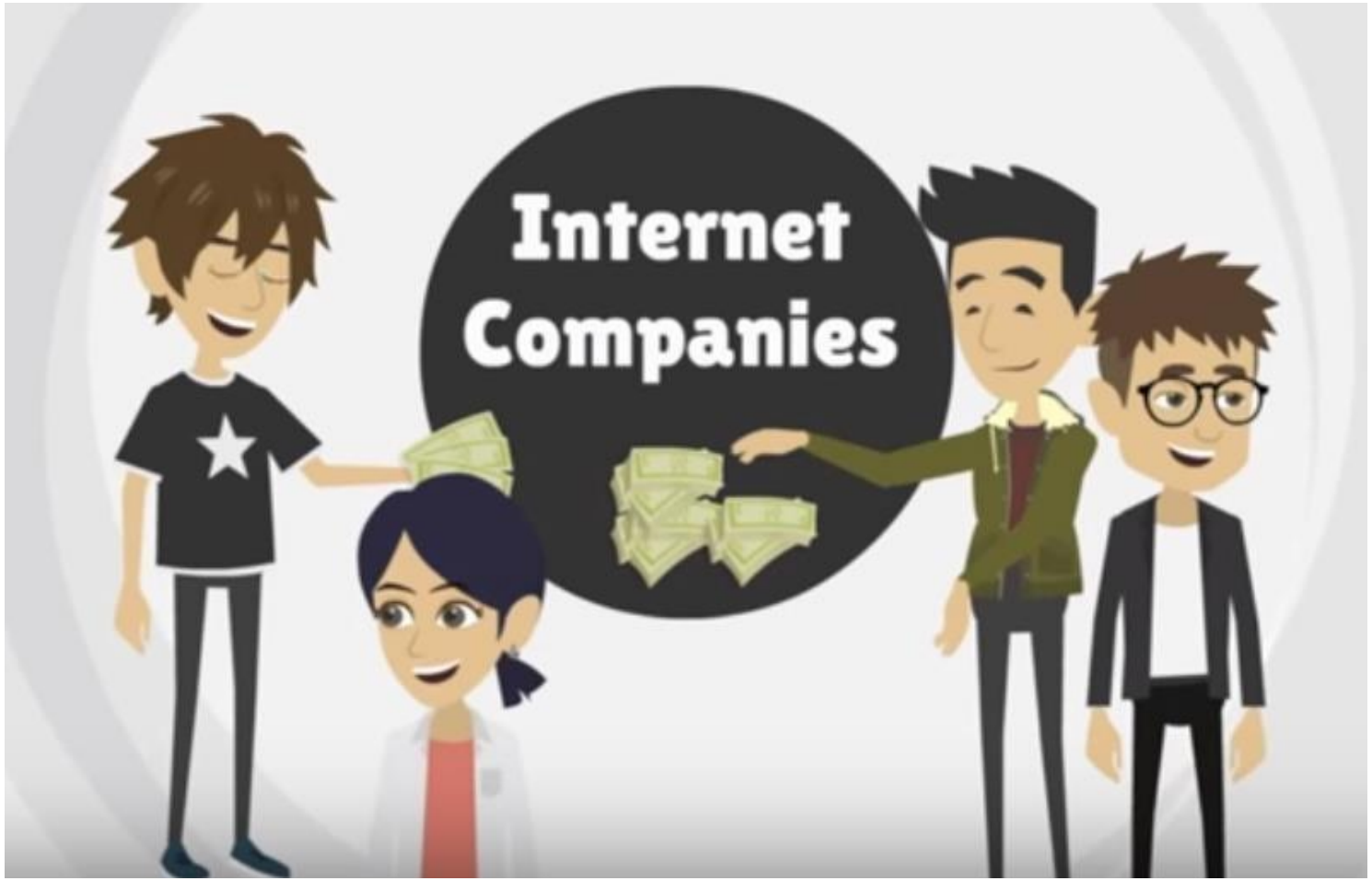
Causal FedBlock : Blockchain based Causal Approach to Robust Federated Learning with Fair Incentivization

Sreya Francis

01

Introduction

Current Scenario



Existing technology - Issues

- Data collection means adopted right now is incredibly privacy invasive
- We give our data for free in return of a free service
- Latency issues
- High transfer costs
- Centralized ownership (Users don’t participate in the current system)
- Very limited data for healthcare research

Mila

Current Issues

- Privacy Concerns
  - We don’t have control over the data we generate!
- We are losing one source of natural income
  - Data is our natural resource and we own it
- Sensitive Product Problem - some services are creepy
  - High risks of theft, embarrassment, resale ……etc
- Centralized control by Big Tech Giants
  - All of our data are controlled by tech giants like google, facebook

Mila

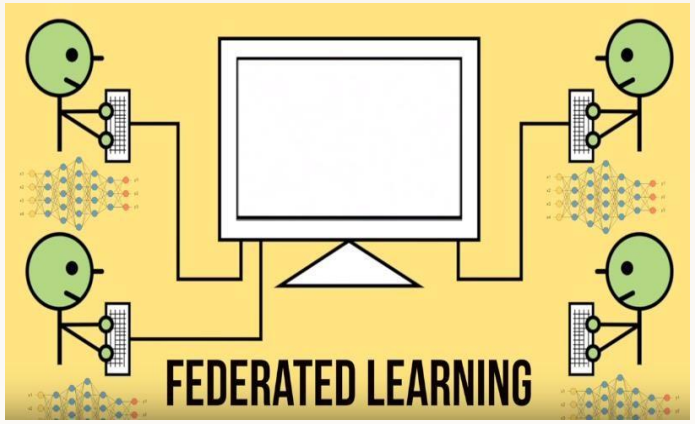
How can we solve this?

- Enhance user privacy
  - We should control our data
- We should be rewarded for the data we own
  - Rewards based on data quality and quantity
- Decentralized power
  - Everyone has control over their data
- Enhance production of sensitive products/models
  - Enhanced privacy would make it easier to collect data related to sensitive fields like healthcare

