CADA: Communication-Adaptive Distributed Adam

Tianyi Chen CHENT18@RPI.EDU
Rensselaer Polytechnic Institute
Ziye Guo GUOZ8@RPI.EDU
Rensselaer Polytechnic Institute
Yuejiao Sun SUNYJ@MATH.UCLA.EDU
University of California, Los Angeles
Wotao Yin WOTAOYIN@MATH.UCLA.EDU
University of California, Los Angeles

Abstract

Stochastic gradient descent (SGD) has taken the stage as the primary workhorse for large-scale machine learning. It is often used with its adaptive variants such as AdaGrad, Adam, and AMSGrad. This paper proposes an adaptive stochastic gradient descent method for distributed machine learning, which can be viewed as the communication-adaptive counterpart of the celebrated Adam method — justifying its name CADA. The key components of CADA are a set of new rules tailored for stochastic gradients that can be implemented to save communication upload. The new algorithms adaptively reuse stale Adam gradients, thus saving communication, and still have convergence rates comparable to original Adam. In numerical experiments, CADA achieves impressive empirical performance at a 60% total communication reduction on average.

1. Introduction

Although simple to use, the plain-vanilla stochastic gradient descent (SGD) method [21] is often sensitive to the choice of hyper-parameters and sometimes suffer from the slow convergence. Among various efforts to improve SGD, adaptive methods such as AdaGrad [5], Adam [14] and AMSGrad [19] have impressive empirical performance, especially in training deep neural networks.

To achieve "adaptivity," these algorithms adaptively adjust the *update direction* or tune the *learning rate*, or, the combination of both. While existing studies on adaptive SGD have focused on the setting where data and computation are both centralized in a single node, this paper considers their implementation in the distributed setting. Since this setting often brings new challenges to machine learning, can we add an *additional dimension of adaptivity* to Adam in this regime?

We consider the setting composed of a central server and a set of M workers in $\mathcal{M} := \{1, \ldots, M\}$, where each worker m has its local data ξ_m from a distribution Ξ_m. Workers may have different data distributions $\{\Xi_m\}$, and they collaboratively solve the following problem

$$\min_{\theta\in\mathbb{R}^p} \mathcal{L}(\theta) = \frac{1}{M}\sum_{m\in\mathcal{M}} \mathcal{L}_m(\theta) \;\text{ with }\; \mathcal{L}_m(\theta) := \mathbb{E}_{\xi_m}[\ell(\theta;\xi_m)]\,, \quad m\in\mathcal{M} \tag{1}$$

where $\theta \in \mathbb{R}^p$ is the sought variable and $\{\mathcal{L}_m, m \in \mathcal{M}\}$ are smooth (but not necessarily convex) functions. We focus on the setting where local data ξ_m at each worker m can not be uploaded to the

server, and collaboration is needed through communication between the server and workers. This setting often emerges due to the data privacy concerns, e.g., federated learning [9, 17].

To solve (1), we can in principle apply the single-node version of the adaptive SGD methods such as Adam [14]: At iteration k, the server broadcasts θ^k to *all* the workers; each worker m computes $\nabla\ell(\theta^k;\xi_m^k)$ using a randomly selected sample or a minibatch of samples $\{\xi_m^k\}\sim\Xi_m$, and then uploads it to the server; and once receiving stochastic gradients from all workers, the server can simply use the aggregated stochastic gradient $\bar{\boldsymbol{\nabla}}^k=\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^k;\xi_m^k)$ to update the parameter via the plain-vanilla single-node Adam. To implement this, however, *all* the workers have to *upload* the fresh $\{\nabla\ell(\theta^k;\xi_m^k)\}$ at each iteration. This prevents the efficient implementation of Adam in scenarios where the communication uplink and downlink are not symmetric, and communication especially upload from workers and the server is costly; e.g., cellular networks [18]. Therefore, *our goal* is to endow an additional dimension of adaptivity to Adam for saving uplink communication.

Related work

Adaptive SGD. Adaptive learning rate methods have been developed that scale the gradient in an entry-wise manner by using past gradients, which include AdaGrad [5, 26], AdaDelta [29] and other variants [15]. This simple technique has markedly improved the performance of SGD. Adaptive SGD methods update the search directions and the learning rates simultaneously using past gradients. Adam [14] and AMSGrad [19] are the representative ones in this category. While these methods are simple-to-use, analyzing their convergence is challenging [19]. Their convergence in the nonconvex setting has been settled only recently [3, 4]. Except [27], most adaptive SGD methods are studied in the single-node setting where data and computation are both centralized.

Communication-efficient SGD. One of the most popular techniques in this category is the periodic averaging, e.g., elastic averaging SGD [30], local SGD (a.k.a. FedAvg) [7, 10, 12, 13, 16, 17, 22, 24] or local momentum SGD [25, 28]. In local SGD, workers perform local model updates independently and the models are averaged periodically. However, except [7, 10, 24], most of local SGD methods follow a pre-determined communication schedule that is nonadaptive. Some of them are tailored for the *homogeneous* settings, where the data are identically distributed over all workers. The effect of data heterogeneity on local update methods has been discussed in, e.g., [13].

The most related line of work to this paper is the lazily aggregated gradient (LAG) approach [2, 23]. Unfortunately, the performance of LAG will be degraded when using stochastic gradients. Our approach generalizes LAG to the regime of running adaptive SGD. Very recently, FedAvg with local adaptive SGD update has been proposed in [20]. When the new algorithm achieves the sweet spot between local SGD and adaptive momentum SGD, the proposed algorithm is very different from ours, and the *averaging period* and the selection of *participating workers* are nonadaptive.

2. CADA: Communication-Adaptive Distributed Adam

We develop a new adaptive SGD algorithm for distributed learning, called **C**ommunication-**A**daptive **D**istributed **A**dam (**CADA**). Akin to the dynamic scaling of every gradient coordinate in Adam, the idea of adaptive communication is motivated by that during distributed learning, not all communication rounds between the server and workers are equally important. So a natural solution is to use a condition that decides whether the communication is important or not, and then adjust the frequency of communication between a worker and the server.

Analogous to the original Adam [14] and AMSGrad [19], our new CADA approach also uses the exponentially weighted stochastic gradient h^{k+1} as the update direction of θ^{k+1}, and leverages the weighted stochastic gradient magnitude v^{k+1} to inversely scale the update direction h^{k+1}. Different from the direct distributed implementation of Adam that incorporates the fresh (thus unbiased) stochastic gradients $\bar{\nabla}^k = \frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^k;\xi_m^k)$, CADA exponentially combines the aggregated stale stochastic gradients $\nabla^k = \frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\hat{\theta}_m^k;\hat{\xi}_m^k)$, where $\nabla\ell(\hat{\theta}_m^k;\hat{\xi}_m^k)$ is either the fresh stochastic gradient $\nabla\ell(\theta^k;\xi_m^k)$, or an old copy when $\hat{\theta}_m^k \neq \theta^k; \hat{\xi}_m^k \neq \xi_m^k$. Informally, with $\alpha_k > 0$ denoting the stepsize at iteration k, CADA has the following update

$$h^{k+1} = \beta_1 h^k + (1-\beta_1)\nabla^k, \quad \text{with} \quad \nabla^k = \frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\hat{\theta}_m^k;\hat{\xi}_m^k) \tag{2a}$$

$$v^{k+1} = \beta_2\hat{v}^k + (1-\beta_2)(\nabla^k)^2 \tag{2b}$$

$$\theta^{k+1} = \theta^k - \alpha_k(\epsilon I + \hat{V}^{k+1})^{-\frac{1}{2}}h^{k+1} \tag{2c}$$

where $\beta_1, \beta_2 > 0$ are the momentum weights, $\hat{V}^{k+1} := \text{diag}(\hat{v}^{k+1})$ is a diagonal matrix whose diagonal vector is $\hat{v}^{k+1} := \max\{v^{k+1}, \hat{v}^k\}$, the constant is $\epsilon > 0$, and I is an identity matrix. To reduce the memory requirement of storing all the stale stochastic gradients $\{\nabla\ell(\theta^k;\xi_m^k)\}$, we can obtain ∇^k by refining the previous aggregated stochastic gradients ∇^{k-1} stored in the server via

$$\nabla^k = \nabla^{k-1} + \frac{1}{M}\sum_{m\in\mathcal{M}^k}\delta_m^k \tag{3}$$

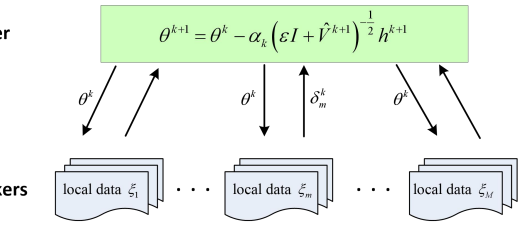

Figure 1: The CADA implementation.

where $\delta_m^k := \nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\hat{\theta}_m^k;\hat{\xi}_m^k)$ is the stochastic gradient innovation, and $\mathcal{M}^k$ is the set of workers that upload the stochastic gradient to the server at iteration k. See CADA's implementation in Figure 1 and the pseudo-code in Algorithm 1.

We formally develop our CADA method, and present the intuition behind its design. To be more precise in our notations, we henceforth use $\tau_m^k \geq 0$ for the *staleness or age of the information* from worker m used by the server at iteration k, e.g., $\hat{\theta}_m^k = \theta^{k-\tau_m^k}$. An age of 0 means "fresh."

The first one termed **CADA1** will calculate two stochastic gradient innovations with one $\tilde{\delta}_m^k := \nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\tilde{\theta};\xi_m^k)$ at the sample ξ_m^k, and one $\tilde{\delta}_m^{k-\tau_m^k} := \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k})$ at the sample $\xi_m^{k-\tau_m^k}$, where $\tilde{\theta}$ is a snapshot of the previous iterate θ that will be updated every D iterations. CADA1 will exclude worker m from $\mathcal{M}^k$ at iteration k if worker m finds

$$\left\|\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}\right\|^2 \leq c\sum_{d=1}^{D}\left\|\theta^{k+1-d} - \theta^{k-d}\right\|^2. \tag{4}$$

If (4) is satisfied, worker m does not upload, and the staleness increases by $\tau_m^{k+1} = \tau_m^k + 1$; otherwise, worker m belongs to $\mathcal{M}^k$, uploads the stochastic gradient innovation δ_m^k, and resets $\tau_m^{k+1} = 1$.

In addition to (4), the second rule that we term **CADA2** will reuse the stale stochastic gradient $\nabla\ell(\theta_m^{k-\tau_m^k};\xi_m^{k-\tau_m^k})$ or exclude worker m from $\mathcal{M}^k$ if worker m finds

$$\left\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta_m^{k-\tau_m^k};\xi_m^k)\right\|^2 \leq c\sum_{d=1}^{D}\left\|\theta^{k+1-d} - \theta^{k-d}\right\|^2. \tag{5}$$

Algorithm 1 Pseudo-code of CADA; red lines are run only by CADA1; blue lines are implemented only by CADA2; not both at the same time.

Input: counter $\{\tau_m^0\}$, stepsize α_k, c, max delay D.
for $k = 0, 1, \ldots, K-1$ **do**
  Server broadcasts θ^k to all workers.
  CADA1: All workers set $\tilde{\theta} = \theta^k$ if $k \bmod D = 0$.
  for Worker $m = 1, 2, \ldots, M$ **do in parallel**
    CADA1:
      Compute $\nabla\ell(\theta^k; \xi_m^k)$ and $\nabla\ell(\tilde{\theta}; \xi_m^k)$.
      Check condition (4) with stored $\tilde{\delta}_m^{k-\tau_m^k}$.
    CADA2:
      Compute $\nabla\ell(\theta^k; \xi_m^k)$ and $\nabla\ell(\theta_m^{k-\tau_m^k}; \xi_m^k)$.
      Check condition (5).
    if (4) or (5) is violated, or, $\tau_m^k \geq D$ **then**
      Upload δ_m^k. ▷ $\tau_m^{k+1} = 1$
    else
      Upload nothing. ▷ $\tau_m^{k+1} = \tau_m^k + 1$
    end if
  end for
  Server updates $\{h^k, v^k\}$ via (2a)-(2b).
  Server updates θ^k via (2c).
end for

If (5) is satisfied, then worker m does not upload, and the staleness increases by $\tau_m^{k+1} = \tau_m^k + 1$; otherwise, worker m uploads the stochastic gradient innovation δ_m^k, and resets the staleness as $\tau_m^{k+1} = 1$. Notice that (5) is evaluated at two different iterates but on the same sample ξ_m^k.

3. Convergence Analysis of CADA

We present the convergence results of CADA. For all the results, we make some basic assumptions, which are standard in analyzing Adam and its variants [3, 14, 19, 27].

Assumption 1 *The loss function $\mathcal{L}(\theta)$ is smooth with the constant L.*

Assumption 2 *Samples $\xi_m^1, \xi_m^2, \ldots$ are independent, and the stochastic gradient $\nabla\ell(\theta; \xi_m^k)$ satisfies $\mathbb{E}_{\xi_m^k}[\nabla\ell(\theta; \xi_m^k)] = \nabla\mathcal{L}_m(\theta)$ and $\|\nabla\ell(\theta; \xi_m^k)\| \leq \sigma_m$. And define $\sigma := \frac{1}{M}\sum_{m\in\mathcal{M}}\sigma_m$.*

We will start with analyzing the expected descent in $\mathcal{L}(\theta^k)$ by applying one CADA update.

