Momentum-Driven Adaptivity: Towards Tuning-Free Asynchronous Federated Learning

Wenjing Yan
The Chinese University of Hong Kong

(Joint work with Xiangyu Zhong, Xiaolu Wang, Yingjun Angela Zhang)

ICML 2025

Outline

1 Background and Motivation

2. Algorithm Design

3. Simulation Results

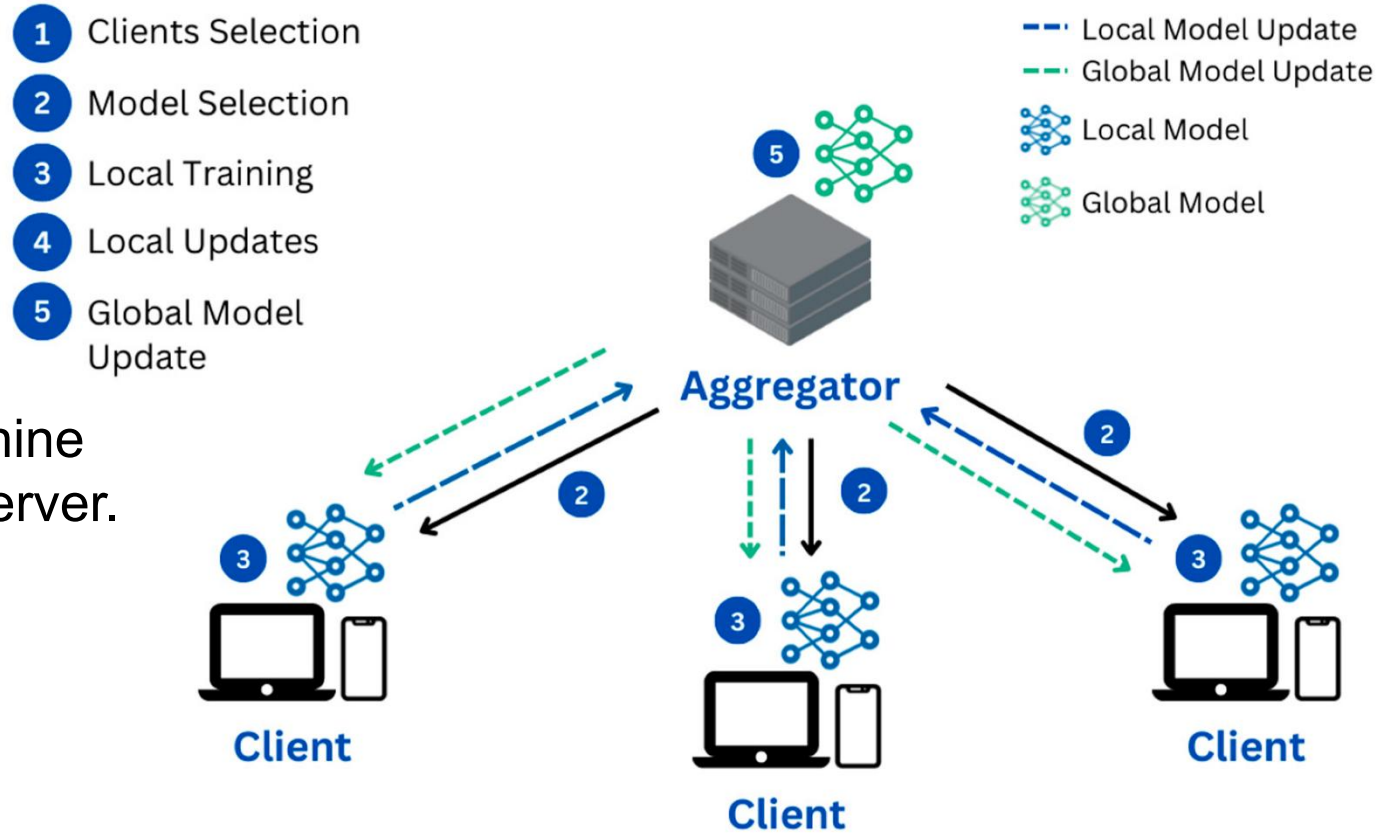
1 Background and Motivation

Problem Formulation of Federated Learning (FL)

- Federated Learning:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta})$$

$$\text{where} \quad f_i(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\xi}_i\sim\mathcal{D}_i}\left[F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right)\right]$$



- Figure from https://encyclopedia.pub/entry/48625

- Multiple clients collaboratively train a machine learning model with the help of a central server.
- Each client performs multiple local update based on its local private data
- Server aggregates the global model

Advantages:

- **Ensures privacy by avoiding raw data sharing**
- **Offers scalability and communication efficiency**

Problem-Parameter-Free Federated Learning

Challenge 1: Data Heterogeneity

- Data Heterogeneity in FL:

