group Customization (FedGC)* model and the *Federated Group Reweighting (FedGR)* aggregation scheme. FedGC is designed to tackle data disparity across client types, elevating FL performance at the client-type level. Building on the state-of-the-art visual prompt tuning technique (Jia et al. 2022), FedGC leverages pretrained Vision Transformers (ViT) to offer *customization* for each client according to its specific type using carefully designed *prompts*. In particular, FedGC leverages two types of prompts: global and type-specific prompts. Conceptually, the global prompts facilitate *general* adjustments to the pretrained ViT, whereas the type-specific prompts provide *customization* that boosts performances for each client-type without requiring distinct models.

On the other hand, FedGR addresses collective fairness considerations, ensuring performance uniformity across client types. Specifically, to protect client data privacy and confidentiality, FedGR first leverages the type-specific feature vector—*averaged over the client's dataset*—to cluster the clients into a predetermined number of groups, circumventing the need for a priori client-type information. Then, based on the clustering result, FedGR introduces a *group-wise reweighting* mechanism that expands prior FL individual-level fairness algorithms to also account for collective-level fairness. In particular, following the approach of Li et al. (2020), FedGR recalibrates the aggregation weight of each client's model updates based on the average performances of the cluster they belong to, in addition to their own performances. To account for potential clustering inaccuracies, we include a scheduling hyperparameter that gradually enhances the influence of *group reweighting* relative to *individual reweighting*. While an overly restricted cluster count will cause some cluster to contain multiple client-types, we find that setting the cluster number equal to or greater than the client-type count results in near-perfect alignment, due to the efficacy of FedGC.

We investigate the task of image classification where each client type corresponds to a distinct data domain, further subjecting them to additional imbalance settings. Extensive experiments over three multi-domain datasets reveal that *FedGCR* surpasses all baselines in *every* assessed metric, including overall accuracy as well as performance uniformity between client-types and between individual clients.

Our contributions are summarized as follows:

- We investigate the scenario of **Federated Learning with Distinct Client Types (FL-DCT)**, considering both *performance* and *fairness* at the collective client-type level.
- We present **Federated learning with Group Customization and Reweighting (FedGCR)**, a novel FL algorithm consisting of the FedGC model which provides customization for distinct client-types via a novel visual prompt tuning design, and the FedGR scheme which enables performance uniformity at the collective level through group reweighting approaches, while also maintaining individual performance uniformity.
- Extensive experiments on three real-world datasets establish the superiority of FedGCR over baseline methods on both performance and fairness, especially under imbalanced settings.

Related Work

Federated Learning

Federated Learning (FL) facilitates collaborative training across multiple clients without revealing private data (McMahan et al. 2017). In FL, clients send model updates to a central server, where they are aggregated using factors like data size proportion for weighting each client's input. The server then redistributes the consolidated model for further training. This standard approach assumes that client data are independent and identically distributed (IID), leading to potential degradation in performance when clients have distinct datasets (Li et al. 2022) and naturally feature biased results (Collins et al. 2021).

Domain Adaptation in FL To handle client data heterogeneity, prior works adopt methods ranging from aligning client data through techniques such as appending a small subset of shared data (Zhao et al. 2018), preprocessing (Sheller et al. 2019), feature alignment (Liu et al. 2021; Jiang, Wang, and Dou 2022), style transfer (Chen et al. 2023), and data augmentation (Yoon et al. 2021; Zhou and Konukoglu 2023), to providing personalized FL (Kulkarni, Kulkarni, and Pant 2020), *i.e.*, customizing the model parameters for each individual client (Li et al. 2021b,c; Tan et al. 2022; Marfoq et al. 2022; Zhong et al. 2023; Sattler, Müller, and Samek 2020; Ghosh et al. 2020). However, these methods come with substantial computational overhead and entail significant memory storage requirements. In particular, clustered FL (Sattler, Müller, and Samek 2020; Ghosh et al. 2020) features completely parallel and duplicative FL training for each cluster of clients. An alternative approach aims to reduce the shared model's sensitivity to distinct inputs (Foret et al. 2020; Jiang, Wang, and Dou 2022), but the trade-off between robustness and sensitivity may compromise performance (Zhang et al. 2019). In contrast, FedGCR adopts novel prompt tuning techniques, enabling a single model to perform type-customized operations, thereby avoiding the aforementioned trade-offs.

Fairness in FL To mitigate biased results, (Mohri, Sivek, and Suresh 2019; Hu et al. 2020; Li et al. 2020, 2021b) aim to encourage a more uniform distribution of the model's performance across clients by adjusting the aggregation scheme. For example, AFL utilizes a min-max optimization to boost the worst-performing clients (Mohri, Sivek, and Suresh 2019), while q-FFL reweights the aggregation to favor clients with higher loss (Li et al. 2020). However, the above works focus on individual clients, whereas this work also considers the systemic bias between distinct groups of clients belonging to different client types. It is worth clarifying that prior works exploring *group fairness* generally aim to mitigates bias between data sample groups *within* each client's private dataset (Zhang, Kou, and Wang 2020; Ezzeldin et al. 2023; Du et al. 2021; Papadaki et al. 2022). In contrast, our work focuses on fairness among groups of clients corresponding to different client-types, and aim for uniform performance between the different types. A recent work (Yue, Nouiehed, and Al Kontar 2023) similarly addresses fair performance across both groups of clients and individual clients. However, it relies on a priori knowledge of the client groups, while our approach automatically infers the client type based on anonymized information. Finally, these works do not address scenario where client's private data exhibits distinct domains differences. In contrast, FedGCR utilizes prompt tuning to overcome the domain disparity through customization for different client-types.

