Supplementary Material for Federated Domain Generalization with Generalization Adjustment

Ruipeng Zhang[1,2], Qinwei Xu[1,2], Jiangchao Yao[1,2], Ya Zhang[1,2,✉], Qi Tian[3], Yanfeng Wang[1,2,✉]
[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
[2]Shanghai AI Laboratory [3]Huawei Cloud & AI
{zhangrp, qinweixu, Sunarker, ya_zhang, wangyanfeng}@sjtu.edu.cn, tian.qi1@huawei.com

A. Proof of Theorems

A.1. Technical Lemmas

Lemma 1. *If we have $\mathcal{E}_{\widehat{D}}(\theta) = \sum_{i=1}^{M} a_i \mathcal{E}_{\widehat{D}_i}(\theta)$, then for any domain T, we have:*

$$d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}, T) = \sum_{i=1}^{M} a_i d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \tag{1}$$

Proof. From the definition of $d_{\mathcal{H}\Delta\mathcal{H}}(\cdot,\cdot)$ in [1], we can get

$$\begin{aligned}
d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}, T) =& 2 \sup_{A \in \mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} |\mathrm{Pr}_{\widehat{D}}(A) - \mathrm{Pr}_T(A)| \\
=& 2 \sup_{A \in \mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left| \sum_{i=1}^{M} a_i \mathrm{Pr}_{\widehat{D}_i}(A) - \mathrm{Pr}_T(A) \right| \\
=& 2 \sup_{A \in \mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \left| \sum_{i=1}^{M} a_i (\mathrm{Pr}_{\widehat{D}_i}(A) - \mathrm{Pr}_T(A)) \right| \\
\leq& 2 \sup_{A \in \mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} \sum_{i=1}^{M} a_i |\mathrm{Pr}_{\widehat{D}_i}(A) - \mathrm{Pr}_T(A)| \\
\leq& 2 \sum_{i=1}^{M} a_i \sup_{A \in \mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} |\mathrm{Pr}_{\widehat{D}_i}(A) - \mathrm{Pr}_T(A)| \\
=& \sum_{i=1}^{M} a_i d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T).
\end{aligned}$$

□

Lemma 2. *For any $\theta \in \Theta$, the expectation risk gap between domain A and domain B is bounded by the domain divergence $d_{\mathcal{H}\Delta\mathcal{H}}(A, B)$.*

$$|\mathcal{E}_A(\theta) - \mathcal{E}_B(\theta)| \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(A, B). \tag{2}$$

Proof. By the definition of $d_{\mathcal{H}\Delta\mathcal{H}}(\cdot,\cdot)$ in [1], we have

$$\begin{aligned}
d_{\mathcal{H}\Delta\mathcal{H}}(A, B) =& 2 \sup_{\theta, \theta' \in \Theta} |\mathrm{Pr}_{x\sim A}[f(x;\theta) \neq f(x;\theta')] \\
&- \mathrm{Pr}_{x\sim B}[f(x;\theta) \neq f(x;\theta')]|
\end{aligned}$$

where $f(x;\theta)$ means the prediction function on data x with model parameter θ. We choose θ' as parameter of the label function, then $f(x;\theta) \neq f(x;\theta')$ means the loss function $\mathcal{L}(x;\theta)$, so we have

$$\begin{aligned}
d_{\mathcal{H}\Delta\mathcal{H}}(A, B) \geq& 2 \sup_{\theta \in \Theta} |\mathrm{Pr}_{x\sim A}[\mathcal{L}(x;\theta)] - \mathrm{Pr}_{x\sim B}[\mathcal{L}(x;\theta)]| \\
\geq& 2|\mathcal{E}_A(\theta) - \mathcal{E}_B(\theta)|
\end{aligned}$$

