Federated Learning

Talay Cheema and Siddharth Swaroop

MLG reading group

11th March 2020

Why learn with distributed data?

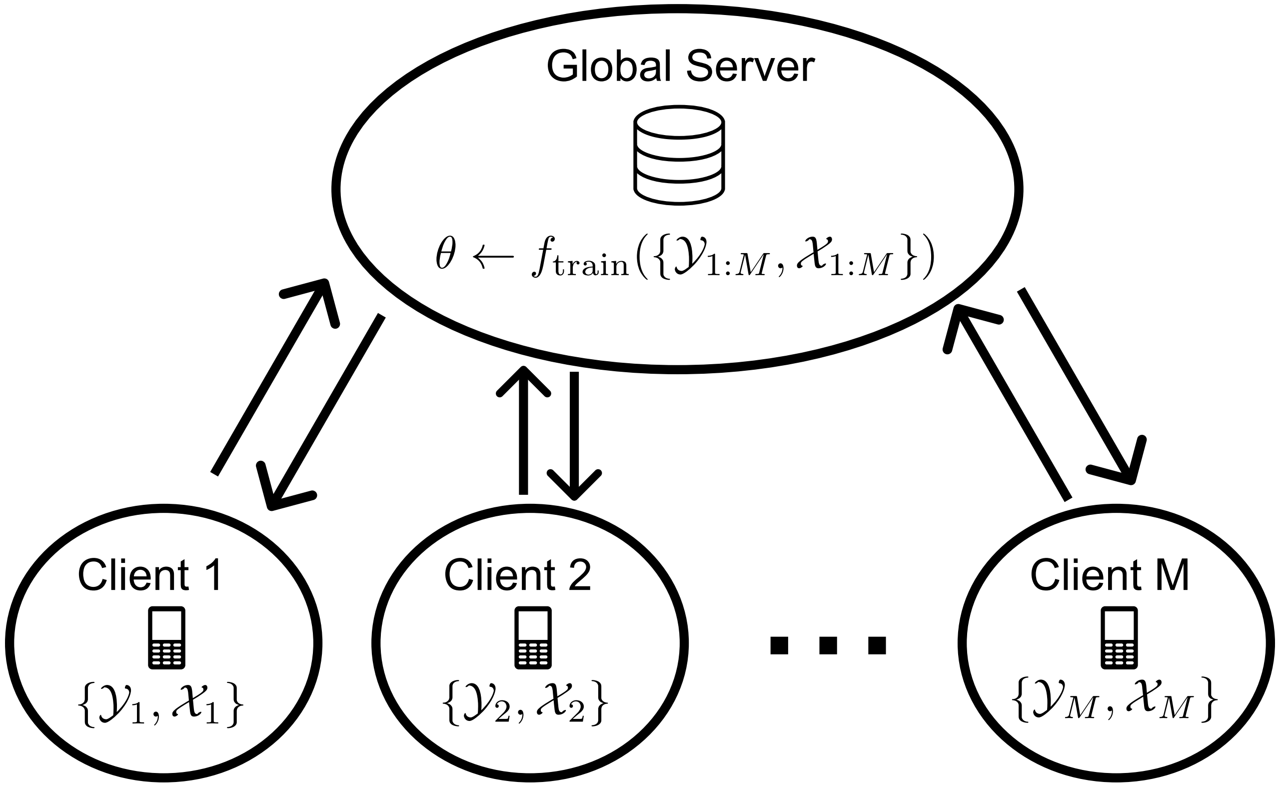
- Data is often distributed across many devices / locations
  - User data on mobile phones
  - Large institutional databases e.g. medical records in hospitals
  - Not P2P

- Communication efficiency is important (big data, low power, low bandwidth)

- Privacy is important – can we get away without asking for user data?

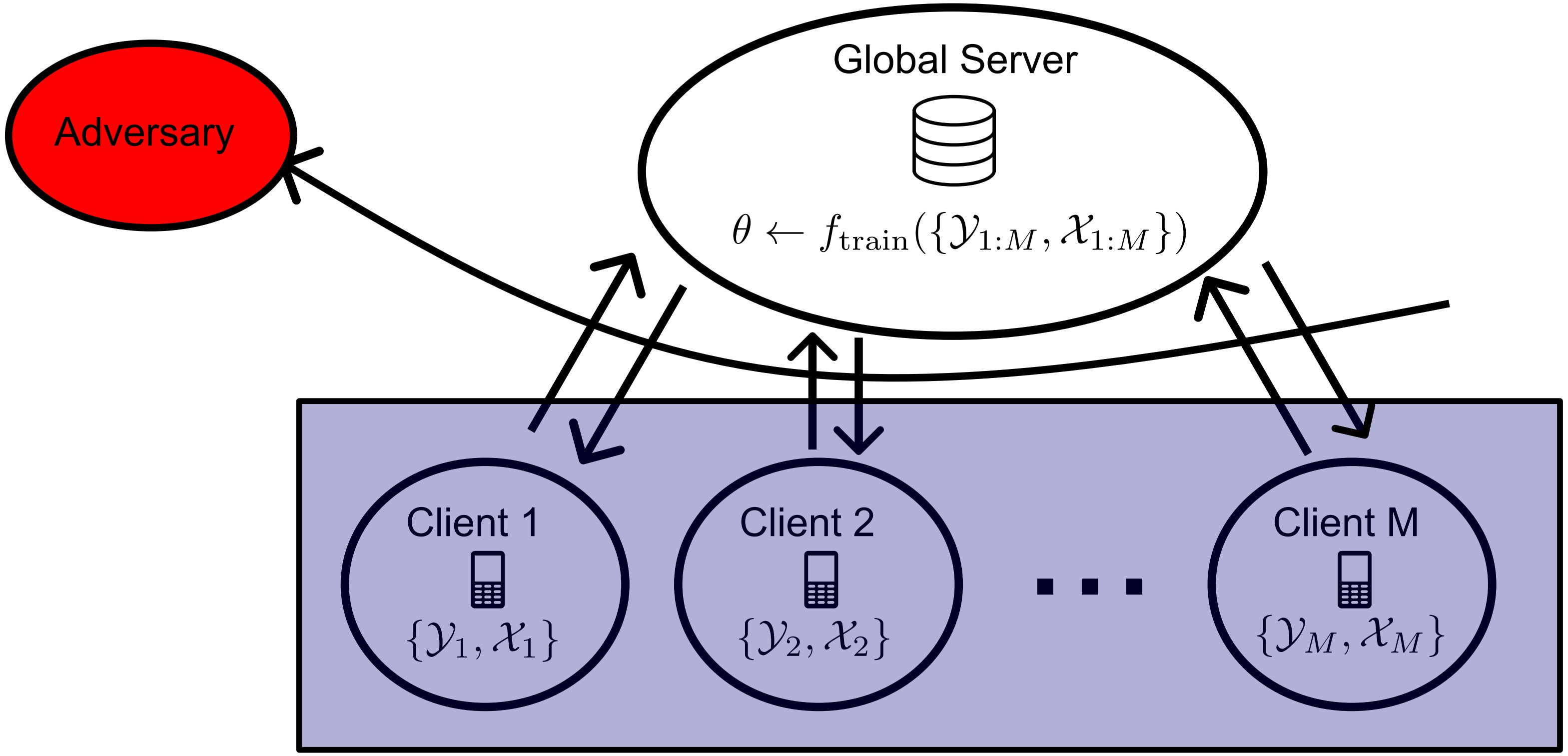
Talk Outline

1. **Motivations and background**
   - Threat models
   - Homomorphic encryption
   - Definition and core challenges
2. SGD-inspired approaches
   - Vanilla SGD
   - Federated Averaging
3. Bayesian federated learning
   - Partitioned Variational Inference
4. Improving security and privacy
   - Secure Multi-Party Computation
   - Differential Privacy

