Federated Learning via Posterior Averaging: A New Perspective and Practical Algorithms

ICLR 2021

5/3/2021

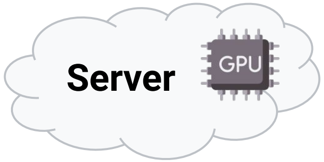
Maruan Al-Shedivat,[1] Jenny Gillenwater,[2] Eric Xing,[1] Afshin Rostamizadeh[2]

[1] Carnegie Mellon University, [2] Google Research

Federated Learning (FL) is usually formulated as a **distributed optimization problem**

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}, \quad f_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})$$

global objective local client objectives



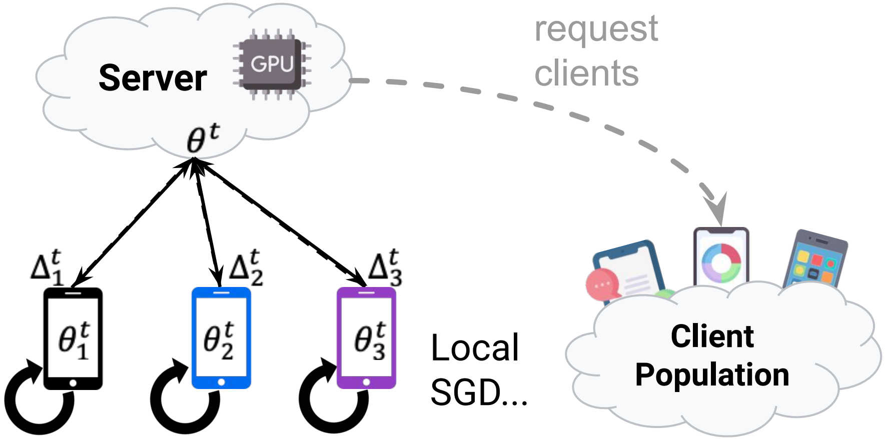
Federated Learning (FL) is usually formulated as a **distributed optimization problem**

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}, \quad f_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})$$

global objective local client objectives

Solve this problem using **FedAvg** (local SGD):

- Optimize the global objective over multiple communication rounds.
- At each round, a subset of clients runs local optimization and communicates with the server.



Federated Learning (FL) is usually formulated as a **distributed optimization problem**

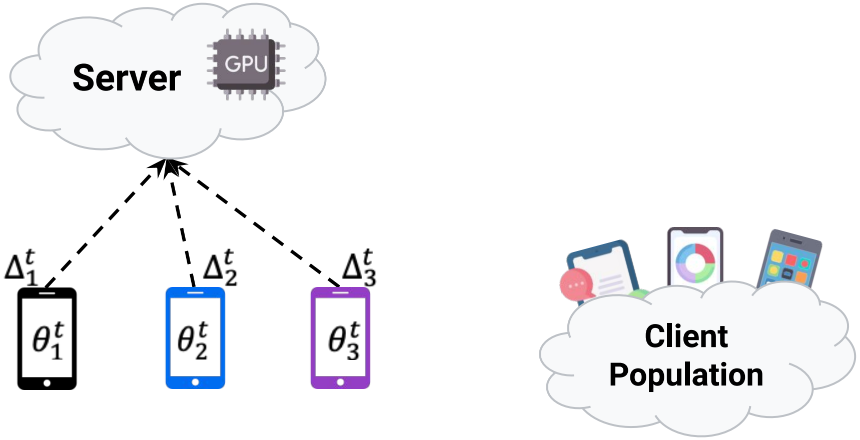
$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}, \quad f_i(\boldsymbol{\theta}) := \frac{1}{n_i}\sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})$$

global objective local client objectives

Solve this problem using **FedAvg** (local SGD):

- Optimize the global objective over multiple communication rounds.
- At each round, a subset of clients runs local optimization and communicates with the server.

