Privacy-Preserving Constrained Domain Generalization via Gradient Alignment

Chris Xing Tian, Haoliang Li, Yufei Wang and Shiqi Wang

Abstract—Deep neural networks (DNN) have demonstrated unprecedented success for various applications. However, due to the issue of limited dataset availability and the strict legal and ethical requirements for data privacy protection, the broad applications of DNN (e.g., medical imaging classification) with large-scale training data have been largely hindered, greatly constraining the model generalization capability. In this paper, we aim to tackle this problem by developing the privacy-preserving constrained domain generalization method, aiming to improve the generalization capability under the privacy-preserving condition. In particular, we propose to improve the information aggregation process on the centralized server side with a novel gradient alignment loss, expecting that the trained model can be better generalized to the "unseen" but related data. The rationale and effectiveness of our proposed method can be explained by connecting our proposed method with the Maximum Mean Discrepancy (MMD) which has been widely adopted as the distribution distance measure. Experimental results on three domain generalization benchmark datasets indicate that our method can achieve better cross-domain generalization capability compared to the state-of-the-art federated learning methods.

Index Terms—Federated learning, domain generalization, gradient alignment.

1 Introduction

Deep neural networks (DNNs) have achieved remarkable success in various applications, such as computer vision, natural language processing, and acoustic verification. For example, in the field of medical imaging classification (e.g., tumour detection and classification, X-ray image analysis), DNN model can even achieve higher diagnosis accuracy compared with human doctors.

The tremendous achievements of DNNs are driven by the availability of the large-scale training data. To guarantee reliable decision support based on artificial intelligence (AI) oriented applications, the large amount of data are indispensable for training purposes [1], [2]. However, reasonably large-scale dataset for some realistic environments (e.g., clinical environment [3], [4]) are infeasible to collect due to two reasons. First, though annotated data can be collected from multiple sources, aggregating the data for training may not be feasible due to the privacy regulations. For example, European General Data Protection Regulation (GDPR) has imposed strict rules regarding the storage and exchange of the health data. Second, even though data access permission can be obtained, collecting large-scale and representative data can still be difficult due to the variation of capturing protocols, device vendors and environments. Thus, the trained DNNs are typically prone to be lack of generalization capability when annotated large-scale data are not available during training stage, especially for the out-of-distribution data which are "unseen" during the training stage.

Tremendous efforts have been devoted to tackling the challenges of privacy and generalization for the DNN. Regarding the issue of privacy, federated learning (FL) [5], [6] was proposed to train DNN based on datasets distributed across multiple domains while preventing data leakage. However, while existing techniques (e.g., [7], [8]) can tackle the setting where data from multiple domains are heterogeneous (i.e., the distribution of data from different sources are different), the trained DNN may not be able to generalize well to the out-of-distribution data. Regarding the issue of generalization, numerous methods have been developed to improve the generalization capability of DNN based on domain generalization [9], [10]. However, the existing techniques require to aggregate the data to conduct domain-shift simulation, which disobeys the privacy regulation rules.

To jointly overcome the aforementioned difficulties, we propose a novel *task-agnostic* domain generalization method based on gradient aggregation, aiming to improve the model generalization capability under the privacy-preserving constraint (i.e., domain generalization under the constraint of privacy-preserving without data sharing from multiple domains). By treating the gradient as a kernel mean embedding from the original data space to the neural tangent kernel space, we conduct distribution alignment through Maximum Mean Discrepancy (MMD) [11] across multiple domains based on the gradient. As such, the new aggregated gradient is equipped with information from multiple domains and is expected to be better generalized to the "unseen" testing data. We conduct extensive experiments on three domain generalization benchmarks with the issue of privacy to evaluate our proposed method. The results show that our method can achieve better generalization capability compared with state-of-the-art FL and domain generalization (DG) techniques under the privacy-preserving setting.

2 Related Works

2.1 Domain Generalization

In the context of mitigating the challenge posed by disparate environmental conditions between training and test data, the concept of domain generalization (DG) has emerged as a promising approach. DG involves leveraging data collected from diverse environmental conditions (source domains) to train a deep neural network (DNN) model that can effectively handle testing data obtained from an unseen yet related condition (unknown target domain). It is important to note that DG shares a close relationship with domain adaptation (DA) [12], where domain shifts are also addressed. However, unlike DA, which assumes access to some (labeled or unlabeled) data samples from the target domain, DG assumes the unavailability of such samples during training. Consequently, DG methods must seek solutions to effectively exploit the information present in multiple source domains accessible during training. The hope is that by distilling shared knowledge from source domains, we can obtain more robust features that can be potentially useful in unseen target domains. Existing techniques in the DG field can be broadly categorized into three streams. The first stream is based on the idea of training a dedicated classifier for each source domain and then combining them to give fused predictions by evaluating the similarity between different source domains and test samples (e.g., [13], [14]). The second stream focuses on extracting shareable information across data through feature representation learning (e.g., [15]) and meta-learning (e.g., [16]). These approaches aim to discover common patterns or representations that are transferable across different domains, enabling the model to generalize well to unseen conditions. The last stream involves employing data augmentation techniques, such as domain randomization [17] and adversarial training [18], [19], to augment the scale of the training data. By introducing variations or perturbations to the data during training, these methods enhance the model's ability to handle diverse environmental conditions.

