FedGen: Generalizable Federated Learning

Praveen Venkateswaran
IBM Research
praveen.venkateswaran@ibm.com

Vinod Muthusamy
IBM Research
vmuthus@us.ibm.com

Vatche Isahagian
IBM Research
vatchei@ibm.com

Nalini Venkatasubramanian
UC Irvine
nalini@ics.uci.edu

ABSTRACT
Existing federated learning models that follow the standard risk minimization paradigm of machine learning often fail to generalize in the presence of spurious correlations in the training data. In many real-world distributed settings, spurious correlations exist due to biases and data sampling issues on distributed devices or clients that can erroneously influence models. Current generalization approaches are designed for centralized training and attempt to identify features that have an invariant causal relationship with the target, thereby reducing the effect of spurious features. However, such invariant risk minimization approaches rely on apriori knowledge of training data distributions which is hard to obtain in many applications. In this work, we present a generalizable federated learning framework called FedGen, which allows clients to identify and distinguish between spurious and invariant features in a collaborative manner without prior knowledge of training distributions. We evaluate our approach on real-world datasets from different domains and show that FedGen results in models that achieve significantly better generalization than current federated learning approaches.

1 INTRODUCTION
The rise in popularity of edge devices (e.g., mobile phones, wearables, sensors) has resulted in the generation of massive amounts of data, exacerbating network bandwidth demands, privacy concerns, and security issues. This has called for distributed model training over a large number of these devices for many machine learning applications. Federated Learning (FL) has emerged as an attractive paradigm, allowing local devices or clients to collaboratively train a shared global model. The typical federated learning paradigm involves two stages - (i) clients train models with their local datasets independently, and (ii) a central server gathers the locally trained models and aggregates them to obtain a shared global model.

A standard approach for federated learning has been FedAvg [28] where parameters of local models are averaged element-wise with weights proportional to sizes of the client datasets. However, FedAvg has been shown to have several shortcomings [19, 49], in particular its performance when the data across clients are non-iid. While several improvements have been proposed to handle non-iid data [25, 29], they rely on empirical risk minimization (ERM) techniques which has been shown by numerous examples to be brittle in generalizing to out-of-distribution (OOD) data.

This is partly due to being influenced by spurious correlations and data biases that fail to hold outside training data distributions [35, 43]. A classic example was highlighted by Beery et al. [5] where a model, trained to classify images of cows in pastures and camels in the desert, failed when the backgrounds were switched because it was influenced by the spurious correlation (i.e., green pastures with cows and sandy deserts with camels) rather than relying on the invariant features (i.e., the cows and camels themselves). In another example, DeGrave et al. [12] showed how models trained to detect COVID-19 from chest radiographs relied on spurious features (e.g. hospital metadata) rather than medical pathology features, and failed when tested in new hospitals. While there has been an increasing effort to improve the generalization of models to out-of-distribution data using different approaches like meta-learning [6], adversarial learning [2], subnetwork selection [52] etc, they are designed for centralized training and there has not been prior work done for generalizability in federated settings.

To demonstrate the sensitivity of FedAvg and associated empirical risk minimization approaches to spurious correlations in the training data, we use the Punctuated SST-2 dataset [10]. It consists of sentences and their binary sentiment labels divided into two training distributions. A punctuation mark, either a ‘!’ or ‘.’, is introduced as a spurious feature with an 80% and 90% spurious correlation with each of the binary sentiment labels in the two training distributions respectively, and only has a 10% correlation in the test distribution (i.e.) out-of-distribution (OOD) setting. Hence, any model relying on this punctuation feature rather than the sentence while predicting the sentiment would do well during training, but perform poorly on the test set where the spurious correlation no longer holds.

We distribute the training data over ten clients and compare the performance of FedAvg on data with the spurious feature against its performance if the spurious feature was not present. Figure 1 shows the training accuracy and the resulting out-of-distribution

(a) Training Accuracy

(b) Test Accuracy

Figure 1: Comparing the generalizability of FedAvg with and without spurious correlations in the training data
test accuracy of FedAvg in the two cases. We observe that when the spurious feature is present, FedAvg is influenced significantly by it, thereby resulting in a higher training accuracy than if it were not present. However, this influence results in poor generalization, and it demonstrates very low test accuracy on the out-of-distribution (OOD) data. The poor generalizability of FedAvg can be clearly seen from the performance of the two cases in Figure 1, since the only difference in the datasets is the presence of a spurious feature.

The example above highlights the drawbacks of existing federated learning approaches. In this work, we address the drawbacks by developing a generalizable federated learning approach called FedGen. Our approach develops a distributed masking function that enables clients to collaboratively identify spurious features or correlations in the data thereby ensuring that the resulting aggregated model relies only on the invariant features. In this work, we focus on classification tasks on time-series and sequential data, such as natural language processing and IoT data where spurious correlations are often present and will extend this to other tasks and domains as part of future work.

Our contributions can be summarized as follows:

- We present a generalizable federated learning framework (FedGen) that can handle the presence of spurious correlations or biases in the training data.
- FedGen uses a distributed masking function that allows the clients to collaboratively identify and distinguish between spurious and invariant features during the training process even as each client trains on only its local data.
- We formally prove the correctness and present results on the convergence of FedGen.
- We compare our framework with standard centralized training as well as existing federated learning approaches on a variety of sequential and time-series datasets from real-world domains and demonstrate the significantly improved generalizability of aggregated models using FedGen.

2 BACKGROUND

Consider a set of $K$ devices or clients, each with data samples drawn from $D_k \in \mathcal{D}$ distributions on $X \times Y$, where $X$ is the set of input features and $Y$ is the target variable. We denote $n_k$ as the number of data samples on device $k \in K$, and $N = \sum_{k=1}^{K} n_k$ as the total number of samples across the $K$ devices.

