

# Adding vs. Averaging in Distributed Primal-Dual Optimization

Chenxin Ma<sup>\*</sup><sup>a</sup> · Virginia Smith<sup>\*</sup><sup>b</sup> · Martin Jaggi<sup>c</sup> · Michael I. Jordan<sup>b</sup> · Peter Richtárik<sup>d</sup> · Martin Takáč<sup>a</sup>



ETH



## overview

**Goal:** Fixing the **communication bottleneck** for **distributed optimization** in supervised ML

### Contributions

COCOA+ is a *primal-dual* framework for distributed optimization

- ✓ efficient additive aggregation of local updates
- ✓ strong convergence guarantees
- ✓ framework: guarantees for arbitrary local solvers
- ✓ significant practical speedup

## setup

Primal problem formulation

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \ell_i(\mathbf{w}^T \mathbf{x}_i)$$

data partitioned by examples  $A_i := \mathbf{x}_i$

primal-dual correspondence  
 $\mathbf{w} = \frac{1}{\lambda n} A \boldsymbol{\alpha}$

Dual problem

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} -\frac{\lambda}{2} \left\| \frac{A \boldsymbol{\alpha}}{\lambda n} \right\|^2 - \frac{1}{n} \sum_{i=1}^n \ell_i^*(-\alpha_i)$$