2.2 Federated Learning

Recent years have seen a rapidly growing number of intelligent devices with AI computing capability, such as smartphones, wearable devices, autonomous vehicles, intelligent CCTV cameras, IoT devices, etc. Those devices, forming a large distributed network, can generate a large amount of heterogeneous data every day. How to fully utilize the local AI computing capability of each device while reducing the data transmission cost or preserving data privacy becomes a new challenge. Traditional AI data processing models, which usually require homogeneous data transmission from some parties to a central party for model training and final build, can hardly be adapted in such scenarios. This gap leads to a growing interest in the Federated Learning (FL) framework. The Federated Learning is firstly proposed in [20], [6] to support training AI models over distributed remote devices or isolated data centers while keeping data localized. In a general Federate Learning setting, there may be tens to potentially millions of distributed clients (remote devices/soiled data islands, etc.), and each client trains the AI model locally using its private dataset. In each FL training round, the clients will share their model information, usually the learned model weights or gradients, instead of the training data, with a central aggregator. The aggregator would aggregate the information from those clients (e.g., through model parameters averaging) to obtain global model parameters, which will be sent back to clients for the next round of training.

One limitation of the aforementioned mechanism is that it does not fully address the underlying challenges associated with system and data heterogeneity, where system heterogeneity refers to the situation where each local server has different computational power, communication bandwidth, etc., which further leads to local-update variation, and data heterogeneity refers to the situation that the data distributions from different local servers are different. Moreover, it is highly likely that the testing data are drawn from the distribution which is different from the data distributions of local servers. Regarding the first issue, in [7], a proximal term on the objective function is introduced for each local server, such that the impact of local-update variation can be mitigated. Regarding the second issue, in [21], [22], a federated transfer learning scheme was proposed, where shareable information across servers can be learned with domain alignment regularization. However, it required that the co-occurrence samples are available between the labeled source domain (i.e., domain for training purpose) and the unlabeled target domain (i.e., domain for testing purpose), which is not desired as we do not have target domain data in hand for real-time applications. As such, its generalization capability to out-of-distribution samples is prohibited. Besides the aforementioned techniques, there are also some works focusing on the federated learning for multi-task setting or non i.i.d setting [23], [24], [25], [26], where the data are heterogeneous. However, they are not designed based on the cross-domain scenario for testing.

In recent years, there exist some works which focus on tackling the problem of domain generalization under the constraint of privacy-preserving (i.e., non-shared data from multiple domains) [27], [28]. In [27], the authors proposed to tackle the problem domain generalization under privacy setting for medical imaging, with the input based on frequency space (with 2D Fourier transformation) and a boundary-oriented meta-optimization strategy, which is task-specific (i.e., tailored for medical image segmentation task). While the method proposed in [28] is task-agnostic, it is built upon FedAvg [6] but requires both trainable classifier models and frozen models during the training stage, which may not be able to scale to different architectures. Our framework only requires conventional local training on clients, which aligns with the standard of federated learning and can be applied to different models and tasks. Moreover, unlike [27] and [28] which focus on local training, our proposed method focuses on aggregation on the central server side.

3 Proposed Method

We propose to study the problem of task-agnostic privacy-preserving constrained domain generalization (PPDG). The architecture is built upon the FL system configuration [5], [6], which is designed to handle data from multiple local servers (i.e., client servers) and then aggregate the information from local servers to a centralized server. Based on the FL settings, the centralized server maintains a global DNN model to coordinate the global learning objective across the framework. Specifically, the objective is to minimize

$$\min_{w} f(w) = \sum_{k=1}^{K} p_k F_k(w), \tag{1}$$

where $F_k(w)$ denotes the objective of deep learning model on the k-th local server, K is the number of local servers, $p_k > 0$, and $\sum_k p_k = 1$. In practice, one can set $p_k = n_k/n$, where n_k and n denote the number of training data in the kth server and the total number of training data, respectively. During training, at the federated round t, the DNN in the local server is updated by receiving the DNN parameters/gradients[1] from the centralized server, and the local servers further conduct DNN model training and send the encrypted gradient to the centralized server for gradient aggregation.

3.1 Distribution Alignment in Neural Tangent Kernel Space

Directly conducting gradient aggregation through averaging process [6] may not benefit the generalization capability of DNN model. One reason may be attributed to the gradient conflict (i.e., $\langle \frac{\partial F_i(w)}{\partial w}, \frac{\partial F_j(w)}{\partial w} \rangle < 0$ for local server i and j) [29] which further leads to negative transfer across different servers. To this end, we propose to improve the generalization capability of DNN with privacy-preserving constraints by proposing a novel gradient aggregation technique.