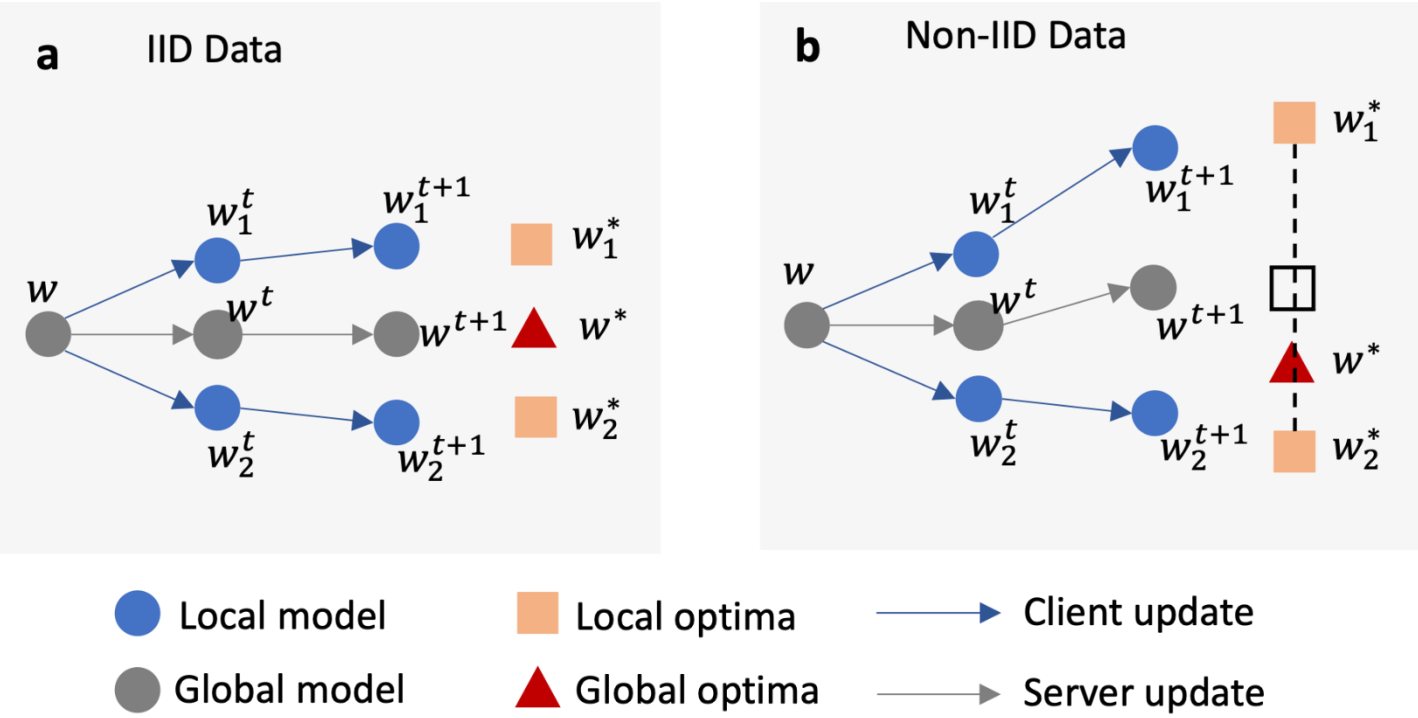
$$f_i(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\xi}_i \sim \mathcal{D}_i} \left[F\left(\boldsymbol{\theta}; \boldsymbol{\xi}_i\right) \right]$$

$$\mathcal{D}_i \neq \mathcal{D}_j \text{ for any } i \neq j$$

Bounded Gradient Dissimilarity

$$\frac{1}{N} \sum_{i=1}^{N} \left\| \nabla f_i(\boldsymbol{\theta}) \right\|^2 \leq B \left\| \nabla f(\boldsymbol{\theta}) \right\|^2 + \sigma_h^2$$

"Client drift": Local updates from individual clients diverge significantly from one another and from the global objective



- **Quantifying this bound is difficult in FL due to privacy and data constraints.**
- **Limit the applicability of FL dynamic environments with varying data distributions.**

- **Figure from https://ar5iv.labs.arxiv.org/html/2103.00710**

Challenge 2: Problem-Specific Hyperparameter Tuning

- Federated Learning:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta})$$

$$\text{where} \quad f_i(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\xi}_i\sim\mathcal{D}_i}\left[F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right)\right]$$

FedAvg Key Steps:

- K steps local update at client i:

$$\boldsymbol{g}_i^{t,k} = \nabla F\left(\boldsymbol{\theta}_i^{t,k};\boldsymbol{\xi}_i^{t,k}\right)$$