□

Lemma 3. *Let $\mathcal{H}$ be the hypothesis space and Θ is the corresponding parameter space, the VC dimension of $\mathcal{H}$ is d. The domain divergence between two domains D_i and D_j on hypothesis space $\mathcal{H}$ is denoted by $d_{\mathcal{H}\Delta\mathcal{H}}(D_i, D_j)$. Then for any $\delta \in (0, 1)$, with probability at least $1-\delta$, $\forall \theta \in \Theta$:*

$$\begin{aligned}
\mathcal{E}_T(\theta) \leq& \sum_{i=1}^{M} a_i \Bigg(\widehat{\mathcal{E}}_{\widehat{D}_i}(\theta) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \\
&+ \sqrt{\frac{\log d + \log 1/\delta}{2N_i}} \Bigg) + \lambda,
\end{aligned} \tag{3}$$

where λ is the optimal combined risk on T and $\widehat{D}$ that can be achieved by the parameters in Θ.

Proof. From the Theorem 2 in [1], we ignore the estimation error from $d_{\mathcal{H}\Delta\mathcal{H}}$ and $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$, and have the generalization bound for domain T and $\widehat{D}$ with the probability at least $1-\delta$:

$$\mathcal{E}_T(\theta) \leq \mathcal{E}_{\widehat{D}}(\theta) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}, T) + \lambda. \tag{4}$$

And we have $\mathcal{E}_{\widehat{D}}(\theta) = \sum_{i=1}^{M} a_i \mathcal{E}_{\widehat{D}_i}(\theta)$ and Lemma 1, then Eq. (4) can be rewritten as the following inequality.

$$\begin{aligned}
\mathcal{E}_T(\theta) &\leq \sum_{i=1}^{M} a_i \mathcal{E}_{\widehat{D}_i}(\theta) + \frac{1}{2}\sum_{i=1}^{M} a_i d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) + \lambda. \\
&\leq \sum_{i=1}^{M} a_i \Big(\widehat{\mathcal{E}}_{\widehat{D}_i}(\theta) + \sqrt{\frac{\log d + \log 1/\delta}{2N_i}} \\
&\quad + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \Big) + \lambda.
\end{aligned}$$

The second inequality considers the generalization bound between $\mathcal{E}_{\widehat{D}_i}(\theta)$ and $\widehat{\mathcal{E}}_{\widehat{D}_i}(\theta)$ on each domain.

□

A.2. Proof of Theorem 1

Theorem 1. *Let θ denote the global model after R round federated learning, θ_i^* and θ_T^* mean the local optimal for each source domain and the unseen target domain, respectively. For any $\delta \in (0, 1)$, the domain generalization gap for the unseen domain T can be bounded by the following equation with a probability of at least $1 - \delta$.*

$$\begin{aligned}
\mathcal{E}_T(\theta) - \mathcal{E}_T(\theta_T^*) \leq \sum_{i=1}^{M} a_i \Bigg(& G_{\widehat{D}_i}(\theta) + d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \\
& + \frac{\sqrt{\log \frac{d}{\delta}} + \sqrt{\log \frac{Md}{\delta}}}{\sqrt{2N_i}} \Bigg) + \lambda
\end{aligned} \tag{5}$$

Proof. For a given $\theta \in \Theta$, with the definition of generalization bound, the following inequality holds with at most $\frac{\delta}{M}$ for each domain $\widehat{D}_i$. (M is the number of domains)

$$\widehat{\mathcal{E}}_{\widehat{D}_i}(\theta) - \mathcal{E}_{\widehat{D}_i}(\theta) > \sqrt{\frac{\log d + \log M/\delta}{2N_i}} \tag{6}$$

Moreover, from Lemma 2, we have $\mathcal{E}_{\widehat{D}_i}(\theta) - \mathcal{E}_T(\theta) \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T)$ for each domain. Then let us consider Eq. (6), we can obtain the following inequalities with the probability at least greater that $1 - \frac{\delta}{M}$.

$$\begin{aligned}
\min_{\theta'} \widehat{\mathcal{E}}_{\widehat{D}_i}(\theta') &\leq \widehat{\mathcal{E}}_{\widehat{D}_i}(\theta) \leq \mathcal{E}_{\widehat{D}_i}(\theta) + \sqrt{\frac{\log d + \log M/\delta}{2N_i}} \\
&\leq \mathcal{E}_T(\theta) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) + \sqrt{\frac{\log d + \log M/\delta}{2N_i}}
\end{aligned}$$

