FL Games: A federated learning framework for distribution shifts

Anonymous authors
Paper under double-blind review

Abstract

Federated learning aims to train predictive models for data that is distributed across clients, under the orchestration of a server. However, participating clients typically each hold data from a different distribution, which can yield to catastrophic generalization on data from a different client, which represents a new domain. In this work, we argue that in order to generalize better across non-i.i.d. clients, it is imperative to only learn correlations that are stable and invariant across domains. We propose FL Games, a game-theoretic framework for federated learning that learns causal features that are invariant across clients. While training to achieve the Nash equilibrium, the traditional best response strategy suffers from high-frequency oscillations. We demonstrate that FL Games effectively resolves this challenge and exhibits smooth performance curves. Further, FL Games scales well in the number of clients, requires significantly fewer communication rounds, and is agnostic to device heterogeneity. Through empirical evaluation, we demonstrate that FL Games achieves high out-of-distribution performance on various benchmarks.

1 Introduction

With the rapid advance in technology and growing prevalence of smart devices, Federated Learning (FL) has emerged as an attractive distributed learning paradigm for machine learning models over networks of computers (Kairouz et al., 2019; Li et al., 2020; Bonawitz et al., 2019). In FL, multiple sites with local data, often known as *clients*, collaborate to jointly train a shared model under the orchestration of a central hub called the *server* while keeping their data private.

While FL serves as an attractive alternative to centralized training because the client data does not need to move to the server, there are several challenges associated with its optimization: 1) *statistical heterogeneity* across clients; 2) *massively distributed with limited communication* i.e, a large number of client devices with only a small subset of active clients at any given time (McMahan et al., 2017; Li et al., 2020). One of the most popular algorithms in this setup, Federated Averaging (FedAVG) (McMahan et al., 2017), allows multiple updates at each site prior to communicating updates with the server. While this technique delivers huge communication gains in i.i.d. (independent and identically distributed) setting, its performance on non-i.i.d. clients is an active area of research. As shown by Karimireddy et al. (2020), client heterogeneity has direct implications on the convergence of FedAVG since it introduces a *drift* in the updates of each client with respect to the server model. Yao et al. (2019) describes the same phenomena by arguing that multiple steps of updates at each client cause *gradient bias* towards model aggregation. While recent works including Li et al. (2019); Karimireddy et al. (2020); Yu et al. (2019); Wang et al. (2020); Li et al. (2020); Lin et al. (2020); Li & Wang (2019); Zhu et al. (2021) have tried to address client heterogeneity through constrained gradient optimization and knowledge distillation, most did not tackle the underlying distribution shift. These methods mostly adapt variance reduction techniques such as Stochastic Variance Reduction Gradients (SVRG) (Johnson & Zhang, 2013) to FL. The bias among clients is reduced by constraining the updates of each client with respect to the aggregated gradients of all other clients. These methods can

at best generalize to interpolated domains and fail to extrapolate well, i.e., generalize to new extrapolated domains [1].

According to Pearl (2018) and Schölkopf (2019), in order to build robust systems that generalize well outside of their training environment, learning algorithms should be equipped with causal reasoning tools. Over the past year, there has been a surge in interest in bringing the machinery of causality into machine learning (Arjovsky et al., 2019; Ahuja et al., 2020; Schölkopf, 2019; Ahuja et al., 2021b; Parascandolo et al., 2020; Robey et al., 2021; Krueger et al., 2021; Rahimian & Mehrotra, 2019). All these approaches have focused on learning causal dependencies, which are stable across training environments and further estimate a truly invariant and causal predictor. However, despite their success, they suffer from several key limitations rendering them unsuitable for deployment in a real-world setup.

Since FL typically consists of a large number of clients, it is natural for data at each client to represent different annotation tools, measuring circumstances, experimental environments, and external interventions. Predictive models trained on such datasets could simply rely on spurious correlations to improve their in-distribution i.i.d. performance. Inspired by this idea and by the recent progress in causal machine learning, we draw connections between OOD generalization and robustness across heterogeneous clients in FL. In particular, we consider IRM Games (Ahuja et al., 2020) from the OOD generalization literature since its formulation shows resemblance to the standard FL setup. However, as discussed above, IRM Games too encounters a few fundamental challenges not just specific to FL but also in a generic ML framework.

- **Sequential dependency.** The underlying game theoretic algorithm in IRM Games is inherently sequential. This causes the time complexity of the algorithm to scale linearly with the number of environments.
- **Oscillations.** IRM Games exhibits large oscillations in the performance metrics as the training progresses. The high frequency of these oscillations makes it difficult to define a valid stopping criterion.
- **Convergence speed.** The convergence rate of IRM Games is slow and hence directly impacts systems with speed or communication cost as a primary bottleneck.

In this study, we take a step towards fixing these limitations and addressing the challenge of client heterogeneity under distribution shifts in FL from a causal viewpoint. Specifically, drawing inspiration from IRM Games, we propose Federated Learning Games (FL Games) for learning causal representations which are stable across clients and further enhancing the generalizability of the trained model across unseen testing domains. Apart from effectively learning causal features across clients, our algorithm FL Games also addresses each of the above challenges, hence providing a robust, efficient, and scalable solution. We summarize our main contributions below.

- We propose a new framework called FL Games for learning causal representations that are invariant across clients in a federated learning setup.
- Inspired from the game theory literature, we equip our algorithm to allow parallel updates across clients, further resulting in superior scalability.
- Using ensembles over client's historical actions, we demonstrate that FL Games appreciably smoothens the observed oscillations.
- By increasing the local computation at each client, we show that FL Games exhibits high communication efficiency
- Empirically, we show that the performance of the invariant predictors found by our approach on unseen OOD clients improves significantly over state-of-art prior works.

[1] Similar to Krueger et al. (2021), we define interpolated domains as the domains which fall within the convex hull of training domains and extrapolated domains as those that fall outside of that convex hull.