The general federated learning problem involves learning a single global statistical model from these $K$ devices under the constraint that device-generated data are stored and processed locally, with only intermediate model parameter updates being communicated periodically with a central server, which aggregates them. The goal is typically to minimize the following objective function:

$$
\min_w F(w), \text{ where } F(w) = \sum_{k=1}^{K} \frac{n_k}{N} f_k(w_k)
$$

where $f_k$ is the local objective function for the $k$th device. The local objective function is often defined by existing approaches using Empirical Risk Minimization (ERM) which minimizes the average training loss in a distribution agnostic manner:

$$
f_k(w_k) = \mathbb{E}_{(x_k, y_k) \sim D_k} [\ell(x_k, y_k); w_k]
$$

where $x_k \in X$, $y_k \in Y$, and $\ell$ is a given loss function. One of the popular aggregation methods is Federated Averaging (FedAvg) by McMahan et al. [28] which uses ERM and averages the local model parameters of the devices element-wise during aggregation as summarized in Algorithm 1. While ERM has been shown to work well in practice for i.i.d. data [44], it has been shown to fail when (i) training distributions on the devices are non-iid and leading each device to optimize its local objective as opposed to the central objective – potentially hurting convergence or even causing divergence of the aggregated model [7], and (ii) test distributions differ significantly from training distributions which is a natural consequence when spurious correlations are present in the training data [43].

Several alternative approaches to FedAvg have been proposed to address the first issue. FedProx proposed by Li et al. [25] limits the impact of local updates by keeping them close to the global model where they add a proximal term to the local client objective functions $f_k^2$:

$$
f_k^2(w_k) + \frac{\mu}{2} ||w_k - w||^2
$$

where the proximal term penalizes large deviations of the client model parameters, $\mu$ is a hyper-parameter and we see that when $\mu = 0$, this reduces to FedAvg. Agnostic Federated Learning (AFL) by Mohri et al. [29] is another variant of FedAvg which optimizes a centralized distribution that is a mixture of client distributions $D_k$ by modeling it as $D_k = \sum \gamma_k D_k$, where $\gamma_k$ is the mixture weight. Their goal is to find a solution that works for any $\gamma_k$. However, these approaches assume that all correlations present in the training distributions are invariant, and therefore cannot handle spurious correlations resulting in aggregated models that are not generalizable as we will demonstrate.

Finding the set of invariant features has emerged as a popular approach for generalization in centralized training settings. Invariant Risk Minimization (IRM) by Arjovsky et al. [4] tries to find an invariant data representation $\Phi(X)$ and assumes that the training data comes from multiple distributions, and that features whose distributions vary across the training data are also likely to vary between the training and test datasets and hence should be treated as spurious correlations. They define the following loss or risk function:

$$
\min_{\Phi} \sum_{D \in \mathcal{D}} L(D(\Phi(X_D))) + \lambda ||\nabla_w L_D(w^\top \Phi(X_D))||^2_2
$$

where the second term serves as a regularizer that balances between predictive power within a distribution (ERM), and the invariance of the predictor across distributions. There have been several extensions to the IRM framework. For instance Ahuja et al. [1] propose a game theoretic approach to IRM, while Krueger et al. [22] introduce the notion of risk extrapolation to encourage strict equality between training risks.

While the goal of identifying the invariant feature set has been shown to be effective for out-of-distribution generalization over a variety of datasets and serves as a base for our approach, IRM based approaches have been shown to have inherent drawbacks [30, 37]. Of particular importance is its reliance on prior knowledge of training data distributions, absent which, its performance is similar to ERM [46]. This knowledge is especially hard to obtain...
Whereas if \( E \) whose prediction probabilities differ across distributions due to Federated Averaging Algorithm 1.

As described in Section 1, generalizable models need to be able to vary variances in the feature weights to emphasize invariant features. From the above intuition, we know that the variance of invariant features is high and that of spurious features would be low. Hence, we see that updating the local masks on each client using Equation 5 results in the masks of invariant features gaining in value on each of the \( K \) clients since their variance is lower than the average (i.e., \( \frac{1}{K} \sum_{k=1}^{K} v(w_{ik}) - \sigma(v(w_{ik})) > 0 \)). Masks of spurious features on the other hand, become progressively negative, since the variance of their weights coupled with the scaling factor is larger than the average which is brought down by invariant features. Since the masks themselves are unbounded, we retain the scale of the original feature values by scaling the masks using the sigmoid function \( \sigma \) which is bounded between \([0, 1]\). Hence, using the masks and the update function in Equation 5 on the input features for each local client results in:

\[
\sigma(m_{ik}) \times X_i \rightarrow \begin{cases} X_i & \text{if } X_i \text{ is invariant} \\ 0 & \text{if } X_i \text{ is spurious} \end{cases}, \forall X_i \in X
\]

where \( \odot \) denotes element-wise multiplication. We then define the local loss on each client in FedGen as:

\[
\min_{w_k} f_k(Z_k, w_k) + \lambda \| \nabla w_k w_k^T f_k(Z_k, w_k) \|^2_2, \quad \text{where}
\]

\[
f_k(Z_k, w_k) = E_{(x_i, y_i) \sim D_k}[f(Z_k, y_i; w_k)], \quad \text{s.t. } Z_k = \sigma(M_k) \odot x_k
\]

The penalty term here like IRM serves as a regularizer for the local models that penalizes doing too well on one data distribution (i.e., possibly relying on spurious features) and rewards doing well across distributions (i.e., relying on invariant features). FedGen then builds a centralized model by communicating and subsequently aggregating both the local model parameters as well as the local masks of the participating devices or clients. We use the same element-wise averaging for aggregation as used by prior work:

\[
w_k^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} w_k^{t+1}, \quad M_k^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} M_k^{t+1}
\]

Aggregating the masks across the \( K \) devices or clients allows them to obtain a consensus on invariant and spurious features across the different data distributions that can be present in their local training data. Masks that are positive (i.e., reflecting invariant features) and spurious features across most clients will continue to be emphasized after aggregation, and likewise for the negative spurious feature masks. The mask aggregation also allows client models relying on the wrong features to obtain the global consensus and ignore spurious features. We define the FedGen central objective function as:

\[
\min_w F(Z, w), \quad \text{where } F(Z, w) = \sum_{k=1}^{K} \frac{n_k}{N} f_k(Z_k, w_k)
\]

We summarize our FedGen approach in Algorithm 2. We next formally prove that FedGen results in an aggregated model that is generalizable in Theorem 1 by formulating a minimax problem and showing that the resulting model minimizes the loss using invariant features, even under the most adverse unseen test distributions.