Mila

Ingredients for the solution

Federated Learning



BlockChain

Causal Machine Learning

Cryptography

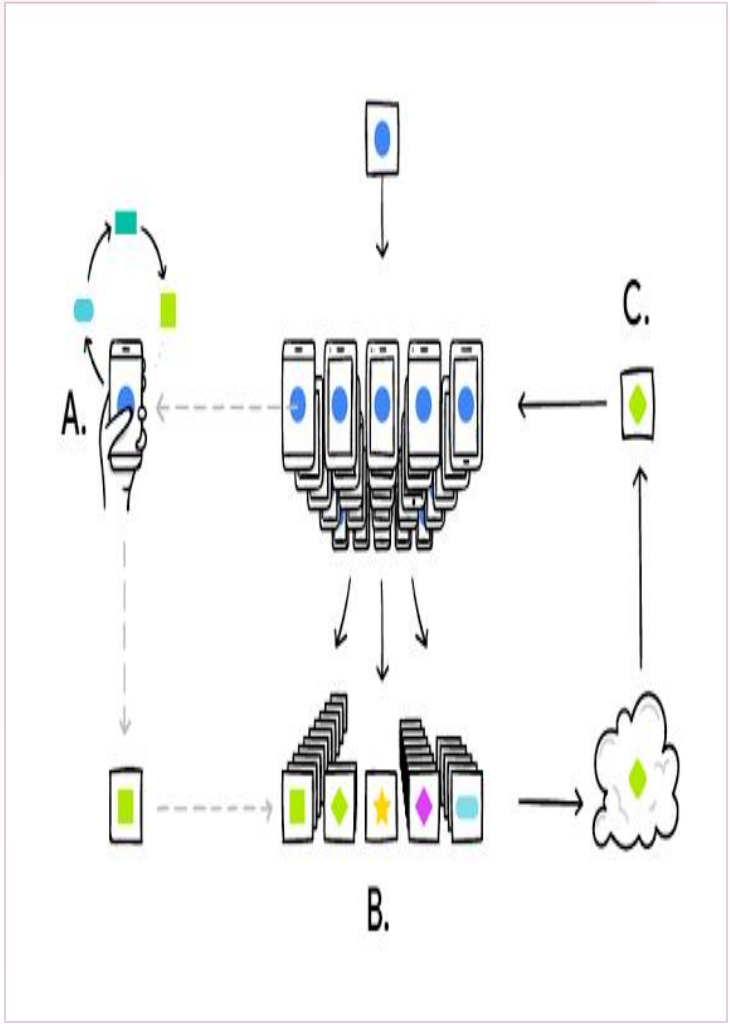
1 Federated Learning

- What is Federated Learning?
- How does it work?
- Federated Learning Platforms

Mila

Federated Learning - Definition

- Idea: machine learning over a distributed dataset
- Federated computation: where a server coordinates a fleet of participating devices to compute aggregations of devices' private data.
- Federated learning: where a shared global model is trained via federated computation.
- Definition: training a shared global model, from a federation of participating devices which maintain control of their own data, with the facilitation of a central server.



Mila

Federated Learning – Brief stepwise overview

- Step 1: Users download a Model
- Step 2: Users train the Model on their own data.
- Step 3: Users upload their Gradients to a server
- Step 4: Gradients are added up to protect privacy.
- Step 5: The Model is updated with the Global Model.

Mila

Federated Learning – Algorithm

Server

Until Converged:

1. Select a random subset (e.g.200) of the (online) clients

2. In parallel, send current parameters **θ(t)** *to those clients*

Selected client K

1.Receive **θ(t)** from server.
2. Run some number of minibatch SGD steps, producing **θ'**
3. Return **θ'-θ(t)** to server.

3. **θ(t+1)** *=* **θ(t)** *+ data-weighted average of client updates*

Mila

Federated Learning – Pros & Cons

Pros:

- Enhanced User Privacy: Users keep their data secret

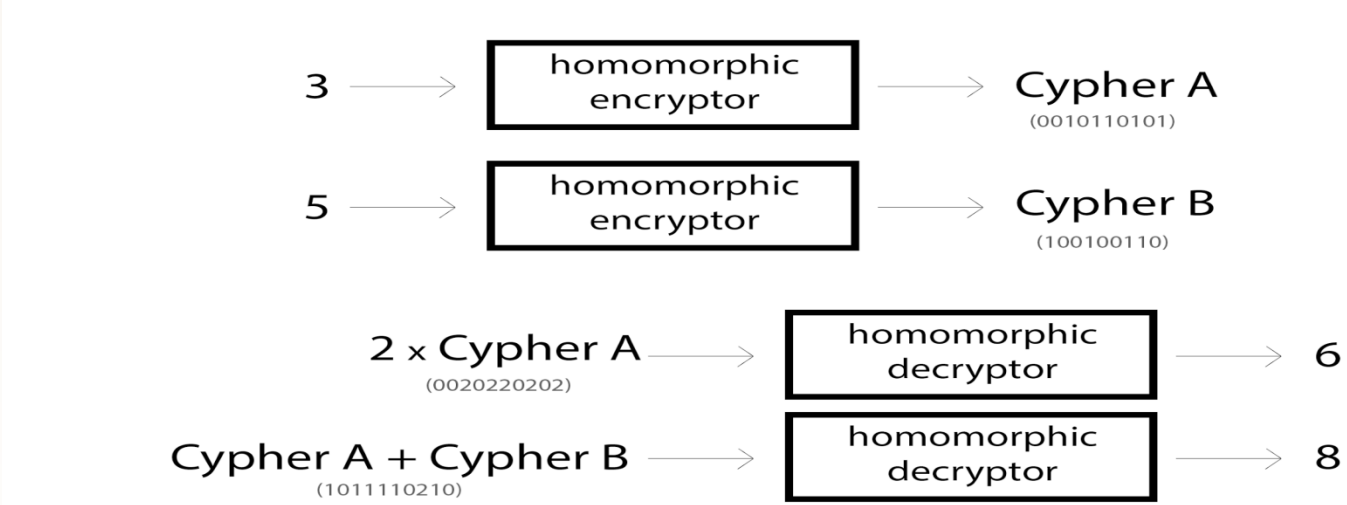
Cons:

- Privacy: Gradients give hints about data
- Theft: Participants can steal the updated models
- No Sensitive Products: Because of theft/privacy issues

Mila

One Possible Solution: Homomorphic Encryption

What is Homomorphic Encryption?



- Homomorphically encrypt the user **gradients** so that the gradient privacy is preserved
- Privacy-Preserving **Deep Neural Network model** (2P-DNN) based on the **Paillier** Homomorphic **Cryptosystem** could be used to enhanced global model privacy
- Hence there is no issue of theft or privacy intrusion in this case

Mila

Reward Calculation

Possible way

- Based on user model performance on validation set
  - To evaluate the validity of user data, we can run a validation check on the user model based on a trusted validation set.
  - Based on the performance on validation set, the users can be rewarded.
  - If the validation accuracy goes below a specified threshold, the data is rejected.
- Pros
  - An easy and fast way to calculate user reward immediately after client side training

- Cons
  - At any given iteration, an honest gradient may update the model in an incorrect direction, resulting in a drop in validation accuracy.
  - This is confounded by the problem that clients may have data that is not accurately modeled by our trusted validation set

Mila

Issues with data in FL

What can go wrong?

