Communication-Efficient Regret-Optimal Distributed Online Convex Optimization

Jiandong Liu, Lan Zhang, Fengxiang He, Chi Zhang, Shanyang Jiang, and Xiang-Yang Li, *Fellow, IEEE*

***Abstract*—Online convex optimization in distributed systems has shown great promise in collaboratively learning on data streams with massive learners, such as in collaborative coordination in robot and IoT networks. When implemented in communication-constrained networks like robot and IoT networks, two critical yet distinct objectives in distributed online convex optimization (DOCO) are minimizing the overall regret and the communication cost. Achieving both objectives simultaneously is challenging, especially when the number of learners n and learning time T are prohibitively large. To address this challenge, we propose novel algorithms in typical adversarial and stochastic settings. Our algorithms significantly reduce the communication complexity of the algorithms with the state-of-the-art regret by a factor of $\mathcal{O}(n^2)$ and $\tilde{\mathcal{O}}(\sqrt{nT})$ in adversarial and stochastic settings, respectively. We are the first to achieve nearly optimal regret and communication complexity simultaneously up to polylogarithmic factors. We validate our algorithms through experiments on real-world datasets in classification tasks. Our algorithms with appropriate parameters can achieve $90\% \sim 99\%$ communication saving with close accuracy over existing methods in most cases. The code is available at https://github.com/GGBOND121382/Communication-Efficient_Regret-Optimal_DOCO.**

***Index Terms*—Communication complexity, distributed online learning, convex optimization.**

I. Introduction

DISTRIBUTED online convex optimization (DOCO) [1], [2], [3], [4] has shown promising performance in collaborative learning on feedback data from clients served sequentially by massive learners. Besides loss or accuracy, a crucial criterion for DOCO algorithms is communication cost, especially in communication-constrained networks like robot and IoT networks. The energy cost of communications is orders of magnitude higher than that of local computation in these networks [5], [6], [7]. In distributed real-time coordination and localization in robot networks [8], [9], for example, robots collaborate to minimize the accumulated loss, measured by the difference between their predictions and the streaming observations of targets. High communication costs can render the algorithm impractical since bandwidth or communication power is typically limited in these applications, leading to quick energy depletion among resource-constrained robots [10].

Existing works have designed several DOCO algorithms with low regret loss, a standard metric for the loss of models [11], [12], [13], or reduced communication complexity [2], [14]. However, the literature lacks a general theory for DOCO algorithms that can simultaneously achieve low regret loss and communication complexity. This work aims to achieve this challenging goal to make DOCO algorithms practical in communication-constrained networks. Each learner measures its (pseudo) regret loss as the difference between the (expected) accumulated loss incurred by its models and that incurred by the best single model. The algorithm's communication complexity is evaluated by the message complexity [12], [15], which refers to the number of messages transmitted in the algorithm.

We consider two typical settings in DOCO: *adversarial* and *stochastic*. In the adversarial setting, feedback data can be arbitrary or adaptive to historical models. This setting covers applications where feedbacks vary vastly over time, such as in distributed tracking of moving targets in sensor networks [8]. In the stochastic setting, each learner's feedback data follow a fixed distribution.[1] This setting has wide applications in statistical learning, inference, and coordination in networks [1], [16].

To understand the limitations of DOCO, we establish communication complexity lower bounds to achieve the minimax regret loss, which represents the optimal worst-case regret loss for DOCO algorithms on arbitrary connected learner networks with arbitrary loss functions. We consider DOCO algorithms with n learners and learning time T. For adversarial DOCO, we prove that the lower bound is linear in T. For stochastic DOCO, the lower bound is nearly linear in n. Generally, the learning time T

[1]The distributions of each learner's obtained feedbacks in stochastic DOCO can be different. We include the results of the i.i.d. stochastic setting where all feedbacks are sampled from the same distribution in Appendix B in the supplementary material, available online.

Manuscript received 3 June 2023; revised 12 May 2024; accepted 18 May 2024. Date of publication 21 May 2024; date of current version 4 October 2024. This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0110400, in part by the Innovation Program for Quantum Science and Technology under Grant 2021ZD0302900, in part by China National Natural Science Foundation under Grant 62132018, in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2023C01029, in part by the Fundamental Research Funds for the Central Universities under Grant WK2150110024, and in part by the Key Laboratory of Internet and industrial integration and innovation (CAICT), MIIT. Recommended for acceptance by D. Grosu. *(Corresponding author: Xiang-Yang Li.)*

Jiandong Liu, Lan Zhang, Chi Zhang, Shanyang Jiang, and Xiang-Yang Li are with the LINKE Lab, University of Science and Technology of China, Hefei 230026, China (e-mail: jdliu@mail.ustc.edu.cn; zhanglan@ustc.edu.cn; gzhnciha@mail.ustc.edu.cn; yang12@mail.ustc.edu.cn; xiangyangli@ustc.edu.cn).

Fengxiang He is with the Artificial Intelligence and its Applications Institute, School of Informatics, University of Edinburgh, EH8 9AB Edinburgh, U.K. (e-mail: F.He@ed.ac.uk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPDS.2024.3403883, provided by the authors.

Digital Object Identifier 10.1109/TPDS.2024.3403883

TABLE I
MINIMAX REGRET BOUNDS, COMMUNICATION COMPLEXITY LOWER BOUNDS FOR REGRET-OPTIMAL DOCO, CURRENTLY STATE-OF-THE-ART (SOTA) COMMUNICATION COMPLEXITY FOR DOCO WITH THE STATE-OF-THE-ART REGRET, AND OUR ALGORITHMS' COMMUNICATION COMPLEXITY

	Adversarial	Stochastic
Minimax regret	$\mathcal{O}(n^{3/2}\sqrt{T})$ (**Cor. 1**)	$\mathcal{O}(\sqrt{nT})$ [13]
Comm. lower bounds	$\Omega(T)$ (**Cor. 1**)	$\Omega(n)$ (**Cor. 2**)
SOTA algs.' comm.	$\mathcal{O}(n^2 T)$ [11]	$\mathcal{O}(n^{3/2}\sqrt{T})$ [13]
Our algs.' comm.	$\boldsymbol{\mathcal{O}(T)}$ **(Cor. 3)**	$\boldsymbol{\tilde{\mathcal{O}}(n)}$ **(Thm. 4)**

The $\tilde{\Omega}$ ($\tilde{\mathcal{O}}$) notation hides polylogarithmic factors in the number of learners n and learning time T when $T >> n$.

is significantly greater than the number of learners n. Therefore, our theory implies that algorithms in stochastic DOCO may be considerably more communication-efficient than those in adversarial DOCO.

Based on the above understanding, we propose novel DOCO algorithms that utilize each communication more effectively. For the adversarial setting, we design the dual-block BFS-tree-aided DOCO (DB-TDOCO) algorithm. DB-TDOCO updates models twice in two fine-tuned blocks each time the learners gather their gradients in a Breadth-First Search (BFS) tree. Compared with the traditional distributed mini-batch algorithm [13], DB-TDOCO enables more frequent and effective model updates. For the stochastic setting, we devise the distributed batch-to-online (DB2O) algorithm. DB2O significantly reduces communication complexity since the learners infrequently update models. Each updated model is obtained from a parallel run of a communication-efficient distributed batch optimization algorithm.

Our proposed algorithms offer notable theoretical advantages. DB-TDOCO and DB2O achieve nearly optimal communication complexity and regret bounds for adversarial and stochastic DOCO, respectively. This is the first time that such results have been achieved. Our results significantly reduce the state-of-the-art communication complexity [11], [13] for algorithms with the currently state-of-the-art regret (cf. Table I). In Section VI, we show that our algorithms outperform the state-of-the-art on typical cycle, grid, and clique networks. In Appendix C, available online, we additionally discuss the communication complexity on networks with specific diameters and establish our algorithms' optimality.

In this paper, we address DOCO within a distributed computing framework [12], [17], where learners can decide when and with whom to communicate. Our algorithms involve learners coordinating to run convergecast and broadcast protocols [17] for exchanging gradients or models. This communication model is pertinent in networks like robot and IoT networks [8], [9], where broadcast and convergecast are applicable across wired or wireless devices [6], [8], [18], [19]. DOCO algorithms in [13], [20] are tailored to this paradigm. In contrast, an alternative body of literature [2], [11], [21] has devised decentralized DOCO algorithms, following the *gossip* pattern, for networks lacking coordination (e.g., unstable wireless networks [12], [22]). In the *gossip* framework, learners communicate with neighbors and locally average their models, representing a specific instance within our broader communication model.

We assess our algorithms by conducting distributed online logistic regression on real-world datasets [23], [24]. Results show that our algorithms achieve substantial communication savings, up to 95% compared to the state-of-the-art [11], [25], while maintaining comparable accuracy in the adversarial setting. In the stochastic setting, communication savings range from 90% to 99% compared to the state-of-the-art [13], with comparable accuracy, achieved through appropriate parameter selection. The code is available at https://github.com/GGBOND121382/Communication-Efficient_Regret-Optimal_DOCO.

II. RELATED WORK

DOCO has been widely adopted in distributed learning systems that require real-time AI service [1], [2], [3], [4], [26]. As many distributed systems operate on communication-constrained networks, communication-efficient and effective learning algorithms have become highly attractive [2], [4], [27]. In adversarial DOCO, the decentralized *gossip* algorithm [11], [12] currently achieves the state-of-the-art regret of $\mathcal{O}(\Gamma n^{3/2}\sqrt{T})$, where Γ measures the connectivity of the learner network, ranging from $\mathcal{O}(1)$ to $\mathcal{O}(n^2)$ for different networks. The worst-case communication cost of *gossip* is $\mathcal{O}(n^2 T)$. Wan et al. [2] introduced the decentralized block online conditional gradient algorithm (D-BOCG), which offers communication savings over *gossip* by a factor of $\mathcal{O}(\sqrt{T})$, with a regret enlarged by a factor of $\mathcal{O}(T^{1/4})$. For stochastic DOCO, the distributed mini-batch algorithm (DMA) [13] achieves the minimax regret with the currently state-of-the-art communication cost of $\mathcal{O}(n^{3/2}\sqrt{T})$. Although DMA was originally designed for i.i.d. stochastic data, it naturally works for learners with diverse feedback distributions and the same bounds hold. Besides message complexity, van der Hoeven et al. [28], Acharya et al. [14], and Tu et al. [4] investigate DOCO with reduced bit complexity based on gradient quantization, a common technique for compressing message bit-length in learning tasks.

Another area of research focuses on the communication complexity lower bound of DOCO. Wan et al. [29] demonstrate that, for the dependence on T, the learners need $\Omega(T)$ communication cost to achieve the minimax regret in adversarial DOCO. van der Hoeven et al. [28] consider a DOCO problem where one learner wakes up at each time and serves a client. They establish the regret lower bounds for this DOCO problem with different budgets for bit complexity. Wang et al. [25] proved the $\Omega(n)$ communication complexity lower bound of distributed stochastic bandits to achieve the minimax regret.

Some of our analysis is inspired by online convex optimization with delayed feedback (OCOD) [20], [30], [31] and batched bandits [32], [33], [34]. On the one hand, OCOD provides an algorithmic framework for online learning in scenarios where feedback data are delayed due to networking or communication constraints as we encounter in DOCO. On the other hand, batched bandits investigate the regret of bandit online optimization where the update times of models are limited. These works provide insights into the regret analysis of DOCO, where learners cannot update models frequently due to a lack of information under communication constraints.