**Algorithm 1** Federated Averaging FedAvg

**Require:** Devices \( k \in K \), local epochs \( E \), learning rate \( \eta \), global model \( w \) randomly initialized

1. \( \text{for } t = 1 \rightarrow T \text{ do} \)
2. Server selects subset \( K_t \) of \( K \) devices at random
3. Server sends \( w^t \) to devices in \( K_t \)
4. \( \text{for each client } k \in K_t \text{ in parallel do} \)
5. Update \( w^t \) for \( E \) epochs of SGD to get \( w^{t+1} \)
6. Send \( w^{t+1} \) back to the server
7. Server aggregates \( w^{t+1} \leftarrow \frac{\sum_{k=1}^{K} n_k}{N} w_k^{t+1} \)

3 FEDGEN: GENERALIZABLE FEDERATED LEARNING

As described in Section 1, generalizable models need to be able to distinguish between spurious and invariant features, thereby ignoring spurious correlations with the target variable present in training data. Given the set of input features \( X \) used by the device models, the set of invariant features \( X^I \) across all devices is one where the target prediction probability is consistent across all data distributions, (i.e., \( p(Y|X_i \in X^I, D) \)) is approximately constant. Conversely, the set of spurious features \( X^S \) consists of features whose prediction probabilities differ across distributions due to the spurious correlations. Hence, it follows that \( X^I \cup X^S = X \), and \( X^I \cap X^S = \emptyset \), (i.e.) a feature cannot be both invariant and spurious.

Our approach to determining whether the \( i \)th feature \( X_i \in X \) is spurious or invariant, is by measuring the stability of its parameter weight \( w_i \). Recent work [17, 46] has shown that if \( X_i \) is a causal or invariant feature, \( w_i \) converges to a fixed magnitude, (i.e., \( \mathbb{E}[Y|X_i] = c \) for some constant value \( c \), across all training epochs. Whereas if \( \mathbb{E}[Y|X_i] \) is changing, \( w_i \) would keep changing as well, and hence spurious features have parameter weights that exhibit high variance. This definition is equivalent to learning features whose correlations with the target variable are stable.

We leverage this intuition for our federated setting and define a set of masks \( M_k = \{ m_{ik}, ..., m_{jk} \} \) over the input features for each of the \( k \in K \) clients, where \( m_{ij} \in \mathbb{R} \) and \( j \) is the number of input features in \( X \). We update the masks during training by using the variances in the feature weights to emphasize invariant features and suppress spurious ones. During each training epoch, we update the local masks for each client \( k \in K \) as:

\[
m_{ik} = m_{ik} + \frac{1}{E} \sum_{t=1}^{E} o(w_{ik}) - \sigma(o(w_{ik})), \forall m_{ij} \in M_k
\]

where hyper-parameter \( \alpha \) serves as a scaling factor, \( o(w_{ik}) \) is the variance of the weights of feature \( X_i \) on client \( k \), and the second term is the average variance of all feature weights on client \( k \). From the above intuition, we know that the variance of invariant features is low and that of spurious features would be high. Hence, we see that updating the local masks on each client using Equation 5 results in the masks of invariant features gaining in value on each of the \( K \) clients since their variance is lower than the average (i.e., \( \frac{1}{K} \sum_{k=1}^{K} v(w_{ik}) - \sigma(v(w_{ik})) > 0 \)). Masks of spurious features on the other hand, become progressively negative, since the variance of their weights coupled with the scaling factor is larger than the average which is brought down by invariant features. Since the masks themselves are unbounded, we retain the scale of the original feature values by scaling the masks using the sigmoid function \( \sigma \) which is bounded between \([0, 1]\). Hence, using the masks and the update function in Equation 5 on the input features for each local client results in:

\[
\sigma(m_{ik}) \times X_i \rightarrow \begin{cases} X_i & \text{if } X_i \text{ is invariant} \\ 0 & \text{if } X_i \text{ is spurious} \end{cases}, \forall X_i \in X
\]

where \( \odot \) denotes element-wise multiplication. We then define the local loss on each client in FedGen as:

\[
\min_{w_k} f_k(Z_k, w_k) + \lambda \| \nabla w_k w_k^T f_k(Z_k, w_k) \|^2_2, \quad \text{where}
\]

\[
f_k(Z_k, w_k) = E_{(x_i, y_i) \sim D_k}[f(Z_k, y_i; w_k)], \quad \text{s.t. } Z_k = \sigma(M_k) \odot x_k
\]

The penalty term here like IRM serves as a regularizer for the local models that penalizes doing too well on one data distribution (i.e., possibly relying on spurious features) and rewards doing well across distributions (i.e., relying on invariant features). FedGen then builds a centralized model by communicating and subsequently aggregating both the local model parameters as well as the local masks of the participating devices or clients. We use the same element-wise averaging for aggregation as used by prior work:

\[
w_k^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} w_k^{t+1}, \quad M_k^{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} M_k^{t+1}
\]

Aggregating the masks across the \( K \) devices or clients allows them to obtain a consensus on invariant and spurious features across the different data distributions that can be present in their local training data. Masks that are positive (i.e., reflecting invariant features) and spurious features across most clients will continue to be emphasized after aggregation, and likewise for the negative spurious feature masks. The mask aggregation also allows client models relying on the wrong features to obtain the global consensus and ignore spurious features. We define the FedGen central objective function as:

\[
\min_w F(Z, w), \quad \text{where } F(Z, w) = \sum_{k=1}^{K} \frac{n_k}{N} f_k(Z_k, w_k)
\]

We summarize our FedGen approach in Algorithm 2. We next formally prove that FedGen results in an aggregated model that is generalizable in Theorem 1 by formulating a minimax problem and showing that the resulting model minimizes the loss using invariant features, even under the most adverse unseen test distributions.