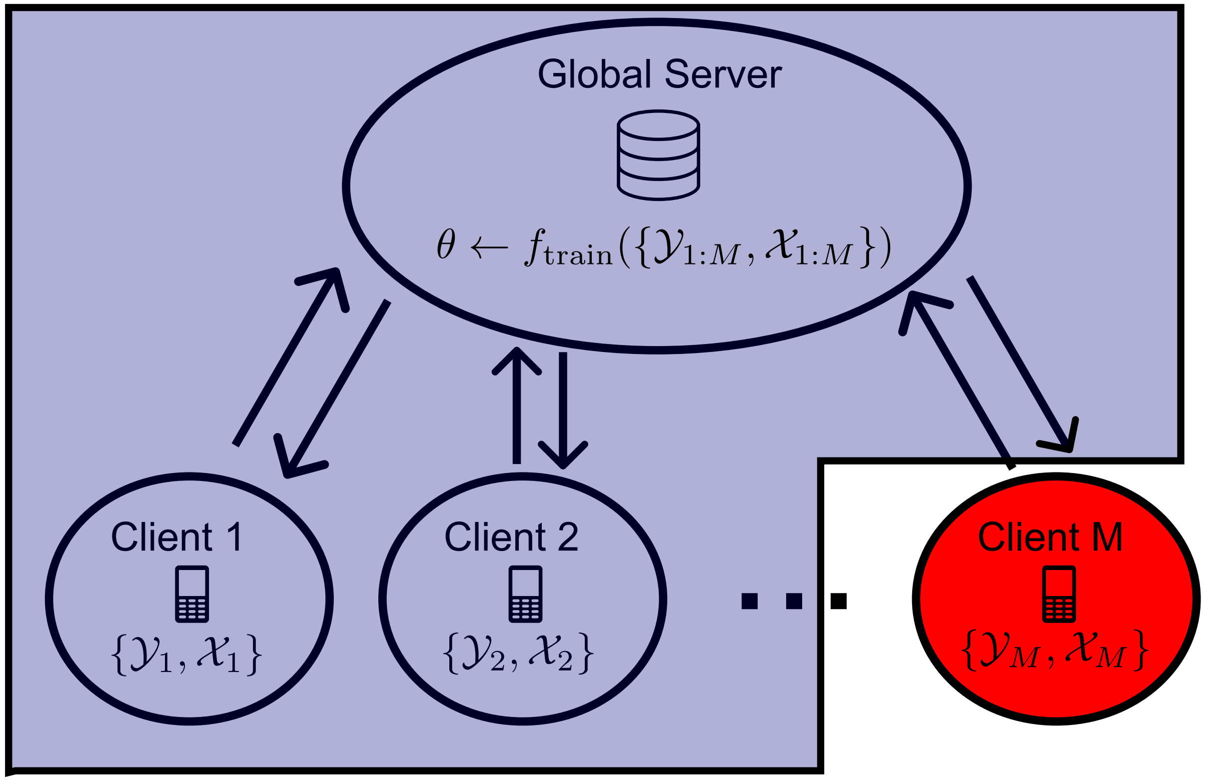


- Learn a global model with parameters θ *efficiently*, *securely* and *fairly* from *private* data
- We will make these terms more precise...