Lemma 1 *Under Assumptions 1 and 2, if $\alpha_k \leq \alpha_{k+1}$, then $\{\theta^k\}$ generated by CADA satisfy*

$$\mathbb{E}[\mathcal{L}(\theta^{k+1})] - \mathbb{E}[\mathcal{L}(\theta^k)] \leq \left(\frac{L}{2} + \beta_1 L\right)\mathbb{E}\left[\|\theta^{k+1} - \theta^k\|^2\right] - \alpha_k(1-\beta_1)\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\boldsymbol{\nabla}^k\right\rangle\right]$$
$$- \alpha_k\beta_1\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^{k-1}), (\epsilon I + \hat{V}^k)^{-\frac{1}{2}}h^k\right\rangle\right] + \alpha_k(2-\beta_1)\sigma^2\mathbb{E}\Big[\sum_{i=1}^{p}\left((\epsilon + \hat{v}_i^{k-D})^{-\frac{1}{2}} - (\epsilon + \hat{v}_i^{k+1})^{-\frac{1}{2}}\right)\Big]. \quad (6)$$

Lemma 1 contains four terms in the RHS of (6): the first term captures the drift of two consecutive iterates; the second and third terms quantify the correlations between the gradient direction $\nabla\mathcal{L}(\theta^k)$ and the *stale* stochastic gradient $\boldsymbol{\nabla}^k$ as well as the *state momentum* stochastic gradient h^k; and, the last term estimates the maximum drift of the adaptive stepsizes over $D+1$ iterations. The following lemma characterizes the regularity of the stale aggregated stochastic gradients $\boldsymbol{\nabla}^k$.

Lemma 2 *Under Assumptions 1 and 2, if the stepsizes satisfy $\alpha_{k+1} \le \alpha_k \le 1/L$, then we have*

$$-\alpha_k\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\boldsymbol{\nabla}^k\right\rangle\right] \le -\frac{\alpha_k}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right]+\frac{6DL\alpha_k^2\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2 + \epsilon^{-\frac{1}{2}}\left(\frac{L}{12}+\frac{c}{2L}\right)\sum_{d=1}^{D}\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]. \tag{7}$$

Lemma 2 justifies the relevance of the stale yet properly selected stochastic gradients. Intuitively, the first term in the RHS of (7) resembles the descent of using SGD with the unbiased gradient, and the second and third terms will diminish if the stepsizes are diminishing.

In view of Lemmas 1 and 2, we introduce the following **Lyapunov function**:

$$\mathcal{V}^k := \mathcal{L}(\theta^k)-\mathcal{L}(\theta^\star)-\sum_{j=k}^{\infty}\alpha_j\beta_1^{j-k+1}\left\langle\nabla\mathcal{L}(\theta^{k-1}),(\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\right\rangle + b_k\sum_{d=0}^{D}\sum_{i=1}^{p}(\epsilon+\hat{v}_i^{k-d})^{-\frac{1}{2}}+\sum_{d=1}^{D}\rho_d\|\theta^{k+1-d}-\theta^{k-d}\|^2 \tag{8}$$

where $\theta^\star$ is the solution of (1), $\{b_k\}_{k=1}^K$ and $\{\rho_d\}_{d=1}^D$ are constants that will be specified in the proof.

The following lemma captures the progress of the Lyapunov function.

Lemma 3 *Under Assumptions 1-2, if $\{b_k\}_{k=1}^K$ and $\{\rho_d\}_{d=1}^D$ in (8) are chosen properly, we have*

$$\mathbb{E}[\mathcal{V}^{k+1}]-\mathbb{E}[\mathcal{V}^k] \le -\frac{\alpha_k(1-\beta_1)}{2}\left(\epsilon+\frac{\sigma^2}{1-\beta_2}\right)^{-\frac{1}{2}}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2\right]+\alpha_k^2C_0 \tag{9}$$

where the constant C_0 depends on the CADA and problem parameters $c,\beta_1,\beta_2,\epsilon,D$, and $L,\{\sigma_m^2\}$.

Lemma 3 is a generalization of SGD's descent lemma. If we set $\beta_1=\beta_2=0$ in (2) and $b_k=0,\rho_d=0,\ \forall d,k$ in (8), then Lemma 3 reduces to that of SGD in terms of $\mathcal{L}(\theta^k)$; see e.g., [1, Lemma 4.4].

Building upon our Lyapunov analysis, we first present the convergence in nonconvex case.

Theorem 4 (nonconvex) *Under Assumptions 1, 2, if we choose $\alpha_k=\alpha=\mathcal{O}(\frac{1}{\sqrt{K}})$ and $\beta_1<\sqrt{\beta_2}<1$, then the iterates $\{\theta^k\}$ generated by CADA satisfy*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]=\mathcal{O}\left(\frac{1}{\sqrt{K}}\right). \tag{10}$$

From Theorem 4, the convergence rate of CADA in terms of the average gradient norms is $\mathcal{O}(1/\sqrt{K})$, which matches that of the plain-vanilla Adam [3, 19].

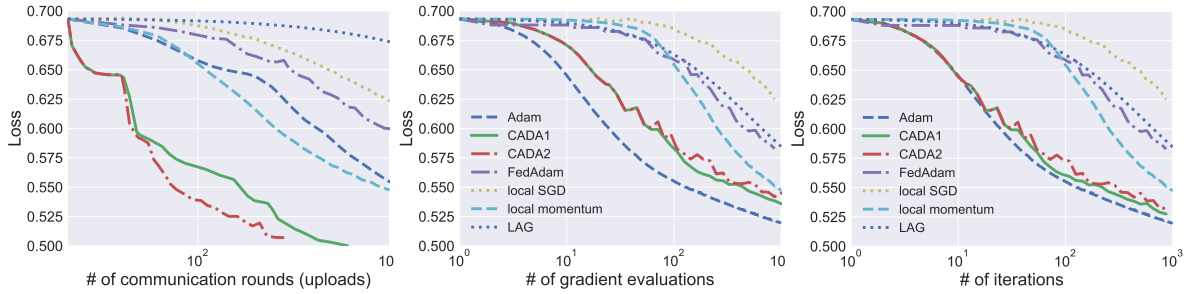

Figure 2: Logistic regression on *covtype* dataset.

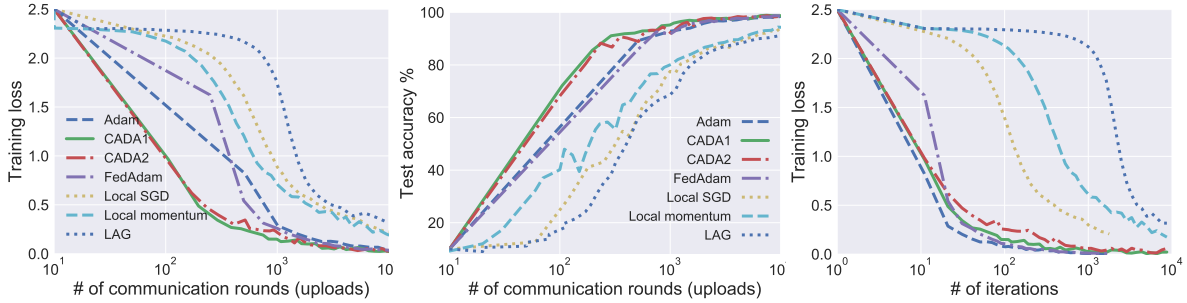

Figure 3: Training Neural network for classification on *mnist* dataset.

4. Simulations

In order to verify our analysis and show the empirical performance of CADA, we conduct simulations using logistic regression and training neural networks. Data are distributed across $M = 10$ workers during all tests. We benchmark CADA with some popular methods Adam [14], stochastic version of LAG [2], local SGD (or FedAvg) [17], local momentum [28] and FedAdam [20]. For local SGD, local momentum and FedAdam, workers perform model update independently, which are averaged over all workers every H iterations. In simulations, stepsizes are optimized for each algorithm by a grid-search. Due to space limitation, please see the data allocation, the detailed choice of parameters, and additional experiments on *CIFAR10* dataset **in Appendix**.

Tests on logistic regression are reported in Figure 2, and tests on training neural networks are reported in Figure 3. In our tests, two CADA variants achieve the similar iteration complexity as the original Adam and outperform all other baselines in most cases. Since our CADA requires two gradient evaluations per iteration, the gradient complexity of CADA is higher than Adam, but still not more than that of other baselines. For logistic regression task, CADA1 and CADA2 save the number of communication uploads by at least one order of magnitude; and for neural network training, the saving is about 60%. Based on this results, the CADA1 and CADA2 rules achieve more saving in terms of communication rounds than the direct stochastic version of LAG.

References

[1] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[2] Tianyi Chen, Georgios Giannakis, Tao Sun, and Wotao Yin. LAG: Lazily aggregated gradient for communication-efficient distributed learning. In *Proc. Conf. Neural Info. Process. Syst.*, pages 5050–5060, Montreal, Canada, Dec 2018.

[3] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of Adam-type algorithms for non-convex optimization. In *Proc. Intl. Conf. Learn. Representations*, New Orleans, LA, May 2019.

[4] Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. On the convergence of Adam and Adagrad. *arXiv preprint:2003.02395*, March 2020.

[5] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Machine Learning Res.*, 12(Jul):2121–2159, 2011.

[6] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[7] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. In *Proc. Conf. Neural Info. Process. Syst.*, pages 11080–11092, Vancouver, Canada, December 2019.

[8] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Conf. Neural Info. Process. Syst.*, pages 315–323, 2013.

[9] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint:1912.04977*, December 2019.

[10] Michael Kamp, Linara Adilova, Joachim Sicking, Fabian Hüger, Peter Schlicht, Tim Wirtz, and Stefan Wrobel. Efficient decentralized deep learning by dynamic model averaging. In *Euro. Conf. Machine Learn. Knowledge Disc. Data.,*, pages 393–409, Dublin, Ireland, 2018.

[11] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Proc. Euro. Conf. Machine Learn.*, pages 795–811, Riva del Garda, Italy, September 2016.

[12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. In *Proc. Intl. Conf. Machine Learn.*, July 2020.

[13] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *arXiv preprint:1909.04715*, September 2019.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980*, December 2014.

[15] Xiaoyu Li and Francesco Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *Proc. Intl. Conf. on Artif. Intell. and Stat.*, pages 983–992, Okinawa, Japan, April 2019.

[16] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *Proc. Intl. Conf. Learn. Representations*, Addis Ababa, Ethiopia, April 2020.

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. Intl. Conf. Artificial Intell. and Stat.*, pages 1273–1282, Fort Lauderdale, FL, April 2017.

[18] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. Wireless network intelligence at the edge. *Proc. of the IEEE*, 107(11):2204–2239, November 2019.

[19] Sashank Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *Proc. Intl. Conf. Learn. Representations*, Vancouver, Canada, April 2018.

[20] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint:2003.00295*, March 2020.

[21] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, September 1951.

[22] Sebastian Urban Stich. Local SGD converges fast and communicates little. In *Proc. Intl. Conf. Learn. Representations*, New Orleans, LA, May 2019.

[23] Jun Sun, Tianyi Chen, Georgios Giannakis, and Zaiyue Yang. Communication-efficient distributed learning via lazily aggregated quantized gradients. In *Proc. Conf. Neural Info. Process. Syst.*, page to appear, Vancouver, Canada, Dec 2019.

[24] Jianyu Wang and Gauri Joshi. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. In *ICML Workshop on Coding Theory for Large-Scale ML*, Long Beach, CA, June 2019.

[25] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. SlowMo: Improving communication-efficient distributed SGD with slow momentum. In *Proc. Intl. Conf. Learn. Representations*, 2020.

[26] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *Proc. Intl. Conf. Machine Learn.*, pages 6677–6686, Long Beach, CA, June 2019.

[27] Yangyang Xu, Colin Sutcher-Shepard, Yibo Xu, and Jie Chen. Asynchronous parallel adaptive stochastic gradient methods. *arXiv preprint:2002.09095*, February 2020.

[28] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *Proc. Intl. Conf. Machine Learn.*, pages 7184–7193, Long Beach, CA, June 2019.

[29] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint:1212.5701*, December 2012.

[30] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging SGD. In *Proc. Conf. Neural Info. Process. Syst.*, pages 685–693, Montreal, Canada, Dec 2015.

Supplementary materials for "CADA: Communication-Adaptive Distributed Adam"

In this supplementary document, we first compare CADA and the direct stochastic extension of LAG, and then present the missing derivations of some claims, as well as the proofs of all the lemmas and theorems in the paper, which is followed by details on our experiments.

Appendix A. Rationale of CADA

A.1. Why LAG with stochastic gradients does not work?

The LAG method [2] modifies the distributed gradient descent update. Instead of communicating with all workers per iteration, LAG selects the subset of workers $\mathcal{M}^k$ to obtain *fresh* full gradients and reuses stale full gradients from others, that is, $\theta^{k+1} = \theta^k - \frac{\eta_k}{M}\sum_{m\in\mathcal{M}\setminus\mathcal{M}^k}\nabla\mathcal{L}_m(\theta^{k-\tau_m^k}) - \frac{\eta_k}{M}\sum_{m\in\mathcal{M}^k}\nabla\mathcal{L}_m(\theta^k)$, where $\mathcal{M}^k$ is adaptively decided by comparing the gradient difference $\|\nabla\mathcal{L}_m(\theta^k) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\|$. Following this principle, the direct (or "naive") stochastic version of LAG selects the subset of workers $\mathcal{M}^k$ to obtain *fresh* stochastic gradients $\nabla\mathcal{L}_m(\theta^k;\xi_m^k)$, $m\in\mathcal{M}^k$. The **stochastic LAG** also follows the distributed SGD update, but it selects $\mathcal{M}^k$ by: if worker m finds the innovation of the fresh stochastic gradient $\nabla\ell(\theta^k;\xi_m^k)$ is small such that it satisfies

$$\left\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2 \le c\sum_{d=1}^{D}\left\|\theta^{k+1-d} - \theta^{k-d}\right\|^2 \tag{11}$$

where $c\ge 0$ and D are pre-fixed constants, then worker m reuses the old gradient, $m\in\mathcal{M}\setminus\mathcal{M}^k$, and sets the staleness $\tau_m^{k+1} = \tau_m^k + 1$; otherwise, worker m uploads the fresh gradient, and sets $\tau_m^{k+1} = 1$.