Prompt Tuning for Visual Tasks

Inspired by prompt tuning techniques in NLP (Liu et al. 2023), Visual Prompt Tuning (VPT) (Jia et al. 2022) aims to leverage large vision models, *i.e.*, Vision Transformers (ViT) (Dosovitskiy et al. 2021) for downstream tasks. Concretely, while ViT processes a sequential array of *image patches*, VPT inserts learnable prompt tokens into the array to be processed by a frozen pretrained ViT alongside the image patches. In essence, the lightweight prompt tokens guides the ViT to achieve customization for the downstream task. Concurrent work (Zhou et al. 2022b,a) develops prompt tuning for vision-language models (*e.g.*, CLIP (Radford et al. 2021)) for improved results on zero-shot image classification tasks. In particular, the work of Zhou et al. (2022a) features a second order customization, where an image encoder processes an input image into secondary prompts that further finetunes the model behaviour. In this work, we improve upon VPT based on such concept and design FedGCR to tackle the novel scenario of FL-DCT.

Methodology

To address the challenges of Federated Learning with Distinct Client Types (FL-DCT), we introduce Federated learning with Group Customization and Reweighting (FedGCR), consisting of a novel Federated Group Customization (FedGC) model, which provides customization over the distinct domain differences at the local level, and a Federated Group Reweighting (FedGR) aggregation scheme, which provides an improved aggregation process that allows for uniform performances at both the collective client type level and the individual client level at the server level. Fig. 2 provides a high-level illustration of the entire framework.

The FedGC Model

Model Design FedGC provides type-based customization by leveraging *prompt tokens*. In particular, a set of learnable embeddings are utilized as *global prompts*, while a fully connected neural network named GC-Net is utilized to project image representations into *type-specific prompts*. As shown in Fig. 2, FedGC consists of a frozen pretrained ViT, another MLP classification head, GC-Net, and a set of learnable *global prompts* denoted as $\mathbf{p} = \{p_1, p_2, \ldots, p_n\}$. By devising a two-stage process, FedGC efficiently leverages ViT to serve as both the image encoder and the image classifier.[2] In the first stage, FedGC crafts a *type-specific prompt* h for the input image. Specifically, image patches are passed through ViT and then projected via GC-Net to derive the type-specific prompt h. In the second stage, the *type-specific prompt* h is added onto the *global prompts* $\bar{\mathbf{p}} = \{p_1 + h, p_2 + h, \ldots\}$. The combined prompts are concatenated to the image patches and passed through the ViT to derive the image representation z, which is then processed by the MLP head to produce the final classification result $\hat{y}$. Intuitively, the global prompts $\mathbf{p}$, shared by all clients, represent a global adjustment to align the ViT for the image classification task, while the type-specific prompts h, crafted by GC-Net for each input image, offer customization based on the data domain of the corresponding client type.

It is worth noting that FedGC features a novel design that sets it apart from existing methods. Compared to VPT (Jia et al. 2022), we cleverly leverage ViT not only as the classifier but also as an image encoder, and offer additional customization by using the ViT embeddings to craft the *type-specific prompts*. This unique approach enables us to take advantage of the powerful Vision Transformers for FL-DCT,

[2]The ViT depiction in Fig. 2 mirrors Dosovitskiy et al. (2021).

