FedLOE: Federated Domain Generalization via Locally Overfit Ensemble

Anonymous authors
Paper under double-blind review

Abstract

In federated learning (FL), clients typically access data from just one distribution. Ideally, the learned models would generalize to out-of-distribution (OOD) data, i.e., domain generalization (DG). However, centralized DG methods cannot easily be adapted in the domain separation context and prior federated DG methods perform poorly when the number of clients is large. To address these challenges, we revisit the classic mixture-of-experts (MoE) idea by viewing each client as an expert on its own dataset. From this perspective, simple federated averaging can be seen as a type of iterative MoE, where the amount of local training determines the strength of each expert. Contrast to the FL communication-performance trade-off, we theoretically demonstrate that in linear cases and empirically validate in deep models that reducing communication frequency can effectively enhance DG performance, surpassing their centralized counterparts (e.g., +4.34% on PACS). Building on this, we further propose an additional MoE strategy to combine the client-specific classifier heads via standard DG objectives. Our proposed FedLOE method can be viewed as an intermediate approach between FedAVG and one-time ensembling. It demonstrates both theoretical soundness and empirical effectiveness. Moreover, FedLOE requires fewer communication rounds, highlighting its practical efficiency and scalability.

1 Introduction

In federated learning (FL), each client owns a portion of the data. For example, in a medical network, each hospital may possess its own set of chest x-rays or tissue sample images, collected using different imaging devices. Similarly, regional datasets capturing economic, crime, or demographic information are often compiled with varying methodologies. As a result, each client's dataset may originate from a distinct domain, leading to what is commonly referred to as *domain separation* (Bai et al., 2023). This setup violates the typical i.i.d. assumption and presents challenges for model training in FL. Additionally, it raises the question: *Can models trained in this setting generalize to clients from previously unseen domains?* This is the central concern of domain generalization (DG), which we consider in the federated setting where domain separation is pronounced and the number of clients is large.

A straightforward approach might be to adapt centralized DG objectives to each client. However, many popular DG methods assume access to multiple domains during training (Arjovsky et al., 2019; Shi et al., 2021; Sun and Saenko, 2016; Li et al., 2018b; Sagawa et al., 2019; Ajakan et al., 2014; Zhou et al., 2021; Li et al., 2018a), which does not hold in the domain separation setting of FL. Existing methods that address DG in FL either do not scale well with many clients (Zhang et al., 2021; Nguyen et al., 2022) or may compromise privacy by sharing frequency spectra of images (Liu et al., 2021).

To address these challenges, we revisit the mixture-of-experts perspective, treating each client as an expert on its local domain. Rather than being a drawback, domain-specific data may enable clients to specialize, potentially offering complementary information across the network. The key challenge then becomes: *How can these expert models be effectively combined to support out-of-domain generalization?*

We consider two combination strategies. First, we explore an *implicit* combination through the standard FedAvg algorithm and find that communication frequency plays an important role in DG performance (see

Section 3). Surprisingly, in some cases, FedAvg with infrequent communication achieves more stable and improved DG performance compared to centralized ERM (which corresponds to communication at every epoch). We hypothesize that frequent communication may lead models to overfit to common spurious correlations, while infrequent communication allows them to retain diverse representations. Our theoretical analysis supports this behavior in a simplified linear setup (Section 3).

Second, we propose an *explicit* ensembling step for combining locally trained classifier heads. This step is motivated by observations in the DG literature: while DG methods can be effective in simple linear settings (Arjovsky et al., 2019; Sun and Saenko, 2016), they often underperform on overparameterized deep models (Gulrajani and Lopez-Paz, 2020; Koh et al., 2021). Moreover, recent studies suggest that the deep feature extractor may already encode useful invariances, but the classifier head can emphasize spurious features (Wald et al., 2022; Rosenfeld et al., 2022). Drawing on these insights, we adopt a two-stage training process that overfits client-specific classifiers and then combines them using a standard DG objective. Our contributions are as follows:

1. We explore a mixture-of-experts interpretation of federated DG and observe that FedAvg with reduced communication frequency can lead to improved out-of-domain generalization under domain separation.
2. We provide a theoretical analysis in a simplified linear setting that sheds light on the conditions under which our approach may outperform centralized ERM.
3. We introduce a two-stage algorithm, FedLOE, that first performs local overfitting followed by aggregation using parameter averaging and a DG objective, respectively.
4. We present empirical results on several real-world datasets, providing evidence that our approach performs comparably or favorably to existing baselines and is robust to varying numbers of clients.

2 Background

Consider a featurizer $g_\theta(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}^d$ with parameters θ and a linear classifier head denoted as $h_\psi(\boldsymbol{z}) : \mathbb{R}^d \to \mathbb{R}^m$ with parameters ψ, where $\boldsymbol{z} = g_\theta(\boldsymbol{x})$. Let $\ell(\cdot, \cdot)$ and $\mathcal{L}(\theta, \psi) := \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(h_\psi(g_\theta(\boldsymbol{x})), y)]$ be the per-sample and expected loss respectively. Let $[A]$ represent the set of integers up to A: $[A] \triangleq \{1, 2, \cdots, A\}$. C denotes the number of clients, K denotes the number of training domains.

2.1 Domain Generalization (DG)

In domain generalization, we will assume there are set of K domain-specific training distributions where the k-th joint distribution is denoted as $p(\boldsymbol{x}, y|k)$ and the marginal probability of each domain is $p(k)$. Additionally, there are one or more domain-specific unseen test distributions denoted by $p(\boldsymbol{x}, y|\tilde{k})$, where $\tilde{k} > K$. The goal of DG is to perform well on these unseen test distributions, which can be formalized as minimizing the expected loss over the set of *unseen* test distributions, i.e.,

$$\min_{\theta,\psi} \sum_{\tilde{k}>K} p(\tilde{k})\mathcal{L}_{\tilde{k}}(\theta, \psi)\,, \tag{1}$$

where $\mathcal{L}_{\tilde{k}}(\theta, \psi) \triangleq \mathbb{E}_{p(\boldsymbol{x},y|\tilde{k})}[\ell(h_\psi(g_\theta(x)), y)]$. Clearly, because the test domains are unknown, a practical proxy objective is simply to use standard empirical risk minimization (ERM) by minimizing the expected loss (i.e., risk) of the training domain distributions:

$$\min_{\theta,\psi} \sum_{k=1}^{K} p(k)\mathcal{L}_k(\theta, \psi)\,, \tag{2}$$

where $\mathcal{L}_k$ are the expected losses of the training distributions. Despite its simplicity, using simple ERM for the DG task has been difficult to beat (Gulrajani and Lopez-Paz, 2020), particularly for training deep non-linear models. Several common approaches add a type of DG regularization term to the ERM objective.

A common approach to DG is through representation learning, including domain-invariant representation learning via kernel methods (Muandet et al., 2013; Ghifary et al., 2016), invariant risk minimization (Arjovsky et al., 2019; Krueger et al., 2020), domain adversarial neural networks (Sun and Saenko, 2016; Li et al., 2018b). Besides domain-invariant learning, several methods add regularizations to the gradient computations (Shi et al., 2021; Rame et al., 2022). Other approaches include distributionally robust optimization (Sagawa et al., 2019), which learns the worst-case distribution scenario of training domains; and meta-learning (Finn et al., 2017; Li et al., 2018a), which is based on the learning-to-learn mechanism to learn general knowledge by constructing meta-learning tasks to simulate domain shift. Most of these methods can be viewed as adding a regularization penalty $r(\theta, \psi)$ with tuning parameter λ to the simple ERM objective:

$$\min_{\theta,\psi} \sum_{k=1}^{K} p(k)\mathcal{L}_k(\theta, \psi) + \lambda r(\theta, \psi)\,. \tag{3}$$

We select three representative examples that we will use in our experiments. In the IRMv1 method from Arjovsky et al. (2019), the regularization term encourages gradients to be zero:

$$r_{\mathrm{irm}}(\theta, \psi) \triangleq \sum_{k} \mathbb{E}_{p_k} \left\| \nabla_{\theta,\psi} \mathcal{L}_k(\theta, \psi) \right\|^2 .$$

Another popular choice is to align the gradients as Fish (Shi et al., 2021) by computing the inner product of the domain-specific loss functions:

$$r_{\mathrm{fish}}(\theta, \psi) \triangleq -\sum_{k \neq k'} \mathbb{E}_{p_k, p_{k'}} \left\langle \nabla_{\theta,\psi} \mathcal{L}_k(\theta, \psi), \nabla_{\theta,\psi} \mathcal{L}_{k'}(\theta, \psi) \right\rangle .$$

The regularization in REx Krueger et al. (2020) seeks to reduce the loss variance across domains:

$$r_{\mathrm{REx}}(\theta, \psi) \triangleq \mathrm{Var}\left[\mathcal{L}_1(\theta, \psi), \mathcal{L}_2(\theta, \psi), \ldots, \mathcal{L}_K(\theta, \psi)\right].$$

2.2 Federated Domain Generalization

In the federated DG setup, we assume C clients, each possessing data from only a single training domain—a specific form of client heterogeneity known as domain separation in Bai et al. (2023).. Formally, client c will have samples from one domain distribution, i.e., $p(\boldsymbol{x}, y|c) \equiv p(\boldsymbol{x}, y|k)$ for some domain k. In our experiments, we will always assume that the number of clients is always larger or equal to the number of training domains $C \geq K$, specifically where C is fairly large in contrast to K, which may only be 4 domains (e.g., in PACS). Importantly, this type of client heterogeneity is distinct from label imbalance, i.e., $\exists k \neq k', p(y|k) \neq p(y|k')$, which is the most common heterogeneity considered in the FL literature. Rather, we assume that the joint distribution could be different, i.e., $\exists k \neq k', p(\boldsymbol{x}, y|k) \neq p(\boldsymbol{x}, y|k')$, but do not assume any particular type of shift between distributions. This is a more general case of client heterogeneity than class imbalance. Like ERM in the centralized case, the standard FL learning algorithm is the FedAvg learning algorithm (McMahan et al., 2017) that minimizes the expected loss on each client and then averages the parameters from all clients.