**Theorem 1.** Given training distributions \( D^I \) and a test distribution \( D^S \), the set of invariant features \( X^I \) is the saddle point of the following minimax problem:

\[
X^I = \min_w \max_{X^I, X^S} F(Z, w; D^S), \quad \text{where } Z = \sigma(M) \odot X
\]
Algorithm 2 Federated Generalization FedGen

Require: Devices \( k \in K \), local epochs \( E \), learning rate \( \eta \), global model \( w \) randomly initialized
Require: Local masks \( M_k \in M \), scaling factor \( a \), warm up epochs \( \varepsilon_{init} \)
1: for \( t = 1 \to T \) do
2: Server selects subset \( K_t \) of \( K \) devices at random
3: Server sends \( w^t \) to devices in \( K_t \)
4: ClientUpdate:
5: for each client \( k \in K \), do
6: Initialize masks \( m_{ik} = 1 \), \( \forall m_{ik} \in M_k \)
7: Initialize mean \( u_{ik} = 0 \), variance \( \nu_{ik} = 0 \)
8: for \( e = 1 \to E \) do
9: Sample batch \( x_k, y_k \)
10: \( \ell_{loc} = [f(\mu(M_k) \odot x_k, y_k); w_k] \) \( \rightarrow \) prediction error
11: \( \ell_1 = ||w_k||_1 \) \( \rightarrow \) L1 regularization
12: \( \ell_{pen} = \lambda \|\nabla w_k^T f_k(Z_k, w_k)\|_2^2 \) \( \rightarrow \) FedGen penalty
13: \( \mathcal{L} = \ell_{loc} + \ell_1 + \ell_{pen} \) \( \rightarrow \) Total loss
14: \( w_k = w_k - \eta \nabla \mathcal{L} \) \( \forall w_k \in K \)
15: \( u_{old} = u_k \)
16: \( u_k = \beta w_k + (1 - \beta) u_{old} \) \( \rightarrow \) Mean estimate
17: \( \nu_k = (1 - \delta)(w_k - u_{old})^2 \) \( \rightarrow \) Variance estimate
18: if \( e > \varepsilon_{init} \) then \( \rightarrow \) Update masks
19: \( m_{ik} \leftarrow \frac{1}{K} \sum_{k=1}^{K} n_k w_{ik}^t - \mu(\nabla w_k), \forall m_{ik} \in M_k \)
20: Server aggregates \( w^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k w_k^{t+1}}{N_k} \)
21: Server aggregates \( M^{t+1} \leftarrow \sum_{k=1}^{K} \frac{M_k^{t+1}}{N_k} \)

where \( F(Z, w) \) is the FedGen central objective function, \( M \) the final aggregated masks, and \( X^I, X^S \) denote the set of invariant and spurious features respectively such that \( X^I \cup X^S = X \), and \( X^I \cap X^S = \emptyset \) (i.e.) they are disjoint.

Proof. For every \( Z \), we can partition it into invariant variables \( Z^I \) and non-invariant variables \( Z^S \) as:

\[
Z^I = \sigma(M) \circ X^I, \quad Z^S = \sigma(M) \circ X^S.
\]

Consider a test distribution \( D^{te} \), where the set of spurious features \( X^S \) are not predictive of the output \( Y \), and only the invariant features \( X^I \) are predictive of \( Y \).

\[
p(Y|Z, D^{te}) = p(Y|Z^I, D^{te}), \quad p(Y|Z, D^{tr}) = p(Y|Z^I, D^{tr})
\]

Therefore, \( F(Z, w) \) is

\[
F(Z, w) = \max_{Z^I} \left\{ H(p(Y|Z^I, D^{te}), p(Y|Z, D^{tr})) \right\}
\]

Recall that \( X^S \) was assumed to be non-predictive of \( Y \). However, in most cases, the spurious feature \( X^S \) would have some predictive power over \( Y \) in the training environment. Hence, from the definition of spurious features \( X^S \), their biased influence on the model performance during training will lead to an increased loss in the worst case test environment:

\[
\max_{X^S} F(Z, w; D^{te}) \geq F(Z, w; D^{tr})
\]

By combining equations (13), (14), and (15), we have:

\[
\max_{X^S} F(Z, w; D^{te}) \geq \max_{X^S} F(X^I, w; D^{tr})
\]

which in turn implies,

\[
X^I = \min_{X^I, X^S} \max_{w} F(Z, w; D^{te})
\]

3.1 FedGen Convergence

In this section, we analyze the convergence of FedGen by first introducing some definitions and assumptions for our convergence analysis.

Definition 1. (Smoothness) The function \( f \) is \( L \)-smooth with constant \( L > 0 \) if \( \forall x_1, x_2 \)

\[
f(x_1) - f(x_2) \leq \langle \nabla f(x_2), x_1 - x_2 \rangle + \frac{L}{2} ||x_1 - x_2||^2
\]

Definition 2. (Convexity) The function \( f \) is \( \mu \)-strongly convex with \( \mu > 0 \) if \( \forall x_1, x_2 \)

\[
f(x_1) - f(x_2) \geq \langle \nabla f(x_2), x_1 - x_2 \rangle + \frac{\mu}{2} ||x_1 - x_2||^2
\]

In order to analyze convergence in heterogeneous non-iid settings with constant step-size (as is usually implemented in practice), we also introduce a notion of dissimilarity between the local device functions in the federated network similar to assumptions made by prior work [25, 45, 51].