2 Related works

Federated learning (FL). A major challenge in federated learning is data heterogeneity across clients where the local optima at each client may be far from that of the global optima in the parameter space. This causes a *drift* in the local updates of each client with respect to the server aggregated parameters and further results in slow and unstable convergence (Karimireddy et al., 2020). Recent works have shown FedAVG to be vulnerable in such heterogeneous settings (Zhao et al., 2018). A subset of these works that explicitly constrains gradients for bias removal are called extra gradient methods. Among these methods, FedProx (Li et al., 2020) imposes a quadratic penalty over the distance between server and client parameters which impedes model plasticity. Others use a form of variance reduction techniques such as SVRG (Johnson & Zhang, 2013) to regularize the client updates with respect to the gradients of other clients (Acar et al., 2021; Li et al., 2019; Karimireddy et al., 2020; Liang et al., 2019; Zhang et al., 2020; Konečnỳ et al., 2016). Karimireddy et al. (2020) communicates to the server an additional set of variables known as control variates which contain the estimate of the update direction for both the server and the clients. Using these control variates, the drift at each client is estimated and used to correct the local updates. On the other hand, Acar et al. (2021) estimates the drift for each client on the server and corrects the server updates. By doing so, they avoid using control variates and consume less communication bandwidth. The general strategy for variance reduction methods is to estimate client drift using gradients of other clients, and then constrain the learning objective to reduce the drift. The extra gradient methods are not explicitly optimized to discover causal features, and thus may fail with out-of-distribution examples outside the aggregated distribution of clients.

To date, only two scientific works, Francis et al. (2021); Tenison et al. (2021) have incorporated the learning of invariant predictors in order to achieve strong generalization in FL. The former adapts masked gradients as in Parascandolo et al. (2020) and the latter builds on IRM to exploit invariance and improve leakage protection in FL. While IRM lacks theoretical convergence guarantees, failure modes of Parascandolo et al. (2020) like formation of dead zones and high sensitivity to small perturbations (Shahtalebi et al., 2021) are also issues when it is applied in FL, rendering it unreliable.

Out-of-distribution (OOD) generalization. Generalization under distributional shift is one of the major challenges faced by machine learning systems, limiting their application in the real world. Recent research including Arjovsky et al. (2019); Ahuja et al. (2020); Schölkopf (2019); Ahuja et al. (2021b); Parascandolo et al. (2020); Robey et al. (2021); Krueger et al. (2021); Rahimian & Mehrotra (2019); Xie et al. (2020); Yao et al. (2022); Ahuja et al. (2021a) have tried to address this challenge by proposing alternative objectives for training mechanisms that are invariant across training environments. IRM (Arjovsky et al., 2019) proposes finding a representation $\phi(X)$ that has good prediction abilities and also elicits an invariant predictor across environments. Works like Krueger et al. (2021); Xie et al. (2020) propose a penalty that is a function of variance of training risks. Ahuja et al. (2020) reformulates IRM as finding the Nash equilibrium of an ensemble game played among environments. Mahajan et al. (2021) argues that learning invariant representations for inputs derived from the same object. Recently, Robey et al. (2021) proposed Model-Based Domain Generalization, which enforces invariance to the underlying transformations of data. Another line of work (Rosenfeld et al., 2020; Kamath et al., 2021; Ahuja et al., 2021a) has theoretically analyzed failures of IRM.

3 Background

3.1 Federated Learning

Federated learning methods involve a cloud server coordinating among multiple client devices to jointly train a global model without sharing data across clients. Denote $\mathcal{S}$ to be the set of client devices where $|\mathcal{S}| = m$. Let N_k denote the number of data samples at client device k, and $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ as it's labelled dataset. Mathematically, the objective of a FL is to approximately minimize the global loss

$$F(w) := \sum_{k=1}^{m} \frac{1}{m} \sum_{i=1}^{N_k} \frac{1}{N_k} \ell(x_i^k, y_i^k, w), \tag{1}$$

where ℓ is the loss function and w is the model parameter.

One of the most popular methods in FL is Federated Averaging (FedAVG), where each client performs E local-updates before communicating its weights with the server. FedAVG becomes equivalent to FedSGD for $E = 1$ wherein weights are communicated after every local update. For each client device k, FedAVG initializes it's corresponding device model w_k^0. Consequently, in round t, each device undergoes a local update on its dataset according to the following $w_k^{t+1} \leftarrow w_k^t - \eta^k \nabla \ell(\mathcal{B}_i^k, w_k^t)$, $\forall B_i^k \subseteq \mathcal{D}_k$ where B_i^k is a sampled mini-batch from D_k at the ith step. All clients' model parameters $\{w_k^{t+1}\}_{k \in \mathcal{S}}$ are then sent to the cloud server which performs a weight average to update the global model w^{t+1} as

$$w^{t+1} \leftarrow \frac{1}{|\mathcal{S}|} \sum_{k \in \mathcal{S}} \frac{N_k}{N} w_k^t,$$

where N_k is the number of samples at client device k and N is the total number of samples from all clients ($N = \sum_{k=1}^{m} N_k$). This aggregated global model is broadcasted with all clients and the above process is repeated till convergence.

3.2 Invariant Risk Minimization Games

Consider a setup comprising datasets $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ from multiple training environments, $k \in \mathcal{E}_{tr}$ with N_k being the number of samples at environment k. The aim of Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) is to jointly train across all these environments and learn a robust set of parameters θ that generalize well to unseen (test) environments $\mathcal{E}_{all} \supset \mathcal{E}_{tr}$. The risk of a predictor f at each environment can be mathematically represented as $R^k(w \circ \phi) = E_{(x,y) \sim D_k} f_\theta((w \circ \phi); x, y)$ where $f_\theta = w \circ \phi$ is the composition of a feature extraction function $\phi : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^d$ and a predictor network, $w : \mathcal{Z} \to \mathbb{R}^k$ where k is the number of classes.

Empirical Risk Minimization (ERM) aims to minimize the average of the losses across all environments. Mathematically, the ERM objective can be formulated as $R^{\text{ERM}}(\theta) = E_{(x,y) \sim \cup_{e \in E} D_e} f_e(\theta; x, y)$. As shown in Arjovsky et al. (2019), ERM fails to generalize to novel domains, which have significant distribution shift as compared to the training environments.

Invariant Risk Minimization (IRM) instead aims to capture invariant representations ϕ such that the optimal predictor w given ϕ is the same across all training environments. Mathematically, they formulate the objective as a bi-level optimization problem

$$\min_{\phi \in \mathcal{H}_\phi, w \in \mathcal{H}_w} \sum_{k \in \mathcal{E}_{tr}} R^k(w \circ \phi) \text{ s.t. } w \in \arg\min_{w \in \mathcal{H}_w} R^k(w \circ \phi), \forall k \in \mathcal{E}_{tr} \tag{2}$$

where $\mathcal{H}_\phi$, $\mathcal{H}_w$ are the hypothesis sets for feature extractors and predictors, respectively. Since each constraint calls an inner optimization routine, IRM approximates this challenging optimization problem by fixing the predictor w to a scalar.