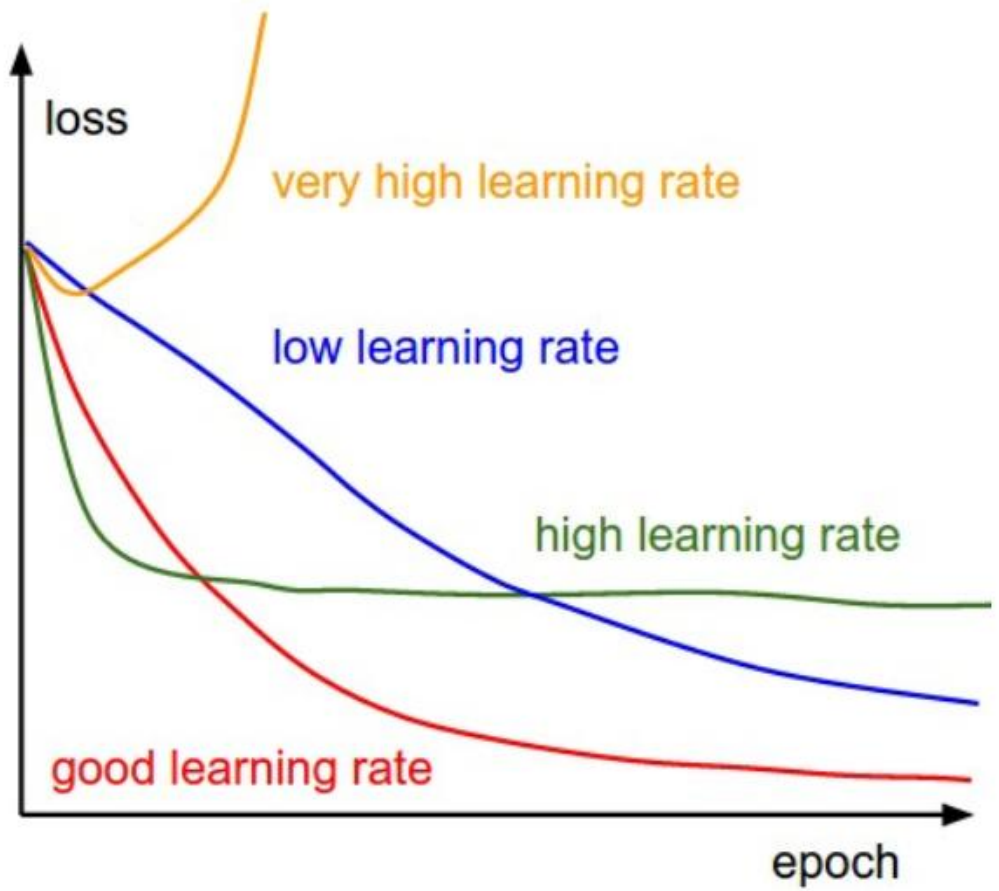
$$\boldsymbol{\theta}_i^{t,k+1} = \boldsymbol{\theta}_i^{t,k} - \eta_t \boldsymbol{g}_i^{t,k}$$

- Global model aggregation at server:

$$\boldsymbol{g}^t = \frac{1}{NK}\sum_{i=1}^{N}(\boldsymbol{\theta}^t - \boldsymbol{\theta}_i^{t,K})$$

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \gamma \boldsymbol{g}^t$$

Stepsize setting is crucial



Challenge 2: Problem-Specific Hyperparameter Tuning

- Federated Learning:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta})$$

$$\text{where} \quad f_i(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\xi}_i\sim\mathcal{D}_i}\left[F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right)\right]$$

- Problem-Specific Constants:
  - L: Smoothness constant
  - σ^2: Stochastic gradient variance
  - B, σ_h^2: coefficients on gradient dissimilarity bound
  - $\Delta := f(\boldsymbol{\theta}_0) - f^*$: Initial suboptimality gap

Assumptions

L-Smoothness

$$\left\|\nabla F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right) - \nabla F\left(\boldsymbol{\delta};\boldsymbol{\xi}_i\right)\right\| \le L\|\boldsymbol{\theta}-\boldsymbol{\delta}\|$$

Stochastic Gradient Variance

$$\mathbb{E}_{\boldsymbol{\xi}_i}\left\|\nabla F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right) - \nabla f_i(\boldsymbol{\theta})\right\|^2 \le \sigma^2$$

Bounded Gradient Dissimilarity

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla f_i(\boldsymbol{\theta})\right\|^2 \le B\left\|\nabla f(\boldsymbol{\theta})\right\|^2 + \sigma_h^2$$

Background and Motivation

Challenge 1: Problem-Specific Hyperparameter Tuning

- Federated Learning:

$$\min_{\theta \in \mathbb{R}^d} \quad f(\theta) := \frac{1}{N} \sum_{i=1}^{N} f_i(\theta)$$

where $f_i(\theta)$:

- Problem-Specific Constants:
  - L: Smoothness constant
  - σ^2: Stochastic gradient variance
  - L, σ_h^2: coefficients on gradient dissimilarity bound
  - $\Delta := f(\theta_0) - f^*$: Initial suboptimality gap

- **Estimating those constants in FL is difficult due to data privacy restrictions and computational complexity.**
- **Problem-specific tuning limits the applicability of those FL approaches in dynamic environments (e.g., IoT, edge devices).**

Assumptions

L-Smoothness

$$\|\nabla F(\theta; \xi_i) - \nabla F(\delta; \xi_i)\| \leq L\|\theta - \delta\|$$

Gradient Variance

$$\mathbb{E}_{\xi_i \sim \mathcal{D}_i} \|\nabla F(\theta; \xi_i) - \nabla f_i(\theta)\|^2 \leq \sigma^2$$

Bounded Gradient Dissimilarity

$$\frac{1}{N} \sum_{i=1}^{N} \|\nabla f_i(\theta)\|^2 \leq B \|\nabla f(\theta)\|^2 + \sigma_h^2$$

Background and Motivation

Our Method: Tuning-Free Asynchornous Federated Learning (AFL)

- Federated Learning:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad f(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\boldsymbol{\theta})$$

$$\text{where} \quad f_i(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{\xi}_i\sim\mathcal{D}_i}\left[F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right)\right]$$

Eliminating the requirement on data heterogeneity bounds

Independent of all problem-specific parameters, enabling **tuning-free**

Assumptions

***L*-Smoothness**

$$\left\|\nabla F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right)-\nabla F\left(\boldsymbol{\delta};\boldsymbol{\xi}_i\right)\right\| \leq L\|\boldsymbol{\theta}-\boldsymbol{\delta}\|$$