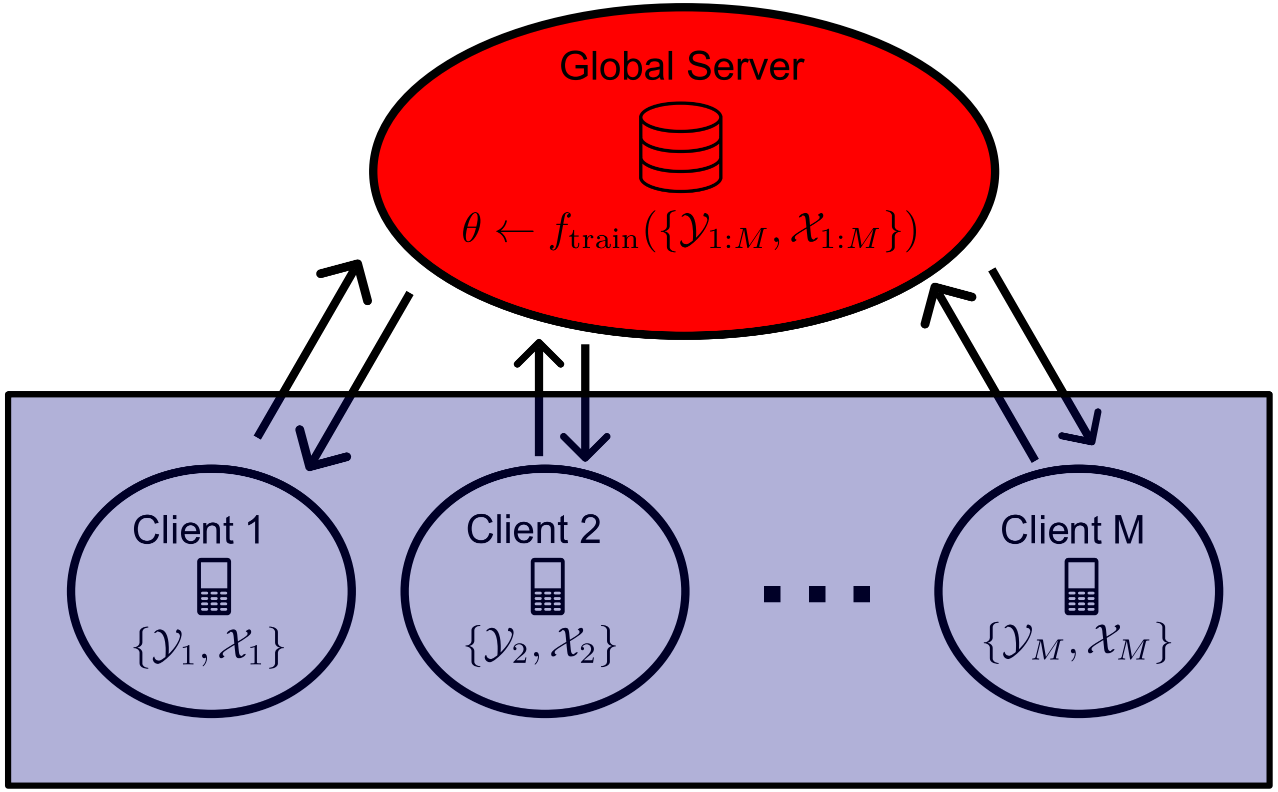
Threat 1 – eavesdropper



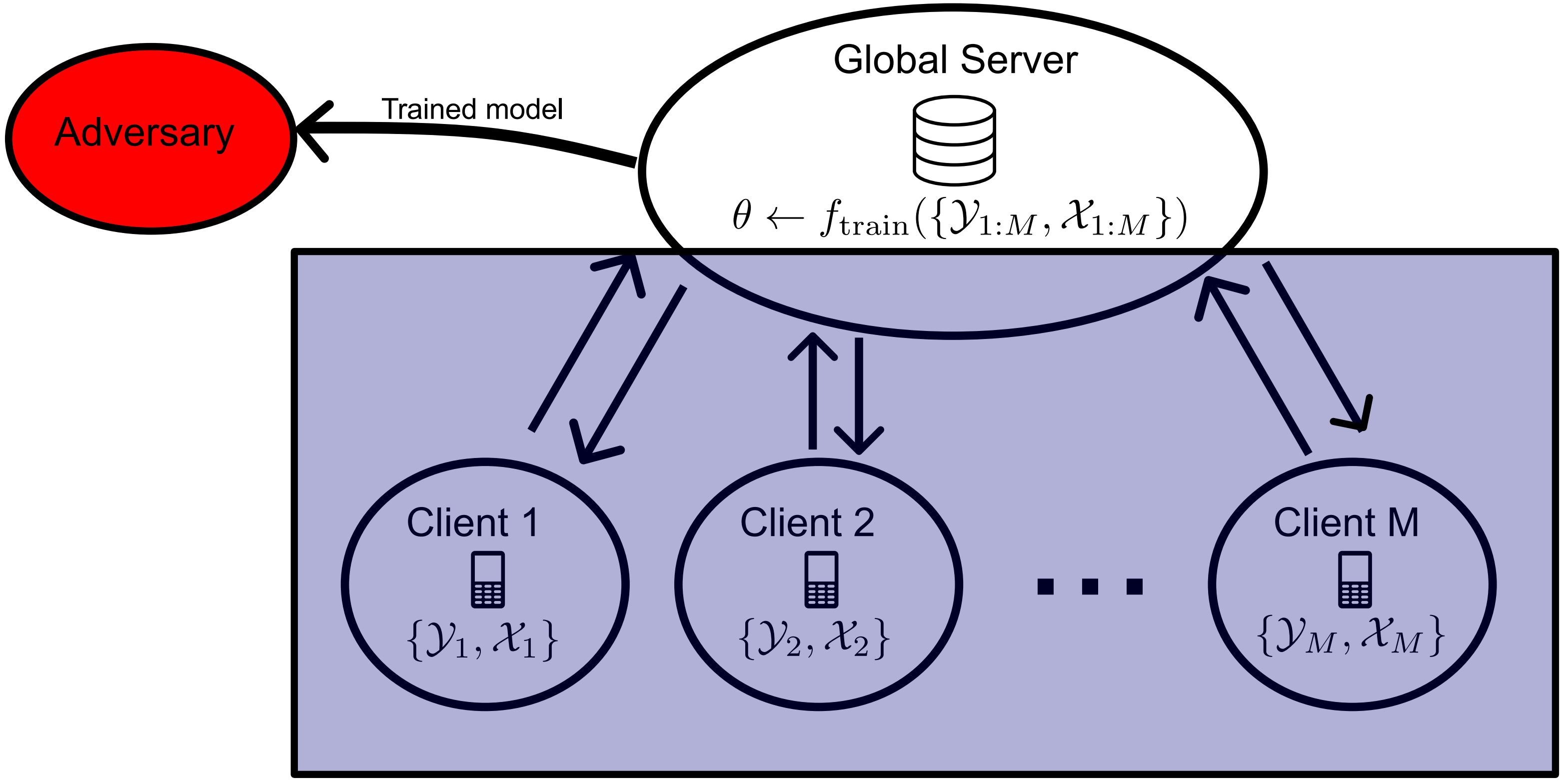
Threat 2 – an adversarial client



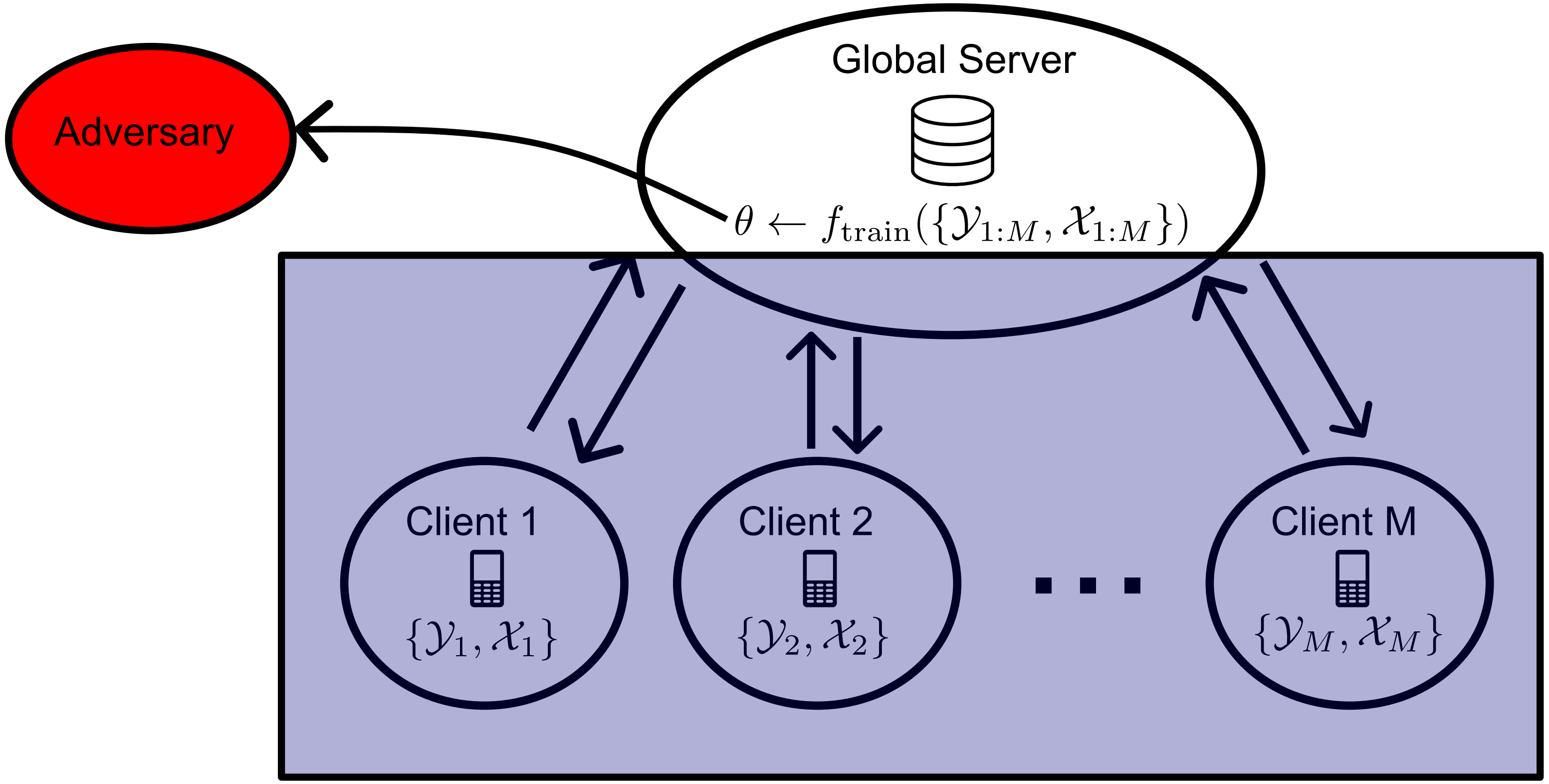
Threat 3 – a curious server

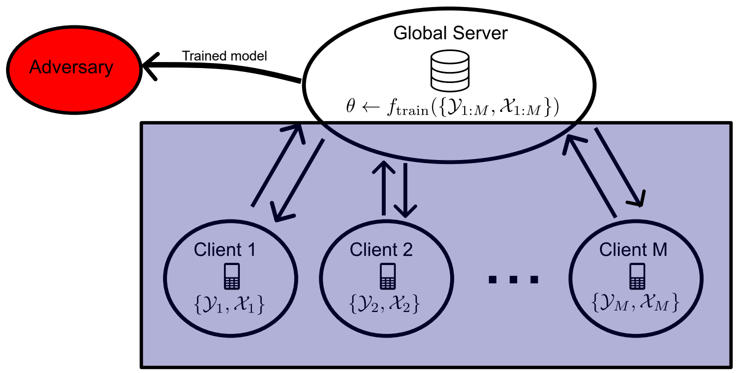


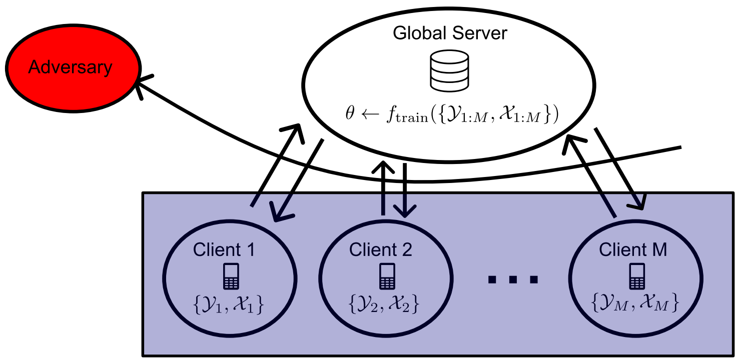
Threat 4.1 – an end user

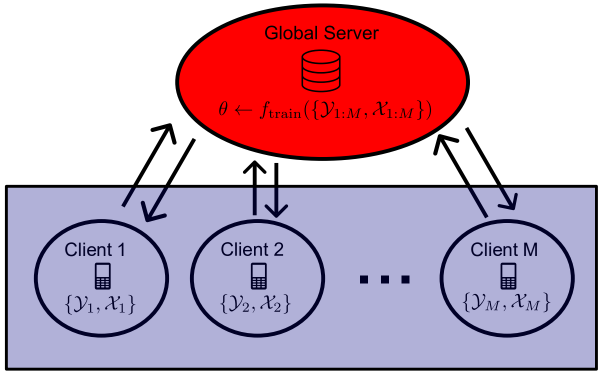


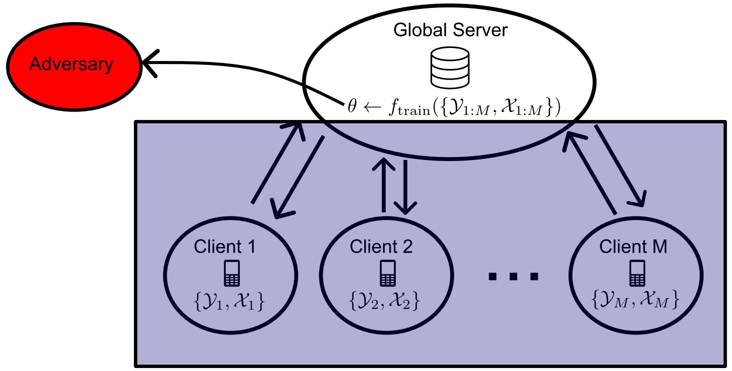
Threat 4.2 – training observations

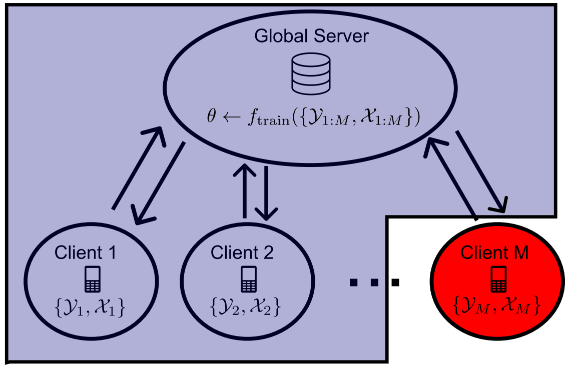



Adversary
Trained model
Global Server
$\theta \leftarrow f_{\text{train}}(\{\mathcal{Y}_{1:M}, \mathcal{X}_{1:M}\})$
Client 1
$\{\mathcal{Y}_1, \mathcal{X}_1\}$
Client 2
$\{\mathcal{Y}_2, \mathcal{X}_2\}$
Client M
$\{\mathcal{Y}_M, \mathcal{X}_M\}$


Adversary
Global Server
$\theta \leftarrow f_{\text{train}}(\{\mathcal{Y}_{1:M}, \mathcal{X}_{1:M}\})$
Client 1
$\{\mathcal{Y}_1, \mathcal{X}_1\}$
Client 2
$\{\mathcal{Y}_2, \mathcal{X}_2\}$
Client M
$\{\mathcal{Y}_M, \mathcal{X}_M\}$


Global Server
$\theta \leftarrow f_{\text{train}}(\{\mathcal{Y}_{1:M}, \mathcal{X}_{1:M}\})$
Client 1
$\{\mathcal{Y}_1, \mathcal{X}_1\}$
Client 2
$\{\mathcal{Y}_2, \mathcal{X}_2\}$
Client M
$\{\mathcal{Y}_M, \mathcal{X}_M\}$


Adversary
Global Server
$\theta \leftarrow f_{\text{train}}(\{\mathcal{Y}_{1:M}, \mathcal{X}_{1:M}\})$
Client 1
$\{\mathcal{Y}_1, \mathcal{X}_1\}$
Client 2
$\{\mathcal{Y}_2, \mathcal{X}_2\}$
Client M
$\{\mathcal{Y}_M, \mathcal{X}_M\}$


Global Server
$\theta \leftarrow f_{\text{train}}(\{\mathcal{Y}_{1:M}, \mathcal{X}_{1:M}\})$
Client 1
$\{\mathcal{Y}_1, \mathcal{X}_1\}$
Client 2
$\{\mathcal{Y}_2, \mathcal{X}_2\}$
Client M
$\{\mathcal{Y}_M, \mathcal{X}_M\}$