- Gamber attack
  - User/Attacker can randomly pick data and maliciously change them
  - User can give garbage input
  - User/Attacker give data that does not contribute to the model
- Omniscient attack
  - Attackers are supposed to know the gradients sent by all the workers
  - Use the sum of all the gradients, scaled by a large negative value,
  - And replace some of the gradient vectors.
- Gaussian attack
  - Some of the gradient vectors are replaced by random vectors sampled from a Gaussian distribution with large variances.

Mila

How to counter adversaries?

Possible ways

- Based on KRUM Algorithm
  - Uses the Euclidean distance to rank the gradients
  - Determines which gradient contributions are removed
  - the top f contributions to the client model that are furthest from the mean client contribution are removed from the aggregated gradient

- Pros
  - specifically designed to counter adversaries in federated learning.

- Cons
  - Not an absolute measure of user contribution
  - Implementation is a bit complicated

Mila

How to ensure validity of gradients?

Possible ways

Let us assume that q out of n vectors are Byzantine/incorrect, where q < n:

Krum's Algo in a nutshell:

$$Krum(\{\bar{v}_i : i \in [n]\}) = \bar{v}_k,$$
$$k = \underset{i \in [n]}{\operatorname{argmin}} \sum_{i \to j} \|\bar{v}_i - \bar{v}_j\|^2,$$

where $i \to j$ is the indices of the $n-q-2$ nearest neighbours of $\bar{v}_i$ in $\{\bar{v}_i : i \in [n]\}$ measured by Euclidean distance.

- Works only when q < n
- Ensure upto 33% protection against adversarial attacks
- Best solution proposed till date

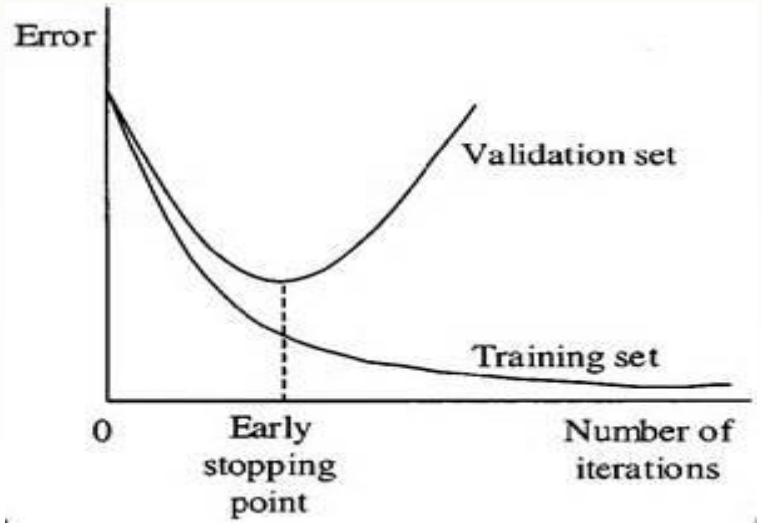
Mila

Proposed Solution to the User Reward Issue

- Data Cost
  - Each User calculates his/her data cost
  - Class id – Ci, Number of samples - Nci
  - Cost per user -> ∑j=1 to k (j*Nci)

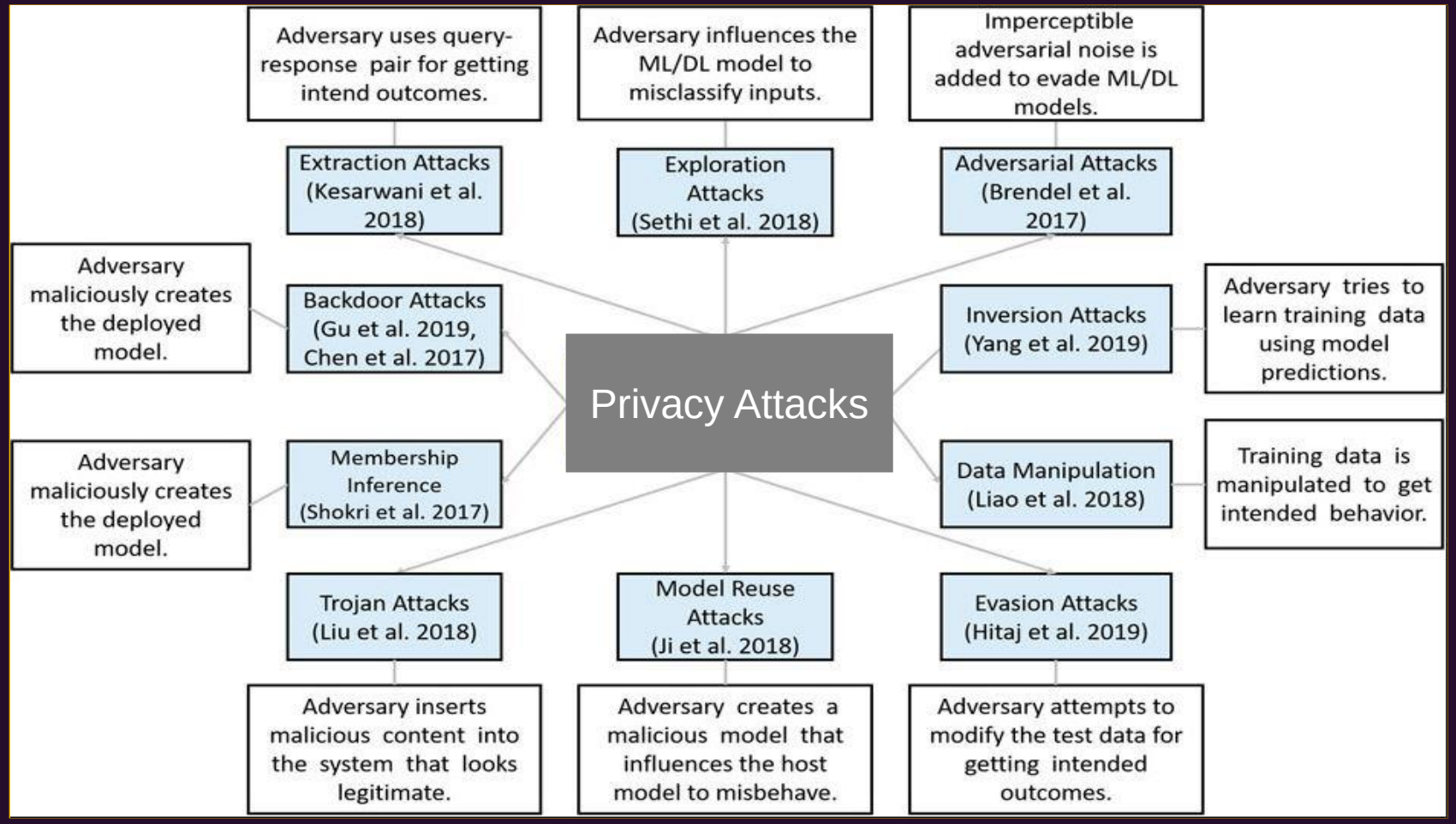
- Generate validation set
  - Based on parameters passed to calculate data cost
  - Automatically generate a validation set with some random samples
  - Samples pertain to user specified classes