Stochastic Gradient Variance

$$\mathbb{E}_{\boldsymbol{\xi}_i}\left\|\nabla F\left(\boldsymbol{\theta};\boldsymbol{\xi}_i\right)-\nabla f_i(\boldsymbol{\theta})\right\|^2 \leq \sigma^2$$

Bounded Gradient Dissimilarity

$$\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla f_i(\boldsymbol{\theta})\right\|^2 \leq B\left\|\nabla f(\boldsymbol{\theta})\right\|^2+\sigma_h^2$$

Outline

1 Background and Motivation

2. Algorithm Design

3. Simulation Results

Momentum-driven Asynchronous Federated Learning (MasFL)

Handling data heterogeneity: Momentum + Control variates

- Local update at client i:

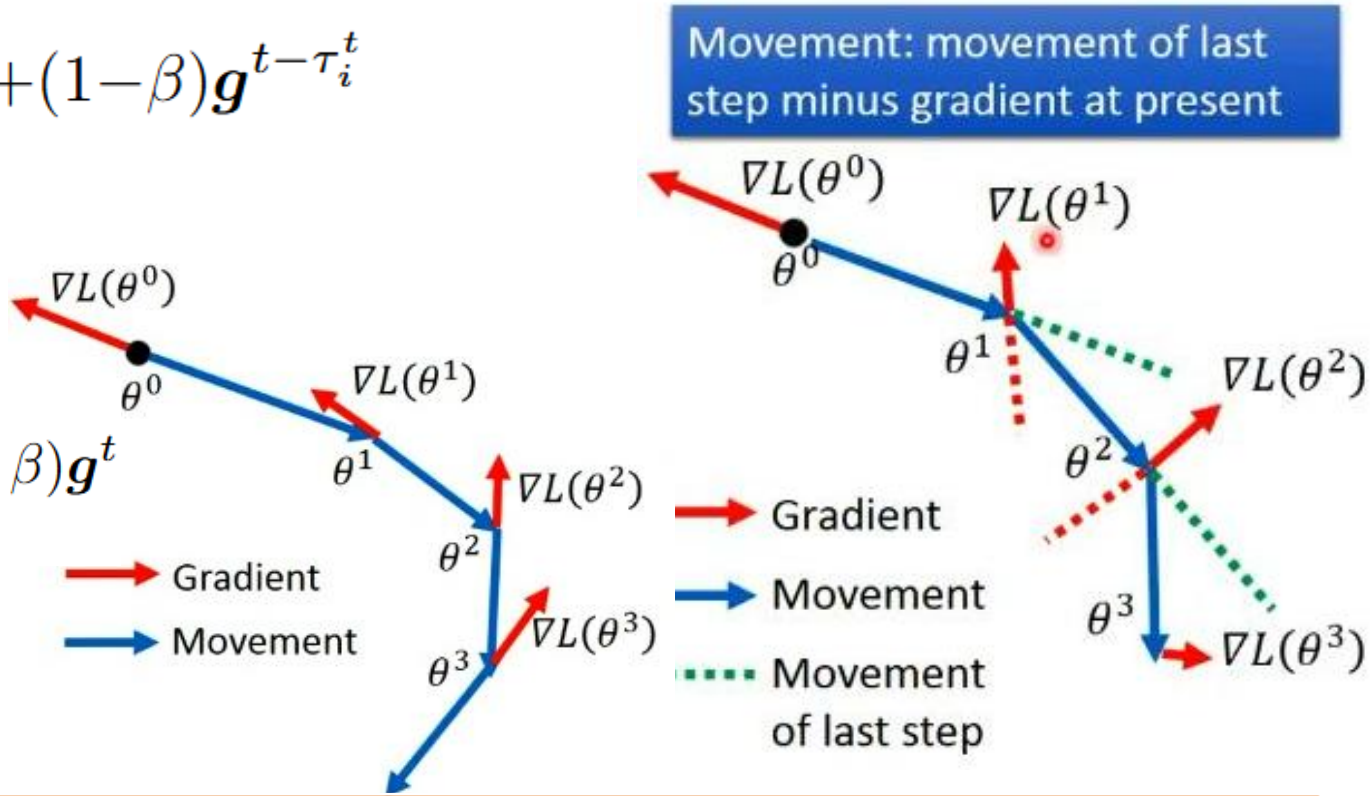
$$\boldsymbol{g}_i^{t,k} = \beta\left(\nabla F\left(\boldsymbol{\theta}_i^{t,k}; \boldsymbol{\xi}_i^{t,k}\right) - \widetilde{\boldsymbol{c}}_i^t + \boldsymbol{c}^{t-\tau_i^t}\right) + (1-\beta)\boldsymbol{g}^{t-\tau_i^t}$$

$$\boldsymbol{\theta}_i^{t,k+1} = \boldsymbol{\theta}_i^{t,k} - \eta \boldsymbol{g}_i^{t,k}$$

- Global aggregation at server:

$$\boldsymbol{g}^{t+1} = \beta\left(\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\boldsymbol{c}_i^{t+1} - \boldsymbol{c}_i^t\right) + \boldsymbol{c}^t\right) + (1-\beta)\boldsymbol{g}^t$$

$$\boldsymbol{c}^{t+1} = \boldsymbol{c}^t + \frac{1}{N}\sum_{i\in\mathcal{S}_t}\left(\boldsymbol{c}_i^{t+1} - \boldsymbol{c}_i^t\right)$$