Using our notation, FedAvg can be compactly formalized as performing a sequence of computational steps denoted by $t \in [T]$ where T is the maximum number of computational epochs, and in each computational step, FedAvg updates the model and sometimes *also* synchronize and average all parameters across clients: for all $c \in [C]$,

$$\left(\psi_c^{t+\frac{1}{2}}, \theta_c^{t+\frac{1}{2}}\right) = \left(\psi_c^t, \theta_c^t\right) - \alpha \nabla \mathcal{L}_c(\psi_c^t, \theta_c^t).$$

$$\left(\psi_c^{t+1}, \theta_c^{t+1}\right) = \begin{cases} \left(\psi_c^{t+\frac{1}{2}}, \theta_c^{t+\frac{1}{2}}\right) & \text{if } t \notin \mathcal{T}, \\ \sum_{c'} \left[\left(\psi_{c'}^{t+\frac{1}{2}}, \theta_{c'}^{t+\frac{1}{2}}\right)\right] & \text{otherwise,} \end{cases} \tag{4}$$

where α is the step size, $\mathcal{T} \subseteq [T]$ denote a set of synchronization indices. If $\mathcal{T} = [T]$ then the synchronization of the sequences is performed every epoch, which corresponds to using mini-batch SGD with mini-batch size C to solve ERM and If $\mathcal{T} = \{T\}$, then (4) amounts to one-shot averaging. A proper choice of the synchronization set $\mathcal{T}$ allows us to obtain an expressive model while keeping the signals of common spurious features relatively low. $\mathcal{T}$ depends on each task and can be determined experimentally. Further, the averaging weights $p(c')$ are chosen based on the training size. Formulated in this way, we notice that FedAvg can be interpreted as a type of local SGD (Stich, 2018), which was originally aimed at solving standard supervised learning (i.e., in-domain accuracy). In Section 3, we show that unlike standard supervised learning tasks—where Stich (2018) show more frequent communication improves the in-domain accuracy—the federated DG accuracy is not monotonic w.r.t. communication frequency.

To the best of our knowledge, only a few works seek to solve federated DG. In particular, Nguyen et al. (2022) proposed FedSR where they enable domain generalization while still respect the distributed and privacy-preserving natures of FL context by enforcing ℓ_2 norm regularizer and a conditional mutual information regularizer on the representation. Liu et al. (2021) propose FedDG, a federated learning paradigm specifically designed for medical image classification. The proposed method requires sharing the amplitude spectrum of images among local clients, which violates the privacy protocol. Zhang et al. (2021) applies generative adversarial network (GAN) (Goodfellow et al., 2020) in the FL context, where it first trains the featurizer and classifier by minimizing the empirical loss, then trains the generator and the discriminator using a GAN-based (Goodfellow et al., 2020) approach, and FedGMA (Tenison et al., 2022) proposes a mask on the gradient on the server side to align the updates among domains.

3 Unraveling the Benefits of Expert Parameter Ensembling for Domain Generalization

In this section, we motivate by a simple experiment showing the frequent communication can lead to out-of-distribution (i.e., DG) overfitting. We then show theoretically that infrequent communication leads to better DG accuracy on a linear structural causal model.

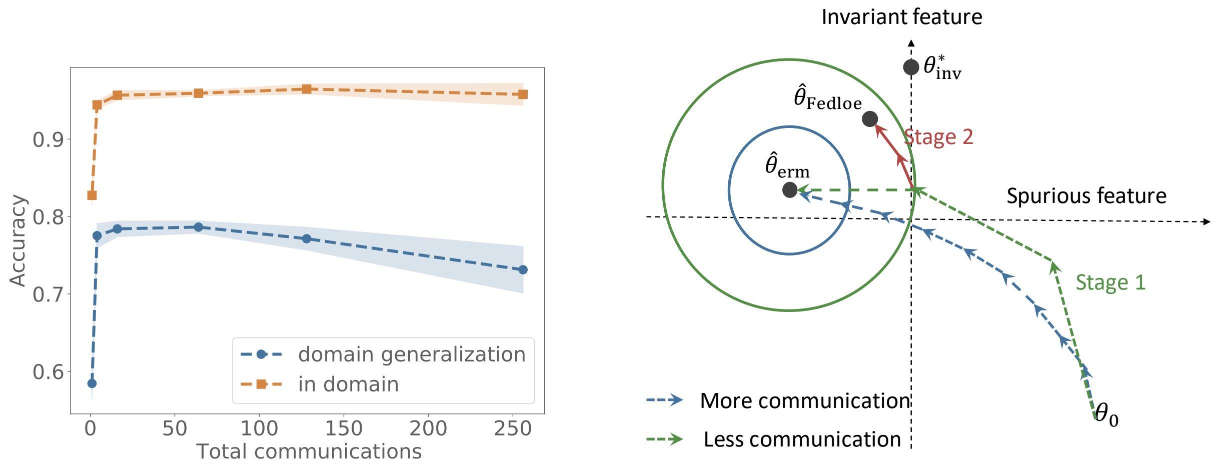
Motivating observation: infrequent communication leads to better DG accuracy We start with a surprising phenomenon related to the effect of communication frequency on DG accuracy when using FedAvg. To demonstrate the phenomena, we conducted an experiment on the PACS dataset using 20 clients with the same model and a fixed computational budget of 256 epochs while allowing the number of communications to vary, i.e., how many times the model parameters are averaged. As seen in Figure 1a, while in-domain accuracy steadily increases and stays nearly constant, the optimal communication frequency for DG is actually a very small number of communications. In terms of DG performance, FedAvg with 4 communications attains the best performance and stability as seen by the low variance. In contrast, its centralized counterpart mini-batch SGD with total communications 256 performs poorly. These results indicate that increasing communication frequency in federated learning can potentially hinder DG performance.

Theoretic analysis of expert parameter ensemble for linear structural equation model We now theoretically validate our empirical finding by analyzing a linear DG problem to provide further theoretic insight into the expert parameter ensembling. Our analysis sheds light on the potential for parameter averaging of locally overfit models (i.e., experts) to indeed improve DG performance. We expect that this result could be extended to more complex linear settings (and possibly non-linear) but focus on a simple clear example here. Concretely, we extend the linear model example from Arjovsky et al. (2019) to multiple invariant and spurious features. Consider the following structural equation model:

$$\boldsymbol{x}_{\mathcal{I}} = \boldsymbol{\xi}_{\mathbf{1}}, \quad y = \boldsymbol{x}_{\mathcal{I}}^{\top}\boldsymbol{\alpha}_{\mathcal{I}} + \xi_2, \quad \boldsymbol{x}_{\mathcal{S}} = y\boldsymbol{\alpha}_{\mathcal{S}} + \boldsymbol{\xi}_{\mathbf{3}}, \tag{5}$$

where $\xi_i, i \in \{1, 2, 3\}$ are exogenous noise variables, $y \in \mathbb{R}$ is the regression target, $\boldsymbol{x}_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ is the invariant feature vector, $\boldsymbol{x}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ is the spurious feature vector, and $\boldsymbol{\alpha}_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ and $\boldsymbol{\alpha}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ are the invariant and spurious parameters, respectively. Note that y is causally generated by $\boldsymbol{x}_{\mathcal{I}}$ while $\boldsymbol{x}_{\mathcal{S}}$ are spurious features because they are generated from y, i.e., they are causal descendants.

To frame the DG problem, we assume that each domain or environment shares the above causal model (including the same $\boldsymbol{\alpha}_{\mathcal{I}}$ and $\boldsymbol{\alpha}_{\mathcal{S}}$) but has domain-specific noise distributions for $\boldsymbol{\xi}_{\mathbf{1}}$ and ξ_2. Additionally, for



(a) While more communication improves *in-domain* test accuracy, more communication may harm out-of-distribution domain generalization accuracy. The left-most points represent one-shot averaging (i.e., only one communication), while the rightmost point corresponds to ERM (i.e., communicate every batch).

(b) This figure illustrates how the infrequent communication of FedLOE's first stage (green) may implicitly regularizes the solution away from the ERM solution $\widehat{\theta}_{\text{erm}}$ but closer to the robust invariant solution $\widehat{\theta}_{\text{inv}}$, and FedLOE's second stage (red) explicitly moves toward the invariant solution via DG objectives while standard FedAvg with frequent communication (blue) converges to the ERM solution.

the FL context, we assume that each client has access to data from a single domain. We formalize these assumptions next.

Assumption 3.1 (Independent domain-specific sub-Gaussian exogenous noises)**.** Noise satisfies

1. Each element of the exogenous noises are independent and sub-Gaussian.
2. $\boldsymbol{\xi_1}$ and ξ_2 are zero mean with domain-specific variances, i.e., $\mathbb{E}[\boldsymbol{\xi_1}] = \mathbf{0}$, $\mathbb{E}[\boldsymbol{\xi_1}\boldsymbol{\xi_1}^\top] = \sigma_1^2(k)I$, $\mathbb{E}[\xi_2] = 0$, and $\mathbb{E}[\xi_2^2] = \sigma_2(k)$.
3. $\boldsymbol{\xi_3}$ is zero mean with a shared variance across domains, i.e., $\mathbb{E}[\boldsymbol{\xi_3}] = \mathbf{0}$, and $\mathbb{E}[\boldsymbol{\xi_3}\boldsymbol{\xi_3}^\top] = \sigma_3 I$.

Assumption 3.2 (Domain separation)**.** Each client $c \in [C]$ has data from a unique non-overlapping domains $k \in [K]$ where $C \equiv K$ in this example.

