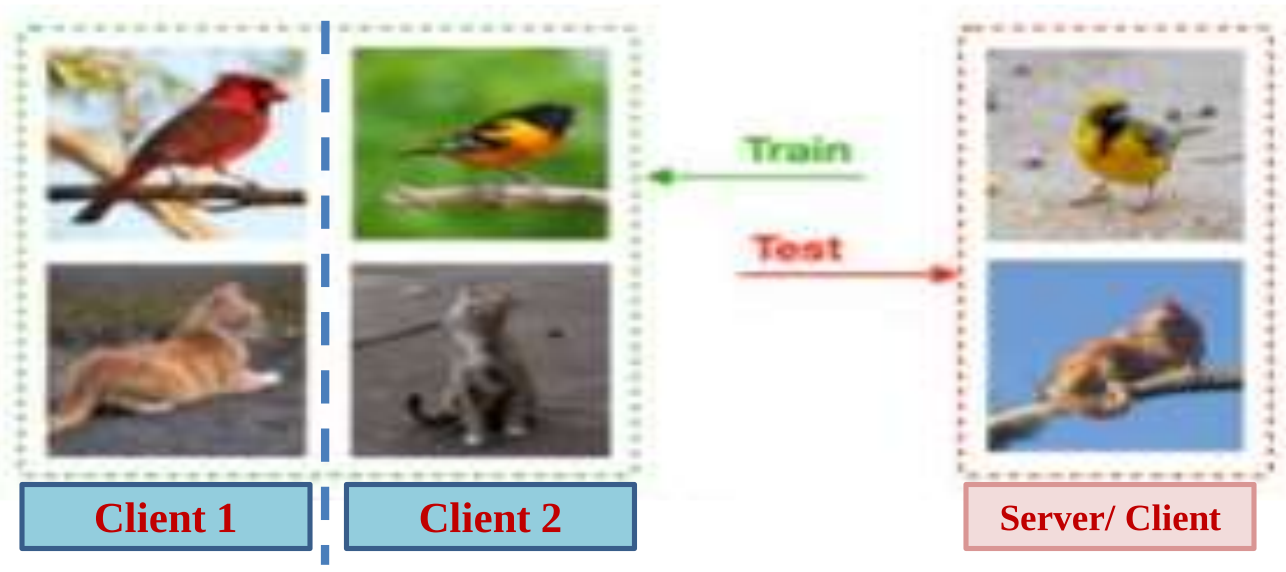
Towards Causal Federated Learning For Enhanced Robustness and Privacy

Sreya Francis, Irene Tenison, Irina Rish
MILA, University Of Montreal

Problem Statement



- Federated learned models are prone to learn easier–to-fit spurious correlations
- Hence fail to generalize out of their training distribution (OOD)
- Poor generalization can lead to higher risks of privacy attacks