Client-server communication is often slow & expensive. How can we speed up training?



Federated Learning (FL) is usually formulated as a **distributed optimization problem**

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}, \quad f_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})$$

global objective

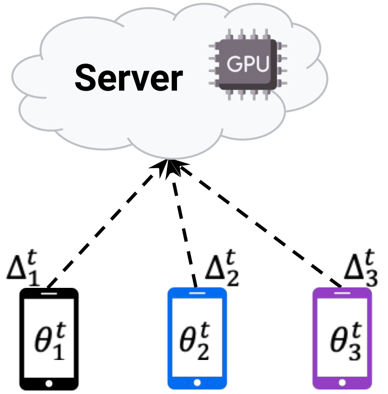
local client objectives

Solve this problem using **FedAvg** (local SGD):

- Optimize the global objective over multiple communication rounds.
- At each round, a subset of clients runs local optimization and communicates with the server.

Client-server communication is often slow & expensive. How can we speed up training?

✔ To speed up (x10-100) we can make clients spend more time at each round on local training (e.g., do more local SGD steps)
⇒ do more local progress, thereby reducing the total number of communication rounds.



Federated Learning (FL) is usually formulated as a **distributed optimization problem**

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}, \qquad f_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})$$

global objective

local client objectives

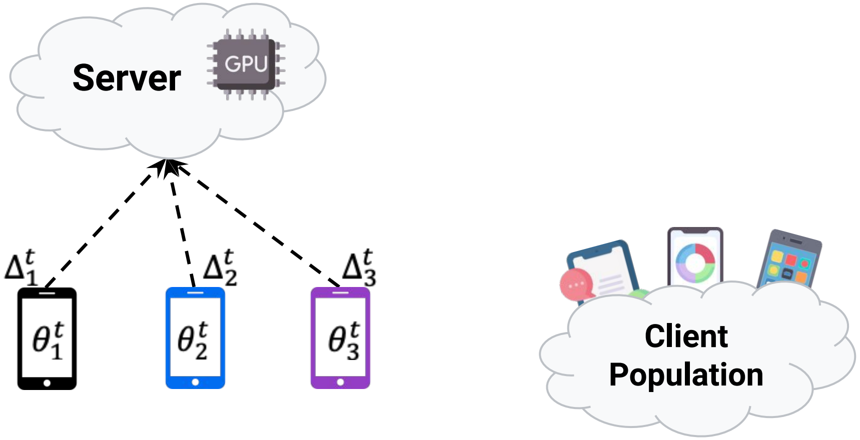
Solve this problem using **FedAvg** (local SGD):

- Optimize the global objective over multiple communication rounds.
- At each round, a subset of clients runs local optimization and communicates with the server.