In the deterministic setting, LAG condition (11) is motivated by the elegant *"larger descent per upload"* rationale, and has proved to be effective [2]. Nevertheless, the observation here is that the two stochastic gradients (11) are evaluated on not just two different iterates (θ^k and $\theta^{k-\tau_m^k}$) but also two different samples (ξ_m^k and $\xi_m^{k-\tau_m^k}$) thus two different loss functions. This is in contrast to the original LAG in [2] where the gradient innovation is evaluated on the same function.

This subtle difference leads to the ineffectiveness of (11). We can see this by expanding the left-hand-side (LHS) of (11) by (see the details in supplemental material)

$$\mathbb{E}\left[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\|^2\right] \ge \frac{1}{2}\mathbb{E}\left[\left\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\mathcal{L}_m(\theta^k)\right\|^2\right] \tag{12a}$$

$$+\frac{1}{2}\mathbb{E}\left[\left[\left\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right\|^2\right]\right] \tag{12b}$$

$$-\mathbb{E}[\|\nabla\mathcal{L}_m(\theta^k) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\|^2]. \tag{12c}$$

Even if θ^k converges, e.g., $\theta^k\to\theta^*$, and thus the right-hand-side (RHS) of (11) $\left\|\theta^{k+1-d} - \theta^{k-d}\right\|^2 \to 0$, the LHS of (11) does not, because the variance inherited in (12a) and (12b) does not vanish yet the gradient difference at the same function (12c) diminishes. Therefore, the key insight here is that the non-diminishing variance of stochastic gradients makes the LAG rule (11) ineffective eventually. This will also be verified in our simulations when we compare CADA with stochastic LAG.

Appendix B. Why CADA rules can work with stochastic gradients?

The rationale of CADA1. In contrast to the non-vanishing variance in LAG rule (see (12)), the CADA1 rule (4) reduces its inherent variance. To see this, we can decompose the LHS of (4) as

the difference of two *variance reduced* stochastic gradients at iteration k and $k-\tau_m^k$. Using the stochastic gradient in SVRG as an example [8], the innovation can be written as

$$\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k} = \left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\tilde{\theta};\xi_m^k) + \nabla\mathcal{L}_m(\tilde{\theta})\right) - \left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k}) + \nabla\mathcal{L}_m(\tilde{\theta})\right). \quad (13)$$

Define the minimizer of (1) as $\theta^\star$. With derivations given in the supplementary document, the expectation of the LHS of (4) can be *upper-bounded* by

$$\mathbb{E}\left[\left\|\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}\right\|^2\right] = \mathcal{O}\left(\mathbb{E}[\mathcal{L}(\theta^k)] - \mathcal{L}(\theta^\star) + \mathbb{E}[\mathcal{L}(\theta^{k-\tau_m^k})] - \mathcal{L}(\theta^\star) + \mathbb{E}[\mathcal{L}(\tilde{\theta})] - \mathcal{L}(\theta^\star)\right). \quad (14)$$

If θ^k converges, e.g., $\theta^k, \theta^{k-\tau_m^k}, \tilde{\theta} \to \theta^*$, the RHS of (14) diminishes, and thus the LHS of (4) diminishes. This is in contrast to the LAG rule (12) *lower-bounded* by a non-vanishing value. Notice that while enjoying the benefit of variance reduction, our communication rule does not need to repeatedly calculate the full gradient $\nabla\mathcal{L}_m(\tilde{\theta})$.

The rationale of CADA2. Similar to CADA1, the CADA2 rule (5) also reduces its inherent variance, since the LHS of (5) can be written as the difference between a *variance reduced* stochastic gradient and a *deterministic* gradient, that is

$$\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) = \left(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\right) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k}). \quad (15)$$

With derivations deferred to the supplementary document, similar to (14) we can also conclude that $\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k)\|^2] \to 0$ as the iterate $\theta^k \to \theta^\star$.

For either (4) or (5), worker m can check it locally with small memory cost by recursively updating the RHS of (4) or (5). In addition, worker m will update the stochastic gradient if the staleness satisfies $\tau_m^k \geq D$. We summarize CADA1 and CADA2 in Algorithm 1.

Appendix C. Missing Derivations

The analysis in this part is analogous to that in [6]. We define an auxiliary function as

$$\psi_m(\theta) = \mathcal{L}_m(\theta) - \mathcal{L}_m(\theta^\star) - \left\langle \nabla\mathcal{L}_m(\theta^\star), \theta - \theta^\star \right\rangle$$

where $\theta^\star$ is a minimizer of $\mathcal{L}$. Assume that $\nabla\ell(\theta;\xi_m)$ is $\bar{L}$-Lipschitz continuous for all ξ_m, we have

$$\|\nabla\ell(\theta;\xi_m) - \nabla\ell(\theta^\star;\xi_m)\|^2 \leq 2\bar{L}\left(\ell(\theta;\xi_m) - \ell(\theta^\star;\xi_m) - \left\langle \nabla\ell(\theta^\star;\xi_m), \theta - \theta^\star \right\rangle\right).$$

Taking expectation with respect to ξ_m, we can obtain

$$\mathbb{E}_{\xi_m}[\|\nabla\ell(\theta;\xi_m) - \nabla\ell(\theta^\star;\xi_m)\|^2] \leq 2\bar{L}\left(\mathcal{L}_m(\theta) - \mathcal{L}_m(\theta^\star) - \left\langle \nabla\mathcal{L}_m(\theta^\star), \theta - \theta^\star \right\rangle\right) = 2\bar{L}\psi_m(\theta).$$

Note that $\nabla\mathcal{L}_m$ is also $\bar{L}$-Lipschitz continuous and thus

$$\|\nabla\mathcal{L}_m(\theta) - \nabla\mathcal{L}_m(\theta^\star)\|^2 \leq 2\bar{L}(\mathcal{L}_m(\theta) - \mathcal{L}_m(\theta^\star) - \left\langle \nabla\mathcal{L}_m(\theta^\star), \theta - \theta^\star \right\rangle) = 2\bar{L}\psi_m(\theta).$$

C.1. Derivations of (12)

By (56), we can derive that

$$\|\theta_1+\theta_2\| \le 2\|\theta_1\|^2+2\|\theta_2\|^2$$

which also implies $\|\theta_1\|^2 \ge \frac{1}{2}\|\theta_1+\theta_2\|^2-\|\theta_2\|^2$.

As a consequence, we can obtain

$$\begin{aligned}
&\mathbb{E}\Big[\big\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\big\|^2\Big]\\
\ge&\frac{1}{2}\mathbb{E}\Big[\big\|\big(\nabla\ell(\theta^k;\xi_m^k)-\nabla\mathcal{L}_m(\theta^k)\big)+\big(\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\big)\big\|^2\Big]\\
&\qquad-\mathbb{E}\Big[\big\|\nabla\mathcal{L}_m(\theta^k)-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\big\|^2\Big]\\
=&\frac{1}{2}\mathbb{E}\Big[\big\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\mathcal{L}_m(\theta^k)\big\|^2\Big]+\frac{1}{2}\mathbb{E}\Big[\big[\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\big\|^2\big]\Big]\\
&\underbrace{+\mathbb{E}\Big[\Big\langle\nabla\ell(\theta^k;\xi_m^k)-\nabla\mathcal{L}_m(\theta^k),\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\Big\rangle\Big]}_{I_3}-\mathbb{E}\Big[\big\|\nabla\mathcal{L}_m(\theta^k)-\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\big\|^2\Big]
\end{aligned}$$

where we used the fact that $I_3=0$ to obtain (12), that is

$$I_3=\mathbb{E}\Big[\Big\langle\mathbb{E}\big[\nabla\ell(\theta^k;\xi_m^k)\big|\Theta^k\big]-\nabla\mathcal{L}_m(\theta^k),\nabla\mathcal{L}_m(\theta^{k-\tau_m^k})-\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\Big\rangle\Big]=0.$$

C.2. Derivations of (14)

Recall that

$$\begin{aligned}
\tilde\delta_m^k-\tilde\delta_m^{k-\tau_m^k}=&\big(\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\tilde\theta;\xi_m^k)+\nabla\mathcal{L}_m(\tilde\theta)\big)-\big(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\ell(\tilde\theta;\xi_m^{k-\tau_m^k})+\nabla\mathcal{L}_m(\tilde\theta)\big)\\
=&\underbrace{\big(\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\tilde\theta;\xi_m^k)+\nabla\psi_m(\tilde\theta)\big)}_{g_m^k}-\underbrace{\big(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\ell(\tilde\theta;\xi_m^{k-\tau_m^k})+\nabla\psi_m(\tilde\theta)\big)}_{g_m^{k-\tau_m^k}}.
\end{aligned}$$

And by (56), we have $\|\tilde\delta_m^k-\tilde\delta_m^{k-\tau_m^k}\|^2\le 2\|g_m^k\|^2+2\|g_m^{k-\tau_m^k}\|^2$. We decompose the first term as

$$\begin{aligned}
\mathbb{E}[\|g_m^k\|^2]\le&2\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)\|^2]+2\mathbb{E}[\|\nabla\ell(\tilde\theta;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)-\nabla\psi_m(\tilde\theta)\|^2]\\
=&2\mathbb{E}[\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)\|^2|\Theta^k]]\\
&+2\mathbb{E}[\|\nabla\ell(\tilde\theta;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)-\mathbb{E}[\nabla\ell(\tilde\theta;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)|\Theta^k]\|^2]\\
\le&4\bar L\mathbb{E}\psi_m(\theta^k)+2\mathbb{E}[\|\nabla\ell(\tilde\theta;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)\|^2]\\
=&4\bar L\mathbb{E}\psi_m(\theta^k)+2\mathbb{E}[\mathbb{E}[\|\nabla\ell(\tilde\theta;\xi_m^k)-\nabla\ell(\theta^\star;\xi_m^k)\|^2|\Theta^k]]\\
\le&4\bar L\mathbb{E}\psi_m(\theta^k)+4\bar L\mathbb{E}\psi_m(\tilde\theta).
\end{aligned}$$

By nonnegativity of ψ_m, we have

$$\begin{aligned}
\mathbb{E}[\|g_m^k\|^2]&\le 4\bar L\sum_{m\in\mathcal{M}}\mathbb{E}\psi_m(\theta^k)+4\bar L\sum_{m\in\mathcal{M}}\mathbb{E}\psi_m(\tilde\theta)\\
&=4M\bar L(\mathbb{E}\mathcal{L}(\theta^k)-\mathcal{L}(\theta^\star))+4M\bar L(\mathbb{E}\mathcal{L}(\tilde\theta)-\mathcal{L}(\theta^\star)).
\end{aligned}\tag{16}$$

Similarly, we can prove

$$\mathbb{E}[\|g_m^{k-\tau_m^k}\|^2]\le 4M\bar L(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k})-\mathcal{L}(\theta^\star))+4M\bar L(\mathbb{E}\mathcal{L}(\tilde\theta)-\mathcal{L}(\theta^\star)).\tag{17}$$

Therefore, it follows that

$$\mathbb{E}[\|\tilde{\delta}_m^k - \tilde{\delta}_m^{k-\tau_m^k}\|^2] \leq 8M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star)) + 8M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)) + 16M\bar{L}(\mathbb{E}\mathcal{L}(\tilde{\theta}) - \mathcal{L}(\theta^\star)).$$

C.3. Derivations of (15)

The LHS of (5) can be written as

$$\begin{aligned}\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) =& \big(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\mathcal{L}_m(\theta^{k-\tau_m^k})\big) - \nabla\mathcal{L}_m(\theta^{k-\tau_m^k}) \\ =& \big(\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\psi_m(\theta^{k-\tau_m^k})\big) - \nabla\psi_m(\theta^{k-\tau_m^k}).\end{aligned}$$

Similar to (16), we can obtain

$$\begin{aligned}&\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) + \nabla\psi_m(\theta^{k-\tau_m^k})\|^2] \\ &\qquad\leq 4M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star)) + 4M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)).\end{aligned}$$

Combined with the fact

$$\begin{aligned}\mathbb{E}[\|\nabla\psi_m(\theta^{k-\tau_m^k})\|^2] &= \mathbb{E}[\|\nabla\mathcal{L}_m(\theta^{k-\tau_m^k}) - \nabla\mathcal{L}_m(\theta^\star)\|^2] \\ &\leq 2\bar{L}\mathbb{E}\psi_m(\theta^{k-\tau_m^k}) \leq 2M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star))\end{aligned}$$

we have

$$\mathbb{E}[\|\nabla\ell(\theta^k;\xi_m^k) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^k)\|^2] \leq 8M\bar{L}(\mathbb{E}\mathcal{L}(\theta^k) - \mathcal{L}(\theta^\star)) + 12M\bar{L}(\mathbb{E}\mathcal{L}(\theta^{k-\tau_m^k}) - \mathcal{L}(\theta^\star)).$$

