Efficient Adaptive Federated Optimization

Su Hyeong Lee [1] Sidharth Sharma [2] Manzil Zaheer [3] Tian Li [1]

[1]University of Chicago [2]Carnegie Mellon University [3]Google DeepMind

Overview

Goal: Adaptive optimization has been shown to accelerate convergence, and be critical to training transformer-based models such as LLMs. However, adaptivity imposes additional constraints on client memory and communication during distributed optimization. Can we develop a strategy to overcome these bottlenecks in federated learning (FL)?

Contributions:

- ✓ Develop a class of efficient jointly adaptive distributed training algorithms (`FedAda`^{2}) to mitigate the restrictions above while retaining full benefits of adaptivity
- ✓ Ensure that `FedAda`^{2}-class algorithms maintains an identical communication complexity as the vanilla FedAvg algorithm
- ✓ Provide robust convergence guarantees for the general, non-convex setting, achieving the same best known convergence rate as prior federated adaptive optimizers

Why is Joint Adaptivity Desirable?

Empirical Perspective: Improves convergence and final accuracies (e.g., Wang et al, 2021; Lee et al, 2024)

Theoretical Perspective: Construction of an artificial problem involving heavy-tailed noise in which adaptivity is paramount

Theorem 1 (Informal)

For μ-strongly convex online global objectives, FedAvg incurs infinite regret in expectation when client stochastic gradient distribution is heavy-tailed (defined as $\mathbb{E}[X^2] = \infty$).

- **Corollary 1:** Introducing client side or joint adaptivity via AdaGrad for the setting in Theorem 1 mitigates suffering infinite regret at every step!
- **Corollary 2:** Even a single client with heavy-tailed gradient noise is able to instantaneously propagate their volatility to the global model, which severely destabilizes distributed learning in expectation.

Moral of story: Advantage of FL is large supply of clients, which enable the trainer to draw from an abundant stream of computational power. Downside is that global model may become strongly impacted by the various gradient distributions induced by local data shards, which must be dealt with carefully (e.g. using jointly adaptive optimizers to mitigate regret).

FedAda^2: Efficient Joint Adaptivity

`FedAda`^{2}: Efficient Adaptive Federated Optimization

```
for $t = 1, \dots, T$ do
    Sample participating clients $\mathcal{S}^t \subset [N]$
    for each client $i \in \mathcal{S}^t$ (in parallel) do
        (Main Idea 1:) Zero Preconditioner Initialization
        for $k = 1, \dots, K$ do
            Draw $g_{i,k}^t \sim \mathcal{D}(x_{i,k-1}^t)$, let $m_k \leftarrow MOM(g_{i,k}^t)$
            (Main Idea 2:) Any Efficient Optimizer
        end for
        $\Delta_i^t = x_{i,K}^t - x_{t-1}$
    end for
    Server Update
end for
```

SM3-ADAGRAD VARIANT:

$$m_k \leftarrow g_{i,k}^t, \quad \mu_k(b) \leftarrow 0 \quad \text{for} \quad \forall b \in \{1, \dots, q\},$$

$$\text{Loop } j : \begin{cases} v_k(j) \leftarrow \min_{b : S_b \ni j} \mu_{k-1}(b) + \left(g_{i,k}^t(j)\right)^2 \\ \mu_k(b) \leftarrow \max\{\mu_k(b), v_k(j)\}, \quad \forall b : S_b \ni j \end{cases}$$

Non-convex Convergence Analysis

Theorem 2

Under some assumptions, FedAda^2 deterministically satisfies

$$\min_{t \in [T]} \|\nabla f(x_{t-1})\|^2 \leq \frac{\Psi_1 + \Psi_2 + \Psi_3 + \Psi_4 + \Psi_5}{\Psi_6}$$

where asymptotically,

$$\psi_1 = \Theta(1), \ \psi_2 = \eta^2 \eta_\ell^2 T, \ \psi_3 = \eta \eta_\ell^2 T, \ \psi_4 = \eta \eta_\ell \log(1 + T\eta_\ell^2)$$

and

$$\psi_5 = \begin{cases} \eta^3 \eta_\ell^3 T \text{ if } \mathcal{O}(\eta_\ell) \leq \mathcal{O}(1) \\ \eta^3 \eta_\ell T \text{ if } \Theta(\eta_\ell) > \Omega(1) \end{cases}, \quad \psi_6 = \begin{cases} \eta \eta_\ell T \text{ if } \mathcal{O}(T\eta_\ell^2) \leq \mathcal{O}(1) \\ \eta \sqrt{T} \text{ if } \Theta(T\eta_\ell^2) > \Omega(1) \end{cases}.$$