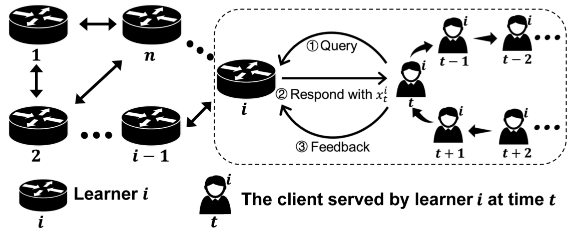


Fig. 1. An example learner network in DOCO and the service pattern of learner $i \in [n]$ in the network.

Despite great contributions in various aspects, the literature lacks a comprehensive analysis of the communication complexity lower bounds for regret-optimal DOCO. Furthermore, there is a lack of algorithms that attain these lower bounds.

III. Problem Formulation

We formalize the problem settings for online convex optimization (OCO) and distributed OCO (DOCO).

Online convex optimization: In OCO, an agent acts as a learner at each time step $t \in [T]$ to optimize an AI model using streaming data. More specifically, the learner handles incoming clients based on a model parameter $x_t \in \mathcal{C}$, where $\mathcal{C} \subset \mathbb{R}^d$ is the feasible region. The client then provides feedback data ξ_t to the learner. The learner derives the loss function $f_t(\cdot) = f(\cdot; \xi_t)$ from a function f and updates x_{t+1} based on $f_t(\cdot)$.

Distributed OCO: DOCO extends OCO in distributed settings, where n agents collaborate in learning using their data streams. These n learners communicate in a connected network, as shown in Fig. 1. At each time $t \in [T]$, each learner $i \in [n]$ serves a client using $x_t^i \in \mathcal{C}$, with $\mathcal{C} \subset \mathbb{R}^d$ denoting the feasible region. The client then sends feedback data ξ_t^i to learner i, which may vary for different i due to clients' unique preferences or noisy feedbacks (e.g., in distributed tracking based on noisy observations). The performance of x_t^i is evaluated by the cumulative loss function across all client feedbacks at time t. More precisely, given a predefined function $f(\cdot; \xi)$ with ξ being an arbitrary feedback, the loss function is expressed as $f_t = \sum_{i=1}^n f_t^i$, where $f_t^i(\cdot) = f(\cdot; \xi_t^i)$. Learner i can communicate with neighbors and update x_{t+1}^i using a partial loss function $f_t^i = f(\cdot; \xi_t^i)$ and messages from neighbors.

Feedback Settings: In DOCO, two typical feedback settings are considered: *adversarial* and *stochastic*. In adversarial DOCO, the clients at time t select feedback $\xi_t^i, i \in [n]$, arbitrarily or adaptively, even based on historical models and messages. In stochastic DOCO, the feedback ξ_t^i is sampled from a distribution $\mathbb{P}_i$ for $i \in [n]$.

Network Topology: Similar to existing DOCO algorithms [2], [11], [13], we consider network topologies as arbitrary undirected connected graphs. Typical topologies in DOCO applications include cycles, grids, and cliques [2], [11], [22].

Communication Pattern: We focus on DOCO algorithms employing the standard static communication pattern [1], [2], [13]. In this setup, learners operate within a static communication pattern, where neighboring learners i and j for $i, j \in [n]$ communicate at specific time points t, determined by the learner network and the total learning time T.

Within this pattern, learners can coordinate to exchange messages such as gradients or models. Two widely adopted

procedure Convergecast($\{y_i\}_{i \in [n]}$, $\mathcal{T}$)
  Each leaf learner i in $\mathcal{T}$ sends $z_i = y_i$ to its parent
  Upon receiving $\{z_i\}_{i \in N_j}$ from all its children, each learner j computes
  $$z_j \leftarrow y_j + \Sigma_{i \in N_j} z_i,$$
  where N_j denotes the children set of learner j
  After computing z_j, learner j sends z_j to its parent if learner j is non-root
end procedure
procedure Broadcast(y, $\mathcal{T}$)
  The root learner in $\mathcal{T}$ sends y to its children
  Upon receiving y from its parent, each non-leaf learner j sends y to its children
end procedure

Fig. 2. Convergecast and Broadcast procedures in a BFS tree $\mathcal{T}$ of the learner network [17], where y_i represents the vector owned by learner $i \in [n]$ and y is owned by the root learner.

communication protocols in DOCO are *convergecast* and *broadcast* [13], [20], facilitating the aggregation and distribution of messages in a BFS tree of the network. The procedures of *convergecast* and *broadcast* are presented in Fig. 2. Both protocols involve an $\mathcal{O}(n)$ communication cost and a delay proportional to the network's diameter. Example applications include distributed real-time localization [8], [9], where robots can convergecast gradients evaluated on the current localization model to a single unit for updates, followed by broadcasting the updated model.

Performance Metrics: The performance of each learner's model sequence is measured using the (pseudo) regret loss. In the adversarial setting, each learner $i \in [n]$ aims to minimize the *regret loss* [11], [12]

$$\mathcal{R}_i(T) \triangleq \Sigma_{t=1}^T f_t(x_t^i) - \min_{x \in \mathcal{C}} \Sigma_{t=1}^T f_t(x). \tag{1}$$

In the stochastic setting, let $\bar{f}_i(x) \triangleq \mathbb{E}_{\xi \sim \mathbb{P}_i}[f(x; \xi)]$ and $\bar{f}(x) \triangleq \sum_{i=1}^n \bar{f}_i(x)$. Each learner i seeks to minimize the *expected pseudo regret loss* [13]

$$\bar{\mathcal{R}}_i(T) \triangleq \mathbb{E}\left[\Sigma_{t=1}^T \left[\bar{f}(x_t^i) - \min_{x \in \mathcal{C}} \bar{f}(x)\right]\right]. \tag{2}$$

For instance, in distributed real-time localization, the loss function quantifies the sum of distances between a prediction of the localization model and all robots' noisy observations. Each robot aims to minimize the regret loss to reduce its cumulative localization error. In this paper, we focus on algorithms with the *minimax regret*, i.e., the optimal regret loss achieved by DOCO algorithms under worst-case networks and feedbacks (or feedback distributions in the stochastic setting).

Another performance metric of a DOCO algorithm is its *communication complexity*. In this paper, the communication complexity is quantified by the number of transmitted messages, i.e., the message complexity [12], [17]. In DOCO algorithms, a message typically represents a gradient or model parameter.

Conditions on Loss Functions: This paper considers standard loss functions that are Lipschitz, convex, smooth, and bounded.

The feasible region $\mathcal{C}$ for the models is assumed to be convex, compact, and adhere to a fatness condition.

Definition 1 (Convexity): A function f is convex within a convex compact set $\mathcal{C} \subset \mathbb{R}^d$ if, for all $x, y \in \mathcal{C}$,

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle.$$

Definition 2 (Lipschitz): A function f is L-Lipschitz for some $L > 0$ in a convex compact set $\mathcal{C}$ if, for all $x, y \in \mathcal{C}$,

$$|f(x) - f(y)| \leq L \|x - y\|.$$

Here, $\|\cdot\|$ denotes the Euclidean norm.

Definition 3 (Smoothness): A function f is L_G-smooth for some $L_G > 0$ if, for all $x, y \in \mathcal{C}$,

$$\|\nabla f(x) - \nabla f(y)\| \leq L_G \|x - y\|.$$

Definition 4 (Bounded function and fat set [35]): A function f is bounded if $|f(x)| \leq M$ for all $x \in \mathcal{C}$, where $M > 0$. A set $\mathcal{C}$ is fat if it contains an ℓ_∞-ball of radius $\Omega(\mathrm{poly}(\frac{1}{d}))$.

The assumptions of Lipschitz, convexity, smoothness, and boundedness on the loss functions, and the convexity and compactness assumptions on the feasible region $\mathcal{C}$, are widely adopted in the DOCO literature [1], [2], [12], [13]. The fatness assumption is essential for cutting-plane-based algorithms [35], [36], ensuring sufficient interior space for search. By the analysis in [35], this assumption holds for sets of the form $\{x \in \mathbb{R}^d : h_i(x) \leq 0, i \in [k]\}$, where each $h_i(\cdot)$ is Lipschitz and smooth for $i \in [k]$. However, this assumption does not hold for constraints containing equalities, e.g., a surface in $\mathbb{R}^d$. We can adapt our analysis and algorithms to these regions by operating on the low-dimensional manifold induced by these equalities, which we leave for future work.

Many real-world loss functions satisfy Definitions 1–4. Logistic loss functions [20] and smoothed hinge loss functions [37] with bounded regions and samples, widely adopted in classification tasks [2], [13], [20], [37], and ridge regression functions utilizing bounded regions and samples [20] popular in regression tasks [1], [20], [38], align with these definitions. A specific instance of loss functions satisfying Definitions 1–4 is the *linear function* with a constant dimension $d > 1$

$$f(x) = \langle x, z \rangle \text{ for some } z \in [0,1]^d,$$
$$\mathcal{C} = \{x \in \mathbb{R}^d_{\geq 0} | 1 \leq \|x\|_1 \leq 2\}. \tag{3}$$

In adversarial DOCO, we assume that f_t^i for $i \in [n]$ and $t \in [T]$ satisfy Definitions 1–4 with common parameters L, L_G, and M. In stochastic DOCO, we assume that $f(x;\xi)$ for all possible $\xi \sim \mathbb{P}_i$, $i \in [n]$, satisfies Definitions 1–4. Specifically, in stochastic DOCO, we also consider broader non-Lipschitz functions satisfying the following milder condition.

Definition 5 (Bounded gradient variance and expected initialization risk [13]): Functions $f(x;\xi)$, $\xi \sim \mathbb{P}_i$ for $i \in [n]$, have bounded gradient variance $\sigma^2 > 0$ and expected initialization risk $R > 0$ in a compact convex $\mathcal{C}$ if, for $x \in \mathcal{C}, i \in [n]$,

$$\begin{cases} \mathbb{E}_{\xi \sim \mathbb{P}_i} \|\nabla f(x;\xi) - \nabla \bar{f}_i(x)\|^2 \leq \sigma^2 \\ \bar{f}_i(\hat{x}) - \bar{f}_i(x^*) \leq R \end{cases},$$

where $\bar{f}_i(x) = \mathbb{E}_{\xi \sim \mathbb{P}_i}[f(x;\xi)]$, $x^* \triangleq \arg\min_{x \in \mathcal{C}} \Sigma_{i=1}^n \bar{f}_i(x)$, and $\hat{x} \triangleq \arg\min_{x \in \mathcal{C}} \|x\|$.

The condition above is a relaxed version of the Lipschitz condition as $f(x;\xi), \xi \sim \mathbb{P}_i$ for $i \in [n]$ have gradient variance L^2 and expected initialization risk $L\|\mathcal{C}\|$ if it is L-Lipschitz, where $\|\mathcal{C}\| \triangleq \max_{x,y \in \mathcal{C}} \|x - y\|$. This condition can accommodate broader loss functions like logistic and ridge regression loss functions with noisy gradients [4], [39].

IV. Communication Complexity Lower Bounds

To determine the limits of DOCO, we establish lower bounds on the communication complexity for regret-optimal algorithms. Let r denote the communication budget, i.e., the maximum number of messages that can be transmitted by the learners. We construct hard instances for DOCO where any algorithm will experience suboptimal regret when r falls below a lower bound. In our analysis, we make no assumptions on the bit-lengths of the messages. Consequently, our lower bounds remain valid even when the bit-lengths of the messages are unbounded. This is because we utilize information-theoretic analysis on feedback information. The learners must acquire information related to feedbacks at specific times and promptly update models to achieve the minimax regret. We present brief outlines of the proofs for our theoretical results in this section and defer the full proofs to Appendices D and E in the supplementary material.