Given this problem context, we consider the OLS estimator of the coefficient $\boldsymbol{\alpha} := [\boldsymbol{\alpha_{\mathcal{I}}}, \boldsymbol{\alpha_{\mathcal{S}}}]^\top$ for predicting y from the concatenated features $\boldsymbol{x} = [\boldsymbol{x_{\mathcal{I}}}, \boldsymbol{x_{\mathcal{S}}}]^\top \in \mathbb{R}^d$:

$$\widehat{\boldsymbol{\alpha}} \in \operatorname*{arg\,min}_{\boldsymbol{\alpha}\in\mathbb{R}^d} \mathbb{E}_{(\boldsymbol{x},y)} \left[y - \boldsymbol{x}^\top\boldsymbol{\alpha}\right]^2 . \tag{6}$$

The *domain-invariant*, i.e., robust solution is to simply ignore the spurious features $\boldsymbol{x_{\mathcal{S}}}$ and only use the linear coefficients of the invariant features, i.e., $\boldsymbol{\alpha}^* := [\boldsymbol{\alpha_{\mathcal{I}}}, \mathbf{0}]^\top$. This solution also corresponds to finding the true causal mechanism that generates y. We compare two distinct cases in this context: (1) the clients communicate every epoch so that the solution converges to the ERM solution (Stich, 2018) or (2) the clients solve their local problems and only communicate once, i.e., one-shot averaging of the model parameters.

Theorem 3.3. *Given the least squares problem* (6) *under Assumption 3.1 and Assumption 3.2, one-shot averaging will put relatively more weight on the invariant features, i.e.,*

$$\frac{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}}\|}{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}}\|} < \frac{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}}\|}{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}}\|}, \tag{7}$$

and OSA will be closer than ERM to the invariant solution $\boldsymbol{\alpha}^* \triangleq [\boldsymbol{\alpha_{\mathcal{I}}}, \mathbf{0}]^\top$, *i.e.,*

$$\|\widehat{\boldsymbol{\alpha}}^{\text{ERM}} - \boldsymbol{\alpha}^*\| > \|\widehat{\boldsymbol{\alpha}}^{\text{OSA}} - \boldsymbol{\alpha}^*\|. \tag{8}$$

Corollary 3.4. *Under the above assumption, the DG risk of ERM is higher than that of OSA. i.e.,*

$$\mathbb{E}\left[(y - \boldsymbol{x}^\top \widehat{\boldsymbol{\alpha}}^{\mathrm{ERM}})^2\right] > \mathbb{E}\left[(y - \boldsymbol{x}^\top \widehat{\boldsymbol{\alpha}}^{\mathrm{OSA}})^2\right] \, , \tag{9}$$

where $(\boldsymbol{x}, y) \sim \mathcal{D}_{\text{test}}$.

While the full proofs are Appendix A and Appendix B respectively, the key to the proof relies on the fact the harmonic mean (the ERM solution) is less than the arithmetic mean (the OSA solution). Theorem 3.3 shows that simple one shot ensemble of local estimators can be more robust than ERM. Furthermore, Appendix B shows that OSA could achieve lower DG risk. This validates our empirical findings and suggests that frequent communication may over-emphasize spurious features. Further theoretic analysis would be required to analyze cases in between these two extremes that are more likely to have better DG performance than either extreme as seen empirically in Figure 1a. Given our empirical results, we expect this type of behavior to hold beyond this specific example.

4 FedLOE: Locally Overfit Ensemble Algorithm for Federated DG

In this section, we establish a framework of new algorithms by creating experts via client-specific training followed by expert ensembling via linear parameter combination. Our method aims to reduce overfitting to common spurious features, induce robust features, and control data privacy loss. Our algorithm is designed in two stages that both have the same structure: (1) construct expert models by overfitting locally and (2) combine experts via linear combination as outlined in Table 1 and illustrated in Figure 1b.

Table 1: FedLOE: An iterative two stage framework for federated DG that locally overfits and ensembles first all model parameters (θ, ψ) and then only the linear classifier head parameters ψ.

	Stage 1: Learning shared features	Stage 2: Robustifying classifier head
Step 1: Locally overfit	SGD on θ and ψ	SGD on ψ
Step 2: Ensemble parameters	Simple average of parameters	Linearly combine via DG objective

In stage one, we draw from the key observation in Section 3 and reinterpret FedAvg with *infrequent* communication as locally overfitting and then averaging the local expert parameters. Infrequent communication helps to avoid common spurious features while still learning an expressive model. Additionally, infrequent communication is better for communication constrained FL on edge devices. Even by itself, stage one increases DG robustness compared to synchronizing every mini-batch (which is equivalent to centralized SGD), which would require the most frequent communications. In stage two, we freeze the feature extractor parameters but allow each client to become an expert by training its own linear classifier head locally on its own dataset (note this requires no communication). Then we ensemble the expert classifier heads using standard DG objectives to make the final ensemble classifier more robust.

4.1 Stage 1: Learning Shared Features via FedAvg with Infrequent Communication

Motivated by Section 3 that frequent averaging may amplify the common spurious features in exploitative experiments, we use FedAvg with infrequent communication schedule (4) with $\mathcal{T} = \mathcal{T}_1$. This can be viewed as overfitting each client model on its own data and ensembling these models via parameter averaging where the communication frequency determines how specialized each expert becomes. $\mathcal{T}_1$ is tuned experimentally and generally is a much smaller set than all possible time points. For example, in Figure 1a, the best $|\mathcal{T}_1| = 4$. By the end of this stage, all clients share a common featurizer $g_{\widehat{\theta}}$ and a linear classifier ψ^{T_1}. An ablation study in Section 5.1 also confirms that if we replace stage 1 with the most frequent communications, i.e., mini batch SGD, the DG accuracy would decrease significantly. This verifies our explanation in the exploratory experiments in Figure 1a that frequent communication induces common spurious features. At the end of the first stage, the featurizer parameters are fixed and we will focus on the last linear layer because prior work has shown that the deep featurizer may be good enough and only the last classifier layer needs to be made robust (Wald et al., 2022).

4.2 Stage 2: Robustifying the Linear Classifier Head

In Stage 2, we follow a similar overfitting and ensembling strategy as in stage 1 but freeze the featurizer and update only the linear classifier head. The key difference is that we compute a specialized linear combination of the classifier heads via DG objectives rather than computing a simple average as in FedAvg.

Stage 2.1: Local overfitting of classifier heads. In this substage, the clients become experts on their own local datasets by training only their classifier heads $\psi_c, c \in [C]$ on their local datasets $\{(\boldsymbol{x}_i^{\text{train}}, y_i^{\text{train}})\}_{i \in \mathcal{D}_c^{\text{train}}}$ while keeping the featurizer $g_{\widehat{\theta}}$ from stage 1 fixed. Specifically, for each $c \in [C]$, we initialize at the current classifier head and use SGD to solve

$$\widehat{\psi}_c \triangleq \arg\min_{\psi_c} \mathcal{L}_c(\hat{\theta}, \psi_c) \tag{10}$$

Thus, by the end of substage 2.1, we obtain C distinct linear classifiers $\widehat{\psi}_1, \dots, \widehat{\psi}_C$, where each $\widehat{\psi}_c \in \mathbb{R}^{m \times d}$.

Stage 2.2: Robust ensemble of classifier heads via DG methods. In this stage, we aim to train a robust linear combination of the expert classifier we get from stage 2.1. We utilize DG objectives such as IRM (Arjovsky et al., 2019) and Fish (Shi et al., 2021) to promote DG performance. This step is done on the aggregation server, thus requiring sending a small portion of the *predictions* to the aggregation server. Notice that we only send m dimensional prediction which are much less than the original D dimensional data. We now explain the specifics of this stage in more detail. First, the server will collect and broadcast the all the linear classifiers $\widehat{\psi}_1, \dots, \widehat{\psi}_C$ to each client. Each client will then compute the predictions using each classifier for a tuning dataset, which is usually 10 times smaller than the training dataset. We will denote the prediction on the c-th client of a local data point $\boldsymbol{x}_c$ using the c'-th client's classifier as $\widehat{y}_{c,c'} \triangleq h_{\hat{\psi}_{c'}}(g_{\widehat{\theta}}(\boldsymbol{x}_c))$. Each client will then send all its predictions and the true label for its tuning dataset to the server. The server will then estimate the optimal robust linear combination of these predictions using a regularized DG objective as follows:

Table 2: PACS: We show that our methods outperform centralized SGD. Our methods as well as FedAvg without frequent communication are more stable.