Definition 3. (B-local dissimilarity) The local functions \( f_k \) are B-locally dissimilar at \( Z \) if \( \exists \Sigma = \|\nabla f_k(Z, w)\|^2 \leq \|\nabla F(Z, w)\|^2 B^2 \).

We further define \( B(w) = \sqrt{\frac{\|\nabla F(Z, w)\|^2}{\|\nabla F(Z, w)\|^2}} \) when \( \|\nabla F(Z, w)\|^2 \neq 0 \). As explained in Li et al.[25], when all the local functions are the same (i.e.) distributions of data sampled from the devices are i.i.d, we have \( B(w) \to 1 \) for all \( w \). However, in the presence of spurious correlations, the data distributions are heterogeneous and therefore \( B > 1 \), and larger the value of \( B(w) \), the more the heterogeneity of the local data. We next make the following assumption on the objective functions.

Assumption 1. Assume that:

\[ Preprint — under submission. \]
Theorem 3. Let Assumption 1 hold. Assume $F(Z, w)$ is $L$-smooth and $\mu$-strongly convex and functions $f_k$ are $B$-locally dissimilar with learning rate $\eta$. Then we have that for $\rho = -\eta(1 - \frac{\mu L_{\mathcal{P}}}{2})$, at iteration $t$ of FedGen, we have the following expected decrease in the objective:

$$
\mathbb{E}[F(Z^{t+1}, w^{t+1})] - F(Z_t, w_t) \leq (1 - 2\mu\rho)(F(Z^t, w^t) - F(Z_t, w_t))
$$

We present the detailed proof in Appendix A.1

Theorem 3. Let Assumption 1 hold. Assume $F(Z, w)$ is $L$-smooth and non-convex and functions $f_k$ are $B$-locally dissimilar with learning rate $\eta$. Also, let $F(Z^0, w^0) - F(Z^*, w^*) = \Delta$, where $w^* = \min_w F(Z, w)$. Then after $T$ iterations of FedGen, we have

$$
\frac{1}{T} \sum_{t=0}^{T-1} \rho \mathbb{E}[||\nabla F(Z^t, w^t)||^2] \leq \Delta
$$

(19)

The proof can be found in Appendix A.2

4 EXPERIMENTS

In this section, we evaluate FedGen on diverse models, clients, and datasets spanning multiple application domains: Natural Language Processing (NLP), Internet-of-Things (IoT), and business process mining (Table 1). Similar to prior work [1, 4, 10, 22], we augment these benchmark datasets with spurious features $X^S$ such that it has a strong spurious correlation with the target labels $Y$ in the training data distributions (i.e.) $p(Y | X^S, D^{tr}) \geq \alpha = 0.8$, and this spurious relationship does not hold in the test distribution (i.e.) $p(Y | X^S, D^{te}) \leq \alpha = 0.1$.

We assess the quality of generalization of the aggregated model as the classification accuracy obtained on the test environment, disjoint from the set of training environments. Hence, an approach that cannot generalize will do well during training but not on the test set, while a generalizable model will continue to perform well on the test data. We also present results from an ablation study of the masking function as well as additional experiments that highlight the robustness of our approach.

### 4.1 Datasets

#### 4.1.1 SST-2: We modify the Stanford Sentiment Treebank (SST-2) [41], a benchmark dataset for binary sentiment analysis with spurious correlations in a similar manner as Choe et al. [10]. We add a spurious feature $X^S$ by pairing each sentiment with a final punctuation mark (, or !) in the corresponding text as follows:

$$
p(Y = 0 | X^S = ., D) = p(Y = 1 | X^S = !, D) = \alpha$$

$$
p(Y = 0 | X^S = !, D) = p(Y = 1 | X^S = ., D) = 1 - \alpha
$$

where we set $\alpha$ as described above. The addition of the punctuation mark as a separate feature, and not as part of the sentence, allows us to isolate the sentence sequence embeddings to mask, while still preserving the structure and meaning of the sentence (Choe et al. [10]). We create two different training distributions with differing strengths of spurious correlations and treat the test data as a third distribution with weak spurious correlation as described above. We distribute the training data among 100 devices or clients.

#### 4.1.2 AG News: To evaluate FedGen for multi-class NLP predictions, we use the AG News dataset [53], which consists of a corpus of news articles and their titles that have been classified into four categories - (1) World, (2) Sports, (3) Business, and (4) Sci/Tech. There are 30,000 training and 1,900 test examples for each class. We again add a spurious feature that pairs each of the four news categories with a corresponding final punctuation mark in the article from the set $\mathcal{P} = \{ ., @, !, \} \}$ as follows:

$$
p(Y = 0 | X^S = ., D) = p(Y = 1 | X^S = !, D) = \alpha$$

$$
p(Y = 2 | X^S = @, D) = p(Y = 3 | X^S = \ldots, D) = \alpha$$

$$
p(Y = y_j | X^S = \text{rand}(\mathcal{P} \setminus \mathcal{P}_j), D) = 1 - \alpha
$$

and set $\alpha$ as before. We divide the training data into two distributions with strong spurious correlations, and a separate test distribution with a weak correlation. We distribute the training data among 100 devices.

#### 4.1.3 Human Activity Recognition (HAR). The HAR dataset [3] consists of smartphone accelerometer and gyroscope readings recorded from 30 individuals performing six activities (walking, standing, sitting, etc.). The model objective is to predict the activity being performed given the sensor readings. The data consists of sequences of 128 time-steps of sensor readings which correspond to any particular activity. We consider the data from the smartphone of each of the individuals to be a client, and divide them into two training and one test distribution such that there is a strong spurious correlation between the specific activity performed and the smartphone model used by individuals in the training test, which does not hold for individuals in the test set as follows:

$$
p(Y = y_j | X^S = S_j, D) = \alpha$$

$$
p(Y = y_j | X^S = \text{rand}(S \setminus S_j), D) = 1 - \alpha
$$

where $y_j$ and $S_j$ refer to the $j$th activity and smartphone model respectively.