A. Communication Complexity for Adversarial DOCO

We begin by examining the communication complexity of adversarial DOCO. We construct a hard instance where the regret loss is $\Omega(n^{3/2}T/\sqrt{r})$ for communication budgets r that are not sufficiently large, as presented in Theorem 1.

Theorem 1: Consider adversarial DOCO with linear loss functions (cf. (3)) and static communication patterns. There exists a learner network and functions $\{f_t^i\}_{t \in [T], i \in [n]}$ such that

$$\mathcal{R}_1(T) = \Omega\left(n^{3/2} \max\{\sqrt{T}, T/\sqrt{r}\}\right), \tag{4}$$

where n is the number of learners, T is the learning time, and r is the communication budget.

As a corollary of Theorem 1, we need $r = \Omega(T)$ to achieve the minimax regret $\mathcal{O}(n^{3/2}\sqrt{T})$.

Corollary 1: Consider adversarial DOCO with loss functions satisfying Definitions 1–4 and static communication patterns. The communication complexity to achieve the minimax regret $\mathcal{O}(n^{3/2}\sqrt{T})$ is $\Omega(T)$ with respect to n and T.

The theory demonstrates that the necessary communication cost in adversarial DOCO increases at a minimum rate of linearly with respect to the learning time T. This result is substantiated by constructing a specific problem instance in which any algorithm will experience suboptimal regret if the communication budget is inadequate. On one hand, the $\Omega(T)$ lower bound is disheartening as it indicates that the learners must communicate frequently during the course of the algorithm. On the other hand, the bound is encouraging as it is independent of the number of learners n. Our bound suggests that the currently state-of-the-art communication complexity of $\mathcal{O}(n^2 T)$ (cf. Table I) may be suboptimal with regard to n.

Proof sketch of Theorem 1

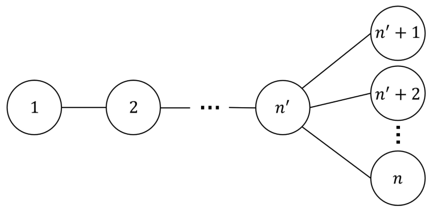


Fig. 3. Claw-shaped network. $n' = \lceil n/2 \rceil$.

1) Constructing the network: We construct the learner network as a claw-shaped one in Fig. 3. In this network, the transmission of each message from a learner $i > n'$ to learner 1 requires $\Omega(n)$ timeslots and communication budget. By the analysis in OCO with delayed feedback [30], the $\Omega(n)$ delay can magnify learner 1's regret by an $\Omega(\sqrt{n})$ factor. This attribute is critical for establishing the $\Omega(n^{3/2}\sqrt{T})$ regret lower bound when the communication budget r is sufficiently large.

2) Constructing loss functions: Let $t_k + 1$ be the first time learner 1 receives information from learner $i > n'$ sent at time $t_{k-1} < t \leq t_k$, and m be the number of such t_k. Let $t_0 = 0$ and $t_{m+1} = T$. We construct linear loss functions as in (3). For learner $i > n'$, we sample $\hat{z}_k$ uniform in $\{0,1\}^d$ for $k \in [m+1]$, and set $f_t^i(x) = \langle x, \hat{z}_k \rangle$ for $t_{k-1} < t \leq t_k$. The constructed loss for learner $i \leq n'$ is zero. Learner 1 then incurs high regret loss since its models are independent of the functions of learners $n'+1$ to n.

3) Lower bounding the regret: We establish that

$$\mathbb{E}[\mathcal{R}_1(T)] = \Omega(nT/\sqrt{m}),$$

based on our loss functions. In the claw-shaped network, $m = \mathcal{O}(\min(T/n, r/n))$. Theorem 1 can then be deduced.

Comparison with existing works: Our lower bound on communication complexity improves upon the result in [29], which proves a lower bound of $\Omega(T)$ based on a hard instance satisfying Definitions 1–4. Notably, their work does not consider the dependence on the number of learners n, and hence does not reveal the suboptimality of the state-of-the-art algorithm's communication complexity of $\mathcal{O}(n^2 T)$ [11]. While Wan et al. [29] employ similar loss functions to ours (cf. (3)), which rely on specific communication time points t_k for $k \in [m]$, they do not consider the influence of transmission delays and network topologies in their construction. In contrast, we obtain tighter bounds by selecting t_k based on the communication budget and transmission delays in a sophisticated network.

B. Communication Complexity for Stochastic DOCO

For the stochastic setting, we prove that a hard instance exists where the regret loss is $\omega(\sqrt{nT})$ for a low communication budget r. We summarize the result in Theorem 2.

Theorem 2: Consider stochastic DOCO with linear loss functions (cf. (3)) and static communication patterns. There exists a learner network and loss functions $f(x; \xi_t^i)$ for $i \in [n]$ and $t \in [T]$, where $\xi_t^i \sim \mathbb{P}_i$ for some distribution $\mathbb{P}_i$, such that

$$\bar{\mathcal{R}}_1(T) = \Omega\left(\max\left\{\min\left\{n^2, nT\right\}, (nT)^{\frac{1}{2-2^{-2r/n}}}\right\}\right), \quad (5)$$

where n is the number of learners, T is the learning time, and r is the communication budget.

Via similar arithmetic computations as in batched bandits [32], we need $r = \Omega(n \ln\ln T)$ to make $\bar{\mathcal{R}}_1(T)$ in (5) equal to $\mathcal{O}(\sqrt{nT})$, the minimax regret, when $T \gg n$.

Corollary 2: Consider stochastic DOCO with loss functions satisfying Definitions 1–4 and static communication patterns. The communication complexity to achieve the minimax regret $\mathcal{O}(\sqrt{nT})$ is $\Omega(n \ln\ln T)$ with respect to n and T.

The theory shows that for stochastic DOCO, the lower bound on communication complexity required to attain the minimax regret is $\tilde{\Omega}(n)$, which is almost insensitive to T. We establish this bound with an instance where any algorithm fails to achieve the optimal regret if their communication budget is below our lower bound. Initially, the tightness of this bound may appear dubious as the regret seems to rise swiftly in T if the communication cost is nearly constant in T. Nevertheless, once this bound is reached, it is highly advantageous as the learning time tends to be lengthy in real-world applications [1], [13].

Proof sketch of Theorem 2.

1) Constructing the network: Similarly, as in adversarial DOCO, we take the claw-shaped learner network in Fig. 3.

2) Constructing loss functions: We sample ℓ uniformly from $[d]$. The loss functions are as in (3), i.e., $f(x; \xi_t^i) = \langle x, \xi_t^i \rangle$. For learner $i \leq n'$, ξ_t^i is an all-zero vector. For learner $i > n'$, the ℓth coordinate of ξ_t^i is drawn from the Bernoulli distribution with mean $\frac{1}{2} - \epsilon$, and the other coordinates are uniform in $\{0,1\}$. We tune the parameter ϵ to maximize learner 1's regret.

3) Lower bounding the regret: Let $t_k, k \in [m]$ be the time points we defined in the proof sketch of Theorem 1. By optimally tuning the parameter ϵ in the loss, we derive that:

$$\mathbb{E}_\ell[\bar{\mathcal{R}}_1(T)] = \Omega(n \min\{n, T\} + n^{\frac{1}{2-2^{-m}}} T^{\frac{1}{2-2^{-m}}}).$$

It holds that $m \leq r/n' \leq 2r/n$ in the claw-shaped network. Theorem 2 then follows.

Comparison with existing works: To our best knowledge, this is the first work that analyzes the communication complexity for general DOCO in the stochastic setting.

V. Communication-Efficient DOCO Algorithm

Building on the insights from our lower bound analysis, we design two novel DOCO algorithms: dual-block BFS-tree-aided DOCO (DB-TDOCO) and distributed batch-to-online (DB2O). These algorithms optimize communication resource utilization, achieving optimal regret and communication complexity in the number of learners n and learning time T, as outlined in Section IV, within polylogarithmic factors. Proof sketches of these results are presented in this section, with details deferred to Appendices G and H in the supplementary material.

In Table II, we compare the performance of our algorithms with state-of-the-art [2], [11], [12], [13]. Our DB-TDOCO and DB2O$_a$ algorithms notably reduce communication complexity while maintaining comparable or lower regret, under the same

TABLE II
REGRET BOUNDS, COMMUNICATION COMPLEXITY, AND ASSUMPTIONS ON LOSS FUNCTION FOR DOCO ALGORITHMS FOR THE DEPENDENCE OF THE NUMBER OF LEARNERS n AND LEARNING TIME T WHEN $T \gg n$. *GOSSIP* [11], [12] AND D-BOCG [2] RESPECTIVELY ACHIEVE STATE-OF-THE-ART REGRET AND COMMUNICATION COMPLEXITY IN ADVERSARIAL DOCO

Feedbacks	DOCO algorithm	Communication complexity	Regret bounds	Assumptions on loss functions
Adversarial	*gossip* [11], [12]	$\mathcal{O}(n^2 T)$	$\mathcal{O}(n^{3/2}\Gamma\sqrt{T})$	Definitions 1 and 2
	D-BOCG [2]	$\mathcal{O}(n^2\sqrt{T})$	$\mathcal{O}(n^{3/2}\Gamma T^{3/4})$	
	DB-TDOCO (topology-independent)	$\mathcal{O}(T)$	$\mathcal{O}(n^{3/2}\sqrt{T})$	
	DB-TDOCO (diameter-dependent)	$\mathcal{O}(nT/D)$	$\mathcal{O}(n\sqrt{DT})$	
Stochastic	DMA [13]	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(\sqrt{nT})$	Definitions 1, 3, and 5
	$\mathbf{DB2O}_a$	$\tilde{\mathcal{O}}(n^{5/4}T^{1/4})$	$\tilde{\mathcal{O}}(\sqrt{nT})$	
	$\mathbf{DB2O}_c$	$\tilde{\mathcal{O}}(n)$	$\tilde{\mathcal{O}}(\sqrt{nT})$	Definitions 1, 4, and 5

The parameter Γ in gossip and D-BOCG is the inverse of the spectral gap of the network's gossip matrix [40], ranging from $\mathcal{O}(1)$ to $\mathcal{O}(n^2)$. DMA [13] is the stochastic DOCO algorithm achieving both state-of-the-art metrics. Our DB-TDOCO algorithm has two variants: one with parameters independent of the network topology, and the other dependent on network diameter $D < n$. DB2O$_a$ and DB2O$_c$ are two variants of our DB2O algorithm with cutting-plane [35] and accelerated gradient descent [41] update rules.

assumptions as *gossip* and DMA, respectively, which achieve top-tier regret for adversarial and stochastic DOCO. They reduce the communication cost of *gossip* and DMA by $\mathcal{O}(n^2)$ and $\tilde{\mathcal{O}}((nT)^{1/4})$ factors, respectively. Furthermore, our DB2O$_c$ reduces the communication cost of DMA by an $\tilde{\mathcal{O}}(\sqrt{nT})$ factor, assuming bounded loss functions and a fat feasible region (cf. Definition 4) while relaxing the smoothness assumption (cf. Definition 3). Notably, our algorithms handle broader functions with milder assumptions compared to those in Section IV. Specifically, DB-TDOCO accommodates convex and Lipschitz loss functions, including non-smooth and unbounded ones like hinge loss functions with bounded regions and samples in SVM-based classifications [11], even with additive noise. Meanwhile, DB2O$_c$ is suitable for convex and bounded functions with bounded gradient variance, such as hinge loss with noisy gradients [39]. Finally, DB-TDOCO and DB2O$_a$ apply to non-fat feasible regions, such as the constrained simplex in linear programming [42], [43]. Our bounds in Section IV remain valid for these milder assumptions as we construct hard instances under stricter conditions that inherently fulfill the milder ones.