- Training



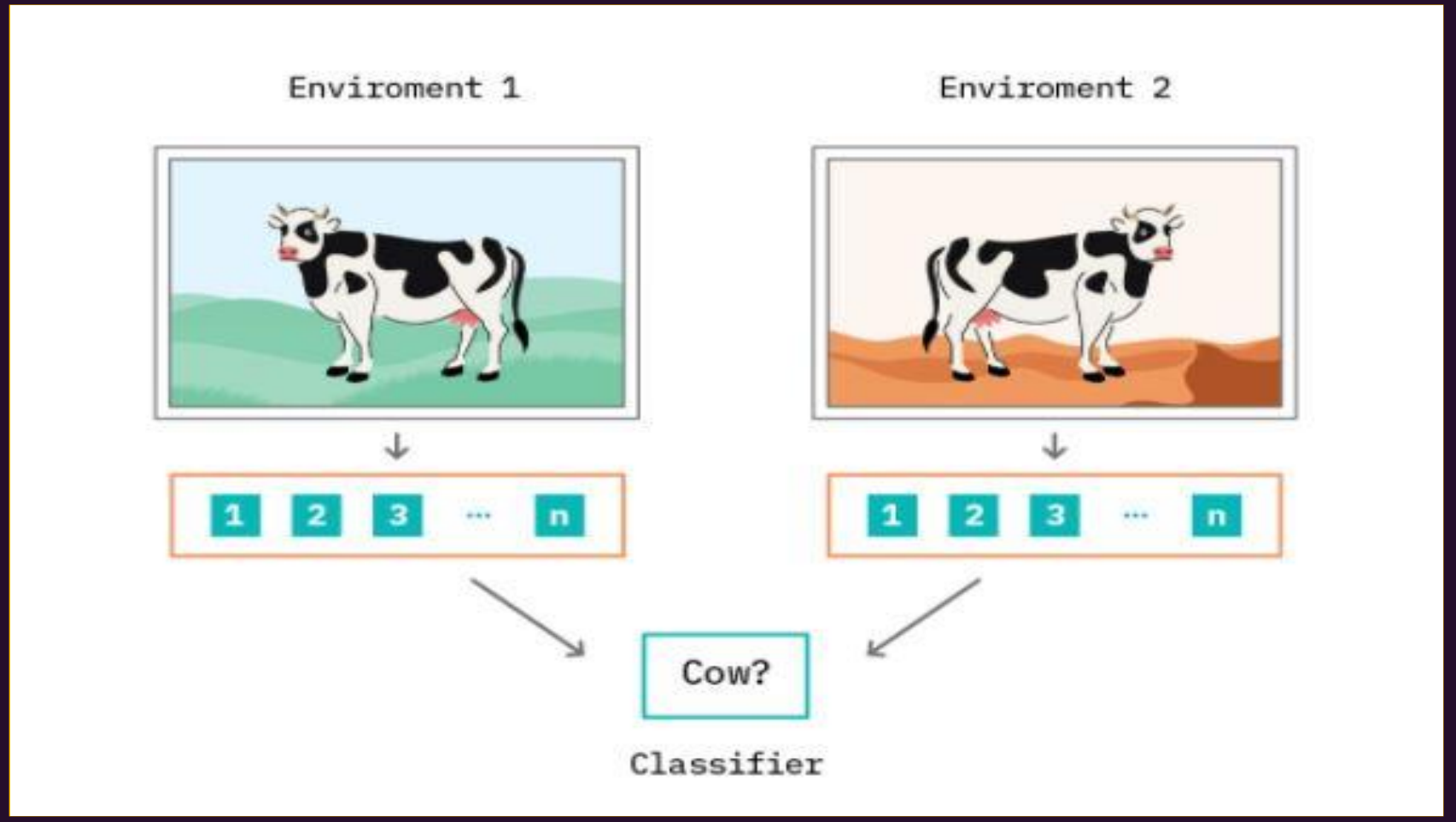
•Stop training before the model over-fits data
•If validation error doesn't go down, user entry is wrong
•If validation error goes down, user entry is valid and pay the user based on calculated data cost

Open Issues In FL



Mila

Open Issues In FL



Mila

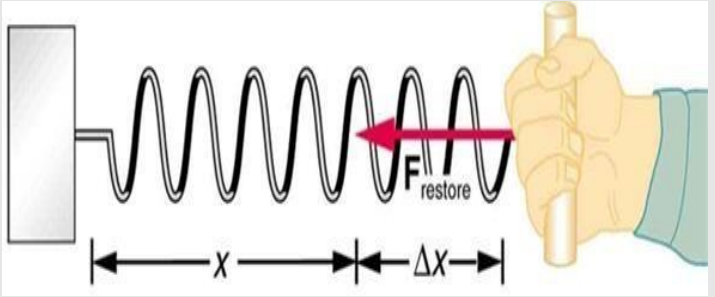
How Can Causal Learning Help?

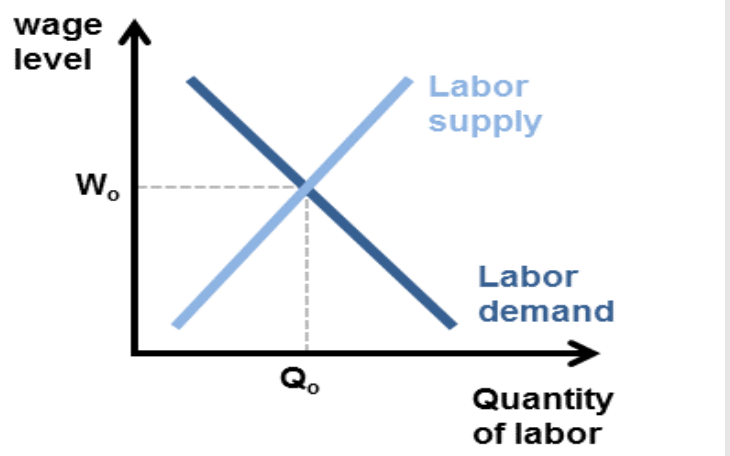
- What is causality?
- Potential Outcomes Framework
- Unobserved Confounds / Simpson’s Paradox
- Structural Causal Model Framework

Cause and Effect

- Questions of cause and effect common in biomedical and social sciences
- Such questions form the basis of almost all scientific inquiry
  - Medicine: drug trials, effect of a drug
  - Social sciences: effect of a certain policy
  - Genetics: effect of genes on disease

- **So what is causality?**
- **What does it mean to *cause* something?**





What is causality?

