Improving Federated Domain Generalization Through Dynamical Weights Calculated from Data Influences on Global Model Update

Zikun Zhou[1], Wen Huang[1*], Xingyi Wang[1], Zhishuo Zhang[2], Zhun Zhang[2], Jian Peng[1], Feihu Huang[3]
[1]Sichuan University
[2]University of Electronic Science and Technology of China
[3]Civil Aviation Flight university of China
zhouzikun@stu.scu.edu.cn {wen, jianpeng }@scu.edu.cn

Abstract

With the popularity of federated learning, federated domain generalization (FedDG) has attracted more and more attentions. Existing works of federated learning indicate that the generalization performance of the global model can be improved when the global model is obtained by aggregating local models according to a suitable weights. However, the existing methods to calculate weights do not fully utilize the data influences on the global model update, which gives us an opportunity to improve the generalization performance of the global model further. In this paper, we propose the method DI (data influences), which utilizes the data influences on the global model update to calculate dynamical weights of local model in each round of training. Specifically, the first component data influences calculator (DIC) of DI calculates the local weights of local model from the influences of each data on the global model update and we introduce the influences function to complete the calculation process. The second component data influences adjuster (DIA) of DI calculates the global weights (which are used in the aggregation process of the global model) from local weights. Extensive experiments indicate that our method improves the generalization performance of models significantly. In particular, our method improves model accuracy on benchmark datasets PACS, Office-Home, and Office-31 by 1.79%, 1.61%, and 2.39% on average, respectively. Source code is publicly available at github.

Code — https://anonymous.4open.science/r/Fed-DI

Introduction

With the popularity of federated learning (Fan et al. 2022; Li et al. 2020a), federated domain generalization (FedDG) has attracted more and more attentions. As demonstrated in Figure 1, FedDG (Liu et al. 2021; Jiang, Wang, and Dou 2022; Zhuang et al. 2022) aims at that: models trained by federated learning not only perform well on source domains involved in training, but also perform well on unseen domains not involved in training. In brief, FedDG aims to improve generalization of models, which is a meaningful but challenging task.

At present, the most effective method of improving FedDG is to aggregate local models according to suitable weights. Specifically, the global model is updated according to local models together with assigned weights. For example, FedAvg (McMahan et al. 2017) assigns weights to local models trained in each source domain by the data volumes of the source domain. Based on the idea that the global model with fair performance across all source domains has better generalization performance, GA (generalization adjuster) (Zhang et al. 2023) dynamically updates the weights of local models by the differences among models of different source domains.

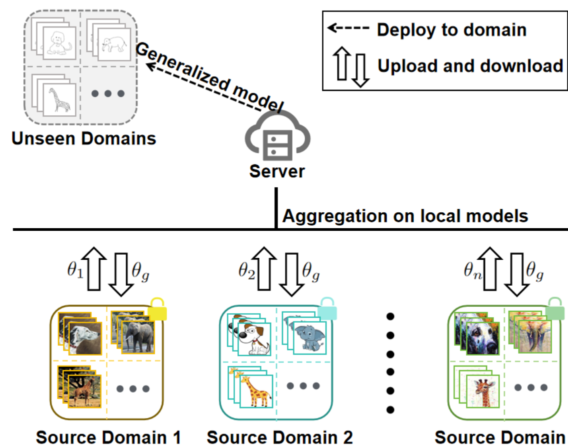


Figure 1: Scenario of FedDG. FedDG aims at generalizing models trained by source domains to unseen domains.

However, existing methods (McMahan et al. 2017; Zhang et al. 2023; Chen et al. 2022; Seunghan et al. 2024; Shenaj et al. 2023) of assigning weights to local models do not fully utilize the influences of data in source domains on the generalization performance of the global model. For example, FedAvg assigns weights to each local model only by the data volumes of source domain.

Assigning weights to local models according to data influences on global model update has great potential to further improve the generalization performance of global model. In each round of training, the global model is an aggregation of local models trained by the data in source domains, result-

[*]Corresponding author

ing in that the generalization performance of global model is inherently linked to the influences of the data on global model update. Consequently, assigning weights by data influences on global model update may unlock new potential for enhancing the global model's generalization performance across diverse domains.

In this paper, by more fully utilizing the data influences on the global model update, we propose a novel method DI to further improve FedDG. The first component of DI is DIC (Data Influences Calculator), which aims at evaluating the contribution of data in source domains to generalization performance of the global model. Specifically, when receiving the global model, the source domain evaluates influences of its data on the global model update by the latest local model and the received global model. The evaluation process is completed by the approximation of the influence function which is introduced in the section of preliminaries, and the local weights obtained are uploaded to the server together with new local models. The second component of DI is DIA (Data Influences Adjuster), which calculates the global weights from the local weights and local models. DIA can effectively guarantee the generalization performance of the global model when global model update is cased by a small amount of data. Our contributions are as follows

- We propose DIC module which can dynamically calculate the local weights of local models by the data influences on the global model update. To precisely evaluate the influences of data in source domains on the global model update, we introduce the approximation of influences function into DIC so that weights of local models can be calculated dynamically.
- We propose DIA module to adjust calculated local weights of local models trained in source domains. DIA can maintain the generalization performance of the global model even when the global model update is caused only by a small portion of data. The module has many advantages such as being easy to implement and sharing no raw data between source domains and the server.
- Through extensive experiments on benchmark datasets including PACS, OfficeHome, and Office-31 datasets, we demonstrate that the proposed method DI significantly outperforms the state-of-the-art weights strategy together with other methods, and the proposed method DI further improves the generalization performance of global model trained by federated learning.