Before introducing our proposed method, we first revisit the problem of improving the generalization capability of machine learning model to the out-of-distribution samples, which has received more and more attention recently [15], [30]. In [30], the authors theoretically proved that the generalization capability can be improved by domain alignment through domain variance reducing, where the domain variance can be defined by summing the MMD distance between domain pairs, which can be given as

$$\sum_{i,j} \|\mu_{P_i} - \mu_{P_j}\|^2_{\mathcal{H}}, \tag{2}$$

where μ_{P_i} and μ_{P_j} denote the kernel embedding of distribution of domain i and j, respectively. One can minimize the domain variance by representing the kernel mean with empirical averaging as $\mu_P = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)$, where ϕ denotes a feature mapping function [11].

Our motivation originates from the recent advance of DNN analysis based on neural tangent kernel [31]. By conducting first-order Taylor expansion of network objective $f(w)$, we can reformulate the objective as

$$f(w) \approx f(w_0) + \bigtriangledown_w f(w_0)^\top (w - w_0). \tag{3}$$

By focusing on the parameter w, the above approximation can be interpreted as a linear model with respect to w, and the feature map $\phi(\cdot)$ is the gradient at the initialization w_0 given as $\phi(x) = \bigtriangledown_w f(x; w_0)$ w.r.t. the data x. Based on the neural tangent kernel space, where the feature map of the original input data x is defined as the corresponding gradient, we can interpret the average gradient as the kernel mean with empirical averaging, given by $\mu_P = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)$, where ϕ represents the feature mapping function based on the neural tangent kernel, which is the gradient. Thus, we can define the domain variance among multiple local servers based on the kernel embedding in the neural tangent kernel space by extending the MMD distance in Eq. (2) as

$$\sum_{i,j} \|grad_i - grad_j\|^2, \tag{4}$$

where $grad_i$ and $grad_j$ denote the gradient sent to the centralized server from local server i and j, respectively.

3.2 Gradient Aggregation

Algorithm 1 Proposed Gradient Aggregation Algorithm

```
input: Gradients sent from local servers: $G = \{grad_i\}$.
Initialize: $\hat{grad}_i = grad_i$, $G = \{\hat{grad}_i\}$
for $\hat{grad}_i \in G$ do
    for $\hat{grad}_j \in G \backslash \{\hat{grad}_i\}$ do
        if $\langle \hat{grad}_i, \hat{grad}_j \rangle < 0$ then
            $\hat{grad}_i = \hat{grad}_i - 2\lambda(\hat{grad}_i - \hat{grad}_j)$
                                                    ▷ Gradient alignment
        end if
    end for
end for
$grad_{agg} = \frac{1}{K}\sum_{i=1}^{K} \hat{grad}_i$     ▷ Aggregated gradient sent
back to local servers
```

The gradient aggregation on the centralized server side optimizes Eq. 4 instead of conducting gradient averaging to avoid possible gradient conflicting. To jointly achieve domain alignment among multiple local servers while preserving discriminative power of DNN learning, we propose to conduct gradient modification based on local server i through gradient descend w.r.t. local server j only if negative transfer between i and j occurs (i.e., $\langle \frac{\partial F_i(w)}{\partial w}, \frac{\partial F_j(w)}{\partial w} \rangle <$ 0), and the modified gradient of local server i w.r.t. local server j is given as

$$\hat{grad}_i = grad_i - 2\lambda(grad_i - grad_j), \tag{5}$$

where λ is the hyper-parameter for gradient descend. We repeat this process across the gradients collected from all local servers in a random order to obtain the respective gradient $\hat{grad}_i$ for local server i. We then conduct gradient averaging, which is for model update. Our proposed method is summarized in Algorithm 1.

It is worth noting that our proposed gradient aggregation does not conflict with the homomorphic encryption, which encrypts the data by preserving the structural transformation of original data [5], [32] and is adopted for gradient communication in the FL setting. For example, by considering homomorphic encryption, Eq. 5 can be reformulated as

$$\hat{E}(grad_i) = E(grad_i) \ominus E(2) \otimes E(\lambda) \otimes (E(grad_i) \ominus E(grad_j)),$$

where $\otimes$ and $\ominus$ denote the subtraction and multiplication operation in the encrypted space. After conducting gradient aggregation on the centralized server side, the modified and encrypted gradient can be sent back to the local servers for decryption and model updating.

Discussion. Conceptually, our proposed method is close to Meta-Learning Domain Generalization (MLDG) [16]. In

[1] We consider gradient in our manuscript.

[16], a first-order Meta-Learning approximation was proposed to reformulate the objective in [33] with a classification loss and a gradient similarity loss. Besides, there also exists methods which perform gradient alignment for the problem of domain generalization. For example, ArgSum and ArgRand [34] aim to explore the consensus of gradient information (i.e., by masking the gradient to 0 if there exists sign contradiction). While ArgSum and ArgRand are based on the setting where data from multiple domains could be shared, their method can be extended in our setting. However, these two methods only explore gradient sign information which may not be able to extract domain invariant knowledge across different clients. In [35] and [36], the authors proposed to maximize the gradient similarity, and minimize gradient variance across domains, which can also be treated as performing gradient alignment for domain generalization. However, [35] and [36] focus on the centralized setting (i.e., data from multiple domains are shared), where the gradient alignment loss is jointly optimized with classification loss in a gradient-of-gradient manner (see Algorithm 2 in [35]). Our proposed method is different from [35] and [36], where the classification loss (i.e., local training) and gradient matching (on the central server) are conducted in different places, as such, no gradient-of-gradient information can be obtained. As such, a novel optimization scheme (i.e., aggregation method) should be developed to tackle the problem of domain generalization with non-shared data from multiple domains.