Theorem 3 (Generalization)

Given client $i \in [N]$, strategy $l \in [Op]$, global timestep r, and local timestep p, assume optimizer strategies satisfy update rule

$$x_{i,p}^{r,l} = x_{i,p-1}^{r,l} - \eta_\ell \sum_{\ell=1}^{p} \frac{a_{i,\ell}^{r,l} g_{i,\ell}^{r,l}}{\vartheta_{i,\ell}^{r,l}(g_{i,1}^{r,l}, \dots, g_{i,\ell}^{r,l})}$$

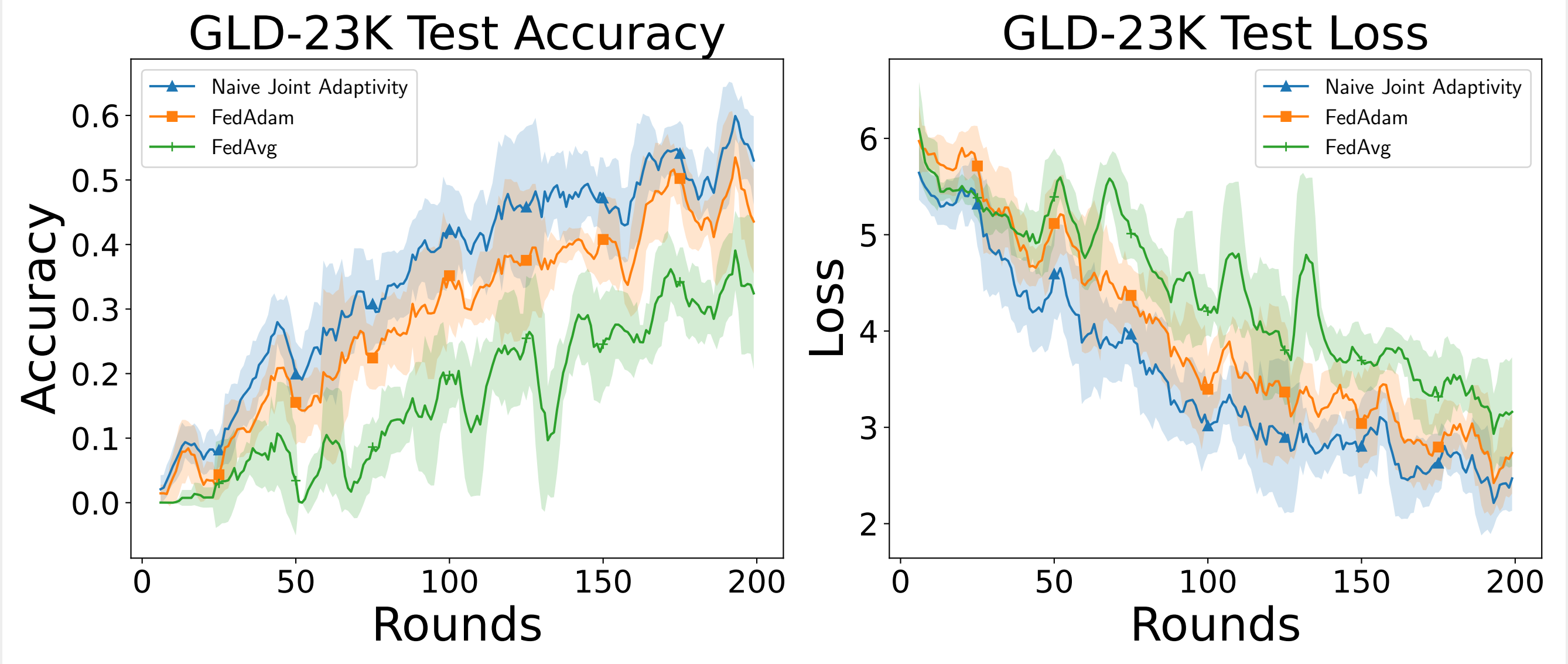