Related Work

In real-world scenarios, the generalization performance of models may be low because the data distribution across different devices and organizations might differ significantly. As a potential method to improve generalization performance, FedDG has attracted more and more attentions. FedDG combines the strengths of federated learning (McMahan et al. 2016; Yang et al. 2019; Kairouz et al. 2021) and domain generalization (Muandet, Balduzzi, and Schölkopf 2013; Blanchard, Lee, and Scott 2011; Torralba and Efros 2011). Specifically, it not only can obtain better generalization performance, but also can preserve data privacy. In FedDG, using proper weights to aggregate local models is the key to improving the generalization performance of the global model (Zhang et al. 2023). FedAvg (McMahan et al. 2017) aggregates local models based on the amount of data per source domain. Yuan (Yuan et al. 2023) introduced a semantic similarity-based attention mechanism that dynamically assigns weights to cross-layer pairs. Zhang (Zhang et al. 2023) proposed the federated generalization adjustment (GA) method to optimize the generalization performance of the global model by dynamically calibrating weights of local models. In FedDG, optimizing the aggregation weights can effectively improve the generalization performance of the global model, which is crucial to addressing the challenges posed by heterogeneous data distribution among source domains. However, existing methods (Li et al. 2020b; Nagalapatti and Narayanam 2021; Zhao et al. 2018) of assigning weights do not utilize the influences of data in source domains on the global model updates, which leaves us an opportunity to further improve the generalization performance of the global model.

Preliminaries

Influence Function

Influence function, a classic technique from robust statistics (Cook and Weisberg 1980), tells us how the model parameters change as we up weight a training point by an infinitesimal amount (Koh and Liang 2017). Due to this ability, influence function is used to quantify the contribution of training data to model update in our method.

Simply, train a model θ on dataset $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ and loss function is $\mathcal{L}(x, y, \theta)$. The trained model is

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \mathcal{L}(x_i, y_i; \theta). \tag{1}$$

To evaluate the influence of i-th data (x_i, y_i) on model $\hat{\theta}$, a new model is trained as follows

$$\hat{\theta}_i = \arg\min_{\theta} \sum_{j=1}^{n} \left[(1-\phi_j)\mathcal{L}(x_j, y_j; \theta)\right], \tag{2}$$

where $1 \leq j \leq n, \phi_j = 0$ except $j = i$. The influence $I_{(i,f)}$ of (x_i, y_i) on the model θ can be obtained as:

$$I_{(i,f)} = f(\hat{\theta}_i) - f(\hat{\theta}), \tag{3}$$

where f is the function (which is tightly related to the application scenario) to quantify the difference between models.

Although applying influence function to measure the influence of data on the model requires repeated training of the model, there is a relatively efficient first-order approximation of equation (3). Thus, through the approximation, we can avoid repeated training of the model, which is elaborated on later.

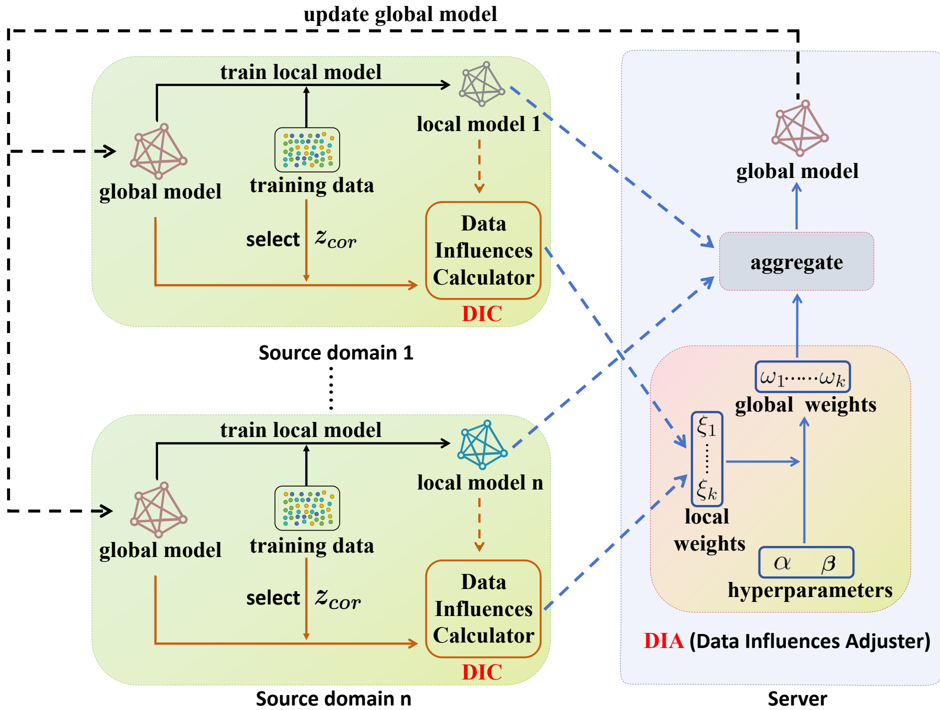


Figure 2: Workflow of DI. DICs calculate local weights from local data, the local model, and the received global model. DIA calculates global weights from local weights. The global mode is obtained by aggregating local models according to global weights.