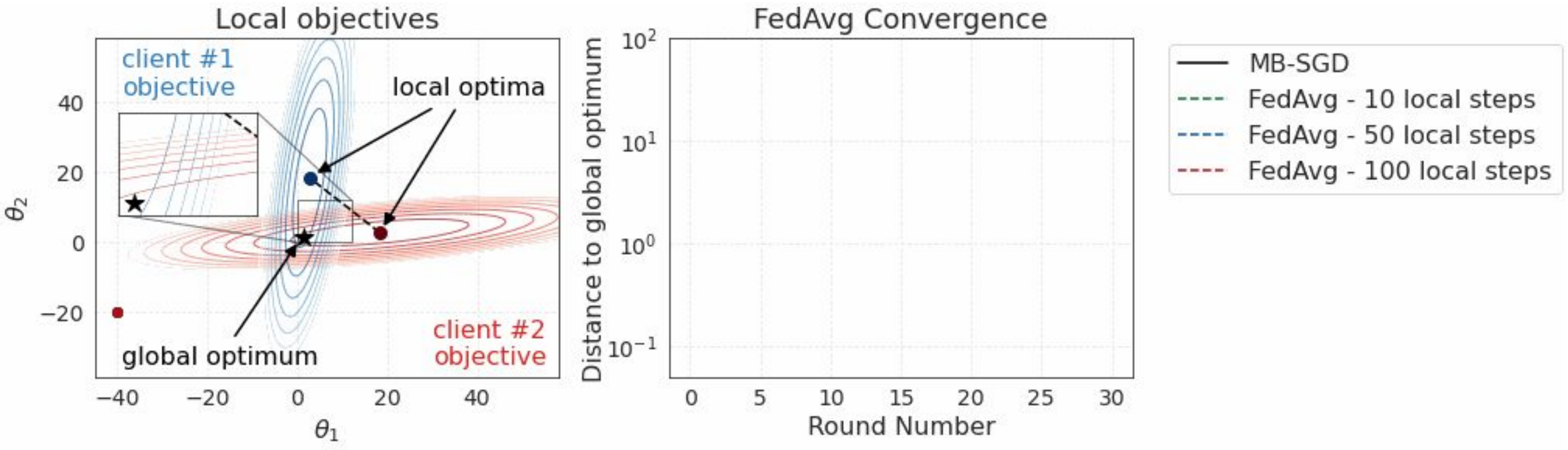
Client-server communication is often slow & expensive. How can we speed up training?

✔ To speed up (x10-100) we can make clients spend more time at each round on local training (e.g., do more local SGD steps) **⇒ do more local progress, thereby reducing the total number of communication rounds.**

✗ Because of client data heterogeneity, it turns out that more local computation per round results in **convergence to inferior models!**



Convergence Issues: Toy Example (Least Squares in 2D)



Least squares: $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}^\star + \varepsilon \Rightarrow \min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 \Rightarrow \min_{\boldsymbol{\theta}} \boldsymbol{\theta}^\top \underbrace{(\mathbf{X}^\top\mathbf{X})}_{:=\mathbf{A}} \boldsymbol{\theta} - \underbrace{(\mathbf{y}^\top\mathbf{X})}_{:=\mathbf{b}} \boldsymbol{\theta}$

Federated Learning (FL) is usually formulated as a **distributed optimization problem**

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}, \quad f_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})$$

global objective   local client objectives

Solve this problem using **FedAvg** (local SGD):

- Optimize the global objective over multiple communication rounds.
- At each round, a subset of clients runs local optimization and communicates with the server.

We propose to approach FL as a distributed **posterior inference problem** (new perspective)

$$\mathbb{P}(\theta \mid D) \propto \prod_{i=1}^{N} \underbrace{\prod_{z \in D_i} \mathbb{P}(z \mid \theta)}_{\text{local likelihood}} \propto \prod_{i=1}^{N} \overbrace{\mathbb{P}(\theta \mid D_i)}^{\text{local posterior}}$$

optima of the FL objective ⇔ modes of the global posterior

Federated Learning (FL) is usually formulated as a **distributed optimization problem**

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \underbrace{\left\{ F(\boldsymbol{\theta}) := \sum_{i=1}^{N} q_i f_i(\boldsymbol{\theta}) \right\}}_{\text{global objective}}, \quad \underbrace{f_i(\boldsymbol{\theta}) := \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}; z_{ij})}_{\text{local client objectives}}$$

Solve this problem using **FedAvg** (local SGD):

- Optimize the global objective over multiple communication rounds.
- At each round, a subset of clients runs local optimization and communicates with the server.