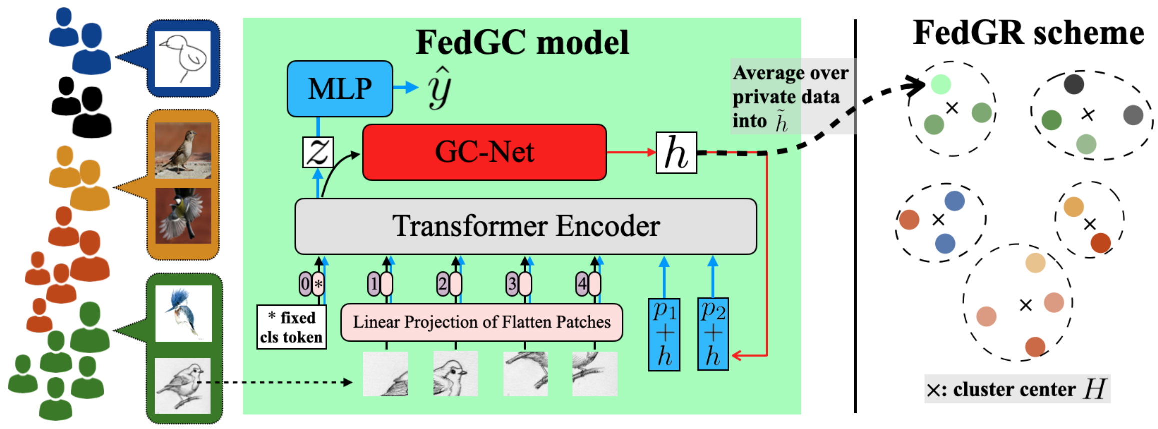

Figure 2: Illustration of FedGCR. The figure presents an example of 5 *client types* (indicated by color) with distinct domains collaborating in FL. FedGC enables customization at the client level through a two-stage process: 1) crafting the *type-specific prompt* h by processing the input image with ViT and GC-Net (black arrows), then combining h with the *global prompts* (p_1 and p_2, red arrow), and 2) process the combined prompts with the image by ViT (blue arrows) into the image representation z, then by MLP for the final classification prediction $\hat{y}$. FedGR performs group reweighting based on cluster results of the *client representations* $\tilde{h}$ (Equation (4)) at the server level, ensuring uniform performance between the *client-types*.

while maintaining a lightweight design by keeping the number of trainable parameters low.

The FedGR Scheme

The conventional FL setting (McMahan et al. 2017) with K clients operates under the following objective

$$\min_{\theta} f(\theta) = \sum_{k=1}^{K} \omega_k L_k(\theta), \tag{1}$$

where

$$L_k(\theta) = \sum_{(\mathbf{x},y)\in\mathcal{D}_k} \ell_{CE}(\theta, \mathbf{x}, y) \tag{2}$$

denotes the local objective of client k, *i.e.*, the empirical risk over the client's local dataset $\mathcal{D}_k$ of image $\mathbf{x}$, label y pairs, with ℓ_{CE} being the cross entropy loss, whereas the proportion of data hold by each client $\omega_k = \frac{|\mathcal{D}_k|}{\sum_\xi |\mathcal{D}_\xi|}$ is used as the the aggregation weight. While such design aligns with the empirical risk of centralized machine learning when client data are independent and identically distributed (IID), it becomes less effective when client data are heterogeneous. With FedGR, we aim to tackle the challenges presented by this heterogeneity, thereby enhancing performance uniformity for Federated Learning with Distinct Client Types (FL-DCT). Following (Li et al. 2020), we modify Equation (1) for an FL-DCT objective f_{DCT} that considers both individual and type-based performance uniformity. With T types each consisting of N_i clients, FedGR utilize a global aggregation objective of

$$\min_{\theta} f_{DCT}(\theta) = \sum_{i=1}^{T}\sum_{j=1}^{N_i} \omega_{ij}(L_{ij}(\theta)^{1-\beta}\tilde{L}_i(\theta)^{\beta})^{q+1}, \tag{3}$$

where the ij subscript denotes the j^{th} the client under the i^{th} type, while $q > 0$ and $\beta \in [0, 1]$ are hyperparameters. Similar to Equation (1), ω_{ij} and L_{ij} denotes the data size portion and local objective of a client, respectively, while $\tilde{L}_i = \frac{1}{N_i}\sum_j^{N_i} L_{ij}$ is the average loss over all clients belonging to type i. By including both L_{ij} and $\tilde{L}_i$, along with the exponent $q > 0$, Equation (3) promotes performance uniformity in both the individual and type-based collective level by assigning larger aggregation weights to the lower performing clients and client-types. Furthermore, β controls the balance between individual and collective fairness.

Anonymized Clustering While directly leveraging explicit client-type information in Equation (3) seem straightforward, requiring clients to supply explicit type information may jeopardize privacy or cause mislabeling. To circumvent these issues, we aim to utilize anonymized clustering to automatically identify the type of each client. Thus, in each communication round, in addition to sending the model update, each client also sends a *client representation* $\tilde{h}$, created through *averaging* the type-specific prompt h over their private dataset. Specifically, to avoid class imbalance revealing client data information, $\tilde{h}$ is calculated as

$$\tilde{h} = \frac{1}{C}\sum_{c}^{C}\frac{1}{|\mathcal{D}_c|}\sum_{\mathbf{x}\in\mathcal{D}_c} h_{\mathbf{x}}, \tag{4}$$

where $\mathcal{D}_c$ indicates the subset of a client data that belongs to class c while $h_{\mathbf{x}}$ indicates the type-specific prompt h crafted from image $\mathbf{x}$ via the ViT encoder and the GC-Net.

Leveraging the $\tilde{h}$ of each client, we utilize Gaussian Mixture Model (GMM) (Xu and Jordan 1996) to cluster clients into a predetermined number of clusters (T').[3]

[3]FedGCR performs well even with $T' \neq T$ (See Section. Ablation).

FedGR thereby operates Equation (3) by the cluster results. Since the clustering algorithm relies on GC-Net to generate prompts that are correctly aligned to the client type, it may be less accurate in the beginning. Therefore, we devise

$$\beta = \delta(1 - \gamma^{r-1}), \tag{5}$$

where r is the round number starting from 1 to R. In the first round, i.e., $r = 1$, β equals to 0. Then, γ^{r-1} exponentially approaches 0 such that β approaches to δ.