- A fundamental question
- Surprisingly, until very recently---maybe the last 30+ years---we have not had a mathematical language of causation. We have not had an arithmetic for representing causal relationships.

The Three Layer Causal Hierarchy

Pearl, Theoretical Impediments to Machine Learning with Seven Sparks from the Causal Revolution, arXiv:1801.04016v1. 11 Jan 2018

Level	Typical Activity	Typical Question	Examples
1. Association $P(y \mid x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y \mid do(x), z)$	Doing, Intervening	What if? What if I do X?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x \mid x', y')$	Imagining, Retrospection	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past 2 years?

A practical definition

Definition: T causes Y iff

changing T leads to a change in Y,

keeping everything else constant.

The causal effect is the magnitude by which Y is changed by a unit change in T.

Called the "interventionist" interpretation of causality.

**Interventionist* definition [http://plato.stanford.edu/entries/causation-mani/]

Keeping everything else constant: Imagine a *counterfactual* world

"What-if" questions

Reason about a world that does not exist.

- What if a system intervention was not done?
- What if an algorithm was changed?
- What if I gave a drug to a patient?

- What is causality?
- Potential Outcomes Framework
- **Unobserved Confounds / Simpson's Paradox**
- Structural Causal Model Framework

The Simpson's paradox:

Consider success rate analysis of kidney stone treatment based on Observational Data

Kidney Stones	Treatment (A)	Treatment (B)
Success Rate	**78%**	**83%**

Which treatment do you think is better?

What if there are unobserved features that matter?

Simpson (1951)

The Simpson's paradox: Treatment B is better overall, but worse for each subgroup!

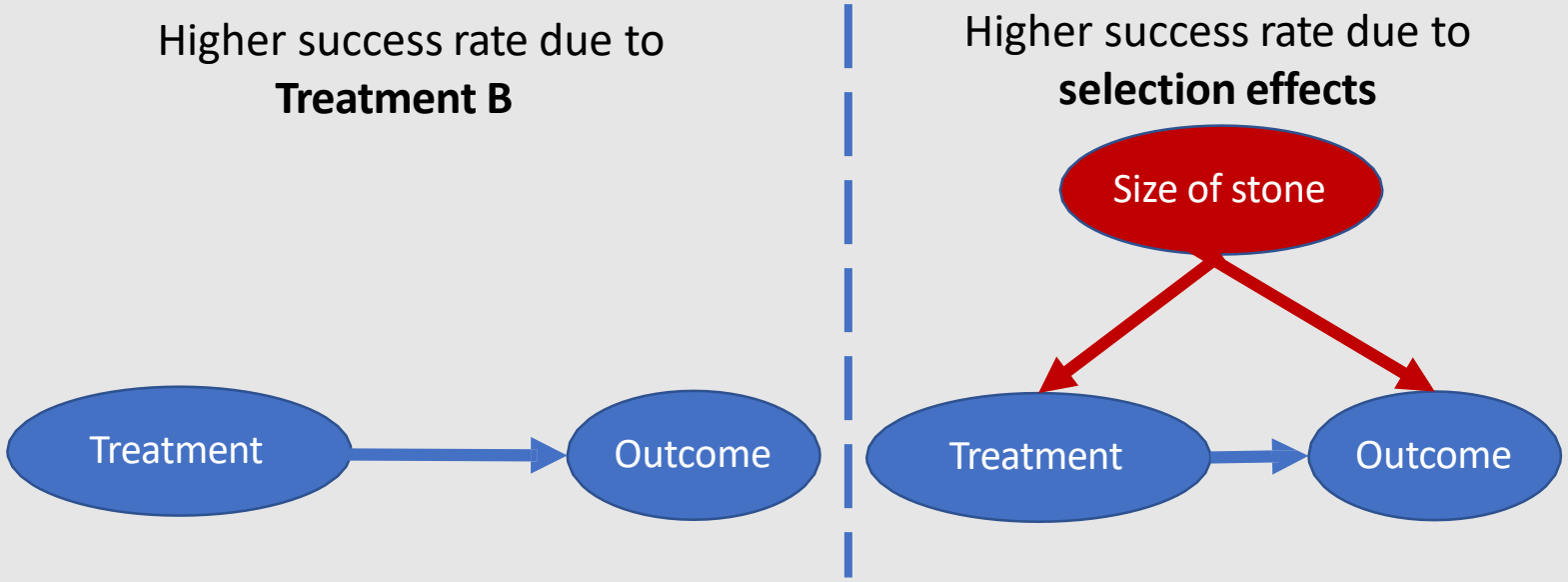
	Treatment (A)	Treatment (B)
Success Rate for small stones	93%	87%
Success Rate for large stones	73%	69%
Overall Success Rate	**78%**	**83%**

So, which is better?

Simpson (1951)

From metrics to decision-making

- **Did the change to treatment B increase success rate for the patients?**
- Answer (as usual):
- Maybe, maybe not (!)



E.g., Treatment B is shown at a different time than A.

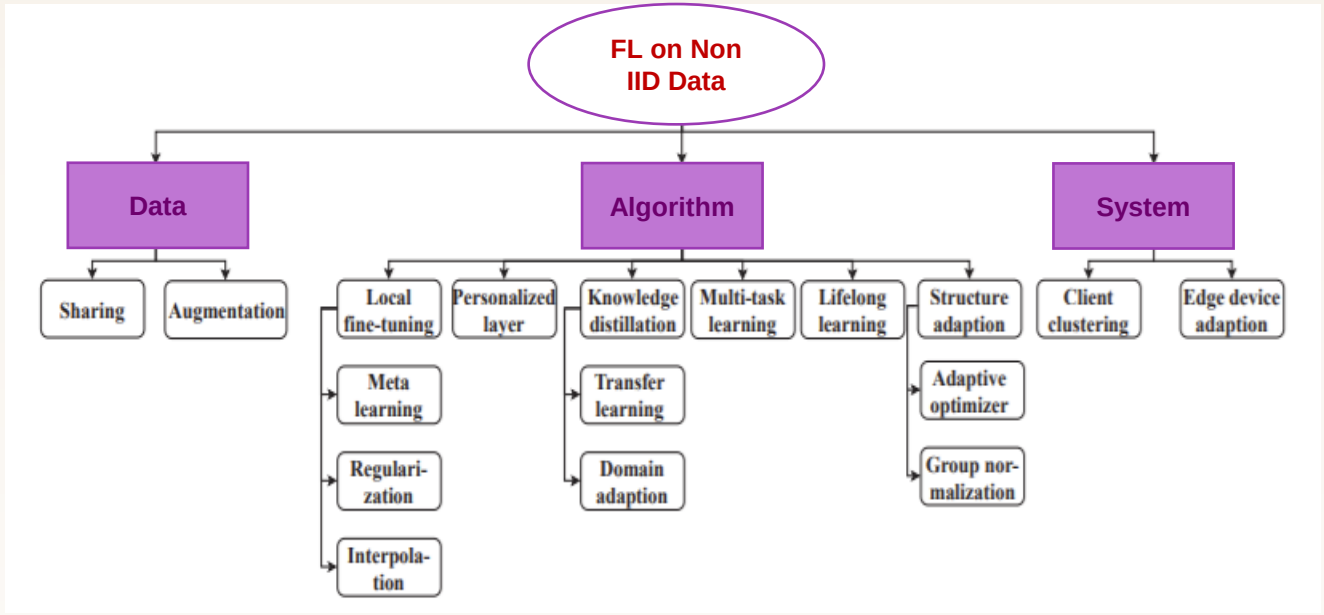
There could be other hidden causal variations.