A. Algorithm Design for the Adversarial Setting

Our adversarial DOCO algorithm is inspired by online convex optimization with delayed feedback (OCOD) [30]. OCOD research shows that a feedback delay of $\Theta(n)$ time points leads to regret loss increasing by an $\mathcal{O}(\sqrt{n})$ factor. This is due to difficulties in the model catching up with variations in other learners' collected adversarial feedbacks in time. In DOCO, through convergecast (cf. Fig. 2), learners can aggregate their gradients at each time to a single learner, requiring $\mathcal{O}(n)$ time points and communication cost. Then, this learner updates and broadcasts models, achieving a minimax regret of $\mathcal{O}(n^{3/2}\sqrt{T})$. Nonetheless, this approach incurs a communication cost of $\mathcal{O}(nT)$. To minimize communication cost to $\mathcal{O}(T)$, learners communicate gradients in blocks, with the block size finely tuned to maintain the minimax regret.

Building on the aforementioned insight, we propose the dual-block BFS-tree-aided DOCO (DB-TDOCO) algorithm, which achieves improved performance by updating models in finely tuned blocks. We divide the learning time into $m+1$ intervals $[t_{k-1}+1, t_k], k \in [m+1]$, where $t_0 = 0$ and $t_{m+1} = T$. Each interval $[t_{k-1}+1, t_k]$ is further divided into two blocks: $I_{2k-1} \triangleq [t_{k-1}+1, t_k - 2\mu]$ and $I_{2k} \triangleq [t_k - 2\mu + 1, t_k]$, where μ denotes the height of a Breadth-First Search (BFS) tree of the learner network. Within each interval $[t_{k-1}+1, t_k]$, learners update models twice in blocks I_{2k-1} and I_{2k} based on gradients of previous blocks communicated in the BFS tree. The model update process of DB-TDOCO is illustrated in Fig. 4, and the detailed algorithm is presented in Algorithm 1.

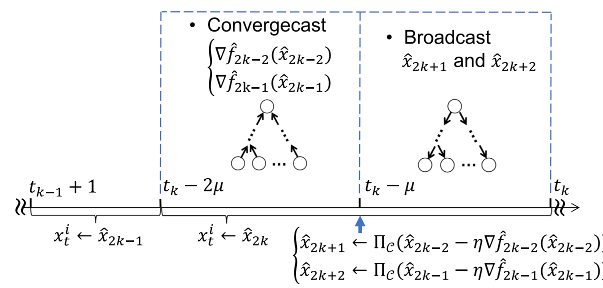


Fig. 4. Model updates in DB-TDOCO. $\hat{f}_{2k-2}$ and $\hat{f}_{2k-1}$ represent block loss functions within blocks $I_{2k-2} \triangleq [t_{k-1} - 2\mu + 1, t_{k-1}]$ and $I_{2k-1} \triangleq [t_{k-1}+1, t_k - 2\mu]$, respectively. Here, $\hat{f}_\ell \triangleq \frac{1}{n\mathcal{I}}\Sigma_{s\in I_\ell} f_s$ for each ℓ, where $\mathcal{I}$ is the largest block size. μ denotes the height of the learner network's BFS tree. $\Pi_{\mathcal{C}}$ projects models onto the feasible region $\mathcal{C}$.

The *dual-block* model updates in DB-TDOCO enable all loss functions to contribute to updating models. The traditional mini-batch updates [13] cannot achieve this target, which neglects feedbacks when learners convergecast/broadcast messages (i.e., during $[t_k - 2\mu, t_k]$ for $k \in [m]$). DB-TDOCO achieves the minimax regret. Besides, we make DB-TDOCO communication-efficient by tuning the number of intervals $m+1$ optimally.

Theoretical advantages of DB-TDOCO.

We study the regret and communication complexity of DB-TDOCO with arbitrary m and $\{t_k\}_{k\in[m]}$ in Theorem 3.

Algorithm 1: Dual-Block BFS-Tree-Aided DOCO (DB-TDOCO).

Input: BFS tree $\mathcal{T}$, number of intervals $m+1$, and time sequence $0 = t_0 < t_1 < \ldots < t_{m+1} = T$

Initialization:

Each learner sets, for $k \in [m+1]$,

$$\begin{cases} I_{2k-1} \leftarrow [t_{k-1}+1, t_k - 2\mu] \\ I_{2k} \leftarrow [t_k - 2\mu + 1, t_k], \end{cases}$$

where μ is the height of $\mathcal{T}$

Each learner sets $\mathcal{I} \leftarrow \max_{\ell \in [2m+2]} |I_\ell|$

Each learner i initializes:

$$x_t^i = \hat{x}_0 = \hat{x}_1 = \hat{x}_2 \leftarrow \arg\min_{x \in \mathcal{C}} \|x\| \text{ for } t \in I_1 \cup I_2$$

for $k = 1$ **to** m **do**

Convergecast:

At time $t_k - 2\mu$, the learners run:

$$\begin{cases} \text{Convergecast}(\{\nabla \hat{f}_{2k-2}^i(\hat{x}_{2k-2})\}_{i \in [n]}, \mathcal{T}) \\ \text{Convergecast}(\{\nabla \hat{f}_{2k-1}^i(\hat{x}_{2k-1})\}_{i \in [n]}, \mathcal{T}), \end{cases}$$

where $\hat{f}_\ell^i \triangleq \frac{1}{n\mathcal{I}} \Sigma_{s \in I_\ell} f_s^i$ for each ℓ and i

At time $t_k - \mu$, the root learner in $\mathcal{T}$ obtains:

$$\begin{cases} \nabla \hat{f}_{2k-2}(\hat{x}_{2k-2}) = \Sigma_{i=1}^n \nabla \hat{f}_{2k-2}^i(\hat{x}_{2k-2}) \\ \nabla \hat{f}_{2k-1}(\hat{x}_{2k-1}) = \Sigma_{i=1}^n \nabla \hat{f}_{2k-1}^i(\hat{x}_{2k-1}) \end{cases}$$

Global Update:

At time $t_k - \mu$, the root learner in $\mathcal{T}$ updates:

$$\begin{cases} \hat{x}_{2k+1} \leftarrow \Pi_{\mathcal{C}}(\hat{x}_{2k-2} - \eta \nabla \hat{f}_{2k-2}(\hat{x}_{2k-2})) \\ \hat{x}_{2k+2} \leftarrow \Pi_{\mathcal{C}}(\hat{x}_{2k-1} - \eta \nabla \hat{f}_{2k-1}(\hat{x}_{2k-1})), \end{cases} \tag{6}$$

where $\Pi_{\mathcal{C}}(x) \triangleq \arg\min_{y \in \mathcal{C}} \|y - x\|$, and $\eta = \Theta(1/\sqrt{m})$

Broadcast:

At time $t_k - \mu$, the learners run:

$$\begin{cases} \text{Broadcast}(\hat{x}_{2k+1}, \mathcal{T}) \\ \text{Broadcast}(\hat{x}_{2k+2}, \mathcal{T}) \end{cases}$$

Local Update:

At time $t_k + 1$, each learner i updates:

$$\begin{cases} x_t^i \leftarrow \hat{x}_{2k+1} \text{ for } t \in I_{2k+1} \\ x_t^i \leftarrow \hat{x}_{2k+2} \text{ for } t \in I_{2k+2}. \end{cases}$$

end for

Theorem 3: Let the loss functions $\{f_t^i\}_{t \in [T], i \in [n]}$ satisfy Definitions 1 and 2, and the time points $\{t_k\}_{k \in [m]}$ satisfy

$$t_k - t_{k-1} > 2\mu + 1 \text{ for } k \in [m+1].$$

DB-TDOCO features $\mathcal{O}(n\mathcal{I}\sqrt{m})$ regret with $\mathcal{O}(mn)$ communication cost, where $2m+2$ is the number of blocks and $\mathcal{I}$ is the largest length of blocks $\{I_k\}_{k \in [2m+2]}$.

Leveraging the results from Theorem 3, DB-TDOCO can attain the optimal regret and communication complexity, as established in Corollary 2, by selecting a specific value of m and $\{t_k\}_{k \in [m]}$. We provide the details in Corollary 3.

Corollary 3: Let the loss functions $\{f_t^i\}_{t \in [T], i \in [n]}$ satisfy Definitions 1 and 2. In DB-TDOCO, by choosing

$$m = \lfloor T/(2n) \rfloor - 1 \text{ and } t_k = 2nk \text{ for } k \in [m], \tag{7}$$

in Algorithm 1, the regret bound is $\mathcal{R}_i = \mathcal{O}(n^{3/2}\sqrt{T})$ for $i \in [n]$, and the communication cost is $\mathcal{O}(T)$.

Furthermore, we discover that DB-TDOCO can surpass the minimax regret $\mathcal{O}(n^{3/2}\sqrt{T})$ by selecting a different value of m and $\{t_k\}_{k \in [m]}$ for networks with specific diameters D.

Corollary 4: Let the loss functions $\{f_t^i\}_{t \in [T], i \in [n]}$ satisfy Definition 1 and 2. In DB-TDOCO, by choosing

$$m = \lfloor T/(3\mu) \rfloor - 1 \text{ and } t_k = 3\mu k \text{ for } k \in [m], \tag{8}$$

we obtain the regret $\mathcal{R}_i = \mathcal{O}(n\sqrt{DT})$ for $i \in [n]$, and the communication cost is $\mathcal{O}(nT/D)$, where μ is the BFS tree's height and D is the network's diameter.

As shown in Corollary 4, DB-TDOCO with parameters specified in (8) reduces the minimax regret by a factor of $\mathcal{O}(\sqrt{n/D})$, where the diameter D varies from 1 to $n-1$. Furthermore, we demonstrate in Appendix C, available online, in the supplementary material that the regret and communication cost of DB-TDOCO in the diameter-dependent setting is optimal.

Proof sketch of Theorem 3.

In DB-TDOCO, for each iteration ℓ, the root learner updates

$$\hat{x}_{\ell+1} = \Pi_{\mathcal{C}}(\hat{x}_{\ell-2} - \eta \nabla \hat{f}_{\ell-2}(\hat{x}_{\ell-2})),$$

where $\eta = \mathcal{O}(1/\sqrt{m})$ and $\hat{f}_\ell \triangleq \frac{1}{n\mathcal{I}} \Sigma_{s \in I_\ell} f_s$. In other words, the next model $\hat{x}_{\ell+1}$ is updated using the delayed loss function $\hat{f}_{\ell-2}$ and model $\hat{x}_{\ell-2}$ via gradient descent. By applying the regret analysis of OCOD [30], [44], we can establish that

$$\Sigma_{k=1}^{2m+2} \hat{f}_k(x_k) - \min_{x \in \mathcal{C}} \Sigma_{k=1}^{2m+2} \hat{f}_k(x) = \mathcal{O}(\sqrt{m}).$$

By multiplying the equation above by $n\mathcal{I}$, we can obtain the regret bound $\mathcal{R}_i$ for $i \in [n]$. As for the communication cost, during each interval $[t_{k-1}+1, t_k]$ for $k \in [m]$, the learners convergecast two averaged gradients by recursively aggregating their children's gradients and broadcast two models. Each run of convergecast or broadcast takes $n-1$ communication cost. Hence, the communication cost of DB-TDOCO is $\mathcal{O}(mn)$.