Appendix D. Proof of Lemma 1

Using the smoothness of $\mathcal{L}(\theta)$ in Assumption 1, we have

$$\begin{aligned}\mathcal{L}(\theta^{k+1}) \leq& \mathcal{L}(\theta^k) + \Big\langle \nabla\mathcal{L}(\theta^k), \theta^{k+1} - \theta^k \Big\rangle + \frac{L}{2}\big\|\theta^{k+1} - \theta^k\big\|^2 \\ =& \mathcal{L}(\theta^k) - \alpha_k \Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k+1})^{-\frac{1}{2}} h^{k+1} \Big\rangle + \frac{L}{2}\big\|\theta^{k+1} - \theta^k\big\|^2. \end{aligned} \tag{18}$$

We can further decompose the inner product as

$$\begin{aligned}&-\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k+1})^{-\frac{1}{2}} h^{k+1} \Big\rangle \\ =& -(1-\beta_1)\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^k)^{-\frac{1}{2}} \boldsymbol{\nabla}^k \Big\rangle \underbrace{-\beta_1 \Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^k)^{-\frac{1}{2}} h^k \Big\rangle}_{I_1^k} \\ &\underbrace{-\Big\langle \nabla\mathcal{L}(\theta^k), \Big((\epsilon I + \hat{V}^{k+1})^{-\frac{1}{2}} - (\epsilon I + \hat{V}^k)^{-\frac{1}{2}}\Big) h^{k+1} \Big\rangle}_{I_2^k}\end{aligned} \tag{19}$$

where we again decompose the first inner product as

$$-(1-\beta_1)\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^k)^{-\frac{1}{2}}\boldsymbol{\nabla}^k\Big\rangle = \underbrace{-(1-\beta_1)\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\boldsymbol{\nabla}^k\Big\rangle}_{I_3^k}$$
$$\underbrace{-(1-\beta_1)\Big\langle \nabla\mathcal{L}(\theta^k), \Big((\epsilon I+\hat{V}^k)^{-\frac{1}{2}}-(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big)\boldsymbol{\nabla}^k\Big\rangle}_{I_4^k}. \quad (20)$$

Next, we bound the terms $I_1^k, I_2^k, I_3^k, I_4^k$ separately.

Taking expectation on I_1^k conditioned on Θ^k, we have

$$\begin{aligned}\mathbb{E}[I_1^k \mid \Theta^k] &= -\mathbb{E}\Big[\beta_1\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\Big\rangle \mid \Theta^k\Big]\\ &= -\beta_1\Big\langle \nabla\mathcal{L}(\theta^{k-1}), (\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\Big\rangle - \beta_1\Big\langle \nabla\mathcal{L}(\theta^k)-\nabla\mathcal{L}(\theta^{k-1}), (\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\Big\rangle\\ &\overset{(a)}{\le} -\beta_1\Big\langle \nabla\mathcal{L}(\theta^{k-1}), (\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\Big\rangle + \alpha_{k-1}^{-1}\beta_1 L\big\|\theta^k-\theta^{k-1}\big\|^2\\ &\overset{(b)}{\le} \beta_1\Big(I_1^{k-1}+I_2^{k-1}+I_3^{k-1}+I_4^{k-1}\Big)+\alpha_{k-1}^{-1}\beta_1 L\big\|\theta^k-\theta^{k-1}\big\|^2 \quad (21)\end{aligned}$$

where follows from the L-smoothness of $\mathcal{L}(\boldsymbol{\theta})$ implied by Assumption 1; and (b) uses again the decomposition (19) and (20).

Taking expectation on I_2^k over all the randomness, we have

$$\begin{aligned}\mathbb{E}[I_2^k] =&\mathbb{E}\Big[-\Big\langle \nabla\mathcal{L}(\theta^k), \Big((\epsilon I+\hat{V}^{k+1})^{-\frac{1}{2}}-(\epsilon I+\hat{V}^k)^{-\frac{1}{2}}\Big)h^{k+1}\Big\rangle\Big]\\ =&\mathbb{E}\Big[\sum_{i=1}^p \nabla_i\mathcal{L}(\theta^k)h_i^{k+1}\Big((\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big]\\ \overset{(d)}{\le}&\mathbb{E}\Big[\|\nabla\mathcal{L}(\theta^k)\|\|h^{k+1}\|\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big]\\ \overset{(e)}{\le}&\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big] \quad (22)\end{aligned}$$

where (d) follows from the Cauchy-Schwarz inequality and (e) is due to Assumption 2.

Regarding I_3^k, we will bound separately in Lemma 2.

Taking expectation on I_4^k over all the randomness, we have

$$\begin{aligned}\mathbb{E}[I_4^k] =&\mathbb{E}\Big[-(1-\beta_1)\Big\langle \nabla\mathcal{L}(\theta^k), \Big((\epsilon I+\hat{V}^k)^{-\frac{1}{2}}-(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big)\boldsymbol{\nabla}^k\Big\rangle\Big]\\ =&-(1-\beta_1)\mathbb{E}\Big[\sum_{i=1}^p \nabla_i\mathcal{L}(\theta^k)\boldsymbol{\nabla}_i^k\Big((\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}\Big)\Big]\\ \le&(1-\beta_1)\mathbb{E}\Big[\|\nabla\mathcal{L}(\theta^k)\|\|\boldsymbol{\nabla}^k\|\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}\Big)\Big]\\ \le&(1-\beta_1)\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}\Big)\Big]. \quad (23)\end{aligned}$$

Taking expectation on (18) over all the randomness, and plugging (21), (22), and (23), we have

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\theta^{k+1})]-\mathbb{E}[\mathcal{L}(\theta^k)] \le & -\alpha_k\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),(\epsilon I+\hat{V}^{k+1})^{-\frac{1}{2}}h^{k+1}\right\rangle\right]+\frac{L}{2}\mathbb{E}\left[\left\|\theta^{k+1}-\theta^k\right\|^2\right]\\ =&\alpha_k\mathbb{E}\left[I_1^k+I_2^k+I_3^k+I_4^k\right]+\frac{L}{2}\mathbb{E}\left[\left\|\theta^{k+1}-\theta^k\right\|^2\right]\\ \le & -\alpha_k(1-\beta_1)\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\boldsymbol{\nabla}^k\right\rangle\right]\\ &-\alpha_k\beta_1\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^{k-1}),(\epsilon I+\hat{V}^{k})^{-\frac{1}{2}}h^k\right\rangle\right]\\ &+\alpha_k\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\left((\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\right)\Big]\\ &+\alpha_k(1-\beta_1)\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\left((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k})^{-\frac{1}{2}}\right)\Big]\\ &+\left(\frac{L}{2}+\alpha_k\alpha_{k-1}^{-1}\beta_1 L\right)\mathbb{E}\left[\|\theta^{k+1}-\theta^k\|^2\right]. \end{aligned}\tag{24}$$

Since $(\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}\le(\epsilon+\hat{v}_i^{k-1})^{-\frac{1}{2}}$, we have

$$\begin{aligned}&\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^k)^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)+(1-\beta_1)\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k})^{-\frac{1}{2}}\Big)\Big]\\ \le&(2-\beta_1)\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\left((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\right)\Big].\end{aligned}\tag{25}$$

Plugging (25) into (24) leads to the statement of Lemma 1.

Appendix E. Proof of Lemma 2

We first analyze the inner produce under CADA2 and then CADA1.

First recall that $\bar{\boldsymbol{\nabla}}^k=\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^k;\xi_m^k)$. Using the law of total probability implies that

$$\begin{aligned}\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\bar{\boldsymbol{\nabla}}^k\right\rangle\right]&=\mathbb{E}\left[\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\bar{\boldsymbol{\nabla}}^k\right\rangle\mid\Theta^k\right]\right]\\ &=\mathbb{E}\left[\left\langle\nabla\mathcal{L}(\theta^k),(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\mathbb{E}\left[\bar{\boldsymbol{\nabla}}^k\mid\Theta^k\right]\right\rangle\right]\\ &=\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right].\end{aligned}\tag{26}$$

Taking expectation on $\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\boldsymbol{\nabla}^k \right\rangle$ over all randomness, we have

$$\begin{aligned}
&-\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\boldsymbol{\nabla}^k \right\rangle\right] \\
=&-\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\bar{\boldsymbol{\nabla}}^k \right\rangle\right] \\
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\frac{1}{M}\sum_{m\in\mathcal{M}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\theta^k;\xi_m^k)\right)\Big\rangle\Big] \\
\overset{(a)}{=}&-\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right] \\
&-\frac{1}{M}\sum_{m\in\mathcal{M}}\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\theta^k;\xi_m^k)\right)\Big\rangle\Big] \qquad (27)
\end{aligned}$$

where (a) uses (26).

Decomposing the inner product, for the CADA2 rule (5), we have

$$\begin{aligned}
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\theta^k;\xi_m^k)\right)\Big\rangle\Big] \\
=&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k}) - \nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right)\Big\rangle\Big] \\
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) - \nabla\ell(\theta^k;\xi_m^k)\right)\Big\rangle\Big] \\
\overset{(b)}{\le}&\frac{L\epsilon^{-\frac{1}{2}}}{12\alpha_k}\sum_{d=1}^{D}\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right] + 6DL\alpha_k\epsilon^{-\frac{1}{2}}\sigma_m^2 \\
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) - \nabla\ell(\theta^k;\xi_m^k)\right)\Big\rangle\Big] \qquad (28)
\end{aligned}$$

where (b) follows from Lemma 6.

Using the Young's inequality, we can bound the last inner product in (28) as

$$\begin{aligned}
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) - \nabla\ell(\theta^k;\xi_m^k)\right)\Big\rangle\Big] \\
\le&\frac{1}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right] + \frac{1}{2}\mathbb{E}\left[\left\|(\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\right\|\left\|\left(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) - \nabla\ell(\theta^k;\xi_m^k)\right)\right\|^2\right] \\
\overset{(g)}{\le}&\frac{1}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right] + \frac{1}{2}\mathbb{E}\left[\left\|(\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\right\|\left\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^k) - \nabla\ell(\theta^k;\xi_m^k)\right\|^2\right] \\
\overset{(h)}{\le}&\frac{1}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right] + \frac{c}{2}\mathbb{E}\left[\left\|(\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}\right\|\sum_{d=1}^{D}\left\|\theta^{k+1-d}-\theta^{k-d}\right\|^2\right] \\
\overset{(i)}{\le}&\frac{1}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right] + \frac{c\epsilon^{-\frac{1}{2}}}{2}\sum_{d=1}^{D}\mathbb{E}\left[\left\|\theta^{k+1-d}-\theta^{k-d}\right\|^2\right] \qquad (29)
\end{aligned}$$

where (g) follows from the Cauchy-Schwarz inequality, and (h) uses the adaptive communication condition (5) in CADA2, and (i) follows since $\hat{V}^{k-D}$ is entry-wise nonnegative.

Similarly for CADA1's condition (4), we have

$$\begin{aligned}
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big(\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})-\nabla\ell(\theta^k;\xi_m^k)\Big)\Big\rangle\Big]\\
=&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big(\nabla\ell(\tilde{\theta};\xi_m^{k-\tau_m^k})-\nabla\ell(\tilde{\theta};\xi_m^k)\Big)\Big\rangle\Big]\\
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big(\tilde{\delta}_m^{k-\tau_m^k}-\tilde{\delta}_m^k\Big)\Big\rangle\Big]\\
\overset{(j)}{\leq}&\frac{L\epsilon^{-\frac{1}{2}}}{12\alpha_k}\sum_{d=1}^{D}\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]+6DL\alpha_k\epsilon^{-\frac{1}{2}}\sigma_m^2\\
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big(\tilde{\delta}_m^{k-\tau_m^k}-\tilde{\delta}_m^k\Big)\Big\rangle\Big]
\end{aligned}\tag{30}$$

where (j) follows from Lemma 6 since $\tilde{\theta}$ is a snapshot among $\{\theta^k,\cdots,\theta^{k-D}\}$.

And the last product in (30) is bounded by

$$\begin{aligned}
&-\mathbb{E}\Big[\Big\langle \nabla\mathcal{L}(\theta^k), (\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big(\tilde{\delta}_m^{k-\tau_m^k}-\tilde{\delta}_m^k\Big)\Big\rangle\Big]\\
\leq&\frac{1}{2}\mathbb{E}\Big[\Big\|\nabla\mathcal{L}(\theta^k)\Big\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\Big]+\frac{c}{2}\mathbb{E}\Big[\Big\|(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}\Big\|\sum_{d=1}^{D}\Big\|\theta^{k+1-d}-\theta^{k-d}\Big\|^2\Big]\\
\overset{(i)}{\leq}&\frac{1}{2}\mathbb{E}\Big[\Big\|\nabla\mathcal{L}(\theta^k)\Big\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\Big]+\frac{c\epsilon^{-\frac{1}{2}}}{2}\sum_{d=1}^{D}\mathbb{E}\Big[\Big\|\theta^{k+1-d}-\theta^{k-d}\Big\|^2\Big].
\end{aligned}\tag{31}$$

Combining (27)-(31) leads to the desired statement for CADA1 and CADA2.