	DG accuracy (by domain)				
Test Domain	Photo	Art	Cartoon	Sketch	Average
Mini-batch SGD	93.37 ± 0.52	81.98 ± 1.71	77.72 ± 0.67	75.98 ± 2.92	82.26
FedDG	96.67 ± 0.28	84.40 ± 0.92	75.77 ± 0.85	74.97 ± 1.74	82.95
FedSR	92.63 ± 0.81	79.73 ± 1.19	73.90 ± 3.08	69.93 ± 0.10	79.05
FedGMA	97.82 ± 0.75	88.17 ± 1.01	77.40 ± 0.94	79.30 ± 0.32	85.67
Our FedAvg	97.70 ± 0.00	$\mathbf{88.98 \pm 0.00}$	78.63 ± 0.01	78.61 ± 0.02	85.98
FedLOE-IRM	97.67 ± 0.77	86.04 ± 0.72	$\mathbf{80.01 \pm 1.01}$	79.01 ± 0.98	85.68
FedLOE-Fish	$\mathbf{98.25 \pm 0.25}$	88.62 ± 0.59	78.53 ± 0.72	$\mathbf{81.01 \pm 0.69}$	**86.60**
FedLOE-REx	97.75 ± 0.94	86.66 ± 0.71	78.87 ± 0.55	79.44 ± 0.70	85.68

$$\widehat{\phi} = \arg\min_{\phi \in \mathbb{R}^C} \sum_c p(c) \mathbb{E}^{\text{tune}}_{p(\hat{y}, y|c)} \left[\ell \left(\sum_{c'=1}^{C} \phi_{c'} \widehat{y}_{c'}, y \right) \right] + \lambda \cdot r \left(\sum_{c'=1}^{C} \phi_{c'} \widehat{y}_{c'}, y \right), \tag{11}$$

where $r(\cdot)$ is a DG regularization term (see Appendix C.1 for detailed expression for each method) and λ is the regularization parameter. From one perspective, this can be seen as training a last "meta" linear layer that combines the outputs of the C client classifiers. From another perspective, because both the classifiers are linear and this meta-combination is linear, the resulting ensemble classifier can be simplified to a single linear classifier by taking a combination of the client classifiers, i.e., $\psi_{\text{ensemble}} = \sum_{c=1}^{C} \widehat{\phi}_c \widehat{\psi}_c$. This second perspective allows us to write both steps of this stage in a similar manner to FedAvg with the important exception that the linear combination weights $\widehat{\phi}$ are learned via the DG objective in Equation (11), i.e.,

$$\forall c,\ \widehat{\psi}_c^t = \begin{cases} \widehat{\psi}_c^{t-1} - \eta_c^{t-1}\nabla_\psi \mathcal{L}_c(\hat{\theta}, \widehat{\psi}_c^{t-1}), & \text{if } t \notin \mathcal{T}_2 \\ \sum_{c'=1}^{C} \widehat{\phi}_{c'}^t \widehat{\psi}_{c'}^{t-1}, & \text{if } t \in \mathcal{T}_2. \end{cases} \tag{12}$$

In summary, our FedLOE provides a general framework for domain generalization methods in the federated context, which can incorporate newly designed centralized DG methods in the second stage to provide better DG accuracy (see Section 5 for experiment results). The framework naturally fits into the FL context with low communication burden. The key idea of the framework is to "free" the clients by letting them overfit in domain and avoid common spurious features; further personalize their classifiers; and encourage invariant predictions by solving a linear classification problems with DG regularizations. See Algorithm 1 in the appendix for more details. Our ablation study in Section 5.1 shows that without stage 2 or training stage 2 with SGD, the DG accuracy would stay the same as FedAvg.

5 Experiments

In this section, we evaluate FedLOE on three real-world datasets PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017) and IWildCam-Wilds (Koh et al., 2021). We choose popular IRM (Arjovsky et al., 2019), Fish (Shi et al., 2021) and REx (Krueger et al., 2020) as the DG regularizer for stage 2, and we compare them with FedAvg (McMahan et al., 2017), FedDG (Liu et al., 2021) FedSR (Nguyen et al., 2022) and FedGMA (Tenison et al., 2022), which were originally designed for solving DG in the FL regime. In Section 5.1, we explore the impact of the number of clients and the effect of stage 2. For all these experiments except for the ablation study, we report average performance as well as the standard error over 5 different runs. We explain the most important experimental settings the experimental setting here but refer the reader to more details in Appendix D.

Table 3: OfficeHome: We show that our methods outperform centralized SGD. Our methods as well as FedAvg without frequent communication are more stable.

	DG accuracy (by domain)				
Test Domain	Art	Clipart	Product	Real-world	Average
Mini-batch SGD	60.06 ± 2.01	50.32 ± 1.25	73.20 ± 1.49	75.63 ± 1.85	64.80
FedDG	58.58 ± 0.74	50.92 ± 0.89	75.41 ± 0.43	75.97 ± 0.98	65.22
FedSR	1.54 ± 0.01	1.53 ± 0.01	1.52 ± 0.01	1.53 ± 0.01	1.53
FedGMA	58.19 ± 2.74	42.33 ± 3.65	68.90 ± 1.64	71.70 ± 2.07	60.28
Our FedAvg	63.70 ± 0.01	50.71 ± 0.02	76.74 ± 0.00	78.74 ± 0.01	67.47
FedLOE-IRM	$\mathbf{65.55 \pm 0.83}$	$\mathbf{51.23 \pm 0.45}$	76.10 ± 0.94	$\mathbf{78.95 \pm 1.03}$	**67.95**
FedLOE-Fish	64.79 ± 0.59	50.79 ± 0.39	$\mathbf{76.93 \pm 0.44}$	78.4 ± 0.51	67.72
FedLOE-REx	64.01 ± 0.29	50.52 ± 0.38	77.32 ± 0.34	78.10 ± 0.42	67.49

5.1 Main Results

We present the experiment results of our methods as well as the baseline methods on PACS in Table 2, OfficeHome in Table 3 and IWildCam in Table 4. We observe that FedAvg outperforms centralized ERM on all three datasets, which validates our theory that frequent communication over different domains is overshooting, and our FedLOE method with both stages is comparable or better than FedAvg on both PACS and OfficeHome datasets. This suggests that in some cases, our second stage of robust ensembling can improve performance. Furthermore, we achieve the highest performance on the leaderboard of the Wilds Benchmark using ResNet-50 without targeted augmentation in this client heterogeneity setting using infrequent-communicate FedAvg. It is also worth noting that our FedLOE-IRM, FedLOE-Fish, and FedLOE-REx methods exhibit less effectiveness compared to FedAvg. This could be attributed to the fact that IRM, Fish, and REx did not demonstrate significant effectiveness in the centralized setting, as indicated by the wilds benchmark leaderboard (Koh et al., 2021).

The influence of number of clients C: In this study, we investigate the performance of various algorithms in the FL context as the number of clients C increases on PACS. As shown in Figure 2, OOD accuracy is plotted against the number of clients, where each client only contains samples from a single domain. The results indicate that: **1)** the performance of FedSR degrades as the number of clients increases. **2)** In contrast, our proposed method, FedLOE, not only achieves higher accuracy, but also exhibits greater robustness to larger client number settings.

The Effect of Stage 2: In Figure 3, the DG accuracy per communication is plotted, illustrating the performance boost over FedAvg on both PACS and OfficeHome datasets. Specifically, the results demonstrate that FedAvg achieves the same accuracy with the same number of communications and computations as our FedLOE-IRM indicating that FedAvg converges.

Table 4: IWildCam: We show that our methods outperform centralized SGD. Our FedAvg with infrequent communication performs better and is more stable.

	F1 score
Mini-batch SGD	31.61 ± 1.88
FedDG	31.15 ± 1.72
FedSR	0.01 ± 0.00
FedGMA	26.05 ± 2.24
Our FedAvg	$\mathbf{33.02 \pm 0.61}$
FedLOE-IRM	31.02 ± 0.65
FedLOE-Fish	31.14 ± 0.91
FedLOE-REx	31.96 ± 0.76

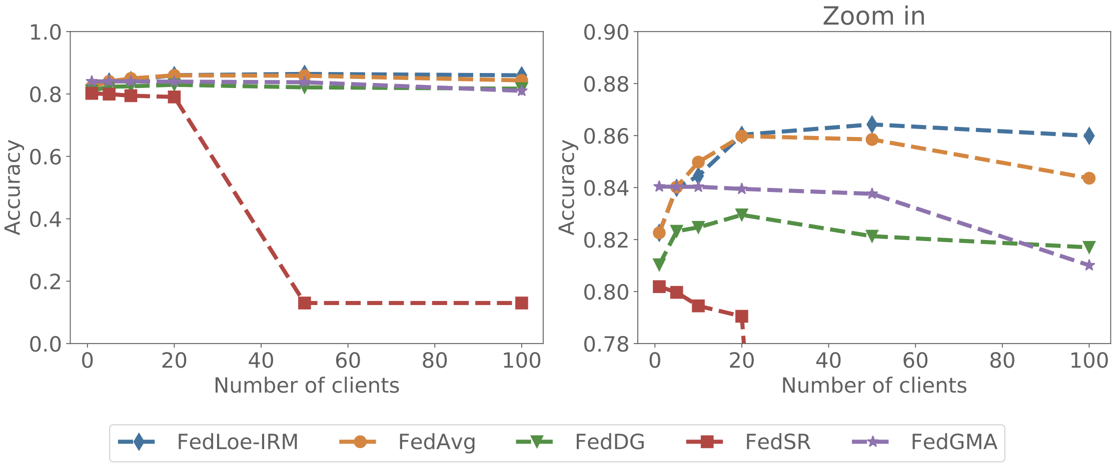


Figure 2: OOD accuracy with changing number of clients C. The second figure is the zoom in of the first figure showing the detail of each methods.

6 Related Work

In this section, we discuss some related approaches.

6.1 Federated Domain Generalization

Federated Domain Generalization (Federated DG) emerges as a natural extension of centralized domain generalization, especially suited for real-world applications where data is inherently distributed across multiple domains. In such settings, the need for robust generalization to unseen domains is both intuitive and

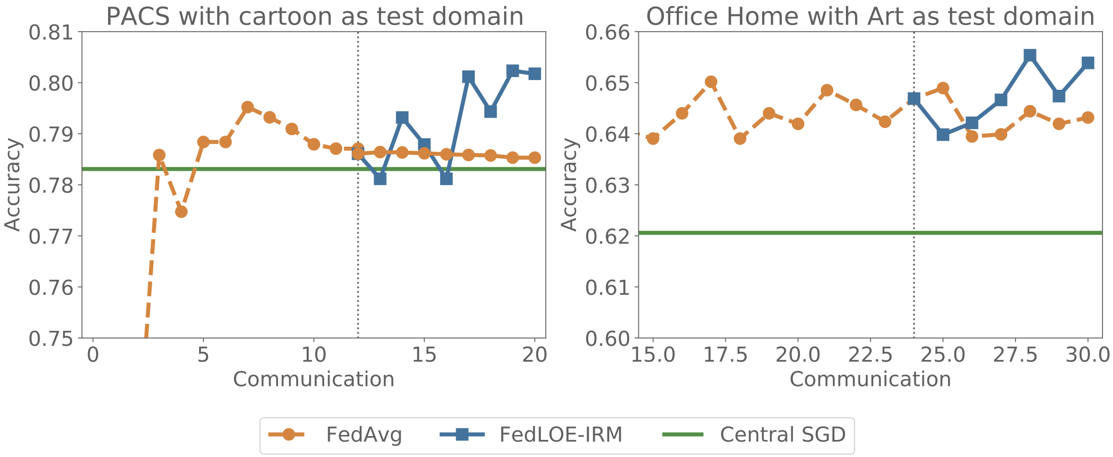


Figure 3: DG accuracy per communication on PACS and OfficeHome dataset, the second stage of our method starts at 12 and 24 respectively (blue).

essential. However, Federated DG introduces two key challenges. First, the presence of non-iid (heterogeneous) data distributions across clients complicates model convergence, even before considering discrepancies between training and testing domains. Second, strict privacy requirements and communication limitations often preclude direct data sharing across domains, thereby rendering many traditional, centralized domain generalization approaches infeasible in federated settings.