Comparison with existing works: By Table II, *gossip* [11], [12] and D-BOCG [2] respectively achieve state-of-the-art regret and communication complexity for adversarial DOCO with convex and Lipschitz loss functions. Under the same assumptions, DB-TDOCO with parameters in (7) significantly reduces both the worst-case regret and communication cost of *gossip* by an $\mathcal{O}(n^2)$ factor. With parameters in (8), DB-TDOCO minimizes regret in networks with small diameters D, while incurring a communication cost of $\mathcal{O}(nT/D)$. Given that *gossip* incurs a communication cost of $\Omega(nT)$ [12], DB-TDOCO reduces this cost by an $\Omega(D)$ factor. Compared to D-BOCG, which boasts the state-of-the-art communication cost ranging from $\Omega(n\sqrt{T})$ to $\mathcal{O}(n^2\sqrt{T})$, DB-TDOCO reduces its worst-case regret by an $\mathcal{O}(n^2 T^{1/4})$ factor. However, this improvement comes at the expense of an increased communication cost by a factor of

$\mathcal{O}(\sqrt{T}/n)$ and $\mathcal{O}(\sqrt{T}/D)$ when employing parameters in (7) and (8), respectively.

DB-TDOCO under milder assumptions: DB-TDOCO can broaden its scope to accommodate milder assumptions, such as unknown Lipschitz constants and unbounded feasible regions $\mathbb{R}^d$, by replacing the model update rule in (6) with gradient methods from [45], [46]. These gradient algorithms yield equivalent regret bounds to gradient descent concerning n and T, with an additional dependence on the norm of the optimal model. Leveraging these gradient algorithms allows DB-TDOCO to achieve comparable regret bounds and communication complexity in DOCO scenarios with unbounded feasible regions $\mathbb{R}^d$ and gradient norms.

The analysis above assumes the BFS tree is pre-built. If the tree is not built beforehand, it can be constructed with $O(n^2)$ communication cost in $\mathcal{O}(n)$ timeslots (e.g., using the mega-merger and flooding algorithms [47]). In this case, the learners incur $\mathcal{O}(n^2)$ extra regret by choosing models as the initialized one before building the tree. The extra communication cost and regret are negligible in DB-TDOCO when $T \gg n$.

B. Algorithm Design for the Stochastic Setting

In the stochastic setting, the well-known online-to-batch conversion scheme [48] shows that online algorithms converge favorably in batch optimization. We extend this concept in the opposite direction, utilizing batch optimization algorithms to develop DOCO algorithms. While a batch algorithm may not ensure low loss for all constituent models, it does yield a final model with low loss, suitable for DOCO tasks. Hence, we require learners to update local models as the output of a batch algorithm runs before the current time point. The key insight behind achieving low regret and communication cost stems from the rapid convergence of the batch algorithm's output to the optimal model as the feedback data increase. Consequently, frequent updates of learners' local models are unnecessary. Instead, we only require learners to train a small number of communication-efficient batch optimization algorithms, which output models at carefully selected time points.

Based on this intuition, we propose the distributed batch-to-online (DB2O) algorithm, which utilizes infrequent but effective model updates to achieve reduced regret and communication cost. Specifically, we leverage a general framework for communication-efficient distributed batch convex optimization (DBCO), namely BFS-tree-aided DBCO (T-DBCO). The learners execute m instances of T-DBCO, with each instance parallelly training models based on convergecast gradients and loss values. At time t_k, the kth instance returns a model x_{k,ν_k+1} for $k \in [m]$, and each learner i then locally updates $x_t^i = x_{k,\nu_k+1}$. We provide the model update chart of DB2O in Fig. 5 and detailed algorithmic descriptions in Algorithm 2.

We implement DB2O by employing communication-efficient T-DBCO updates with a limited number of T-DBCO instances m. Specifically, by adopting the *cutting-plane* [35] or AGD [41] update rule $\mathcal{A}$, DB2O achieves optimal regret, with polylogarithmic factors in n and T, where $m = \tilde{\mathcal{O}}(1)$. Moreover, DB2O,

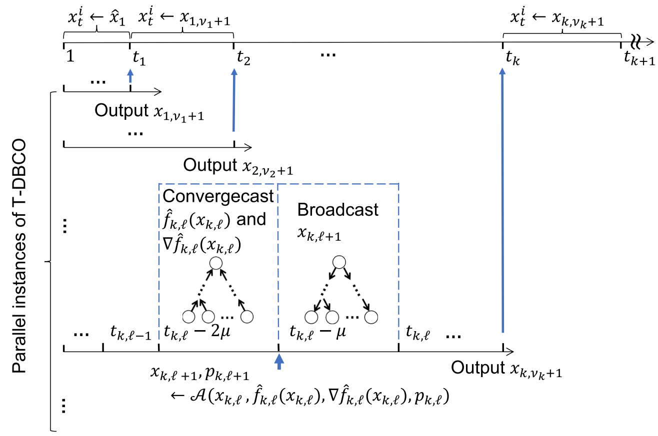


Fig. 5. Model updates in DB2O. The kth T-DBCO instance yields x_{k,ν_k+1} as learners' local models from time t_k+1 to t_{k+1}, where $k \in [m]$. $\mathcal{A}$ represents the model update rule in T-DBCO. $p_{k,\ell}$ and $\hat{f}_{k,\ell}$ denote the parameters of $\mathcal{A}$ and the average loss functions, respectively, within the ℓth mini-batch ($[t_{k,\ell-1}+1, t_{k,\ell}-2\mu]$) of the kth T-DBCO instance.

when integrated with *cutting-plane* or AGD, significantly reduces the state-of-the-art communication cost [13] by approximately a factor of $\tilde{\mathcal{O}}((nT)^\alpha)$ for $\alpha > 0$.

Theoretical advantages of DB2O.

We analyze the regret and communication cost of DB2O with *cutting-plane* and AGD in Theorems 4 and 5. The details of update and initialization rules of *cutting-plane* and AGD are deferred to Appendix A in the supplementary material, available online.

Theorem 4: Let the loss functions $f(x;\xi), \xi \sim \mathbb{P}_i$ for $i \in [n]$ satisfy Definitions 1, 4, and 5. We choose update and initialization rules $\mathcal{A}$ and $\mathcal{A}_{init}$ as in *cutting-plane*, the number of T-DBCO instances $m = 1 + \lceil \ln\ln T \rceil$, the time sequence as

$$\begin{cases} t_1 = \lceil (2\mu+2)C_1 d \ln(nT) \rceil \\ t_k = t_1 + \left\lceil (T-t_1)^{\frac{2-2^{-k+2}}{2-2^{-m+1}}} \right\rceil, \text{for } 2 \le k \le m, \end{cases}$$

and the batch size as

$$b_k = \lfloor t_k / \left(C_1\, d \ln\left(dn/\epsilon_k\right)\right) \rfloor \text{ for } k \in [m],$$

where μ is the BFS tree's height, $C_1 > 0$ is a constant, and $\epsilon_k = d^{3/2} \ln(nt_k) \sqrt{\frac{n}{t_k - (2\mu+1)C_1\, d \ln(nt_k)}}$. Then, the regret bound of DB2O is $\tilde{\mathcal{O}}(dn^2 + d^{3/2}\sqrt{nT})$, and the communication cost is $\tilde{\mathcal{O}}(dn)$, where n is the number of learners, T is the learning time, and d is the model dimension.

Theorem 5: Let the loss functions $f(x;\xi), \xi \sim \mathbb{P}_i$ for $i \in [n]$ satisfy Definitions 1, 3, and 5. We choose update and initialization rules $\mathcal{A}$ and $\mathcal{A}_{init}$ as in AGD, the number of T-DBCO instances $m = \lceil \ln\ln n \rceil + \lceil \ln\ln T \rceil$, the time sequence as

$$\begin{cases} t_k = 2\mu + n + \left\lfloor n^{1+\frac{2}{3}\cdot\frac{2^k-1}{2^{m'}-1}} \right\rfloor, & \text{for } k \le m' \\ t_k = t_{m'} + \left\lceil (T-t_{m'})^{\frac{2-2^{-k+1+m'}}{2-2^{-m+m'}}} \right\rceil, & \text{for } k > m', \end{cases}$$

Algorithm 2: Distributed Batch-to-Online (DB2O).

Input: Update rule $\mathcal{A}$, initialization rule $\mathcal{A}_{init}$ for $\mathcal{A}$, BFS tree $\mathcal{T}$, number of T-DBCO instances m, time sequence $0 = t_0 < \ldots < t_{m+1} = T$, and batch sizes $\{b_k\}_{k\in[m]}$

$\triangleright$ Details of $\mathcal{A}$ and $\mathcal{A}_{init}$ are deferred to Appendix A, available online

Initialization:

Each learner i initializes:

$$x_t^i = \hat{x}_1 \leftarrow \arg\min_{x\in\mathcal{C}} \|x\| \text{ for } t \in [1, t_1]$$

Each learner sets μ as the height of $\mathcal{T}$

Each learner sets, for $k \in [m]$,

$$\begin{cases} x_{k,1} \leftarrow \hat{x}_1 \\ t_{k,\ell} \leftarrow \ell \cdot b_k \text{ for } \ell \le \nu_k = \lfloor t_k/b_k \rfloor \\ p_{k,1} \leftarrow \mathcal{A}_{init}(\nu_k) \end{cases}$$

for each k **in** $[m]$ **do**

T-DBCO Update:

for $\ell = 1$ **to** ν_k **do**

Convergecast:

At time $t_{k,\ell} - 2\mu$, the learners run:

$$\begin{cases} \text{Convergecast}(\{\hat{f}_{k,\ell}^i(x_{k,\ell})\}_{i\in[n]}, \mathcal{T}) \\ \text{Convergecast}(\{\nabla\hat{f}_{k,\ell}^i(x_{k,\ell})\}_{i\in[n]}, \mathcal{T}), \end{cases}$$

where $\hat{f}_{k,\ell}^i = \frac{1}{b_k - 2\mu}\Sigma_{s=t_{k,\ell-1}+1}^{t_{k,\ell}-2\mu} f_s^i$

At time $t_{k,\ell} - \mu$, the root learner obtains:

$$\begin{cases} \hat{f}_{k,\ell}(x_{k,\ell}) = \Sigma_{i=1}^n \hat{f}_{k,\ell}^i(x_{k,\ell}) \\ \nabla\hat{f}_{k,\ell}(x_{k,\ell}) = \Sigma_{i=1}^n \nabla\hat{f}_{k,\ell}^i(x_{k,\ell}) \end{cases}$$

Global Update:

At time $t_{k,\ell} - \mu$, the root learner in $\mathcal{T}$ updates:

$$x_{k,\ell+1}, p_{k,\ell+1} \leftarrow \mathcal{A}(x_{k,\ell}, \hat{f}_{k,\ell}(x_{k,\ell}), \nabla\hat{f}_{k,\ell}(x_{k,\ell}), p_{k,\ell}).$$

Broadcast:

At time $t_{k,\ell} - \mu$, the learners run:

$$\text{Broadcast}(x_{k,\ell+1}, \mathcal{T})$$

end for

Local Update:

At time $t_k + 1$, each learner i updates:

$$x_t^i \leftarrow x_{k,\nu_k+1} \text{ for } t \in [t_k + 1, t_{k+1}]$$

end for

and the batch sizes as

$$\begin{cases} b_k = 2\mu + n, & \text{for } k \le m' \\ b_k = \max\left\{2\mu + n, \left\lfloor \frac{t_k}{C_2 n^{1/4} t_k^{1/4} + 1} \right\rfloor\right\}, & \text{for } k > m', \end{cases}$$

where μ is the BFS tree's height, $m' = \lceil \ln\ln n \rceil$, and $C_2 > 0$ is a constant. Then, the regret bound of DB2O is $\tilde{\mathcal{O}}(n^2 + \sqrt{nT})$, and the communication cost is $\tilde{\mathcal{O}}(n^{5/3} + n^{5/4}T^{1/4})$.