Appendix F. Proof of Lemma 3

For notational brevity, we re-write the Lyapunov function (8) as

$$\begin{aligned}
\mathcal{V}^k:=\mathcal{L}(\theta^k)-\mathcal{L}(\theta^\star)-c_k\Big\langle\nabla\mathcal{L}(\theta^{k-1}),(\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\Big\rangle\\
+b_k\sum_{d=0}^{D}\sum_{i=1}^{p}(\epsilon+\hat{v}_i^{k-d})^{-\frac{1}{2}}+\sum_{d=1}^{D}\rho_d\|\theta^{k+1-d}-\theta^{k-d}\|^2
\end{aligned}\tag{32}$$

where $\{c_k\}$ are some positive constants.

Therefore, taking expectation on the difference of $\mathcal{V}^k$ and $\mathcal{V}^{k+1}$ in (32), we have (with $\rho_{D+1} = 0$)

$$\begin{aligned}
\mathbb{E}[\mathcal{V}^{k+1}] - \mathbb{E}[\mathcal{V}^k] =& \mathbb{E}[\mathcal{L}(\theta^{k+1})] - \mathbb{E}[\mathcal{L}(\theta^k)] - c_{k+1}\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k+1})^{-\frac{1}{2}} h^{k+1}\right\rangle\right] \\
&+ c_k \mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^{k-1}), (\epsilon I + \hat{V}^{k})^{-\frac{1}{2}} h^{k}\right\rangle\right] \\
&+ b_{k+1}\sum_{d=0}^{D}\sum_{i=1}^{p}(\epsilon + \hat{v}_i^{k+1-d})^{-\frac{1}{2}} - b_k \sum_{d=0}^{D}\sum_{i=1}^{p}(\epsilon + \hat{v}_i^{k-d})^{-\frac{1}{2}} \\
&+ \rho_1 \mathbb{E}\left[\|\theta^{k+1} - \theta^k\|^2\right] + \sum_{d=1}^{D}(\rho_{d+1} - \rho_d)\mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right] \\
\overset{(a)}{\leq}& (\alpha_k + c_{k+1})\mathbb{E}\left[I_1^k + I_2^k + I_3^k + I_4^k\right] - c_k \mathbb{E}\left[I_1^{k-1} + I_2^{k-1} + I_3^{k-1} + I_4^{k-1}\right] \\
&+ b_{k+1}\sum_{i=1}^{p}\mathbb{E}\left[(\epsilon + \hat{v}_i^{k+1})^{-\frac{1}{2}}\right] - b_k \sum_{i=1}^{p}\mathbb{E}\left[(\epsilon + \hat{v}_i^{k-D})^{-\frac{1}{2}}\right] \\
&+ \sum_{d=1}^{D}(b_{k+1} - b_k)\sum_{i=1}^{p}\mathbb{E}\left[(\epsilon + \hat{v}_i^{k+1-d})^{-\frac{1}{2}}\right] + \left(\frac{L}{2} + \rho_1\right)\mathbb{E}\left[\|\theta^{k+1} - \theta^k\|^2\right] \\
&+ \sum_{d=1}^{D}(\rho_{d+1} - \rho_d)\mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right] \qquad (33)
\end{aligned}$$

where (a) uses the smoothness in Assumption 1 and the definition of $I_1^k, I_2^k, I_3^k, I_4^k$ in (19) and (20).

Note that we can bound $(\alpha_k + c_{k+1})\mathbb{E}\left[I_1^k + I_2^k + I_3^k + I_4^k\right]$ the same as (19) in the proof of Lemma 1 In addition, Lemma 2 implies that

$$\begin{aligned}
\mathbb{E}[I_3^k] \leq& -\frac{1-\beta_1}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}}}\right] \\
&+ (1-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12\alpha_k} + \frac{c}{2}\right)\sum_{d=1}^{D}\mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right] + (1-\beta_1)\frac{6DL\alpha_k\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2. \qquad (34)
\end{aligned}$$

Therefore, plugging Lemma 1 with α_k replaced by α_k+c_{k+1} into (33), together with (34), leads to

$$\begin{aligned}\mathbb{E}[\mathcal{V}^{k+1}]-\mathbb{E}[\mathcal{V}^k] \le & -(\alpha_k+c_{k+1})\left(\frac{1-\beta_1}{2}\right)\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right]\\
&+(\alpha_k+c_{k+1})(1-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12\alpha_k}+\frac{c}{2}\right)\sum_{d=1}^{D}\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]\\
&+(\alpha_k+c_{k+1})(1-\beta_1)\frac{6DL\alpha_k\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2\\
&+((\alpha_k+c_{k+1})\beta_1-c_k)\mathbb{E}\left[I_1^{k-1}+I_2^{k-1}+I_3^{k-1}+I_4^{k-1}\right]\\
&+(\alpha_k+c_{k+1})(2-\beta_1)\sigma^2\mathbb{E}\Big[\sum_{i=1}^{p}\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big]\\
&+b_{k+1}\sum_{i=1}^{p}\mathbb{E}\Big[(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big]-b_k\sum_{i=1}^{p}\mathbb{E}\Big[(\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}\Big]\\
&+\sum_{d=1}^{D}(b_{k+1}-b_k)\sum_{i=1}^{p}\mathbb{E}\Big[(\epsilon+\hat{v}_i^{k+1-d})^{-\frac{1}{2}}\Big]\\
&+\sum_{d=1}^{D}(\rho_{d+1}-\rho_d)\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]\\
&+\left(\frac{L}{2}+\rho_1+(\alpha_k+c_{k+1})\alpha_{k-1}^{-1}\beta_1 L\right)\mathbb{E}\left[\|\theta^{k+1}-\theta^k\|^2\right]. \qquad (35)\end{aligned}$$

Select $\alpha_k\le\alpha_{k-1}$ and $c_k:=\sum_{j=k}^{\infty}\alpha_j\beta_1^{j-k+1}\le(1-\beta_1)^{-1}\alpha_k$ so that $(\alpha_k+c_{k+1})\beta_1=c_k$ and

$$\begin{aligned}(\alpha_k+c_{k+1})(1-\beta_1)&\le(\alpha_k+(1-\beta_1)^{-1}\alpha_{k+1})(1-\beta_1)\\
&\le\alpha_k(1+(1-\beta_1)^{-1})(1-\beta_1)=\alpha_k(2-\beta_1).\end{aligned}$$

In addition, select b_k to ensure that $b_{k+1}\le b_k$. Then it follows from (35) that

$$\begin{aligned}\mathbb{E}[\mathcal{V}^{k+1}]-\mathbb{E}[\mathcal{V}^k]\le & -\frac{\alpha_k(1-\beta_1)}{2}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2_{(\epsilon I+\hat{V}^{k-D})^{-\frac{1}{2}}}\right]+(2-\beta_1)\alpha_k^2\frac{6DL\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2\\
&+(2-\beta_1)\alpha_k\epsilon^{-\frac{1}{2}}\left(\frac{L}{12\alpha_k}+\frac{c}{2}\right)\sum_{d=1}^{D}\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]\\
&+\left(\frac{(2-\beta_1)^2}{(1-\beta_1)}\alpha_k\sigma^2-b_k\right)\mathbb{E}\Big[\sum_{i=1}^{p}\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big]\\
&+\left(\frac{L}{2}+\rho_1+(1-\beta_1)^{-1}L\right)\mathbb{E}\left[\|\theta^{k+1}-\theta^k\|^2\right]\\
&+\sum_{d=1}^{D}(\rho_{d+1}-\rho_d)\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right] \qquad (36)\end{aligned}$$

where we have also used the fact that $-(\alpha_k + c_{k+1})\left(\frac{1-\beta_1}{2}\right) \leq -\frac{\alpha_k(1-\beta_1)}{2}$ since $c_{k+1} \geq 0$.

If we choose $\alpha_k \leq \frac{1}{L}$ for $k = 1, 2 \ldots, K$, then it follows from (36) that

$$\begin{aligned}
&\mathbb{E}[\mathcal{V}^{k+1}] - \mathbb{E}[\mathcal{V}^k] \\
\leq& -\frac{\alpha_k(1-\beta_1)}{2}\left(\epsilon + \frac{\sigma^2}{1-\beta_2}\right)^{-\frac{1}{2}} \mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2\right] + (2-\beta_1)\frac{6\alpha_k^2 DL\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2 \\
&+ \underbrace{\left(\frac{(2-\beta_1)^2}{(1-\beta_1)}\alpha_k\sigma^2 - b_k\right)}_{A^k}\mathbb{E}\Big[\sum_{i=1}^{p}\left((\epsilon + \hat{v}_i^{k-D})^{-\frac{1}{2}} - (\epsilon + \hat{v}_i^{k+1})^{-\frac{1}{2}}\right)\Big] \\
&+ \left(\frac{L}{2} + \rho_1 + (1-\beta_1)^{-1}L\right)\mathbb{E}\left[\|\theta^{k+1} - \theta^k\|^2\right] \\
&+ \sum_{d=1}^{D}\underbrace{\left((2-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12} + \frac{c\alpha_k}{2}\right) + \rho_{d+1} - \rho_d\right)}_{B_d^k}\mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right]. \qquad (37)
\end{aligned}$$

To ensure $A^k \leq 0$ and $B_d^k \leq 0$, it is sufficient to choose $\{b_k\}$ and $\{\rho_d\}$ satisfying (with $\rho_{D+1} = 0$)

$$\begin{aligned}
&\frac{(2-\beta_1)^2}{(1-\beta_1)}\alpha_k\sigma^2 - b_k \leq 0, \quad k = 1, \cdots, K \\
&(2-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12} + \frac{c\alpha_k}{2}\right) + \rho_{d+1} - \rho_d \leq 0, \quad d = 1, \cdots, D.
\end{aligned}$$

Solve this system of linear equations and get

$$b_k = \frac{(2-\beta_1)^2}{(1-\beta_1)L}\sigma^2, \quad k = 1, \cdots, K \qquad (38)$$

$$\rho_d = (2-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12} + \frac{c}{2L}\right)(D-d+1), \quad d = 1, \cdots, D \qquad (39)$$

plugging which into (37) leads to the conclusion of Lemma 3.

Appendix G. Proof of Theorem 4

From the definition of $\mathcal{V}^k$, we have for any k, that

$$\begin{aligned}
\mathbb{E}[\mathcal{V}^k] &\geq \mathcal{L}(\theta^k) - \mathcal{L}(\theta^*) - c_k\left\langle\nabla\mathcal{L}(\theta^{k-1}), (\epsilon I + \hat{V}^k)^{-\frac{1}{2}}h^k\right\rangle + \sum_{d=1}^{D}\rho_d\|\theta^{k+1-d} - \theta^{k-d}\|^2 \\
&\geq -|c_k|\left\|\nabla\mathcal{L}(\theta^{k-1})\right\|\left\|(\epsilon I + \hat{V}^k)^{-\frac{1}{2}}h^k\right\| \\
&\geq -(1-\beta_1)^{-1}\alpha_k\sigma^2\epsilon^{-\frac{1}{2}} \qquad (40)
\end{aligned}$$

where we use Assumption 2 and Lemma 7.

By taking summation on (37) over $k = 0, \cdots, K-1$, it follows from that

$$\begin{aligned}
&\frac{\alpha(1-\beta_1)}{2}\left(\epsilon+\frac{\sigma^2}{1-\beta_2}\right)^{-\frac{1}{2}}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2\right]\\
\leq&\frac{\mathbb{E}[\mathcal{V}^1]-\mathbb{E}[\mathcal{V}^{K+1}]}{K}+(2-\beta_1)\frac{6\alpha^2DL\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2+\frac{(2-\beta_1)^2}{(1-\beta_1)}\sigma^2pD\epsilon^{-\frac{1}{2}}\frac{\alpha}{K}\\
&+\left(\frac{L}{2}+\rho_1+(1-\beta_1)^{-1}L\right)\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\theta^{k+1}-\theta^k\|^2\right]\\
\overset{(a)}{\leq}&\frac{\mathbb{E}[\mathcal{V}^1]}{K}+(2-\beta_1)\frac{6\alpha^2DL\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2+(1-\beta_1)^{-1}\sigma^2\epsilon^{-\frac{1}{2}}\frac{\alpha}{K}+\frac{(2-\beta_1)^2}{(1-\beta_1)}\sigma^2pD\epsilon^{-\frac{1}{2}}\frac{\alpha}{K}\\
&+\left(\frac{L}{2}+\rho_1+(1-\beta_1)^{-1}L\right)p(1-\beta_2)^{-1}(1-\beta_3)^{-1}\alpha^2 \qquad (41)
\end{aligned}$$

where (a) follows from (40) and Lemma 8.