Momentum: Accumulating past gradients across iterations and clients

Momentum-driven Asynchronous Federated Learning (MasFL)

Algorithm 1 MasFL: Procedures at Central Server

Require: Initial model $\boldsymbol{\theta}^0$, control variates $\boldsymbol{c}_i^0 = \frac{1}{K}\sum_{k=0}^{K-1}\nabla F\left(\boldsymbol{\theta}^0;\boldsymbol{\xi}_i^{-1,k}\right)$ for any i, $\boldsymbol{c}^0 = \frac{1}{N}\sum_i \boldsymbol{c}_i^0$, momentum $\boldsymbol{g}^0 = \boldsymbol{c}^0$, global learning rate γ, local learning rate η, and momentum parameter β

for $t = 0, \cdots, T-1$ **do**

Randomly selected a set of clients $\mathcal{S}_t$

Update

$$\boldsymbol{c}_i^{t+1} = \begin{cases} \widetilde{\boldsymbol{c}}_i^{t+1}, & \text{if } i \in \mathcal{S}_t \\ \boldsymbol{c}_i^t, & \text{otherwise} \end{cases}$$

Aggregate momentum

$$\boldsymbol{g}^{t+1} = \beta\left(\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\boldsymbol{c}_i^{t+1} - \boldsymbol{c}_i^t\right) + \boldsymbol{c}^t\right) + (1-\beta)\boldsymbol{g}^t$$

Update global model $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \gamma\boldsymbol{g}^{t+1}$

Aggregate control variate

$$\boldsymbol{c}^{t+1} = \boldsymbol{c}^t + \frac{1}{N}\sum_{i\in\mathcal{S}_t}\left(\boldsymbol{c}_i^{t+1} - \boldsymbol{c}_i^t\right)$$

Send $\boldsymbol{\theta}^{t+1}$ and $\beta\boldsymbol{c}^{t+1} + (1-\beta)\boldsymbol{g}^{t+1}$ to all clients

end for

Algorithm 2 MasFL: Procedures at Client i

Receive $\boldsymbol{\theta}^{t-\tau_i^t}$ and $\beta\boldsymbol{c}^{t-\tau_i^t} + (1-\beta)\boldsymbol{g}^{t-\tau_i^t}$ from server. Set $\boldsymbol{\theta}_i^{t,0} = \boldsymbol{\theta}^{t-\tau_i^t}$

for $k = 0, \cdots, K-1$ **do**

Compute

$$\boldsymbol{g}_i^{t,k} = \beta\left(\nabla F\left(\boldsymbol{\theta}_i^{t,k};\boldsymbol{\xi}_i^{t,k}\right) - \widetilde{\boldsymbol{c}}_i^t + \boldsymbol{c}^{t-\tau_i^t}\right) + (1-\beta)\boldsymbol{g}^{t-\tau_i^t}$$

Update local model $\boldsymbol{\theta}_i^{t,k+1} = \boldsymbol{\theta}_i^{t,k} - \eta\boldsymbol{g}_i^{t,k}$

end for

Send $\widetilde{\boldsymbol{c}}_i^{t+1} = \frac{1}{K}\sum_{k=0}^{K-1}\nabla F\left(\boldsymbol{\theta}_i^{t,k};\boldsymbol{\xi}_i^{t,k}\right)$ to the server

Adaptive Momentum-driven Asynchronous Federated Learning (MasFL)

Key Techniques: Normalized Gradient Descent

- Traditional gradient descent:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}_t)$$

- The stepsize constraint is:

$$e.g.\ \eta \leq 1/L$$

The step size η must be small enough to account for the gradient's magnitude, which is governed by L. Large gradients can lead to overshooting and instability.

- Normalized gradient descent:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \frac{\nabla f(\boldsymbol{x}_t)}{\|\nabla f(\boldsymbol{x}_t)\|}$$

The gradient is normalized, so the step size η no longer depends on how large the gradient is (irrespective of L). Consistent step sizes allows NGD to navigate flat regions, steep regions, and saddle points more effectively.

Steep region → NGD scales down the update step → prevents overshooting

Flat region → NGD amplifies the update step → avoids stagnation

Algorithm 3 AdaMasFL: Procedures at Central Server

Require: Initial model $\boldsymbol{\theta}^0$, control variates $\boldsymbol{c}_i^0 = \frac{1}{K}\sum_{k=0}^{K-1}\nabla F\left(\boldsymbol{\theta}^0;\boldsymbol{\xi}_i^{-1,k}\right)$ for any i, $\boldsymbol{c}^0 = \frac{1}{N}\sum_i \boldsymbol{c}_i^0$, momentum $\boldsymbol{g}^0 = \boldsymbol{c}^0$, global learning rate γ, local learning rate η, and momentum parameter β.
for $t = 0, \cdots, T-1$ **do**
  Randomly selected a set of clients $\mathcal{S}_t$
  Update

$$\boldsymbol{c}_i^{t+1} = \begin{cases} \widetilde{\boldsymbol{c}}_i^{t+1}, & \text{if } i \in \mathcal{S}_t \\ \boldsymbol{c}_i^t, & \text{otherwise} \end{cases}$$