6.2 Model Ensemble

Model ensemble methods aim to improve generalization by combining predictions from multiple models, often trained under varying conditions or on different subsets of data. In the context of Federated DG, ensemble methods have been leveraged to account for domain shifts without requiring centralized data sharing. For instance, FedEM (Marfoq et al., 2021) ensembles personalized models from different clients to form a more generalizable global predictor. FedCE (Cai et al., 2023) continues this line of work by introducing a contrastive ensembling framework that encourages consistency across local models while maintaining domain-specific diversity. These approaches highlight the strength of ensemble-based strategies in mitigating the effects of domain heterogeneity and enhancing robustness to unseen target domains. However, ensembling can incur increased computational and communication overhead, particularly in resource-constrained federated environments, making scalability a key concern.

7 Conclusion and Discussion

Our paper starts with a novel mixture-of-experts perspective on FL DG in the domain separation context where each client is viewed as an expert. Our method, while preserving privacy, still shares model parameters to the server. We could further perform differential privacy (Dwork et al., 2006) to protect the privacy.

The key observation is that locally overfitting and then refitting can actually avoid collusion among experts so that they do not find a common set of spurious features—as noted in several recent works. Given our perspective and these observations, we design a novel federated DG method FedLOE that combines two stages. The first stage uses the natural and implicit combination strategy of federated averaging to learn a good deep featurizer. The second stage focuses on making the linear classifier head robust by iteratively locally overfitting the heads and then optimizing the best linear combination of these heads using standard DG objectives. Given that the second stage employs standard DG objectives as subroutines, it offers flexibility that allows us to integrate any future methods aimed at enhancing DG accuracy. More broadly, this work suggests that client heterogeneity in FL may actually be beneficial for OOD generalization in contrast to the usual assumption that data heterogeneity impairs FL methods.

References

Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Ruqi Bai, Saurabh Bagchi, and David I. Inouye. Benchmarking algorithms for domain generalization in federated learning, 2023. URL https://openreview.net/forum?id=IsCg7qoy8i9.

Luxin Cai, Naiyue Chen, Yuanzhouhan Cao, Jiahuan He, and Yidong Li. Fedce: Personalized federated learning method based on clustering ensembles. In *Proceedings of the 31st ACM international conference on multimedia*, pages 1625–1633, 2023.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex), 2020. URL https://arxiv.org/abs/2003.00688.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018b.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1013–1023, 2021.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34: 15434–15447, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR, 2013.

A Tuan Nguyen, Ser-Nam Lim, and Philip HS Torr. Fedsr: Simple and effective domain generalization method for federated learning. *International Conference on Learning Representations*, 2022.

Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR, 2022.

Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. Domain-adjusted regression or: ERM may already learn features sufficient for out-of-distribution generalization. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=Ypo0AckYW8.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer vision–ECCV 2016 workshops: Amsterdam, the Netherlands, October 8-10 and 15-16, 2016, proceedings, part III 14*, pages 443–450. Springer, 2016.

Irene Tenison, Sai Aravind Sreeramadas, Vaikkunth Mugunthan, Edouard Oyallon, Eugene Belilovsky, and Irina Rish. Gradient masked averaging for federated learning. *arXiv preprint arXiv:2201.11986*, 2022.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. Malign overfitting: Interpolation and invariance are fundamentally at odds. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Liling Zhang, Xinyu Lei, Yichun Shi, Hongyu Huang, and Chao Chen. Federated learning with domain generalization. *arXiv preprint arXiv:2111.10487*, 2021.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6xHJ37MVxxp.

Appendix

Table of Contents

A Proof of Theorem 3.3

Proof. At the solution $\widehat{\boldsymbol{\alpha}}$, we have

$$\begin{cases} \left.\frac{\partial L}{\partial \widetilde{\boldsymbol{\alpha}}_{\mathcal{I}}}\right|_{\widetilde{\boldsymbol{\alpha}}_{\mathcal{I}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}} = 0, \\ \left.\frac{\partial L}{\partial \widetilde{\boldsymbol{\alpha}}_{\mathcal{S}}}\right|_{\widetilde{\boldsymbol{\alpha}}_{\mathcal{S}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}} = 0. \end{cases}$$

For all invariant features, we have

$$\begin{aligned} \left.\frac{\partial L}{\partial \widetilde{\boldsymbol{\alpha}}_{\mathcal{I}}}\right|_{\widetilde{\boldsymbol{\alpha}}_{\mathcal{I}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}} &= 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}(\boldsymbol{x}^\top \widehat{\boldsymbol{\alpha}} - y)\right] \\ &= 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}(\boldsymbol{x}^\top \widehat{\boldsymbol{\alpha}} - \boldsymbol{x}_{\mathcal{I}}^\top \boldsymbol{\alpha}_{\mathcal{I}} - \xi_2)\right] \\ &\overset{(a)}{=} 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}(\boldsymbol{x}^\top \widehat{\boldsymbol{\alpha}} - \boldsymbol{x}_{\mathcal{I}}^\top \boldsymbol{\alpha}_{\mathcal{I}})\right] \\ &= 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}(\boldsymbol{x}_{\mathcal{I}}^\top (\widehat{\boldsymbol{\alpha}}_{\mathcal{I}} - \boldsymbol{\alpha}_{\mathcal{I}}) + \boldsymbol{x}_{\mathcal{S}}^\top \widehat{\boldsymbol{\alpha}}_{\mathcal{S}})\right] = 0, \end{aligned} \tag{13}$$

therefore, there holds

$$\mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}}^\top (\boldsymbol{\alpha}_{\mathcal{I}} - \widehat{\boldsymbol{\alpha}}_{\mathcal{I}})\right] = \mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{S}}^\top \widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\right], \tag{14}$$

which amounts to

$$\mathbb{E}\left[\boldsymbol{\xi}_{\mathbf{1}}\boldsymbol{\xi}_{\mathbf{1}}^\top (\boldsymbol{\alpha}_{\mathcal{I}} - \widehat{\boldsymbol{\alpha}}_{\mathcal{I}})\right] = \mathbb{E}\left[\boldsymbol{\xi}_{\mathbf{1}}\boldsymbol{\xi}_{\mathbf{1}}^\top \boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^\top \widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\right]. \tag{15}$$

For all spurious features, we have

$$
\begin{aligned}
\frac{\partial L}{\partial \widetilde{\boldsymbol{\alpha}}_{\mathcal{S}}}\bigg|_{\widetilde{\boldsymbol{\alpha}}_{\mathcal{S}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}} &= 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{S}}(\boldsymbol{x}^{\top}\widehat{\boldsymbol{\alpha}}-y)\right] \\
&= 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{S}}(\boldsymbol{x}^{\top}\widehat{\boldsymbol{\alpha}}-\boldsymbol{x}_{\mathcal{I}}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}-\xi_2)\right] \\
&= 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{I}}^{\top}\left(\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\right)+\boldsymbol{x}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}})\right]-2\mathbb{E}\left[\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}\xi_2+\xi_2^2\boldsymbol{\alpha}_{\mathcal{S}}+\boldsymbol{\xi_3}\xi_2\right] \\
&\overset{\text{(b)}}{=} 2\mathbb{E}\left[\boldsymbol{x}_{\mathcal{S}}(\boldsymbol{x}_{\mathcal{I}}^{\top}\left(\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\right)+\boldsymbol{x}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}})\right]-2\mathbb{E}\left[\xi_2^2\boldsymbol{\alpha}_{\mathcal{S}}\right] \\
&= 2\mathbb{E}\left[(y\boldsymbol{\alpha}_{\mathcal{S}}+\boldsymbol{\xi_3}))(\boldsymbol{\xi_1}^{\top}\left(\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\right)+(y\boldsymbol{\alpha}_{\mathcal{S}}+\boldsymbol{\xi_3}))^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}})\right]-2\mathbb{E}\left[\xi_2^2\boldsymbol{\alpha}_{\mathcal{S}}\right] \\
&= 2\mathbb{E}\left[\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}\left\{\boldsymbol{\xi_1}^{\top}\left(\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\right)\right\}\right]+2\mathbb{E}\big[\big[(\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}+\xi_2)^2\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}+\boldsymbol{\xi_3}\boldsymbol{\xi_3}^{\top}\big]\,\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\big\}\big]-2\mathbb{E}\left[\xi_2^2\boldsymbol{\alpha}_{\mathcal{S}}\right] \\
&= 0,
\end{aligned}
$$

where (b) results from the independence of $\boldsymbol{\xi_1}, \xi_2$ and $\boldsymbol{\xi_3}$. Therefore, there holds

$$
\mathbb{E}\left[\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}\left\{\boldsymbol{\xi_1}^{\top}\left(\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\right)\right\}\right]+\mathbb{E}\big[\big[(\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}+\xi_2)^2\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}+\boldsymbol{\xi_3}\boldsymbol{\xi_3}^{\top}\big]\,\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\big\}\big]=\mathbb{E}\left[\xi_2^2\boldsymbol{\alpha}_{\mathcal{S}}\right]. \tag{16}
$$