Specifically, if we choose a constant stepsize $\alpha := \frac{\eta}{\sqrt{K}}$, where $\eta > 0$ is a constant, and define

$$\tilde{C}_1 := (2-\beta_1)6DL\epsilon^{-\frac{1}{2}} \qquad (42)$$

and

$$\tilde{C}_2 := (1-\beta_1)^{-1}\epsilon^{-\frac{1}{2}}+\frac{(2-\beta_1)^2}{(1-\beta_1)}D\epsilon^{-\frac{1}{2}} \qquad (43)$$

and

$$\tilde{C}_3 := \left(\frac{L}{2}+\rho_1+(1-\beta_1)^{-1}L\right)(1-\beta_2)^{-1}(1-\beta_3)^{-1} \qquad (44)$$

and

$$\tilde{C}_4 := \frac{1}{2}(1-\beta_1)\left(\epsilon+\frac{\sigma^2}{1-\beta_2}\right)^{-\frac{1}{2}} \qquad (45)$$

we can obtain from (41) that

$$\begin{aligned}
\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla\mathcal{L}(\theta^k)\|^2\right]\leq&\frac{\frac{\mathcal{L}(\theta^0)-\mathcal{L}(\theta^*)}{K}+\frac{\tilde{C}_1}{M}\sum_{m\in\mathcal{M}}\sigma_m^2\alpha^2+\tilde{C}_2p\sigma^2\frac{\alpha}{K}+\tilde{C}_3p\alpha^2}{\alpha\tilde{C}_4}\\
\leq&\frac{\mathcal{L}(\theta^0)-\mathcal{L}(\theta^*)}{K\alpha\tilde{C}_4}+\frac{\tilde{C}_1\alpha}{\tilde{C}_4M}\sum_{m\in\mathcal{M}}\sigma_m^2+\tilde{C}_2p\frac{\sigma^2}{K\tilde{C}_4}+\frac{\tilde{C}_3p\alpha}{\tilde{C}_4}\\
=&\frac{(\mathcal{L}(\theta^0)-\mathcal{L}(\theta^*))C_4}{\sqrt{K}\eta}+\frac{C_1\eta}{\sqrt{K}M}\sum_{m\in\mathcal{M}}\sigma_m^2+\frac{C_2p\sigma^2}{K}+\frac{C_3p\eta}{\sqrt{K}}
\end{aligned}$$

where we define $C_1 := \tilde{C}_1/\tilde{C}_4$, $C_2 := \tilde{C}_2/\tilde{C}_4$, $C_3 := \tilde{C}_3/\tilde{C}_4$, and $C_4 := 1/\tilde{C}_4$.

Appendix H. Convergence under Polyak-Łojasiewicz condition

Next we present the convergence results under a slightly stronger assumption on the loss $\mathcal{L}(\theta)$.

Assumption 3 *The loss function $\mathcal{L}(\theta)$ satisfies the Polyak-Łojasiewicz (PL) condition with the constant $\mu > 0$, that is $\mathcal{L}(\theta) - \mathcal{L}(\theta^*) \leq \frac{1}{2\mu} \|\mathcal{L}(\theta)\|^2$.*

The PL condition is weaker than the strongly convexity, which does not even require convexity [11]. And it is satisfied by a wider range of problems such as least squares for underdetermined linear systems and logistic regression, and also certain types of neural networks.

We next establish the convergence of CADA under this condition.

Theorem 5 (PL-condition) *Under Assumptions 1-3, if we choose the stepsize as $\alpha_k = \frac{2}{\mu(k+K_0)}$ for a given constant K_0, then θ^K generated by Algorithm 1 satisfies*

$$\mathbb{E}\left[\mathcal{L}(\theta^K)\right] - \mathcal{L}(\theta^\star) = \mathcal{O}\left(\frac{1}{K}\right). \tag{46}$$

Theorem 5 implies that under the PL-condition of the loss function, the CADA algorithm can achieve the global convergence in terms of the loss function, with a fast rate $\mathcal{O}(1/K)$.

By the PL-condition of $\mathcal{L}(\theta)$, we have

$$\begin{aligned}
&-\frac{\alpha_k(1-\beta_1)}{2}\left(\epsilon+\frac{\sigma^2}{1-\beta_2}\right)^{-\frac{1}{2}}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta^k)\right\|^2\right]\\
\leq&-\alpha_k\mu(1-\beta_1)\left(\epsilon+\frac{\sigma^2}{1-\beta_2}\right)^{-\frac{1}{2}}\mathbb{E}\left[\mathcal{L}(\theta^k)-\mathcal{L}(\theta^\star)\right]\\
\overset{(a)}{\leq}&-2\alpha_k\mu\tilde{C}_4\Big(\mathbb{E}[\mathcal{V}^k]+c_k\left\langle\nabla\mathcal{L}(\theta^{k-1}),(\epsilon I+\hat{V}^k)^{-\frac{1}{2}}h^k\right\rangle-b_k\sum_{d=0}^{D}\sum_{i=1}^{p}(\epsilon+\hat{v}_i^{k-d})^{-\frac{1}{2}}-\sum_{d=1}^{D}\rho_d\|\theta^{k+1-d}-\theta^{k-d}\|^2\Big)\\
\overset{(b)}{\leq}&-2\alpha_k\mu\tilde{C}_4\mathbb{E}[\mathcal{V}^k]+2\alpha_k^2\mu\tilde{C}_4(1-\beta_1)^{-1}\sigma^2\epsilon^{-\frac{1}{2}}+2\alpha_k\mu\tilde{C}_4b_k\sum_{d=0}^{D}\sum_{i=1}^{p}\mathbb{E}\left[(\epsilon+\hat{v}_i^{k-d})^{-\frac{1}{2}}\right]\\
&+2\alpha_k\mu\tilde{C}_4\sum_{d=1}^{D}\rho_d\mathbb{E}[\|\theta^{k+1-d}-\theta^{k-d}\|^2]
\end{aligned}\tag{47}$$

where (a) uses the definition of $\tilde{C}_4$ in (45), and (b) uses Assumption 2 and Lemma 7.

Plugging (47) into (36), we have

$$\begin{aligned}
&\mathbb{E}[\mathcal{V}^{k+1}]-\mathbb{E}[\mathcal{V}^k]\\
\le&-2\alpha_k\mu\tilde{C}_4\mathbb{E}[\mathcal{V}^k]+(2-\beta_1)\frac{6\alpha_k^2DL\epsilon^{-\frac{1}{2}}}{M}\sum_{m\in\mathcal{M}}\sigma_m^2\\
&+\frac{(2-\beta_1)^2}{(1-\beta_1)}\alpha_k\sigma^2\mathbb{E}\Big[\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big]\\
&+b_{k+1}\sum_{i=1}^p\mathbb{E}\Big[(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big]-(b_k-2\alpha_k\mu\tilde{C}_4b_k)\sum_{i=1}^p\mathbb{E}\Big[(\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}\Big]\\
&+\sum_{d=1}^D(b_{k+1}-b_k+2\alpha_k\mu\tilde{C}_4b_k)\sum_{i=1}^p\mathbb{E}\Big[(\epsilon+\hat{v}_i^{k+1-d})^{-\frac{1}{2}}\Big]\\
&+\left(\frac{L}{2}+\rho_1+(1-\beta_1)^{-1}L\right)p(1-\beta_2)^{-1}(1-\beta_3)^{-1}\alpha_k^2+2\alpha_k^2\mu\tilde{C}_4(1-\beta_1)^{-1}\sigma^2\epsilon^{-\frac{1}{2}}\\
&+\sum_{d=1}^D\left((2-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12}+\frac{c\alpha_k}{2}\right)+\rho_{d+1}-\rho_d+2\alpha_k\mu\tilde{C}_4\rho_d\right)\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right]. \qquad (48)
\end{aligned}$$

If we choose b_k to ensure that $b_{k+1}\le(1-2\alpha_k\mu\tilde{C}_4)b_k$, then we can obtain from (48) that

$$\begin{aligned}
&\mathbb{E}[\mathcal{V}^{k+1}]-\mathbb{E}[\mathcal{V}^k] \qquad (49)\\
\le&-2\alpha_k\mu\tilde{C}_4\mathbb{E}[\mathcal{V}^k]+\frac{\tilde{C}_1}{M}\sum_{m\in\mathcal{M}}\sigma_m^2\alpha_k^2+\tilde{C}_3p\alpha_k^2+2\mu\tilde{C}_4(1-\beta_1)^{-1}\sigma^2\epsilon^{-\frac{1}{2}}\alpha_k^2\\
&+\left(\frac{(2-\beta_1)^2}{(1-\beta_1)}\alpha_k\sigma^2-(1-2\alpha_k\mu\tilde{C}_4)b_k\right)\mathbb{E}\Big[\sum_{i=1}^p\Big((\epsilon+\hat{v}_i^{k-D})^{-\frac{1}{2}}-(\epsilon+\hat{v}_i^{k+1})^{-\frac{1}{2}}\Big)\Big]\\
&+\sum_{d=1}^D\left((2-\beta_1)\epsilon^{-\frac{1}{2}}\left(\frac{L}{12}+\frac{c\alpha_k}{2}\right)+\rho_{d+1}-\rho_d+2\alpha_k\mu\tilde{C}_4\rho_d\right)\mathbb{E}\left[\|\theta^{k+1-d}-\theta^{k-d}\|^2\right].
\end{aligned}$$

If $\alpha_k\le\frac{1}{L}$, we choose parameters $\{b_k,\rho_d\}$ to guarantee that

$$\frac{(2-\beta_1)^2}{(1-\beta_1)L}\sigma^2-\Big(1-\frac{2\mu\tilde{C}_4}{L}\Big)b_k\le 0,\quad\forall k \qquad (50)$$

$$(2-\beta_1)\left(\frac{L}{12}+\frac{c}{2L}\right)\epsilon^{-\frac{1}{2}}+\rho_{d+1}-\Big(1-\frac{2\mu\tilde{C}_4}{L}\Big)\rho_d\le 0,\quad d=1,\cdots,D \qquad (51)$$

and choose β_1,β_2,ϵ to ensure that $1-\frac{2\mu\tilde{C}_4}{L}\ge 0$.

Then we have

$$\mathbb{E}[\mathcal{V}^{k+1}] \le \left(1-2\alpha_k\mu\tilde{C}_4\right)\mathbb{E}[\mathcal{V}^k] + \underbrace{\left(\frac{\tilde{C}_1}{M}\sum_{m\in\mathcal{M}}\sigma_m^2 + \tilde{C}_3 p + 2\mu\tilde{C}_4(1-\beta_1)^{-1}\sigma^2\epsilon^{-\frac{1}{2}}\right)}_{\tilde{C}_5}\alpha_k^2$$

$$\le \prod_{j=0}^{k}(1-2\alpha_j\mu\tilde{C}_4)\mathbb{E}[\mathcal{V}^0] + \sum_{j=0}^{k}\alpha_j^2\prod_{i=j+1}^{k}(1-2\alpha_i\mu\tilde{C}_4)\tilde{C}_5. \tag{52}$$

If we choose $\alpha_k = \frac{1}{\mu(k+K_0)\tilde{C}_4} \le \frac{1}{L}$, where K_0 is a sufficiently large constant to ensure that α_k satisfies the aforementioned conditions, then we have

$$\begin{aligned}
\mathbb{E}[\mathcal{V}^K] \le& \mathbb{E}[\mathcal{V}^0]\prod_{k=0}^{K-1}(1-2\alpha_k\mu\tilde{C}_4) + \tilde{C}_5\sum_{k=0}^{K-1}\alpha_k^2\prod_{j=k+1}^{K-1}(1-2\alpha_j\mu\tilde{C}_4)\\
\le& \mathbb{E}[\mathcal{V}^0]\prod_{k=0}^{K-1}\frac{k+K_0-2}{k+K_0} + \frac{\tilde{C}_5}{\mu^2\tilde{C}_4^2}\sum_{k=0}^{K-1}\frac{1}{(k+K_0)^2}\prod_{j=k+1}^{K-1}\frac{j+K_0-2}{j+K_0}\\
\le& \frac{(K_0-2)(K_0-1)}{(K+K_0-2)(K+K_0-1)}\mathbb{E}[\mathcal{V}^0] + \frac{\tilde{C}_5}{\mu^2\tilde{C}_4^2}\sum_{k=0}^{K-1}\frac{(k+K_0-1)}{(k+K_0)(K+K_0-2)(K+K_0-2)}\\
\le& \frac{(K_0-1)^2}{(K+K_0-1)^2}\mathbb{E}[\mathcal{V}^0] + \frac{\tilde{C}_5 K}{\mu^2\tilde{C}_4^2(K+K_0-1)^2}\\
=& \frac{(K_0-1)^2}{(K+K_0-1)^2}(\mathcal{L}(\theta^0)-\mathcal{L}(\theta^\star)) + \frac{\tilde{C}_5 K}{\mu^2\tilde{C}_4^2(K+K_0-2)^2}
\end{aligned}$$

from which the proof is complete.

Appendix I. Supporting Lemmas

Define the σ-algebra $\Theta^k = \{\theta^l, 1 \le l \le k\}$. For convenience, we also initialize parameters as $\theta^{-D}, \theta^{-D+1}, \ldots, \theta^{-1} = \theta^0$. Some basic facts used in the proof are reviewed as follows.

Fact 1. Assume that $X_1, X_2, \ldots, X_n \in \mathbb{R}^p$ are independent random variables, and $EX_1 = \cdots = EX_n = 0$. Then

$$\mathbb{E}\left[\left\|\sum_{i=1}^{n}X_i\right\|^2\right] = \sum_{i=1}^{n}\mathbb{E}\left[\|X_i\|^2\right]. \tag{53}$$

Fact 2. (Young's inequality) For any $\theta_1, \theta_2 \in \mathbb{R}^p, \varepsilon > 0$,

$$\left\langle\theta_1, \theta_2\right\rangle \le \frac{\|\theta_1\|^2}{2\varepsilon} + \frac{\varepsilon\|\theta_2\|^2}{2}. \tag{54}$$

As a consequence, we have

$$\|\theta_1+\theta_2\|^2 \le \left(1+\frac{1}{\varepsilon}\right)\|\theta_1\|^2 + (1+\varepsilon)\|\theta_2\|^2. \tag{55}$$

Fact 3. (Cauchy-Schwarz inequality) For any $\theta_1, \theta_2, \ldots, \theta_n \in \mathbb{R}^p$, we have

$$\Big\| \sum_{i=1}^n \theta_i \Big\|^2 \le n \sum_{i=1}^n \|\theta_i\|^2. \tag{56}$$

Lemma 6 *For $k - D \le l \le k - \tau_m^k$, if $\{\theta^k\}$ are the iterates generated by CADA, we have*

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right) \right\rangle\right] \\
\le &\frac{L\epsilon^{-\frac{1}{2}}}{12\alpha_k} \sum_{d=1}^D \mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right] + 6DL\alpha_k\epsilon^{-\frac{1}{2}}\sigma_m^2
\end{aligned} \tag{57}$$

and similarly, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\mathcal{L}_m(\theta^l) - \nabla\ell(\theta^l;\theta^{k-\tau_m^k})\right) \right\rangle\right] \\
\le &\frac{L\epsilon^{-\frac{1}{2}}}{12\alpha_k} \sum_{d=1}^D \mathbb{E}\left[\|\theta^{k+1-d} - \theta^{k-d}\|^2\right] + 3DL\alpha_k\epsilon^{-\frac{1}{2}}\sigma_m^2.
\end{aligned} \tag{58}$$

Proof: We first show the following holds.