#### 4.1.4 Water Quality Detection. The dataset [14] comprises of water quality data collected from different cities across the United States. It includes sensor readings such as turbidity, conductivity, pH, temperature, DO2, in addition to environmental attributes like humidity and precipitation, which correspond to binary labels depicting the occurrence of a contamination event. We divide the training data into two distributions, clustered the humidity into

| Dataset | Devices/ Clients | Train Seq. | Test Seq. | #Spurious |
|---------|------------------|------------|----------|-----------|
| SST-2   | 100              | 67,350     | 873      | 1         |
| AG News | 100              | 120,000    | 7600     | 1         |
| HAR     | 30               | 7352       | 2947     | 2         |
| Water Quality | 120     | 132,227    | 14,237   | 1         |
| BPIC 2018 | 100         | 306,615    | 63,692   | 1         |

Table 1: Summary of FedGen Datasets

Preprint — under submission.
We distribute the training data among 120 clients corresponding to variants and augment the process trace sequences with a spurious contamination labels \((L)\) and \((H)\) values and spuriously correlate them to the contamination labels \((C, \emptyset)\) as follows:

\[
p(Y = C \mid X^S \in H, D) = p(Y = \emptyset \mid X^S \in L, D) = \alpha
\]

\[
p(Y = C \mid X^S \in L, D) = p(Y = \emptyset \mid X^S \in H, D) = 1 - \alpha
\]

We distribute the training data among 120 clients corresponding to the different cities.

### 4.1.5 Business Process Mining:
We use a benchmark process mining dataset (BPIC 2018)\(^1\) which consists of payment applications from German farmers to the European Agricultural Guarantee Fund collected over a period of three years. Each application is processed by one of four departments, and to evaluate our approach in a multi-distribution setting, we consider each department as a distribution and use three of them for training and the fourth as our out-of-distribution (OOD) test data. We consider two business process variants and augment the process trace sequences with a spurious feature denoting the area of the farm. We cluster the area into low \((L)\) and high \((H)\) values and spuriously correlate them to the two business process variants \((A, B)\) as follows:

\[
p(Y = A \mid X^S \in H, D) = p(Y = B \mid X^S \in L, D) = \alpha
\]

\[
p(Y = A \mid X^S \in L, D) = p(Y = B \mid X^S \in H, D) = 1 - \alpha
\]

and define \(\alpha\) as before. We distribute the training data among 100 clients.

### 4.2 Benchmarks
We compare FedGen to four benchmark approaches:

- **ERM**: Standard empirical risk minimization approach that minimizes the average loss over the entire training data. We use this as a comparison with centralized training.
- **FedAvg**: Federated averaging approach by McMahan et al. \([28]\) where model parameters are averaged element-wise.
- **FedProx**: Approach by Li et al. \([25]\) to handle heterogeneous non-iid data in federated settings using a proximal term to keep local models close to the global model.
- **Inv-FedAvg**: FedAvg trained on data without any spurious features. This approach reflects the setting where spurious correlations are not present and hence reflects the best performance.

### 4.3 Implementation Details
We implement FedGen, ERM, FedAvg, and Inv-FedAvg using the PyTorch library \([31]\) and implement FedProx based on their publicly available code\(^2\).

### Table 2: Accuracy achieved by comparison approaches for all datasets

| Algorithm       | SST-2 Train | SST-2 Test | AG News Train | AG News Test | HAR Train | HAR Test | Water Quality Train | Water Quality Test | BPIC 2018 Train | BPIC 2018 Test |
|-----------------|-------------|------------|---------------|-------------|-----------|----------|---------------------|------------------|----------------|----------------|
| ERM             | 85.6±0.1    | 96.1±0.3   | 92.3±0.1      | 63.4±0.5    | 98.1±0.3  | 73.8±0.5 | 88.7±0.3            | 65.3±0.3         | 77.1±0.7       | 73.7±0.4       |
| FedAvg          | 83.3±0.1    | 96.1±0.1   | 91.5±0.6      | 61.3±0.2    | 98.0±0.5  | 70.1±1.6 | 89.3±0.7            | 63.8±1.6         | 77.1±2.1       | 71.7±1.3       |
| FedProx         | 75.0±1.1    | 96.1±0.1   | 91.3±0.8      | 61.1±0.3    | 96.7±0.5  | 71.5±0.6 | 87.7±1.1            | 66.1±0.7         | 73.9±1.1       | 71.2±0.5       |
| **FedGen (ours)** | **58.3±0.2** | **60.9±0.3** | **81.8±0.3**  | **82.3±0.1** | **94.2±1.2** | **87.5±0.3** | **64.3±0.2** | **85.6±1.1** | **74.8±0.8** | **86.1±0.4** |
| Inv-FedAvg      | 66.6±0.2    | 62.5±0.2   | 82.1±0.4      | 82.8±0.1    | 93.0±0.4 | 89.9±0.5 | 74.9±1.1            | 88.1±0.6         | 78.0±0.3       | 88.7±0.3       |

\(^1\)https://data.4tu.nl/articles/BPI_Challenge_2018/12688355

\(^2\)https://github.com/litian96/FedProx

Preprint — under submission.
across all five datasets. The candidate local epochs we consider are \( E \in \{20, 40, 60, 80, 100, 120, 140\} \). We run each approach till it achieves convergence and report the test accuracy achieved in Figure 2. We observe that the performance of FedAvg and Inv-FedAvg deteriorates with longer local training, thus showcasing its sensitivity to hyperparameter settings, which matches prior observations made in literature. FedProx only partially alleviates this problem and also demonstrates a drop in accuracy for larger number of epochs. On the other hand, we observe that FedGen often benefits from longer training, since it gets to further emphasize invariant features.