We propose to approach FL as a distributed **posterior inference problem** (new perspective)

$$\mathbb{P}(\theta \mid D) \propto \prod_{i=1}^{N} \underbrace{\prod_{z \in D_i} \mathbb{P}(z \mid \theta)}_{\text{local likelihood}} \propto \prod_{i=1}^{N} \overbrace{\mathbb{P}(\theta \mid D_i)}^{\text{local posterior}}$$

optima of the FL objective ⇔ modes of the global posterior

Key idea:

given that any global posterior decomposes into a product of local posteriors ⇒

run local posterior inference, then multiplicatively aggregate posteriors

Overcoming the Challenges of Posterior Inference

To make posterior inference tractable, we propose:

➔ Use Gaussian approximation

$$\hat{\boldsymbol{\mu}} := \left(\sum_{i=1}^{N} n_i \hat{\boldsymbol{\Sigma}}_i^{-1} \right)^{-1} \left(\sum_{i=1}^{N} n_i \hat{\boldsymbol{\Sigma}}_i^{-1} \hat{\boldsymbol{\mu}}_i \right)$$

Overcoming the Challenges of Posterior Inference

To make posterior inference tractable, we propose:

- ➔ Use Gaussian approximation
- ➔ Use SG-MCMC for local inference

$$\hat{\boldsymbol{\mu}} := \left(\sum_{i=1}^{N} n_i \hat{\boldsymbol{\Sigma}}_i^{-1}\right)^{-1} \left(\sum_{i=1}^{N} n_i \hat{\boldsymbol{\Sigma}}_i^{-1} \hat{\boldsymbol{\mu}}_i\right)$$

local posterior covariances

local posterior means

Overcoming the Challenges of Posterior Inference

To make posterior inference tractable, we propose:

➔ Use Gaussian approximation

$$\hat{\boldsymbol{\mu}} := \left(\sum_{i=1}^{N} n_i \hat{\boldsymbol{\Sigma}}_i^{-1}\right)^{-1} \left(\sum_{i=1}^{N} n_i \hat{\boldsymbol{\Sigma}}_i^{-1} \hat{\boldsymbol{\mu}}_i\right)$$

local posterior covariances

local posterior means

➔ Use SG-MCMC for local inference

➔ Convert matrix inversion into a stochastic optimization problem, which is solved over multiple communication rounds

Federated Posterior Averaging (FedPA): A Practical Algorithm

On the server:

1. Distribute the initial state to clients
2. Collect & average deltas from clients
3. Take a gradient step:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \left[\sum_{i=1}^{N} \frac{n_i}{n} \underbrace{\hat{\boldsymbol{\Sigma}}_i^{-1} (\boldsymbol{\theta}_t - \hat{\boldsymbol{\mu}}_i)}_{:= \boldsymbol{\Delta}_i} \right]$$

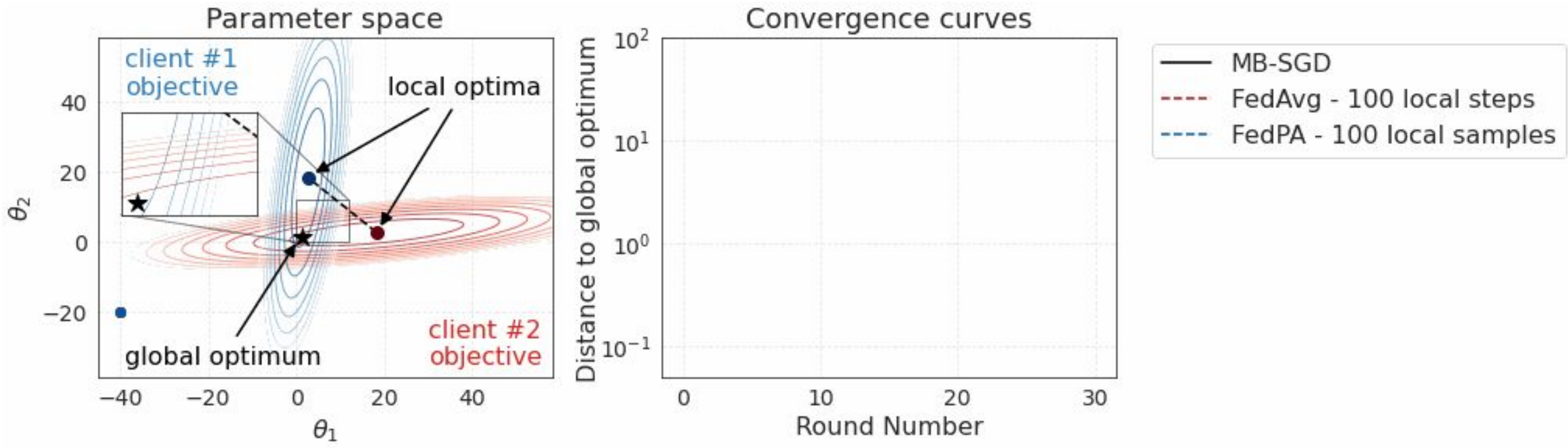
Identical to (generalized) FedAvg!
[Reddi*, Charles*, et al., ICLR 2021]