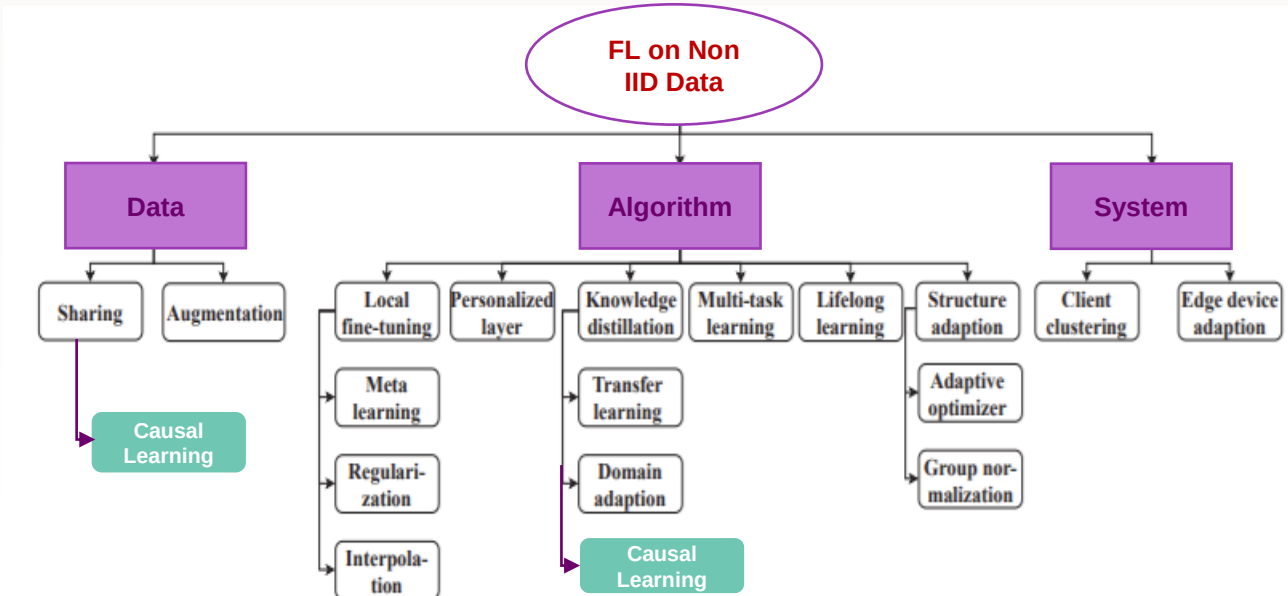
Making sense of such data can be too complex.

Unobserved confounds are a threat to causal reasoning!

02

Some approaches to merge CL and FL


FL on Non IID Data
Data
Algorithm
System
Sharing
Augmentation
Local fine-tuning
Personalized layer
Knowledge distillation
Multi-task learning
Lifelong learning
Structure adaption
Client clustering
Edge device adaption
Meta learning
Transfer learning
Adaptive optimizer
Regulari-zation
Domain adaption
Group nor-malization
Interpola-tion


FL on Non IID Data
Data
Algorithm
System
Sharing
Augmentation
Local fine-tuning
Personalized layer
Knowledge distillation
Multi-task learning
Lifelong learning
Structure adaption
Client clustering
Edge device adaption
Causal Learning
Meta learning
Transfer learning
Adaptive optimizer
Regulari-zation
Domain adaption
Group nor-malization
Interpola-tion
Causal Learning

FL- Bilevel Constrained Optimization Problem
Optimal Data Rep and Optimal Classifier

IRM – Single Optimization Loop with trick of using constant classifier and introducing new penalty term to loss function

Sum over env(Error + Penalty)

Measures how well model performs in each env + measures how much perf could be improved in each env with one gradient step.
Punish each gradients when large improvement in 1 env within 1 epoch of learning.

02

Proposed Approach 1 - CausalFed

Causal Fed

Algorithm 1 CausalFed

ServerCausalUpdate:

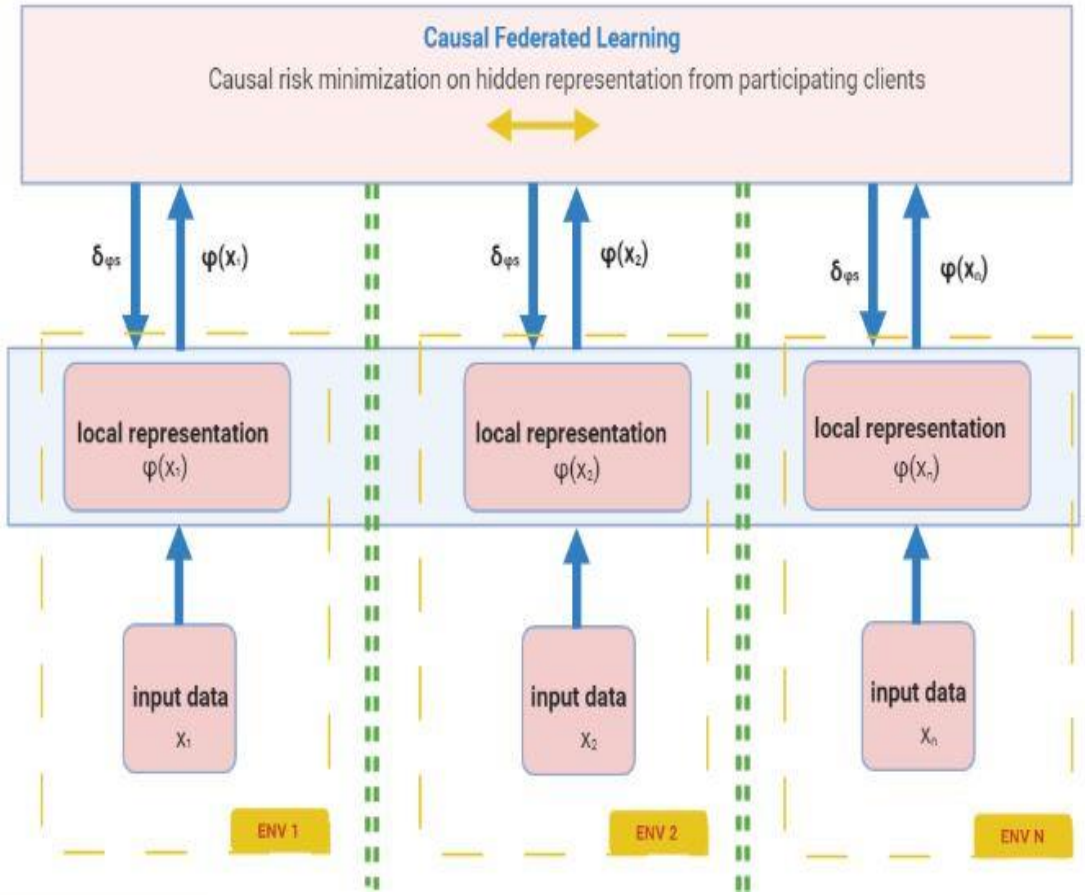
Initialize $\mathbf{W}_0^s$
for each server epoch, t = 1,2,..k **do**
  Select random set of S clients
  Share initial model with the selected clients
  for each client $k \in S$ **do**
    $(\phi(x_t^k), \mathbf{Y}^k) \leftarrow ClientRepresentation(k, \mathbf{W}_t^k)$
    Evaluate loss $\mathcal{L}_k$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