  Aggregate local updates $\overline{\boldsymbol{g}}^t = \frac{1}{S}\sum_{i\in\mathcal{S}_t}\boldsymbol{\Delta}_i^t$
  Update global model $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \gamma\overline{\boldsymbol{g}}^t$
  Aggregate momentum

$$\boldsymbol{g}^{t+1} = \beta\left(\frac{1}{S}\sum_{i\in\mathcal{S}_t}\left(\boldsymbol{c}_i^{t+1} - \boldsymbol{c}_i^t\right) + \boldsymbol{c}^t\right) + (1-\beta)\boldsymbol{g}^t$$

  Aggregate control variate

$$\boldsymbol{c}^{t+1} = \boldsymbol{c}^t + \frac{1}{N}\sum_{i\in\mathcal{S}_t}\left(\boldsymbol{c}_i^{t+1} - \boldsymbol{c}_i^t\right)$$

  Download $\boldsymbol{\theta}^{t+1}$ and $\beta\boldsymbol{c}^{t+1}+(1-\beta)\boldsymbol{g}^{t+1}$ to all clients
end for

Algorithm 4 AdaMasFL: Procedures at Client i

Receive $\boldsymbol{\theta}^{t-\tau_i^t}$ and $\beta\boldsymbol{c}^{t-\tau_i^t} + (1-\beta)\boldsymbol{g}^{t-\tau_i^t}$ from server. Set $\boldsymbol{\theta}_i^{t,0} = \boldsymbol{\theta}^{t-\tau_i^t}$
for $k = 0, \cdots, K-1$ **do**
  Compute

$$\boldsymbol{g}_i^{t,k} = \beta\left(\nabla F\left(\boldsymbol{\theta}_i^{t,k};\boldsymbol{\xi}_i^{t,k}\right) - \widetilde{\boldsymbol{c}}_i^t + \boldsymbol{c}^{t-\tau_i^t}\right) + (1-\beta)\boldsymbol{g}^{t-\tau_i^t}$$

  Update local model

$$\boldsymbol{\theta}_i^{t,k+1} = \boldsymbol{\theta}_i^{t,k} - \eta\frac{\boldsymbol{g}_i^{t,k}}{\|\boldsymbol{g}_i^{t,k}\|}$$

end for
Send $\boldsymbol{\Delta}_i^t = \frac{1}{\eta K}\left(\boldsymbol{\theta}^{t-\tau_i^t} - \boldsymbol{\theta}_i^{t,K}\right)$ and $\widetilde{\boldsymbol{c}}_i^{t+1} = \frac{1}{K}\sum_{k=0}^{K-1}\nabla F\left(\boldsymbol{\theta}_i^{t,k};\boldsymbol{\xi}_i^{t,k}\right)$ to the server

3 Theoretical Results

Convergence of MasFL

Theorem 1. *Suppose that Assumptions 1 and 2 holds. Let* $\{\boldsymbol{\theta}^t\}_{t=1}^T$ *be the global iterates generated by MasFL. Set* $\beta = \sqrt{\frac{SK}{T}}$ *and* $\gamma = \frac{1}{4L}\sqrt{\frac{SK}{T}}$. *Define* $a := \tau_{\max}^2\beta^2 + 20e^2\eta^2K^2L^2$. *If the condition* $1 - 4a - \sqrt{\frac{SK}{T}} \geq 0$ *is satisfied, i.e.,* $\eta \leq \frac{\sqrt{T-\sqrt{SKT}-4SK\tau_{\max}^2}}{4\sqrt{5}eKL\sqrt{T}}$, *then it holds that*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{\theta}^t)\|^2 \leq \mathcal{O}\left(\frac{L\Delta+\sigma^2}{\sqrt{SKT}} + \frac{\sigma^2}{T} + \frac{L^2\sigma^2}{NKT}\right)$$

for sufficiently large T, *where* e *denotes Euler's number and* $\Delta := f(\boldsymbol{\theta}^0) - f^*$ *represents the initial optimality gap with* $f^* = \min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) > -\infty$.

Assumptions

L-Smoothness

$$\|\nabla F(\boldsymbol{\theta};\boldsymbol{\xi}_i) - \nabla F(\boldsymbol{\delta};\boldsymbol{\xi}_i)\| \leq L\|\boldsymbol{\theta}-\boldsymbol{\delta}\|$$

Stochastic Gradient Variance

$$\mathbb{E}_{\boldsymbol{\xi}_i}\|\nabla F(\boldsymbol{\theta};\boldsymbol{\xi}_i) - \nabla f_i(\boldsymbol{\theta})\|^2 \leq \sigma^2$$

3 Theoretical Results

Convergence of AdaMasFL

Theorem 2. *Suppose that Assumptions 1 and 2 holds. Let* $\{\theta^t\}_{t=1}^T$ *be the global iterates generated by AdaMasFL. Set* $\gamma = \frac{(SK)^{1/4}}{T^{3/4}}$, $\eta = \frac{1}{K\sqrt{T}}$, *and* $\beta = \sqrt{\frac{SK}{T}}$, *then it holds that*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\theta^t)\right\| \leq \mathcal{O}\left(\frac{\Delta + L + \sigma + \sqrt{L\sigma}}{(SKT)^{\frac{1}{4}}} + \sigma\sqrt{\frac{SK}{T}} + \frac{\sqrt{L\sigma\tau_{\max}}}{T^{\frac{3}{8}}(SK)^{\frac{1}{8}}} + \bar{\tau}\sigma\sqrt{\frac{S}{T}} + \tau_{\max}L\frac{(SK)^{\frac{1}{4}}}{T^{\frac{3}{4}}} + \bar{\tau}\sqrt{L\sigma\tau_{\max}}\frac{(SK)^{\frac{3}{8}}}{T^{\frac{7}{8}}}\right)$$

for sufficiently large T.