Method

Overview

The workflow of our method DI is demonstrated in Figure 2. In brief, after receiving the global model, the source domains train local models, and DICs calculate local weights by local data, the local model, and the received global model. The trained local models and calculated local weights are uploaded to aggregation server and then DIA calculates the global weights by local weights. To obtain the new global model, the server aggregates local models according to the global weights and local models. Finally, the new global model is sent to source domains again to start next round of training.

Calculation of Local Weights by DIC

DICs calculate local weights of each local model by influences of data in source domains on global model update. The concrete formula of calculating local weights is as follows

$$\xi_k = \sum_{i=1}^{N_k} \frac{\mathcal{I}(z_i, \hat{\theta}_k^r)}{N_k} G(z_{cor}, \theta_g^{r-1}). \tag{4}$$

Here, ξ_k represents the local weight of local model of k-th domain. N_k represents the number of data in k-th domain. $\mathcal{I}(z_i, \hat{\theta}_k^r)$ is used to quantify the influences of data z_i on process of local model update from θ_g^{r-1} to $\hat{\theta}_k^r$, where θ_g^{r-1} represents the global model obtained in $r-1$ round and $\hat{\theta}_k^r$ represents local model of k-th domain in r round. $G(z_{cor}, \theta_g^{r-1})$ is combined with $\mathcal{I}(\cdot,\cdot)$ to quantify the data influences on the change of model's generalization performance, where z_{cor} is a dataset randomly selected from the k-th source domain.

We use the influence function (Koh and Liang 2017) to obtain $\mathcal{I}(z_i, \hat{\theta}_k^r)$ of equation (4). Assume that there are N_k training samples $z_1, z_2 ... z_{N_k}$ in k-th source domain, where $z_i = (x_i, y_i)$, and let $L(z_i, \theta)$ represent the loss function of sample z_i under model parameters θ, then the empirical risk is

$$R(\theta) = \frac{1}{N_k} \sum_{i=1}^{N_k} L(z_i, \theta). \tag{5}$$

Based on Empirical Risk Minimization (ERM), we can obtain the model parameters as follows

$$\hat{\theta} = \arg\min_{\theta} R(\theta). \tag{6}$$

To investigate the effect of a particular training sample z_p on the model θ, we increase the weight of the sample z_p in the dataset by an infinitesimal quantity ϵ. In this case, the model parameters derived from the ERM would become

$$\hat{\theta}_{\epsilon, z_p} = \arg\min_{\theta} \Big(R(\theta) + \epsilon L(z_p, \theta) \Big). \tag{7}$$

The evaluation of the relationship (7) between the change in model and the change in training sample weights can be achieved by influence function. First, define the variable $\Delta_\epsilon = \hat{\theta}_{\epsilon, z_p} - \hat{\theta}$ to measure the change in the model

θ. Note that $\hat{\theta}$ is the model obtained from ERM before increasing weights of the sample, and therefore it is independent of ϵ. We have $\frac{d\hat{\theta}_{\epsilon,z_p}}{d\epsilon} = \frac{d\Delta_\epsilon}{d\epsilon}$. $\hat{\theta}_{\epsilon,z_p}$ is the minimization result of equation (7). Thus, the first-order derivative of $\Big(R(\theta) + \epsilon L\big(z_p, \theta\big)\Big)$ with respect to θ is zero, i.e., $\nabla R(\hat{\theta}_{\epsilon,z_p}) + \epsilon \nabla L(z_p, \hat{\theta}_{\epsilon,z_p}) = 0$.

When ϵ is an infinitesimal quantity, $\hat{\theta}_{\epsilon,z_p}$ tends to $\hat{\theta}$. We do a first-order Taylor expansion and Δ_ϵ can be derived as follows,

$$\Delta_\epsilon \approx -\left[\nabla^2 R(\hat{\theta}) + \epsilon \nabla^2 L(z_p, \hat{\theta})\right]^{-1} \cdot \left[\nabla R(\hat{\theta}) + \epsilon \nabla L(z_p, \hat{\theta})\right]. \tag{8}$$

Based on the equation (8) and (Ling 1984), we have

$$\mathcal{I}\left(z_p, \hat{\theta}\right) = \frac{d\hat{\theta}_{\epsilon,z_p}}{d\epsilon}\bigg|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla L\left(z_p, \hat{\theta}\right), \tag{9}$$

where

$$H_{\hat{\theta}} = \nabla^2 R\left(\hat{\theta}\right) = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L\left(z_i, \hat{\theta}\right). \tag{10}$$

In equation (4), we use $\mathrm{I}(\cdot,\cdot)$ to estimate the degree of influence of sample z_i on the local model update. A larger value of $\mathrm{I}(\cdot,\cdot)$ indicates that sample z_i has an larger influence on the trained local model. The sum $\sum_{i=1}^{N_k} \mathcal{I}(z_i, \hat{\theta}_k^r)$ denotes the influences of the source domain dataset on the local model.