DB2O facilitates learners in achieving nearly optimal regret by parallelizing a few instances of communication-efficient T-DBCO. By Theorem 4, DB2O with *cutting-plane* almost attains the optimal regret and communication complexity in n and T (the primary focus of this paper) established in Corollary 2 when $T \gg n$. Notably, it stands as the first algorithm to achieve almost optimal regret and communication complexity that is nearly independent of T. However, its regret and communication complexity do additionally depend on the model dimension d, which proves to be suboptimal when compared to the state-of-the-art DMA algorithm with no dependence on d [13], cf. Table I. In contrast, DB2O with AGD nearly achieves optimal regret in n, T, and d. While its communication cost is not optimal in n and T, it remains independent of d and outperforms the state-of-the-art [13], offering a potential advantage in high-dimensional DOCO tasks.

Proof sketch of Theorems 4 and 5.

For DB2O with *cutting-plane*, denote the risk of model $x \in \mathcal{C}$ as $\text{Risk}(x) \triangleq \bar{f}(x) - \min_{x^*\in\mathcal{C}} \bar{f}(x^*)$, where $\bar{f}(x) = \Sigma_{i=1}^n \mathbb{E}_{\xi\sim\mathbb{P}_i}[f(x;\xi)]$. By (2), we obtain, for $j \in [n]$

$$\begin{aligned} \bar{\mathcal{R}}_j(T) &= \mathbb{E}\left[\text{Risk}(\hat{x}_1)t_1\right] + \mathbb{E}\left[\Sigma_{k=1}^m \text{Risk}(x_{k,\nu_k+1})(t_{k+1} - t_k)\right] \\ &= \mathcal{O}(nt_1) + \mathbb{E}\left[\Sigma_{k=1}^m \text{Risk}(x_{k,\nu_k+1})(t_{k+1} - t_k)\right], \quad (9) \end{aligned}$$

where the second equality follows from $\text{Risk}(\hat{x}_1) = \mathcal{O}(n)$ according to Definition 5. For *cutting-plane*, b_k specified in Theorem 4 ensures that

$$\mathbb{E}\left[\text{Risk}(x_{k,\nu_k+1})\right] \le \mathcal{O}(1) \cdot \epsilon_k.$$

Plugging the values of t_k and the risk of x_{k,ν_k+1} for $k \in [m]$ into (9), we obtain that $\bar{\mathcal{R}}_j(T) = \tilde{\mathcal{O}}(dn^2 + d^{3/2}\sqrt{nT})$.

Each run of T-DBCO requires $\mathcal{O}(t_k/b_k)$ convergecast and broadcast operations and the communication cost of convergecast/broadcast is $\mathcal{O}(n)$. The communication cost for each run of T-DBCO in Algorithm 2 equals $\mathcal{O}(nt_k/b_k) = \tilde{\mathcal{O}}(dn)$ and Algorithm 2 runs $\mathcal{O}(\ln\ln T)$ instances of T-DBCO. Thus the overall communication cost is $\tilde{\mathcal{O}}(dn)$.

In DB2O with AGD, b_k specified in Theorem 5 ensures that

$$\begin{cases} \mathbb{E}\left[\text{Risk}(x_{k,\nu_k+1})\right] = \mathcal{O}\left(\frac{n^3}{(t_k - 2\mu - n)^2}\right) & \text{if } t_k \le 2\mu + n + n^{5/3} \\ \mathbb{E}\left[\text{Risk}(x_{k,\nu_k+1})\right] = \mathcal{O}\left(\sqrt{\frac{n}{t_k - 2\mu - n}}\right), & \text{otherwise.} \end{cases}$$

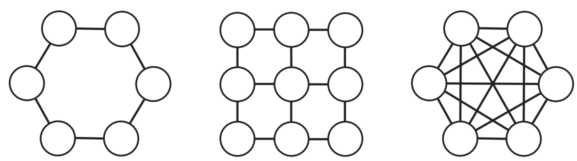
The rest of the proof is similar to that for *cutting-plane*.

Comparison with existing works: DB2O significantly reduces the communication complexity of the state-of-the-art DMA algorithm [13] concerning n and T. First, DB2O with *cutting-plane* reduces DMA's communication complexity by a factor of $\tilde{\mathcal{O}}(\sqrt{nT}/d)$ while maintaining minimax regret within polylogarithmic factors in n and T and factors in $d^{3/2}$. This advantage is particularly evident in scenarios with numerous learners and prolonged learning times. Compared to DMA, DB2O with *cutting-plane* requires bounded loss functions and a fat feasible region, relaxing the smoothness assumption (cf. Table II). Second, DB2O with AGD nearly achieves minimax regret and reduces

TABLE III
REGRET BOUNDS AND COMMUNICATION COMPLEXITY FOR DOCO ALGORITHMS ON n-NODE CYCLE, GRID, AND CLIQUE NETWORKS FOR THE DEPENDENCE OF THE NUMBER OF LEARNERS n AND LEARNING TIME T WHEN $T \gg n$

Feedbacks	DOCO algorithms	Regret bounds			Communication complexity		
		Cycle	Grid	Clique	Cycle	Grid	Clique
Adversarial	*gossip* [11], [12]	$\mathcal{O}(n^{7/2}\sqrt{T})$	$\tilde{\mathcal{O}}(n^{5/2}\sqrt{T})$	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(nT)$	$\mathcal{O}(nT)$	$\mathcal{O}(n^2T)$
	D-BOCG [2]	$\mathcal{O}(n^{7/2}T^{3/4})$	$\tilde{\mathcal{O}}(n^{5/2}T^{3/4})$	$\mathcal{O}(n^{3/2}T^{3/4})$	$\mathcal{O}(n\sqrt{T})$	$\mathcal{O}(n\sqrt{T})$	$\mathcal{O}(n^2\sqrt{T})$
	DB-TDOCO (topology-independent)	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(T)$	$\mathcal{O}(T)$	$\mathcal{O}(T)$
	DB-TDOCO (diameter-dependent)	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(n^{5/4}\sqrt{T})$	$\mathcal{O}(n\sqrt{T})$	$\mathcal{O}(T)$	$\mathcal{O}(\sqrt{n}T)$	$\mathcal{O}(nT)$
Stochastic	DMA [13]	$\mathcal{O}(\sqrt{nT})$	$\mathcal{O}(\sqrt{nT})$	$\mathcal{O}(\sqrt{nT})$	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(n^{3/2}\sqrt{T})$	$\mathcal{O}(n^{3/2}\sqrt{T})$
	$\mathbf{DB2O}_c$	$\tilde{\mathcal{O}}(\sqrt{nT})$	$\tilde{\mathcal{O}}(\sqrt{nT})$	$\tilde{\mathcal{O}}(\sqrt{nT})$	$\tilde{\mathcal{O}}(n)$	$\tilde{\mathcal{O}}(n)$	$\tilde{\mathcal{O}}(n)$
	$\mathbf{DB2O}_a$	$\tilde{\mathcal{O}}(\sqrt{nT})$	$\tilde{\mathcal{O}}(\sqrt{nT})$	$\tilde{\mathcal{O}}(\sqrt{nT})$	$\tilde{\mathcal{O}}(n^{5/4}T^{1/4})$	$\tilde{\mathcal{O}}(n^{5/4}T^{1/4})$	$\tilde{\mathcal{O}}(n^{5/4}T^{1/4})$

Topology-independent and diameter-dependent DB-TDOCO variants are implemented with parameters in Eqs. (7) and (8), respectively. DB2Oc and DB2Oa refer to DB2O with cutting-plane and AGD, respectively.



(a) Cycle network (b) Grid network (c) Clique network

Fig. 6. Illustration of some typical network topologies.

TABLE IV
CONNECTIVITY PARAMETER Γ FOR *GOSSIP* [11], [12] IN ADVERSARIAL DOCO IN n-NODE CYCLE, GRID, AND CLIQUE NETWORKS

Network topology	Connectivity parameter Γ
Cycle	$\mathcal{O}(n^2)$
Grid	$\mathcal{O}(n\ln n)$
Clique	$\mathcal{O}(1)$

DMA's communication complexity by a factor of $\tilde{\mathcal{O}}((nT)^{1/4})$ for loss functions under the same assumptions. This reduction is dimension-free, making DB2O with AGD more advantageous over DMA in high-dimensional tasks compared to DB2O with *cutting-plane*.

DB2O under milder assumptions: For DB2O with *cutting-plane*, the strict requirements on loss functions outlined in Definitions 1, 4, and 5 can be relaxed without compromising effectiveness. This relaxation involves ensuring that the expected loss function $\bar{f}(x) = \Sigma_{i=1}^{n}\mathbb{E}_{\xi\sim\mathbb{P}_i}[f(x;\xi)]$ adheres to Definitions 1 and 4, while bounding the variances of both the loss values and gradients of $f(x;\xi), \xi \sim \mathbb{P}_i$ for $i \in [n]$ [35]. Consequently, the regret and communication complexity of the corresponding DB2O implementation remain unaffected. Similarly, for DB2O with AGD, the prerequisites on feasible regions to be compact can be relaxed, enabling it to operate in scenarios with an unbounded feasible region $\mathbb{R}^d$ [41]. This relaxation aligns with the analysis in Theorem 5 concerning n and T, allowing DB2O with AGD to maintain the same regret and communication complexity under these relaxed conditions.

Similar to the adversarial setting, if the BFS tree is not built beforehand, it will incur $O(n^2)$ extra communication cost and regret loss when building the tree and using the initialized model. In this case, the communication cost of DB2O with *cutting-plane* becomes $\tilde{O}(n^2 + dn)$, which is still the lowest in terms of n and T in the literature.

VI. ALGORITHM ANALYSIS ON TYPICAL NETWORKS

In this section, we compare the regret loss and communication cost of our algorithms with the state-of-the-art [2], [11], [12], [13] on typical cycle, grid, and clique networks. We present the networks in Fig. 6 and summarize the comparisons of regret bounds and communication complexity in Table III. It shows that our proposed algorithms achieve reduced communication cost in n and T without compromising regret on these networks compared to algorithms with state-of-the-art regret [11], [12], [13].

A. DOCO for the Adversarial Setting

In the adversarial setting, we assess the performance of DB-TDOCO against *gossip* [11], [12] and D-BOCG [2], which respectively achieve the state-of-the-art regret and communication complexity. The impact of network topologies differs between *gossip* (D-BOCG) and DB-TDOCO. According to the analysis in [2], [11], [22], *gossip* and D-BOCG's regret and communication complexity are influenced by network connectivity, measured by the spectral gap of the network's gossip matrix [40] and the number of edges. However, DB-TDOCO's performance remains unaffected by the network topology when parameters in (7) are chosen. Conversely, with parameters in (8), DB-TDOCO's performance is governed by the network's diameter.

The regret bound of *gossip* scales as $\mathcal{O}(\Gamma n^{3/2}\sqrt{T})$, where Γ measures the connectivity of the network [22]. Specifically, in *gossip*, the network topology is parameterized by its gossip matrix A [40], a weighted adjacency matrix of the learner network. The parameter Γ is the inverse of A's spectral gap, which equals $\frac{1}{1-\lambda_2(A)}$, where $\lambda_2(A)$ denotes the second largest eigenvalue of A. Values of Γ for n-node cycle, grid, and clique networks are presented in Table IV [22]. The communication cost of *gossip* is linear in the network's edge number E and learning time T. Note that $E = \mathcal{O}(n)$ for cycle and grid networks and $E = \mathcal{O}(n^2)$ for the clique network. The regret bounds and communication complexity of *gossip* in Table III then follow.