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^l), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right) \right\rangle\right] \\
\overset{(a)}{=}&\mathbb{E}\left[\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^l), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right) \right\rangle \middle| \Theta^l\right]\right] \\
\overset{(b)}{=}&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^l), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \mathbb{E}\left[\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k}) \middle| \Theta^l\right] \right\rangle\right] \\
=&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^l), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\mathcal{L}_m(\theta^l) - \nabla\mathcal{L}_m(\theta^l)\right) \right\rangle\right] = 0
\end{aligned} \tag{59}$$

where (a) follows from the law of total probability, and (b) holds because $\hat{V}^{k-D}$ is deterministic conditioned on Θ^l when $k - D \le l$.

We first prove (57) by decomposing it as

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right) \right\rangle\right] \\
\overset{(c)}{=}&\mathbb{E}\left[\left\langle \nabla\mathcal{L}(\theta^k) - \nabla\mathcal{L}(\theta^l), (\epsilon I + \hat{V}^{k-D})^{-\frac{1}{2}} \left(\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right) \right\rangle\right] \\
\overset{(d)}{\le}&L\mathbb{E}\left[\left\|(\epsilon I + \hat{V}^{k-D})^{-\frac{1}{4}}\right\| \left\|\theta^k - \theta^l\right\| \left\|(\epsilon I + \hat{V}^{k-D})^{-\frac{1}{4}} \left(\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\right) \right\|\right] \\
\overset{(e)}{\le}&\frac{L\epsilon^{-\frac{1}{2}}}{12D\alpha_k} \underbrace{\mathbb{E}\left[\|\theta^k - \theta^l\|^2\right]}_{I_1} + \frac{6DL\alpha_k\epsilon^{-\frac{1}{2}}}{2} \underbrace{\mathbb{E}\left[\|\nabla\ell(\theta^l;\xi_m^k) - \nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\|^2\right]}_{I_2}
\end{aligned} \tag{60}$$

where (c) holds due to (59), (d) uses Assumption 1, and (e) applies the Young's inequality.

Applying the Cauchy-Schwarz inequality to I_1, we have

$$\begin{aligned}
I_1 =&\mathbb{E}\Big[\Big\| \sum_{d=1}^{k-l} (\theta^{k+1-d} - \theta^{k-d}) \Big\|^2\Big] \\
\le&(k-l) \sum_{d=1}^{k-l} \mathbb{E}\Big[\|\theta^{k+1-d} - \theta^{k-d}\|^2\Big] \le D \sum_{d=1}^{D} \mathbb{E}\Big[\|\theta^{k+1-d} - \theta^{k-d}\|^2\Big].
\end{aligned} \tag{61}$$

Applying Assumption 2 to I_2, we have

$$\begin{aligned} I_2 =&\mathbb{E}\Big[\big\|\nabla\ell(\theta^l;\xi_m^k)-\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\big\|^2\Big] \\ =&\mathbb{E}\Big[\big\|\nabla\ell(\theta^l;\xi_m^k)\big\|^2\Big]+\mathbb{E}\Big[\big\|\nabla\ell(\theta^l;\xi_m^{k-\tau_m^k})\big\|^2\Big]\le 2\sigma_m^2 \end{aligned} \tag{62}$$

where the last inequality uses Assumption 2. Plugging (61) and (62) into (60), it leads to (57).

Likewise, following the steps to (60), it can be verified that (58) also holds true. ■

Lemma 7 *Under Assumption 2, the parameters $\{h^k,\hat{v}^k\}$ of CADA in Algorithm 1 satisfy*

$$\|h^k\|\le\sigma,\quad\forall k;\quad \hat{v}_i^k\le\sigma^2,\quad\forall k,i \tag{63}$$

where $\sigma:=\frac{1}{M}\sum_{m\in\mathcal{M}}\sigma_m$.

Proof: Using Assumption 2, it follows that

$$\|\boldsymbol{\nabla}^k\|=\left\|\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|\le\frac{1}{M}\sum_{m\in\mathcal{M}}\left\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|\le\frac{1}{M}\sum_{m\in\mathcal{M}}\sigma_m=\sigma. \tag{64}$$

Therefore, from the update (2a), we have

$$\|h^{k+1}\|\le\beta_1\|h^k\|+(1-\beta_1)\|\boldsymbol{\nabla}^k\|\le\beta_1\|h^k\|+(1-\beta_1)\sigma.$$

Since $\|h^1\|\le\sigma$, if follows by induction that $\|h^{k+1}\|\le\sigma,\ \forall k$.

Using Assumption 2, it follows that

$$\begin{aligned} (\nabla_i^k)^2=&\left(\frac{1}{M}\sum_{m\in\mathcal{M}}\nabla_i\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right)^2 \\ \le&\frac{1}{M}\sum_{m\in\mathcal{M}}\left(\nabla_i\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right)^2 \\ \le&\frac{1}{M}\sum_{m\in\mathcal{M}}\left\|\nabla\ell(\theta^{k-\tau_m^k};\xi_m^{k-\tau_m^k})\right\|^2=\frac{1}{M}\sum_{m\in\mathcal{M}}\sigma_m^2\le\sigma^2. \end{aligned} \tag{65}$$

Similarly, from the update (2b), we have

$$\hat{v}_i^{k+1}\le\max\{\hat{v}_i^k,\beta_2\hat{v}_i^k+(1-\beta_2)(\nabla_i^k)^2\}\le\max\{\hat{v}_i^k,\beta_2\hat{v}_i^k+(1-\beta_2)\sigma^2\}.$$

Since $v_i^1=\hat{v}_i^1\le\sigma^2$, if follows by induction that $\hat{v}_i^{k+1}\le\sigma^2$. ■

Lemma 8 *Under Assumption 2, the iterates $\{\theta^k\}$ of CADA in Algorithm 1 satisfy*

$$\left\|\theta^{k+1}-\theta^k\right\|^2\le\alpha_k^2 p(1-\beta_2)^{-1}(1-\beta_3)^{-1} \tag{66}$$

where p is the dimension of θ, $\beta_1<\sqrt{\beta_2}<1$, and $\beta_3:=\beta_1^2/\beta_2$.

Proof: Choosing $\beta_1 < 1$ and defining $\beta_3 := \beta_1^2/\beta_2$, it can be verified that

$$\begin{aligned}
|h_i^{k+1}| &= \left|\beta_1 h_i^k + (1-\beta_1)\nabla_i^k\right| \beta_1|h_i^k| + |\nabla_i^k| \\
&\le \beta_1\left(\beta_1|h_i^{k-1}| + |\nabla_i^{k-1}|\right) + |\nabla_i^k| \\
&\le \sum_{l=0}^{k}\beta_1^{k-l}|\nabla_i^l| = \sum_{l=0}^{k}\sqrt{\beta_3}^{k-l}\sqrt{\beta_2}^{k-l}|\nabla_i^l| \\
&\overset{(a)}{\le} \left(\sum_{l=0}^{k}\beta_3^{k-l}\right)^{\frac{1}{2}}\left(\sum_{l=0}^{k}\beta_2^{k-l}(\nabla_i^l)^2\right)^{\frac{1}{2}} \\
&\le (1-\beta_3)^{-\frac{1}{2}}\left(\sum_{l=0}^{k}\beta_2^{k-l}(\nabla_i^l)^2\right)^{\frac{1}{2}} \qquad (67)
\end{aligned}$$

where (a) follows from the Cauchy-Schwartz inequality.

For $\hat{v}_i^k$, first we have that $\hat{v}_i^1 \ge (1-\beta_2)(\nabla_i^1)^2$. Then since

$$\hat{v}_i^{k+1} \ge \beta_2\hat{v}_i^k + (1-\beta_2)(\nabla_i^k)^2$$

by induction we have

$$\hat{v}_i^{k+1} \ge (1-\beta_2)\sum_{l=0}^{k}\beta_2^{k-l}(\nabla_i^l)^2. \qquad (68)$$

Using (67) and (68), we have

$$\begin{aligned}
|h_i^{k+1}|^2 \le& (1-\beta_3)^{-1}\left(\sum_{l=0}^{k}\beta_2^{k-l}(\nabla_i^l)^2\right) \\
\le& (1-\beta_2)^{-1}(1-\beta_3)^{-1}\hat{v}_i^{k+1}.
\end{aligned}$$

From the update (2c), we have

$$\begin{aligned}
\|\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k\|^2 &= \alpha_k^2\sum_{i=1}^{p}\left(\epsilon+\hat{v}_i^{k+1}\right)^{-1}|h_i^{k+1}|^2 \\
&\le \alpha_k^2 p(1-\beta_2)^{-1}(1-\beta_3)^{-1} \qquad (69)
\end{aligned}$$

which completes the proof. ■

Appendix J. Additional Numerical Results

J.1. Simulation setup

In order to verify our analysis and show the empirical performance of CADA, we conduct experiments in the logistic regression and training neural network tasks, respectively.

In logistic regression, we tested the **covtype** and **ijcnn1** in the main paper, and **MNIST** in the supplementary document. In training neural networks, we tested **MNIST** dataset in the main paper, and **CIFAR10** in the supplementary document. To benchmark CADA, we compared it with some state-of-the-art algorithms, namely ADAM [14], stochastic LAG, local momentum [25, 28], local SGD (or FedAvg) [17] and FedAdam [20].

All experiments are run on a workstation with an Intel i9-9960x CPU with 128GB memory and four NVIDIA RTX 2080Ti GPUs each with 11GB memory using Python 3.6.

J.2. Simulation details

J.2.1. LOGISTIC REGRESSION.

Objective function. For the logistic regression task, we use either the logistic loss for the binary case, or the cross-entropy loss for the multi-class class, both of which are augmented with an ℓ_2 norm regularizer with the coefficient $\lambda = 10^{-5}$.

Data pre-processing. For *ijcnn1* and *covtype* datasets, they are imported from the popular library LIBSVM[1] without further preprocessing. For *MNIST*, we normalize the data and subtract the mean. We uniformly partition *ijcnn1* dataset with 91,701 samples and *MNIST* dataset with 60,000 samples into $M = 10$ workers. To simulate the heterogeneous setting, we partition *covtype* dataset with 581,012 samples randomly into $M = 20$ workers with different number of samples per worker.

For *covtype*, we fix the batch ratio to be 0.001 uniformly across all workers; and for *ijcnn1* and *MNIST*, we fix the batch ratio to be 0.01 uniformly across all workers.

Choice of hyperparameters. For the logistic regression task, the hyperparameters in each algorithm are chosen by hand to roughly optimize the performance of each algorithm. We list the values of parameters used in each test in Tables 1-3.

Algorithm	**stepsize** α	**momentum weight** β	**averaging interval** H/D
FedAdam	$\alpha_l = 100$ $\alpha_s = 0.02$	0.9	$H = 10$
Local momentum	0.1	0.9	$H = 10$
ADAM	0.005	$\beta_1 = 0.9$ $\beta_2 = 0.999$	/
CADA1&2	0.005	$\beta_1 = 0.9$ $\beta_2 = 0.999$	$D = 100$, $c = 5e^{-4}$
Local SGD	0.1	/	$H = 10$
Stochastic LAG	0.1	/	$c = 0.006$

Table 1: Choice of parameters in *covtype*.

1. https://www.csie.ntu.edu.tw/ cjlin/libsvm/

Algorithm	**stepsize** α	**momentum weight** β	**averaging interval** H/D
FedAdam	$\alpha_l = 100$ $\alpha_s = 0.03$	0.9	$H = 10$
Local momentum	0.1	0.9	$H = 20$
ADAM	0.01	$\beta_1 = 0.9$ $\beta_2 = 0.999$	/
CADA	0.01	$\beta_1 = 0.9$ $\beta_2 = 0.999$	$D = 100, \quad c = 10$
Local SGD	0.1	/	$H = 10$
Stochastic LAG	0.1	/	$c = 0.1$

Table 2: Choice of parameters in *ijcnn1*.

Algorithm	**stepsize** α	**momentum weight** β	**averaging interval** H/D
FedAdam	$\alpha_l = 0.1$ $\alpha_s = 0.02$	0.9	$H = 40$
Local momentum	0.1	0.9	$H = 40$
ADAM	0.0005	$\beta_1 = 0.9$ $\beta_2 = 0.999$	/
CADA1&2	0.0005	$\beta_1 = 0.9$ $\beta_2 = 0.999$	$D = 100, \quad c = 5e^{-5}$
Local SGD	0.1	/	$H = 40$
Stochastic LAG	0.1	/	$c = 0.1$

Table 3: Choice of parameters in binary-class *MNIST* (digits 3 and 5).