### 4.4.3 Convergence Rate

In this experiment we compare the global model convergence achieved by all the comparison approaches. We report the convergence rate for the local training epochs \( E \) that yields the best final aggregated model accuracy over the test set for each of the approaches. Figure 3 shows the convergence rates of all the approaches across the five datasets. We observe that the convergence rate achieved by the approaches is quite similar, and that FedProx converges the quickest, followed by FedGen and FedAvg. This demonstrates that FedGen can result in significantly more generalizable models without a loss in runtime performance compared to existing approaches.

### 4.4.4 Ablation Study

We perform an ablation experiment to measure the contribution of the components of our masking function and the FedGen penalty term as shown in Table 3. We first remove the hyperparameter \( \alpha \) used to scale the masks, and observe that there is a drop in accuracy across all three datasets since the degree of spuriousness may not be fully captured, and hence the local models can get influenced by spurious features. We next remove the masking function \( \sigma(M) \), essentially reducing the loss function to that used by FedAvg. This is reflected by the accuracy values, which are similar to those achieved by FedAvg, thereby demonstrating that our proposed masking function is the primary reason for model generalization. We finally remove the penalty term while optimizing model weights (Equation 7) and again observe a drop in accuracy across all datasets since the regularization is important to ensure that the model does not get influenced by spurious correlations in any one distribution.

### 4.4.5 Varying Number of Training Distributions

We measure the robustness of FedGen to varying numbers of distributions in the training data using the AG News dataset. We augment the dataset with more distributions with differing levels of spurious correlations which are similar to those achieved by FedAvg, thereby demonstrating that our proposed masking function is the primary reason for model generalization. We finally remove the penalty term while optimizing model weights (Equation 7) and again observe a drop in accuracy across all datasets since the regularization is important to ensure that the model does not get influenced by spurious correlations in any one distribution.

---

**Table 3: Ablation Study**

| Algorithm            | SST-2 | AG News | HAR   | Water Quality | BPIC 2018 |
|----------------------|-------|---------|-------|---------------|-----------|
| FedGen (ours)        | 82.3  | 82.1    | 82.1  | 82.2          | 82.2      |
| FedAvg               | 61.3  | 61.0    | 60.3  | 60.0          | 60.0      |
| FedProx              | 61.1  | 60.7    | 60.6  | 60.3          | 60.3      |
| Inv-FedAvg           | 82.8  | 82.6    | 82.8  | 82.2          | 82.8      |

**Table 4: Varying Number of Training Distributions**

| Algorithm | Number of Distributions |
|-----------|-------------------------|
| ERM       | 2  | 4  | 6  | 8  |
|           | 63.4 | 63.0 | 62.6 | 62.0 |
| FedAvg    | 61.3 | 61.0 | 60.3 | 60.0 |
| FedProx   | 61.1 | 60.7 | 60.6 | 60.3 |
| FedGen (ours) | 82.3 | 82.1 | 82.1 | 82.2 |
| Inv-FedAvg| 82.8 | 82.6 | 82.8 | 82.8 |
with $\alpha > 0.8$. Table 4 shows that the performance of FedGen and Inv-FedAvg remains consistent even with an increased number of distributions, demonstrating FedGen’s robustness, while the other approaches are impacted to a small extent.

5 RELATED WORK

As edge devices like phones, sensors, and wearables grow in capabilities and popularity, it is increasingly attractive to train models locally using federated learning [24]. To improve the performance of the popular FedAvg approach [28] in statistically heterogeneous settings, FedProx [25] and Agnostic Federated Learning [29] are two approaches that we described in Section 2. Karimireddy et al. propose SCAFFOLD [20] which uses control variates (variance reduction) to correct for any distribution drift or changes in the local clients. However, the heterogeneity assumed by these approaches does not extend to spurious correlations as we have demonstrated.

Additionally, heuristic approaches have been proposed to tackle the heterogeneity by sharing the local device data or server-side proxy data. However, in addition to placing a burden on network bandwidth, sending local data to the server as in Jeong et al.[18] violates the key privacy assumption of federated learning, and sending globally-shared proxy data to all clients (Zhao et al.[54], Huang et al.[16]) requires effort to carefully generate the data.

There has also been work on improving the communication costs of FedAvg. Wang et al.[49] propose FedMA which matching the neurons of client models before averaging them, thereby reducing the communication needed. CMFL is an approach by Luping et al.[27] where every client checks if its update is relevant for model improvement and reduces the number of irrelevant updates to the server, while FedAsync (Xie et al.[50]) improves training time by focusing on handling stragglers. However, these approaches assume that the training and test datasets come from the same population, which may not happen in many real-world settings, thereby calling for generalizable approaches.

Prior work on machine learning generalization have predominantly looked at computer vision challenges. Data augmentation techniques aim to make the model more robust by training using instances obtained from neighbouring domains hallucinated from the training domains, and thus make the network ready for these neighbouring domains. Shankar et al. [40] augment data using instances perturbed along directions of domain change and use a second classifier to capture this. Volpi et al. [48] apply this to single domain data, while Carlucci et al. [9] apply augmentation to images during training by simultaneously solving an auxiliary unsupervised jigsaw puzzle alongside.

To obtain stable predictions for unseen test distributions, a desirable model relies on only invariant features [39] and causal model discovery [32] aims to find an underlying causal graph to obtain the invariant feature set. Peters et al.[33] adopt statistical tests for mining the invariant set, claiming that the residual of the invariant method should be i.i.d. while Rojas-Carulla et al.[36] perform this for non-linear settings. Decomposition based approaches represent the parameters of the network as the sum of a common parameter and domain-specific parameters during training [11], Khosla et al. [21] applied decomposition to domain generalization by retaining only the common parameter for inference. Li et al. [23] extended this work to Convolutional Neural Networks (CNNs) where each layer of the network was decomposed into common and specific low-rank components. Piratla et al. [34] recently proposed a more efficient approach that decomposes only the last layer, imposes loss on both the common and domain-specific parameters, and constrains the two parts to be orthogonal.