Our formulation is also similar to projecting conflicting gradients (PCGrad) [37], which is designed for multi-task learning, at a high level. In [37], the cosine similarity of gradients between two tasks are evaluated. If the value of gradient similarity is negative, PCGrad proceeds to replace one gradient by projecting it onto the normal plane of another gradient. However, there are two limitations which prevents PCGrad from being applied to our setting, 1) it involves division process when computing cosine similarity of gradient, the training process may not be stable if the gradient vanishes (i.e., values of gradient close to zero) at any local servers, which leads to 0 divided by 0; 2) even if there is not gradient vanishing, it is still difficult for PCGrad to be applied in homomorphic encryption based federated learning setting due to the division operation involved [38]. Nevertheless, we also show in the experimental section that our proposed method can achieve better performance compared with PCGrad.

We are also aware that more recently, [26] proposed a FL method where clients transmit Jacobian matrices to improve model performance in the non-IID FL setting. While some desired performance was reported in [26], our proposed method is different compared with [26] on two folds: 1) we focus on cross-domain FL scenario, 2) we only require the client to tranmit gradient information to the central server, which can be easier to adapt to different FL architectures (based on the standard of federated learning [39]).

3.3 Convergence Analysis

In this section, we conduct convergence analysis of our proposed gradient aggregation method. For simplicity, we assume that two local servers are involved in the federated learning training process, where the gradient sent from two local servers are denoted as $grad_1$ and $grad_2$, respectively. We assume that we first conduct gradient modification on $grad_1$ followed by $grad_2$, if needed. We further denote $\hat{grad}_1$ and $\hat{grad}_2$ as the modified gradients, respectively. We focus on the convergence analysis on local server 1 as a showcase.

At each update, we have three cases:

1) $\langle grad_1, grad_2 \rangle > 0$
2) $\langle grad_1, grad_2 \rangle < 0$ and $\langle \hat{grad}_1, grad_2 \rangle > 0$
3) $\langle grad_1, grad_2 \rangle < 0$ and $\langle \hat{grad}_1, grad_2 \rangle < 0$

For case 1), there is no need to conduct gradient modification based on our setting, for case 2), we only modify $grad_1$ by keeping $grad_2$ unchanged, for case 3), we modify both $grad_1$ and $grad_2$. Now we are ready to present our analysis.

Theorem 1. *We assume the loss function $\mathcal{L}$ is convex and differentiable, and the gradient of $\mathcal{L}$ is L-Lipschitz with $L > 0$. Then, the model update rule with our proposed gradient modification method will converge to the optimal value.*

Proof. If case 1), we can apply gradient descent (e.g., stochastic gradient descent) which leads to a standard deep neural network optimization.

If case 2), $grad_1$ will be modified as

$$\hat{grad}_1 = (1 - 2\lambda)grad_1 + 2\lambda grad_2, \tag{6}$$

and $grad_2$ will keep unchanged. the model parameters will then be updated as

$$w^* = w - \frac{\eta}{2}[(1 - 2\lambda)grad_1 + (1 + 2\lambda)grad_2]. \tag{7}$$

As we have assumed that $\mathcal{L}$ is Lipschitz continuous, by further denoting $t = \frac{\eta}{2}$, where $t \leq \frac{1}{L}$ based on the Lipschitz continuous property, we can conduct a quadratic expansion of $\mathcal{L}$ around $\mathcal{L}(w)$ and obtain the following inequality:

$$\begin{aligned}
&\mathcal{L}(w^*) \leq \mathcal{L}(w) + \triangledown\mathcal{L}(w)^\top(w^* - w) + \frac{1}{2}L\|w^* - w\|^2 \\
&\leq \mathcal{L}(w) + grad_1(-t[(1 - 2\lambda)grad_1 + (1 + 2\lambda)grad_2]) \\
&+ \frac{1}{2}L\|(-t[(1 - 2\lambda)grad_1 + (1 + 2\lambda)grad_2])\|^2 \\
&\leq \mathcal{L}(w) + (2\lambda^2 - \frac{1}{2})t\|grad_1 - grad_2\|^2.
\end{aligned} \tag{8}$$

As we can see, since $t > 0$, as long as $2\lambda^2 - \frac{1}{2} < 0$, we can have $\mathcal{L}(w^*) < \mathcal{L}(w)$ which implies that the objective function value strictly decreases with each iteration unless $grad_1 = grad_2$.