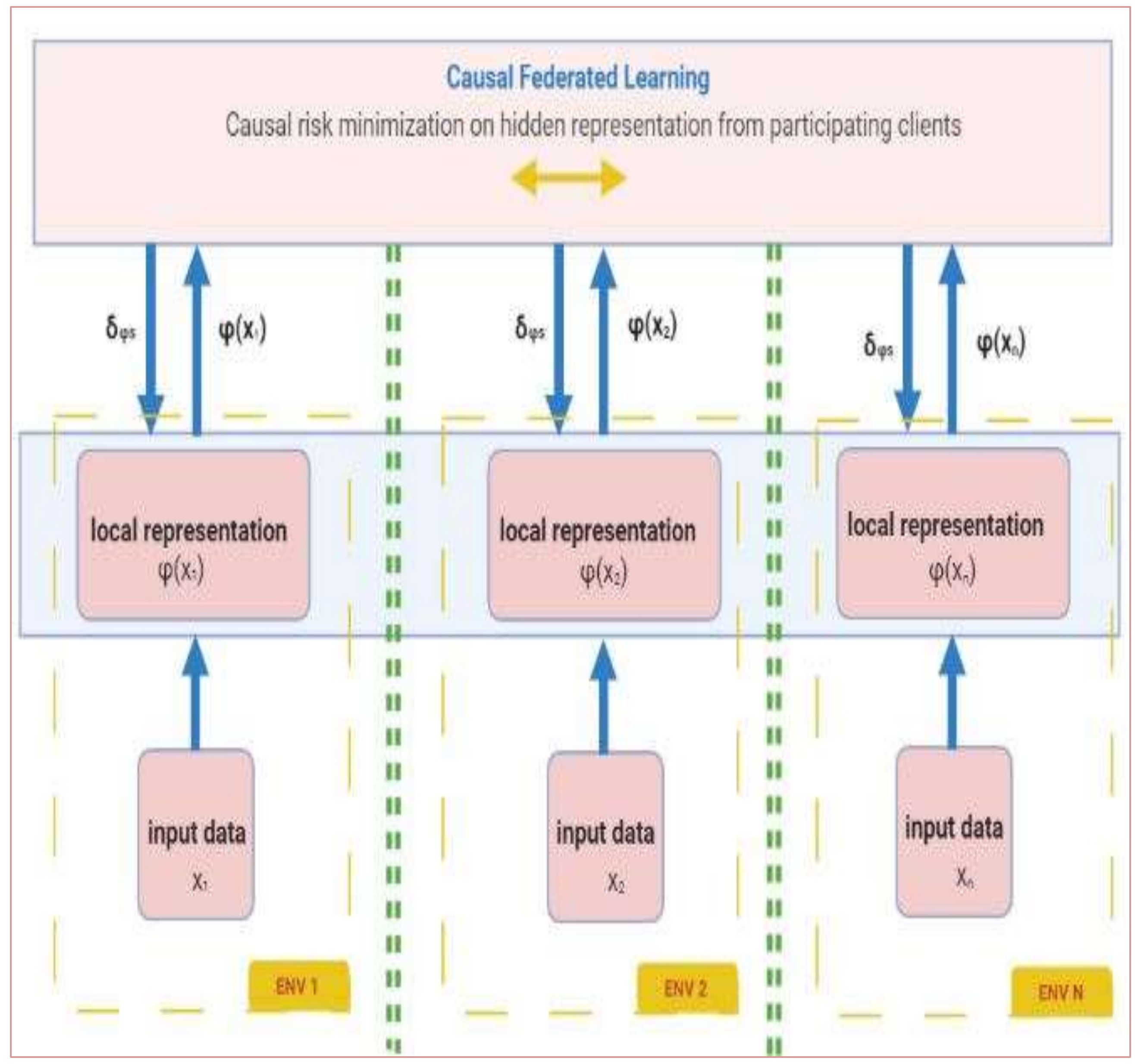
Correlation Vs Causation

- Spurious correlations are correlations that we do not expect to hold in the future use cases
- Minimizing individual client training error leads to absorbing all the correlations found in training data.
- However, spurious correlations stemming from data biases are unrelated to the causal explanation
- Problem: identify which properties of the training data describe spurious correlations (landscapes and contexts), and which properties represent the phenomenon of interest (animal shapes).

Causal Risk Minimization (CRM)

- Learn invariant / causal features common to all participating client environments
- Find a data representation such that the optimal classifier on top of that representation matches for all participating client environments.
- Keeping the client data private, we propose 2 approaches to enhance OOD (Out of Distribution) Accuracy and privacy of the final learned model.