Note that FedGR also calculates the *cluster centers* $\mathbf{H}$, the mean of $\tilde{h}$ over each cluster, and returns them to all clients to facilitate the *group customization loss* detailed below.

The Local Learning Objective

Finally, we detail the local learning objective utilized in FedGCR. In particular, we modify Equation (2) and replace ℓ_{CE} with a more sophisticated

$$\ell = \ell_{CE} + \lambda_1 \ell_{GC} + \lambda_2 \ell_{RA}, \tag{6}$$

where the *group customization loss* ℓ_{GC} is added to better guide the prompt customization processes in FedGC and the *representation alignment loss* ℓ_{RA} to prevent the local training process results in image representations that deviates from the shared model. In particular, while the cross-entropy loss ℓ_{CE} provides provides supervision learning over the classification predicted $\hat{y}$, both ℓ_{GC} and ℓ_{inv} leverages a contrastive learning approach to guide the *type-specific prompt* h and the image representation z with server aggregated results, respectively.

The Group Customization Loss FedGCR employs the group customization loss ℓ_{GC} to guide the creation of type-specific prompts h using GC-Net. Our aim is twofold. On the one hand, we aim to foster a self-supervised mechanism enabling GC-Net to generate h that can be aptly and accurately clustered according to the client's data domain. Thus, leveraging the cluster center prompts H, we incorporate a contrastive learning loss design (Oord, Li, and Vinyals 2018) in ℓ_{GC} to guide each client's type-specific prompts h to become more aligned with its current clusters. On the other hand, we wish spur the continual improvement of customization by GC-Net. Thus, we device another term using each client's averaged prompt from the previous round $\tilde{h}_{prev}$ as a negative sample within the contrastive loss design. Intuitively, the design presents a balance between *exploitation* of the current clustering result and *exploration* of a better clustering, leading to enhanced customization. Formally, the group customization loss is written as

$$\ell_{GC} = -\log \frac{\exp(h \cdot H_i/\tau)}{\exp(h \cdot \tilde{h}_{prev}/\tau) + \sum \exp(h \cdot H_t/\tau)}, \tag{7}$$

where i denotes the cluster which the client belongs to, H_i is the i^{th} cluster center, $\sum$ indicates summation over all clusters ($t = 1$ to T'), and τ denotes the temperature parameter.

The Representation Alignment Loss Lastly, to prevent client representation drifting hampering the convergence of FL training (Jiang, Wang, and Dou 2022; Li, He, and Song 2021), we use the *representation alignment loss* ℓ_{RA}, a contrastive loss on the image representation z. In particular, for each communication round, we denote the local FedGC parameters in the prior round as θ_{prev}, the received parameters as θ_0, and the image representations derived by the corresponding parameters as z_{prev} and z_0, respectively.

$$\ell_{RA} = -\log \frac{exp(z \cdot z_0/\tau)}{exp(z \cdot z_0/\tau) + exp(z \cdot z_{prev}/\tau)}, \tag{8}$$

where the current parameters (producing z) align with the global model (z_0) and improve over the previous result (z_{prev}).

Algorithm 1: FedGCR for FL with Distinct Client Types.

Input: ViT, initial parameter $\theta^0 \equiv \{\psi^0, \phi^0, \mathbf{p}^0\}$ (GC-Net, MLP, global prompts). Hyperparameters $\eta, T', K, M, \lambda_1, \lambda_2, \gamma, \tau, q$. Client data and corresponding proportion denoted by $\mathcal{D}$ and ω.

ServerExecutes:

1: Initialize cluster center $\mathbf{H}^0$ as NULL.
2: **for** each round r in $1, \ldots, R$ **do**
3:   **for** each client k in $1, \ldots, K$ **in parallel do**
4:     $L_k^r, \tilde{h}_k^r, \theta_k^r \leftarrow$ **LocalUpdate**$(\theta^{r-1}, \mathbf{H}^{r-1}, \mathcal{T}[k])$
5:   $\mathbf{H}^r, \mathcal{T} \leftarrow Cluster(\{\tilde{h}_k^r\}_{k=1}^K, T')$
6:   Prepare $\{N_i\}$ and $(\cdot)_k \leftrightarrow (\cdot)_{ij}$ by $\mathcal{T}$ for $\theta_k, \omega_k, L_k^r$.
7:   $\beta \leftarrow 0.5 - 0.5\gamma^{r-1}$
8:   $L'_{ij} = (L_{ij}^r)^{1-\beta}(\frac{1}{N_i}\sum_{k=1}^{N_i} L_{ik}^r)^\beta$
9:   $\theta^r \leftarrow \sum_{i=1}^{T'}\sum_{j=1}^{N_i} \omega_{ij} \left(L'_{ij}\right)^{q+1} \theta_{ij}$

LocalUpdate$(\theta_0, \mathbf{H}, i)$:

1: Initialize $\ell \leftarrow 0$; Obtain z_0 by θ_0
2: **for** each epoch m in $1, \ldots, M$ **do**
3:   $L \leftarrow 0$; $\{\psi, \phi, \mathbf{p}\} \leftarrow \theta_{m-1}$,
4:   **for** each $(\mathbf{x}, y)$ in $\mathcal{D}$ **do**
5:     $h \leftarrow$ GC-Net$(\psi,$ ViT$(\mathbf{x}))$
6:     $z \leftarrow$ ViT$(\mathbf{x} \mid \mathbf{p} + h)$
7:     $\ell_{CE} \leftarrow \ell_{CE} + CrossEntropy(MLP(\phi, z), y)$
8:     **if** $\mathbf{H}$ is NULL **then**
9:       $L \leftarrow L + \ell_{CE}$
10:     **else**
11:       $\ell_{GC} \leftarrow$ Equation (7) with $(h, \tilde{h}, \mathbf{H}, i, \tau)$
12:       $\ell_{RA} \leftarrow$ Equation (8) with (z, z_0, z_{prev}, τ)
13:       $L \leftarrow L + (\ell_{CE} + \lambda_1 \ell_{GC} + \lambda_2 \ell_{RA})$
14:   $\theta_m \leftarrow \theta_{m-1} - \eta \nabla L$
15: $z_{prev} \leftarrow z$; $\tilde{h} \leftarrow$ Equation (4)
16: **return** $L, \tilde{h}, \theta_M$

The Full FedGCR Algorithm

Algorithm 1 outlines the FedGCR procedure for FL with Distinct Client Types. It consists of two main components: **ServerExecute** and **LocalUpdate**. In the server side, after the initialization of the cluster center (Line 1), the server first enforces the update of the local clients (Line 4). This update leverages the parameter θ, the cluster centers $H = H_1, \ldots, H_{T'}$ where T' is the number of client types, and the client cluster id $\mathcal{T}[k]$ for client k computed in the previous step. Afterward, the algorithm updates the T' clusters given the updated aggregated representation $\tilde{h}_k^r$ for each client k (Line 6). As a result, we obtain the updated cluster centers H^r at step r and the cluster indexes $\mathcal{T}$ that encode the anonymized clustering results. The final step (Line 9) is the group reweighting aggregation described in Equation (3). These steps repeat for the R server aggregation rounds.

The **LocalUpdate** component performs the local update of the client model. The process first operates the GC-Net$(\psi, \cdot)$ module (Line 5) to derive the type-specific prompt h. It then retrieves the image representation z, with $\mathbf{x}|\mathbf{p}+h$ denoting the concatenation of the image patches ($\mathbf{x}$) and the combined prompts ($\mathbf{p}+h$) in Line 6. Then, the algorithm updates the three components of the loss: ℓ_{CE} (Line 7), ℓ_{GC} (Line 11), and ℓ_{RA} (Line 12). Lastly, it updates the parameter θ. The process proceeds iteratively through the client training epochs and returns the loss, the average (and anonymized) client representation $\tilde{h}$, and the parameter θ.

Experiments

Experiment Settings

Datasets We evaluate the proposed FedGCR for FL-DCT on the three multi-domain image classification datasets. Digit-Five (Zhou et al. 2020) combines MNIST, SVHN, USPS, SynthDigits, and MNIST-M to exhibit 5 numeric digit image domains. DomainNet (Peng et al. 2019) includes 6 real-world image types including Clipart, Infograph, Painting, Quickdraw, Real (*i.e.*, photographs), and Sketch. PACS (Li et al. 2017) consists of 4 different types of pictures including Photo, Art Painting, Cartoon, and Sketch.

Domain Imbalance Factor (DIF) In this study, we investigate FL-DCT by using each *domain* as a distinct *client-type*. In particular, each client holds private data from a singular domain within the multi-domain datasets. We further scrutinize the imbalanced setting through the application of the *domain imbalance factor (DIF)* (Cui et al. 2019), where the client count of each client-type N_i is organized in a geometric series, with the DIF being the quotient of the highest and lowest number. We analyze scenarios with DIF values of $1, 5, 10$ (DIF$=1$ is a balanced setting).

Baseline Methods We compare the proposed FedGCR with the following baselines. (i) The **original** FL algorithm, FedAvg (McMahan et al. 2017); (ii) **fairness-FL** methods, proposed to improve the fairness/uniformity of individual client performances: q-FFL (Li et al. 2020), AFL (Mohri, Sivek, and Suresh 2019), and TERM (Li et al. 2021a); (iii) **domain-FL** methods, proposed to enhance domain adaptation: Harmo-FL (Jiang, Wang, and Dou 2022), FedSAM (Qu et al. 2022), and FedMix (Yoon et al. 2021).

Evaluation Metrics We conduct evaluations on both *performance* and *fairness*. In particular, we measure the performance by the average classification accuracy over all clients (Avg). For fairness, we follow (Li et al. 2020) to measure the standard deviations between individual and type performances. Namely, we measure the standard deviation between the averaged accuracy of each client-types (σ_{type}) and that between individual clients (σ_{client}). The objective is to achieve higher values in Avg and lower values in σ_{type} and σ_{client}.