Ideas from cryptography

- Related secure computation problems have been studied since the 80s
- We could adapt the earliest attempt to our case with

$$f_{\text{train}} = \max(\theta^{(k-1)}, x)$$

- One client with q-bit feature x
- Asymmetric cipher on n-bit integers:
  - server and client can encrypt with $E(\cdot)$
  - but only the server can decrypt with $D(\cdot)$
- Need to send $n + (2^q + 1)\frac{n}{2} + 1$ bits, three rounds of communication

A. C. Yao, "Protocols for secure computations," *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, Chicago, IL, USA, 1982, pp. 160-164.

Homomorphic encryption

- How far can we get without letting the server decrypt?
- Rough sketch of protocol:
  - Clients encrypt features and send to server
  - Server runs training algorithm on ciphertext
  - Server sends model to clients
  - Clients decrypt model and return it to server
- This would guarantee security against:
  - T1 – eavesdropper
  - T3 – curious server
- But we need the ciphertext equivalent of plaintext operations…

Towards homomorphic encryption: ElGamal

- ElGamal is based on a cyclic group G of order q with generator g
- i.e. elements of G are $1, g, g^2, \dots, g^{q-1}$
- Public key: $(G, q, g, h = g^k)$ Private key: k
- Encryption function: draw a random r from $\{0\colon q-1\}$ and do
$$x \to (g^r, x \cdot h^r)$$
- Decryption function: $g^{r^k} = h^r$ so $g^{r^{q-k}} = h^{-r}$. Hence do
$$(x \cdot h^r) \cdot (g^r)^{q-k} = x \cdot (h^r \cdot h^{-r}) = x$$
- As long as G satisfies certain properties (Decisional Diffie Hellman assumption), it will be hard to get any information on x from the public key and ciphertext.