Approach 1 - CausalFed



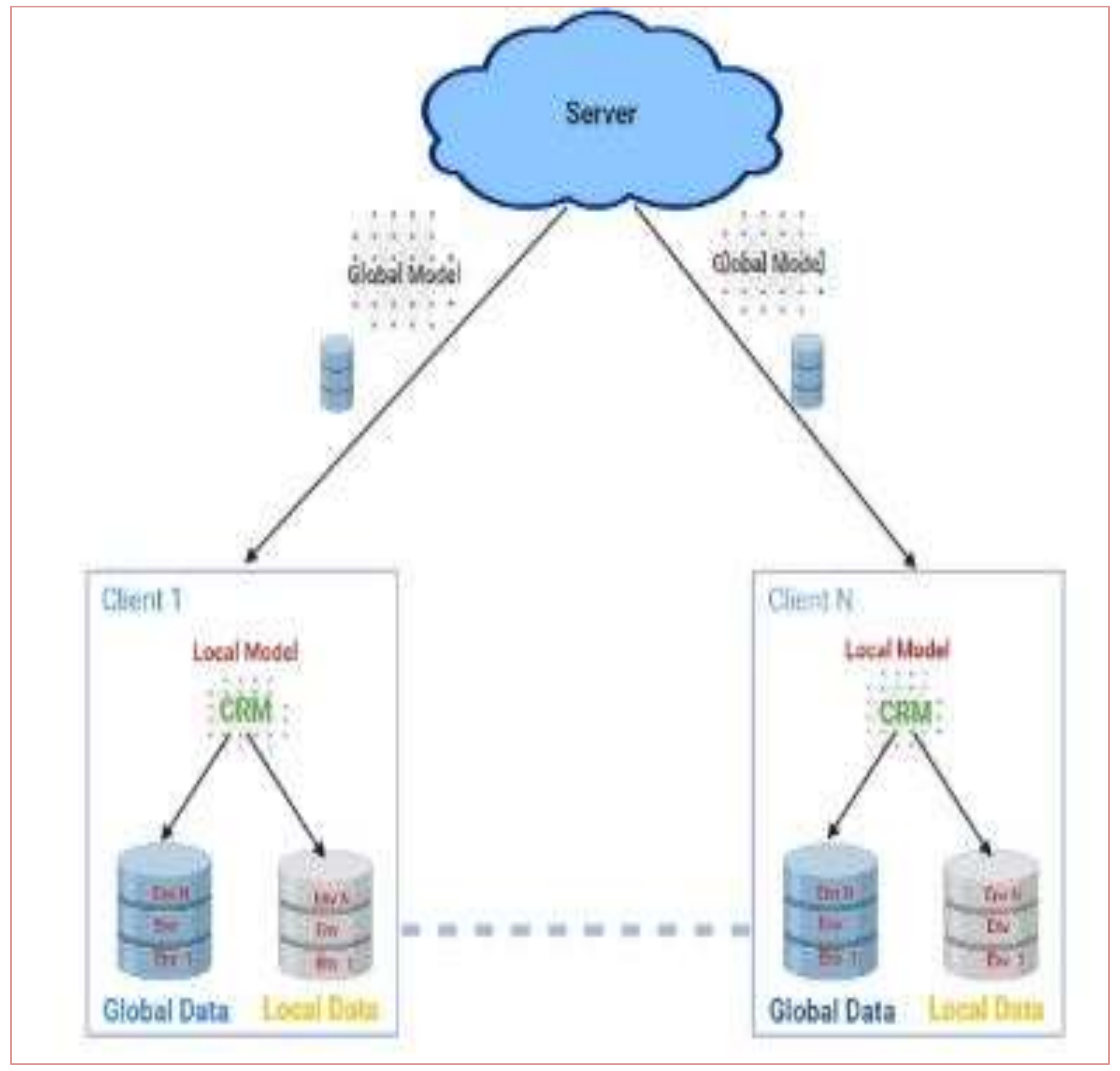
Algorithm 1 CausalFed

ServerCausalUpdate:
Initialize $\mathbf{W}_0^s$
for each server epoch, t = 1,2,..k **do**
  Select random set of S clients
  Share initial model with the selected clients
  for each client $k \in S$ **do**
    $(\phi(x_t^k), \mathbf{Y}^k) \leftarrow \textit{ClientRepresentation}(k, \mathbf{W}_t^k)$
    Evaluate loss $\mathcal{L}_k$
  end for
  $\mathcal{L}_s = \sum_k^S \mathcal{L}_k + \lambda \sum_k^S \left\|\nabla \mathcal{L}_k\right\|^2$
  $\mathbf{W}_{t+1}^s \leftarrow \mathbf{W}_t^s - \eta \nabla \mathcal{L}_s$
end for
$\mathbf{W}_t^k \leftarrow \textit{ClientUpdate}(\nabla \mathcal{L}_s)$

ClientRepresentation($\mathbf{W}_t^k$):
if k is first client to start training **then**
  $\mathbf{W}_t^k \leftarrow$ initial weights from server
else
  $\mathbf{W}_t^k \leftarrow \mathbf{W}_{t-1}^{k-1}$ from the previous $\textit{ClientUpdate}(\nabla \mathcal{L}_s)$
end if
for each local client epoch, i=1,2,..k **do**
  Calculate hidden representation $\phi(x_t^k)$
end for
return $\phi(x_t^k)$ and $\mathbf{Y}^k$ to server

ClientUpdate:
for each client $k \in S$ **do**
  $\mathbf{W}_{t+1}^k \leftarrow \mathbf{W}_t^k - \eta \nabla \mathcal{L}_s$
end for
return $\mathbf{W}_{t+1}^k$ to server