Information: local

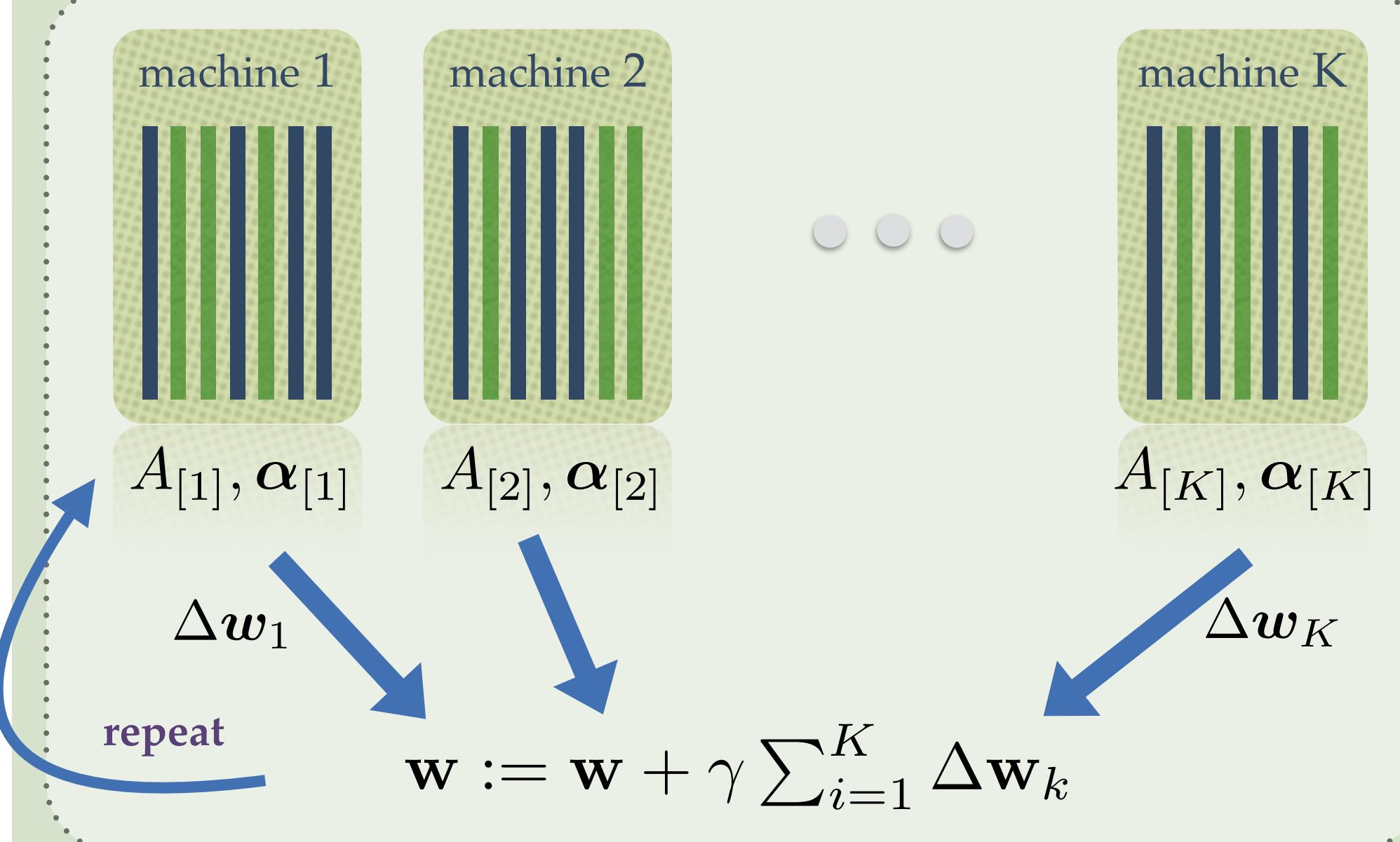
shared

$$\ell_i^*(s) := \sup_{t \in \mathbb{R}} \{st - \ell_i(t)\}$$

## COCOA+

### Main Idea

Propose a *local subproblem* to allow additive, data dependent aggregation



### Local Subproblem

$$\begin{aligned} \max_{\Delta \boldsymbol{\alpha}_{[k]} \in \mathbb{R}^n} & \mathcal{G}_k^{\sigma'}(\Delta \boldsymbol{\alpha}_{[k]}; \mathbf{w}, \boldsymbol{\alpha}_{[k]}) \\ \mathcal{G}_k^{\sigma'}(\Delta \boldsymbol{\alpha}_{[k]}; \mathbf{w}, \boldsymbol{\alpha}_{[k]}) &:= -\frac{1}{n} \sum_{i \in \mathcal{P}_k} \ell_i^*(-\alpha_i - (\Delta \boldsymbol{\alpha}_{[k]})_i) \\ & - \frac{1}{K} \frac{\lambda}{2} \|\mathbf{w}\|^2 - \frac{1}{n} \mathbf{w}^T A \Delta \boldsymbol{\alpha}_{[k]} - \frac{\lambda}{2} \sigma' \left\| \frac{1}{\lambda n} A \Delta \boldsymbol{\alpha}_{[k]} \right\|^2 \end{aligned}$$

- ▷  $\sigma'$  measure of difficulty of data partitioning
- ▷  $\sigma' := \gamma K \in [1, K]$  safe in practice

### CoCoA+ Framework

**Input:** Aggregation parameter  $\gamma \in (0, 1]$ , subproblem parameter  $\sigma'$   
**Initialize:**  $\boldsymbol{\alpha}^{(0)} := \mathbf{0} \in \mathbb{R}^n$ ,  $\mathbf{w}^{(0)} := \mathbf{0} \in \mathbb{R}^d$   
**for**  $t = 0, 1, 2, \dots$  **do**  
    **for**  $k \in \{1, 2, \dots, K\}$  **in parallel over workers do**  
        **call local solver:** compute a  $\Theta$ -approximate solution  $\Delta \boldsymbol{\alpha}_{[k]}$  of the local subproblem  $\mathcal{G}_k^{\sigma'}(\Delta \boldsymbol{\alpha}_{[k]}, \mathbf{w}, \boldsymbol{\alpha}_{[k]})$   
        **update**  $\boldsymbol{\alpha}_{[k]}^{(t+1)} := \boldsymbol{\alpha}_{[k]}^{(t)} + \gamma \Delta \boldsymbol{\alpha}_{[k]}$   
        **return**  $\Delta \mathbf{w}_k := \frac{1}{\lambda n} A \Delta \boldsymbol{\alpha}_{[k]}$   
    **end for**  
    **reduce**  $\mathbf{w}^{(t+1)} := \mathbf{w}^{(t)} + \gamma \sum_{k=1}^K \Delta \mathbf{w}_k$   
**end for**

▷ flexible: can use *arbitrary* local solver

## convergence results

### Local $\Theta$ -Approximation

For  $\Theta \in [0, 1]$ , we assume the local solver finds a (possibly) randomized approximate solution satisfying:

$$\begin{aligned} \mathbb{E}[\mathcal{G}_k^{\sigma'}(\Delta \boldsymbol{\alpha}_{[k]}^*) - \mathcal{G}_k^{\sigma'}(\Delta \boldsymbol{\alpha}_{[k]})] \\ \leq \Theta (\mathcal{G}_k^{\sigma'}(\Delta \boldsymbol{\alpha}_{[k]}^*) - \mathcal{G}_k^{\sigma'}(\mathbf{0})) \end{aligned}$$