If case 3), $grad_1$ and $grad_2$ will be modified as

$$\hat{grad}_1 \quad = \quad (1 - 2\lambda)grad_1 + 2\lambda grad_2, \tag{9}$$

$$\hat{grad}_2 \quad = \quad 2\lambda \hat{grad}_1 + (1 - 2\lambda)grad_2, \tag{10}$$

respectively. Similar to the case 2), we can perform a quadratic expansion around $\mathcal{L}(w)$ as

$$\mathcal{L}(w^*) \approx \mathcal{L}(w) + \triangledown\mathcal{L}(w)^\top(w^* - w) + \frac{1}{2}L\|w^* - w\|^2,$$

where $\nabla\mathcal{L}(w) = grad_1$, L is the Lipschitz constant, and $w^* - w = \frac{1}{2}(\hat{grad}_1 + \hat{grad}_2)$, $\hat{grad}_1 = (1-2\lambda)grad_1 + 2\lambda grad_2, \hat{grad}_2 = 2\lambda\hat{grad}_1 + (1-2\lambda)grad_2$.

Now, we substitute the update rule expression for w^* and simplify the inequality using the constraint $t \leq \frac{1}{L}$, which leads to

$$\mathcal{L}(w^*) \leq \mathcal{L}(w) + (8\lambda^4 - \frac{1}{2})t\|grad_1 - grad_2\|^2. \tag{11}$$

As long as $8\lambda^4 - \frac{1}{2} < 0$, we can also have $\mathcal{L}(w^*) < \mathcal{L}(w)$ with each iteration unless $grad_1 = grad_2$.

In summary, we show that optimal value can be obtained in all the cases. This completes the proof. □

Our analysis can be extended to the cases where we have multiple local servers. Particularly, one can treat the gradient from the ith local server as $grad_1$ the weighted summed gradients from the remaining servers as $grad_2$.

3.4 Implementation

Our proposed gradient aggregation algorithm for privacy-preserving constrained domain generalization (PPDG) only relies on the gradient information of each client, which is task-agnostic. Intuitively, each client can send gradient to the central server every iteration, however, such mechanism inevitably increases communication burden between central server and the local server, which is not practical in FL. We thus follow [6] to reduce the computation cost on each client (i.e., increase the number of iterations of training on each client before sending the information to the central server). In this case, we consider to approximate the gradient as $\omega_T - \omega_0$ (in a form of gradient descent), where ω_0 denotes the initial parameters of the client model, and ω_T denote the model parameters after T iterations. We found such strategy to be quite effective on different benchmark datasets.

4 Experiment

We first evaluate our model on the WILDS benchmark [40], which contains a variety of datasets capturing real-world distribution shifts across a diverse range of modalities. We consider three challenging datasets in WILDS benchmark, namely Camelyon17, Poverty, and FMow where the data are *all with privacy concerns* (e.g., Camelyon17 based on healthcare application, and Poverty, FMow based on satellite imagery which can be related to homeland security), under the Federated learning settings. We consider the *gradient/weight-aggregation-based* FL baselines, including **FedAvg** [6], **FedProx** [41] and **COPA** [28], as well as the domain generalization methods, including **AgrSum** and **AgrRand** [34], which aim to explore the consensus gradients across domains and can be extended to the FL setting. Besides, we also adopt **DeepAll** and **PCGrad** [37] (where PCGrad is designed for multi-task learning but can be extended to the FL setting) as baselines for comparison.

Noted that other state-of-the-art non-federated domain generalization baselines are not applicable in our case as they require aggregating data from multiple servers to create a domain shift scenario.

4.1 Results on Camelyon17

The Camelyon17 dataset contains 450,000 scanned patches of breast cancer metastases in lymph-node sections. The data are collected from 5 hospitals, and each hospital can be treated as a single domain. The objective of Camelyon17 is to predict the presence of tumor tissue in the scanned patch. As shown in [40], the variations in data collection and processing brought from different hospital deployments can greatly degrade the performance of tumor tissue prediction. We follow the setting in [40] for our evaluation, where the training data contain scanned patches from three different hospitals, and the validation and test set consist of data from the rest hospitals. We utilize the training set to train our proposed method and evaluate and validation and test set, respectively.

Setting. We follow the experimental protocols proposed in [40], using the DenseNet-121 as the network for model training. As for the model training on local server, we set the learning rate to be 0.001, L2-regularization strength to be 0.01, the batch size to be 32 and adopt SGD optimizer with momentum 0.9. We trained the model for 10 rounds. Each round all clients join the training and the local training epochs is set to 1. We choose the epoch with the highest accuracy in validation split, and report the corresponding test accuracy. We set hyperparameter λ of PPDG to 0.1, and for FedProx baseline, we tune the parameter of the proxy term μ in a large range and set it to 0.1 where the best performance can be achieved.

Results. We first conduct performance comparisons with FL methods. As can be observed from Table 1, our proposed method can achieve better performance compared with FedAvg and FedProx in a large margin. Such observation is reasonable due to the domain alignment strategy for gradient aggregation, such that shared information among domains can be better exploited. We also notice that FedProx generally achieves slightly better performance when compared to FedAvg. One possible reason for this improvement could be that the proximal term in FedProx encourages the updated model weights to stay close to the original model weights, which might contribute to enhanced common knowledge learning.