We denote the local optimal on each source domain i as θ_i^*. If we choose a specific parameter $\theta_T^* = \min_\theta \mathcal{E}_t(\theta)$ which is the local optimal on the unseen domain T, the above third inequality still holds. Then we can rewrite the above inequalities into:

$$\widehat{\mathcal{E}}_{\widehat{D}_i}(\theta_i^*) \leq \mathcal{E}_T(\theta_T^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) + \sqrt{\frac{\log d + \log M/\delta}{2N_i}} \tag{7}$$

Considering on each domain, Eq. (7) holds. By a similar derivation process, we can obtain the inequality between T and $\widehat{D}$ with the probability at least greater that $1 - \delta$.

$$\begin{aligned}
\sum_{i=1}^{M} a_i \widehat{\mathcal{E}}_{\widehat{D}_i}(\theta_i^*) \leq \mathcal{E}_T(\theta_T^*) + \sum_{i=1}^{M} a_i \Bigg(& \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \\
& + \sqrt{\frac{\log d + \log M/\delta}{2N_i}} \Bigg)
\end{aligned} \tag{8}$$

Combining the Eq.(8) and Lemma 3, we have Theorem 1 with the global model θ after federated learning.

$$\begin{aligned}
\mathcal{E}_T(\theta) - \mathcal{E}_T(\theta_T^*) &\leq \sum_{i=1}^{M} a_i \Bigg(\widehat{\mathcal{E}}_{\widehat{D}_i}(\theta) - \widehat{\mathcal{E}}_{\widehat{D}_i}(\theta_i^*) + d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \\
&\quad + \frac{\sqrt{\log \frac{d}{\delta}} + \sqrt{\log \frac{Md}{\delta}}}{\sqrt{2N_i}} \Bigg) + \lambda \\
&= \sum_{i=1}^{M} a_i \Bigg(G_{\widehat{D}_i}(\theta) + d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{D}_i, T) \\
&\quad + \frac{\sqrt{\log \frac{d}{\delta}} + \sqrt{\log \frac{Md}{\delta}}}{\sqrt{2N_i}} \Bigg) + \lambda
\end{aligned} \tag{9}$$

□

B. Main Code of GA

We release the pytorch-style pseudo code of our GA method based on FedAvg, and other methods can also be applied by simply replace the function "`client_train`" for local training algorithms, "`client_eval`" for client evaluation and "`FedAvg`" for federated aggregation algorithms by their own. Our codes have been uploaded within the supplementary materials and will be released publicly after the paper is accepted.

```
def GA(gen_gaps, domain_weights, d):
    '''
    adjust the domain weights by the
        generalization gaps and step
        size d
    '''
    mean_gap = mean(gen_gaps)
    gen_gaps = gen_gaps - mean_gap

    # adjust the weights by the gaps and
        step size
```

```
    for i in range(M):
        new_domain_weights[i] =
            domain_weights[i] +
            gen_gaps[i] / max(gen_gaps) *
            d

    # normalize the new domain weights
    for i in range(M):
        new_domain_weights[i] /=
            sum(new_domain_weights)
    return new_domain_weights

def main():
    # initialize the datasets for each
        source domain
    datasets = get_data()

    # initialize the parameters of
        global model
    global_model = get_model()

    # initialize the local models
    local_models = [get_model() for i in
        range(M)]
    broadcast(global_model, local_models)

    # initialize the domain weights
    domain_weights = [1/M for i in
        range(M)]

    # federated learning
    for r in range(R):
        for i in range(M): # client
            # evaluate on global model
            loss_global[r][i] =
                client_eval(global_model,
                datasets[i]['val'])

            # generalization gap on global
                model theta_r
            gen_gaps[r][i] =
                loss_global[r][i] -
                loss_local[r-1][i]

            # local training on theta_i_r
            local_models[i] =
                client_train(local_models[i],
                datasets[i]['train'],
                local_epochs, t)
            loss_local[r][i] =
                client_eval(local_models[i],
                datasets[i]['val'])

        # Generalization Adjustment
        domain_weights = GA(gen_gaps[r],
            domain_weights, d*(R-r)/R)

        # parameter aggregation
        global_model =
            FedAvg(local_models,
            domain_weights)

        broadcast(global_model,
            local_models)

if __name__ == '__main__':
    main()
```