Another approach is to pose the generalization problem as a meta-learning task which has been studied either in the context of few-shot supervised learning methods which adapt using small amounts of labeled data from the new domain [15], distribution shifts in only test domains [13], or only considering label shifts [26]. There have also been some prior work on generalization for other applications like business process predictions [47] and medical diagnosis using human annotated spurious features [42].

6 CONCLUSION

We present an approach to improve the generalizability of models trained using federated learning. FedGen overcomes the brittleness of current empirical risk minimization based approaches in the presence of spurious correlations in the training data, while also not needing any prior knowledge of training distributions unlike existing generalization approaches. We develop a masking function for the local clients that allows them to collaboratively identify and distinguish between invariant and spurious features during the training process. The final aggregated model thus relies on the invariant feature set, thereby ensuring its robustness and generalizability to unseen test distributions. We prove that our masking function minimizes loss even under the most adverse test distributions. We show that FedGen achieves good generalization and outperforms existing federated learning approaches on a variety of real-world datasets. We believe that this work motivates the need to improve generalizability in federated learning and as part of future work, we will work on improving the communication costs and extend our approach to other models that satisfy the minimax optimization such as GANs, as well as other prediction tasks.

REFERENCES

[1] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. 2020. Invariant risk minimization games. In International Conference on Machine Learning. PMLR, 145–151.
[2] Isabela Albuquerque, João Monteiro, Tiago H Falk, and Joannis Mitliagkas. 2019. Adversarial target-invariant representation learning for domain generalization. arXiv preprint arXiv:1911.00804 (2019).
[3] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. 2013. A public domain dataset for human activity recognition using smartphones. In Euro. Vol 3. 3.
[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019).
[5] Sara Beery, Grant Van Horn, and Pietro Perona. 2018. Recognition in terra incognita. In Proceedings of the European Conference on Computer Vision (ECCV). 456–473.
[6] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 2019. A meta-transfer objective for learning to disentangle causal mechanisms. arXiv preprint arXiv:1901.10912 (2019).
[7] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dimitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046 (2019).
[8] Sebastian Candás, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. arXiv preprint arXiv:1812.01097 (2018).
[9] Fabio M Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. 2019. Domain generalization by solving jigsaw puzzles. In Proceedings
With Assumption 1 and B-local dissimilarity, we get

\[ F(Z^{t+1}, w^{t+1}) - F(Z^t, w^t) \leq \langle \nabla F(Z^t, w^t) + \frac{L}{2} \rangle \left| w^{t+1} - w^t \right|^2 \]  

(20)

Then summing up Equation 28 over global iteration \( T \), we obtain

\[ \frac{1}{T} \sum_{t=0}^{T-1} \rho \mathbb{E} \left[ \left| \nabla F(Z^t, w^t) \right|^2 \right] \leq F(Z^0, w^0) - F(Z^*, w^*) \]  

(31)

**A.3 Implementation Details**

We simulate the federated learning setup (1 server and \( N \) devices) on a commodity machine with 7 Intel® Xeon® E5-2690 v4 CPUs and an NVIDIA® Tesla P100 GPU.

We describe the implementation details of the experiments in the paper as described in Section 4. We searched for hyper-parameters based on the values in Table 5, and chose the configurations with the best performance for each of the approaches.

| Hyper-parameter                  | Range            |
|----------------------------------|------------------|
| Learning rate                    | \( [10^{-5}, 10^{-1}] \) |
| Rounds                           | \( [11, 201] \)  |
| Epochs                           | \( [5, 150] \)   |
| Regularization weight            | \( [10^{-6}, 10^{-2}] \) |
| Penalty                          | \( [10^{-8}, 10^{-4}] \) |
| Scaling factor                   | \( [10^{-2}, 10] \) |
| Proximal term (FedProx)          | \( [10^{-3}, 10^{-1}] \) |

**Table 5: Hyper-parameter ranges tried for all comparison approaches**

For both the SST-2 and AG News datasets, we used a 3-layer (50-50-2) MLP with ReLU activations, based on the average of the 50-dimensional GloVe (glove.6B.50d) embedding word vectors as inputs. These are pre-trained word vector embeddings that have been trained on Wikipedia articles. We achieve similar results with the 300-dimensional word vectors as well (glove.6B.300d). For SST-2, we use a learning rate of 0.001 and set the penalty and regularizer weight for FedGen to be \( 10^{-4} \) and \( 10^{-6} \) respectively with a scaling factor of 10. The regularizer weight for ERM, FedAvg, Inv-FedAvg and FedProx was set to \( 10^{-6} \) and the proximal term was \( 10^{-3} \).

For AG News, we also use a learning rate of 0.001 and set the penalty and regularizer weight for FedGen to be 1 and \( 10^{-6} \) respectively with a scaling factor of 10. The same regularizer weight was used for the other approaches and the proximal term was \( 10^{-3} \).

For the HAR dataset, we use an LSTM with one layer and hidden size of 100, followed by a linear layer of size 100 with ReLU activation. We use a batch size of 20 on the clients and a learning rate of 0.01. The regularizer weight for all approaches was set to \( 10^{-3} \) and FedGen’s penalty was 10 with a scaling factor of 10. We used the same proximal term as above.

For the Water Quality dataset, we used a 3-layer MLP (200-200-1) with ReLU activations. The learning rate was 0.07 and the regularizer weight was \( 10^{-6} \). We set FedGen’s penalty to 1 with a scaling factor of 10.

For BPI 2018, we use an LSTM with two layers and a hidden size of 120 followed by a linear layer of size 120 with ReLU activation. We use a batch size of 5% on the clients. The learning rate was 0.005 and the regularizer weight was \( 7\cdot10^{-6} \). We set the penalty of FedGen to 1 and the scaling factor to 10.