Towards homomorphic encryption: ElGamal

- ElGamal is based on a cyclic group G of order q with generator g
- i.e. elements of G are $1, g, g^2, \ldots, g^{q-1}$
- Public key: $(G, q, g, h = g^k)$ Private key: k
- Encryption function: draw a random r from $\{0\colon q-1\}$ and do

$$x \rightarrow (g^r, x \cdot h^r)$$

$$E(x_1) \cdot E(x_2) = (g^{r_1}, x_1 \cdot h^{r_1}) \cdot (g^{r_2}, x_2 \cdot h^{r_2}) = (g^{r_1+r_2}, (x_1 \cdot x_2) \cdot h^{r_1+r_2}) = E(x_1 \cdot x_2)$$

- We can do additions or multiplications without decrypting, but not both (“partially homomorphic”)
  - (And we need the same secrete key across clients)

Fully homomorphic encryption

- FHE exists with some limitations on accuracy
  - Need polynomial approximations to e.g. activation functions
- But it slows down computations substantially
  - Two days for binary classification by logistic regression (3 vs 8 MNIST)

Table 2 Running 10-fold cross-validation on compressed MNIST dataset with 1500 samples and 196 features

Training method	# iterations	Avg. training time	Avg. AUC	Avg. AUC (unencrypted)
GD + σ_3	10	48.76 h	0.974	0.977

Chen, H., Gilad-Bachrach, R., Han, K. *et al.* Logistic regression over encrypted data from fully homomorphic encryption. *BMC Med Genomics* **11,** 81 (2018)

Federated learning

"**Federated learning** is a machine learning setting where multiple entities (clients) collaborate in *solving a machine learning problem*, under the *coordination of a central server* or service provider. Each client's *raw data is stored locally* and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective."

Kairouz, P., McMahan, H. B., *et al.* Advances and Open Problems in Federated Learning (2018)

Core Challenges

- Expensive communication
- Statistical heterogeneity (non-IID splits)
- Systems heterogeneity (clients dropping out)
- Privacy concerns

Li, T., Sahu, A. K., Talwalkar, A., Smith, V., Federated Learning: Challenges, Methods and Future Directions (2019)

Objective

$$\min_{w} f(w)$$

Conventional setup:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)\,, \qquad f_i(w) \text{ is e.g. loss on each datapoint}$$

Federated learning:

$$f(w) = \sum_{m=1}^{M} \frac{n_m}{n} F_m(w)\,, \qquad F_m(w) \coloneqq \frac{1}{n_m}\sum_{i\in P_m} f_i(w)$$

Talk Outline

1. Motivations and background
   - Threat models
   - Homomorphic encryption
   - Definition and core challenges
2. **SGD-inspired approaches**
   - Vanilla SGD
   - Federated Averaging
3. Bayesian federated learning
   - Partitioned Variational Inference
4. Improving security and privacy
   - Secure Multi-Party Computation
   - Differential Privacy

Vanilla SGD

Core Challenges

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

At Global Server, iteration i:

Send $w^{(i)}$ to a client m

Receive Δw_m from client

$w^{(i+1)} \leftarrow w^{(i)} + \Delta w_m$

At client m:

Receive $w^{(i)}$

Return $\Delta w_m = -\eta \widehat{\nabla} \ell_m\left(w^{(i)}\right)$

- Not parallelised: slow
- Communication-inefficient

Parallelised SGD

Core Challenges
1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

At Global Server, iteration i:

Choose random subset of clients C

Send $w^{(i)}$ to each client $\in C$

Receive Δw_m from each client

$w^{(i+1)} \leftarrow w^{(i)} + \sum_{m \in \mathrm{C}} \Delta w_m$

At client m:

Receive $w^{(i)}$

Return $\Delta w_m = -\eta \widehat{\nabla} \ell_m \left(w^{(i)} \right)$

- Parallelised
- Communication-inefficient

Federated Averaging

Core Challenges

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

At Global Server, iteration i:

Choose random subset of clients C

Send $w^{(i)}$ to each client $\in C$

Update w_m from each client $m \in C$

$w^{(i+1)} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} w_m$

At client m:

Receive $\mathrm{w} \leftarrow w^{(i)}$

Over E epochs, split into minibatches:

$w \leftarrow w - \eta \widehat{\nabla} \ell_m(w)$

Return w

- Parallelised
- Communication-efficient

McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *AISTATS* 2017

Federated Averaging

Core Challenges

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

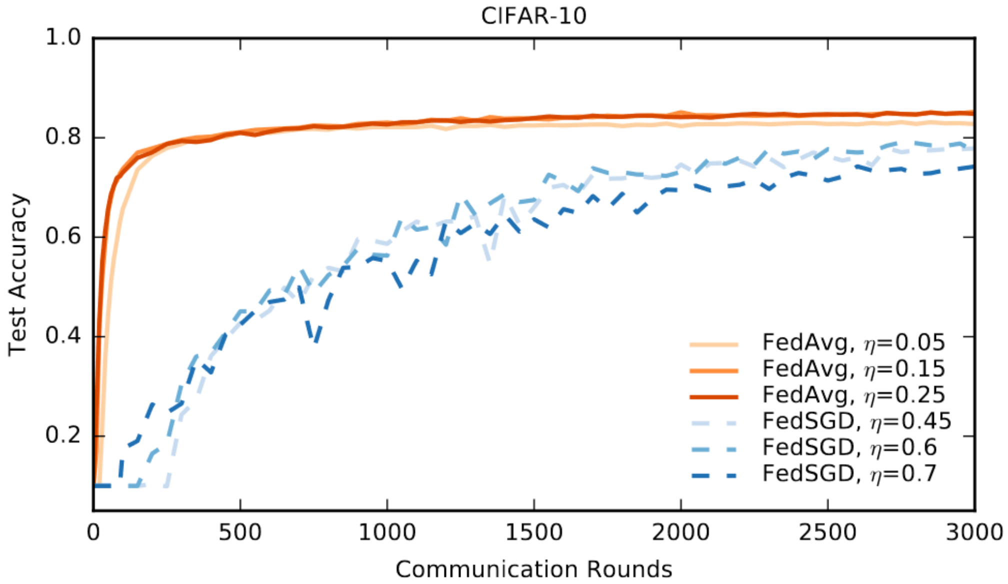


Figure 4: Test accuracy versus communication for the CIFAR10 experiments. FedSGD uses a learning-rate decay of 0.9934 per round; FedAvg uses $B = 50$, learning-rate decay of 0.99 per round, and $E = 5$.

Table 3: Number of rounds and speedup relative to baseline SGD to reach a target test-set accuracy on CIFAR10. SGD used a minibatch size of 100. FedSGD and FedAvg used $C = 0.1$, with FedAvg using $E = 5$ and $B = 50$.