Implementation Details We employ ViT-B/16 (Dosovitskiy et al. 2021) pretrained on ImageNet (Deng et al. 2009). Other configurations include: server rounds $R = 50$, client training epochs $M = 1$, hyperparameters $\lambda_1 = 0.5, \lambda_2 = 0.1, \tau = 0.5, \gamma = 0.5, q = 1$, and learning rate $\eta = 1 \times 10^{-3}$ using the AdamW optimizer (Loshchilov and Hutter 2017). The cluster number T' is, by default, set to the domain count for each dataset. The entire experiment is conducted in PyTorch and executed on a single NVIDIA Tesla V100 GPU.

Main Results

Table 1 and Table 2 present the experiment results of Digit-Five and DomainNet, respectively. The results reveal that *FedGCR* achieves both the best performance *and* the greatest fairness in terms of performance uniformity among all compared methods. It is worth noting that past FL methods for (individual) fairness shows slightly better uniformity than those devoted to domain adaptation in both domain-wise and client-wise metrics. However, both groups are significantly outperformed by FedGCR, demonstrating the superiority of FedGCR in achieving *fairness*. Similarly, domain adaptation methods find better average accuracy than the methods for fairness, yet are still outperformed by FedGCR due to the customization power of FedGC. Finally, we observe that for each method, results on all three metrics degrades as the DIF increases, revealing the impact of imbalanced setting. Nevertheless, FedGCR degrades the least compares to all baselines, such that the gap between the baselines and FedGCR are the widest in the most imbalanced case of DIF $= 10$.

Qualitative Studies

Sensitivity Tests on Cluster Count We assess the accuracy of the anonymized clustering and FedGCR performances under different cluster counts in Digit-Five (with $T = 5$ client-types). Table 3 shows the clustering accuracy, the percentage of clients correctly grouped with the majority client-type in their cluster. We observe that 100% accuracy is attained whenever cluster count is larger than the number of types $T' \geq T = 5$. This suggests that FedGCR can function accurately by setting an adequately large T', even when the exact number of client-types is unspecified.

Table 4, present FedGCR performances with varying cluster count T' for DIF$=1$ and DIF$=10$. Comparing the results of different $T' \neq 5$ in Table 4 to $T' = T = 5$ in Table 1, we find that 1) FedGCR outperforms all baselines even under $T' < T$, where incorrect clustering is guaranteed. This may be explained by FedGCR providing clustering based on contrastive loss of data representations $\tilde{h}$, such that the most disparate types are clustered for FedGC to provide customization for the most required differences, and FedGR reweights between client-types that has the most disparate performance differences. In addition, we observe that providing slightly more cluster can result in the best performance, where $T' = 6$ consistently finds the best results. As the clustering accuracy is 100% under these conditions, we find that the clusters subdivide the client-type groups with larger inter-type differences, thereby providing even better customization and reweighting.

Method	DIF=1			DIF=5			DIF=10		
	Avg(↑)	σ_{type}(↓)	σ_{client}(↓)	Avg(↑)	σ_{type}(↓)	σ_{client}(↓)	Avg(↑)	σ_{type}(↓)	σ_{client}(↓)
FedAvg	68.19 (0.26)	19.76 (0.06)	19.56 (0.48)	63.39 (0.25)	24.77 (0.16)	22.23 (0.33)	60.62 (1.13)	26.76 (1.09)	21.84 (0.95)
AFL	69.15 (0.73)	17.03 (0.79)	16.41 (0.75)	65.57 (0.39)	21.34 (0.13)	18.89 (0.03)	61.81 (0.95)	23.58 (0.49)	18.70 (0.27)
q-FFL	66.78 (0.95)	17.03 (0.79)	19.38 (1.09)	64.16 (0.83)	24.60 (0.85)	21.80 (0.91)	60.69 (0.92)	25.52 (0.77)	20.56 (0.72)
TERM	51.48 (28.11)	19.14 (1.12)	19.0 (0.86)	62.58 (0.61)	25.18 (0.54)	23.05 (0.49)	59.52 (0.46)	27.02 (0.38)	22.22 (0.33)
Harmo-FL	74.06 (1.01)	18.52 (0.31)	18.72 (1.01)	69.17 (0.86)	24.15 (0.81)	21.22 (0.84)	64.33 (1.88)	26.95 (1.1)	21.28 (1.21)
FedSAM	74.76 (1.01)	18.52 (0.31)	18.64 (0.48)	69.75 (0.13)	23.81 (0.25)	20.87 (0.11)	64.06 (1.47)	27.82 (0.81)	21.75 (0.98)
FedMix	51.44 (28.30)	19.01 (0.35)	19.0 (0.35)	62.64 (0.08)	25.23 (0.42)	22.92 (0.03)	59.46 (0.09)	27.37 (0.11)	22.47 (0.18)
FedGCR	**85.82** (0.07)	**8.55** (0.37)	**8.55** (0.37)	**85.11** (0.09)	**9.86** (0.22)	**8.40** (0.05)	**82.74** (0.47)	**12.26** (0.36)	**8.52** (0.19)

Table 1: Digit-Five results. Best results in bold, standard deviation over 3 repeated runs are shown in the parentheses.