Tuning-Free AFL

All hyperparameters (η: local learning rate,γ: global learning rate, and β: momentum parameter) in Algorithm AdaMasFL are explicated determined by system-predefined constants: S (the number of participation clients), K (local update times), T (iteration times)

3 Comparisons with Prior Work

Table 1: Comparisons of AFL algorithms for handling heterogeneous data.
(Convergence Rate = The convergence rate of different algorithms in terms of $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f(\boldsymbol{\theta}^t)\right\|$; Additional Assumptions = Additional assumptions aside from Assumptions 2.1–2.3; BDH = Bounded data heterogeneity define in (1); BG = Bounded gradient that $\|\nabla f_i(\boldsymbol{\theta})\| \leq G,\ \forall i, \boldsymbol{\theta}$.)

Algorithms	Convergence Rate[1]	Additional Assumptions	Stepsize Restrictions	Stepsize-related Problem-Parameters
FedBuff (Nguyen et al., 2022)	$\mathcal{O}\left(\frac{K^{1/4}\sigma_g}{(ST)^{1/4}}+\sqrt{\frac{\tau_{\max}\bar{\tau}}{T}}\right)$	BDH	$\eta\gamma \leq \frac{1}{4K\tau_{\max}^{3/2}}$	$\tau_{\max}$
CA^{2}FL (Wang et al., 2023)	$\mathcal{O}\left(\frac{1}{(SKT)^{1/4}}+\sqrt{\frac{\tau_{\max}}{T}}\right)$	BDH	$\eta\gamma \leq \frac{S}{36K\tau_{\max}^2 L^2}, \eta \leq \frac{1}{36K\sqrt{\tau_{\max}}L}$	$\tau_{\max}, L$
DeFedAvg-nIID (Wang et al., 2024a)	$\mathcal{O}\left(\frac{1}{(SKT)^{1/4}}+\frac{1}{\sqrt{KT}}\right)$	BDH, BG	$\eta\gamma \leq \frac{1}{4LK\tau_{\max}}, \eta \leq \frac{1}{4\sqrt{3}LK}$	$\tau_{\max}, L$
FADAS (Wang et al., 2024c)	$\mathcal{O}\left(\frac{1}{(ST)^{1/4}}+\sqrt{\frac{\tau_{\max}\bar{\tau}}{T}}\right)$	BDH, BG	$\eta\gamma \leq \min\left\{\frac{\epsilon^2 S(N-1)}{180C_G^2 N(N-S)\tau_{\max}^2 KL}, \frac{\sqrt{\epsilon^3 S(N-1)}}{12\sqrt{C_G N(N-S)\tau_{\max}^2 KL}}\right\}, \eta \leq \frac{\sqrt{\epsilon}}{\sqrt{360C_G}\tau_{\max}^2 KL}^2$	$\tau_{\max}, L, G$
MasFL	$\mathcal{O}\left(\frac{\sqrt{\kappa}}{(SKT)^{1/4}}+\sqrt{\frac{\kappa}{T}}\right)^3$	–	$\beta=\sqrt{\frac{SK}{T}}, \gamma=\frac{\beta}{4L}, \eta \leq \frac{\sqrt{T-\sqrt{SKT}-4SK\tau_{\max}^2}}{4\sqrt{5}eKL\sqrt{T}}$	$\tau_{\max}, L$
AdaMasFL	$\mathcal{O}\left(\frac{1}{(SKT)^{1/4}}+\frac{\sqrt{\tau_{\max}}}{T^{3/8}}\right)$	–	$\beta=\sqrt{\frac{SK}{T}}, \gamma=\frac{(SK)^{1/4}}{T^{3/4}}, \eta=\frac{1}{K\sqrt{T}}$	–

[1] For the convergence rate defined in terms of $\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}^t)\|^2$, we can readily obtained the corresponding rate with respect to $\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\boldsymbol{\theta}^t)\|$ by taking square root on both sides of the associated bound. This operator is verified by the following fact: $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{\theta}^t)\| = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\sqrt{\|\nabla f(\boldsymbol{\theta}^t)\|^2} \leq \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\mathbb{E}\|\nabla f(\boldsymbol{\theta}^t)\|^2} \leq \sqrt{\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\boldsymbol{\theta}^t)\|^2}$, where the first and second inequalities utilizes Jensen's inequality as the square root function is concave.
[2] $C_G := \eta KG + \epsilon$ and $\epsilon > 0$ is an adaptive optimization parameter.
[3] $\kappa := \frac{3-4a}{1-4a-\sqrt{SK/T}}$ and $a := \tau_{\max}^2\beta^2 + 20e^2\eta^2K^2L^2$.