Since $\mathcal{I}(\cdot,\cdot)$ does not typically represent the change of generalization performance when model is updated from θ_g^{r-1} to $\hat{\theta}_k^r$, we introduce $\mathrm{G}(\cdot,\cdot)$ to quantify the influences of data on the generalization performance. When we only use one data sample z_c^1, equation (9) can be rewritten as a closed-form expression to quantify the influence of data sample z_c^1 on the change of model's generalization performance (Koh and Liang 2017) i.e., $-\nabla_\theta L(z_c^1, \theta_g^{r-1}) H_{\hat{\theta}_k^r}^{-1} \nabla_\theta L(z_i, \hat{\theta}_k^r)$. To precisely quantify the influences of data on change of model's generalization performance, we random pick a portion of the source domain dataset as the core dataset z_{cor} to calculate the influences. Thus, $G(z_{cor}, \theta_g^{r-1})$ is

$$G(z_{cor}, \theta_g^{r-1}) = \frac{\sum_{i=1}^{N_c} \nabla_\theta L(z_c^i, \theta_g^{r-1})}{N_c}. \tag{11}$$

Calculation of Global Weights by DIA

The local weights are calculated locally in source domains so the calculation of these local weights does not take other source domains into account. To bridge the gap, we propose DIA to calculate global weights of local models from local weights. In particular, the global weight $\bar{w}_k$ of the k-th local model consists of two terms $w_{(k,1)}$ and $w_{(k,2)}$, namely

$$\bar{w}_k = w_{(k,1)} + w_{(k,2)}. \tag{12}$$

The first term $w_{(k,1)}$ is to trade off the calculated local weights and totally equal weight and is as follows

$$w_{(k,1)} = \frac{1}{n} + \frac{1}{2^\alpha}(\xi_k^{'} - \frac{1}{n}), \tag{13}$$

Algorithm 1: Training process of method DI

Input: datasets of n source domains $D_1, D_2, ..., D_n$, initial global model θ_g^0, total training round R, local training algorithm $Tra\,(\theta, D)$ with input global model θ and dataset D.

Output: global model θ_g^R

- **server:** send initial global model θ_g^0 to all source domains
- **for** each $r \in [1, R]$ **do**
  - **k-th source domain**$(1 \le k \le n)$:
    - train the local model $\hat{\theta}_k^r$ on the k-th domain: $\hat{\theta}_k^r = Tra\,(\theta_r^g, D_k)$.
    - get the empirical loss on client D_k: $R\left(\hat{\theta}_k^r, D_k\right)$
    - compute the $\mathcal{I}(z_i, \hat{\theta}_k^r)$ and $G(z_{cor}, \theta_g^{r-1})$.
    - k-th source domain calculates ξ_k and sends it to server.
  - **server:**
    - calculate ω_k, $1 \le k \le n$.
    - aggregate $\hat{\theta}_k^r$ with ω_k to get a new global model: $\theta_g^r = \sum_{k=1}^n \omega_k \cdot \hat{\theta}_k^r$
    - broadcast θ_g^r to all source domains.
- **end for**

---

where $\xi_k^{'} = \frac{\xi_k}{\sum_{i=1}^n \xi_i}$, $1 \le k \le n$ and $\alpha \in [0, +\infty]$. Specifically, when $\alpha = 0$, global weights are allocated entirely according to $\xi_k^{'}$. When α approaches $+\infty$, global weights are $1/n$, namely totally equal weights.

The second term $w_{(k,2)}$ is to adjust $w_{(k,1)}$ when the amount of data in the source domain is small or there is noise in z_{cor}. The second term is

$$w_{(k,2)} = \frac{1}{n} + \frac{1}{2^\beta}(e^{-\varphi_k^{'}} - \frac{1}{n}). \tag{14}$$

$w_{(k,2)}$ prevents the global weights from being excessively biased by the loss of source domain. In particular, the local optimization of k-th domain is formulated as

$$\hat{\theta}_k = \arg\min_\theta \frac{1}{N_k}\sum_{i=1}^{N_k} L\left(z_i, \theta_k\right).$$

We define $\varphi_k = \sum_{i=1}^{N_k} L\big(f(x_i^k, \theta_j), y_i^k\big)$, where f is classification function of source domain, x_i^k together with y_i^k is i-th data in k-th source domain. And then $\varphi_k^{'} = \frac{\varphi_k}{\sum_{i=1}^n \varphi_i}$

Finally, DIA normalizes the calculated global weights $\bar{w}_j$ as follows

$$\omega_k = \frac{\bar{w}_k}{\sum_{i=1}^n \bar{w}_i}. \tag{15}$$

The training process of DI is described in Algorithm 1.

Experiments

In this section, we compare our method with state-of-the-art methods in terms of generalization performance and robustness through extensive experiments. And then, we show the effect of each component on generalization performance by ablation experiments.

Datasets. The experiments are performed on three domain-adaptive benchmark datasets, including Office-Home (Venkateswara et al. 2017) including 4 domains, Office-31 (Saenko et al. 2010) including 3 domains, and PACS (Li et al. 2017) including 4 domains.