Acc.	80%	82%	85%
SGD	18000 (—)	31000 (—)	99000 (—)
FedSGD	3750 (4.8×)	6600 (4.7×)	N/A (—)
FedAvg	280 (64.3×)	630 (49.2×)	2000 (49.5×)

McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *AISTATS* 2017

Federated Averaging

Core Challenges
1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

At Global Server, iteration i:

Choose random subset of clients C

Send $w^{(i)}$ to each client $\in C$

Update w_m from each client $m \in C$

$w^{(i+1)} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} w_m$

At client m:

Receive w $\leftarrow w^{(i)}$

Over E epochs, split into minibatches:

$$w \leftarrow w - \eta \widehat{\nabla} \ell_m(w)$$

Return w

- Hyperparameter tuning required
- No convergence guarantees
  - Can diverge (non-IID)!
- Compression of messages possible (Konečný et al., 2017)
- Deployed at scale! (Bonawitz et al., 2019)

McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *AISTATS* 2017
Konečný et al., “Federated Learning: Strategies for Improving Communication Efficiency”, 2017
Bonawitz et al., “Towards Federated Learning at Scale: System Design”, *SysML* 2019

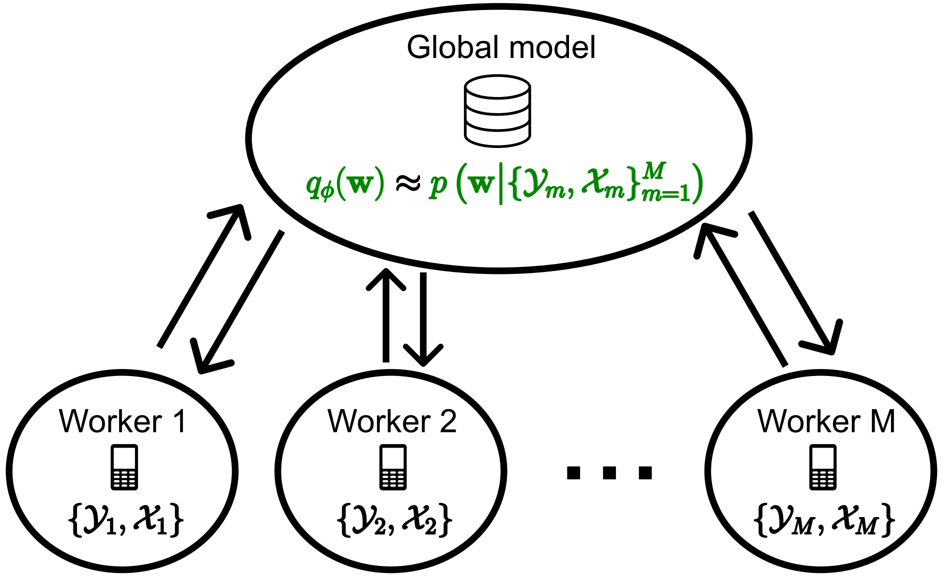
Talk Outline

1. Motivations and background
    - Threat models
    - Homomorphic encryption
    - Definition and core challenges
2. SGD-inspired approaches
    - Vanilla SGD
    - Federated Averaging
3. **Bayesian federated learning**
    - Partitioned Variational Inference
4. Improving security and privacy
    - Secure Multi-Party Computation
    - Differential Privacy

Bayesian FL

Core Challenges

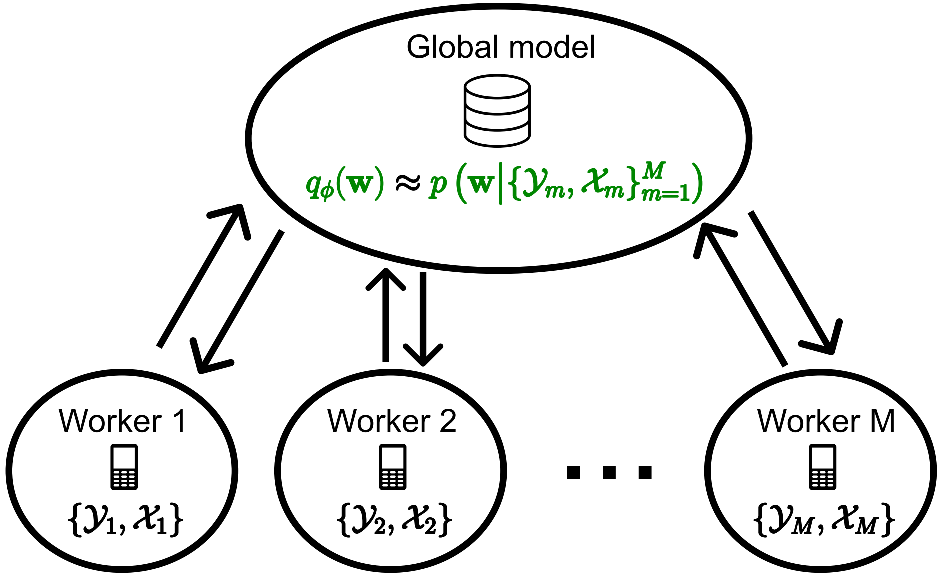
1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns



Bayesian FL

Core Challenges
1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

- SGD ↔ Global VI
- Variational methods
  - Stochastic natural-gradient EP
  - Partitioned VI
  - Store client states locally
- Bayesian Committee Machine
  - Communicate once

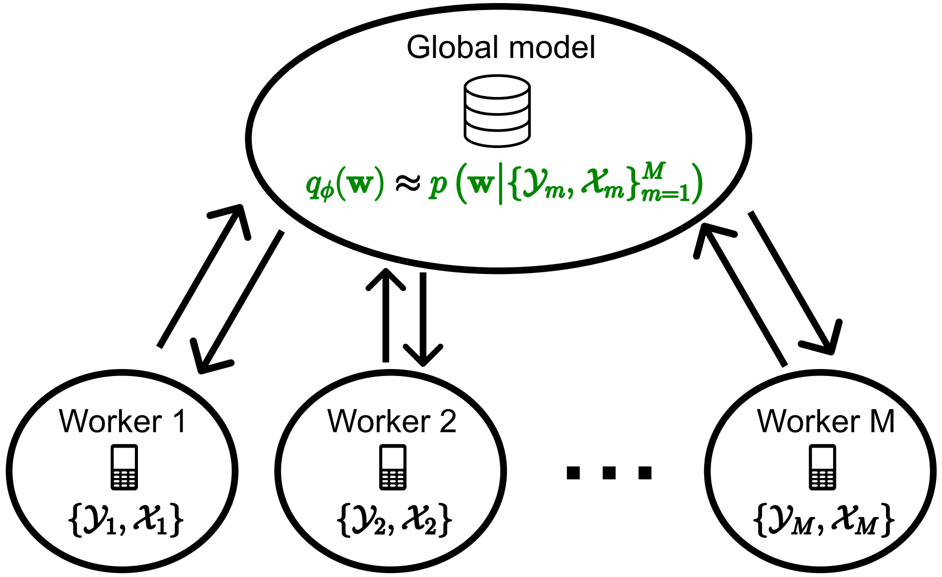


Hasenclever et al., "Distributed Bayesian Learning with Stochastic Natural Gradient Expectation Propagation and the Posterior Server," *JMLR* 2017
Bui et al., "Partitioned Variational Inference: A unified framework encompassing federated and continual learning," 2018
Tresp, "A Bayesian committee machine," *Neural computation* 2000

Partitioned VI

Core Challenges

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns



$$q_\phi(\mathbf{w}) \approx p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$$

$$\frac{p_0(\mathbf{w}) \prod_m t_m(\mathbf{w})}{Z} \approx \frac{p_0(\mathbf{w}) \prod_m p(\mathcal{Y}_m|\mathbf{w}, \mathcal{X}_m)}{p(\mathcal{Y}|\mathcal{X})}$$

Bui et al., "Partitioned Variational Inference: A unified framework encompassing federated and continual learning," 2018