On the clients:

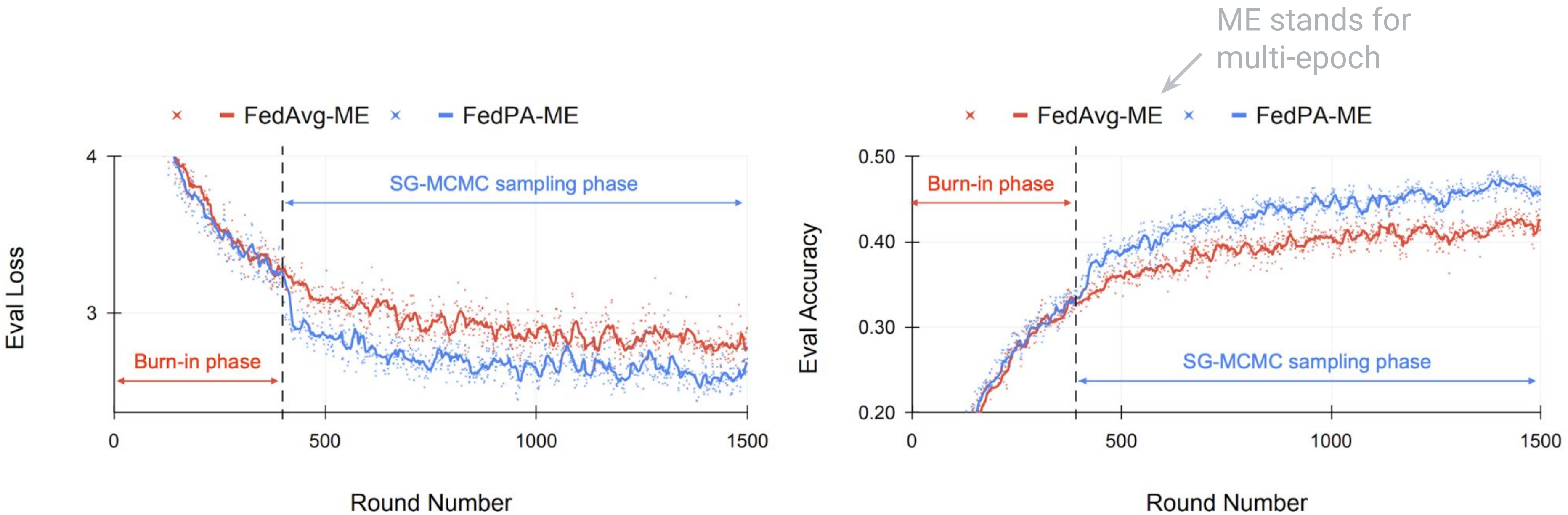
1. Run SGD-based MCMC
2. As new samples arrive, keep computing deltas $\hat{\boldsymbol{\Sigma}}_i^{-1} (\boldsymbol{\theta}_t - \hat{\boldsymbol{\mu}}_i)$
3. Send the final deltas to the server

Similar to FedAvg, we run SGD on the clients, but we compute deltas differently

Does Posterior Inference Work?