Subsequently, we discuss the performance comparisons with centralized learning. As we can see, FedAvg and FedProx achieve poorer performance compared with DeepAll baseline in average, which is reasonable due to the data variation across domains. Such results are also consistent with performance in other FL based tasks (e.g., [7]). While COPA could achieve better performance compared with FedAvg and FedProx, its generalization capability is still not desired compared with our proposed method. Nevertheless, our proposed method can achieve a competitive performance compared with DeepAll baseline and with better performance on validation set, which is reasonable since our method can be interpreted as conducting domain alignment by mapping the data to the neural kernel space, such that the shareable information across domains can be learned, further bringing benefits to generalization capability.

Last but not the least, our proposed method can also achieve better performance compared with the gradient based methods AgrSum, AgrRand and PCGrad, which are

TABLE 1
Results on Camelyon17 dataset.

Method	Validation. (%)	**Test. (%)**	**Average. (%)**
DeepAll	87.4	76.8	82.1
FedAvg	80.4	70.2	75.3
FedProx	80.1	71.4	75.8
AgrSum	87.4	71.1	79.3
AgrRand	88.9	68.3	78.6
PCGrad	85.9	70.0	77.5
COPA	88.0	71.6	79.8
PPDG	**89.0**	**73.0**	**81.0**

TABLE 2
Results on Poverty dataset.

Method	Val Pearson r		**Test. Pearson** r	
	Average	Worst	Average	Worst
DeepAll	0.81	0.52	0.75	0.39
FedAvg	0.71	0.31	0.69	0.13
FedProx	0.71	0.29	0.68	0.08
AgrSum	0.56	0.21	0.59	0.20
AgrRand	0.58	0.28	0.59	0.15
PCGrad	0.70	0.34	0.74	0.10
COPA	0.73	0.25	**0.80**	0.21
PPDG	**0.74**	**0.34**	0.79	**0.23**

TABLE 3
Results on FMOW dataset.

Method	Val Accuracy (%)		Test. Accuracy (%)	
	Average	Worst	Average	Worst
DeepAll	60.1	50.2	53.4	32.7
FedAvg	57.8	47.7	52.1	32.9
FedProx	56.5	45.2	50.8	31.9
AgrSum	52.9	45.7	47.3	27.4
AgrRand	53.1	46.8	47.2	28.7
PCGrad	**60.1**	49.6	53.7	32.8
COPA	60.0	47.6	51.3	29.7
PPDG	59.6	**50.4**	**53.8**	**33.9**

designed for domain generalization and multi-task learning task but can also be extended to the FL setting. We observe that significant improvement can be achieved by using our proposed, which further shows the superiority of our proposed PPDG.

4.2 Results on Poverty

The Poverty dataset assembles satellite imagery and survey data (utilized as ground truth) at 19,669 villages from 23 African countries between 2009 and 2016. Poverty is for regression task which aims to predict the real-valued asset wealth index of an area, given its satellite imagery, which is essential for targeted humanitarian efforts in poor regions, especially for much of the developing world where ground-truth measurement of poverty are lacking because of the high field surveys cost. The whole dataset contains 46 different domains: 23 different countries with 2 regions (urban and rural) for each country. The train split contains 26 domains, the validation and test splits divide the rest 20 domains equally. We use the train split for model training and evaluate the performance on validation and test splits by computing Pearson correlation (r) between the predicted and ground-truth asset index, the worst group result evaluates the model's generalization ability over the Urban and rural region shift [40].

Setting. We follow [40] by using the ResNet-18 as the training network, a batch size of 64, and Adam optimizer with an initial learning rate of 0.001 that decays by 0.96 per epoch. We trained the model for 200 epochs and reported the epoch result with highest validation peason-r value along with the corresponding test pearson-r value. We set the hyper-parameter μ of FedProx and λ of PPDG to 0.1. As the number of train domains here is large (with 26 domains), we follow [6] by choosing a selection ratio 0.5 to randomly select 13 clients (i.e., domains) to join each training round with one local epoch training.

Results. We report the result in Table. 2. We see that PPDG obtains the highest validation and test performance under both average and worst sections compared with other federated learning based baselines, especially on test split with an improvement of 0.1 ahead over the FedAvg. The centralized DeepAll baseline shows significant better performance, which is reasonable since it can directly access data from all domains and use them during training stage instead of only performing gradient aggregation.

4.3 Results on FMow

Similar to the Poverty dataset, the FMow dataset also contains satellite images collected from 5 different regions in 16 consecutive years. The task is to predict the land-usage type (62 categories in total such as shopping mall,residential units etc.) from the satellite image. The objective is to generalize the trained model to satellite imagery taken in the future which may be shifted due the infrastructure development across time. Such predictions can contribute to global-scale monitoring of sustainability and economic challenges, aiding policy and humanitarian efforts in applications such as deforestation tracking. We follow the setting in [40] for training, validation and test domain split.

Settings. We follow [40] to train a DenseNet-121 model for the task. The initial learning rate is to set to 10^{-4} that dacays by 0.96 per epoch and the batch size is set to 32. We randomly select 5 domains to join each training round with one local training epoch. We set λ of PPDG to 0.05 and μ of FedProx to 0.01 (we tune μ in a large range and report the best performance we can achieve). For evaluation, we report the average accuracy to evaluate the model's ability to generalize over years, and the worst-case accuracy to measure the model's generalization performance across regions under a time shift.