D-BOCG reduces *gossip*'s communication cost by a factor of $\mathcal{O}(\sqrt{T})$ but increases the regret by a factor of $\mathcal{O}(T^{1/4})$.

DB-TDOCO's regret bounds and communication complexity are established in Corollaries 3 and 4. When using parameters in (7), DB-TDOCO's performance is independent of the network topology. With parameters in (8), DB-TDOCO's performance depends on the network's diameter D. Note that $D = \mathcal{O}(n)$ for the cycle network, $D = \mathcal{O}(\sqrt{n})$ for the grid network, and $D = \mathcal{O}(1)$ for the clique network. The regret bounds and communication complexity of DB-TDOCO in this case then follow from Corollary 4.

Table III shows that both variants of DB-TDOCO require lower communication cost than *gossip*. DB-TDOCO with parameters in (7) reduces the communication cost proportionally to the network's edge number. With parameters from (8), DB-TDOCO reduces the communication cost by a factor of $\mathcal{O}(\frac{E \cdot D}{n})$, where n is the number of learners, E is the edge number of the network, and D is the diameter of the network. Regarding regret, DB-TDOCO with (7) matches *gossip*'s regret on the clique network. In other settings, DB-TDOCO reduces *gossip*'s regret by a factor of $\mathcal{O}(n^{\alpha})$, where α ranges from 0.5 to 2. Compared to D-BOCG, DB-TDOCO reduces its regret by a factor ranging from $\mathcal{O}(T^{1/4})$ to $\mathcal{O}(n^2 T^{1/4})$, with the communication cost enlarged by a factor of $\mathcal{O}(\sqrt{T}/n^{\alpha})$ for $0.5 \leq \alpha \leq 2$.

B. DOCO for the Stochastic Setting

The DOCO algorithms in the stochastic setting are not affected by the networks' topologies. The regret bounds and communication complexity of our DB2O and DMA algorithms remain unchanged on all three networks in Fig. 6. As summarized in Table III, DB2O reduces the communication complexity of DMA by an $\tilde{\mathcal{O}}((nT)^{\alpha})$ factor in terms of the number of learners n and learning time T, where $\alpha = 0.5$ for DB2O with *cutting-plane* and $\alpha = 0.25$ for DB2O with AGD. These results demonstrate that the advantage of our DB2O algorithms is consistent and significant regardless of the network topologies.

VII. Experiments

In this section, we present experimental results comparing the communication efficiency of our algorithms and the state-of-the-art in achieving comparable classification accuracy.

A. Implementation Details

We conduct distributed online logistic regression tasks. Each learner's error rate at each time is evaluated on the current data. We average all learners' error rates across the learning time as the algorithm's online classification error rate[2].

Data and preprocessing: We utilize two real-world datasets from the LIBSVM repository [49]: `covtype.binary` (`covtype` for short) [23] and `epsilon` [24]. The `covtype` dataset contains 581,012 samples with 54 features, while `epsilon` is a high-dimensional dataset with 2,000 features and 500,000 samples. We preprocess each dataset by scaling all features to $[-1, 1]$ and each sample to unit length. We assign positive samples to half of the learners and negative samples to the other half in uniform size.[3] For the stochastic settings, we randomly order each learner's dataset and reveal the tth sample to them at each time t. The default learning time equals each learner's data size. In the adversarial setting, we generate the data streams inspired by [50] as follows: 1) we divide the learning time into three equal-length phases; 2) we reveal one sample with its label flipped to each learner at each time during the second phase, while revealing the original samples during the first and third phases. With this construction, the optimal model varies in different phases.

Baselines: We compare our algorithms with DMA [13] and the mini-batch *gossip* algorithm [11].[4] These baselines achieve state-of-the-art regret loss with the currently state-of-the-art communication cost. Additionally, we include D-BOCG [2] as a baseline for the adversarial setting, which achieves the currently state-of-the-art communication complexity with a suboptimal regret bound.

Implementations: We conduct experiments on networks with representative topologies, including the cycle network where each learner has two neighbors and the all-connected clique network. Based on [13], [22] and our analysis in Section V, the evaluated algorithms achieve the best theoretical regret bounds on the clique network with high communication cost. They feature higher regret bounds on the loosely connected cycle network with lower communication cost. We choose the feasible region $\mathcal{C}$ as an Euclidean ball with a radius of 20. Similar to [2], we set the stepsize parameter as $cT^{-3/4}$ for D-BOCG, where c is chosen optimally from $\{10^{-2}, 10^{-1}, \ldots, 10^{5}\}$ in each run. For DB-TDOCO and *gossip* [11], we take the stepsize $cT^{-1/2}$ with the optimal c in $\{10^{-2}, 10^{-1}, \ldots, 10^{5}\}$. For DB2O with AGD and *cutting-plane* (referred to as DB2O$_a$ and DB2O$_c$) and DMA, we use the hyperparameters as specified in [13], [41], [51]. To adjust the communication budget of *gossip*, D-BOCG, and DMA, we modify the batch size used to compute the mini-batch gradients. For DB-TDOCO, we adjust the communication budget by modifying the interval size $t_1 = t_2 - t_1 = \ldots = t_m - t_{m-1}$ in Algorithm 1. Similarly, for DB2O$_c$ and DB2O$_a$, we tune the communication budget by modifying the constants C_1 and C_2 in Theorems 4 and 5.

B. Experimental Results

DOCO for the adversarial setting: We investigate how the error rates evolve with the communication budget on 8-node and 32-node learner networks in the adversarial setting. Fig. 7(a)–(d) demonstrate that DB-TDOCO requires only around 5% of the communication cost of *gossip* and D-BOCG to achieve a target error rate of ≤ 0.4 on all networks and datasets. Furthermore, on the 32-node cycle network, DB-TDOCO converges to lower error rates than *gossip* and D-BOCG as the budget increases.

We further analyze the evolution of the communication cost of the algorithms required to achieve a low target error rate

[2]The code is available at https://github.com/GGBOND121382/Communication-Efficient_Regret-Optimal_DOCO

[3]We defer the experiments where all learners' data are i.i.d. stochastic to Appendix B-C in the supplementary material, available online.

[4]We modify *gossip* [11] into a mini-batch version, where the learners communicate and update models every τ timeslots using their averaged gradients, and τ represents the batch size.

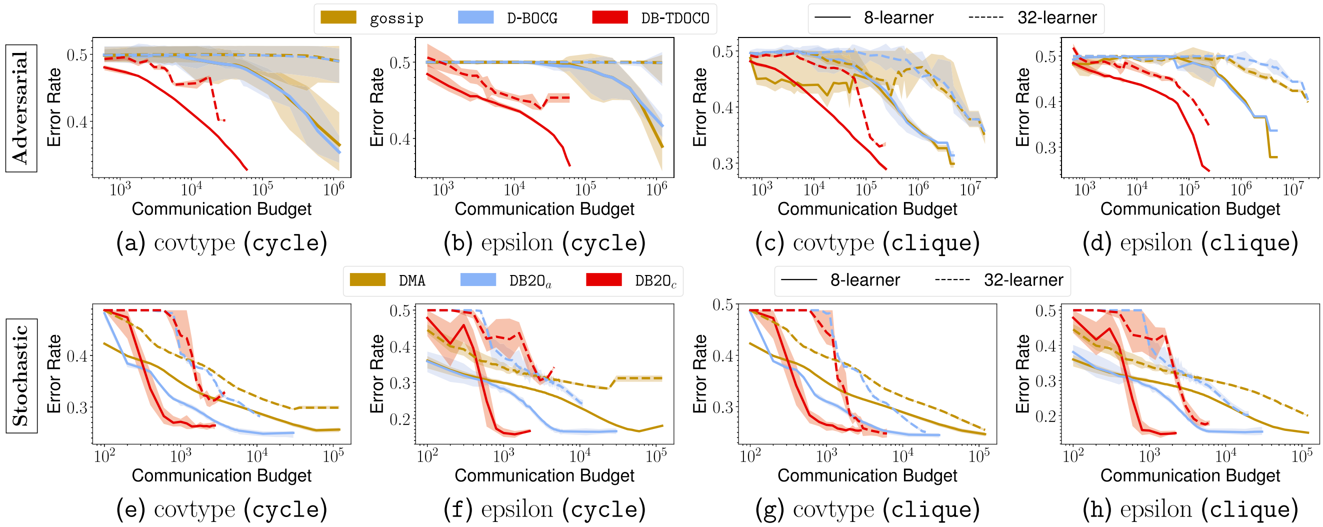


Fig. 7. Online classification error rates in DOCO. The line indicates the averages, and the shaded area is where all error rates lie in across 10 runs.

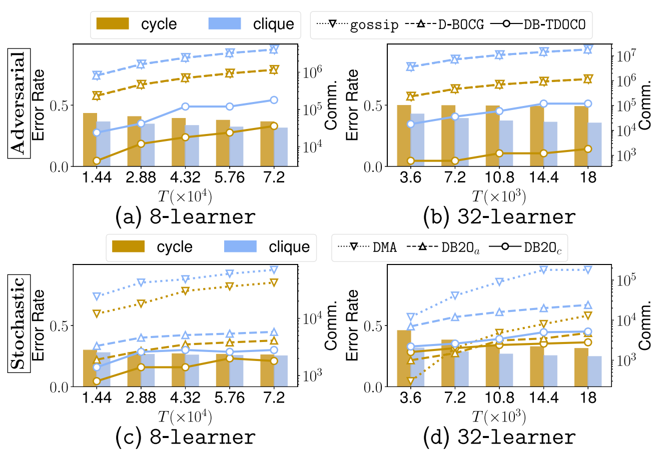


Fig. 8. Communication cost to achieve the target error rates. The curves are the minimum communication cost to achieve the error rates specified by the bars. The target error rate is the maximum of the evaluated algorithms' optimal error rates for each T and network.

with respect to the learning time T. For this analysis, we utilize the covtype dataset. In Fig. 8(a), we observe that on 8-node networks, DB-TDOCO incurs less than 5% of the communication cost of *gossip* (D-BOCG) across various values of T and network topologies. Similarly, on 32-node networks, Fig. 8(b) illustrates that DB-TDOCO necessitates less than 1% of the communication cost incurred by both *gossip* and D-BOCG.

DOCO for the stochastic setting: Fig. 7(e)–(h) illustrate that on 8-node networks, DB2O$_a$ and DB2O$_c$ require only around 10% and 1% of the communication cost of DMA to converge to close and steady error rates. On 32-node networks, DB2O$_a$ and DB2O$_c$ generally need approximately 50% and 10% of the communication cost of DMA to converge to a low error rate ≤ 0.3. The advantages of DB2O$_a$ and DB2O$_c$ are more prominent on 8-node networks since the learning time T is significantly larger than the number of learners n (cf. Theorems 4 and 5). On the 32-node cycle network and epsilon dataset, DB2O$_c$ produces higher error rates than DB2O$_a$ and DMA because T is not large enough to mitigate the impact of the high dimension d.

Consequently, we investigate the necessary communication cost of these algorithms to achieve a low target error rate, considering different T on various networks using covtype. Fig. 8(c) illustrates that on 8-node networks, DB2O$_a$ and DB2O$_c$ require approximately 10% and 5% of the communication cost of DMA, respectively. On the 32-node clique network, Fig. 8(d) shows that the communication costs of DB2O$_a$ and DB2O$_c$ consistently remain lower than DMA. Their communication savings peak at $T = 18,000$, surpassing 80% and 97%, respectively. However, on the 32-node cycle network, DB2O$_a$ and DB2O$_c$ fail to save on communication over DMA for $T \leq 7,200$ as the update times are not sufficient to mitigate the impact of high dimension d and learner count n. Nevertheless, as T reaches 18,000, DB2O$_a$ and DB2O$_c$ achieve over 60% and 80% savings compared to DMA.