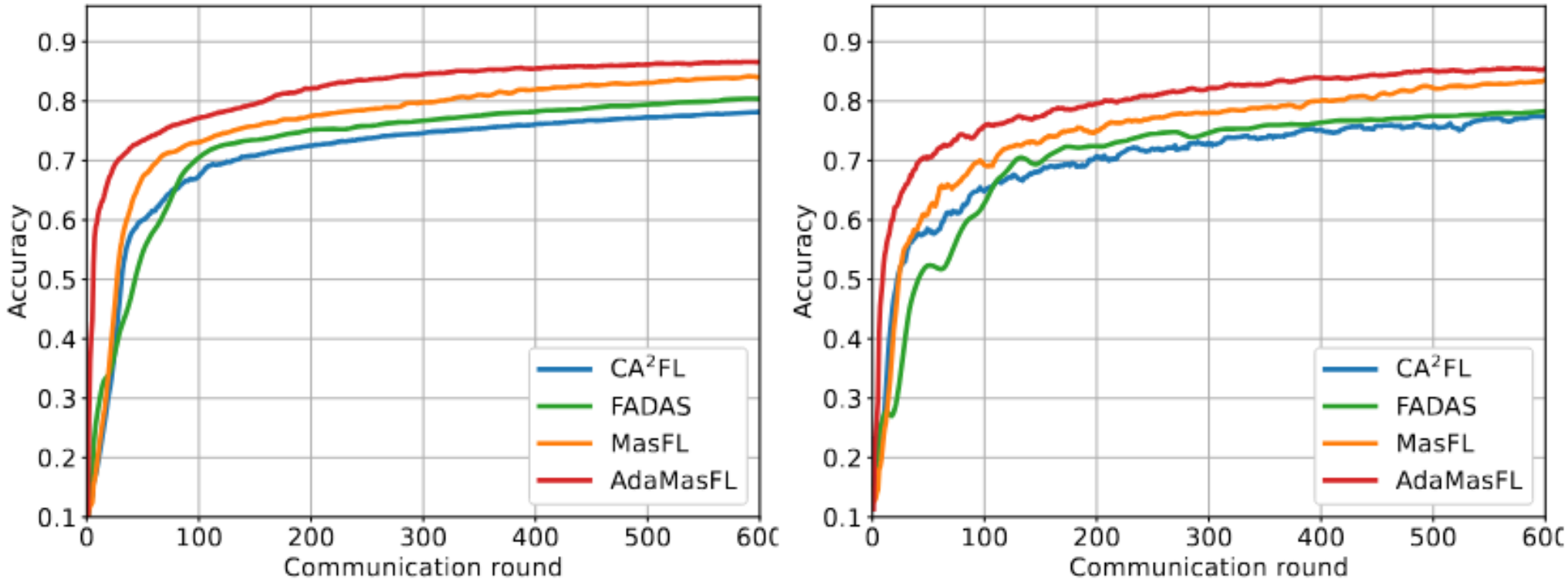
Outline

1 Background and Motivation

2. Algorithm Design

3. Simulation Results

3 Comparisons with Prior Work



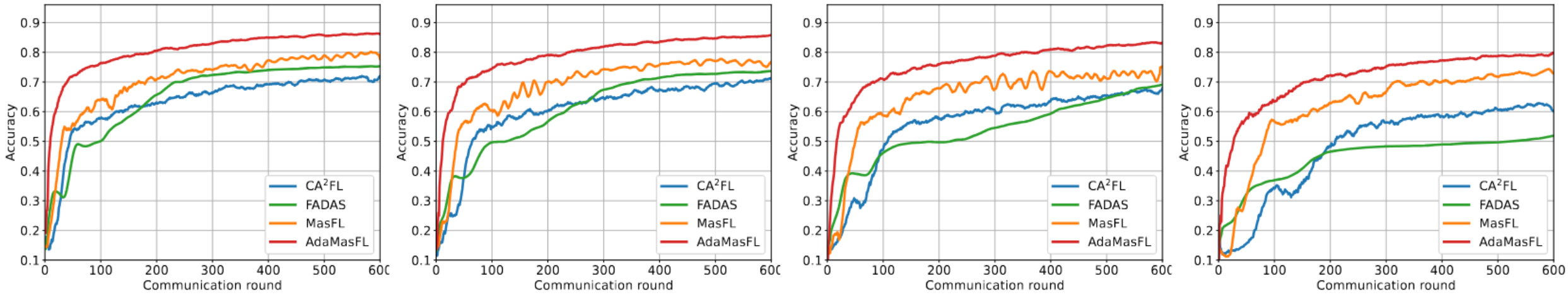
(a) i.i.d. (b) non-i.i.d.

Figure 1. Test accuracy versus the number of communication rounds on the FMNIST dataset (CNN, Dir(0.5)).

Our approach achieves start-of-the-art performance while eliminating the tedious process of stepsize tuning

The stepsizes of AdaMasFL are set directly based on the guidance of Theorem 2. The stepsize of all baselines are perfectly tuned by grid search.

3 Comparisons with Prior Work



(a) $M_c = 30$ (b) $M_c = 40$ (c) $M_c = 60$ (d) $M_c = 100$

Figure 2. Test accuracy on the non-i.i.d. FMNIST dataset under varying levels of asynchrony

- *Mc* denotes the number of clientsperforming local updates concurrently. a larger *Mc* enables more frequent global aggregations, resulting in greater asynchronous delays.
- In Figure 2, we fix the learning rate settings of all algorithmsto those used in Figure 1 and evaluate their performance as Mc increases.

AdaMasFL demonstrates ex-ceptional robustness to varying levels of asynchrony, main-taining nearly consistent performance despite increasing delays

THANKS

Thanks!