Results. We can find in Table 3 that our method achieves the best performance on both test and validation sets among all baselines in terms of the worst-case accuracy, and ranks the second in terms of the average accuracy on validation set. The results further justify the superiority of our proposed method.

4.4 Results on Other Datasets

Besides only considering datasets from WILDS benchmark with privacy issue, we further evaluation on two other datasets, RMNIST and TerraInc, from Domainbed benchmark [42]. Specifically, we follow the Domainbed benchmark by using LeNet for RMNIST, and ResNet-18/50 for TerraInc, where we randomly split each source domain into training and validation set in a ratio of 9:1 to tune the hyperparameter by setting $\lambda = 0.001$ for RMNIST, and $\lambda = 0.2$ for TerraInc. As for FedProx, we set $\mu = 0.1$ where the best performance could be obtained. As we can see, our proposed method can generally outperform other methods, which shows that our proposed method is model agnostic and can be generalized to various datasets.

TABLE 4
Results on RMNIST dataset.

RMNIST	0	15	30	45	60	75	Avg.
DeepAll	94.0	98.7	98.4	98.5	98.5	94.6	97.1
FedAvg	82.9	95.6	96.6	96.8	96.2	86.3	92.4
FedProx	79.6	94.7	95.8	95.9	94.6	82.7	90.5
PCGrad	86.1	**97.3**	93.8	95.0	**97.5**	88.4	93.0
AgrSum	72.9	94.3	96.4	96.4	93.9	78.8	88.8
AgrRand	72.7	94.3	96.4	96.4	93.8	79.0	88.8
COPA	83.0	96.0	97.4	97.0	96.1	**88.6**	93.0
PPDG	**84.0**	96.7	**97.7**	**97.8**	97.1	87.4	**93.4**

TABLE 5
Results on TerraInc dataset.

TerraInc	Loc.100	Loc.38	Loc.43	Loc.46	Avg.
		Resnet-18			
DeepAll	49.8	31.3	47.1	37.2	41.4
FedAvg	46.5	38.6	40.2	27.3	38.2
FedProx	43.8	38.1	39.5	29.0	37.6
PCGrad	46.7	41.1	40.4	27.3	38.9
AgrSum	**51.6**	40.7	38.6	35.3	41.6
AgrRand	50.0	39.6	38.3	**35.4**	40.8
COPA	46.8	40.6	42.4	29.5	39.8
PPDG	49.0	**42.0**	**47.6**	32.7	**42.8**
		Resnet-50			
DeepAll	56.0	48.0	54.6	43.3	50.5
FedAvg	**59.2**	48.1	43.6	32.9	45.9
FedProx	50.5	41.4	41.0	32.2	41.3
PCGrad	57.2	46.0	41.7	33.2	44.5
AgrSum	56.7	47.1	40.1	36.3	45.1
AgrRand	57.4	46.2	39.1	36.9	44.9
COPA	59.0	48.2	44.6	33.1	46.2
PPDG	57.2	**48.7**	**49.9**	**37.7**	**48.4**

4.5 Hyperparameter analysis

We now examine the sensitivity of hyperparameter based on the TerraInc dataset. Specifically, our investigation involves assessing the impact of the hyperparameter λ over a broad range (i.e., $[0.01, 0.02, 0.05, 0.1, 0.2, 0.5]$) and the findings of the resulting hyperparameter analysis are presented in the Figure 1 (where the results are reported in the format (λ, average ACC)). The outcomes reveal that the performance is unsatisfactory when λ is relatively small, which is reasonable as it may not lead to gradient modification using our proposed approach. Conversely, the performance improves as λ increases. Nevertheless, λ cannot be excessively large (i.e., $\lambda = 0.5$ in our case) due to the risk of non-convergence during optimization, as demonstrated in our theoretical analysis.

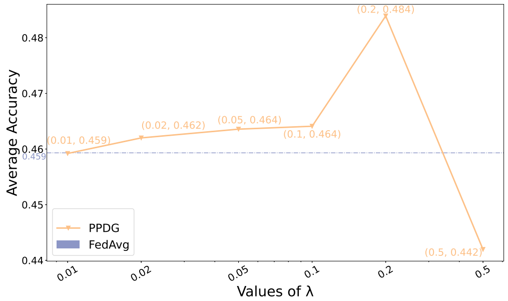

Fig. 1. Hyperparameter Analysis on λ on TerraInc dataset. The dash line denotes the result of FedAvg baseline.

4.6 Ablation study on gradient conflict.

We further propose to conduct an ablation study by excluding the consideration of gradient conflict (i.e., always applying gradient alignment loss between two domains) on TerraInc dataset. The findings in Table 6 indicate that although the results still surpass the FedAvg baseline, the performance decreases in comparison to the outcome obtained from solely considering gradients that lead to negative transfer. This outcome is reasonable because if the gradients between two domains are excessively similar, an overfitting problem may occur, thereby further increasing the gradient distance between the two domains where negative transfer may transpire.