Invariant Risk Minimization Games (IRM Games) is an algorithm based on an alternate game theoretic reformulation of the optimization objective in equation 2. It endows each environment with its own predictor $w^k \in \mathcal{H}_w$ and aims to train an ensemble model $w^{av}(z) = \frac{1}{|\mathcal{E}_{tr}|} \sum_{k=1}^{|\mathcal{E}_{tr}|} w^k(z)$ for each $z \in \mathcal{Z}$ s.t. w^{av} satisfies the following optimization problem

$$\min_{w^{av},\phi\in\mathcal{H}_\phi} \sum_k R^k(w^{av}\circ\phi)$$
$$\text{s.t. } w^k \in \arg\min_{w'_k\in\mathcal{H}_w} R^k\left(\frac{1}{|\mathcal{E}_{tr}|}(w'_k + \sum_{q\in\mathcal{E}_{tr},q\neq k} w^q)\circ\phi\right), \forall k\in\mathcal{E}_{tr} \tag{3}$$

The constraint in equation 3 is equivalent to the Nash equilibrium of a game with each environment k as a player with action w^k, playing to maximize its utility $R^k(w^{av},\phi)$. While there are different algorithms in the game-theoretic literature to compute the Nash equilibrium, the resultant non-zero sum continuous game is solved using the best response dynamics (BRD) with clockwise updates and is referred to as V-IRM Games. In this training paradigm, players take turns according to a fixed cyclic order and only one player is allowed to change its action at any given time (for more details, refer to the supplement). Fixing ϕ to an identity map in V-IRM Games is also shown to be very effective and is called F-IRM Games.

4 Federated Learning Games (FL Games)

OOD generalization is often typified using the notion of data-generating environments. Arjovsky et al. (2019) formalizes an environment as a data generating distribution representing a particular location, time, context, circumstances and so forth. Distinct environments are assumed to share some overlapping causal features such that the corresponding causal mechanisms are invariant across environments, but the distribution of some of the causal variables may vary. Spurious variables denote the unstable features which vary across environments. This concept of data-generating environments can be related to FL by considering each client as producing data generated from a different environment. However despite this equivalence, existing OOD generalization techniques can't be directly applied to FL. Apart from the FL-specific challenges, these approaches also suffer from several key limitations in non-FL domains (Rosenfeld et al., 2020; Nagarajan et al., 2020), further rendering them unfit for practical deployment. As a consequence, developing causal inference models for FL which are inspired from invariant prediction in OOD generalization, are bound to inherit the failures of the latter.

In this work, we consider one such popular OOD generalization technique, IRM Games as it formulation bears close resemblance to a standard FL setup. However, as discussed, IRM Games too suffers from various challenges, which impede its deployment in a generic ML framework, specifically in FL. In the following section, we elaborate on each of these limitations and discuss the corresponding modifications required to overcome them. Further, inspired from the game theoretic formulation of IRM Games, we propose FL Games which forfeits it's failures and can recover the causal mechanisms of the targets, while also providing robustness to changes in the distribution across clients.

4.1 Challenges in Federated Learning

Data Privacy. Consider a FL system with m client devices, $\mathcal{S} = \{1, 2, ..., m\}$. Let N_k denote the number of data samples at client device k, and $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{N_k}$ as it's labelled dataset. The constraint of each environment in IRM Games can be used to formulate the local objective of each client. In particular, each client $k \in \mathcal{S}$ now serves as a player, competing to learn $w^k \in \mathcal{H}_w$ by optimizing its local objective, i.e.

$$w^k \in \arg\min_{w'_k\in\mathcal{H}_w} R^k\left(\frac{1}{|\mathcal{S}|}(w'_k + \sum_{\substack{q\in\mathcal{E}_{tr}\\ q\neq k}} w^q)\circ\phi\right), \forall w'_k\in\mathcal{H}_w \tag{4}$$

However, the upper-level objective of IRM Games requires optimization over the dataset pooled together from all environments. Centrally hosting the data at the server or sharing it across clients contradicts the objective of FL. Hence, we propose using FedSGD (McMahan et al., 2017) to optimize ϕ. Specifically, each client k computes and broadcasts

$$g^k = \nabla R^k\Big(\frac{1}{|\mathcal{S}|}(w'_k + \sum_{q\neq k} w^k)\circ\phi\Big)$$

to the server, which then aggregates these gradients and applies the update rule $\phi_{t+1} = \phi_t - \eta \sum_{k \in \mathcal{S}} \frac{N_k}{N} g^k$ where g^k is computed over $C\%$ batches locally and $N = \sum_{k \in S} N_k$.

Similar to IRM Games, we call this approach V-FL Games. The variant with $\phi = I$ is called F-FL Games.

Sequential dependency As discussed, IRM Games poses IRM objective as finding the Nash equilibrium of an ensemble game across environments and adopts the classic best response dynamics (BRD) algorithm to compute it. This approach is based on playing clockwise sequences wherein players take turns in a fixed cyclic order, with only one player being allowed to change its action at any given time t (Details in the supplement). In order to choose its optimal action for the first time, the last scheduled player N has to wait for all the remaining players from $1, 2, ...N-1$ to play their strategies. This linear scaling of time complexity with the number of players poses a major challenge solving the game in FL.

By definition, in classic BRD, the best responses of any player determines the best responses of the remaining players. Thus, in a distributed learning paradigm, the best responses of each client (player) need to be transmitted to all the other clients. This is infeasible from a practical standpoint as clients are usually based on slow or expensive connections and message transmissions can frequently get delayed or result in information loss. As shown on lines 20 and 26(green) of Algorithm 1, we modify the classic BRD algorithm by allowing simultaneous updates at any given round t. However, a client now best responds to the optimal actions played by it's opponent clients in round $t-1$ instead of t. We refer to this approximation of BRD in F-FL Games and V-FL Games as *parallelized* F-FL Games and, *parallelized* V-FL Games respectively.