Chaining (15) and (16), we have

$$
\left\{\begin{array}{l}
\mathbb{E}\left[\boldsymbol{\xi_1}\boldsymbol{\xi_1}^{\top}\left(\boldsymbol{\alpha}_{\mathcal{I}}-\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\right)\right]=0, \\
\mathbb{E}\left[\left\{\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}\left[\boldsymbol{\xi_1}^{\top}\left(\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}-\boldsymbol{\alpha}_{\mathcal{I}}\right)\right]-\xi_2^2\right\}\boldsymbol{\alpha}_{\mathcal{S}}\right]+\mathbb{E}\left[\left[(\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}+\xi_2)^2\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}+\boldsymbol{\xi_3}\boldsymbol{\xi_3}^{\top}\right]\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\right]=0.
\end{array}\right. \tag{17}
$$

Solving this, we have

$$
\left\{\begin{array}{l}
\boldsymbol{\alpha}_{\mathcal{I}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}+\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}, \\
\mathbb{E}\left[\left\{\left(\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}}\right)^2\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}+\xi_2^2\right\}\boldsymbol{\alpha}_{\mathcal{S}}\right]=\mathbb{E}\left[\left[(\boldsymbol{\xi_1}^{\top}\boldsymbol{\alpha}_{\mathcal{I}})^2+\xi_2^2)\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}+\boldsymbol{\xi_3}\boldsymbol{\xi_3}^{\top}\right]\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\right].
\end{array}\right. \tag{18}
$$

Using the assumption $\mathbb{E}[\xi_i\xi_i^{\top}]=\Sigma_i$,

$$
\left\{\begin{array}{l}
\boldsymbol{\alpha}_{\mathcal{I}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}+\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}, \\
\left(\sigma_2^2-\boldsymbol{\alpha}_{\boldsymbol{\mathcal{S}}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\sigma_2^2\right)\boldsymbol{\alpha}_{\mathcal{S}}=\Sigma_3\widehat{\boldsymbol{\alpha}}_{\mathcal{S}},
\end{array}\right. \tag{19}
$$

(19) suggests our assumption on the noises are mild, as long as one of the noise $\xi_2, \boldsymbol{\xi_3}$ depends on the environment, the regressed spurious feature $\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}$ will depend on the environment. In particular, given that the noise $\boldsymbol{\xi_3}$ are centered i.i.d sub-Gaussian noise, we have

$$
\left\{\begin{array}{l}
\boldsymbol{\alpha}_{\mathcal{I}}=\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}+\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}, \\
\left(\sigma_2^2-\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\sigma_2^2\right)\boldsymbol{\alpha}_{\mathcal{S}}=\sigma_3^2\widehat{\boldsymbol{\alpha}}_{\mathcal{S}},
\end{array}\right. \tag{20}
$$

then we have

$$
\boldsymbol{\alpha}_{\mathcal{S}}^{\top}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}=\frac{\sigma_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2}{\sigma_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2+\sigma_3^2}. \tag{21}
$$

Combining (21) with (20), we have

$$
\left\{\begin{array}{l}
\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}=\frac{\sigma_3^2}{\sigma_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2+\sigma_3^2}\cdot\boldsymbol{\alpha}_{\mathcal{I}}, \\
\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}=\frac{\sigma_2^2}{\sigma_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2+\sigma_3^2}\cdot\boldsymbol{\alpha}_{\mathcal{S}},
\end{array}\right. \tag{22}
$$

Therefore, we have

$$
\frac{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}\|}{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}\|}=\frac{\sigma_3^2}{\sigma_2^2}\cdot\frac{\|\boldsymbol{\alpha}_{\mathcal{I}}\|}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|}.
$$

Consider regressing solely on the invariant correlations $\widehat{y}=\sum_{i\in\mathcal{I}}x_i\widehat{\alpha}_i$ yields

$$
\widehat{\boldsymbol{\alpha}}\in\underset{\alpha_i\in\mathbb{R},i\in\mathcal{I}}{\arg\min}\;\mathbb{E}\left\|y-\boldsymbol{x}_{\mathcal{I}}^{\top}\boldsymbol{\alpha}\right\|^2, \tag{23}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{\alpha}} &= \mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}(\boldsymbol{x}_{\mathcal{I}}^\top \boldsymbol{\alpha} - y)\right] \\
&= \mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}(\boldsymbol{x}_{\mathcal{I}}^\top \boldsymbol{\alpha} - \boldsymbol{x}_{\mathcal{I}}^\top \boldsymbol{\alpha}_{\mathcal{I}} - \xi_2)\right] \\
&\overset{\text{(a)}}{=} \mathbb{E}\left[\boldsymbol{x}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}}^\top(\boldsymbol{\alpha} - \boldsymbol{\alpha}_{\mathcal{I}})\right] = 0.
\end{aligned} \tag{24}
$$

where in (a), we used the independence of $\boldsymbol{\xi_1}$ and ξ_2, and the fact $\boldsymbol{\xi_1}, \xi_2$ are centered. Therefore, $\widehat{\boldsymbol{\alpha}} = [\boldsymbol{\alpha_{\mathcal{I}}}, \mathbf{0}]^\top$. Consider using ERM to solve (6), i.e., using all the data $(x_i, y_i)_{i\in[dK}$, we have

$$
\sigma_2^2 = \mathbb{E}[\xi_2^2] = \mathbb{E}_\eta\left\{\mathbb{E}_{(\boldsymbol{x},y)}\left[\xi_2^2(\eta)\,\middle|\,\eta = i\right]\right\} = \frac{1}{K}\sum_{i=1}^{K}\sigma_2^2(i). \tag{25}
$$

where we assume η satisfying for all $i \in [K]$, there holds

$$
\mathbb{P}(\eta = i) = \frac{1}{K}.
$$

Notice that this is a fair assumption given the data from different distribution is the same. If training data sizes are different, then the distribution of η can be adjusted accordingly. Similarly, there holds

$$
\sigma_3^2 = \frac{1}{K}\sum_{i=1}^{K}\sigma_3^2(i). \tag{26}
$$

Therefore, we have for ERM,

$$
\begin{cases}
\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}} = \frac{\sum_{i=1}^{K}\sigma_3^2(i)}{\sum_{i=1}^{K}\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sum_{i=1}^{K}\sigma_3^2(i)} \cdot \boldsymbol{\alpha}_{\mathcal{I}}, \\
\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}} = \frac{\sum_{i=1}^{K}\sigma_2^2(i)}{\sum_{i=1}^{K}\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sum_{i=1}^{K}\sigma_3^2(i)} \cdot \boldsymbol{\alpha}_{\mathcal{S}}.
\end{cases} \tag{27}
$$

Consider each client locally use their own domain data to solve (6), and then one shot averaging the solutions, we have

$$
\begin{cases}
\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}} = \frac{1}{K}\sum_{i=1}^{K}\frac{\sigma_3^2(i)}{\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sigma_3^2(i)} \cdot \boldsymbol{\alpha}_{\mathcal{I}}, \\
\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}} = \frac{1}{K}\sum_{i=1}^{K}\frac{\sigma_2^2(i)}{\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sigma_3^2(i)} \cdot \boldsymbol{\alpha}_{\mathcal{S}},
\end{cases} \tag{28}
$$

Comparing the solutions of ERM and one-shot averaging, when $\sigma_3(i) = \sigma_3$, we have

$$
\begin{aligned}
\frac{\sum_{i=1}^{K}\sigma_3^2}{\sum_{i=1}^{K}\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sum_{i=1}^{K}\sigma_3^2} &= \frac{K}{\sum_{i=1}^{K}(\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2/\sigma_3^2 + 1)} \\
&\leq \frac{1}{K}\sum_{i=1}^{K}\frac{1}{(\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2/\sigma_3^2 + 1)} \\
&= \frac{1}{K}\sum_{i=1}^{K}\frac{\sigma_3^2}{\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sigma_3^2},
\end{aligned} \tag{29}
$$

where the inequality follows from the fact that harmonic mean is less than the arithmetic mean. Therefore, there holds,

$$
\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}} \leq \widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}}. \tag{30}
$$

Further, we have

$$\begin{aligned}\frac{\sum_{i=1}^K \sigma_2^2(i)}{\sum_{i=1}^K \sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sum_{i=1}^K \sigma_3^2} &= \left(1 - \frac{\sum_{i=1}^K \sigma_3^2}{\sum_{i=1}^K \sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sum_{i=1}^K \sigma_3^2}\right) \cdot \frac{1}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2} \\ &\geq \left(1 - \frac{1}{K}\sum_{i=1}^K \frac{\sigma_3^2}{\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sigma_3^2}\right) \cdot \frac{1}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2} \\ &= \frac{1}{K}\sum_{i=1}^K \frac{\sigma_2^2(i)}{\sigma_2^2(i)\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + \sigma_3^2}, \end{aligned} \tag{31}$$

which suggests

$$\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}} \geq \widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}}. \tag{32}$$

Given that $\sigma_2(i), i \in [K]$ are different, we conclude the above inequality is strict, i.e.,

$$\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}} < \widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}} \quad \text{and} \quad \widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}} > \widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}}. \tag{33}$$

Recall that $\boldsymbol{\alpha}^* := [\boldsymbol{\alpha}_{\mathcal{I}}, \mathbf{0}]^\top$, then (33) implies

$$\|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}} - \boldsymbol{\alpha}_{\mathcal{I}}\|^2 > \|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}} - \boldsymbol{\alpha}_{\mathcal{I}}\|^2 \quad \text{and} \quad \|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}} - \mathbf{0}_{\mathcal{S}}\|^2 > \|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}} - \mathbf{0}_{\mathcal{S}}\|^2. \tag{34}$$