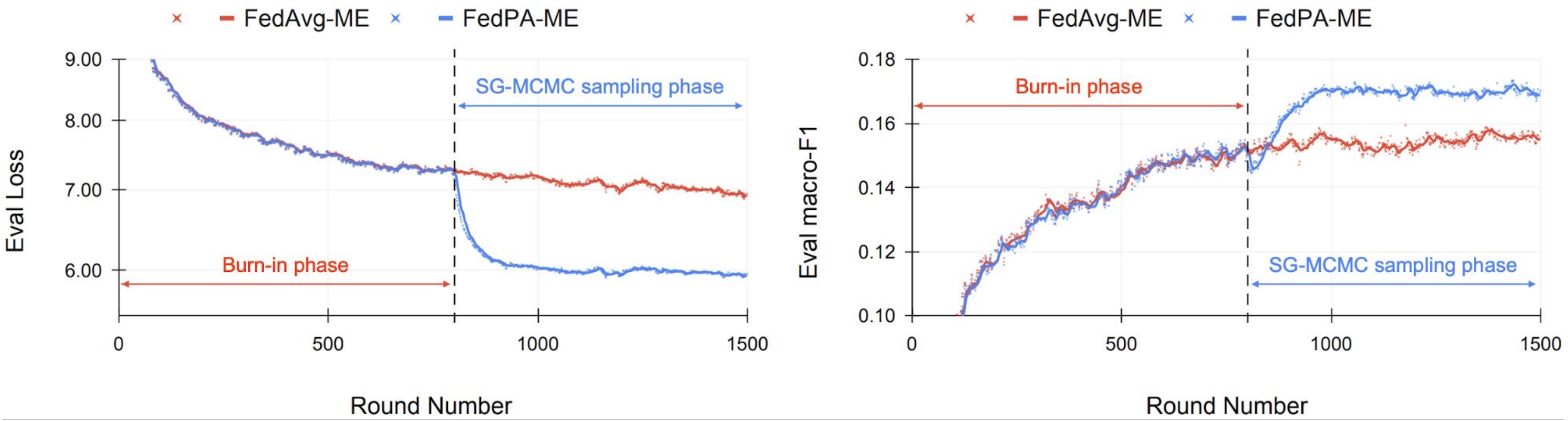
Works Well on Real Benchmarks: Federated CIFAR100



- ➔ **Task:** 100 class image classification, 500 clients (model: ResNet-18).
- ➔ We “burn-in” FedPA by running it in the FedAvg regime for 400 rounds.
- ➔ Starting round 400, we switch to FedPA computation of client deltas.

Works Well on Real Benchmarks: Federated StackOverflow LR

ME stands for multi-epoch

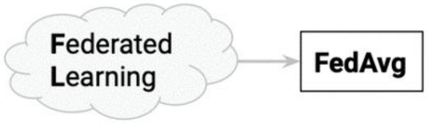


- ➔ **Task:** 500 class multi-label classification, bag of words features, 300K+ clients.
- ➔ We “burn-in” FedPA by running it in the FedAvg regime for 800 rounds.
- ➔ Starting round 800, we switch to FedPA computation of client deltas.

Concluding Thoughts

- ➔ Federated learning can be approached as a probabilistic inference problem, which allows us to design new efficient FL algorithms + re-interpret well-known FedAvg
- ➔ Bayesian ML/DL is typically used for quantification of predictive uncertainty. Turns out, it is also quite useful in distributed, communication-limited settings.

The classical view of FL:



Learn More About Federated Posterior Averaging

Poster: #27

Paper: https://arxiv.org/abs/2010.05273

Code: https://github.com/alshedivat/fedpa

60-minute talk: https://bit.ly/3w2PUTp

Blog post: https://bit.ly/3d83Jaj

Thank you!

Questions?

Email: alshedivat@cs.cmu.edu
Twitter: @alshedivat