Main Results

Baseline Methods. We compare our method with representative methods in the domain generalization and federated learning. Specifically, RSC (Representation Self-Challenging) (Huang et al. 2020) method is a representative regularisation-based method of domain generalization. AM (Amplitude Mix) (Xu et al. 2021) is a representative Fourier-based augmentation method. AM can make the model not overly dependent on the low-level statistical properties and can enable the model to capture the essential features better. FedAvg (Federated Averaging) (McMahan et al. 2017) is a representative federated learning method (Li et al. 2023; Lim et al. 2020; Lyu, Yu, and Yang 2020) designed for data heterogeneity. It improves the generalization performance of the global model by bringing regularisation effects into averaging process. Scaffold (Stochastic Controlled Averaging) (Karimireddy et al. 2020) is another heterogeneity method designed for data heterogeneity. GA (Zhang et al. 2023) improves generalization performance of the global model by aggregating local models through suitable weights.

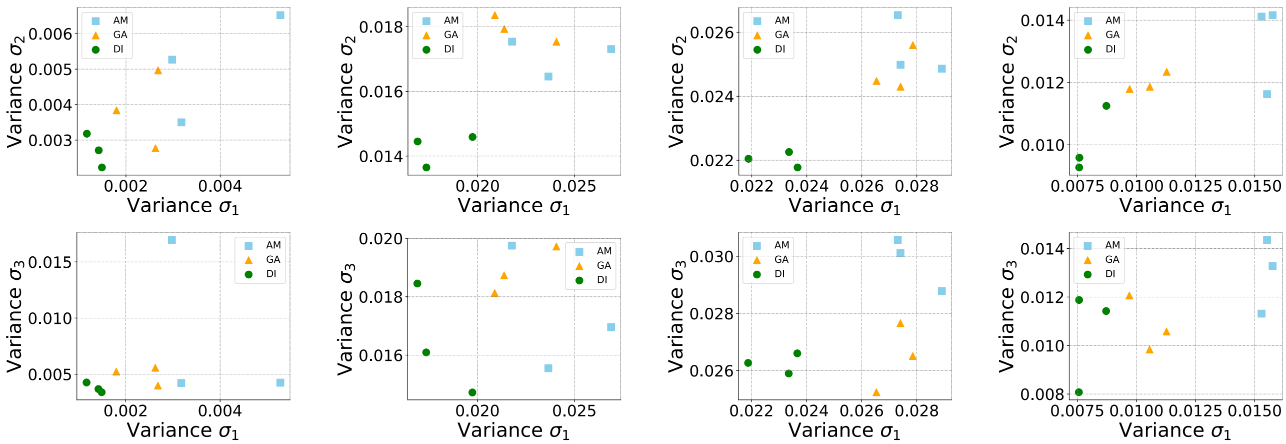
Training Details. As for the training details, we follow the setting of GA (Zhang et al. 2023). Specifically, the trained model is the ImageNet pre-trained ResNet18. The number of epochs of source domains is 5 and the number of training rounds is 40. For fairness, the batch size and learning rate in all experiments are 16 and 0.001 respectively. To achieve the best performance of baseline methods, the number of training rounds of baseline methods is set to 100. To obtain stable experiment results, we run all experiments three times and report the mean of model accuracy.

Similar to other methods of federated domain generalization, we evaluate generalization performance by the domain-by-domain exclusion test. Specifically, we select one domain as the target domain (the unseen domain) in turn and the rest domains are source domains for training.

In federated learning, data scarcity is a common issue, and the performance of many methods degrades under low-volume training sets (Chen et al. 2023; Xiao et al. 2023). To explore the robustness of our method, we gradually decrease the proportion of training set so that we can

Method	Percent	OfficeHome					Office-31			
		P	A	C	R	Avg	A	D	W	Avg
AM	22.5%	54.97	43.63	38.91	57.05	48.64	52.39	86.74	80.58	73.23
	45%	57.10	45.61	41.77	59.55	51.00	53.64	95.24	82.39	77.09
	90%	59.94	46.21	44.59	60.96	52.92	54.28	97.65	88.38	78.10
AM +DI	22.5%	**55.90**	**45.77**	**39.25**	**58.73**	**49.91**	**53.14**	**89.82**	**85.87**	**76.27**
	45%	**58.83**	**47.89**	**42.93**	**62.74**	**53.09**	**54.92**	**97.51**	**90.52**	**80.98**
	90%	**61.68**	**49.19**	**45.62**	**64.92**	**55.35**	**55.30**	**98.79**	**95.26**	**83.11**
FedAvg	22.5%	56.59	44.00	37.39	56.98	48.74	52.53	89.02	80.79	74.86
	45%	59.16	47.12	40.11	61.15	51.88	53.89	97.12	86.45	79.15
	90%	62.60	49.80	43.06	64.83	55.07	55.42	97.92	89.05	80.79
FedAvg +DI	22.5%	**57.33**	**45.00**	**38.19**	**59.53**	**50.01**	**53.23**	**91.09**	**85.74**	**76.68**
	45%	**60.58**	**52.55**	**41.22**	**63.24**	**54.39**	**55.28**	**98.19**	**91.36**	**81.61**
	90%	**63.30**	**51.23**	**44.45**	**66.52**	**56.37**	**57.03**	**99.20**	**94.80**	**83.67**
RSC	22.5%	56.40	43.13	37.18	56.86	48.39	52.24	88.68	82.64	74.52
	45%	58.53	46.58	40.19	60.95	51.56	54.18	96.79	85.11	78.49
	90%	62.30	49.73	44.81	63.77	55.15	55.94	98.26	91.28	81.82
RSC +DI	22.5%	**57.15**	**45.02**	**37.37**	**59.99**	**49.88**	**52.86**	**91.77**	**86.07**	**76.90**
	45%	**59.76**	**49.70**	**40.48**	**63.00**	**53.23**	**55.21**	**97.45**	**93.04**	**81.90**
	90%	**62.97**	**50.62**	**45.84**	**66.61**	**56.51**	**56.39**	**99.20**	**96.40**	**83.99**
Scaffold	22.5%	56.31	44.74	37.54	57.57	49.04	52.93	90.23	86.58	76.58
	45%	58.74	46.90	39.54	61.03	51.55	54.46	97.79	90.43	80.89
	90%	61.94	49.07	43.91	64.20	54.78	54.38	98.33	95.21	82.64
Scaffold +DI	22.5%	**57.16**	**45.34**	**37.95**	**59.59**	**50.01**	**54.25**	**91.09**	**87.30**	**77.54**
	45%	**60.13**	**48.84**	**41.05**	**63.50**	**53.15**	**54.81**	**98.19**	**92.83**	**81.94**
	90%	**62.74**	**50.72**	**44.18**	**66.94**	**56.14**	**55.84**	**99.13**	**96.39**	**83.78**