Oscillations. As demonstrated in Ahuja et al. (2020), when a neural network is trained using the IRM Games objective (equation 3), the training accuracy initially stabilizes at a high value and eventually starts to oscillate. The setup, over which these observations are made, involves two training environments with varying degrees of spurious correlation. The environments are constructed so that the degree of correlation of color with the target label is very high. The explanation for these oscillations attributes to the significant difference among the data of the training environments. In particular, after a few steps of training, the individual model of the environment with higher spurious correlation (say $\mathcal{E}_1$) is positively correlated with the color while the other is negatively correlated. When it is the turn of former environment to play its optimal strategy, it tries to exploit the spurious correlation in its data and increase the weights of the neurons which are indicative of color. On the contrary, the latter tries to decrease the weights of features associated with color since the errors that backpropogate are computed over the data for which exploiting spurious correlation does not work (say $\mathcal{E}_2$). This continuous swing and sway among individual models results in oscillations.

Despite the promising results, with model's performance metrics oscillating to and from at each step, defining a reasonable stopping criterion becomes challenging. As shown in various game theoretic literature including Herings & Predtetchinski (2017); Barron et al. (2010); Fudenberg et al. (1998); Ge et al. (2018), BRD can often oscillate. Computing the Nash equilibrium for general games is non-trivial and is only possible for a certain class of games (e.g., concave games) (Zhou et al., 2017). Thus, rather than alleviating oscillations completely, we propose solutions to reduce them significantly to better target valid stopping points. We propose a two-way ensemble approach wherein apart from maintaining an ensemble across clients (w^{av}), each client k also responds to the ensemble of historical models (memory) of its opponents. Intuitively, a moving ensemble over the historical models acts as a smoothing filter, which helps to reduce drastic variations in strategies.

Based on the above motivation, we reformulate the optimization objective of each client (Equation 4) to adapt the two-way ensemble learning mechanism (refer to line 4(red) in Algorithm 1). Formally, we maintain queues (a.k.a. buffer) at each client which store its historically played actions. In each iteration, a client best responds to a uniform distribution over the past strategies of its opponents. The global objective at the server remains unchanged. Mathematically, the new local objective of each client $k \in \mathcal{E}_{tr}$ can be stated as

$$w^k \in \arg\min_{w'_k \in \mathcal{H}_w} R^k \left(\frac{1}{|\mathcal{S}|} (w'_k + \sum_{\substack{q \in \mathcal{S} \\ q \neq k}} w^q + \sum_{\substack{p \in \mathcal{S} \\ p \neq k}} \frac{1}{|\mathcal{B}_p|} \sum_{j=1}^{|\mathcal{B}_p|} w_j^p) \circ \phi \right) \tag{5}$$

where $\mathcal{B}_q$ denotes the buffer at client q and w_j^q denotes the jth historical model of client q. We use the same buffer size for all clients. Moreover, as the buffer reaches its capacity, it is renewed based on first in first out (FIFO) manner. Note that this approach doesn't result in any communication overhead since a running sum over historical strategies can be calculated in $\mathcal{O}(1)$ time by maintaining a prefix sum. This variant is called F-FL Games (Smooth) or V-FL Games (Smooth) based on the constraint on ϕ.

Algorithm 1 *Parallelized* FL Games (Smooth+Fast)

Notations: $\mathcal{S}$ is the set of N clients; $\mathcal{B}_k$ and $\mathcal{P}_k$ denote the buffer and information set containing copies of $\mathcal{B}_i, \forall i \neq k \in \mathcal{S}$ at client, k respectively.
PredictorUpdate(k):
  /* Two-way ensemble game to update predictor at each client k */
  $w_k \leftarrow \text{SGD}\left[\ell_k \left\{ \frac{1}{|\mathcal{E}_{tr}|} (w'_k + \sum_{\substack{q \in \mathcal{E}_{tr} \\ q \neq k}} w^q + \sum_{\substack{p \in \mathcal{E}_{tr} \\ p \neq k}} \frac{1}{|\mathcal{B}_p|} \sum_{j=1}^{|\mathcal{B}_p|} w_j^p) \circ \phi \right\}\right]$
  Insert w_k to $\mathcal{B}_k$, discard oldest model in $\mathcal{B}_k$ if full
  return w_k
RepresentationUpdate(k):
  /* Gradient Descent (GD) over entire local dataset at client k */
  for every batch $b \in \mathcal{B}$ **do**
    Compute $\nabla \ell_k(w_{\text{cur}}^{\text{av}} \circ \phi_{\text{cur}}; b)$; Add in $\nabla \phi_k$
  return $\nabla \phi_k$
Server executes:
  Initialize $w_k, \forall k \in \mathcal{S}$ and ϕ
  while round $\leq$ max-round **do**
    /* Update representation ϕ at even round parity */
    if round is even **then**
      if Fixed-Phi **then**
        $\phi_{\text{cur}} = I$
      if Variable-Phi **then**
        for each client $k \in \mathcal{S}$ in parallel **do**
          $\nabla \phi_k = \text{RepresentationUpdate(k)}$
        /* Update representation ϕ */
        $\phi_{\text{next}} = \phi_{\text{cur}} - \eta \left(\sum_{k \in \mathcal{S}} \frac{N_k}{\sum_{j \in \mathcal{S}} N_j} \nabla \phi_k \right)$
        $\phi_{\text{cur}} = \phi_{\text{next}}$
    else
      for each client $k \in \mathcal{S}$ in parallel **do**
        $w_k \leftarrow \text{PredictorUpdate(k)}$
      /* Client k updates its information set $\mathcal{P}_k$ by updating copies of predictors of other clients */
      Communicate $\forall k, \mathcal{P}_k \leftarrow \{w_i, \forall i \neq k \in S\}$
    round $\leftarrow$ round $+ 1$
    $w_{\text{curr}}^{\text{av}} = \frac{1}{N} \sum_{k \in \mathcal{S}} w_{\text{curr}}^k$

Convergence speed. As discussed, FL Games has two variants: F-FL Games and, V-FL Games with the former being an approximation of the latter ($\phi = I$). While both the approaches exhibit superior performance on a variety of benchmarks, the latter has shown its success in a variety of large scale tasks like language modeling (Peyrard et al., 2021). Despite being theoretically grounded, V-IRM Games suffers from slower convergence due to an additional round for optimization of ϕ.