where

$$0 < m_l \leq \vartheta_{i,\ell}^{r,l}(g_{i,1}^{r,l}, \dots, g_{i,\ell}^{r,l}) \leq M_l \quad \text{and} \quad 0 < a_l \leq a_{i,\ell}^{r,l} \leq A_l$$

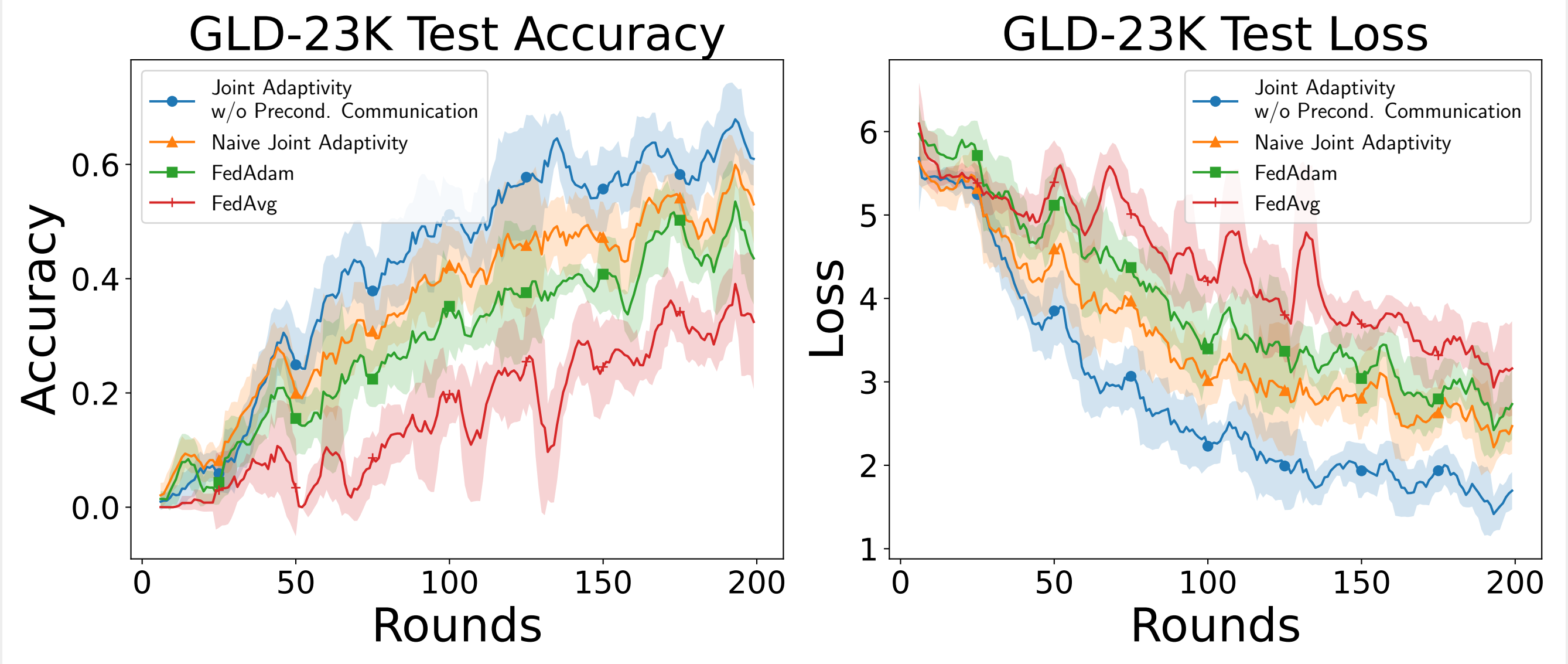
for all possible values of i, ℓ, r, l. If $1 \leq K(O_l^i) \leq K$ and $0 < \Xi^- < w(O_l^i) < \Xi^+$, then the bound in **Theorem 2** holds.