Table 1: Accuracy comparison of the OfficeHome and Office-31. The accuracy is the average accuracy obtained by performing domain-by-domain exclusion test. "+DI" indicates the use of method DI. "Percent" is the portion of training set in the whole dataset.



(a) Product (b) Art (c) Clipart (d) Real

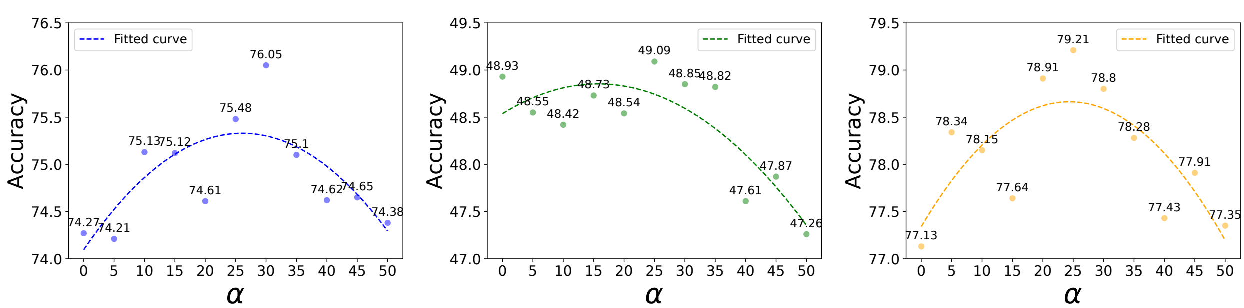
Figure 3: Generalisation performance comparison among AM, GA, and DI. The variance σ_1, σ_2, and σ_3 are the variance of model accuracy when 22.5%, 44.5%, and 90% training data are used, respectively. The scatter represents the variance of the model accuracy over each domain (including source domains and target domain).

observe the performance change of our method when data volume decreases. Specifically, the validation set contains 10% of source domain data, and training set contains 90%, 45%, and 22.5% of source domain data, respectively.

Experiment results are shown in Table 1 and Table 2. The experiment results indicate that the combination of DI with different algorithms achieves consistent and significant improvements in generalization performance. Specifically, DI has the largest accuracy improvement 2.27% on AM, and DI has the smallest accuracy improvement 1.39% on Scaffold. There are also 2.15% and 1.9% accuracy improvement on FedAvg and RSC, respectively. DI has a significant improvement in all experiments, which indicates that DI is able to improve the generalization performance of models.

As for robustness, DI has considerable accuracy improvements in experiments on average when the data volume is low. Specifically, DI improves accuracy by 2.14% in AM when 22.5% of data are used. DI has the smallest accuracy improvement 1.89% on Scaffold while it has the maximum accuracy improvement 2.67% on FedAvg. DI also improves accuracy improvement by 2.06% on RSC. According to these experiment results, even with a limited amount of training data, DI still can obtain appropriate weights to enhance the generalization performance of models, which indicates that DI has strong robustness.

To further explore the ability of DI to improve the generalisation performance of models, we perform more experiments. In federated learning, high generalization performance means that model accuracy is high in all domains including training domains and unseen target domains. Thus, the variance of model accuracy on all domains should be small if model generalization is high. We run experiment three times on OfficeHome to compare variances of model accuracy of DI, GA, and AM. The variances of model accuracy on all domains are shown on Figure 3. It can be seen that DI can decrease the variances of model accuracy on all domains considerably, which indicates that DI can improve



(a) Office-31 (b) OfficeHome (c) PACS

Figure 4: Ablation experiment of hyperparameter α in DI. The basic algorithm for the experiment is AM, and the amount of experimental data is uniformly set to 22.5%.