Method	DIF=1			DIF=5			DIF=10		
	Avg(↑)	σ_{type}(↓)	σ_{client}(↓)	Avg(↑)	σ_{type}(↓)	σ_{client}(↓)	Avg(↑)	σ_{type}(↓)	σ_{client}(↓)
FedAvg	82.03 (0.07)	14.84 (0.14)	15.17 (0.21)	80.69 (0.44)	16.52 (0.59)	11.81 (0.31)	77.67 (0.54)	19.80 (0.59)	13.76 (0.31)
AFL	81.70 (0.24)	14.25 (0.10)	14.43 (0.49)	80.99 (0.27)	15.46 (0.08)	11.18 (0.09)	79.61 (0.06)	16.83 (0.19)	11.09 (0.04)
q-FFL	81.86 (0.32)	15.04 (0.17)	15.34 (0.15)	81.17 (0.60)	16.43 (0.68)	11.76 (0.38)	78.07 (0.52)	18.74 (0.47)	12.59 (0.35)
TERM	81.93 (0.35)	14.85 (0.19)	15.13 (0.3)	80.56 (0.05)	16.47 (0.06)	11.87 (0.03)	77.67 (0.47)	19.53 (0.78)	13.63 (0.43)
Harmo-FL	83.16 (0.10)	14.52 (0.23)	14.89 (0.5)	82.79 (0.47)	16.47 (0.05)	11.50 (0.23)	79.56 (0.22)	18.64 (0.21)	12.53 (0.35)
FedSAM	83.67 (0.12)	13.77 (0.10)	13.77 (0.21)	82.96 (0.35)	15.95 (0.57)	11.37 (0.32)	81.04 (0.28)	17.44 (0.46)	11.74 (0.39)
FedMix	81.78 (0.47)	15.12 (0.18)	15.35 (0.05)	80.57 (0.05)	16.35 (0.04)	11.86 (0.06)	77.55 (0.30)	19.46 (0.67)	13.63 (0.43)
FedGCR	**84.08** (0.29)	**10.89** (0.53)	**11.18** (0.65)	**83.67** (0.19)	**13.18** (0.07)	**9.04** (0.35)	**81.88** (0.52)	**14.88** (0.08)	**9.07** (0.17)

Table 2: DomainNet results. Best results in bold, standard deviation over 3 repeated runs are shown in the parentheses.

T'	3	4	5	6	7
DIF= 1	60.00%	80.00%	100.00%	100.00%	100.00%
DIF= 5	69.23%	92.31%	100.00%	100.00%	100.00%
DIF= 10	81.82%	95.45%	100.00%	100.00%	100.00%

Table 3: Clustering accuracy with Digit-Five.

	DIF=1			DIF=10		
	Avg	σ_{type}	σ_{client}	Avg	σ_{type}	σ_{client}
$T' = 3$	85.65	8.67	8.60	82.62	8.88	8.81
$T' = 4$	85.44	8.62	8.60	82.49	12.63	8.77
$T' = 6$	**85.86**	**8.45**	**8.16**	**82.99**	**11.92**	**8.36**
$T' = 7$	85.84	8.39	8.14	82.52	12.51	8.62

Table 4: Results with different cluster count T' in Digit-Five

Ablation Study Table 5 presents the ablation results of FedGCR in Digit-Five (under DIF= 10). In particular, the first row of Table 5 presents plain FL (*i.e.*, FedAvg), whereas FedGC and FedGR are both provided with anonymized clustering to operate correctly. We observe that FedGC provides a more substantial improvement in both performance and uniformity than FedGR since the former directly interacts with the distinct type condition, while the latter indirectly provides reweighting as a remedy. However, FedGR still provides a substantial improvement, allowing FedGCR to surpass baseline methods by a more notable margin.

FedGR	FedGC	Avg(↑)	σ_{domain}(↓)	σ_{client}(↓)
–	–	60.62	26.76	21.84
✓	–	63.01	23.03	17.75
–	✓	79.49	16.57	11.61
✓	✓	**82.74**	**12.26**	**8.52**

Table 5: Ablation results comparing FedGR and FedGC performances in Digit-Five under DIF= 10.

Conclusion

This paper introduces *Federated learning with Group Customization and Reweighting (FedGCR)*, a novel Federated Learning approach that addresses the challenges posed by client-type diversity, encouraging more uniform performance among groups of clients. FedGCR consists of two core components: FedGC for customized client-type adaptation and FedGR for ensuring fairness across different client types. Leveraging ViT, FedGC employs global and type-specific prompts to tailor model for each client type. Meanwhile, FedGR adopts a group-wise reweighting mechanism to ensure uniform performance across client types, promoting collective fairness without a priori knowledge of client types, thus ensuring data privacy. Extensive experiments on real-world datasets confirm its superiority for both performance and fairness metrics, demonstrating its effectiveness in accounting for client diversity and promoting fairness across groups of clients. While we have explored a static setting of client types, our work leads to future explorations on scenarios where client types are dynamic or even hybrid, presenting fascinating opportunities for future works.