**Theorem.** Let  $\ell_i(\cdot)$  be *L-Lipschitz*. Obtain suboptimality  $\epsilon$ , after  $T$  iterations, with:

- ▷ CoCoA, averaging  $\gamma = 1/K$

$$T \geq \tilde{\mathcal{O}} \left( \frac{K}{1-\Theta} \left( \frac{8L^2}{\lambda K \epsilon} + \tilde{c} \right) \right)$$

- ▷ CoCoA+, adding  $\gamma = 1$

$$T \geq \tilde{\mathcal{O}} \left( \frac{1}{1-\Theta} \left( \frac{8L^2}{\lambda \epsilon} + \tilde{c} \right) \right)$$

**Theorem.** Let  $\ell_i(\cdot)$  be  $1/\mu$ -smooth. Obtain suboptimality  $\epsilon$ , after  $T$  iterations, with:

- ▷ CoCoA, averaging  $\gamma = 1/K$

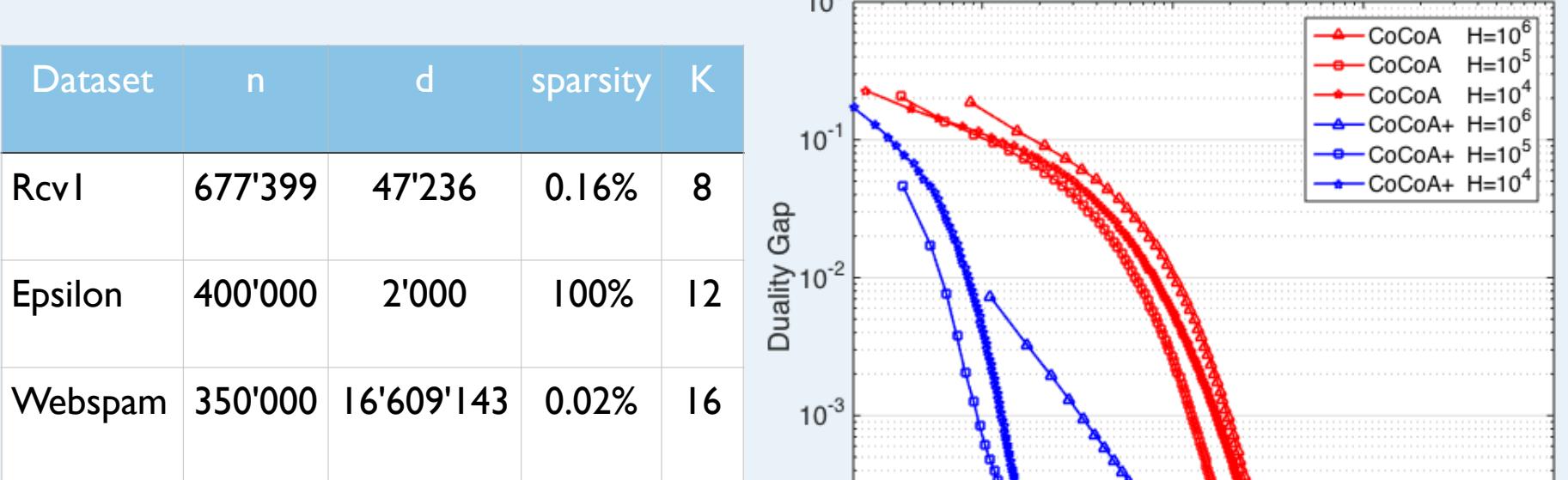
$$T \geq \frac{1}{1-\Theta} \frac{\lambda \mu K + 1}{\lambda \mu} \log \frac{1}{\epsilon}$$