Therefore, we have

$$\begin{aligned}&\|\widehat{\boldsymbol{\alpha}}^{\text{ERM}} - \boldsymbol{\alpha}^*\|^2 \\ &= \|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}} - \boldsymbol{\alpha}_{\mathcal{I}}\|^2 + \|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}} - \mathbf{0}_{\mathcal{S}}\|^2 \\ &> \|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}} - \boldsymbol{\alpha}_{\mathcal{I}}\|^2 + \|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}} - \mathbf{0}_{\mathcal{S}}\|^2 \\ &= \|\widehat{\boldsymbol{\alpha}}^{\text{OSA}} - \boldsymbol{\alpha}^*\|. \end{aligned} \tag{35}$$

In addition, we have

$$\sum_{i=1}^K \frac{1}{K}\frac{\sigma_2^2(i)}{\sigma_3^2} \geq \sum_{i=1}^K \frac{\sigma_2^2(i)}{\sigma_3^2} q_i$$

where $q_i, i \in [K]$ are defined as

$$q_i \triangleq \frac{A_i}{\frac{1}{K}\sum_{k=1}^K A_k}, \qquad \text{where} \qquad A_i \triangleq \frac{1}{\sigma_2^2(i)/\sigma_3^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|^2 + 1}.$$

Therefore, the robust to spurious ratio reads

$$\begin{aligned}\frac{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{ERM}}\|}{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{ERM}}\|} &= \frac{\sigma_3^2}{\frac{1}{K}\sum_{i=1}^K \sigma_2^2(i)} \cdot \frac{\|\boldsymbol{\alpha}_{\mathcal{I}}\|}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|} \\ &= \frac{1}{\frac{1}{K}\sum_{i=1}^K \frac{\sigma_2^2(i)}{\sigma_3^2}} \cdot \frac{\|\boldsymbol{\alpha}_{\mathcal{I}}\|}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|} \\ &\leq \frac{1}{\sum_{i=1}^K \frac{\sigma_2^2(i)}{\sigma_3^2} q_i} \cdot \frac{\|\boldsymbol{\alpha}_{\mathcal{I}}\|}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|} \\ &= \frac{\frac{1}{K}\sum_{i=1}^K A_i}{\frac{1}{K}\sum_{i=1}^K \frac{\sigma_2^2(i)}{\sigma_3^2} A_i} \cdot \frac{\|\boldsymbol{\alpha}_{\mathcal{I}}\|}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|} \\ &= \frac{1}{\frac{1}{K}\sum_{i=1}^K \frac{\sigma_2^2(i)}{\sigma_3^2} \frac{A_i}{\frac{1}{K}\sum_{k=1}^K A_k}} \cdot \frac{\|\boldsymbol{\alpha}_{\mathcal{I}}\|}{\|\boldsymbol{\alpha}_{\mathcal{S}}\|} \\ &= \frac{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{I}}^{\text{OSA}}\|}{\|\widehat{\boldsymbol{\alpha}}_{\mathcal{S}}^{\text{OSA}}\|}. \end{aligned} \tag{36}$$

□

f

B Additional proof that OSA will have better DG accuracy than ERM from linear example

Corollary B.1. *Under the assumptions of Theorem 3.3, then the DG risk of ERM is higher than for OSA:*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{test}}\left[(y-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}}^{ERM})^2\right] > \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{test}}\left[(y-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}}^{OSA})^2\right] . \tag{37}$$

Proof. We first derive the covariances for a single domain:

$$\begin{aligned}
\Sigma_{\mathcal{I},\mathcal{I}} &= \sigma_1^2 I \qquad (38)\\
\Sigma_{\mathcal{S},\mathcal{S}} &= \mathbb{E}[((\boldsymbol{\alpha}_{\mathcal{I}}^T\xi_1+\xi_2)\boldsymbol{\alpha}_{\mathcal{S}}+\xi_3)((\boldsymbol{\alpha}_{\mathcal{I}}^T\xi_1+\xi_2)\boldsymbol{\alpha}_{\mathcal{S}}+\xi_3)^T] \qquad (39)\\
&= \boldsymbol{\alpha}_{\mathcal{I}}^T\mathbb{E}[\xi_1\xi_1^T]\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^T+\mathbb{E}[\xi_2^2]\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^T+\mathbb{E}[\xi_3\xi_3^T] \qquad (40)\\
&= (\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\sigma_1^2+\sigma_2^2)\boldsymbol{\alpha}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}^T+\sigma_3^2 I \qquad (41)\\
\Sigma_{\mathcal{I},\mathcal{S}} &= \mathbb{E}[x_{\mathcal{I}}x_{\mathcal{S}}^T] = \mathbb{E}[\xi_1((\boldsymbol{\alpha}_{\mathcal{I}}^T\xi_1+\xi_2)\boldsymbol{\alpha}_{\mathcal{S}}+\xi_3)^T] \qquad (42)\\
&= \mathbb{E}[\xi_1\xi_1^T\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^T+\xi_1\xi_2\boldsymbol{\alpha}_{\mathcal{S}}^T+\xi_1\xi_3^T] \qquad (43)\\
&= \sigma_1^2\boldsymbol{\alpha}_{\mathcal{I}}\boldsymbol{\alpha}_{\mathcal{S}}^T. \qquad (44)
\end{aligned}$$

We now decompose the test-domain risk where the domain index is not in the training dataset, i.e., $i \notin [C]$, as follows:

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(y-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}})^2\right]\\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(\boldsymbol{x}_{\boldsymbol{\mathcal{I}}}^\top\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}+\xi_2(i)-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}})^2\right]\\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(\boldsymbol{x}^\top\boldsymbol{\alpha}^*+\xi_2(i)-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}})^2\right]\\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(\boldsymbol{x}^\top(\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}}))^2\right]+\sigma_2^2(i)\\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}})^\top\boldsymbol{x}\boldsymbol{x}^\top(\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}})\right]+\sigma_2^2(i)\\
&= (\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}})^\top\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[\boldsymbol{x}\boldsymbol{x}^\top\right](\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}})+\sigma_2^2(i) \qquad (45)\\
&= (\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}})^\top\Sigma_{\boldsymbol{x},\boldsymbol{x}}(i)(\boldsymbol{\alpha}^*-\widehat{\boldsymbol{\alpha}})+\sigma_2^2(i) \qquad (46)\\
&= \begin{bmatrix}(1-\hat{w}_{\mathcal{I}})\boldsymbol{\alpha}_{\mathcal{I}}\\ \hat{w}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}\end{bmatrix}^\top\begin{bmatrix}\Sigma_{\mathcal{I},\mathcal{I}}(i) & \Sigma_{\mathcal{I},\mathcal{S}}(i)\\ \Sigma_{\mathcal{S},\mathcal{I}}(i) & \Sigma_{\mathcal{S},\mathcal{S}}(i)\end{bmatrix}\begin{bmatrix}(1-\hat{w}_{\mathcal{I}})\boldsymbol{\alpha}_{\mathcal{I}}\\ \hat{w}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}\end{bmatrix}+\sigma_2^2(i) \qquad (47)\\
&= \begin{bmatrix}(1-\hat{w}_{\mathcal{I}})\boldsymbol{\alpha}_{\mathcal{I}}\\ \hat{w}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}\end{bmatrix}^\top\begin{bmatrix}\sigma_1^2(i)I_{\mathcal{I}} & \sigma_1^2(i)\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}\boldsymbol{\alpha}_{\boldsymbol{\mathcal{S}}}{}^\top\\ \sigma_1^2(i)\boldsymbol{\alpha}_{\boldsymbol{\mathcal{S}}}\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}{}^\top & \left(\|\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}\|^2\sigma_1^2(i)+\sigma_2^2(i)\right)\boldsymbol{\alpha}_{\boldsymbol{\mathcal{S}}}\boldsymbol{\alpha}_{\boldsymbol{\mathcal{S}}}^\top+\sigma_3^2 I_{\mathcal{S}}\end{bmatrix}\begin{bmatrix}(1-\hat{w}_{\mathcal{I}})\boldsymbol{\alpha}_{\mathcal{I}}\\ \hat{w}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}\end{bmatrix}+\sigma_2^2(i) \qquad (48)\\
&= (1-\hat{w}_{\mathcal{I}})^2\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\sigma_1^2(i)+2\hat{w}_{\mathcal{S}}(1-\hat{w}_{\mathcal{I}})\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^2\sigma_1^2(i)\\
&\quad+\hat{w}_{\mathcal{I}}^2((\|\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}\|^2\sigma_1^2(i)+\sigma_2^2(i))\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^4+\sigma_3^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^2)+\sigma_2^2(i) \qquad (49)
\end{aligned}$$

Note that the algorithm specific terms are only via $\hat{w}_{\mathcal{I}}$ and $\hat{w}_{\mathcal{S}}$, because all other terms are constant w.r.t. to the learned parameters $\widehat{\alpha}$. Finally, from the previous theorem proof ((33)), we know that

$$\begin{aligned}
0 \le (1-\hat{w}_{\mathcal{I}}^{\text{OSA}}) < (1-\hat{w}_{\mathcal{I}}^{\text{ERM}}) \le 1 \qquad (50)\\
0 \le \hat{w}_{\mathcal{S}}^{\text{OSA}} < \hat{w}_{\mathcal{S}}^{\text{ERM}} \le 1\,. \qquad (51)
\end{aligned}$$

Combining these facts with Eqn. 49, we can easily arrive at the result:

$$
\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(y-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}}^{\text{OSA}})^2\right] && (52)\\
&= (1-\hat{w}_{\mathcal{I}}^{\text{OSA}})^2\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\sigma_1^2(i) + 2\hat{w}_{\mathcal{S}}^{\text{OSA}}(1-\hat{w}_{\mathcal{I}}^{\text{OSA}})\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^2\sigma_1^2(i)\\
&\qquad + \left(\hat{w}_{\mathcal{I}}^{\text{OSA}}\right)^2((\|\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}\|^2\sigma_1^2(i)+\sigma_2^2(i))\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^4+\sigma_3^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^2)+\sigma_2^2(i) && (53)\\
&< (1-\hat{w}_{\mathcal{I}}^{\text{ERM}})^2\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\sigma_1^2(i) + 2\hat{w}_{\mathcal{S}}^{\text{ERM}}(1-\hat{w}_{\mathcal{I}}^{\text{ERM}})\|\boldsymbol{\alpha}_{\mathcal{I}}\|_2^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^2\sigma_1^2(i)\\
&\qquad + \left(\hat{w}_{\mathcal{I}}^{\text{ERM}}\right)^2((\|\boldsymbol{\alpha}_{\boldsymbol{\mathcal{I}}}\|^2\sigma_1^2(i)+\sigma_2^2(i))\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^4+\sigma_3^2\|\boldsymbol{\alpha}_{\mathcal{S}}\|_2^2)+\sigma_2^2(i) && (54)\\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{test}}}\left[(y-\boldsymbol{x}^\top\widehat{\boldsymbol{\alpha}}^{\text{ERM}})^2\right]\ . && (55)
\end{aligned}
$$

□

C FedLOE Algorithm Pseudo-Code

We present our training method FedLOE in Algorithm 1.