Method	Percent	PACS				
		P	A	C	S	Avg
AM	22.5%	91.10	77.36	71.31	72.92	78.17
	45%	92.29	82.71	73.17	75.08	80.81
	90%	93.31	83.15	77.68	81.78	83.98
AM +DI	22.5%	**93.11**	**80.78**	**72.61**	**74.71**	**80.30**
	45%	**93.99**	**82.99**	**74.67**	**77.81**	**82.36**
	90%	**94.51**	**84.08**	**78.70**	**82.88**	**85.04**
FedAvg	22.5%	88.28	68.23	71.56	65.42	73.37
	45%	91.24	76.87	72.37	73.88	78.59
	90%	92.71	78.24	77.80	81.13	82.47
FedAvg +DI	22.5%	**94.05**	**73.55**	**72.11**	**68.71**	**77.10**
	45%	**92.67**	**79.77**	**73.63**	**76.01**	**80.52**
	90%	**95.43**	**79.20**	**78.13**	**81.49**	**83.56**
RSC	22.5%	90.54	68.11	**72.54**	67.91	74.77
	45%	91.06	77.80	75.11	74.78	79.68
	90%	92.73	80.08	77.85	81.49	83.03
RSC +DI	22.5%	**92.93**	**74.20**	72.13	**68.55**	**76.95**
	45%	**92.95**	**79.47**	**76.88**	**76.08**	**81.34**
	90%	**94.93**	**81.54**	**77.85**	**83.22**	**84.38**
Scaffold	22.5%	89.70	68.78	72.49	65.41	74.09
	45%	91.06	77.19	**76.38**	77.26	80.47
	90%	92.51	79.28	77.84	**82.23**	82.96
Scaffold +DI	22.5%	**92.52**	**73.73**	**72.92**	**68.41**	**76.89**
	45%	**92.24**	**80.37**	75.13	**78.16**	**81.47**
	90%	**94.23**	**80.59**	**78.10**	81.95	**83.93**

Table 2: Accuracy comparison of PACS. The accuracy is the average accuracy obtained by performing domain-by-domain exclusion test. "+DI" indicates the use of method DI. "Percent" is the portion of training set in the whole dataset.

the generalization performance significantly.

In Table 3, we compare the accuracy improvement of DI with the accuracy improvement of GA. The comparison results indicate that DI obtains better accuracy improvement than GA on all datasets. In particular, the greatest accuracy improvement of GA is 1.81% while the greatest accuracy improvement of DI is 3.31%. That is, the accuracy improvement of DI is 1.5% higher than that of GA in the best case. The least accuracy improvement of GA is 0.26% while the least accuracy improvement of DI is 1.05%. That is, the accuracy improvement of DI is 0.79% higher than that of GA in the worst case. According to the comparison results, the model generalization is improved more by DI (namely, adjusting weights of local models through data influences on global model update) than by GA (adjusting weights of local models through fairness).

Ablation Experiments

Ablation Study of the Hyperparameter α. In this experiment, we explore the impact of hyperparameter α on the model accuracy. The used algorithm in the experiment is AM, and the amount of experiment data is uniformly set to 22.5%. We plot the model accuracy in Figure 4 and fit a curve to demonstrate the change trend of model accuracy. The experiment results show a clear trend: as α increases, the model accuracy increases first and then decreases over all datasets. The trend indicates that hyperparameter α is a necessary and effective parameter which can maximize model accuracy.

Method	PACS	Office-31	OfficeHome
AM +GA	+0.28	+1.81	+0.44
AM +DI	**+1.58**	**+3.31**	**+1.93**
FedAvg +GA	+0.94	+1.14	+0.17
FedAvg +DI	**+2.25**	**+2.63**	**+1.63**
RSC +GA	+0.89	+1.00	+0.49
RSC +DI	**+1.74**	**+2.58**	**+1.50**
Scaffold +GA	+0.43	+0.26	+0.56
Scaffold +DI	**+1.57**	**+1.05**	**+1.38**

Table 3: Accuracy improvement comparison between DI and GA. The accuracy improvement is the average of three experiments. "+DI" indicates the use of method DI. "+GA" indicates the use of method GA.

Algorithm	DIC	DIA	Acc
FedAvg	✗	✗	**77.76**
FedAvg	✗	✓	**78.24**
FedAvg	✓	✗	**79.24**
FedAvg	✓	✓	**80.16**

Table 4: Ablation experiment of major components in DI.

Components Ablation Experiments. We conduct an ablation experiment to assess the impact of two important components of DI: DIC (Data Influences Calculator) and DIA (Data Influences Adjustment). The experiments are performed on PACS and the basic method of experiments is FedAvg. The experiment results are shown in Table 4. The experiment results indicate that each component can improve the generalization performance of models alone and the combination of the two components can improve the generalization performance more. Specifically, DIA increases model accuracy from 77.76% to 78.24% alone. DIC increases model accuracy by 1.48% from 77.76% to 79.24% alone. The combination between DIA and DIC suprisingly increases model accuracy by 2.4%. The accuracy improvements demonstrate that the components of DI are necessary and effective.

Conclusions

In this paper, we propose the method DI to improve federated domain generalization. DI has two components DIC and DIA. Specifically, DIC calculates weights of local models through influences of data in source domains on the global model update and DIA adjusts the weights calculated by DIC in case these weights are overly dependent on a small portion of the data. Comprehensive comparison experiments and ablation experiments show that our method can improve federated domain generalization significantly.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful comments. Wen Huang is corresponding author of this paper. This work is supported in part by China Postdoctoral Science Foundation, PR China (Grant. 2024M752210), MOE (Ministry of Education in China) Liberal arts and Social Sciences Foundation (Grant. 24XJCZH004), Key R&D Program of Sichuan Province, China (2023YFG0115, 2023YFG0112), basic research projects of The Sichuan Provincial Industrial Development Fund: "Container Cloud Computing Infrastructure Platform", "Next-generation Big Data Visualization Based Decision-making and Analysis Platform" (Grant. 2023JB06), Cooperative Program of Sichuan University and Zigong, PR China (Grant. 2022CDZG-6).