- ▷ CoCoA+, adding  $\gamma = 1$

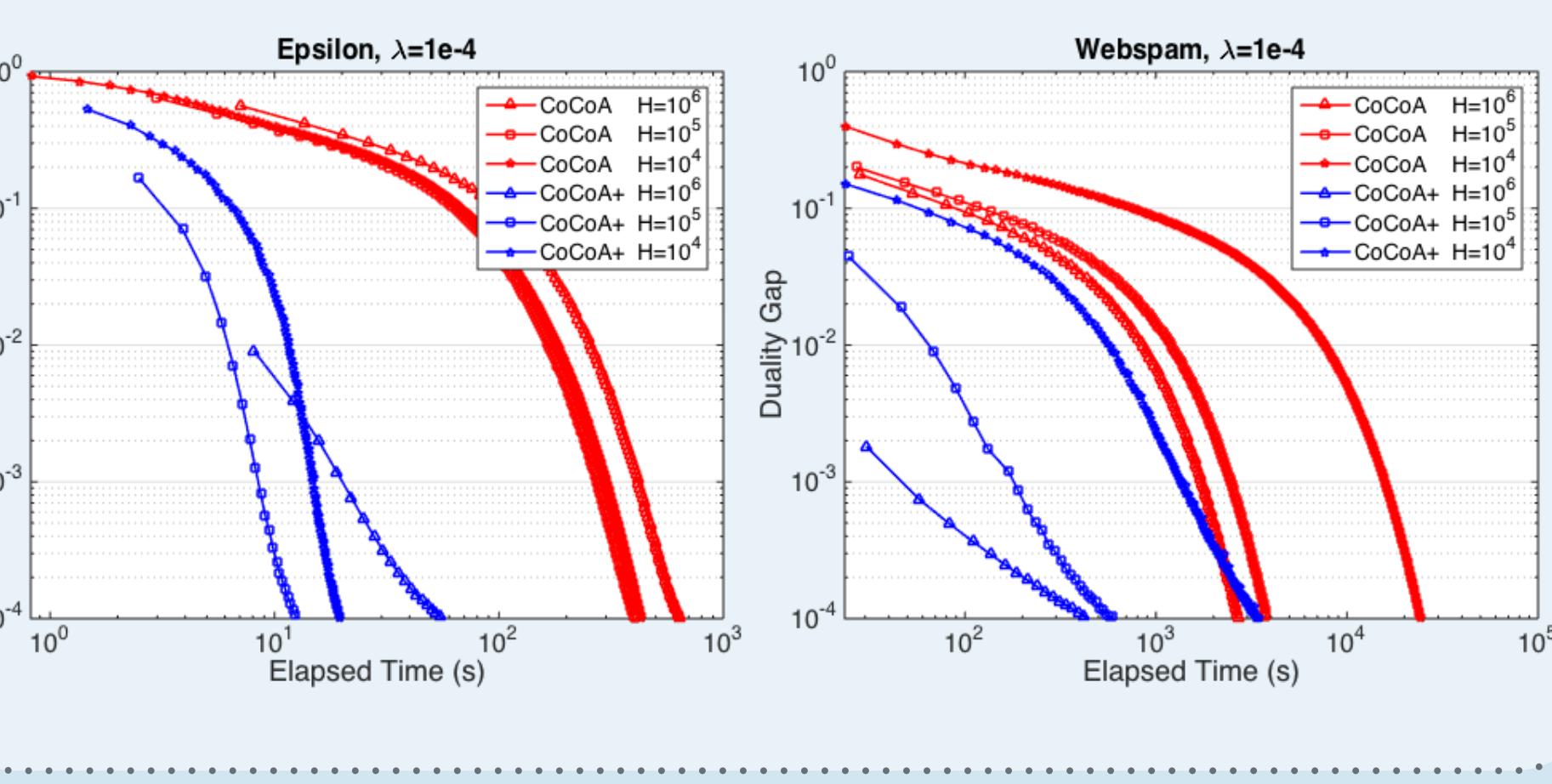
$$T \geq \frac{1}{1-\Theta} \frac{\lambda \mu + 1}{\lambda \mu} \log \frac{1}{\epsilon}$$

- ▷ CoCoA+ rate independent of  $K$
- ▷ applies also to *duality gap*

## experiments in Spark



(example: hinge-loss SVM)