Approach 2 - CausalFedGSD



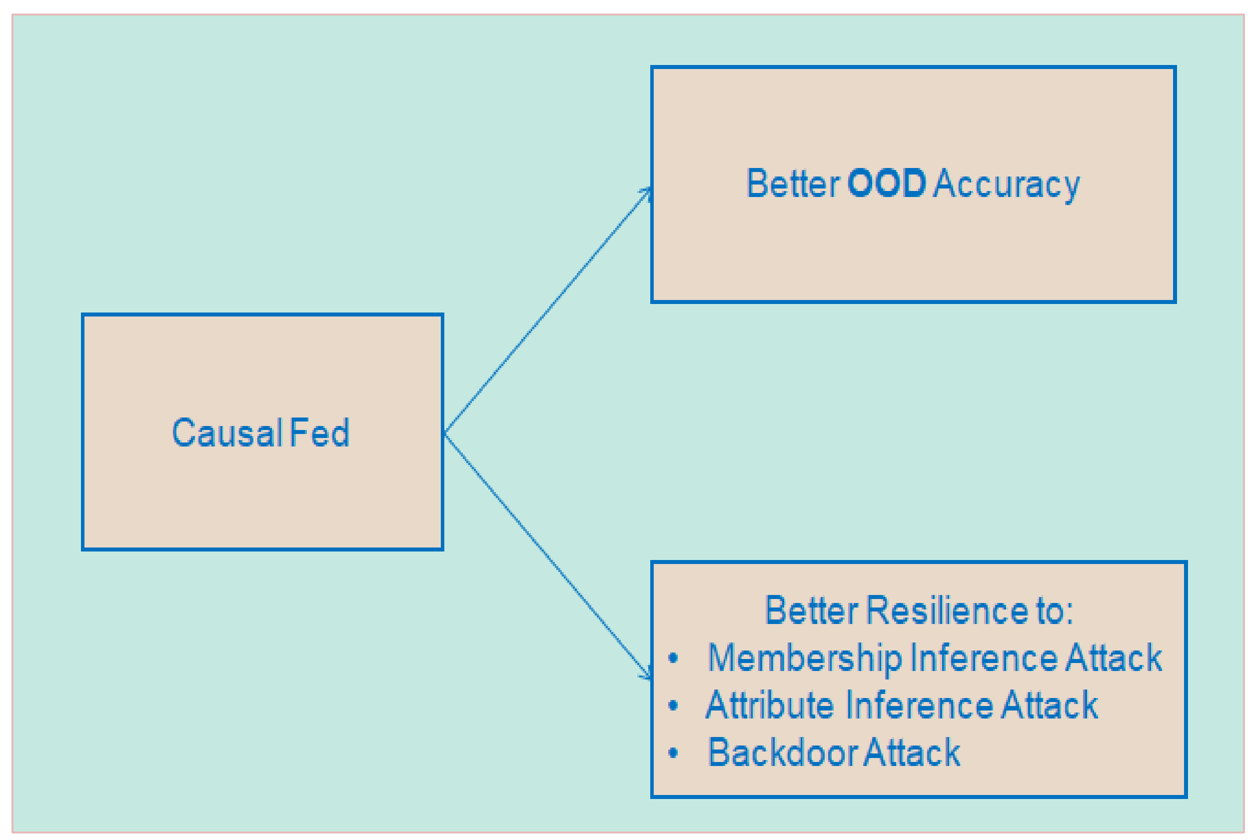
Algorithm 2 CausalFedGSD

ServerUpdate:
G ← distribution over all environments present in server
Initialize $\mathbf{W}_0$
Initialize random portion of G as G_0
for each server epoch, t = 1,2,..k **do**
  Select random set of S clients
  Share G_0 and initial model with the selected clients
  for each client $k \in S$ **do**
    $\mathbf{W}_{t+1}^k = \textit{ClientUpdate}(k, \mathbf{W}_t)$
  end for
  $\mathbf{W}_{t+1} = \sum_{k=1}^K \frac{n_k}{n} \mathbf{W}_{t+1}^k$
end for

ClientUpdate(W):
$\mathcal{E}_{tr} \in$ [Client Env] $\bigcup$[Global Env]
for each local client epoch, t=1,2,..k **do**
  $L_{\mathrm{IRM}}(\Phi, \mathbf{W}_t^k) = \sum_{e \in \mathcal{E}_{tr}} R^e(\mathbf{W} \circ \Phi) + \lambda \cdot \mathbb{D}(\mathbf{W}, \Phi, e)$
  $\mathbf{W}_t^k = \mathbf{W}_t^k - \eta \nabla L_{\mathrm{IRM}}(\mathbf{W}_t^k)$
end for
return W to server

Experimental Setup (Ex: Colored MNIST)

Advantages of Proposed Approaches



OOD (Out Of Distribution) Test Results

Dataset	Arch	Fed-Avg	Fed-ERM	CausalFed-RM	CausalFed-IRM
Colored MNIST	ResNet18	11%	10.2 %	**65.62 %**	60.3 %
Rotated MNIST	ResNet18	82.7%	82.9 %	**90.2 %**	89.1 %
Rotated FMNIST	LeNet	72%	71.6 %	**74.6 %**	73.9 %

Inference Attack Leakage

Dataset	Fed-Avg	Fed-ERM	CausalFed-RM	CausalFed-IRM
Colored MNIST	79.21 %	79.45 %	58.57 %	56.9 %
Rotated MNIST	84.4 %	85.24 %	68.3 %	64.4 %
Rotated FMNIST	76.61 %	78.23 %	57.55 %	55.7 %