Typically, clients (e.g. mobile devices) posses fast processors and computational resources and have datasets that are much smaller compared to the total dataset size. Hence, utilizing additional local computation is essentially free compared to communicating with the server. To improve the efficiency of our algorithm, we propose replacing the stochastic gradient descent (SGD) over ϕ by a full-batch gradient descent (GD) (line 9 (yellow) of Algorithm 1). This allows ϕ to be updated according to gradients accumulated across the entire dataset, as opposed to gradient step over a mini-batch. Intuitively, now at each gradient step, the resultant ϕ takes large steps in the direction of its global optimum, resulting in fast and stable training. Note that the classifiers at each client are still updated over one mini-batch. This variant of FL Games is referred to V-FL Games (Fast).

5 Experiments and Results

5.1 Datasets

In Ahuja et al. (2020), IRM Games was tested over a variety of benchmarks which were synthetically constructed to incorporate color as a spurious feature. These included Colored MNIST, Colored Fashion MNIST and Colored Dsprites dataset. We utilize the same datasets for our experiments. Additionally, we create another benchmark, Spurious CIFAR10 with a data generating process resembling that of Colored MNIST. In this dataset, instead of coloring the images to establish spurious correlation, we add small black patches at various locations in the image. These locations are spuriously correlated with the label. Details on each of the datasets can be found in the supplement. In all the results, we report the mean performance of various baselines over 5 runs. The performance of oracle on each of these datasets is 75% for train and test sets.

Terminologies: In the following analysis, the terms 'Sequential' and 'Parallel' denote BRD with clockwise playing sequences and simultaneous updates respectively (Lines 20 and 26 of Algorithm 1). We use FL Games as an umbrella term that constitutes all the discussed algorithmic modifications. F-FL Games and V-FL Games refer to the privacy preserving variants of FL Games. The approach used to smoothen out the oscillations (Line 4 of Algorithm 1) is denoted by F-FL Games (Smooth) or V-FL Games (Smooth) depending on the constraint on ϕ. The fast variant with high convergence speed is typified as V-FL Games (Smooth+Fast) (Line 9 of Algorithm 1).

We compare these algorithmic variants across fixed and variable ϕ separately as shown in the Table **??**. Clearly, across all benchmarks, the FL baselines FedSGD, FedAVG (McMahan et al., 2017), FedBN (Li et al., 2021) and FedProx (Li et al., 2020) are unable to generalize to the test set, with FedBN exhibiting superior performance compared to the others. Intuitively, these approaches latch onto the spurious features to make prediction, hence leading to poor generalization over novel clients. Below, we study the performance of FL Games and it's variants individually across the four benchmark datasets.

Colored MNIST (Table 1). We observe that our baseline approach i.e. sequential F-FL Games and V-FL Games achieve 66.56 ± 1.58 and 63.78 ± 1.58 percent testing accuracy, respectively. F-FL Games (Smooth) (Parallel) achieves the highest testing accuracy i.e 67.21 ± 2.98 amongst the fixed ($\phi = I$) variants of FL Games. Similarly, V-FL Games (Parallel) achieves the highest testing accuracy across the variable variants of FL Games i.e. 68.34 ± 5.24 percent. Clearly, all modifications in FL Games individually achieve high testing accuracy, hence eliminating the spurious correlations unlike state-of-the-art FL techniques.

Colored Fashion MNIST (Table 1). Similar to the results on Colored MNIST, all modifications in FL Games achieve high testing accuracy. F-FL Games (Smooth) and V-FL Games achieve the highest test accuracy of 71.81 ± 1.60 and 69.90 ± 1.31 percent across the fixed and the variable variants respectively. While the former was solely designed to smoothen the performance curves, it gives an additional benefit of higher OOD performance.

Spurious CIFAR10 (Table 2). F-FL Games (Smooth) (Parallel) and V-FL Games (Smooth+Fast) achieve the highest testing accuracy of 54.71 ± 2.13 and 50.94 ± 3.28 percent in the fixed and the variable categories respectively. Similar to the results on other datasets, all variants within FL Games achieve high testing accuracy.

Table 1: Comparison of methods in terms of training and testing accuracy (mean ± std deviation) across Colored MNIST and Colored Fashion MNIST. 'Seq.' and 'Par.' are abbreviations for sequential and parallel respectively.

			Colored MNIST		Colored Fashion MNIST	
		Algorithm	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
Baselines		FedSGD	84.88 ± 0.16	10.45 ± 0.60	83.49 ± 1.22	20.13 ± 8.06
		FedAVG	84.45 ± 2.69	12.52 ± 4.34	86.23 ± 0.63	13.33 ± 2.07
		FedBN	99.75 ± 0.11	47.16 ± 3.76	99.79 ± 0.17	41.24 ± 1.87
		FedPROX	99.56 ± 0.38	29.31 ± 0.89	99.87 ± 0.12	29.41 ± 0.36
Fixed	Seq.	F-FL Games	55.76 ± 2.03	66.56 ± 1.58	75.13 ± 1.38	68.40 ± 1.83
		F-FL Games (Smooth)	62.83 ± 5.06	66.83 ± 1.83	75.18 ± 0.37	**71.81 ± 1.60**
	Par.	F-FL Games	58.03 ± 6.22	67.14 ± 2.95	71.71 ± 8.23	69.73 ± 2.12
		F-FL Games (Smooth)	61.07 ± 1.71	**67.21 ± 2.98**	72.81 ± 4.51	71.36 ± 4.19
Variable	Seq.	V-FL Games	56.40 ± 0.03	63.78 ± 1.58	69.90 ± 4.56	**69.90 ± 1.31**
		V-FL Games (Smooth+Fast)	61.03 ± 3.11	65.81 ± 3.28	75.10 ± 0.48	69.85 ± 1.22
	Par.	V-FL Games	52.89 ± 8.03	**68.34 ± 5.24**	66.33 ± 9.39	69.85 ± 3.42
		V-FL Games (Smooth+Fast)	63.11 ± 3.02	65.73 ± 1.53	71.89 ± 5.58	69.41 ± 5.49
		Optimal	75	75	75	75

Table 2: Comparison of methods in terms of training and testing accuracy (mean ± std deviation) across Spurious CIFAR10 and Colored Dsprites. 'Seq.' and 'Par.' are abbreviations for sequential and parallel respectively.