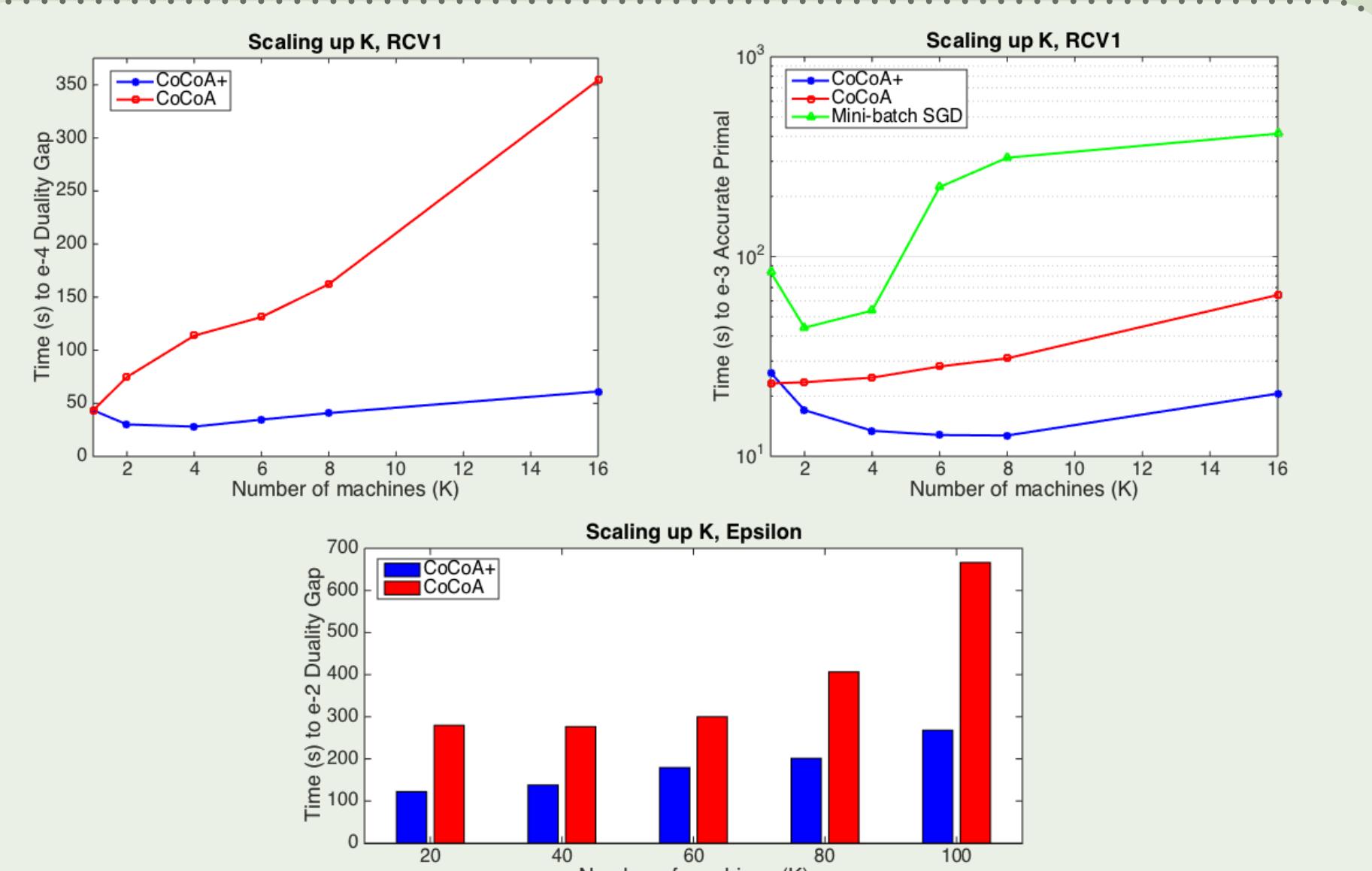


\*Experiments with SDCA as a local solver -> reduces to DisDCA-p [1]

H = number of local updates per round

code at: [github.com/gingsmith/cocoa](https://github.com/gingsmith/cocoa)

## scaling



## references

- [1] Yang. *Trading computation for communication: Distributed stochastic coordinate ascent*. NIPS, 2013.
- [2] Shalev-Shwartz and Zhang. *Stochastic dual coordinate ascent methods for regularized loss minimization*. JMLR, 14:567–599, 2013.
- [3] Jaggi et. al., *Communication-Efficient Distributed Dual Coordinate Ascent*. NIPS, 2014.