Experiments (e.g. GLD-23K)

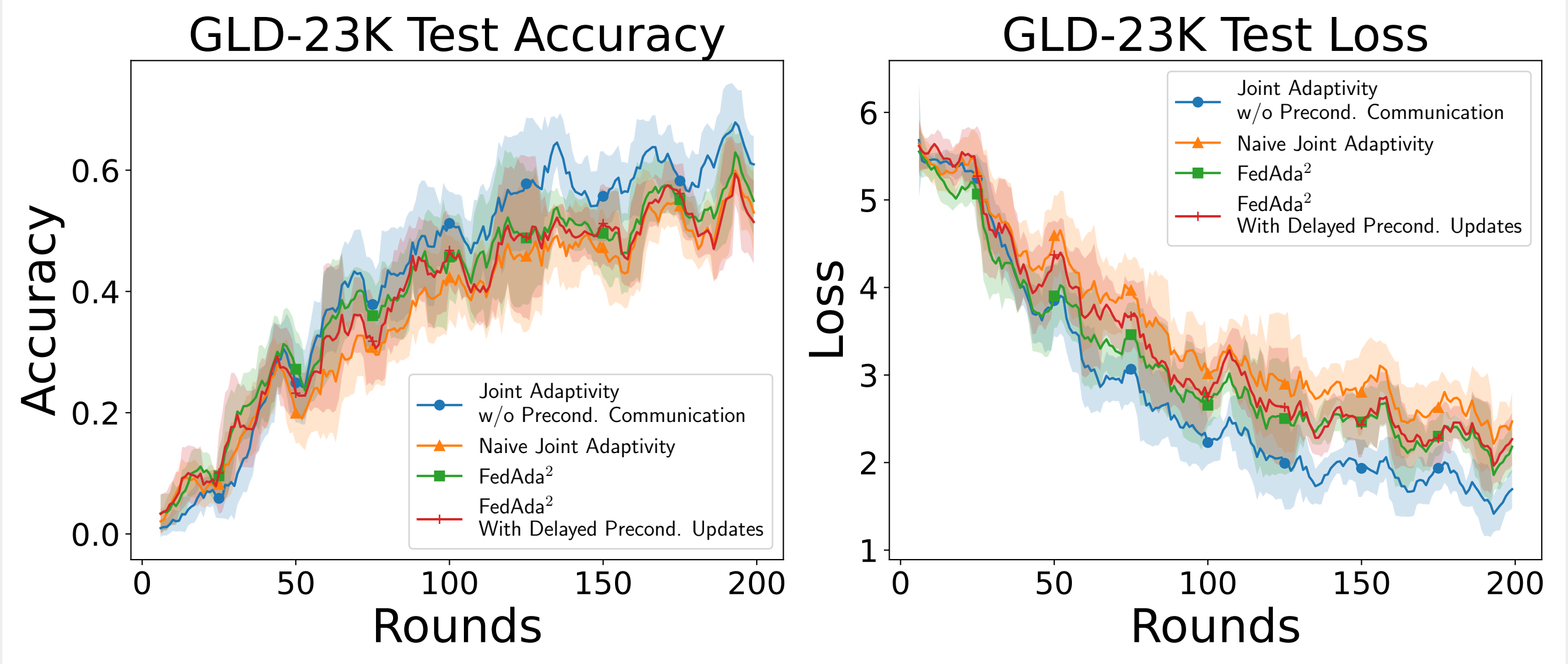
Naive Joint Adaptivity > FedAvg, FedAdam



Naive Joint Adaptivity ≈ Adaptivity w/o Preconditioner Transmission



`FedAda`^{2} (Efficient) ≈ Naive Joint Adaptivity (Costly)



Future Work

1. Generalize full gradient convergence results to stochastic gradients
2. Elucidate the link between attention mechanisms and heavy-tailed gradient noise, and propose additional optimizers
3. Explore empirical performance of blended optimization, identifying settings in which mixing optimizer strategies (e.g. using client side SGD & Adam in the same round) are advantageous for distributed learning