			Spurious CIFAR10		Colored Dsprites	
		Algorithm	Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
Baselines		FedSGD	84.79 ± 0.17	12.57 ± 0.55	99.15 ± 1.10	24.12 ± 2.00
		FedAVG	85.41 ± 1.45	13.11 ± 1.82	99.21 ± 1.35	22.56 ± 2.34
		FedBN	95.24 ± 2.34	25.16 ± 4.06	98.09 ± 1.45	25.19 ± 1.78
		FedPROX	99.67 ± 0.13	24.76 ± 2.67	85.17 ± 1.95	11.12 ± 0.99
Fixed	Seq.	F-FL Games	50.36 ± 2.78	45.36 ± 4.33	53.98 ± 3.67	52.89 ± 4.41
		F-FL Games (Smooth)	64.02 ± 2.08	45.54 ± 1.04	52.87 ± 3.30	61.45 ± 7.11
	Par.	F-FL Games	55.06 ± 2.04	52.07 ± 1.60	52.88 ± 2.78	56.50 ± 6.23
		F-FL Games (Smooth)	56.98 ± 4.09	**54.71 ± 2.13**	53.65 ± 2.11	**62.76 ± 5.97**
Variable	Seq.	V-FL Games	61.72 ± 7.39	46.07 ± 6.01	51.36 ± 5.32	62.84 ± 7.20
		V-FL Games (Smooth+Fast)	50.37 ± 4.97	**50.94 ± 3.28**	51.55 ± 3.20	68.23 ± 4.56
	Par.	V-FL Games	50.41 ± 3.31	50.43 ± 3.04	53.56 ± 4.91	65.87 ± 6.84
		V-FL Games (Smooth+Fast)	45.83 ± 2.44	49.89 ±5.66	54.25 ± 2.05	**68.91 ± 6.47**
		Optimal	75	75	75	75

Colored Dsprites (Table 2). F-FL Games (Smooth) (Parallel) and V-FL Games (Smooth+Fast) achieve the highest testing accuracy of 62.76 ± 5.97 and 68.91 ± 6.47 percent across the fixed and variable variants of FL Games. Further, we discover that the mean performance of V-FL Games is superior to all algorithms with fixed representation ($\phi = I$). This accentuates the importance of V-FL Games over F-FL Games, especially over complex and larger datasets where learning ϕ becomes imperative.

In all the above experiments, both of our end approaches: *parallelized V-FL Games (Smooth+Fast)* and *parallelized* F-FL Games (Smooth) are able to perform better than or at par with the other variants. *These algorithms were primarily designed to overcome the challenges faced by causal FL systems while retaining their original ability to learn causal features.* Hence, the benefits provided by these approaches in terms of 1) robust predictions; 2) scalability; 3) fewer oscillations and 4) fast convergence are not at cost of performance. While the former is demonstrated by Tables 1 and 2, the latter three are detailed in Section 5.3.

5.2 Interpretation of Learned Features

In order to explain the predictions of our parametric model, we use LIME (Ribeiro et al., 2016) to learn an interpretable model locally around each prediction. Specifically, using LIME we construct an interpretable map of our input image whereby all the pixels relevant for prediction are denotes by 1 and the others as 0.

Figure 1 shows LIME masks for FedAVG (ii) and our method (iii) on test image from the COLORED MNIST dataset (i). Clearly, the former focuses solely on the background to make the model prediction. However, the latter uses causal features in the image (like shape, stroke, edges and curves) along with some pixels from the background (noise) to make predictions. This demonstrates the reasoning behind robust generalization of FL GAMES as opposed to state-of-the-art FL techniques.

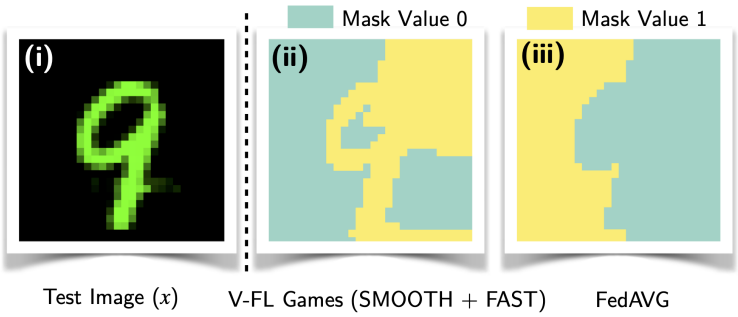


Figure 1: (i) Test image used as input to the interpretable model; (ii) LIME mask corresponding to model prediction using our method V-FL GAMES (SMOOTH+FAST); (iii) LIME mask corresponding to model prediction using FEDAVG;

5.3 Ablation Analysis

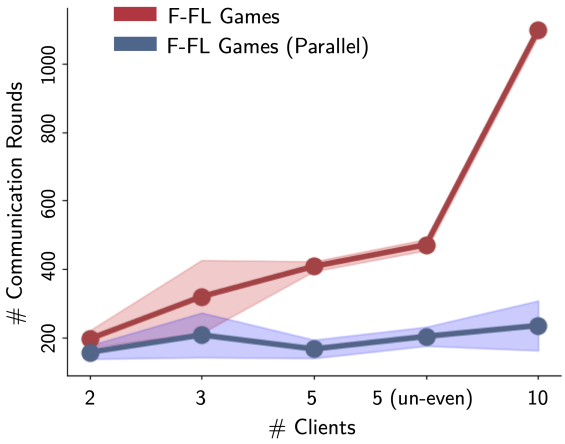
In this section, we analyze the effect of each of our algorithmic modifications using illustrative figures and computational experiments on the COLORED MNIST dataset. The results on other datasets are similar and are discussed in the supplement.

5.3.1 Effect of Simultaneous BRD

We examine the effect of replacing the classic best response dynamics as in Ahuja et al. (2020) with the simultaneous best response dynamics. For the same, we use a more practical environment: (a) more clients are involved, and (b) each client has fewer data. Similar to Choe et al. (2020), we extended the COLORED MNIST dataset by varying the number of clients between 2 and 10. For each setup, we vary the degree of spurious correlation between 70% and 90%) for training clients and merely 10% in the testing set. A more detailed discussion of the dataset is provided in the supplement. For F-FL GAMES, it can be observed from Figure 2(a), as the number of clients in the FL system increase, there is a sharp increase in the number of communication rounds required to reach equilibrium. However, the same doesn't hold true for *parallelized* F-FL GAMES. Further, *parallelized* F-FL GAMES is able to reach a comparable or higher test accuracy as compared to F-FL GAMES with significantly lower communication rounds (refer to Figure 2(b)).