TABLE 6
Comparison between our PPDG and gradient alignment without considering gradient conflict under the setting on TerraInc dataset where the hyperparameter $\lambda = 0.2$.

Method	Acc
FedAvg	45.9
PPDG w/o gradient conflict	47.0
PPDG	48.4

4.7 Statistical Analysis

Besides only reporting results based on only average accuracy/Pearson Correlation, we further perform Test of Significance by using paired-sample t-test on Camelyon17 and RMNIST by using p-value at the 5% significance level. The results are reported in Table 7. From the results, it is evident that h = 1 for all baseline methods, suggesting that it is appropriate to reject the null hypothesis that no difference exists between our proposed approach and the baseline methods. Consequently, we can assert that our proposed method yields significant improvements.

5 Conclusion

In this paper, we focus on the domain generalization problem under the constraint of privacy-preserving settings. In particular, motivated by the theory of kernel embedding on the neural kernel tangent space, we propose a novel gradient aggregation method, which can better extract the shareable information among the data from multiple local servers. We perform experimental studies on various challenging datasets coming from WIDLS and Domainbed benchmarks for classification and regression. The results justify the effectiveness of our proposed method.

TABLE 7
The results of the t-test, which compares our proposed method with the baseline methods, are presented in a format denoted as p/h. The p-value obtained from the t-test is represented by p. A value of h = 1 indicates a statistically significant difference between our proposed method and the baseline, while a value of h = 0 indicates no significant difference.

Method	FedAvg	AgrRand	ArgSum	PCGrad	COPA
Camelyon17	0.0024/1	0.0164/1	0.0043/1	1.63e-4/1	0.0367/1
RMNIST	0.0033/1	1.03e-4/1	1.01e-4/1	0.043/1	0.0212/1

References

[1] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of big data*, vol. 2, no. 1, pp. 1–21, 2015.

[2] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[3] H. Guan and M. Liu, "Domain adaptation for medical image analysis: a survey," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 3, pp. 1173–1185, 2021.

[4] H. Yang, A. S. Coyner, F. Guretno, I. H. Mien, C. S. Foo, J. P. Campbell, S. Ostmo, M. F. Chiang, and P. Krishnaswamy, "A minimally supervised approach for medical image quality assessment in domain shift settings," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 1286–1290.

[5] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTAT*, 2017.

[7] W. Li, F. Milletarì, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. J. Cardoso *et al.*, "Privacy-preserving federated brain tumour segmentation," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2019, pp. 133–141.

[8] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[9] Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *NeurIPS*, 2019.

[10] H. Li, Y. Wang, R. Wan, S. Wang, T.-Q. Li, and A. C. Kot, "Domain generalization for medical imaging classification with linear-dependency regularization," *NeurIPS*, 2020.

[11] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[12] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[13] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *ECCV*, 2014.

[14] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci, "Best sources forward: domain generalization through source-specific nets," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 1353–1357.

[15] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *ICML*, 2013.

[16] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *AAAI*, 2018.

[17] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IROS*. IEEE, 2017.

[18] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," in *NuerIPS*, 2018.

[19] K. Zhou, Y. Yang, T. M. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation." in *AAAI*, 2020.

[20] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[21] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 2552–2559.

[22] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[23] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," *arXiv preprint arXiv:1705.10467*, 2017.

[24] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, and S. Seshan, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2020, pp. 104–115.

[25] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972–5984, 2021.

[26] K. Yue, R. Jin, R. Pilgrim, C.-W. Wong, D. Baron, and H. Dai, "Neural tangent kernel empowered federated learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 783–25 803.

[27] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "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*, 2021, pp. 1013–1023.

[28] G. Wu and S. Gong, "Collaborative optimization and aggregation for decentralized domain generalization and adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6484–6493.

[29] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," *arXiv preprint arXiv:1810.11910*, 2018.

[30] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *CVPR*, 2018.

[31] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," *arXiv preprint arXiv:1806.07572*, 2018.

[32] C. Gentry *et al.*, *A fully homomorphic encryption scheme*. Stanford university Stanford, 2009, vol. 20, no. 9.

[33] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017, pp. 1126–1135.

[34] L. Mansilla, R. Echeveste, D. H. Milone, and E. Ferrante, "Domain generalization via gradient surgery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6630–6638.

[35] Y. Shi, J. Seely, P. H. Torr, N. Siddharth, A. Hannun, N. Usunier, and G. Synnaeve, "Gradient matching for domain generalization," *arXiv preprint arXiv:2104.09937*, 2021.

[36] A. Rame, C. Dancette, and M. Cord, "Fishr: Invariant gradient variances for out-of-distribution generalization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 347–18 377.

[37] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[38] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, pp. 1–36, 2014.

[39] T. Zhang and S. Mao, "An introduction to the federated learning standard," *GetMobile: Mobile Computing and Communications*, vol. 25, no. 3, pp. 18–22, 2022.

[40] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao *et al.*, "Wilds: A benchmark of in-the-wild distribution shifts," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5637–5664.

[41] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[42] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020.