C. More Experimental Results

C.1. Compared with more FedDG methods under several settings

FedDG is a cross-silo FL problem that each client contains a large scale of data with unique data distribution, and it aims to solve the out-of-domain generalization problem in FL. We follow the FedDG setting from ELCFS [5] that each client corresponds to one domain, which is also the same as FedSR [6], and CCST [4]. And our GA is contemporaneous with FedSR [6], FedASAM+SWA [2] (namely FedASAM* in Table 1), and CCST [4], which are published and open-sourced after the submission deadline of CVPR2023. Therefore, we add comparisons with these advanced SOTA FedDG methods in Table 1. GA can still improve the performance on top of them. However, we do appreciate the suggestion of reviewers that one domain can correspond to multiple clients, and implement such experiments in Table 1. In experiments, each domain of data is partitioned into 10 clients, and we randomly select 10 clients to participate in the training per round. From the results, we can find that our GA can still provide gain for the large-scale FL.

Table 1. Results with more clients & with more advanced SOTAs.

Dataset	*more clients*		*suggested SOTAs*			
	FedAvg	Best[1]	ELCFS	FedSR	FedASAM*	CCST
PACS	80.33	81.62	84.07	83.70	82.04	83.48
with GA	**81.99**	**82.72**	**84.88**	**84.66**	**83.57**	**84.35**
OfficeHome	63.38	64.08	62.88	64.29	64.32	64.25
with GA	**64.40**	**65.04**	**64.60**	**64.65**	**64.80**	**65.42**

C.2. Comparison with TTDA.

The application scenarios of Federated Domain Generalization (FedDG) and test time domain adaptation (TTDA) are similar, which both focus on the performance on the unseen target clients with domain shifts. Generally speaking, despite similarity, FedDG and TTDA are at different stages. FedDG aims to improve the out-of-domain generalization during the training of global model, while TTDA aims to

[1] Best performance of other baselines in the Table 1 of submitted manuscript to save space (ELCFS for PACS and HarmoFL for OfficeHome).

better adapt the trained global modal to the new client with domain shift. FedDG and TTDA complement each other, and the more generalized global model from FedDG has better adaptive effects on TTDA. Given the orthogonality of FedDG and TTDA, we can apply GA on all TTDA methods. In Table 2, we implement two well-known TTDA methods: Domain Specific Batch Normalization (DSBN) [3] and Test-time adaptation by entropy minimization (Tent) [7]. From Table 2, we can see GA can also improve the performance with TTDA.

Table 2. Combination with two TTDA methods.

Method	PACS					OfficeHome				
	P	A	C	S	Avg,	P	A	C	R	Avg.
DSBN	96.26	82.23	80.99	77.50	84.25	73.34	56.49	53.64	73.03	64.13
+*GA*	96.56	83.18	81.21	80.11	**85.26**	72.91	57.50	54.99	73.85	**64.81**
Tent	96.92	85.94	83.06	91.39	86.83	74.63	57.95	56.48	74.67	65.92
+*GA*	97.16	86.77	83.98	83.28	**87.80**	74.53	59.79	56.61	75.36	**66.57**

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 1

[2] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIII*, pages 654–672. Springer, 2022. 3

[3] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, pages 7354–7362, 2019. 4

[4] Junming Chen, Meirui Jiang, Qi Dou, and Qifeng Chen. Federated domain generalization for image recognition via cross-client style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 361–370, 2023. 3

[5] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. *CVPR*, 2021. 3

[6] A. Tuan Nguyen, Philip Torr, and Ser-Nam Lim. Fedsr: A simple and effective domain generalization method for federated learning. *NeurIPS 36 (NeurIPS)*, 2022. 3

[7] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 4