Partitioned VI

Core Challenges

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

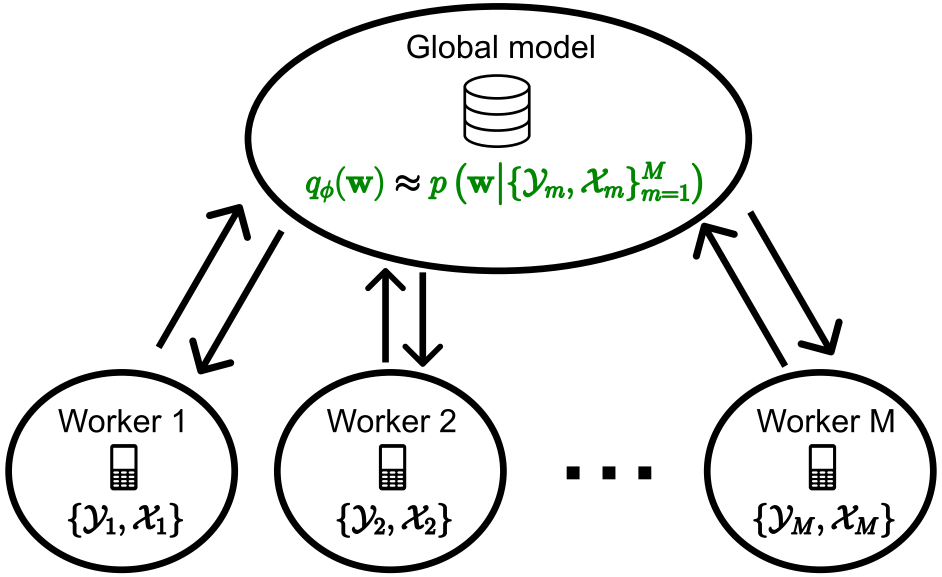
At Global server, iteration i:

Choose random subset of clients C

Send $q_\phi^{(i)}(w)$ to each client $\in$ C

Receive $\Delta t_m(w)$ from each client

$q_\phi^{(i+1)}(w) \leftarrow q_\phi^{(i)}(w) \prod_{m \in C} \Delta t_m(w)$



At client m:

$$\min_{\phi^*} \mathcal{KL}\left(q_{\phi^*}^{(\text{new})}(\mathbf{w}) \middle\| \frac{q_\phi(\mathbf{w}) p(\mathcal{Y}_m|\mathbf{w}, \mathcal{X}_m)}{t_m^{(\text{old})}(\mathbf{w})}\right)$$

compare with:

$$\min_{\phi} \mathcal{KL}\left(q_\phi(\mathbf{w}) || p(\mathbf{w}|\mathcal{Y}, \mathcal{X})\right)$$

$$q_\phi(\mathbf{w}) \approx p(\mathbf{w}|\mathcal{Y}, \mathcal{X})$$

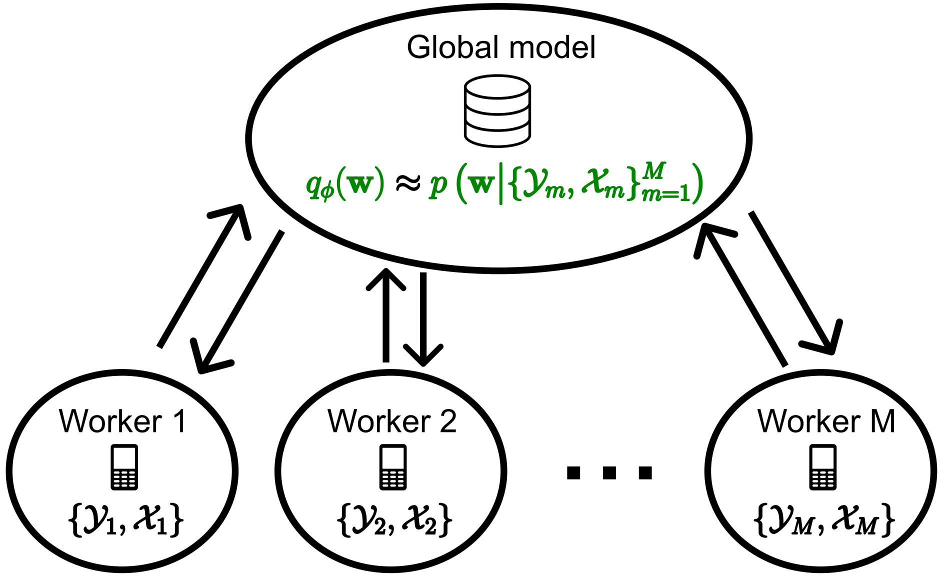
$$\frac{p_0(\mathbf{w}) \prod_m t_m(\mathbf{w})}{Z} \approx \frac{p_0(\mathbf{w}) \prod_m p(\mathcal{Y}_m|\mathbf{w}, \mathcal{X}_m)}{p(\mathcal{Y}|\mathcal{X})}$$

Bui et al., “Partitioned Variational Inference: A unified framework encompassing federated and continual learning,” 2018

Partitioned VI

Core Challenges

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns



At Global server, iteration i:

Choose random subset of clients C

Send $q_\phi^{(i)}(w)$ to each client $\in$ C

Receive $\Delta t_m(w)$ from each client

$q_\phi^{(i+1)}(w) \leftarrow q_\phi^{(i)}(w) \prod_{m \in C} \Delta t_m(w)$

At client m:

$$\min_{\phi^*} \mathcal{KL}\left(q_{\phi^*}^{(\text{new})}(\mathbf{w}) \middle\| \frac{q_\phi(\mathbf{w}) p(\mathcal{Y}_m | \mathbf{w}, \mathcal{X}_m)}{t_m^{(\text{old})}(\mathbf{w})} \right)$$

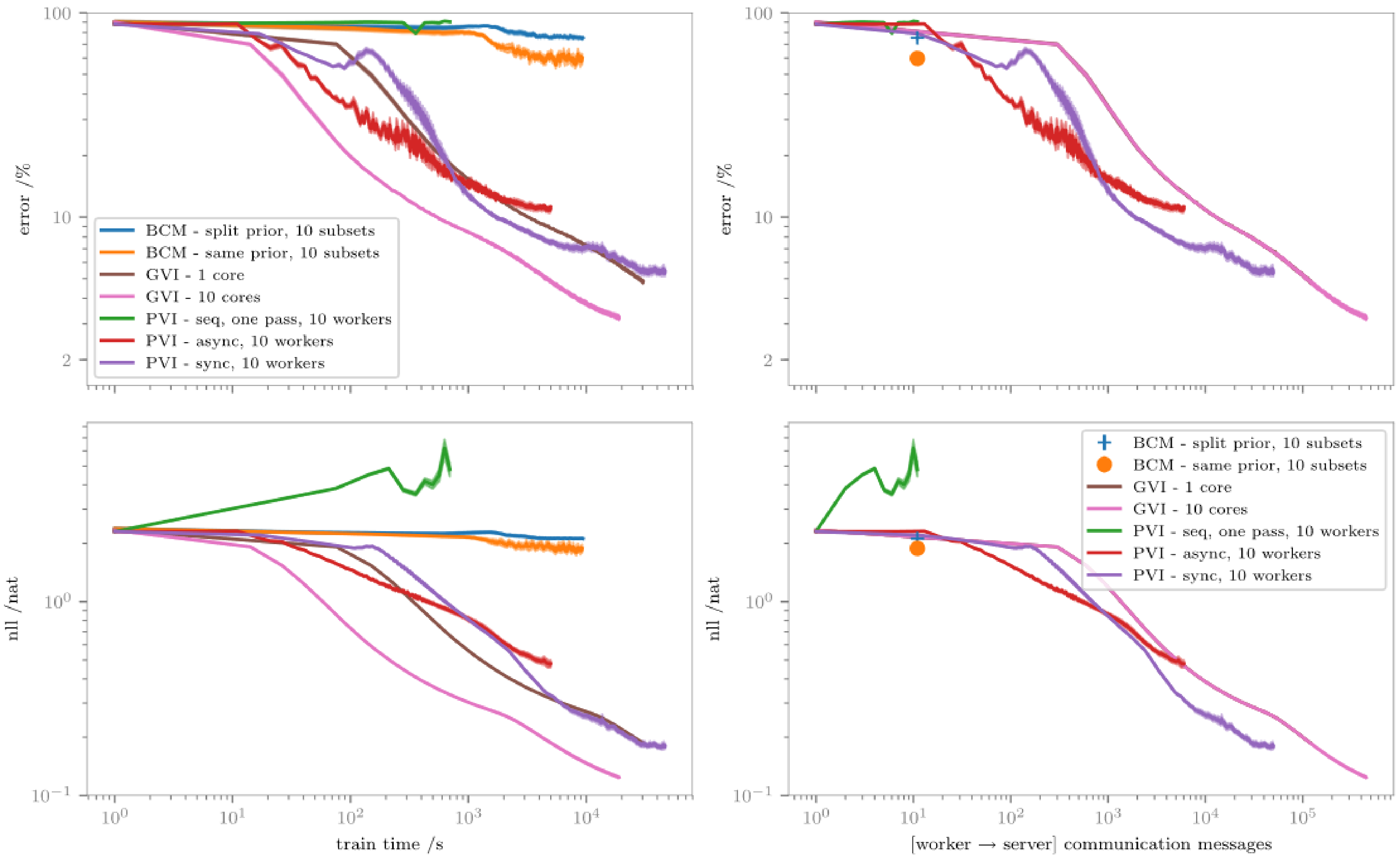
Return $\Delta t_m(w) = \dfrac{q_{\phi^*}^{\text{new}}(w)}{q_\phi^{\text{old}}(w)}$