Acknowledgments

This work was supported in part by the National Science and Technology Council, Taiwan, under grant NSTC-112-2223-E-002-015, and by the Ministry of Education, Taiwan, under grant MOE 112L9009.

This work of EP was supported by the spoke "FutureHPC & BigData" of the ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing funded by European Union – NextGenerationEU.

References

Chen, J.; Jiang, M.; Dou, Q.; and Chen, Q. 2023. Federated Domain Generalization for Image Recognition via Cross-Client Style Transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 361–370.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, 2089–2099. PMLR.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9268–9277.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Du, W.; Xu, D.; Wu, X.; and Tong, H. 2021. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, 181–189. SIAM.

Ezzeldin, Y. H.; Yan, S.; He, C.; Ferrara, E.; and Avestimehr, S. 2023. Fairfed: Enabling group fairness in federated learning.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2020. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597.

Hu, Z.; Shaloudegi, K.; Zhang, G.; and Yu, Y. 2020. Fedmgda+: Federated learning meets multi-objective optimization. *arXiv preprint arXiv:2006.11489*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, 709–727. Springer.

Jiang, M.; Wang, Z.; and Dou, Q. 2022. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1087–1095.

Kominers, S. D.; and Wu, L. 2023. Threads Foreshadows a Big — and Surprising — Shift in Social Media. Harvard Business Review.

Kulkarni, V.; Kulkarni, M.; and Pant, A. 2020. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 794–797. IEEE.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 965–978. IEEE.

Li, Q.; He, B.; and Song, D. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10713–10722.

Li, T.; Beirami, A.; Sanjabi, M.; and Smith, V. 2021a. Tilted Empirical Risk Minimization. In *International Conference on Learning Representations*.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021b. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, 6357–6368. PMLR.

Li, T.; Sanjabi, M.; Beirami, A.; and Smith, V. 2020. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*.

Li, X.; JIANG, M.; Zhang, X.; Kamp, M.; and Dou, Q. 2021c. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *International Conference on Learning Representations*.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Liu, Q.; Chen, C.; Qin, J.; Dou, Q.; and Heng, P.-A. 2021. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1013–1023.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Marfoq, O.; Neglia, G.; Vidal, R.; and Kameni, L. 2022. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, 15070–15092. PMLR.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Mohri, M.; Sivek, G.; and Suresh, A. T. 2019. Agnostic federated learning. In *International Conference on Machine Learning*, 4615–4625. PMLR.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Papadaki, A.; Martinez, N.; Bertran, M.; Sapiro, G.; and Rodrigues, M. 2022. Minimax demographic group fairness in federated learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 142–159.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1406–1415.

Pierce, D. 2023. Can ActivityPub save the internet? The Verge.

Qu, Z.; Li, X.; Duan, R.; Liu, Y.; Tang, B.; and Lu, Z. 2022. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, 18250–18280. PMLR.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Sattler, F.; Müller, K.-R.; and Samek, W. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8): 3710–3722.

Sheller, M. J.; Reina, G. A.; Edwards, B.; Martin, J.; and Bakas, S. 2019. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*, 92–104. Springer.

Tan, Y.; Long, G.; Ma, J.; Liu, L.; Zhou, T.; and Jiang, J. 2022. Federated Learning from Pre-Trained Models: A Contrastive Learning Approach. In *NeurIPS*.

Xu, L.; and Jordan, M. I. 1996. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8(1): 129–151.

Xu, X.; Wu, Z.; Verma, A.; Foo, C. S.; and Low, B. K. H. 2023. FAIR: Fair collaborative active learning with individual rationality for scientific discovery. In *International Conference on Artificial Intelligence and Statistics*, 4033–4057. PMLR.

Yoon, T.; Shin, S.; Hwang, S. J.; and Yang, E. 2021. FedMix: Approximation of Mixup under Mean Augmented Federated Learning. In *9th International Conference on Learning Representations, ICLR*.

Yue, X.; Nouiehed, M.; and Al Kontar, R. 2023. Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2(1): 10–23.

Zhang, D. Y.; Kou, Z.; and Wang, D. 2020. FairFL: A Fair Federated Learning Approach to Reducing Demographic Bias in Privacy-Sensitive Classification Models. In *2020 IEEE International Conference on Big Data (Big Data)*, 1051–1060.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Zhong, A.; He, H.; Ren, Z.; Li, N.; and Li, Q. 2023. FedDAR: Federated Domain-Aware Representation Learning. In *The Eleventh International Conference on Learning Representations*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

Zhou, K.; Yang, Y.; Hospedales, T.; and Xiang, T. 2020. Learning to generate novel domains for domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, 561–578. Springer.

Zhou, T.; and Konukoglu, E. 2023. FedFA: Federated Feature Augmentation. In *The Eleventh International Conference on Learning Representations*.