ClientRepresentation($\mathbf{W}_t^k$):

if k is first client to start training **then**
  $\mathbf{W}_t^k \leftarrow$ initial weights from server
else
  $\mathbf{W}_t^k \leftarrow \mathbf{W}_{t-1}^k$ from the previous $ClientUpdate(\nabla \mathcal{L}_s)$
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



Mila

03

Proposed Approach 2 - CausalFedGSD

Causal Fed GSD

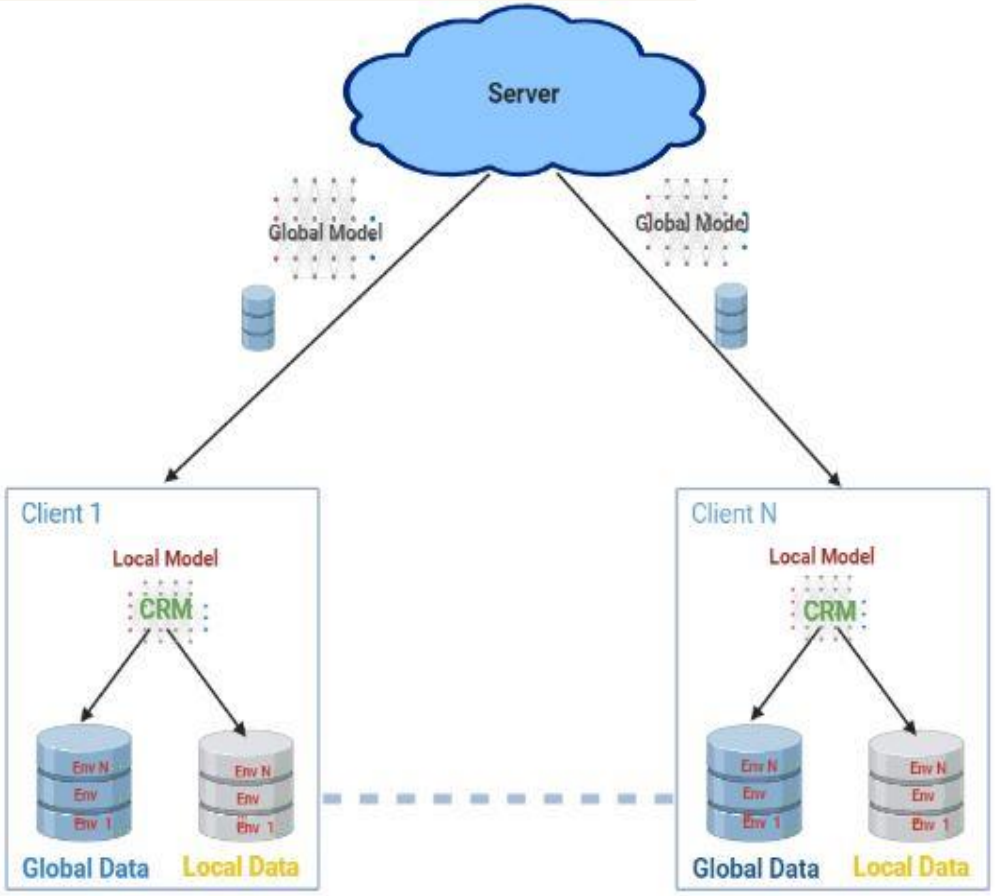
Algorithm 2 CausalFedGSD

ServerUpdate:

G ← distribution over all environments present in server
Initialize 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**
    $w_{t+1}^k = ClientUpdate(k, w_t)$
  end for
  $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
end for

ClientUpdate(w):

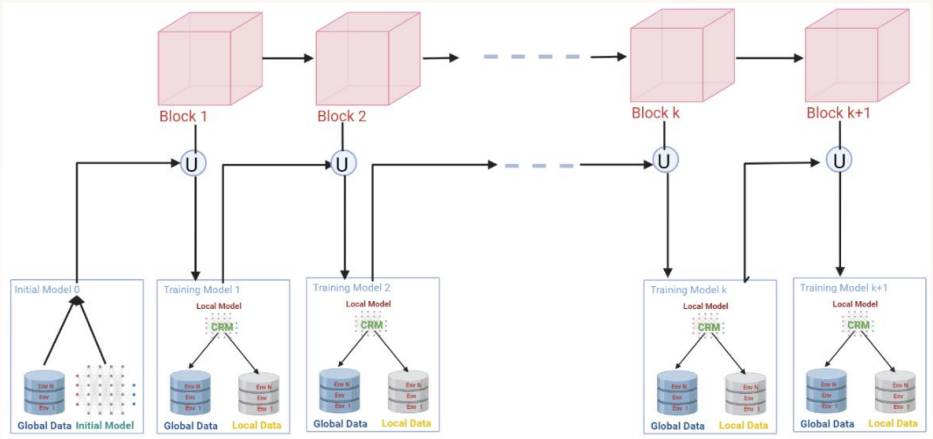
$\mathcal{E}_u \in$ [Client Env] $\bigcup$ [Global Env]
for each local client epoch, t=1,2,..k **do**
  $L_{IRM}(\Phi, w_t^k) = \sum_{e \in \mathcal{E}_u} R^e(w \circ \Phi) + \lambda \cdot \mathbb{D}(w, \Phi, e)$
  $w_t^k = w_t^k - \eta \nabla L_{IRM}(w_t^k)$
end for
return w to server