$$q_\phi(\mathbf{w}) \approx p(\mathbf{w} | \mathcal{Y}, \mathcal{X})$$

$$\frac{p_0(\mathbf{w}) \prod_m t_m(\mathbf{w})}{Z} \approx \frac{p_0(\mathbf{w}) \prod_m p(\mathcal{Y}_m | \mathbf{w}, \mathcal{X}_m)}{p(\mathcal{Y} | \mathcal{X})}$$

Bui et al., "Partitioned Variational Inference: A unified framework encompassing federated and continual learning," 2018

Partitioned VI



(a) Error and NLL vs train time

(b) Error and NLL vs communication cost

Figure 5: Performance on the test set in the federated MNIST experiment with a non-iid distribution of training points across ten workers, i.e. each worker has access to digits of only one class.

Bui et al., "Partitioned Variational Inference: A unified framework encompassing federated and continual learning," 2018

Talk Outline

1. Motivations and background
   - Threat models
   - Homomorphic encryption
   - Definition and core challenges
2. SGD-inspired approaches
   - Vanilla SGD
   - Federated Averaging
3. Bayesian federated learning
   - Partitioned Variational Inference
4. **Improving security and privacy**
   - Secure Multi-Party Computation
   - Differential Privacy

Secure Multi-Party Computation

- “Parties jointly compute a function over inputs while keeping those inputs secure”
  - Homomorphic Encryption
  - Secure Aggregation

- Adds communication rounds & computational cost

Secure Multi-Party Computation

- “Parties jointly compute a function over inputs while keeping those inputs secure”
  - Homomorphic Encryption
  - Secure Aggregation
- Adds communication rounds & computational cost

<u>Secure Aggregation</u>

- Combines cryptographic techniques
  - Secret sharing, key agreement, authenticated encryption, signature scheme, public key infrastructure, …
- Protects against honest-but-curious server, adversarial server
- (Up to) 4 rounds of communication
- Cubic computational cost for server, quadratic for clients

Bonawitz et al., "Practical Secure Aggregation for Privacy-Preserving Machine Learning," 2017

Differential Privacy

Definition 2. *A randomized function $\mathcal{K}$ gives* ϵ-differential privacy *if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S] \quad (+\boldsymbol{\delta}) \tag{1}$$

- "Learn as much as possible from a group while learning as little as possible about any individual in it"

Dwork, "Differential Privacy," 2016

Differential Privacy

Definition 2. *A randomized function $\mathcal{K}$ gives ϵ-*differential privacy *if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S] \quad (+\boldsymbol{\delta}) \tag{1}$$

- "Learn as much as possible from a group while learning as little as possible about any individual in it"
- Achieved by adding (Gaussian) noise
- Global vs Local vs Hybrid
- Combining with Secure MPC

Dwork, "Differential Privacy," 2016

Meta-learning and Federated learning

- Key assumption so far:
  learning a *single global model*

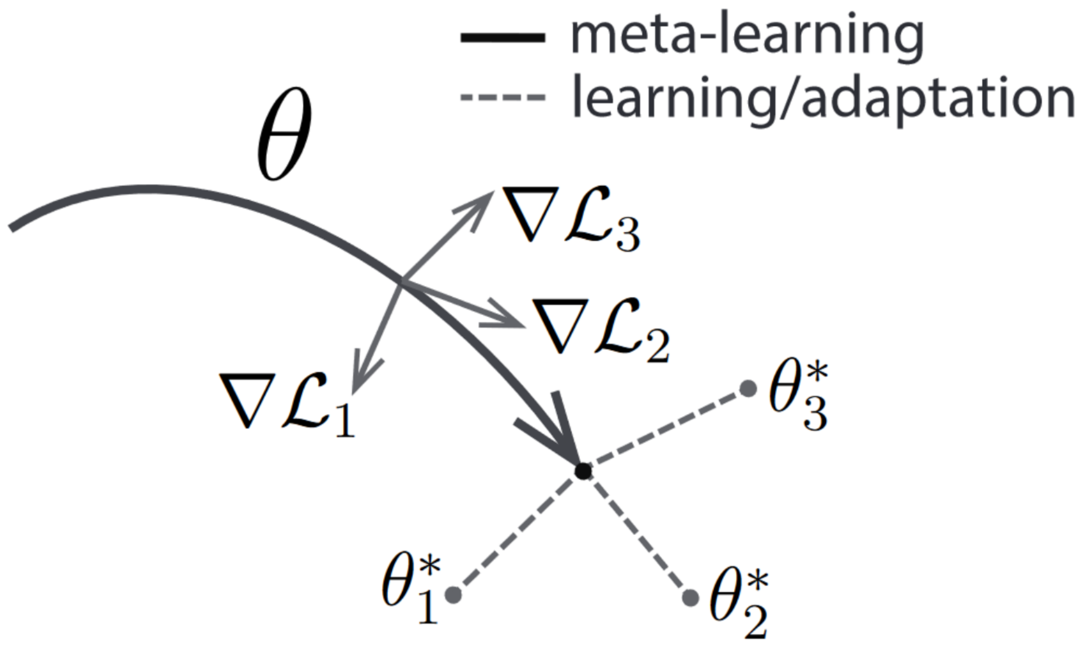
- What if *personalised local models* are better?

Meta-learning and Federated learning

- Key assumption so far:
  learning a *single global model*

- What if *personalised local models* are better?

- Locally fine-tune: cf MAML



Finn et al., "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," *ICML* 2017

Future work

<u>Core Challenges</u>

1. Expensive communication
2. Statistical heterogeneity (non-IID splits)
3. Systems heterogeneity (clients dropping out)
4. Privacy concerns

- Designing algorithms that tackle **all** core challenges
- Communication-accuracy Pareto frontier
- Differential Privacy for FL
- Modelling systems heterogeneity
- Beyond supervised learning
- Benchmarks

Talk Outline

1. Motivations and background
   - Threat models
   - Homomorphic encryption
   - Definition and core challenges
2. SGD-inspired approaches
   - Vanilla SGD
   - Federated Averaging
3. Bayesian federated learning
   - Partitioned Variational Inference
4. Improving security and privacy
   - Secure Multi-Party Computation
   - Differential Privacy