VIII. Conclusion

DOCO provides effective algorithmic frameworks for learning tasks with numerous learners and streaming data. Two significant performance bottlenecks for DOCO are regret loss and communication complexity when implemented in communication-constrained networks. It is challenging to simultaneously achieve low regret and communication complexity, especially when the number of learners n and learning time T are extensive. In this paper, we design novel algorithms in typical adversarial and stochastic settings. Our algorithms nearly achieve the minimax regret and reduce the state-of-the-art algorithms' communication cost by a factor of $\mathcal{O}(n^2)$ and $\tilde{\mathcal{O}}(\sqrt{nT})$ in adversarial and stochastic settings, respectively. Furthermore, we prove that the communication complexity of our algorithms is nearly optimal. Extensive experiments validate that our proposed algorithms can achieve $90\% \sim 99\%$ communication saving over the state-of-the-art with close accuracy in most cases.

REFERENCES

[1] S. Lin, M. Dedeoglu, and J. Zhang, "Accelerating distributed online meta-learning via multi-agent collaboration under limited communication," in *Proc. 22nd Int. Symp. Theory Algorithmic Found. Protocol Des. Mobile Netw. Mobile Comput.*, 2021, pp. 261–270.

[2] Y. Wan, W. Tu, and L. Zhang, "Projection-free distributed online convex optimization with $O(\sqrt{T})$ communication complexity," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 9818–9828.

[3] S. Hosseinalipour, A. Nayak, and H. Dai, "Power-aware allocation of graph jobs in geo-distributed cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 749–765, Apr. 2020.

[4] Z. Tu, X. Wang, Y. Hong, L. Wang, D. Yuan, and G. Shi, "Distributed online convex optimization with compressed communication," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, 2022, Art. no. 2499.

[5] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Approximate dynamic programming for communication-constrained sensor network management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, Aug. 2007.

[6] J. Xu, K. Li, and G. Min, "Reliable and energy-efficient multipath communications in underwater sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 7, pp. 1326–1335, Jul. 2012.

[7] K. Yuan, Q. Ling, and Z. Tian, "Communication-efficient decentralized event monitoring in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2198–2207, Aug. 2015.

[8] A. Ahmad, G. Lawless, and P. U. Lima, "An online scalable approach to unified multirobot cooperative localization and object tracking," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1184–1199, Oct. 2017.

[9] G. Zhan and W. Shi, "LOBOT: Low-cost, self-contained localization of small-sized ground robotic vehicles," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 744–753, Apr. 2013.

[10] Á. Serra-Gómez, B. Brito, H. Zhu, J. J. Chung, and J. Alonso-Mora, "With whom to communicate: Learning efficient communication for multi-robot collision avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 11770–11776.

[11] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. A. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2483–2493, Nov. 2013.

[12] J. Lei, P. Yi, Y. Hong, J. Chen, and G. Shi, "Online convex optimization over erdos-renyi random networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 15591–15601.

[13] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *J. Mach. Learn. Res.*, vol. 13, pp. 165–202, 2012.

[14] J. Acharya, C. D. Sa, D. J. Foster, and K. Sridharan, "Distributed learning with sublinear communication," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 40–50.

[15] B. Sidik, R. Puzis, P. Zilberman, and Y. Elovici, "PALE: Time bounded practical agile leader election," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 470–485, Feb. 2020.

[16] G. Best, M. Forrai, R. R. Mettu, and R. Fitch, "Planning-aware communication for decentralised multi-robot coordination," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1050–1057.

[17] N. Santoro, *Design and Analysis of Distributed Algorithms. Wiley Series on Parallel and Distributed Computing*, Hoboken, NJ, USA: Wiley, 2007.

[18] Z. Qin, D. Wu, Z. Xiao, B. Fu, and Z. Qin, "Modeling and analysis of data aggregation from convergecast in mobile sensor networks for industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4457–4467, Oct. 2018.

[19] J. Bassil, A. Makhoul, B. Piranda, and J. Bourgeois, "Distributed size-constrained clustering algorithm for modular robot-based programmable matter," *ACM Trans. Auton. Adapt. Syst.*, vol. 18, no. 1, pp. 1:1–1:21, 2023.

[20] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 5451–5452.

[21] D. Yuan, Y. Hong, D. W. C. Ho, and S. Xu, "Distributed mirror descent for online composite optimization," *IEEE Trans. Autom. Control.*, vol. 66, no. 2, pp. 714–729, Feb. 2021.

[22] A. Nedic, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.

[23] R. Collobert, S. Bengio, and Y. Bengio, "A parallel mixture of SVMs for very large scale problems," *Neural Comput.*, vol. 14, no. 5, pp. 1105–1114, 2002.

[24] G. Yuan, C. Ho, and C. Lin, "An improved GLMNET for L1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 13, pp. 1999–2030, 2012.

[25] Y. Wang, J. Hu, X. Chen, and L. Wang, "Distributed bandit learning: Near-optimal regret with efficient communication," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–31.

[26] X. Wang, Z. Ning, L. Guo, S. Guo, X. Gao, and G. Wang, "Online learning for distributed computation offloading in wireless powered mobile edge computing networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 8, pp. 1841–1855, Aug. 2022.

[27] X. Su, Y. Zhou, L. Cui, and J. Liu, "On model transmission strategies in federated learning with lossy communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1173–1185, Apr. 2023.

[28] D. van der Hoeven, H. Hadiji, and T. van Erven, "Distributed online learning for joint regret with communication constraints," in *Proc. Int. Conf. Algorithmic Learn. Theory*, PMLR, 2022, pp. 1003–1042.

[29] Y. Wan, G. Wang, W.-W. Tu, and L. Zhang, "Projection-free distributed online learning with sublinear communication complexity," 2022, *arXiv:2103.11102v2*.

[30] P. Joulani, A. György, and C. Szepesvári, "Online learning under delayed feedback," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1453–1461.

[31] I. Bistritz, Z. Zhou, X. Chen, N. Bambos, and J. H. Blanchet, "Online EXP3 learning in adversarial bandits with delayed feedback," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11345–11354.

[32] H. Esfandiari, A. Karbasi, A. Mehrabian, and V. S. Mirrokni, "Regret bounds for batched bandits," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7340–7348.

[33] Z. Gao, Y. Han, Z. Ren, and Z. Zhou, "Batched multi-armed bandits problem," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 501–511.

[34] V. Perchet, P. Rigollet, S. Chassang, and E. Snowberg, "Batched bandit problems," in *Proc. 28th Conf. Learn. Theory*, 2015, p. 1456.

[35] I. Usmanova, M. Kamgarpour, A. Krause, and K. Y. Levy, "Fast projection onto convex smooth constraints," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 10476–10486.

[36] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3/4, pp. 231–357, 2015.

[37] D. van der Hoeven, "Exploiting the surrogate gap in online multiclass classification," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, Art. no. 802.

[38] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5149–5164, Oct. 2015.

[39] C. Li, P. Zhou, L. Xiong, Q. Wang, and T. Wang, "Differentially private distributed online learning," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1440–1453, Aug. 2018.

[40] A. Koloskova, S. U. Stich, and M. Jaggi, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 3478–3487.

[41] C. Hu, J. T. Kwok, and W. Pan, "Accelerated gradient methods for stochastic optimization and online learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Curran Associates, Inc., 2009, pp. 781–789.

[42] Z. Lei, X. Ye, Y. Wang, D. Li, and J. Xu, "Efficient online model adaptation by incremental simplex tableau," in *Proc. AAAI Conf. Artif. Intell.*, AAAI Press, 2017, pp. 2161–2167.

[43] A. Destounis, D. Tsilimantos, M. Debbah, and G. S. Paschos, "Learn2MAC: Online learning multiple access for URLLC applications," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2019, pp. 1–6.

[44] S. Bubeck, "Introduction to online optimization," Lecture Notes, vol. 2, pp. 1–86, 2011.

[45] A. Jacobsen and A. Cutkosky, "Unconstrained online learning with unbounded losses," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2023, pp. 14590–14630.

[46] A. Cutkosky, "Artificial constraints and hints for unbounded online learning," in *Proc. Conf. Learn. Theory*, PMLR, 2019, pp. 874–894.

[47] B. Awerbuch, "Optimal distributed algorithms for minimum weight spanning tree, counting, leader election and related problems (detailed summary)," in *Proc. 19th Annu. ACM Symp. Theory Comput.*, 1987, pp. 230–240.

[48] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. dissertation, Dept. Comput. Sci., The Hebrew Univ. Jerusalem, Jerusalem, Israel, Jul. 2007, pp. 1–162.

[49] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, 2011.

[50] G. Wang, D. Zhao, and L. Zhang, "Minimizing adaptive regret with one gradient per iteration," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 2762–2768.

[51] K. M. Anstreicher, "Towards a practical volumetric cutting plane method for convex programming," *SIAM J. Optim.*, vol. 9, no. 1, pp. 190–206, 1998.

Jiandong Liu received the BS degree in information security from Xidian University, Xi'an, China, in 2018, and the MS degree in cybersecurity from the University of Science and Technology of China, Hefei, China, in 2021. He is currently working toward the PhD degree in cybersecurity with the School of Cybersecurity, University of Science and Technology of China, Hefei, China. His research interests include online learning and privacy-preserving computation.

Lan Zhang received the bachelor's and PhD degrees from Tsinghua University, China. She is currently a professor with the School of Computer Science and Technology, University of Science and Technology of China. Her research interests include mobile computing, privacy protection, and data sharing and trading.

Fengxiang He is a lecturer with Artificial Intelligence and its Applications Institute, School of Informatics, University of Edinburgh. His research interest is in trustworthy AI, particularly deep learning theory and explainability, theory of decentralised learning, privacy in machine learning, symmetry in machine learning, learning theory in game-theoretical problems, and their applications in economics. He is an area chair of ICML, NeurIPS, UAI, AISTATS, and ACML, and an associate editor of *IEEE Transactions on Technology and Society*.

Chi Zhang received the BEng degree in computer science and technology from the University of Science and Technology of China (USTC) with the honor of The Talent Program in Computer and Information Science and Technology, in 2017, and the PhD degree in computer science and technology from USTC, in 2023. His main research interest is data center networking, edge computing and algorithms. Many of his works have been published in top conferences and journals, such as IEEE INFOCOM, ACM MobiHoc, and *IEEE/ACM Transactions on Networking*.

Shanyang Jiang received the BS degree in Internet of Things engineering from the Hefei University of Technology, Hefei, China, in 2018, and the MS degree from the School of Data Science, University of Science and Technology of China, Hefei, China, in 2021. He is currently working toward the PhD degree with the School of Data Science, University of Science and Technology of China, Hefei, China. His research intersets include crowdsourcing and mechanism design.

Xiang-Yang Li (Fellow, IEEE) received the bachelor's degree from the Department of Computer Science, Tsinghua University, China, in 1995, and the MS and PhD degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, in 2000 and 2001, respectively. He is a professor, the executive dean of the College of Information Science and Artificial Intelligence, the executive dean of the School of Computer Science and Technology, University of Science and Technology of China. He is an ACM fellow (2019), an ACM distinguished scientist (2014). He was a full professor with Computer Science Department of IIT. He served as a co-chair of ACM China Council and editors of several premium journals. His research interests include Artificial Intelligence of Things (AIoT), privacy and security of AIOT, and data sharing and trading.