Table 3: COLORED MNIST: Comparison of methods in terms of mean number of rounds required to reach equilibrium.

		ALGORITHM	# Rounds
Fixed	Seq.	F-FL GAMES	224.0
		F-FL GAMES (SMOOTH)	175.0 (1.3×)
	Par.	F-FL GAMES	195.4 (1.1×)
		F-FL GAMES (SMOOTH)	**92.2** (2.4×)
Variable	Seq.	V-FL GAMES	544.4
		V-FL GAMES (SMOOTH+FAST)	292.6 (1.9×)
	Par.	V-FL GAMES	499.1 (1.1×)
		V-FL GAMES (SMOOTH+FAST)	**225.5** (2.4×)



(a)

Type	# clients	Train Accuracy	Test Accuracy
Sequential	2	53.81 ± 4.14	65.68 ± 1.89
	3	54.9 ± 4.37	66.33 ± 1.24
	5	57.20 ± 2.07	66.53 ± 0.55
	5 (uneven)	58.09 ± 2.21	65.30 ± 2.08
	10	59.39 ± 1.41	66.57 ± 1.02
Parallel	2	57.95 ± 3.46	66.57 ± 2.99
	3	59.67 ± 5.46	65.35 ± 3.73
	5	61.82 ± 4.29	65.53 ± 3.85
	5 (uneven)	56.96 ± 5.61	66.15 ± 3.95
	10	55.24 ± 2.88	67.49 ± 3.02

(b)

Figure 2: Colored MNIST: (a) Number of communication rounds required to achieve to achieve Nash equilibrium versus the number of clients in the FL setup; (b) Comparison of F-FL Games and F-FL Games (Parallel) with increasing clients in terms of training and testing accuracy (mean ± std deviation).

5.3.2 Effect of a memory ensemble

As shown in Figure 3(left), compared to F-FL Games, F-FL Games (Smooth) reduces the oscillations significantly. In particular, while in the former, performance metrics oscillate at each step, the oscillations in the later are observed after an interval of roughly 50 rounds. Further, F-FL Games (Smooth) seems to envelop the performance curves of F-FL Games. As a result, apart from reducing the frequency of oscillations, F-FL Games (Smooth) also achieves higher testing accuracy compared to F-FL Games. The observations are consistent across the *parallelized* variants. Further, as observed from Table 3, F-FL Games (Smooth) and *parallelized* F-FL Games (Smooth) significantly reduce the number of communication rounds required to reach equilibrium, further underscoring the efficacy of our proposed methodology.

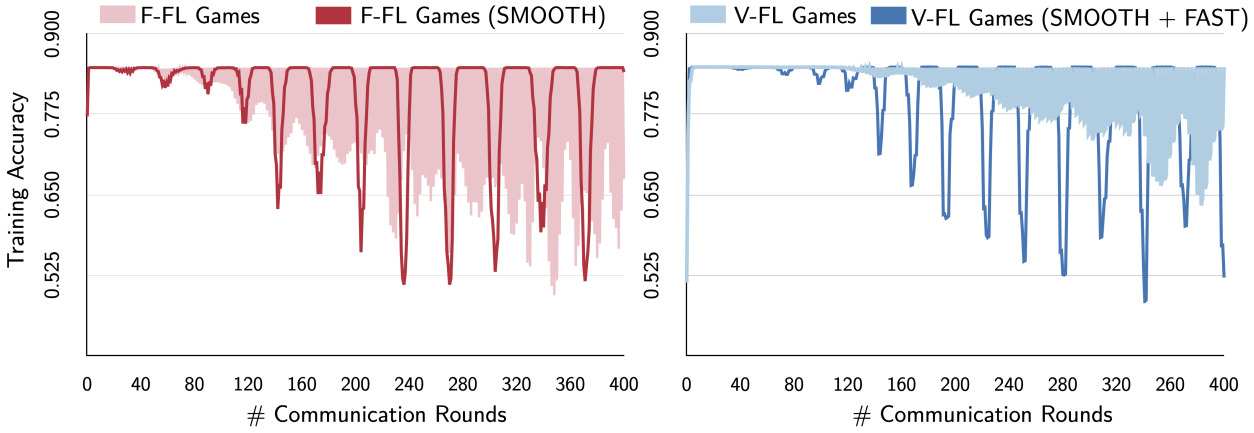


Figure 3: Colored MNIST: Training accuracy of (left) F-FL Games and F-FL Games (Smooth) for a buffer size of 5; (right) V-FL Games and V-FL Games (Smooth + Fast) with buffer size as 5 versus the number of communication rounds

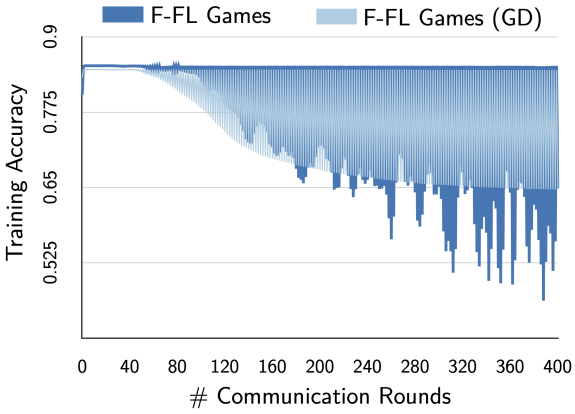
5.3.3 Effect of using Gradient Descent (GD) for ϕ

Communication costs are the principal constraints in FL setup. Edge devices like mobile phones and sensor are bandwidth constrained and require more power for transmission and reception as compared to remote computation. As observed from Figure 3(right), V-FL Games (Smooth+Fast) is able to achieve signif-

icantly higher testing accuracy in fewer communication rounds as compared to V-FL Games. Consistent results are also reported in Table 3, where both sequential and parallel variants of V-FL Games (Smooth +Fast) result in a significant improvement ($\sim 2\times$) in the number of rounds required.

5.3.4 Effect of exact best response

FedAVG provides the flexibility to train communication efficient and high quality models by allowing more local computation at each client. This is particularly detrimental in scenarios with poor network connectivity, wherein communicating at every short time span is infeasible.