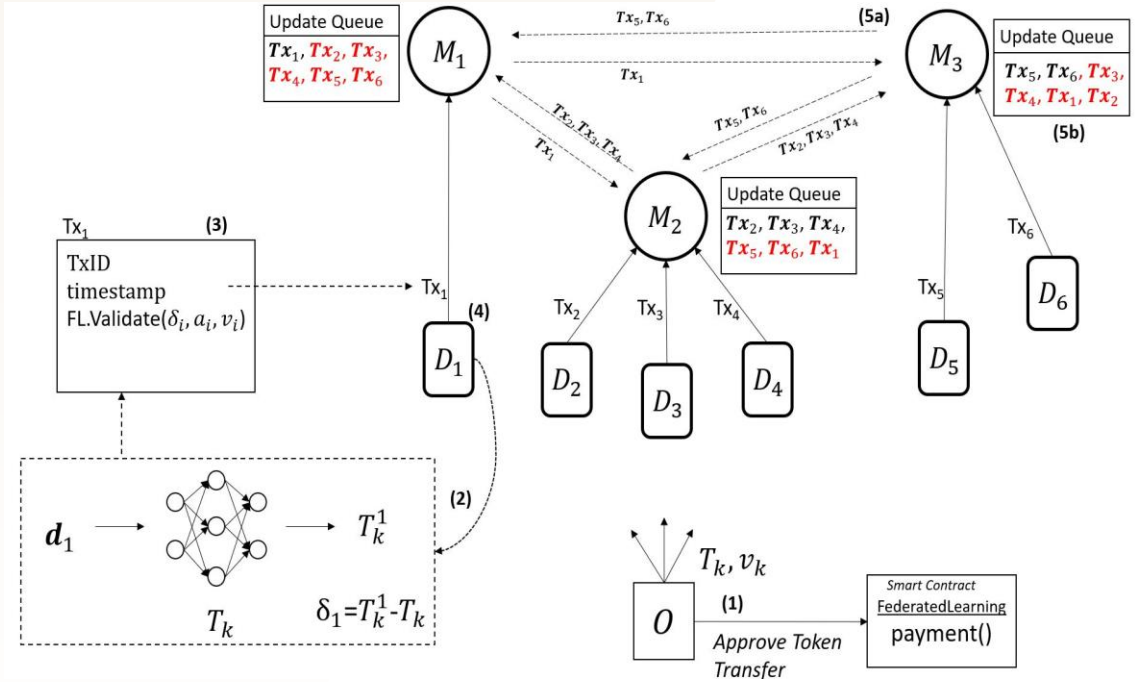


04

CausalFedBlock Architecture

Causal Fed Block

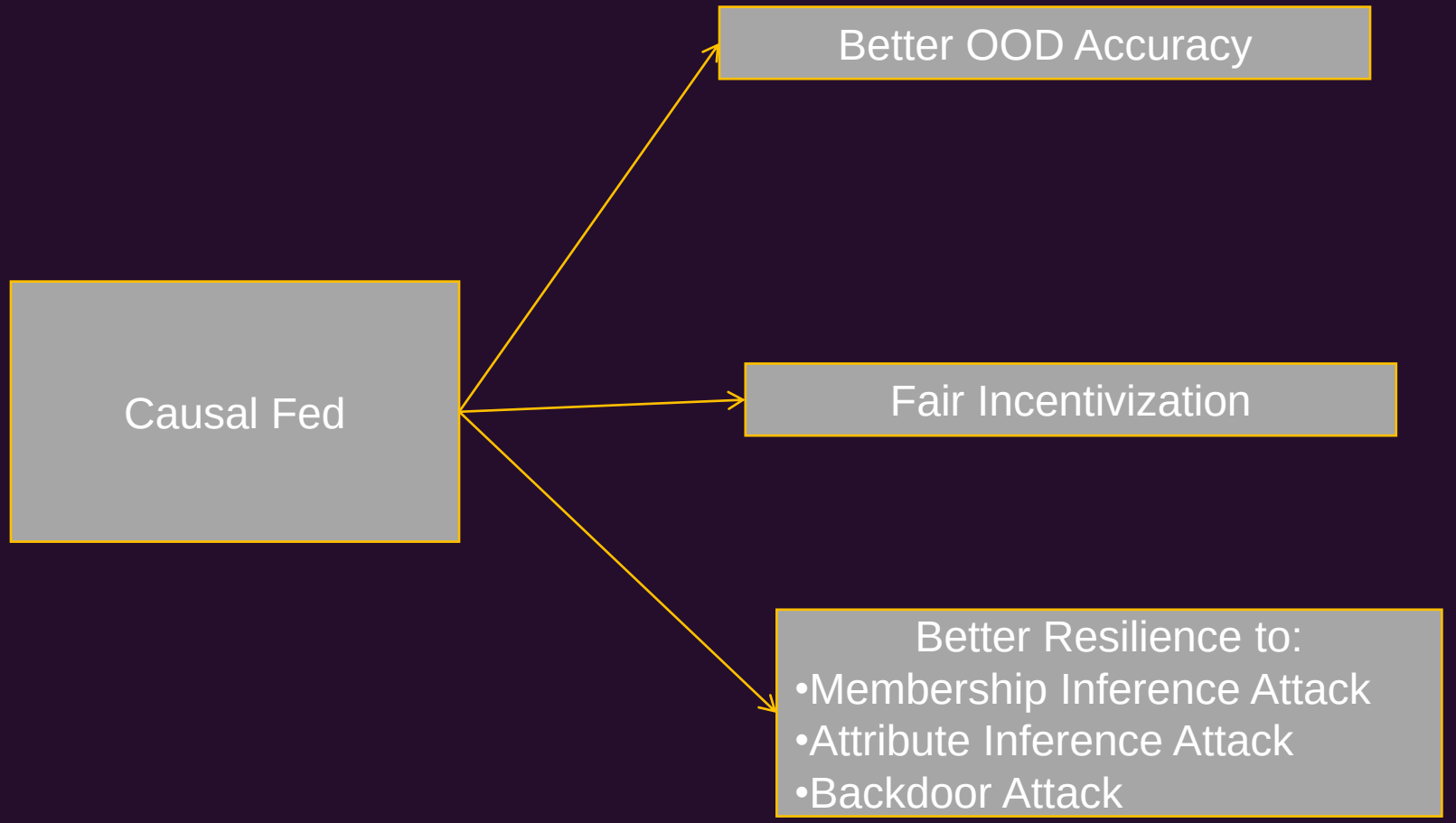




05

Advantages

Advantages of Proposed Approach



Mila

Thank You