Additional results. Due to space limitation, some simulations results have not been covered in the main paper. Figures 4 and 5 show the performance of all the considered algorithms on *ijcnn1* and *covtype* datasets averaged over 10 Monte Carlo runs. And Figure 6 demonstrates the binary classification performance of all the considered algorithms on *MNIST* dataset averaged over 10 Monte Carlo runs. The shadow region in Figures 4-6 represents one standard deviation.

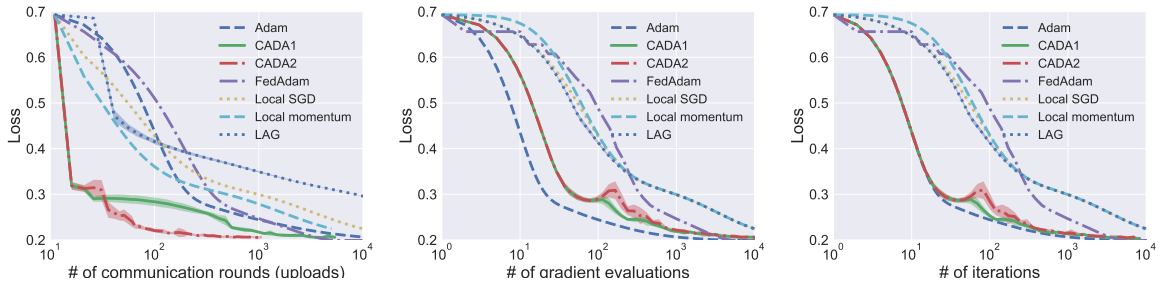

Figure 4: Binary classification on *ijcnn1* dataset averaged over 10 Monte Carlo runs. Shadow region represents one standard deviation.

J.2.2. TRAINING NEURAL NETWORKS.

For training neural networks, we use the cross-entropy loss for all the tests.

Neural network models. For *MNIST* dataset, we use a convolutional neural network with two convolution-ELUmaxpooling layers (ELU is a smoothed ReLU) followed by two fully-connected layers. The first convolution layer is $5 \times 5 \times 20$ with padding, and the second layer is $5 \times 5 \times 50$

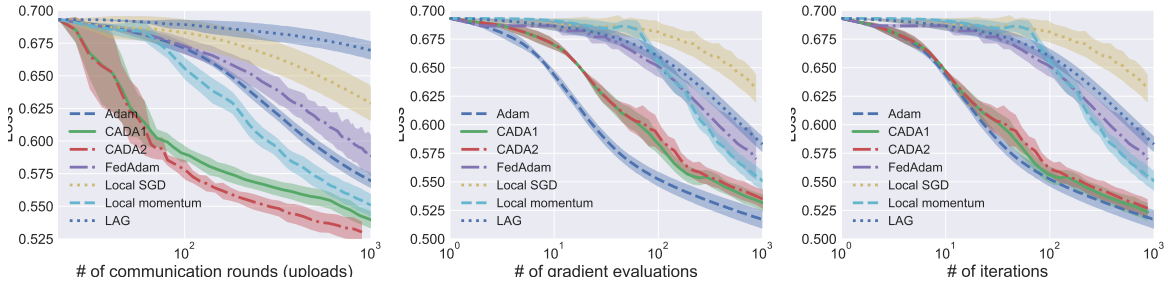

Figure 5: Binary classification on *covtype* dataset averaged over 10 Monte Carlo runs. Shadow region represents one standard deviation.

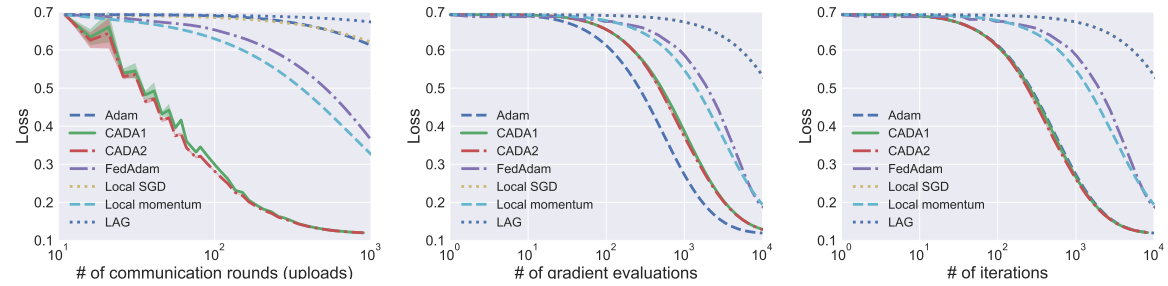

Figure 6: Binary classification on *MNIST* dataset averaged over 10 Monte Carlo runs. Shadow region represents one standard deviation.

with padding. The output of second layer is followed by two fully connected layers with one being 800×500 and the other being 500×10. The output goes through a softmax function. For *CIFAR10* dataset, we use the popular neural network architecture *ResNet20* [2] which has 20 and roughly 0.27 million parameters. We do not use a pre-trained model.

Data pre-processing. We uniformly partition *MNIST* and *CIFAR10* datasets into $M = 10$ workers. For *MNIST*, we use the raw data without preprocessing. The minibatch size per worker is 12. For *CIFAR10*, in addition to normalizing the data and subtracting the mean, we randomly flip and crop part of the original image every time it is used for training. This is a standard technique of data augmentation to avoid over-fitting. The minibatch size for *CIFAR10* is 50 per worker.

Choice of hyperparameters. For *MNIST* dataset which is relatively easy, the hyperparameters in each algorithm are chosen by hand to optimize the performance of each algorithm. We list the values of parameters used in each test in Table 4.

2. https://github.com/akamaster/pytorch_resnet_cifar10

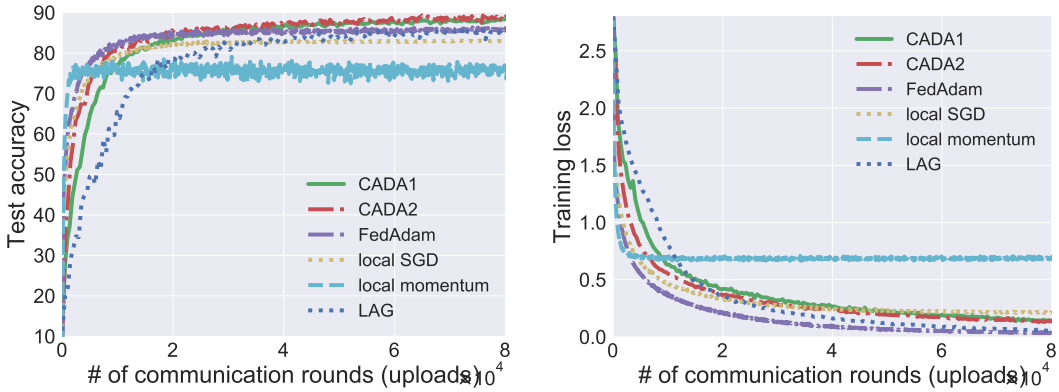

Figure 7: Testing accuracy and training loss versus the communication uploads on *CIFAR10*.

Algorithm	**stepsize** α	**momentum weight** β	**averaging interval** H/D
FedAdam	$\alpha_l = 0.1$ $\alpha_s = 0.001$	0.9	$H = 2$
Local momentum	0.1	0.9	$H = 8$
ADAM	0.0005	$\beta_1 = 0.9$ $\beta_2 = 0.999$	/
CADA1&2	0.0005	$\beta_1 = 0.9$ $\beta_2 = 0.999$	$D = 100, \quad c = 1000$
Local SGD	0.1	/	$H = 8$
Stochastic LAG	0.1	/	$c = 1000$

Table 4: Choice of parameters in multi-class *MNIST*.

For *CIFAR10* dataset, we search the best values of hyperparameters from the following search grid on a per-algorithm basis to optimize the testing accuracy versus the number of communication rounds. The chosen values of parameter are listed in Table 5.

Local SGD: $\alpha \in \{0.1, 0.01, 0.001\}$; $H \in \{4, 6, 8\}$.
FedAdam: $\alpha_s \in \{0.1, 0.01, 0.001\}$; $\alpha_l \in \{1, 0.5, 0.1\}$; $H \in \{4, 6, 8\}$.
Local momentum: $\alpha \in \{0.1, 0.01, 0.001\}$; $H \in \{4, 6, 8\}$.
CADA1: $\alpha \in \{0.1, 0.01, 0.001\}$; $c \in \{0.05, 0.1, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8\}$.
CADA2: $\alpha \in \{0.1, 0.01, 0.001\}$; $c \in \{0.05, 0.1, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8\}$.
LAG: $\alpha \in \{0.1, 0.01, 0.001\}$; $c \in \{0.05, 0.1, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8\}$.

Algorithm	**stepsize** α	**momentum weight** β	**averaging interval** H/D
FedAdam	$\alpha_l = 0.1$ $\alpha_s = 0.1$	0.9	$H = 4$
Local momentum	0.1	0.9	$H = 6$
CADA1	0.1	$\beta_1 = 0.9$ $\beta_2 = 0.99$	$D = 50, c = 1.5$
CADA2	0.1	$\beta_1 = 0.9$ $\beta_2 = 0.99$	$D = 50, c = 0.3$
Local SGD	0.1	/	$H = 6$
Stochastic LAG	0.1	/	$c = 0.05$

Table 5: Choice of parameters in *CIFAR10*.

Additional results. In addition to the results presented in the main paper on *MNIST* dataset in Figure 3, we report a new set of simulations on the image classification task on *CIFAR10* in Figures

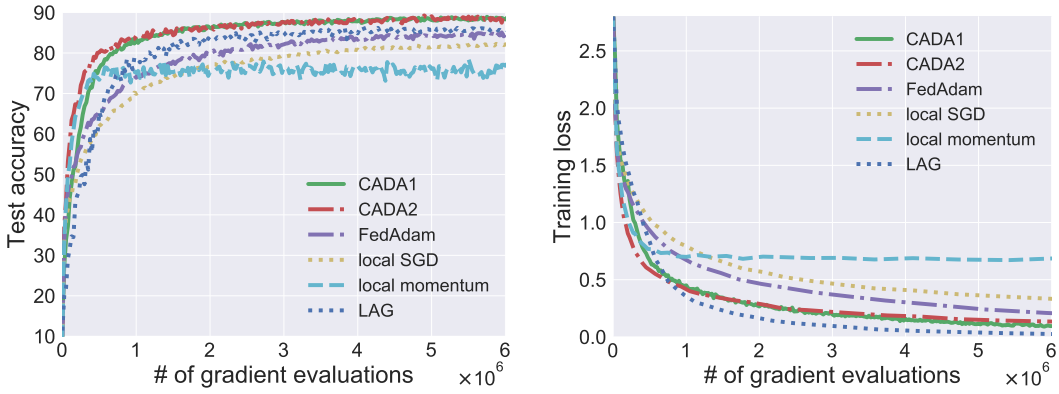

Figure 8: Testing accuracy and training loss versus the gradient evaluation on *CIFAR10*.

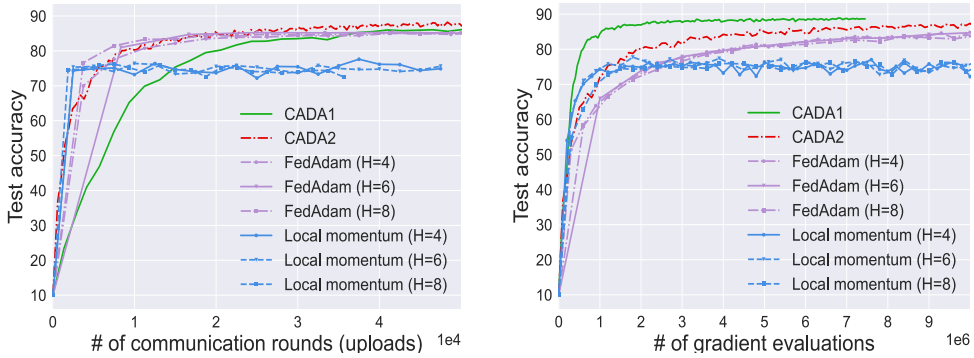

Figure 9: FedAdam and local momentum on *CIFAR10* dataset under different interval H.

7-9. Figure 7 reports the testing accuracy and training loss versus the number of communication uploads. Figure 8 demonstrates the testing accuracy and training loss versus the number of stochastic gradient evaluations. Figure 9 compares the performance of FedAdam and local momentum on *CIFAR10* dataset under different averaging interval H.

Different from the logistic regression case, we observe that FedAdam has very impressive performance in training deep neural networks. In Figure 7, CADA1 and CADA2 require slightly more number of communication rounds than FedAdam at the initial stage of learning, but achieve at least 3-4% higher accuracy in the steady stage than the comparators. Local momentum method achieves a reasonable accuracy with the fewest number of communication, but the test accuracy does not get further improvement. This reduced test accuracy is common among local SGD-type methods, which has also been studied theoretically; see e.g., [7]. In Figure 8, CADA1 and CADA2 require fewer number of stochastic gradient evaluations to achieve certain testing accuracy or training loss than the comparators that are based on multiple local updates. This implies that our CADA methods do not reduce communication at the expense of sacrificing computation overhead. In Figure 9, FedAdam and local momentum under a larger averaging interval H have faster convergence speed at the initial stage, but they reach slightly lower accuracy compared with that under a smaller H.