Algorithm 1 FedLOE: Federated DG via Local Overfitting and Ensembling

```
Input: Training datasets $\{x_c^{\text{train}}, y_c^{\text{train}}\}_{c=1}^C$ and fine-tuning datasets $\{x_c^{\text{tune}}, y_c^{\text{tune}}\}_{c=1}^C$ on each of the $C$
  clients; iterations for stage 1 and 2: $T_1, T_2$; communication indices $\mathcal{T}_1$ and $\mathcal{T}_2$ for stage 1 and 2; local
  computations $E_1$ and $E_2$.
  # Stage 1: Few shot Federated Averaging to Learn $\theta$: (4)
  # Stage 2: Robustifying the Linear Classifier Head
  for $t \in \{T_1+1,\dots,T_1+T_2\}$ do
    If $t \notin \mathcal{T}_2$, # Stage 2.1: Locally overfit classifier heads, distribute to all clients
    {Client} $\forall c, \widehat{\psi}_c^t = \text{Mini-batch SGD}\left(\widehat{\theta}, \widehat{\psi}_c^{t-1}; E_2\right)$
    {Client → Server} $\forall c, \widehat{\psi}_c^t$        {Server → Client} $\left\{\widehat{\psi}_c^t\right\}_{c=1}^C$
    Else if $t \in \mathcal{T}_2$, # Stage 2.2: Robust training
    # Step 1 Server collect predictions on fine tuning dataset
    {Client} $\forall c, \left\{\widehat{y}_{c,c'}^t \triangleq g_{\widehat{\theta}}(x_c^{\text{tune}})\left(\widehat{\psi}_{c'}^t\right)^\top\right\}_{c'\in[C]}$        {Client → Server} $\forall c, \{\widehat{y}_{c,c'}^t\}_{c'\in[C]}$
    # Step 2 Train invariant manager.
    {Server} solves Equation (11) to obtain $\widehat{\phi}^t$        {Server → Client} $\widehat{\psi}^t = \sum_c \widehat{\phi}_c^t\widehat{\psi}_c^t$
  end for
  Return: $(\widehat{\theta}, \widehat{\psi}) = (\widehat{\theta}^{T_1}, \widehat{\psi}^{T_1+T_2})$
```

C.1 Penalty choices in the FL setting

For invariant manager training, in every iteration $t \in \mathcal{T}_2$, recall here (11) for convenience

$$
\widehat{\phi}^t = \operatorname*{arg\,min}_{\phi\in\mathbb{R}^C} \sum_c p(c)\mathbb{E}^{\text{tune}}_{p(\hat{y}^t,y^t|c)}\left[\ell\left(\sum_{c'=1}^C \phi_{c'}\widehat{y}_{c'}^t, y^t\right)\right] + \lambda\cdot r\left(\sum_{c'=1}^C \phi_{c'}\widehat{y}_{c'}^t, y^t\right),
$$

where λ is the penalty parameter and $r(\cdot)$ is a penalty term to induce invariance, and it can be chosen using different methods as in (3). For example, to encourage $\sum_{c=1}^C \phi_j\widehat{y}_{c,j}$ to be simultaneously optimal for different clients domain $c \in [C]$, we choose r_{dg} as r_{irmv}, where it enforces the gradients to be zero Arjovsky et al. (2019),

$$
r_{\text{irm}}(\theta,\psi) \triangleq \sum_c \mathbb{E}_{p_c}\left\|\nabla_{w|w=1.0}\ell\left(\sum_{c=1}^C w\phi_j\widehat{y}_{c,j}, y_c^{\text{tune}}\right)\right\|^2 .
$$

We can also choose r_{dg} as r_{fish}, where it align the gradients as Fish Shi et al. (2021),

$$r_{\text{fish}}(\theta, \psi) \triangleq - \sum_{c \neq c'} \mathbb{E}_{p_c, p_{c'}} \left\langle \nabla_\phi \ell \left(\sum_{c=1}^{C} \phi_j \widehat{y}_{c,j}, y_c^{\text{tune}} \right), \nabla_\phi \ell \left(\sum_{c=1}^{C} \phi_j \widehat{y}_{c',j}, y_{c'}^{\text{tune}} \right) \right\rangle .$$

Further, we can choose r_{dg} as r_{REx}, where it reduces the loss variance to be 0

$$r_{\text{REx}}(\theta, \psi) \triangleq \sum_{c} \mathbb{V}_{p_c} \left[\ell \left(\sum_{j=1}^{C} \phi_j Y_{c,j}, y_c^{\text{tune}} \right) \right] .$$

D Additional Experimental Details

D.1 Dataset

PACS Gulrajani and Lopez-Paz (2020) is an image dataset designed for domain generalization classification, comprising a total of 4 distinct domains: Photo ($1,670$ images), Art Painting ($2,048$ images), Cartoon ($2,344$ images), and Sketch ($3,929$ images). Within each domain, there are 7 categories present. For this dataset, we use each domain as test domain in one run, and we care about the average classification accuracy.

OfficeHome Gulrajani and Lopez-Paz (2020) is also an image dataset designed for domain generalization classification containing 4 domains namely Art ($2,427$ images), Clipart ($4,365$ images), Product ($4,439$ images), and Real-World ($4,357$ images) images. Within each domain, there are 65 categories present, which makes it a harder setting than PACS. Similar to PACS, for this dataset, we use each domain as test domain in one run, and we care about the average classification accuracy.

IWildCam Koh et al. (2021) consists of a diverse collection of wild animals captured by 343 camera traps situated across various natural habitats worldwide. This dataset offers a multi-class classification task, comprising a total of $203,029$ data samples representing 182 distinct animal species. Our primary objective is to achieve high classification accuracy, particularly for rare animal species. Due to this goal, we employ the macro-F1 score as our chosen metric.

D.2 Data Partitioning

In our experiments on the PACS and OfficeHome datasets, we adopt a specific approach. We select one domain as the test domain for each experiment, while the remaining 3 domains serve as the training domains. Initially, we split 5% of the training dataset to create a validation dataset, and another 5% is allocated as an in-domain test dataset.

Subsequently, we divide the remaining 90% of the dataset among the clients, ensuring that each client exclusively contains data from a single domain. For each client, we utilize 90% of its training data as $(\boldsymbol{x}^{\text{train}}, y^{\text{train}})$, while the remaining 10% is designated as $(\boldsymbol{x}^{\text{tune}}, y^{\text{tune}})$. This partitioning strategy enables us to train and evaluate our models on distinct domains while maintaining a balanced and controlled experimental setup.

On the IWildCam dataset, we adhere to the official splits method provided by the wild benchmark Koh et al. (2021) for consistency and comparability. Furthermore, we partition the training data among the clients to ensure that each training domain is present in only one client. This deliberate distribution strategy guarantees maximum domain heterogeneity within the client set.

D.3 Neural Network Structure

In all our experiments, we use ResNet-50 He et al. (2016) as our featurizer, excluding the last fully connected layer. The featurizer produces an output of size 2048. Subsequently, each client maintains their own linear classifier. In stage 1 of our approach, the linear classifiers belonging to each client are averaged with default weight of FedAvg. In stage 2, we take this a step further by averaging the linear classifiers with trained weights.

D.4 Model Selection, Early Stopping and Other Hyperparameters

This section, we report the hyperparameters used in the experiments. Please refer to Table 5.

Table 5: Hyperparameters for our infrequent FedAvg and our FedLOE

Dataset	PACS	OfficeHome	IWildCam
Hyperparameter searching	Leave-one-domain-out-cross-validation		Best performance on held-out validation dataset
Stopping critierion	Fix iteration		Best performance on held-out validation dataset
Batch size	128	128	128
Num of iteration at stage 1	256	256	5120
Stage 1 total communication	8	8	80
Stage 1 optimizer	ADAM	ADAM	ADAM
Stage 1 learning rate	0.00005	0.00003	0.00005
Stage 2 num of iteration (FedLOE only)	8	8	320
Stage 2 total communication (FedLOE only)	4	4	5
Stage 2 optimizer (FedLOE only)	SGD	SGD	SGD
Stage 2 learning rate (FedLOE only)	0.05	0.05	0.05
Stage 2 weight decay (FedLOE only)	0.0005	0.0005	0.0005
IRM λ (FedLOE-IRM only)	100	100	100
Fish meta lr (FedLOE-Fish only)	0.05	0.05	0.01
REx λ (FedLOE-Rex only)	100	100	100

E Code Repository for Reproducibility

Please see the code here: https://anonymous.4open.science/r/fedloe-542D/.