References

Blanchard, G.; Lee, G.; and Scott, C. 2011. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample. In Shawe-Taylor, J.; Zemel, R.; Bartlett, P.; Pereira, F.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Chen, C.; Feng, X.; Zhou, J.; Yin, J.; and Zheng, X. 2023. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*.

Chen, J.; Li, J.; Huang, R.; Yue, K.; Chen, Z.; and Li, W. 2022. Federated Transfer Learning for Bearing Fault Diagnosis With Discrepancy-Based Weighted Federated Averaging. *IEEE Transactions on Instrumentation and Measurement*, 71: 1–11.

Cook, R. D.; and Weisberg, S. 1980. Characterizations of an Empirical Influence Function for Detecting Influential Cases in Regression. *Technometrics*, 22(4): 495–508.

Fan, Z.; Wang, Y.; Yao, J.; Lyu, L.; Zhang, Y.; and Tian, Q. 2022. FedSkip: Combatting Statistical Heterogeneity with Federated Skip Aggregation. In *2022 IEEE International Conference on Data Mining (ICDM)*, 131–140.

Huang, Z.; Wang, H.; Xing, E. P.; and Huang, D. 2020. Self-challenging Improves Cross-Domain Generalization. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 124–140. Cham: Springer International Publishing. ISBN 978-3-030-58536-5.

Jiang, M.; Wang, Z.; and Dou, Q. 2022. HarmoFL: Harmonizing Local and Global Drifts in Federated Learning on Heterogeneous Medical Images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1): 1087–1095.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2): 1–210.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 5132–5143. PMLR.

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1885–1894. PMLR.

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 (ICCV)*.

Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; and He, B. 2023. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *IEEE Transactions on Knowledge and Data Engineering*, 35(4): 3347–3366.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020a. Federated Optimization in Heterogeneous Networks. In Dhillon, I.; Papailiopoulos, D.; and Sze, V., eds., *Proceedings of Machine Learning and Systems*, volume 2, 429–450.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020b. Federated Optimization in Heterogeneous Networks. In Dhillon, I.; Papailiopoulos, D.; and Sze, V., eds., *Proceedings of Machine Learning and Systems*, volume 2, 429–450.

Lim, W. Y. B.; Luong, N. C.; Hoang, D. T.; Jiao, Y.; Liang, Y.-C.; Yang, Q.; Niyato, D.; and Miao, C. 2020. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Communications Surveys Tutorials*, 22(3): 2031–2063.

Ling, R. F. 1984. Residuals and Influence in Regression. *Technometrics*, 26(4): 413–415.

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 (CVPR)*, 1013–1023.

Lyu, L.; Yu, H.; and Yang, Q. 2020. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, J., eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 1273–1282. PMLR.

McMahan, H. B.; Moore, E.; Ramage, D.; and y Arcas, B. A. 2016. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2(2).

Muandet, K.; Balduzzi, D.; and Schölkopf, B. 2013. Domain Generalization via Invariant Feature Representation. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 10–18. Atlanta, Georgia, USA: PMLR.

Nagalapatti, L.; and Narayanam, R. 2021. Game of Gradients: Mitigating Irrelevant Clients in Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10): 9046–9054.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting Visual Category Models to New Domains. In Daniilidis, K.; Maragos, P.; and Paragios, N., eds., *Computer Vision – ECCV 2010*, 213–226. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-15561-1.

Seunghan, Y.; Seokeon, C.; Hyunsin, P.; Sungha, C.; and Yun, S. 2024. Client-agnostic learning and zero-shot adaptation for federated domain generalization. US Patent App. 18/238,998.

Shenaj, D.; Fanì, E.; Toldo, M.; Caldarola, D.; Tavera, A.; Michieli, U.; Ciccone, M.; Zanuttigh, P.; and Caputo, B. 2023. Learning Across Domains and Devices: Style-Driven Source-Free Domain Adaptation in Clustered Federated Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 444–454.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR 2011*, 1521–1528.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xiao, X.; Tang, Z.; Xiao, B.; et al. 2023. A survey on privacy and security issues in federated learning. *Chin. J. Comput*, 46(5): 1019–1044.

Xu, Q.; Zhang, R.; Zhang, Y.; Wang, Y.; and Tian, Q. 2021. A Fourier-Based Framework for Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 14383–14392.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2): 1–19.

Yuan, J.; Ma, X.; Chen, D.; Wu, F.; Lin, L.; and Kuang, K. 2023. Collaborative Semantic Aggregation and Calibration for Federated Domain Generalization. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12528–12541.

Zhang, R.; Xu, Q.; Yao, J.; Zhang, Y.; Tian, Q.; and Wang, Y. 2023. Federated Domain Generalization With Generalization Adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3954–3963.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated Learning with Non-IID Data. *CoRR*, abs/1806.00582.

Zhuang, W.; Gan, X.; Zhang, X.; Wen, Y.; Zhang, S.; and Yi, S. 2022. Federated Unsupervised Domain Adaptation for Face Recognition. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6.