(a)

Local steps (in %)	Train Accuracy	Test Accuracy
1.71%	53.04 ± 1.85	65.05 ± 1.60
4.27%	55.12 ± 4.76	61.37 ± 5.02
6.84%	53.46 ± 3.66	61.64 ± 3.67
8.55%	54.13 ± 5.2	61.07 ± 5.89
19.66%	54.38 ± 6.93	59.17 ± 7.37
29.91%	53.86 ± 8.14	58.40 ± 7.19
49.57%	53.59 ± 8.56	59.22 ± 6.05
70.09%	54.25 ± 8.21	58.66 ± 7.09
80.34%	53.14 ± 9.86	57.93 ± 6.23
89.74%	54.61 ± 7.32	57.76 ± 6.54
100.00%	54.56 ± 6.56	57.02 ± 5.17

(b)

Figure 4: Colored MNIST: (a) Effect on Training accuracy of doing a gradient descent on each client for updating the predictor versus the standard training paradigm i.e. F-FL Games; (b) Impact of increasing the number of local steps (C) for updating the predictor on the training and testing accuracy (mean ± std deviation). When the number of local steps (C) reaches 100%, it is equivalent to gradient descent as shown in (a)

Inspired by FedAVG, we study the effect of increasing the amount of local computation at each client. Specifically, in F-FL Games, each client updates its predictor based on a step of stochastic gradient descent over its mini-batch. We modify this setup by allowing each client to run a few steps of stochastic gradient descent locally ($C\%$). When the number of local steps at each client reaches is maximum (training data size/ mini-batch size) or $C = 100\%$, the scenario becomes equivalent to a gradient descent (GD) over the training data. From Table 4(b), it is evident that as the number of local steps increase i.e. each client **exactly** best responds to its opponents, the testing accuracy at equilibrium starts to decrease. When the local computation reaches 100%, i.e. each client updates its local predictor based on a GD over its data, F-FL Games exhibits converges (as shown in Figure 4(a)). FL Games is guaranteed to exhibit convergence and good out-of-distribution generalization behavior (Ahuja et al., 2021b) despite increasing local computations. Although the testing accuracy at convergence is lower compared to the standard setup, this approach opens avenues for practical deployment of the approach in FL.

6 Conclusion

In this work, we develop a novel framework based on Best Response Dynamics (BRD) training paradigm to learn invariant predictors across clients in Federated learning (FL). Inspired from Ahuja et al. (2020), the proposed method called Federated Learning Games (FL Games) learns causal representations which have good out-of-distribution generalization on new train clients or test clients unseen during training. We investigate the high frequency oscillations observed using BRD and equip our algorithm with a memory of historical actions. This results in smoother performance metrics with significantly lower oscillations. FL Games exhibits high communication efficiency as it allows parallel computation, scales well in the number of clients and results in faster convergence. Future directions include theoretically analyzing the smoothed best response dynamics, as it might have potential implications for other game-theoretic based machine learning frameworks.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021a. URL `https://openreview.net/forum?id=jlchsFOLfeF`.

Kartik Ahuja, Karthikeyan Shanmugam, and Amit Dhurandhar. Linear regression games: Convergence guarantees to approximate out-of-distribution solutions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1270–1278. PMLR, 2021b.

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

E Barron, Rafal Goebel, and R Jensen. Best response dynamics for continuous games. *Proceedings of the American Mathematical Society*, 138(3):1069–1083, 2010.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019.

Yo Joong Choe, Jiyeon Ham, and Kyubyong Park. An empirical study of invariant risk minimization. *arXiv preprint arXiv:2004.05007*, 2020.

Sreya Francis, Irene Tenison, and Irina Rish. Towards causal federated learning for enhanced robustness and privacy. *arXiv preprint arXiv:2104.06557*, 2021.

Drew Fudenberg, Fudenberg Drew, David K Levine, and David K Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

Hao Ge, Yin Xia, Xu Chen, Randall Berry, and Ying Wu. Fictitious gan: Training gans with historical models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134, 2018.

P Jean-Jacques Herings and Arkadi Predtetchinski. Best-response cycles in perfect information games. *Mathematics of Operations Research*, 42(2):427–433, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

Peter Kairouz et al. Advances and open problems in federated learning. dec. 10. *arXiv preprint arXiv:1912.04977*, 2019.

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pp. 4069–4077. PMLR, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

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). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1227–1231. IEEE, 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *arXiv preprint arXiv:2006.07242*, 2020.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pp. 7313–7324. PMLR, 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*, pp. 1273–1282. PMLR, 2017.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*, 2020.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *arXiv preprint arXiv:1801.04016*, 2018.

Maxime Peyrard, Sarvjeet Singh Ghotra, Martin Josifoski, Vidhan Agarwal, Barun Patra, Dean Carignan, Emre Kiciman, and Robert West. Invariant language modeling. *arXiv preprint arXiv:2110.08413*, 2021.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Alexander Robey, George Pappas, and Hamed Hassani. Model-based domain generalization. *Advances in Neural Information Processing Systems*, 34, 2021.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. *arXiv preprint arXiv:2010.05761*, 2020.

Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

Soroosh Shahtalebi, Jean-Christophe Gagnon-Audet, Touraj Laleh, Mojtaba Faramarzi, Kartik Ahuja, and Irina Rish. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. *arXiv preprint arXiv:2106.02266*, 2021.

Irene Tenison, Sreya Francis, and Irina Rish. Gradient masked federated optimization. *arXiv preprint arXiv:2104.10322*, 2021.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*, 2020.

Chuanlong Xie, Haotian Ye, Fei Chen, Yue Liu, Rui Sun, and Zhenguo Li. Risk variance penalization. *arXiv preprint arXiv:2006.07544*, 2020.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. Federated learning with unbiased gradient aggregation and controllable meta updating. *arXiv preprint arXiv:1910.08234*, 2019.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pp. 7184–7193. PMLR, 2019.

Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *arXiv preprint arXiv:2005.11418*, 2020.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Zhengyuan Zhou, Panayotis Mertikopoulos, Aris L Moustakas, Nicholas Bambos, and Peter Glynn. Mirror descent learning in continuous games. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5776–5783. IEEE, 2017.

Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. *arXiv preprint arXiv:2105.10056*, 2021.